

T.C.

ALTINBAŞ ÜNİVERSİTESİ

Lisansüstü Eğitim Enstitüsü

Elektrik ve Bilgisayar Mühendisliği

SAWUML-UML TABANLI YAZILIM MİMARİSİ TANIMLAMA DİLİ VE ARAÇ KÜMESİ

Mehmet Alp KÖSE

Doktora Tezi

Danışman

Doç. Dr. Mert ÖZKAYA

İstanbul, 2022

SAWUML-UML TABANLI YAZILIM MİMARİSİ TANIMLAMA DİLİ VE ARAÇ KÜMESİ

Mehmet Alp KÖSE

Elektrik ve Bilgisayar Mühendisliği

Doktora Tezi

ALTINBAŞ ÜNİVERSİTESİ

2022

"Mehmet Alp KÖSE" tarafından hazırlanmış ve 13/5/2022 tarihinde sunulmuş "SAWUML-UML TABANLI YAZILIM MİMARİSİ TANIMLAMA DİLİ ve ARAÇ KÜMESİ" başlıklı tez Elektrik ve Bilgisayar Mühendisliği Anabilim Dalında Doktora Tezi olarak **oy birliği** ile kabul edilmiştir.

_____

Doç. Dr. Mert ÖZKAYA

Danışman

Tez Savunma Sınavı Jüri Üyeleri:

Doç. Dr. Mert ÖZKAYA	Mühendislik Fakültesi, Yeditepe Üniversitesi	_____
Dr. Öğr. Üyesi Mesut ÇEVİK	Mühendislik ve Mimarlık Fakültesi, Altınbaş Üniversitesi	_____
Dr. Öğr. Üyesi Abdullahi Abdu İBRAHİM	Mühendislik ve Mimarlık Fakültesi, Altınbaş Üniversitesi	_____
Dr. Öğr. Üyesi Taymaz AKAN	Mühendislik Fakültesi, Topkapı Üniversitesi	_____
Doç. Dr. Adil Deniz DURU	Spor Bilimleri Fakültesi, Marmara Üniversitesi	_____

Bu tezin Doktora.tezi olarak bütün şartları sağladığını beyan ederim.

Tezin Lisansüstü Eğitim Enstitüsüne teslim tarihi: ____/____/____

Bu tezdeki tüm bilgilerin akademik kurallara ve etik davranışlara uygun olarak edinildiğini ve sunulduğunu beyan ederim. Ayrıca, bu kuralların ve davranışların gerektirdiği şekilde, bu çalışmada, orijinal olmayan tüm materyalleri ve sonuçları tamamen alıntı yaptığımı ve referans gösterdiğimi de beyan ederim.

Mehmet Alp KÖSE

ÖZET

SAWUML-UML TABANLI YAZILIM MİMARİSİ TANIMLAMA DİLİ VE ARAÇ KÜMESİ

KÖSE, Mehmet Alp

Doktora, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi

Danışman: Doç. Dr. Mert ÖZKAYA

Tarih: 05/2022

Sayfa: 377

Mimari tanımlama dili (*Architecture Description Language-ADL*) olan SAwUML, UML modelleme dilinden esinlenmiştir. UML'in yapısal ve davranışsal modellemelerini bir araya getiren SAwUML, yapısal modelleme için UML'in bileşen diyagramından; davranışsal modelleme için sıralama veya aktivite diyagramından yararlanılmıştır. Sıralama ve aktivite diyagramlarında kontrata dayalı tasarım yaklaşımı kullanılmıştır. SAwUML modellemesi için MetaEdit+ modelleme editörü kullanılmıştır. SAwUML kullanılarak geliştirilen modelden, MetaEdit+ editöründe geliştirilen dönüştürücüler sayesinde modelin java iskelet kodu ve ProMeLa kodu elde edilir daha sonra SPIN model denetleyicisinde, modelin biçimsel denetlemesi yapılır. Ayrıca, elde edilen ProMeLa kodundan tersine mühendislikle SAwUML modeli elde etmek için java dilinde yazılmış bir program geliştirilmiştir; böylece tasarımcı, tersine mühendislik programı sayesinde kolaylıkla alternatif davranışsal model (sıralı diyagramdan aktivite diyagramı veya tam tersi) oluşturabilir.

Anahtar Kelimeler: SAwUML, ProMeLa, SPIN, UML, Tersine Mühendislik

ABSTRACT

SAWUML-UML BASED SOFTWARE ARCHITECTURE DESCRIPTION LANGUAGE AND TOOL SET

KOSE, Mehmet Alp

PhD, Electrical and Computer Engineering, Altınbaş University,

Supervisor: Assoc. Prof. Dr. Mert OZKAYA

Date: 05/2022

Pages: 377

SAwUML is an architecture description language (ADL) inspired by the UML modeling language. SAwUML combines the structural and behavioral modeling of UML by using the component diagram of UML for structural modeling and by using sequence or activity diagram for behavioral modeling. The Design-by-Contract approach is used for behavioral modeling. MetaEdit+ modeling editor is used for modeling SAwUML. Using generators that are developed in the MetaEditor+, the java skeleton and ProMeLa codes are obtained from the SAwUML model. Using SPIN's ProMeLa, formal verification of the model is done. In addition, a reverse engineering program is written in java language to obtain the SAwUML model from the ProMeLa code; thus, the designer can easily create an alternative behavioral model (from sequential diagram to activity diagram or vice versa) using the reverse engineering program.

Keywords: SAwUML, ProMeLa, SPIN, UML, Reverse Engineering

İÇİNDEKİLER

Sayfa

ÖZET ..... **v**

ABSTRACT ..... **vi**

TABLO LİSTESİ ..... **ix**

ŞEKİL LİSTESİ ..... **x**

1.GİRİŞ **1**

1.1 BENZER ÇALIŞMALAR 5

1.1.1 Mimari Dilleri 5

1.1.2 UML Tabanlı Modelleme Dilleri 7

2. SAwUML **9**

2.1 PORT DAVRANIŞLARI 11

2.1.1 İstemci Port (Required Port) Davranışları 11

2.1.2 Sağlayıcı Port (Provided Port) Davranışları 12

3. AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ SAwUML **15**

4. METAEDIT+ **20**

5. PROMELA DÖNÜŞTÜRÜCÜ ALGORİTMASI **27**

6. JAVA DÖNÜŞTÜRÜCÜ ALGORİTMASI **31**

7.VAKA ANALİZLERİ **36**

7.1 GAZ İSTASYONU MODELİ 36

7.2 GAZ İSTASYONU MODELİ (AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ) 41

7.3 AEGİS VAKA ANALİZİ 50

7.4 AEGİS VAKA ANALİZİ (AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ) 73

7.5 AYA İNİŞ VAKA ANALİZİ 119

7.6 AYA İNİŞ VAKA ANALİZİ (AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ) 132

8. TERSİNE MÜHENDİSLİK İLE SAwUML OLUŞTURMA **157**

9. DEĞERLENDİRME 167

9.1 BİÇİMSEL SINAMA (FORMAL VERİFİCATION) 167

10. TARTIŞMA 172

11. SONUÇ 174

REFERANSLAR 176

EK A 181

EK B 195

EK C 201

EK D 212

EK E 223

TABLO LİSTESİ

Sayfa

Tablo 3.1: SAwUML aktivite diyagramı notasyon seti. ……16

Tablo 9.1: SPIN Model Denetleyicisi Sonuçları. ……167

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: SAwUML'de bileşenlerin birbirleri ile bağlantısı. ……9

Şekil 2.2: Bileşenin (*component*) özellikleri. ……9

Şekil 2.3: İstemci port (*required port*) diyagramı. ……10

Şekil 2.4: Sağlayıcı port (*provided port*) diyagramı. ……10

Şekil 2.5: İstemci portun *(required port*) bağlantı ismi ve bellek boyutu özellikleri. ……11

Şekil 2.6: İstemci port (*required port*) için çoklu istek ve yanıt mesaj davranışı. ……12

Şekil 2.7: Sağlayıcı port (*provided port*) için istek ve yanıt mesaj davranışı. ……13

Şekil 2.8: Sağlayıcı port (*provided port*) için çoklu istek ve yanıt mesaj davranışı. ……14

Şekil 3.1: Bileşen diyagramından aktivite diyagramına erişim. ……15

Şekil 3.2: Bileşen (*component*) kulvarı ve özellikleri. ……17

Şekil 3.3: Ön/sonraki koşul notasyonu ve özellikleri. ……17

Şekil 3.4: İstek göndermek için istemci portu (*required port*) olan bir bileşenin(*component*) aktivitesi ve özellikleri. ……18

Şekil 3.5: İstek almak için sağlayıcı portu (*provided port*) olan bir bileşenin (*component*) aktivitesi ve özellikleri. ……18

Şekil 3.6: Yanıt göndermek için sağlayıcı portu (*provided port*) olan bir bileşenin (*component*) aktivitesi ve özellikleri.....19

Şekil 3.7: Yanıt almak için istemci portu (*required port*) olan bir bileşenin (*component*) aktivitesi ve özellikleri.....19

Şekil 4.1: MetaEdit+ bileşenleri ve bağlantıları. 20

Şekil 4.2: MetaEdit+'da oluşturulan grafik tipleri. 20

Şekil 4.3: Component_Diagram grafiğinin object, relationship ve role tipleri. 21

Şekil 4.4: Component_Diagram grafiğin oluşturan parçaların birbirleriyle olan bağlantıları...21

Şekil 4.5: Component_Diagram grafiğindeki object'lerden ulaşılabilecek alt grafik türleri. ...22

Şekil 4.6: Sequence_required grafiğinde kullanılan bileşenlerin tipleri ve isimleri.....22

Şekil 4.7: Sequence_provided grafiğinde kullanılan bileşenlerin tipleri ve isimleri. 23

Şekil 4.8: Activity-Diagram grafiğinde kullanılan bileşenlerin tipleri ve isimleri. 23

Şekil 4.9: Component_Diagram isimli grafik tipinde oluşturulmuş diyagram. 24

Şekil 4.10: Sıralama diyagramlı (*sequence diagram*) SAwUML modelinin, java ve ProMeLa dönüştürülmelerinin sonuçları. 25

Şekil 4.11: Kullanıcı tanımlı bir LTL örneği.....26

Şekil 7.1: SAwUML'de oluşturulan bir *Customer*'ı olan gaz istasyon modeli.....36

Şekil 7.2: *Customer* bileşeninin verileri ve özellikleri. 36

Şekil 7.3: *Cashier* bileşeninin verileri ve özellikleri. 37

Şekil 7.4: *Pump* bileşeninin verileri ve özellikleri. 37

Şekil 7.5: *Customer*'ın *Cashier*'a bağlandığı istemci portun (*required port*) özellikleri. 38

Şekil 7.6: *Cashier*'in *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri. 38

Şekil 7.7: *Cashier*'in *Pump*'a bağlandığı istemci portun (*required port*) özellikleri. 39

Şekil 7.8: *Pump*'ın *Cashier*'e bağlandığı sağlayıcı portunun (*provided port*) özellikleri. 40

Şekil 7.9: *Customer*'ın *Pump*'a bağlandığı istemci portun (*required port*) özellikleri. 40

Şekil 7.10: *Pump*'ın *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri. 41

Şekil 7.11: *Customer-Cashier* aktivite diyagramı. 42

Şekil 7.12: *Cashier-Pump* aktivite diyagramı. 43

Şekil 7.13: *Customer-Pump* aktivite diyagramı. 44

Şekil 7.14: SAwUML'de oluşturulan aegis modeli. 51

Şekil 7.15: *ExperimentController* bileşeninin verileri ve özellikleri. 51

Şekil 7.16: *DoctrineAuthoring* bileşeninin verileri ve özellikleri. 52

Şekil 7.17: *TrackServer* bileşeninin verileri ve özellikleri. 53

Şekil 7.18: *DoctrineValidation* bileşeninin verileri ve özellikleri. 53

Şekil 7.19: *GeoServer* bileşeninin verileri ve özellikleri. 54

Şekil 7.20: *DoctrineReasoning* bileşeninin verileri ve özellikleri. 54

Şekil 7.21: *DoctrineAuthoring*'in *ExperimentController*'a bağlanan istemci port (*required port*) özellikleri. 55

Şekil 7.22: *ExperimentController*'ın *DoctrineAuthoring*'a bağlanan sağlayıcı bağlantı (*provided port*) özellikleri. 56

Şekil 7.23: *DoctrineValidation*'ın *ExperimentController*'a bağlanan istemci portun (*required port*) özellikleri. 57

Şekil 7.24: *ExperimentController*'ın *DoctrineValidation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 58

Şekil 7.25: *TrackServer*'ın *ExperimentController*'e bağlanan istemci portun (*required port*) özellikleri. 59

Şekil 7.26: *ExperimentController*'ın *TrackServer*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 60

Şekil 7.27: *DoctrineValidation*'ın *DoctrineAuthoring*'a bağlanan istemci portun (*required port*) özellikleri. 61

Şekil 7.28: *DoctrineAuthoring*'in *DoctrineValidation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 62

Şekil 7.29: *DoctrineValidation*'ın *TrackServer*'a bağlanan istemci portun (*required port*) özellikleri. 63

Şekil 7.30: *TrackServer*'in *DoctrineValidation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 64

Şekil 7.31: GeoServer'ın *DoctrineAuthoring*'a bağlanan istemci portun (*required port*) özellikleri. 65

Şekil 7.32: *DoctorineAuthoring*'in *GeoServer*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 66

Şekil 7.33: *DoctrineReasoning*'ın *DoctrineAuthoring*'a baglanan istemci portun (*required port*) özellikleri. 67

Şekil 7.34: *DoctorineAuthoring*'in *DoctrineReasoning*'e bağlanan sağlayıcı portun (*provided port*) özellikleri. 68

Şekil 7.35: *DoctrineReasoning*'ın *GeoServer*'a bağlanan istemci portun (*required port*) özellikleri. 69

Şekil 7.36: *GeoServer*'ın *DoctrineReasoning*'e bağlanan sağlayıcı portun (*provided port*) özellikleri. 70

Şekil 7.37: *GeoServer*'ın *TrackServer*'a bağlanan istemci portun (*required port*) özellikleri.. 71

Şekil 7.38: *TrackServer*'ın *GeoServer*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 72

Şekil 7.39: *DoctrineAuthoring-ExperimentController* aktivite diyagramı.....73

Şekil 7.40: *DoctrineValidation-ExperimentController* aktivite diyagramı.....75

Şekil 7.41: *TrackServer-ExperimentController* aktivite diyagramı.....77

Şekil 7.42: *DoctrineValidation-DoctrineAuthoring* aktivite diyagramı.....79

Şekil 7.43: *DoctrineValidation-TrackServer* aktivite diyagramı.....81

Şekil 7.44: *GeoServer-TrackServer* aktivite diyagramı.....83

Şekil 7.45: *GeoServer-DoctrineAuthoring* aktivite diyagramı.....85

Şekil 7.46: *DoctrineReasoning-DoctrineAuthoring* aktivite diyagramı.....87

Şekil 7.47: *DoctrineReasoning-GeoServer* aktivite diyagramı.....89

Şekil 7.48: SAwUML'de oluşturulan Aya İniş modeli.....120

Şekil 7.49: *DataStore* bileşeninin verileri ve özellikleri.....120

Şekil 7.50: *Calculation* bileşeninin verileri ve özellikleri.....121

Şekil 7.51: *UserInterface* bileşeninin verileri ve özellikleri.....121

Şekil 7.52: *Calculation*'ın *DataStore*'a bağlanan istemci portun (*required port*) özellikleri.....122

Şekil 7.53: *DataStore*'un *Calculation*'a bağlanan sağlayıcı portun (*provide port*) özellikleri.....123

Şekil 7.54: *DataStore*'ın *Calculation*'a bağlanan istemci portun (*required port*) özellikleri. 124

Şekil 7.55: *Calculation*'nın *DataStore*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 125

Şekil 7.56: *UserInterface*'in *DataStore*'a bağlanan istemci portun (*required port*) özellikleri. 127

Şekil 7.57: *DataStore*'un *UserInterface*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 128

Şekil 7.58: *Calculate'*in *UserInterface*'e bağlanan istemci portun (*required port*) özellikleri. 129

Şekil 7.59: *UserInterface*'in *Calculation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 130

Şekil 7.60: *Calculate'*in *UserInterface*'e bağlanan istemci portun (*required port*) özellikleri. 131

Şekil 7.61: *UserInterface*'in *Calculation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri. 131

Şekil 7.62: *Calculation - DataStore* aktivite diyagramı. 132

Şekil 7.63: *DataStore - Calculation* aktivite diyagramı. 135

Şekil 7.64: *UserInterface - DataStore* aktivite diyagramı. 137

Şekil 7.65: *Calculation - UserInterface* aktivite diyagramı. 139

Şekil 7.66: *Calculation - UserInterface* aktivite diyagramı. 140

Şekil 8.1: Tersine model için tek müşterili gaz istasyon modelinin .pml dosyasının seçilimi. 158

Şekil 8.2: Seçilen .pml uzantılı dosyanın çalıştırılması. 158

Şekil 8.3: Seçilen bir müşterili gaz istasyon modeline ait .pml uzantılı dosyanın çalıştırılması sonucu elde edilen bileşenler ve o bileşenlere ait özellikler. 159

Şekil 8.4: .mxm uzantılı dosyanın elde edilmesi ve MetaEdit+ editörüne yüklenmesi. 160

Şekil 8.5: Tersine programlama ile elde edilmiş, aktivite diyagramı ile genişletilmiş tek müşterili gaz istasyonu SAwUML modeli. 160

Şekil 8.6: Aktivite diyagramı ile genişletilmiş tek müşterili gaz istasyonu SAwUML modelinin ana grafiği. 161

Şekil 8.7: Tek müşterili gaz istasyonu SAwUML modelinin Cashier-Pump aktivite diyagramını gösteren alt grafiği. 161

Şekil 8.8: Tek müşterili gaz istasyonu SAwUML modelinin Customer-Cashier aktivite diyagramını gösteren alt grafiği. 162

Şekil 8.9: Tek müşterili gaz istasyonu SAwUML modelinin Customer-Pump aktivite diyagramını gösteren alt grafiği. 162

Şekil 8.10: Final.mxm uzantılı dosyanın elde edilmesi ve MetaEdit+ editörüne yüklenmesi. 163

Şekil 8.11: Tersine programlama ile elde edilmiş, tek müşterili gaz istasyonu SAwUML modeli. 163

Şekil 8.12: *Cashier*'in *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri...164

Şekil 8.13: *Pump*'ın *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri. 164

Şekil 8.14: *Pump*'ın *Cashier*'e bağlandığı sağlayıcı portun (*provided port*) özellikleri. 165

Şekil 8.15: *Customer*'ın *Pump*'a bağlandığı istemci portun (*required port*) özellikleri. 165

Şekil 8.16: *Customer*'ın *Cashier*'a bağlandığı istemci portun (*required port*) özellikleri. 166

Şekil 8.17: *Cashier*'in *Pump*'a bağlandığı istemci portun (*required port*) özellikleri. 167

Şekil 9.1: gas1.pml de SPIN model denetleyicide pan -r komutu kullanılarak tespit edilen kilitlenme (*deadlock*) hatasının gösterimi. 168

Şekil 9.2: Gaz istasyon modelinde, SPIN model denetleyicisi tarafından pan komutu kullanılarak tespit edilen eksiklik hatasının (*incompleteness error*) gösterimi. 169

Şekil 9.3: Gaz istasyon modelinde, SPIN model denetleyicisi tarafından pan -r komutu kullanılarak tespit edilen eksiklik hatasının (*incompleteness error*) detaylı gösterimi. 169

Şekil 9.4: Kullanıcı tarafından gaz istasyon modelinde eklenen LTL özelliği. 170

Şekil 9.5: Gaz istasyon modelinde, hata vermeyen LTL özelliği. 170

Şekil 9.6: Gaz istasyon modelinde, hata veren bir LTL özelliği. 171

Şekil E.1: Tersine mühendislik için kullanılan java dosyaları ve klasörler. 223

1.GİRİŞ

Yazılım mimarisi, bileşenler, bileşenlerin davranış özelliklerini ve birbirleri ile olan etkileşimlerini kapsayan bir sistem yapısıdır [1]-[3]. Yazılım mimarisi, bir sistemin hangi bileşenlerden oluştuğu ve bu bileşenlerin birlikte çalışıp çalışmadığı, bileşenlerin ne tür arayüzlere sahip olacağı, bileşenler arası iletişimin ne olacağı ile ilgilidir. Özellikle büyük ölçekli yazılımları tasarlamak için yazılım mimarisi kullanmak gereklidir. Yazılım mimarisine dayanarak tasarlanan bir model ekonomik olur çünkü tasarımdan kaynaklanacak hatalar tasarım detaylandırılmadan tespit edilir, yazılım belirli bir plan için de tasarlandığı için yazılımı geliştiren kişiler değişse bile yazılımın bakımı kolay olur, yapılacak güncellemelerle başka bir yazılım projesinde tekrar kullanılabilir. Yazılım mimarisini oluşturmak için Mimari Tanımlama Dili (*Architecture Description Language-ADL*) kullanılır [4]-[6]. Günümüze kadar onlarca mimari tanımlama (*ADL*) dilleri geliştirilmiştir. ADL biçimsel (*formal*) özellikleri olan bir dildir ve sistemi oluşturan bileşenlerinin yapıları, davranışları ve bağlantılarını sistemin yazılım mimarisi açısından soyut bir seviyede tarif eder. Genel olarak, geliştirilen ADL'ler belirli bir alan (*domain*) için tasarlanmış olması ve/veya sistemin modellenmesi için kullanılan biçimsel (*formal*) özellikli dillerin (pi-calculus vb.) bilinmesi gerekeceğinden ve öğrenilmesinin belirli bir zaman alacağından dolayı çoğu tasarımcı tarafından kullanılması tercih edilmemektedir [7]. Bunun yanı sıra, yarı biçimsel olarak kabul edilebilen ve 90'ların başında geliştirilmeye başlanan genel amaçlı modelleme dili olan UML [8], günümüzde tasarımcılar tarafında en çok tercih edilen modelleme dilidir.

UML diyagramlardan oluşan bir modelleme dilidir. Kullanıcıya, tasarım için değişik bakış açılarını (mantıksal, fiziksel, geliştirme, proses vb.) farklı diyagramlarla sunmaktadır. UML diyagramları, yapısal ve davranışsal olarak ikiye ayrılır. Yapısal diyagramlar, sistemi oluşturan nesneleri, o nesnelerin niteliklerini, sahip oldukları işlemleri ve nesnelerin birbirleri ile olan ilişkilerini gösteren statik yapılardır. Yapısal diyagramlar, sınıf diyagramı, bileşen diyagramı, nesne diyagramı, birleşik yapı diyagramı, dağılım diyagramı, paket diyagramı ve profil diyagramından oluşur. Davranışsal diyagramlar, sistemi oluşturan nesnelerin birbirleri ile olan etkileşimlerini ve bundan dolayı nesnelerin içinde meydana gelen değişiklikleri gösteren

diyagramlardır. Davranışsal diyagramlar, aktivite diyagramı, durum diyagramı, kullanım senaryosu diyagramı, etkileşime bakış diyagramı, sıralama diyagramı, zaman akış diyagramı ve iletişim diyagramından oluşur.

Reggio et al.[9] yaptığı çalışmada 275 katılımcıya (%40'ı akademiden %60'ı endüstriden olmak üzere) UML diyagramlarından hangilerinin daha fazla kullanıldığı sorulmuştur. Çalışmaya göre en fazla bilinen diyagramlar sırasıyla; kullanım senaryosu diyagramı (97%), sınıf diyagramı, durum diyagramı ve sıralama diyagramı (96%) ve aktivite diyagramı (92%) olmuştur. Aynı çalışmaya göre en fazla kullanılan diyagramlar sıralama diyagramı (%79), sınıf diyagramı (%78), kullanım senaryosu diyagramı (77%), durum diyagramı (70 %) ve aktivite diyagramı (67%) olmaktadır.

Grosmann et al.[10], 131 katılımcı üzerinden yaptığı çalışmada, en fazla kullanılan diyagramlar sırasıyla; kullanım senaryosu diyagramı (93%), sınıf diyagramı (93%), sıralama diyagramı (89%), durum diyagramı (63%) ve aktivite diyagramı (60%) olmaktadır.

Zeichick [11] çalışmasına göre en fazla kullanılan diyagramlar sırasıyla; kullanım senaryosu diyagramı (89%), sınıf diyagramı (75%), sıralama diyagramı (75%), durum diyagramı (52%) ve aktivite diyagramı (52%) olmaktadır.

Ozkaya [12] çalışmasında, yazılım mimarisi modelleri semantik açıdan biçimsel (*formal*) ve biçimsel olmayan (*informal*) olarak değerlendirilmiştir. Biçimsel (*formal*) modeller, matematiksel kurallara ve mantığa bağlı olarak tanımlanmış dilleri (CSP, pi-calculus, ProMeLa gibi) kullanarak karmaşık sitemlerin doğru bir şekilde test edilmesini sağlayan biçimsel (*formal*) semantikten oluşmuştur. Biçimsel olmayan modeller ise (düz yazı kullanarak veya kutular ve çizgilerden oluşturulan modeller hariç) düzgün tanımlanmış sözdizimleri ve yarı biçimsel semantikten oluşmuş bir notasyona sahiptirler (UML gibi). 28 ülkeden 115 katılımcıyla yapılan bu çalışmada biçimsel (*formal*) olmayan modelleme dillerinin yazılım mimarisi modellemesinde kullanılması daha yaygın çıkmıştır. Biçimsel (*formal*) olmayan modelleme dillerinden en fazla kullanılan ise (83%) oranı ile UML olmuştur.

Her ne kadar tasarımcılar açısından popüler olsa da, UML yazılım mimarisi modellemesi açısından yetersiz kalmaktadır. Öncelikle, UML yarı biçimsel bir semantiği olmasından dolayı biçimsel (*formal*) bir analizi desteklememektedir, yazılım mimari modelin geçerliliği ve doğrulaması yapılamamaktadır, UML tabanlı geliştirilen ADL'ler bu eksikliği gidermeye çalışmıştır [13].

UML tabanlı geliştirilen ADL'ler de yapısal tasarımlar için genelde sınıf diyagramları kullanılmıştır. Sınıf diyagramları sistemdeki bileşenleri göstermesi açısından kullanışlı olsa da bileşenlerin birbirleri ile olan bağlantıları ile ilgili detay vermez. Bileşenlerin birbirleri ile hangi çeşit bir ara yüz ile bağlandığı, bağlantı yapan bileşenlerin aralarında nasıl bir etkileşim protokolü olduğu ve birbirleri ile nasıl bir iletişim şekli olduğu (senkron, asenkron gibi) bilgiler sınıf diyagramında belirtilmesi kolay değildir. Davranışsal tasarımlar olarak genelde durum diyagramı kullanılmıştır. Ancak durum diyagramı sadece bir bileşenin durumunu gösterdiği için sistemin bütününü göstermesi açısında kullanılması verimli değildir. Aktivite veya sıralama diyagramı kullanılması tercih edilir [14].

Bu çalışmada, SAwUML [13] adında bir ADL geliştirildi. UML'in tasarımcılar arasında yaygın olarak kullanılması, UML'in genel amaçlı bir modelleme dili olması nedeniyle, SAwUML bir UML tabanlı ADL olarak geliştirildi. SAwUML'in önemli bir özelliği hem yapısal hem de davranışsal modellemeyi bir arada gösterebilmesidir. Bileşenlerin birbirleri ile ilişkisini ve bağlantı çeşidini en iyi gösteren diyagram olan bileşen diyagramı yapısal modellemede kullanılmıştır. Davranışsal modellemede ise kullanıcıya iki alternatif diyagram sunulmuştur. Bunlar aktivite ve sıralama diyagramlarıdır. Yapısal modellemede her bir bileşen diğer bileşene istemci port (*required port*) veya sağlayıcı port (*provided port*) ile bağlanır. İstemci porta sahip bileşen, bağlandığı sağlayıcı porttan istekte bulunur. Bileşenlerin portlar arasındaki ilişkileri, metotlar vasıtasıyla parametrelerin karşı tarafa iletilmesi, iletilen yerde belirli hesaplamaların yapılıp; döndürülen bir değerle yanıt verilmesi şeklinde olur. SAwUML'de bileşenlerin arasındaki istek ve cevap şeklindeki dinamik ilişkileri, durum diyagramı veya aktivite diyagramında gösterilir. İstemci porttan yapılan istek, bağlı olduğu sağlayıcı porttan verilen cevapla tamamlanır. Mesajların metotlar vasıtasıyla gönderilmesi ve alınması sırasında doğrulamayı arttırmak için kontrata dayalı tasarım (*design by contract*) yaklaşımı kullanılıyor

[15]. Hoara [16] mantığının kullanıldığı kontrata dayalı tasarımda ön koşul (*pre-conditon*) ve son koşul (*post-condition*) kullanılır. SAwUML'de son koşul yerine veri ataması kullanılmıştır. Bu sayede, model doğrulama yaparken olasılıkları azaltarak biçimsel *(formal*) analiz yapmayı kolaylaştırmış olmaktadır.

İstemci portu (*required port*) olan bileşen, sağlayıcı port (*provided port*) ile bağlı olduğu bileşene istek göndermeden önce ön koşul (*pre-condition*) şartının sağlanması gerekir. Bu şart sağlandıktan gerekirse, metot çağrısını yaptığı parametrelerinin ataması yapılır ve istek gönderilir. İsteğe gelen cevap ise başka bir ön koşul (*pre-condition*) ile değerlendirilir; eğer o ön koşulu sağlıyorsa, son koşul (*post-condition*) olarak gerekirse veri ataması gerçekleştirilir. Sağlayıcı porta (*provided port*) gelen istek, buradaki ön şartı (*pre-condition*) sağlaması halinde, sağlayıcı portu olan bileşendeki gerekli veri atamaları yapılır son koşul (*post-condition*) olarak; daha sonra varsa dönüş değeri ile birlikte metot çağrısını sağlayıcı portun (*provided port*) iletişimde olduğu istemci portu (*required port*) olan bileşene cevap olarak iletilir.

SAwUML, yazılım mimari tasarımı yapmak için bir modelleme editörü kullanıyor. Modelleme editorü vasıtasıyla tasarlanan mimari yazılım modeli için daha sonra kullanıcı tanımlı özellikleri belirtmek için doğrusal zamansal mantık (*Linear Temporal Logic-LTL*) kullanılabilir. SAwUML modelleri, SPIN ProMeLa biçimsel (*formal*) doğrulama diline otomatik olarak dönüştürülerek, önceden tanımlanmış (*pre-defined*) kilitlenme (*deadlock*), eksiklik (*incompleteness*) gibi özellikler ile, kullanıcı tanımlı doğrusal zaman mantık (*LTL*) özellikleri test edilir.

SAwUML'de kod dönüştürücü geliştirilmiştir. Kullanıcı isterse, yüksek seviyeli bir dil olan java'ya geliştirdiği modeli dönüştürebilir. Adaptör (*adapter*) tasarım örüntüsü kullanılarak oluşturulan kodda, birbirleri ile ilişkisi bulunan bileşenlerin mesaj isteğinde bulunup, cevap alabilmesi için adaptör (*adapter*) sınıfı kullanılır. Oluşturulan kod, tasarımın dinamik ve yapısal özelliklerini kapsar. Kullanıcı, başlangıç seviyesinde oluşturulan bu koda yeni modüller ekleyerek (GUI, ağ, veritabanı bağlantısı vb.) programı geliştirebilir.

SAwUML'de dönüştürücüden elde edilen ProMeLa dosyasından tersine mühendislik yaparak, tasarım modeli elde edilebiliyor. Tersine mühendislik için geliştirilen yazılım java programlama

dili ve SQLite veritabanı kullanılarak oluşturulmuştur. Kullanıcı, bu sayede SAwUML modelden elde ettiği ProMeLa dosyasından aynı yapısal modele sahip olmakla beraber ister sıralı diyagrama sahip isterse de aktivite diyagrama sahip davranışsal modele geri dönüştürebiliyor. Tersine mühendislik çok detaylı modellerde farklı bir davranışsal modele geçmek ve oradan da modelin java kodunu elde etmek kullanıcı kolaylığı açısından elverişli olmaktadır. Ayrıca, ProMeLa üzerinden, ön ve/veya son koşullar üzerinden yapılacak değişikler, tersine mühendislik sayesinde model tasarımı üzerinde kolaylıkla gösterilir.

SAwUML'de ve diğer çalışmalarda modelin doğrulanmasında biçimsel (*formal*) semantik bir dil olan ProMeLa kullanılmıştır. Model denetleyicisi SPIN [17] ve onun desteklediği dil olan ProMeLa'nın kullanıcı açısından avantajları vardır. Öncelikle, "process algebra" tabanlı diğer biçimsel (*formal*) dillerden farklı olarak C programlama dilini andıran notasyona sahiptir. Bu durum kullanıcı açısından öğrenilmesini ve kullanılmasını kolaylaştırır. ProMeLa dilinde kullanılan değişken bildirimi, veri atama, eğer-aksi takdirde deyimleri (*if-else statements*), döngüler, kullanıcı açısından oldukça benzer gelmektedir. SPIN, 1991 yılında beri ücretsiz olarak kullanılabilmektedir.

1.1 BENZER ÇALIŞMALAR

SAwUML'e benzer çalışmalar, mimari diller ve UML tabanlı diller olarak iki kısımda incelenmiştir.

1.1.1 Mimari Dilleri

Yapılan [18] çalışmasında Mimari dilleri belirli kategorilere ayırarak değerlendirmişlerdir. Bunlar, dil tanımı, dil özellikleri ve araç desteği olmak üzere ana kategorilerdir. Dil tanımı kategorisini; fonksiyonel olmayan özellikler (örneğin, performans, güvenilirlik, vb.), biçimsel (*formal*) dil kullanılması ve grafik/yazısal desteğinin olması bakımından değerlendirilmiştir. Dil özellikler kategorisini; çoklu bakış açısı bakımından (örneğin, mantıksal, yapısal, fiziksel, vb.), dili isteğe göre düzenleme ve genişletme ile, programlama çerçevesinin (*programming framework*) olması açısından değerlendirilmiştir. Son olarak araç desteği kategorisi ise; otomatik analizin olması, yazılım mimari merkezli tasarımı desteklemesi, bakış açıları

yönetiminin olması, işbirliği desteği olması, sürüm desteğinin olması ve bilgi yönetimi desteğinin olması açısından değerlendirilmiştir. Bu kategorilere göre yapılan Ozkaya [4] çalışmasında 125 mimari dil analiz edilmiştir. Sadece 5 tane mimari dil (%4) görsel olarak yapısal ve davranışsal tasarım kararlarını bir modelleme aracı ile desteklerken, biçimsel (*formal*) metotlar kullanarak kapsamlı analizler yapabilmektedir. Bunlar; KADL [19], Modechart [20], Statemate [21], X-MAN [22], and TADL [23].

KADL, yapısal özellikler için sınıf diyagramını, davranışsal özellikler için sembolik geçiş sistemini (*symbolic transition system*) kullanıyor. Kapsamlı model denetleme özelliği (SyCLAP[24]) varken, eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak zamansal mantık (temporal logic) kullanıyor.

Modechart, yapısal özellikler için çizgi ve kutular kullanırken, davranışsal özellikler için durum diyagramı (*state chart*) kullanıyor. Kapsamlı model denetleme özelliği varken, eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak birinci tanımlı mantık (first order logic) kullanıyor.

Statemate, hem yapısal özellikler için hem de davranışsal özellikler için durum diyagramı (*state chart*) kullanıyor. Kapsamlı model denetleme özelliği (SPIN-ProMeLa) varken, eksiklik (*incompleteness*) hatası denetlemesi özelliği ve kullanıcı tanımlı özelliği yoktur.

X-Man, yapısal özellikler için çizgi ve kutular kullanırken, davranışsal özellikler için kontratlara dayalı tasarım (*design by contract*) kullanıyor. Kapsamlı model denetleme özelliği varken (X-Man), eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak kontratlara dayalı tasarım (*design by contract*) kullanıyor.

TADL, yapısal özellikler için çizgi ve kutular kullanırken, davranışsal özellikler için zamanlı otomat (*timed automata*) kullanıyor. Kapsamlı model denetleme özelliği varken (UPPAAL [25]), eksiklik (*incompleteness*) hatası denetlemesi özelliği varken kullanıcı tanımlı özelliği yoktur.

1.1.2 UML Tabanlı Modelleme Dilleri

Pataricza et al. [26], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği ve eksiklik (*incompleteness*) hatası denetlemesi özelliği vardır. Kullanıcı tanımlı özellik olarak durum diyagramı kullanıyor.

Cabot et al. [27], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için nesne kısıtlama dilini (*object constraint languag -OBL*) durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği varken eksiklik (*incompleteness*) hatası denetlemesi özelliği ve kullanıcı tanımlı özelliği yoktur.

Apvrille et al. [28], yapısal özellikler için SysML'nin blok diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği varken (RT-LOTOS [29], UPPAAL) eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak mantık formülü (*logic formula*) kullanıyor.

V-ProMeLa [30], yapısal özellikler için gerçek zamanlı (RT-UML) [31] kullanırken, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği varken (SPIN-ProMeLa) eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak doğrusal zamansal mantık (*Linear Temporal Logic-LTL*) kullanıyor.

Bose [32], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği varken (SPIN-ProMeLa), eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak durum diyagramı kullanıyor.

Siveroni et al. [33], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği varken (SPIN-ProMeLa), eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak doğrusal zamansal mantık (*Linear Temporal Logic-LTL*) kullanıyor.

McUmber et al. [34], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği (SPIN-ProMeLa) ve eksiklik (*incompleteness*) hatası denetlemesi özelliği vardır. Kullanıcı tanımlı özelliği yoktur.

Campbell et al.[35], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği (SPIN-ProMeLa) ve eksiklik (*incompleteness*) hatası denetlemesi özelliği vardır. Kullanıcı tanımlı özellik olarak doğrusal zamansal mantık (*Linear Temporal Logic-LTL*) kullanıyor.

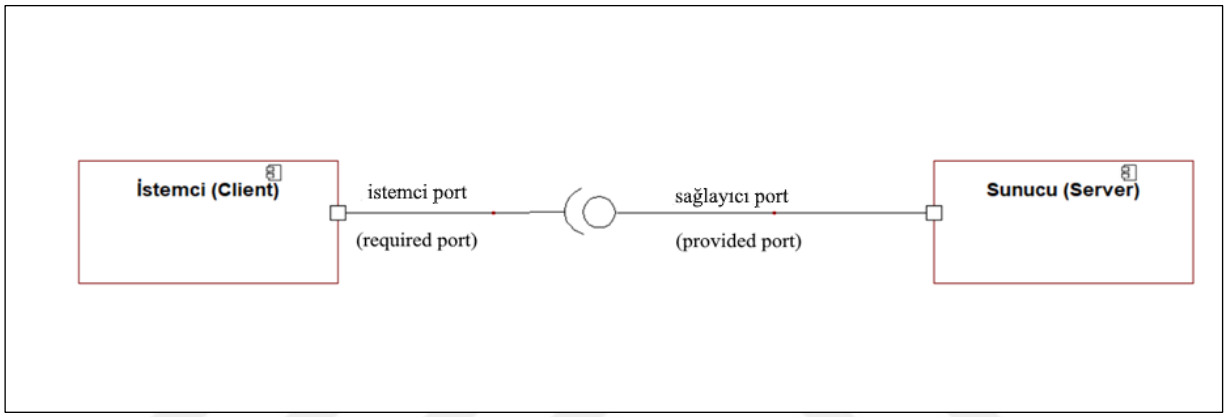
Beek et al.[36,37], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği (UMC) varken eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak UCTL isimli, UML tabanlı zaman çatallanmalı zamansal mantık (*branching time temporal logic*) kullanıyor.

Lilius et al. [38], yapısal özellikler için sınıf diyagramını, davranışsal özellikler için durum diyagramını kullanıyor. Kapsamlı model denetleme özelliği varken (SPIN-ProMeLa), eksiklik (*incompleteness*) hatası denetlemesi özelliği yoktur. Kullanıcı tanımlı özellik olarak, durum diyagramlarına eklenen ve daha sonra doğrusal zamansal mantık (*Linear Temporal Logic-LTL*) çevrilip SPIN'de denetlenebilen stereotipi (*stereotype*) kullanılıyor.

2. SAwUML

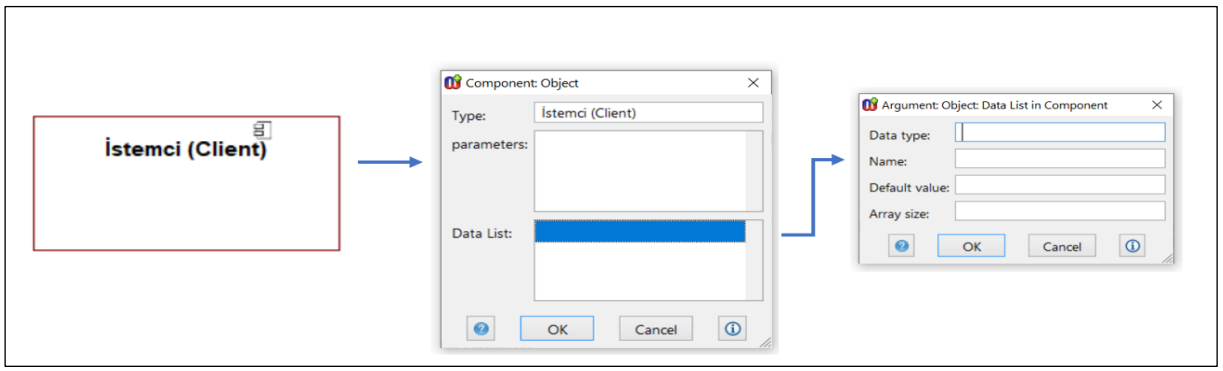
Geliştirdiğimiz SAwUML dili, UML'in bileşen diyagramından esinlenmiştir. Her bir bileşen birbirleri ile istemci (*required port*) ve sağlayıcı (*provided port*) portlar ile bağlanırlar.

İstemci port (*required port*) yarım daire olarak sembolize edilirken; sağlayıcı port (*provided port*) dairesel olarak sembolize edilir (Şekil 2.1).



Şekil 2.1: SAwUML'de bileşenlerin birbirleri ile bağlantısı

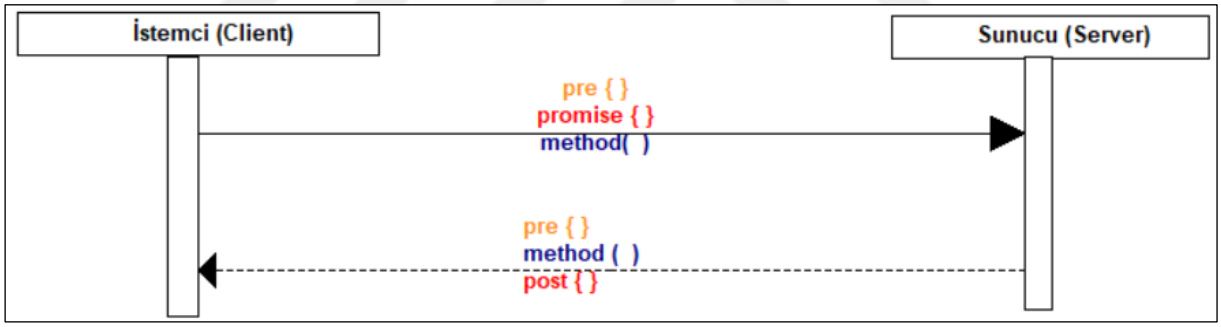
Dikdörtgen kutu şeklinde sembolize edilen her bir bileşen (*component*), tasarlanan sistem içinde bağımsız bir sınıfı (*class*) temsil etmektedir. MetaEdit+'de [39] geliştirilen bileşenin (*component*) sembolize ettiği dikdörtgen kutuya tıklandığında bileşenin ismini (*Type*), parametrelerini (*parameters*) ve veri listesini (*Data List*) gösteren diyalog kutusu (*dialog box*) çıkar (Şekil 2.2).



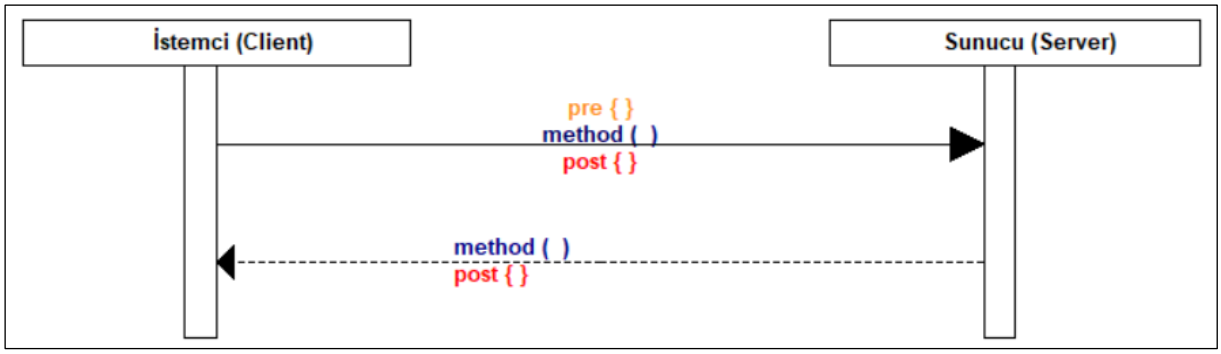
Şekil 2.2: Bileşenin (*component*) özellikleri.

Yazılım mimarisinde kullanılan her bir bileşenin (*component*) ismi, diğer bileşenlerden (*component*) ayırt edilebilmesi için farklı olmalıdır. Her bileşen, tasarımcıların bileşenlere bilgi aktarabileceği bir dizi parametreye sahip olabilir. Her bir bileşendeki (*component*) veri listesi (*Data List*), bileşenin (*component*) veri değişkenlerini (*data variables*) belirtir. Bu veri değişkenleri (*data variables*), o bileşenin (*component*) durumunu ve bağlantı noktalarında kullanılan metotlardaki (*port method*) parametrelere atayacağı değerleri temsil eder. Veri listesinde (*Data List*) tıklama yapılarak ikinci bir diyalog kutusu açılır. Burada veri değişkeninin tipi (*Data type*) integer, char veya bool olarak, ismi (*Name*), varsayılan değeri (*Default value*) ve varsa dizi boyutu (*Array size*) belirlenir.

Bileşenlerin istemci portu (*required port*) gösteren yarım daireye veya sağlayıcı portu gösteren (*provided port*) tam daireye sağ tıklandığında Şekil 2.3 ve Şekil 2.4'de örneği görünen MetaEdit+'de oluşturulmuş alt diyagramlara erişilir.

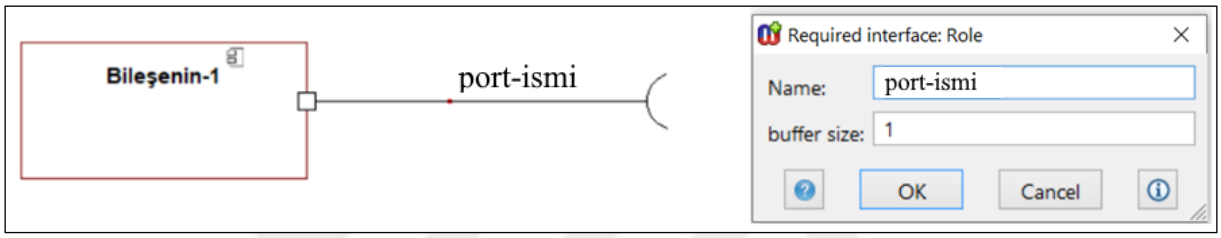


Şekil 2.3: İstemci port (*required port*) diyagramı.



Şekil 2.4: Sağlayıcı port (*provided port*) diyagramı.

Şekil 2.5'de görüldüğü gibi yarım daire ile kırmızı nokta arasındaki çizgiye sol tıklandığında bir diyalog kutusu ortaya çıkar, tasarımcı burada port ismini ve bellek (*buffer*) boyutu girer. Bellek boyutu sıfır (0) ise, birbirleri ile portlar sayesinde bağlı olan bileşenler (*component*) arasındaki iletişim senkron olacak demektir. Birbirleri ile bağlı olan bu bileşenlerin (*component*) karşılıklı olarak metotları vasıtasıyla göndereceği ve alacağı veriler eş zamanlı olacaktır. Bellek boyutu sıfırdan büyük olursa birbirleri ile bağlı olan bileşenler arasındaki iletişim asenkron olur ve metotlar vasıtasıyla iletilecek ve alınacak veriler bellekte tutulur.



Şekil 2.5: İstemci portun (*required port*) bağlantı ismi ve bellek boyutu özellikleri.

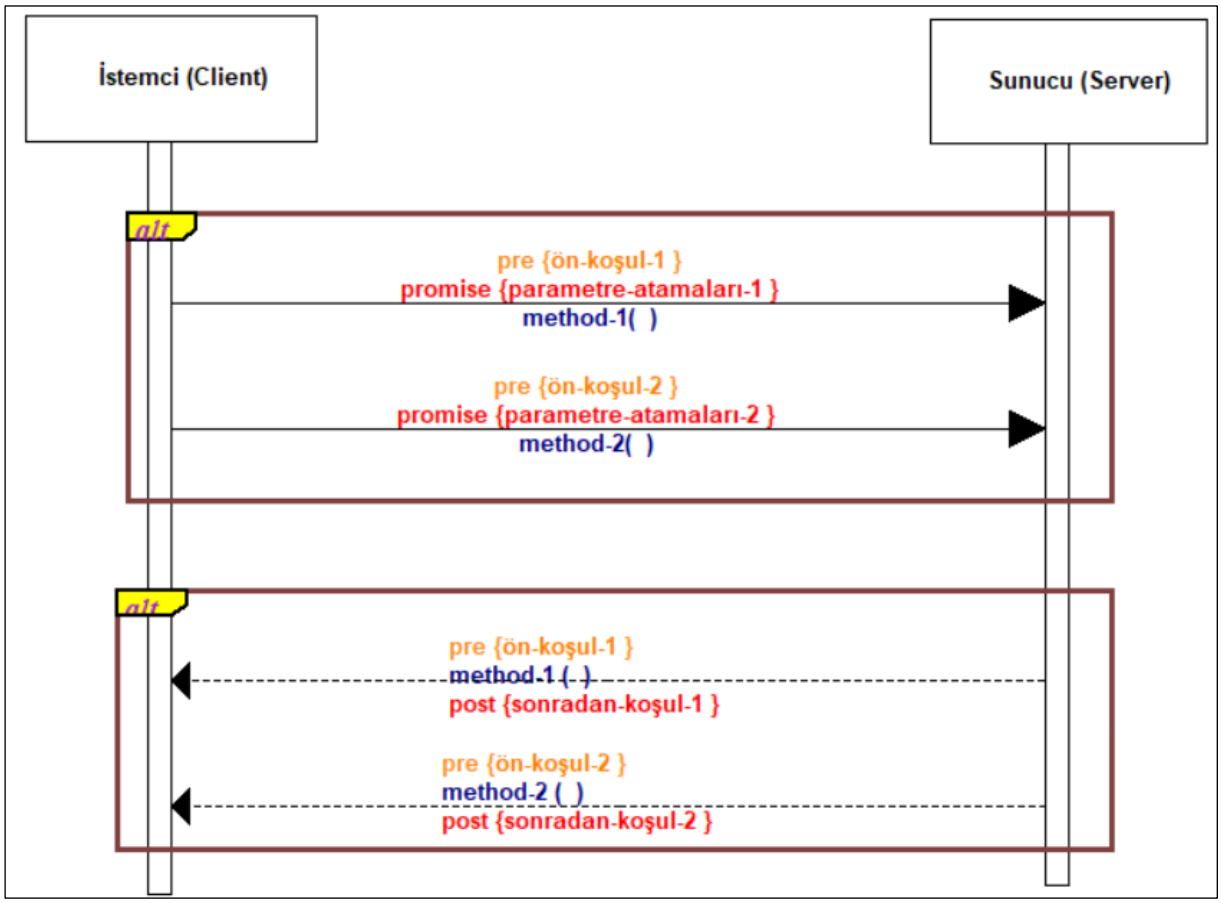
2.1 PORT DAVRANIŞLARI

Portlarda sağ tıklayınca oluşan diyagramlar (Şekil 2.3 ve Şekil 2.4), UML'in sıralamalı diyagramından (*sequence diagram*) esinlenmiştir. Her bir diyagramda bağlantıyı oluşturan bileşen (*component*) çiftleri temsilen zaman çizelgesi (*lifeline*) bulunur. İstemci portu (*required port*) olan bileşen solda; sağlayıcı portu (*provided port*) olan bileşen sağda gösterilir.

2.1.1 İstemci Port (Required Port) Davranışları

İstemci portu (*required port*) olan bileşeni temsil eden zaman çizelgesinden, sağlayıcı portu (*provided port*) olan bileşeni temsil eden zaman çizelgesine düz ok bağlantısı ile mesaj iletmeyi; tersi yönde kesikli ok bağlantısı ile yanıt almayı gösterir (Şekil 2.3). İki bileşen (*component*) arasında mesaj iletilmesi (metot çağrısı) yapılabilmesi için bir kontrata uyulması gerekir. Bu kontrat, "pre" ile gösterilen edilen ön koşul şartı ve "promise" ile gösterilen çağrısı yapılacak metodun parametrelerinin değerlerinin atanmasından oluşur. Metot vasıtasıyla mesajın iletilmesi için, öncelikle "pre" ön koşul şartının sağlanması gerekir. Eğer bu şart sağlanırsa, metodun (varsa) parametrelerinin değerlerinin atanması gerçekleştirilir (promise). Daha sonra metot çağrısı sağlayıcı porta (*provided port*) gönderilir ve metot çağrısı cevabı asenkron olarak

beklenir. Bu arada, istemci porttan (*required port*) başka metot çağrıları yapılır. Ancak, aynı metot çağrısının tekrar yapılabilmesi için o metot çağrısı cevabının gelmesi gerekir. Gelen mesaj cevap kontratı, "pre" ön şartı ve "post" ile gösterilen bileşenin (*component*) verilerinin yeni değerlerinin atanabileceği durumdan oluşur. Eğer, "pre" ön şartı sağlanırsa "post" durumunda gerekirse bileşenin verilerinin yeni değerleri atanır. Şekil 2.6'da gösterildiği gibi aynı porttan birden fazla istek ve yanıt mesajı almak mümkündür.

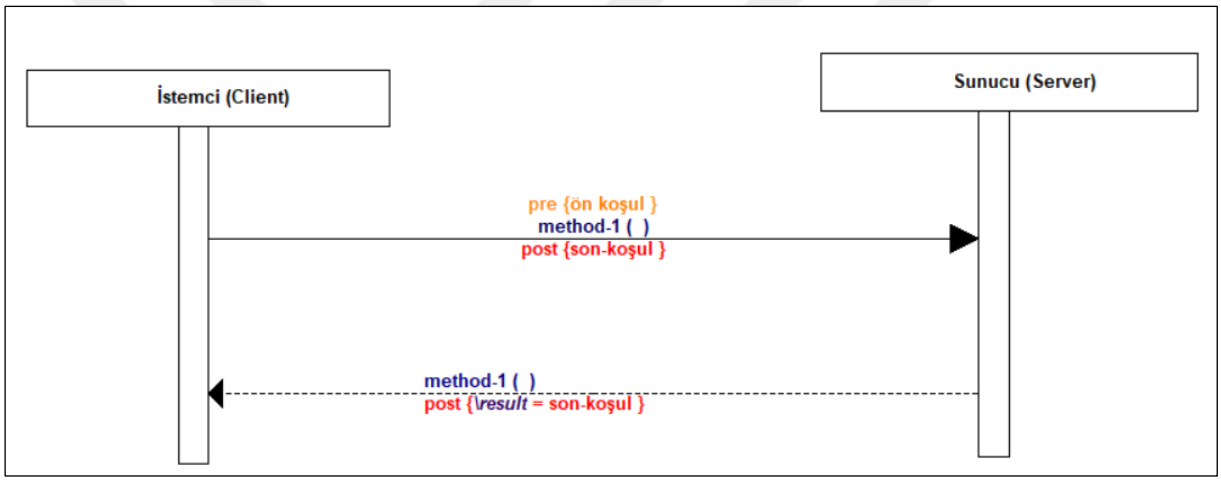


Şekil 2.6: İstemci port (*required port*) için çoklu istek ve yanıt mesaj davranışı.

2.1.2 Sağlayıcı Port (Provided Port) Davranışları

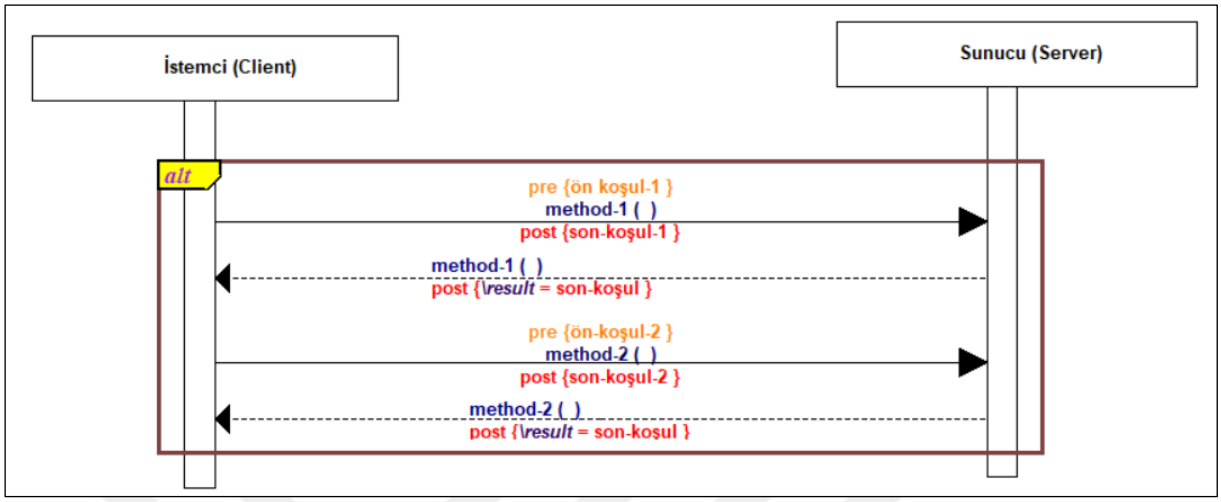
Bileşen (*component*) diyagramında, sağlayıcı port (*provided port*) sembolüne sağ tıklayınca, sağlayıcı port (*provided port*) grafiğine erişilir. Bu diyagramda aynı istemci port grafiği gibi, sol tarafta istemci portu (*required port*) olan bileşeni temsil eden zaman çizelgesinden, sağlayıcı portu (*provided port*) olan bileşeni temsil eden zaman çizelgesine düz ok bağlantısı ile mesajın

alınmasını (metot çağrısının alınması); tersi yönde ise kesikli ok bağlantısı ile yanıt vermeyi gösterir (Şekil 2.7). Sağlayıcı porta (*provided port*), istemci porttan (*required port*) mesajın iletilmesi için ön koşul şartının sağlanması gerekir. Ön koşul şartı sağlandıktan sonra, son koşul şartlarına göre (*post-condition*) bileşenin (sağlayıcı porta bağlı) verilerine gerekirse yeni değerler atanır. Mesajın (metot çağrısının), varsa dönüş değeri "\result" değişkenine atanarak (kesikli okla gösterilen) mesaj tekrar bağlı olduğu gerekli porta (*required port*) iletilir. Sağlayıcı porta (*provided port*) gelen ve iletilen mesajlar bir bütün aksiyon halinde olurken, istemci porttan (*required port*) giden ve gelen mesajlar ayrı aksiyonlar olarak gerçekleşir.



Şekil 2.7: Sağlayıcı port (*provided port*) için istek ve yanıt mesaj davranışı.

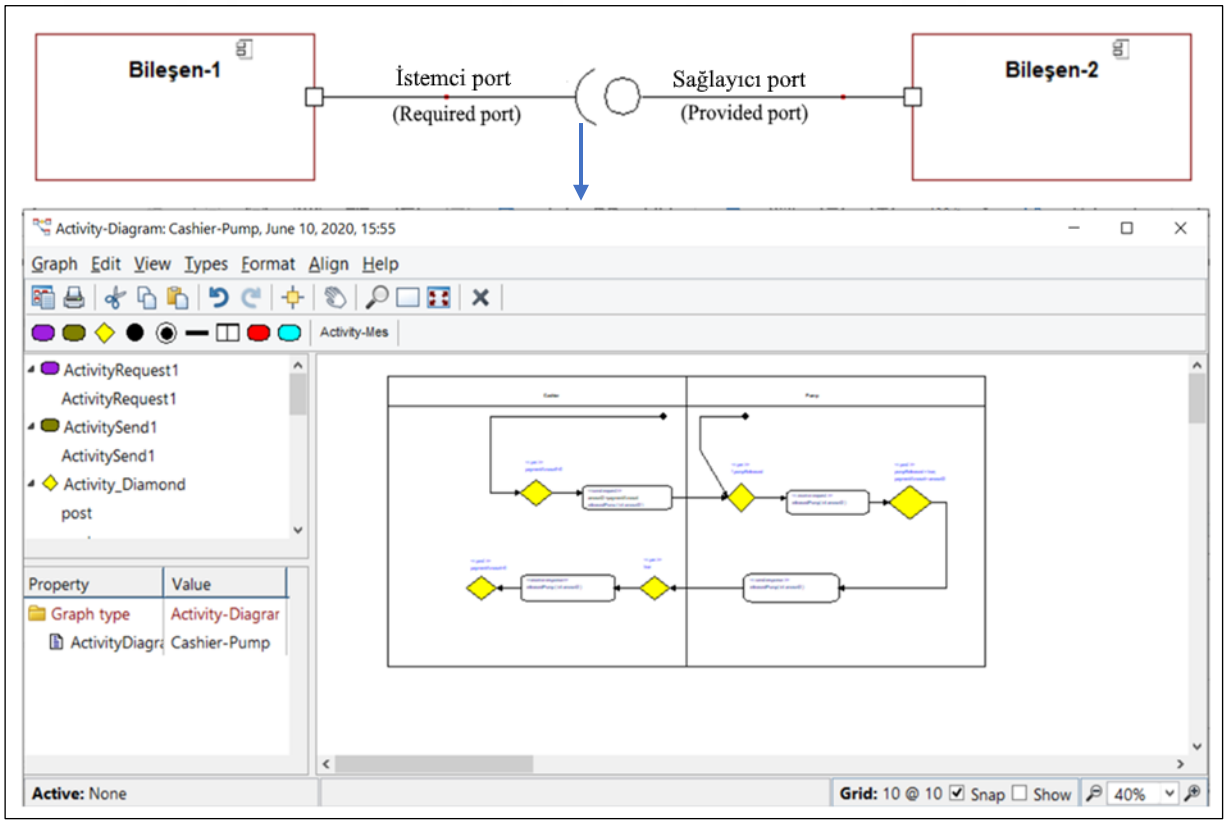
Şekil 2.8'de gösterildiği gibi bir sağlayıcı porttan (*provided port*) birden fazla istek ve yanıt mesajı almak mümkündür. Her bir istek ve yanıt mesaj çiftleri rastgele olarak işleme alınır.



Şekil 2.8: Sağlayıcı port (*provided port*) için çoklu istek ve yanıt mesaj davranışı.

3. AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ SAwUML

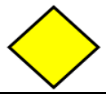
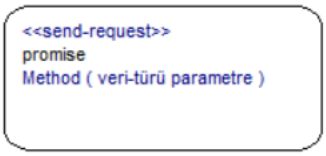
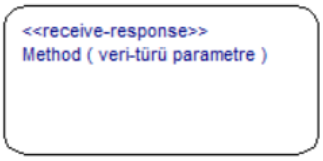
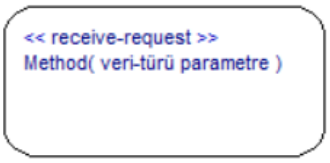
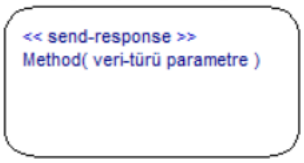
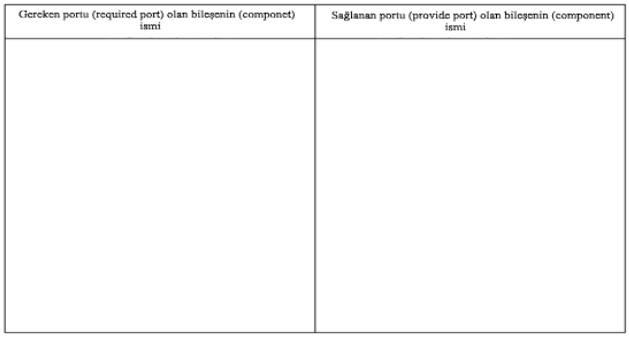
Aktivite diyagram ile genişletilmiş SAwUML'in davranış modeli [40], UML'in aktivite diyagramından esinlenmiştir. Bu sayede, kullanıcı SAwUML'in davranış modeli için sıralamalı diyagramı (*sequence diagram*) veya aktivite diyagramından (*activity diagram*) birini tercih etme imkanı olur. SAwUML'in davranış modelini temsil eden bileşen diyagramında, istemci porta (*required port*) sağ tıklayınca Şekil 3.1'de görüldüğü gibi SAwUML'in aktivite diyagramına erişilir.



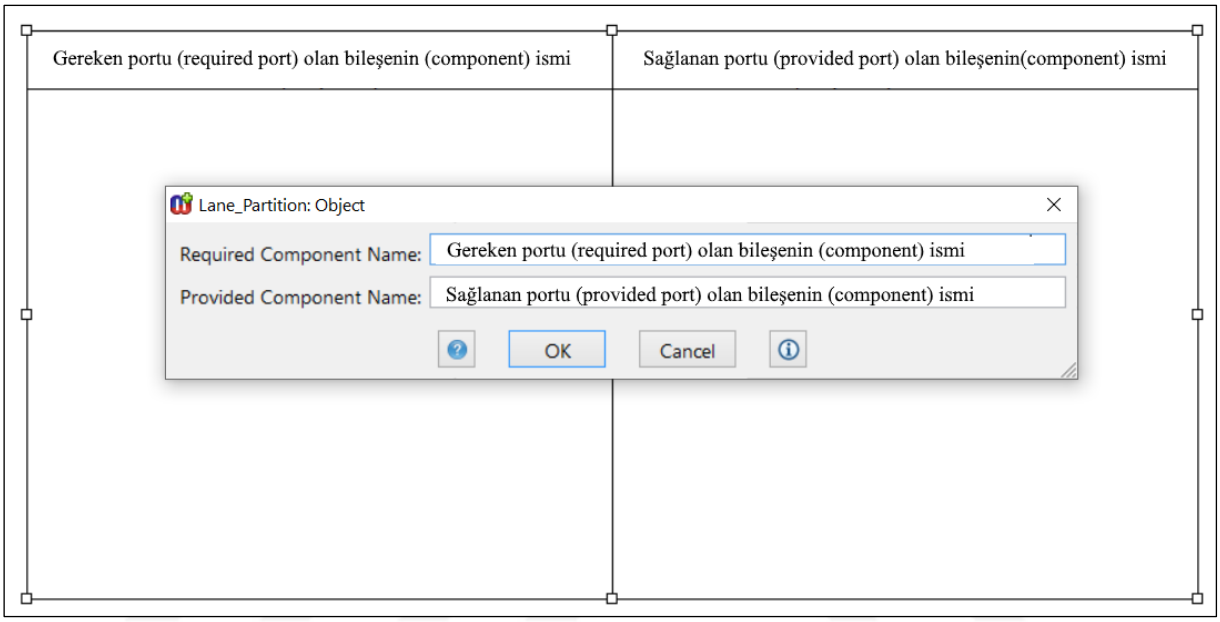
Şekil 3.1: Bileşen diyagramından aktivite diyagramına erişim.

SAwUML aktivite diyagramının notasyon seti UML'in aktivite diyagram notasyon setinden esinlenmiştir. Tablo 3.1'de bu notasyon seti gösterilmiştir.

Tablo 3.1: SAwUML aktivite diyagramı notasyon seti.

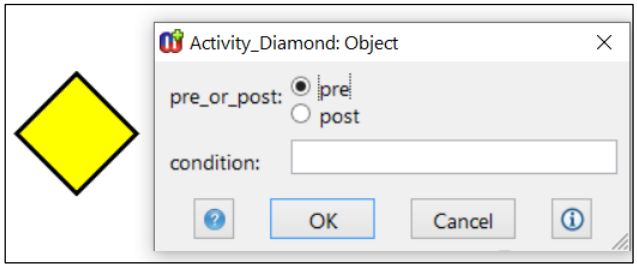
Notasyon	Görsel Sembol
ön/sonraki koşul (*pre/post condition*)	
bileşenin (*component*), istemci porttan (*required port*) istek gönderme aktivitesi	
bileşenin (*component*), istemci porttan (*required port*) yanıt alma aktivitesi	
bileşenin (*component*), sağlayıcı porttan (*provided port*) istek alma aktivitesi	
bileşenin (*component*), sağlayıcı porttan (*provided port*) yanıt gönderme aktivitesi	
bileşen (*component*) kulvarı	
başlangıç son	

SAwUML aktivite diyagramı oluşturulmasında öncelikle bileşen kulvarı kullanılır. Bileşen kulvarında istemci portu (*required port*) olan bileşenin aktiviteleri sol tarafta; sağlayıcı portu (*provided port*) olan bileşenin aktiviteleri sağ tarafta yer alır (Şekil 3.2).



Şekil 3.2: Bileşen (*component*) kulvarı ve özellikleri.

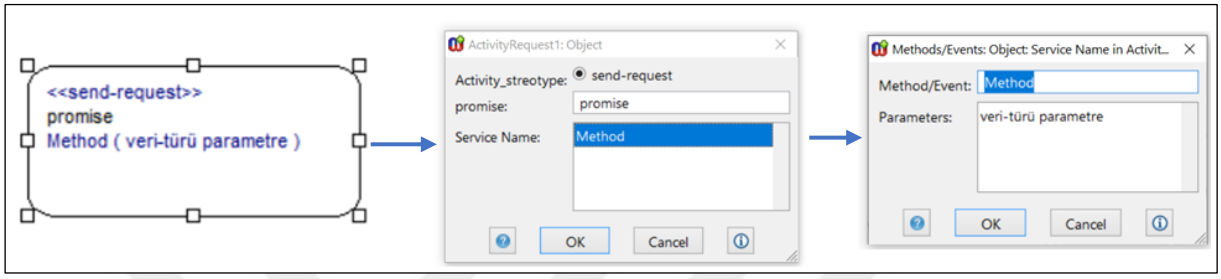
Bileşenin (*component*) portu vasıtasıyla bağlı olduğu diğer bileşene (*component*) metot çağrıları yaptığında ön /sonraki koşul olarak Şekil 3.3'deki sembol kullanılır. Tasarımcı, koşulun türünü (ön veya sonraki) ve koşulu "eşkenar dörtgen" şeklindeki sembolün üzerine tıklayarak açılacak diyalog kutusunda belirtir.



Şekil 3.3: Ön/sonraki koşul notasyonu ve özellikleri.

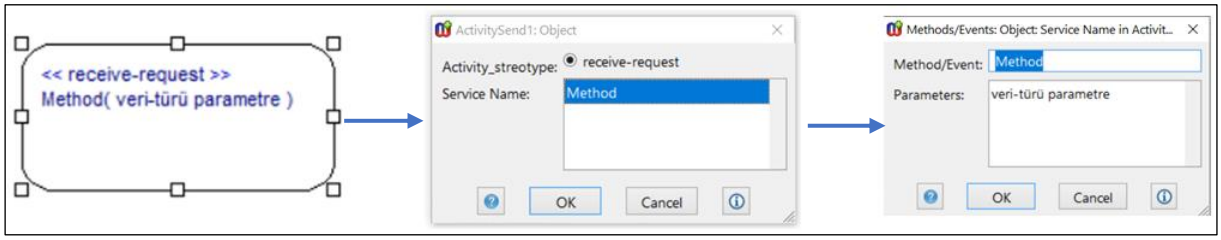
İstek göndermek için istemci portu (*required port*) olan bir bileşenin (*component*) aktivitesine Şekil 3.4'de gösterildiği gibi notasyonun üzerine tıklayınca açılan diyalog kutularına, "promise"

özelliğine metot parametrelerinin değerlerinin atanması olmak üzere, metot ismi ve parametreleri eklenir. Eğer, Şekil 3.3 de gösterilen ön/sonraki koşul (*pre/post condition*) notasyonunda ön (*pre*) şartıyla gösterilen koşulu sağlarsa, istek gönderilme aktivitesi gerçekleşir.



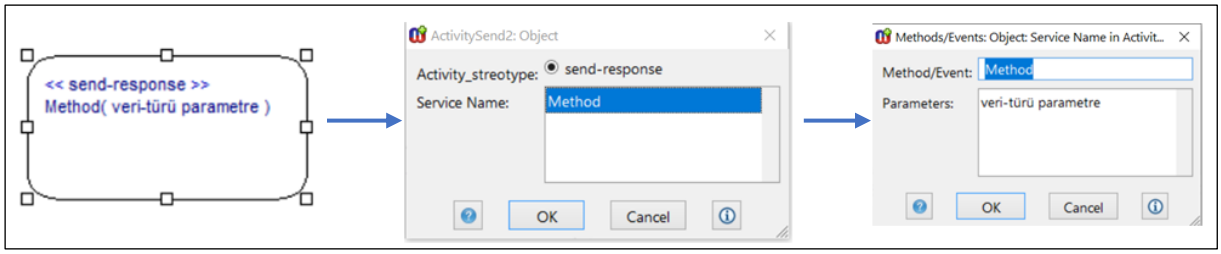
Şekil 3.4: İstek göndermek için istemci portu (*required port*) olan bir bileşenin(*component*) aktivitesi ve özellikleri.

İstemci porttan (*required port*) istek gönderilme aktivitesi gerçekleştirilirse, sağlayıcı porttan (*provided port*) o isteğin alınma aktivitesi, sağlayıcı porttaki (*provided port*) ön/sonraki koşul (*pre/post-condition*) notasyonunda ön (*pre*) şartıyla gösterilen koşulları sağlarsa gerçekleşir. Şekil 3.5'de sağlayıcı porttaki (*provided port*) aktivitenin özellikleri gösterilmektedir.



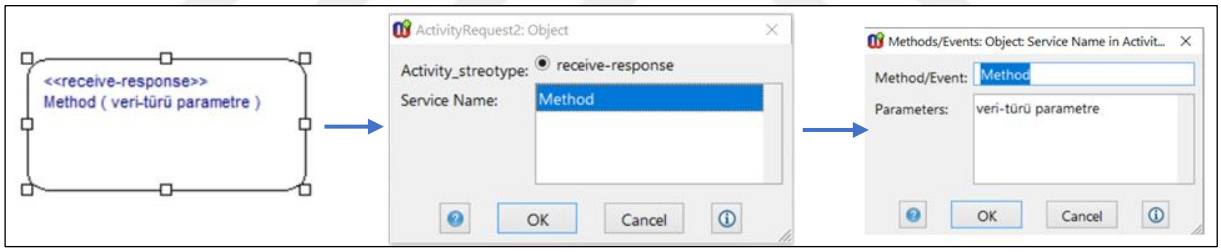
Şekil 3.5: İstek almak için sağlayıcı portu (*provided port*) olan bir bileşenin (*component*) aktivitesi ve özellikleri.

Sağlayıcı portta (*provided port*) aktivite gerçekleşirse, sağlayıcı porta (*provided port*) bağlı olan bileşendeki verilere (*data*) gerekirse yeni değerler atanması ön/sonraki koşul (*pre/post-condition*) notasyonunda sonraki (*post*) şartıyla gösterilir; daha sonra da Şekil 3.6'da gösterilen yanıt gönderme aktivitesi gerçekleştirilir. Bu yanıt yollama aktivitesi ile birlikte varsa ön/sonraki koşul (*pre/post-condition*) notasyonunda sonraki (*post*) şartında metodun dönüş değeri belirtilir.



Şekil 3.6: Yanıt göndermek için sağlayıcı portu (*provided port*) olan bir bileşenin (*component*) aktivitesi ve özellikleri.

Sağlayıcı porttan (*provided port*) gönderilen yanıt aktivitesi gerçekleşirse, istemci porta (*required port*) alınan yanıt, öncelikle ön/sonraki koşul (*pre/post-condition*) notasyonunda sonraki (*pre*) şartları sağlayıp sağlamadığı kontrol edilir. Şartları sağlıyorsa, Şekil 3.7'de gösterilen yanıt alma aktivitesi, varsa ön/sonraki koşul (*pre/post-condition*) notasyonunda sonraki (*post*) şartında gösterilen istemci porta (*required port*) bağlı olan bileşenin son durumunu gösteren notasyonla birlikte gerçekleştirilir.

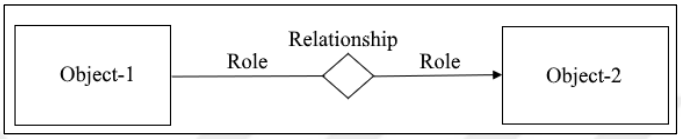


Şekil 3.7: Yanıt almak için istemci portu (*required port*) olan bir bileşenin (*component*) aktivitesi ve özellikleri.

Tablo 3.1'de gösterilen başlangıç ve son notasyonları, aktivitenin başlangıcını ve sonunu göstermektedir.

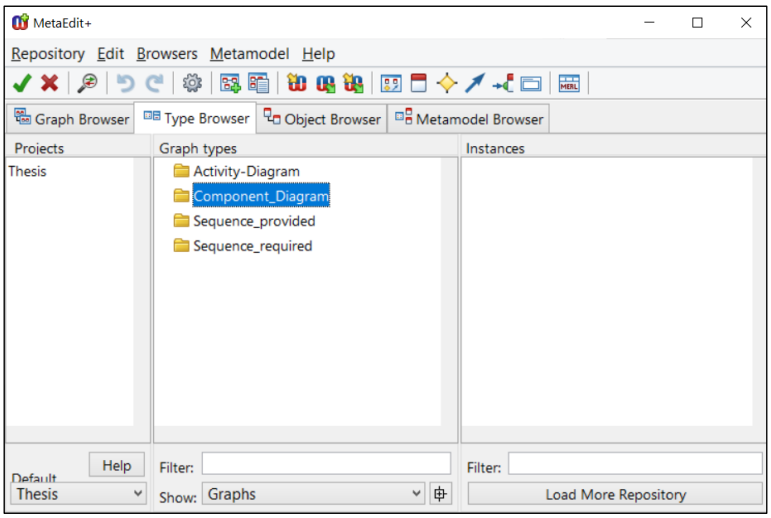
4. METAEDIT+

SAwUML için oluşturduğumuz meta-model ve dönüştürücüleri, MetaEdit+ yazılım aracından faydalanılarak oluşturulmuştur. MetaEdit+ model editöründe oluşturulan modeller, Şekil 4.1'de görüldüğü gibi "object", "relationship" ve "role" isimli parçaların bağlantıları ile oluşturulmuştur.

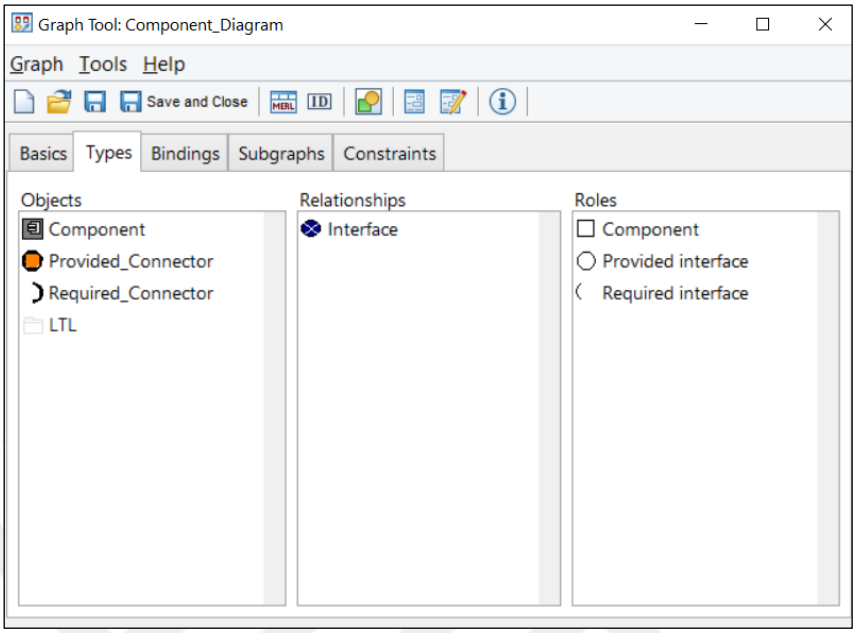


Şekil 4.1: MetaEdit+ bileşenleri ve bağlantıları.

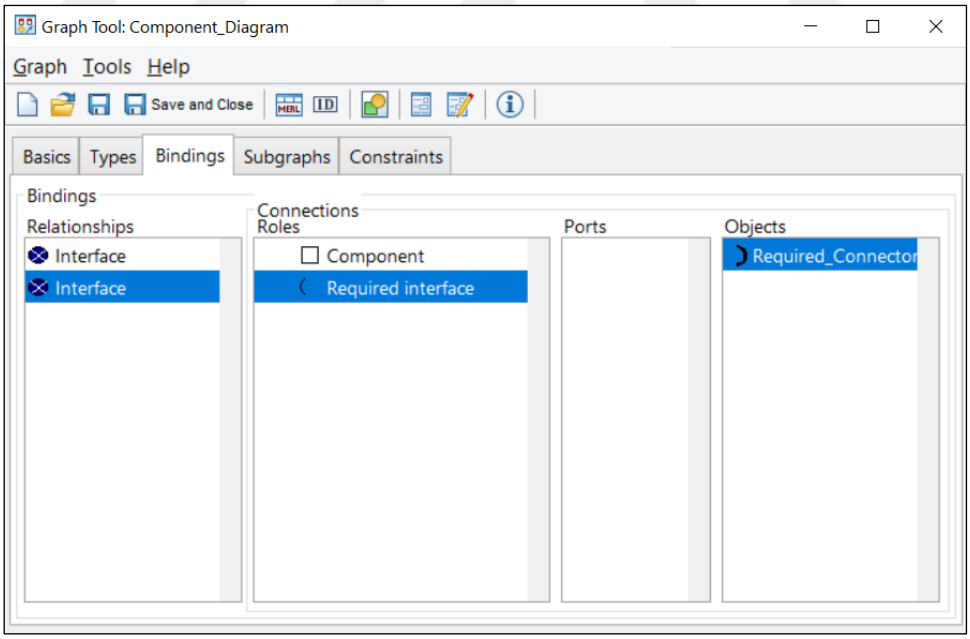
Kullanıcı, MetaEdit+ aracında, öncelikler grafik türlerini belirler (Şekil 4.2). Sonra her bir grafik türü için object, relationship ve role tiplerini oluşturur (Şekil 4.3); daha sonra her bir object'in diğer object'ler ile olan bağlantılarını temel olarak Şekil 4.1'de de gösterilen görülen kurala göre belirler (Şekil 4.4).



Şekil 4.2: MetaEdit+'da oluşturulan grafik tipleri.

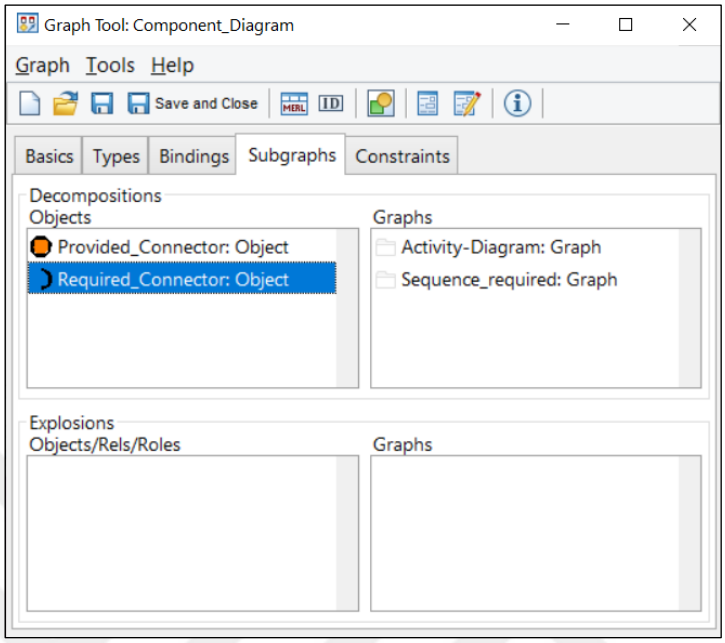


Şekil 4.3: Component_Diagram grafiğinin object, relationship ve role tipleri.



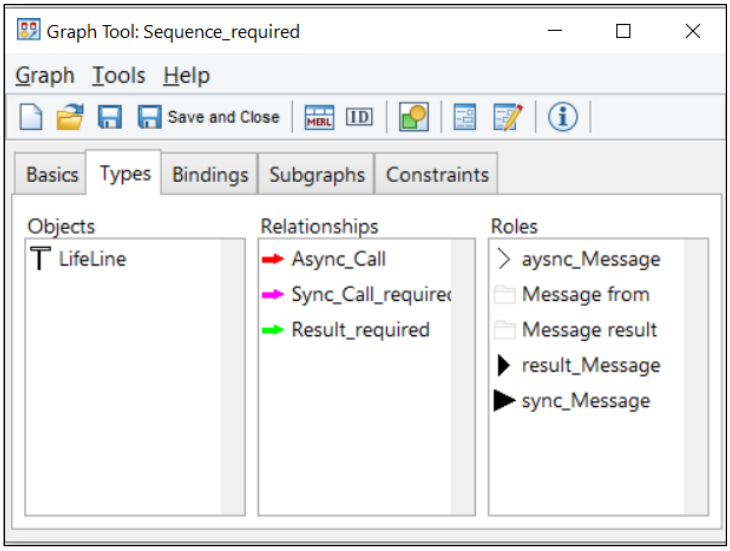
Şekil 4.4: Component_Diagram grafiğini oluşturan parçaların birbirleriyle olan bağlantıları.

Subgraphs sekmesinden, ilgili object'lerden alt grafik olarak istenilen bir veya birden fazla grafik bağlanır (Şekil 4.5).

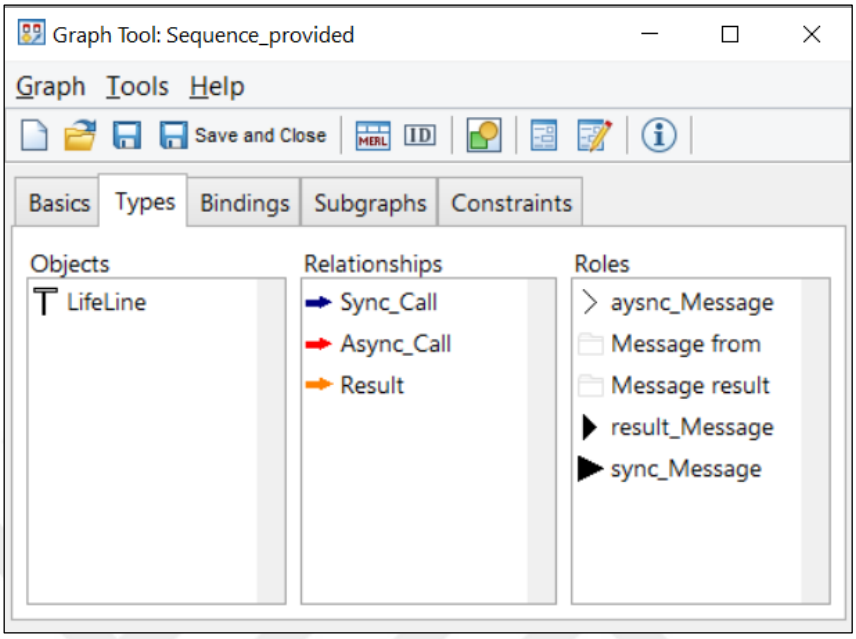


Şekil 4.5: Component_Diagram grafiğindeki object'lerden ulaşılabilecek alt grafik türleri.

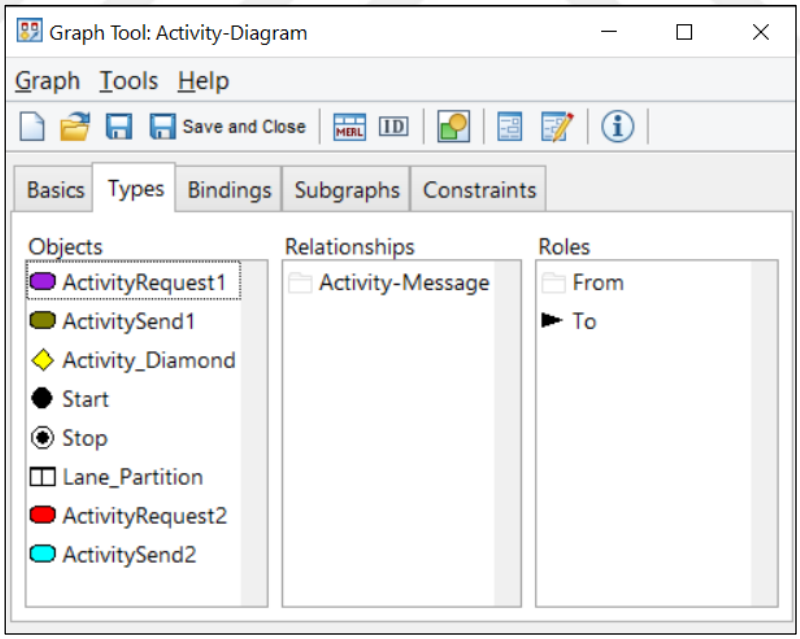
Alt grafikleri (*subgraph*) oluşturan diğer grafiklerin object, relationship ve role tipleri aşağıdaki Şekil 4.6, Şekil 4.7 ve Şekil 4.8'de gösterilmiştir.



Şekil 4.6: Sequence_required grafiğinde kullanılan bileşenlerin tipleri ve isimleri.



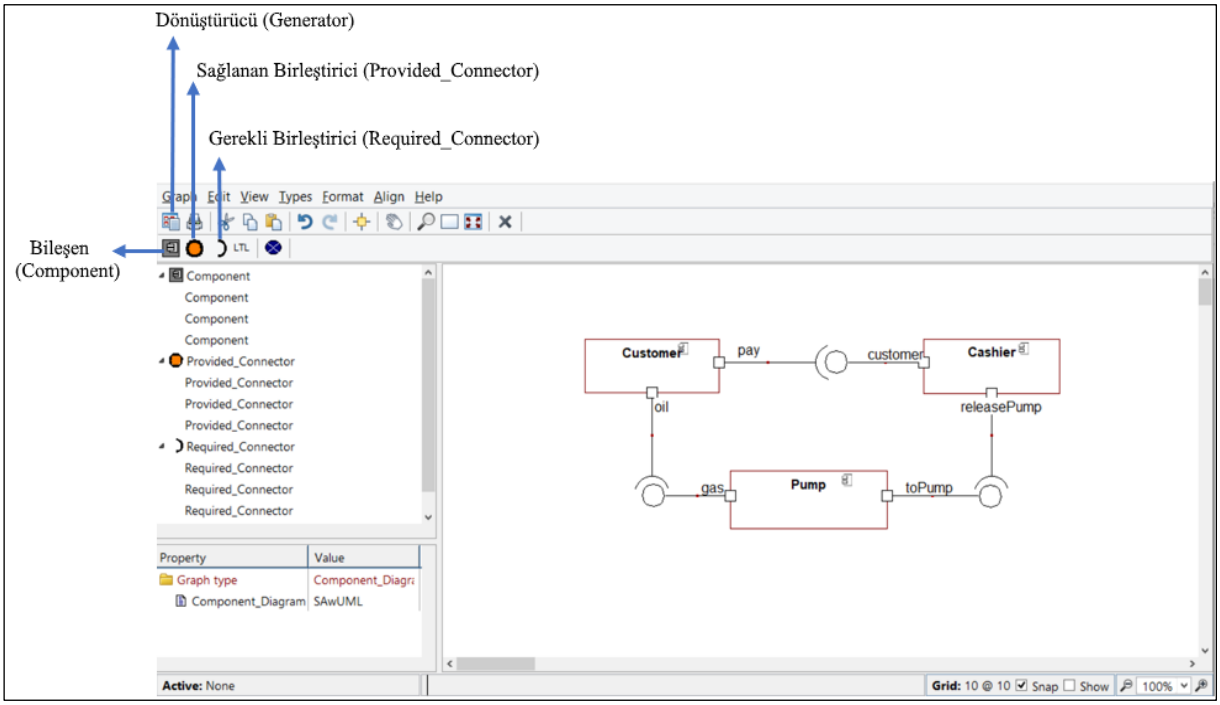
Şekil 4.7: Sequence_provided grafiğinde kullanılan bileşenlerin tipleri ve isimleri.



Şekil 4.8: Activity-Diagram grafiğinde kullanılan bileşenlerin tipleri ve isimleri.

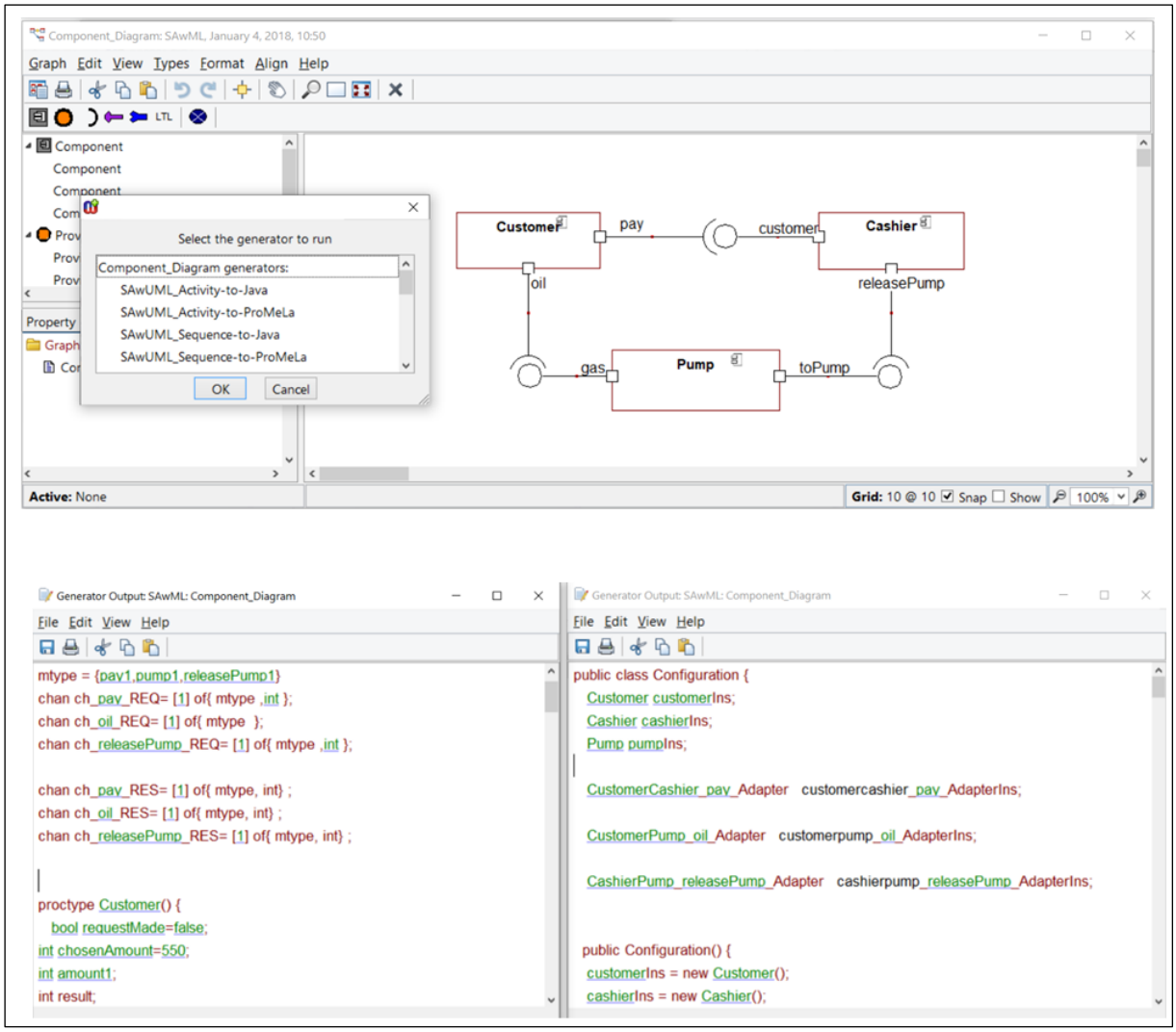
Bir grafik tipi için bileşenleri ve birbirleri ile ilgili bağlantıları oluşturulduktan sonra MetaEdit+ editöründe o grafik türü için diyagramlar çizilir. MetaEdit+ editöründe, araç çubuğundaki

simgelere tıklayıp, tekrar editörün çizim alanına tıklayınca ilgili object oluşur. İki object arasında bağlantı çizmek için (object'ler arasındaki tanımlı bağlantıya göre), object'e sağ tıklayıp connect opsiyonu seçilir, diğer object'e tıklayınca bağlantı kurulmuş olur. Şekil 4.9'da MetaEdit+ editöründe, Componet, Required_Connector ve Provided_Connector isimli object'leri kullanılarak oluşturulmuş diyagram gösterilmektedir.



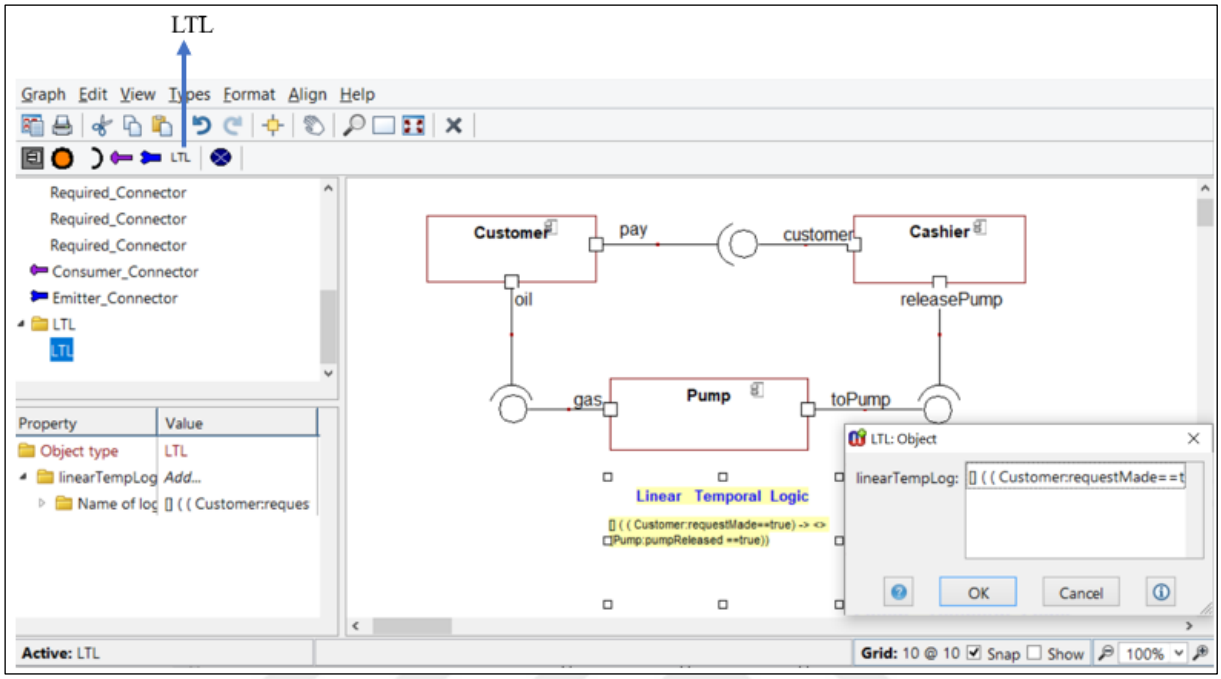
Şekil 4.9: Component_Diagram isimli grafik tipinde oluşturulmuş diyagram.

MetaEdit+ aracında, dönüştürücü (*generator*) için MERL yazılım dili kullanılır. MERL yorumlanır (*scripting*) bir dildir. MetaEdit+ editöründe oluşturulmuş her bir bileşenine, bileşenler arası bağlantılara ve bileşenlerin ve bağlantılarının sahip oldukları verilere (*data*) MERL yazılım dil sayesinde erişilebilir. Şekil 4.9'da gösterilen dönüştürücü (*generator*) simgesine tıklandığında, MetaEdit+ editöründe oluşturulmuş SAwUML modelinin ProMeLa veya java diline dönüştürülmesi seçeneğinin olduğu bir diyalog kutusu çıkar. Kullanıcı bu diyalog kutusundan istediği dönüştürücüyü seçerek çalıştırır (Şekil 4.10).



Şekil 4.10: Sıralama diyagramlı (*sequence diagram*) SAwUML modelinin, java ve ProMeLa dönüştürülmelerinin sonuçları.

LTL simgesine tıklayınca, grafik editor alanında bir diyalog kutusu açılır ve kullanıcı tanımlı doğrusal zamansal mantık (*Linear Temporal Logic -LTL*) özellikleri girilebilir (Şekil 4.11). ProMeLa dönüştürücüyü çalıştırdıktan sonra, LTL özellikleri ProMeLa dilindeki LTL sözdizimine göre çevrilir ve SAwUML modelden elde edilen ProMeLa modele gömülü olarak eklenir.



Şekil 4.11: Kullanıcı tanımlı bir LTL örneği.

5. PROMELA DÖNÜŞTÜRÜCÜ ALGORİTMASI

SAwUML ile tasarlanan sistemlerin özelliklerini kapsamlı bir analizi yapmak için SPIN model denetleyicisinin ProMeLa diline dönüştüren Liste 5.1 ‘deki algoritma kullanıyor.

Liste 5.1: ProMeLa dönüştürücü algoritması.

```
1   Model-ProMeLa
2    mytype = {
3     ∀  bileşen ∈ Model
4       ∀  port ∈ bileşen.port_tipleri
5          ∀  metot ∈ port
6               metot_ismi
7    }
8
9    ∀  bileşen ∈ Model
10     ∀ port ∈ bileşen.istemci_port
11       chan ch_port.ismi_REQ = [port.bellek_boyutu] of {…};
12       chan ch_port.ismi_RES = [port.bellek_boyutu] of {…};
13
14    ∀  bileşen ∈ Model
15     proctype bileşen.isim {
16        ∀  veri ∈ bileşen
17            veri.tip  veri.isim = veri.varsayılan_değer;
18        ∀  parametre ∈ bileşen
19            parametre.tip  parametre.isim = parametre.varsayılan_değer;
20
21        ∀ port ∈ bileşen.port
22          ∀ metot ∈ port
23             ∀ parametre ∈ metot
24               parametre.tip  parametre.isim;
25
26         ∀ port ∈ bileşen.istemci_port
27          ∀ metot ∈ port
28            bool  responseWaiting_port.ismi,metot.ismi =false;
29            int  result_port.ismi,metot.ismi;
```

Liste 5.1: ProMeLa dönüştürücü algoritması (devam).

```
30
31        do
32          ∀ port ∈ bileşen.port
33              EĞER  port ∈  sağlayıcı_port
34                ∀ metot ∈ port.metot
35                  :: ch_port.ismi_REQ ? metot.ismi , metot.parametreler →
36                  if
37                    ∀ m_davranış ∈ metot.davranış_listesi
38                      ::  m_davranış.pre_istek→
39                          m_davranış.post_istek;
40                          m_davranış.post_yanıt,result;
41                          ch_port.ismi_RES ! metot.ismi, result;
42                      :: ! ( ∀ m_davranış.pre_istek) →
43                          ch_port.ismi_REQ ! metot.ismi , metot.parametreler;
44                  fi;
45                EĞER  port  ∈  istemci_port
46                    ∀ metot ∈ port.metot
47                      ∀  m_istek_davranış ∈ metot.istek_davranış_listesi
48                          :: m_istek_davranış.pre ∧ responseWaiting==false →
49                              responseWaiting_port.ismi,metot.ismi=true;
50                          ∀ parametre ∈ metot.parametre
51                              parametre=parameter_m_istek_davranış.promise;
52                          ch_port.ismi_REQ ! metot.ismi, metot.parametreler;
53                        :: ch_port.ismi_RES ! metot.ismi, result→
54                        if
55                         ∀  m_yanıt_davranış ∈ metot.yanıt_davranış_listesi
56                          :: m_yanıt_davranış.pre → yanıt.post;
57                          :: ! ( ∀ m_yanıt_davranış.pre) →assert(false);
58                        fi;
59                        responseWaiting=false;
60            od
61          }
```

Bu algoritmaya göre her bir süreç (*process*) bir bileşeni (*component*) temsil etmektedir. Bu süreçler, koşut zamanlı olarak birbirleriyle etkileşimli olarak çalıştırılıyor. Satır 2-7 arasında, mytpe operatörüne bileşenlerin metot isimleri ekleniyor. Bileşenlerin birbirleriyle

mesajlaşmalarında (istek/yanıt) kullanılan kanalların (*channels*), ilk parametresi olarak metot ismi kullanılıyor; bu sayede bir portta birden fazla mesajlaşma olduğunda, mesajların karışmaması sağlanıyor. Bileşenlerin her istemci portunda (*required port*), ProMeLa kanalları (*channels*) çifti oluşturuluyor (satırlar 9-12). Bu kanallar çifti, istemci port (*required port*) ile onun karşısındaki sağlayıcı port (*provided port*) arasındaki mesajlaşmayı sağlıyor. Kullanıcı her bir kanal çiftinin bellek boyutunu belirleyebilir. Bellek boyutunun sıfır (0)'dan büyük olması durumunda kanalın, belirtilen boyuta kadar mesajları biriktirebileceği belirtilir (asenkron mesajlaşma); bellek boyutunun sıfır olması durumunda (senkron mesajlaşma) mesajların kanaldan doğrudan iletilmesini sağlar (*a randevouz channel*). Her bir bileşenin (*component*) verileri, parametreleri ve bağlantı noktalarındaki metotlarının parametrelerin bildirimi yapılır (satırlar 14-24). Ayrıca, bileşenin (*component*) istemci portu varsa (*required port*) responseWaiting ve result değişkenlerinin bildirimi yapılır (satır 26-29). "responseWaiting" değişkeni, bileşen sürecinin (prosess) istemci porttaki (*required port*) metot çağrısının yapılıp yapılmamasını kontrol eder (satır 28). "result" değişkenine, istemci portun (*required port*) karşısındaki sağlayıcı portundan (*provided port*) gelen metot çağrısının yanıt değeri atanır (satır 29). Daha sonra ProMeLa'nın do …od notasyonu içinde sonsuz döngü oluşturulur (satırlar 31-60).

Eğer sonsuz döngünün içindeki bileşenin (component) portu, sağlayıcı port (*provided port*) ise (satırlar 32-44); karşılığı olan istemci portundan (*required port*) gelen mesajın alındığı kontrol edilir (satır 35), eğer alınmışsa if … fi arasındaki satırlar çalıştırılır (satırlar 36-44). Eğer, ön şart sağlanırsa (satır 38), bileşenin verileri güncellenir (satır 39). Dönüş değerinin (result) ataması yapılır (satır 40); daha sonra, bileşenin sağlayıcı portuna (*provided port*) karşı gelen istemci portuna (*required port*), hesaplanan dönüş değeri (result) ile birlikte metot çağrısı kanal (*channel*) vasıtasıyla yanıt olarak gönderilir (satır 41). Eğer, ön şart sağlanmazsa sağlayıcı porta (*provided port*) değerlendirilmek üzere kanal (*channel*) vasıtasıyla tekrar metot çağrısı gönderilir (satırlar 42-43).

Eğer sonsuz döngünün içindeki bileşenin (*component*) portu istemci port (*required port*) ise 45-59 arası satırlar çalışır. Satır 48'deki şartlar sağlanırsa "responseWaiting" değişkeni "true" olarak atanır ve istemci portun karşısındaki sağlayıcı porta metot çağrısı isteğinde bulunulur

(satır 51-52). “responseWaiting” değişkeninin “true” olarak atanması sayesinde, metot çağrısı yanıtının alınıp işlenmesinden önce tekrar metot çağrısının istek olarak gönderilmesi önlenmiş olur. Karşısındaki sağlayıcı portundan (*provided port*) gelen metot çağrısında ön şart sağlanırsa, bileşenin (*component*) durumu güncellenir (satır 56), eğer ön şart sağlanmazsa istemci porta (*required port*) gelen metot çağrı davranışının hatalı olduğuna dair mesaj oluşturulur (satır 57). Bileşenin, sağlayıcı porta (*provided port*) kanal (*channel*) vasıtasıyla metot çağrı isteğinde bulunması için “responseWaiting” değişkeni “false” olarak atanır (satır 59).

6. JAVA DÖNÜŞTÜRÜCÜ ALGORİTMASI

Java dönüştürücü algoritması, Liste 6.1'de gösterildiği gibi dört kısımdan oluşmaktadır. Modeldeki her bir bileşen (*component*), java dönüştürücüsünde bir "bileşen" sınıfı (*class*) olarak tanımlanmaktadır. Birleşenlerin birbirleriyle yaptıkları mesajlaşma için adapter sınıfı (*class*) kullanılmaktadır. Her bir arayüz (interface), bileşenlerin karşılıklı mesajlaşmak için kullandıkları ortak metotlardan oluşmaktadır. İlgili adapter sınıfına karşılık gelen arayüzü implement (*implements*) edilir. Ana sınıf (*class*), içerisinde main metodu bulunan sınıftır, ve java uygulaması bu sınıftan başlatılır.

Liste 6.1: Java dönüştürücü algoritması genel yapısı.

```
1   Arayüzler
2   Adapter Sınıfları
3   Bileşen Sınıfları
4   Ana Sınıf
```

Liste 6.2'de gösterildiği gibi, istemci port (*required port*) ile sağlayıcı port (*provided port*) vasıtasıyla karşılıklı iletişim halindeki her bir bileşen (*component*) çiftinden bir arayüz oluşturulur (satır 4). Bu bileşen çiftinin ortak metodu, arayüz metodunu oluşturur (satır 5).

Liste 6.2: Arayüz algoritması.

```
1   ∀ bileşen ∈ model
2   ∀ metot ∈ bileşen1.istemci_port ∩ bileşen2.sağlayıcı_port
3   ∀ parametre ∈ metot.parametreListesi
4   interface bileşen1-ismi_bileşen2-ismi_Link {
5   public metot.geriDönüşTipi metot (parametre.tipi parametre ismi) throws
    Exception;
6   }
```

Liste 6.3'de gösterildiği gibi, istemci port (*required port*) ile sağlayıcı port (*provided port*) vasıtasıyla karşılıklı iletişim halindeki her bir bileşen (*component*) çiftinden bir adapter sınıfı (*class*) oluşturulur ve aynı çiftin arayüzü implement (*implements*) edilir (satır 5-6). Her bir adapter sınıftaki (*class*), yapıcı (*constructor*) bileşen çiftlerinden sağlayıcı portu olanın

instance'sını parameter olarak alır (satır 9). Implement ettiği arayüzün metodunun içinden, bu instance vasıtasıyla, ilgili bileşenin metodu çağrılır (satır 13-25).

Liste 6.3: Adapter sınıfı algoritması.

```
1   ∀ bileşen ∈ model
2    ∀ istemci_port ∈ bileşen1
3     ∀ metot ∈ bileşen1.istemci_port ∩ bileşen2.sağlayıcı_port
4       ∀ parametre ∈ metot.parametreListesi
5      class bileşen1-ismi_bileşen2-ismi_Required
6            implements bileşen1-ismi_bileşen2-ismi_Link
7     {
8       bileşen2-ismi  bileşen2-ismi_ins;
9       bileşen1-ismi_bileşen2-ismi_Required (bileşen2-isim bileşen_arg ) throws Exception
10     {
11           bileşen2-ismi_ins = bileşen_arg;
12     }
13     EĞER metot.geriDönüşTipi ∈ VOID
14      public metot.geriDönüşTipi metot (parametre.tipi parametre ismi) {
15         try {
16                bileşen2-ismi_ins.metot ( parametre.ismi) ;
17         }
18         catch ( Exception e) { e.PrintStackTrace ( ); }
19       }
20       EĞER metot.geri-dönüş-tipi ∉ VOID
21            public metot.geriDönüşTipi metot (parametre.tipi parametre ismi)
22              throws   Exception {
23                   return bileşen2-ismi_ins.metot ( parametre.ismi) ;
24            }
25    }
```

Liste 6.4'de gösterildiği gibi, modeldeki her bileşenden bir bileşen sınıfı oluşturulur. Her bileşin verileri ve varsa parametreleri değişken olarak atanır (satır 2-8). İstemci portu (*required port*) olan her bileşenin oluşturduğu sınıftaki metotlardan herhangi birinin dönüş tipi int ise satır 9'daki gibi "result" değişkeni atanır. Bileşenin an az bir tane istemci portu (*required port*) varsa "responeWaiting" değişkeni atanır (satır 10-11). Bileşenin her istemci portu (*required port*) için ilgili arayüz değişkeni deklarasyonu yapılır (satır 12-13). Bileşene, metot çağrısı ile gelen her istek için ön şart kontrol edilir (satır 15), eğer ön şartı sağlıyorsa gerekli atamalar yapılır (satır 16). Metodun dönüş değeri "void" değilse, yanıt gönderilir (satır 17-21). Bileşenin, metot çağrısı

ile istek gönderebilmesi için ön şartı kontrol edilir (satır 32). Eğer ön şart sağlanırsa; metodun parametrelerine gerekli atama yapılır ve ilgili adapter sınıfının metodu çağrılır (satır 36). Eğer metot dönüş tipi "void" değilse, "result" değişkenine atanır. Bileşene metot çağrısı gelen yanıt, ön şartı sağlarsa (satır 41), gerekli atamalar yapılır (satır 42-44). Bileşen (*component*), run metodu çağrısı ile her istemci portun metotları sonsuz döngü içerisinde çağrılır (satır 48-58).

Liste 6.4: Bileşen sınıf algoritması.

```
1   ∀ bileşen ∈ model
2    ∀ veri ∈ bileşen.veriListesi
3    ∀ metot ∈ bileşen1.istemci_port ∩ bileşen2.sağlayıcı_port
4     ∀ parametre ∈ metot.parametreListesi
5    ∀ port ∈ bileşen.port
6     class bileşenismi {
7      veri.tipi veri.ismi =  veri.varsayılanDeğer;
8      parametre.tipi parametre.ismi = parametre.varsayılanDeğer;
9      EĞER   ∃  metot.geriDönüşTipi ∈ int
10           ∧  ∀ port ∈ bileşen.istemci_port
11      private   int result = 0;
12     EĞER  ∃  port ∈ bileşen.istemci_port
13       private boolean responseWaiting = false;
14     EĞER  port ∈ bileşen.istemci_port
15      bileşen1-ismi_bileşen2-ismi_Link    bileşen1-ismi_bileşen2-ismi_LinkInstance;
16     EĞER  port ∈ bileşen.sağlayıcı_port
17      ∀ metot ∈ port.metot
18     public metot.geriDönüşTipi metot (parametre.tipi parametre ismi) throws
    Exception
19      {
20          ∀ mdavranış ∈ metot.davranışListesi
21            if ( mdavranış.preistek) {
22               mdavranış.postistek
23            }
24           EĞER metot.geriDönüşTipi ∉ VOID
25             return mdavranış.postyanıt,result;
26      }
27    EĞER  port ∈ bileşen.istemci_port
28     public metot.geriDönüşTipi metot (parametre.tipi parametre ismi) throws
    Exception
29     {
30          ∀ mistek_davranış ∈  metot.davranışListesi
```

Liste 6.4: Bileşen sınıf algoritması (devam).

```
31          if ( m_istek_davranış && ! responseWaiting) {
32              ∀ parametre ∈ metot.parametre
33                  parametre=parameter_m_istek_davranış.promise;
34                  EĞER metot.geriDönüşTipi ∈ VOID
35                    bileşen1-ismi_bileşen2-ismi_LinkInstance.metot (parametre.ismi);
36                  EĞER metot.geriDönüşTipi ∉ VOID
37                    result =
38                       bileşen1-ismi_bileşen2-ismi_LinkInstance.metot (parametre.ismi);
39          }
40           ∀ m_yanıt_davranış ∈ metot.yanıtDavranışListesi
41            if ( m_yanıt_davranış.pre) {
42                  m_yanıt_davranış.post;
43            }
44           responseWaiting = false;
45           EĞER metot.geriDönüşTipi ∉ VOID
46             return result;
47     }
48    public void run ( ) {
49        while (true) {
50          ∀  port ∈ bileşen.sağlayıcı_port
51            ∀ metot ∈ bağlantı_noktası.metot
52              try {
53                  metot (parameter.initialValue) ;
54              }
55              catch ( Exception e) { e.PrintStackTrace ( ); }
56        }
57     }
58  }
```

Liste 6.5'de gösterildiği gibi, Configuration isimli ana sınıfda (class), öncelikle her bir bileşen sınıfının (*class*) örnekleri (*instance*) oluşturulur (satır 7). İstemci portu (*required port*) olan bileşen (*component*) ile karşısında sağlayıcı portu (*provided port*) olan bileşen (component) çiftinden oluşan adapter sınıfının (class) örnekleri (*instance*) oluşturulur, parametre olarak sağlayıcı portu (*provided port*) olan bileşenin (*component*) örneği (*instance*) konulur (satır 9-10). İstemci portu (*required port*) olan bileşenin (component) örneğinin (*instance*) arayüz değişkenine adapter sınıfın (*class*) örneği (*instance*) atanır (satır 11).

Her bir istemci portu (*required port*) olan bileşenin (*component*) örneği (*instance*), start metodunu çağırarak bileşenin Liste 6.4 satır 48'deki run metodunun çalıştırılması sağlanır.

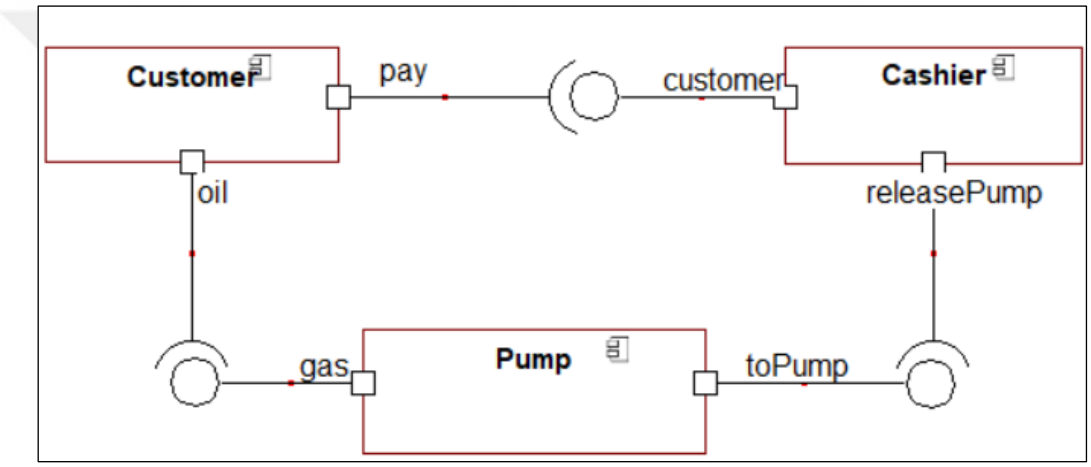
Liste 6.5: Ana sınıf algoritması.

```
1   public class Configuration {
2   ∀ bileşen ∈ model
3    bileşen_ismi  bileşen_ismi_Ins;
4    ∀ metot ∈ bileşen1.istemci_port ∩ bileşen2.sağlayıcı_port
5      bileşen1-ismi_bileşen2-ismi_Required    bileşen1-ismi_bileşen2-ismi;
6     public Configuration ( ) {
7       bileşen_ismi_Ins = new bileşen_ismi ( );
8       try {
9            bileşen1-ismi_bileşen2-ismi =
10               new  bileşen1-ismi_bileşen2-ismi_Required ( bileşen2-ismi_Ins );
11           bileşen1-ismi_Ins.bileşen1-ismi_bileşen2-ismi_LinkInstance =
12                                                   bileşen1-ismi_bileşen2-ismi;
13     }
14     catch ( Exception e) { e.PrintStackTrace ( ); }
15     ∀ bağlantı_noktası  ∈ bileşen1.istemci_port
16       bileşen1-ismi_Ins.start ( ) ;
17    }
18    public static void main (String args[ ] )  throws  Exception {
19       new Configuration ( );
20    }
21  }
```

7.VAKA ANALİZLERİ

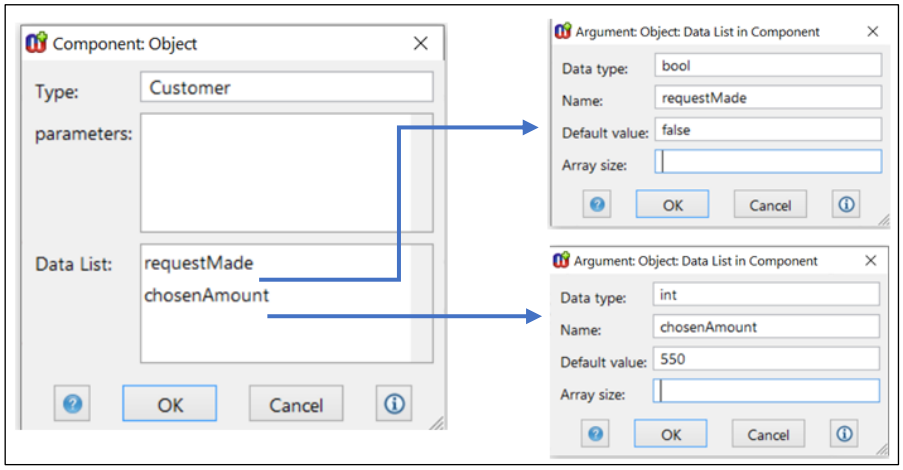
7.1 GAZ İSTASYONU MODELİ

Bir müşterili gaz istasyon modeli [41] birbirleri ile etkileşimli üç bileşenden (*component*) oluşuyor. Bu bileşenler *Customer, Pump* ve *Cashier* isimli bileşenlerdir. *Customer*, yakıtı *Pump*'dan, eğer *Cashier* ödeme yaparsa alabilmekte; *Pump* ödemenin yapılıp yapılmadığını *Cashier*'den gelen bilgiden almaktadır. Şekil 7.1'de gaz istasyon modeli gösterilmektedir.

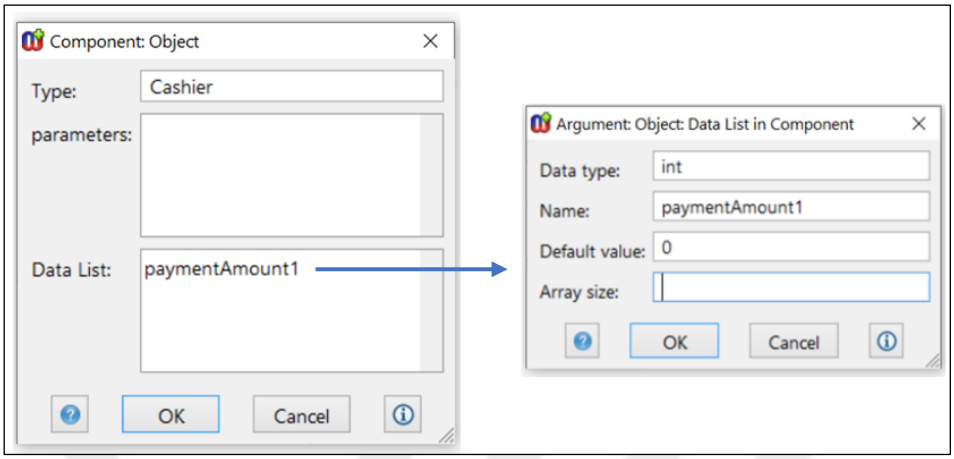


Şekil 7.1: SAwUML'de oluşturulan bir *Customer*'ı olan gaz istasyon modeli.

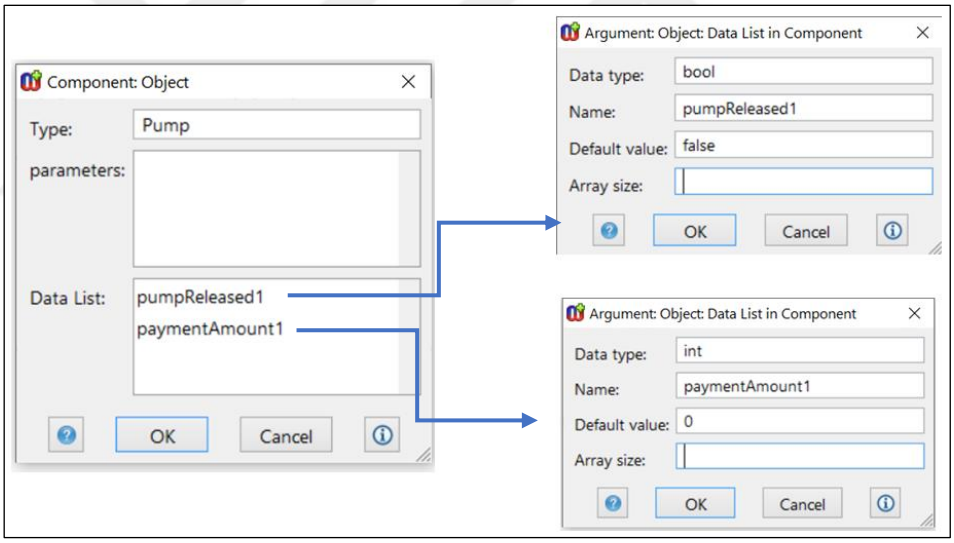
Bileşenlerin verileri (*data*) ve özellikleri aşağıdaki resimlerde gösterilmiştir.



Şekil-7.2: *Customer* bileşeninin verileri ve özellikleri.

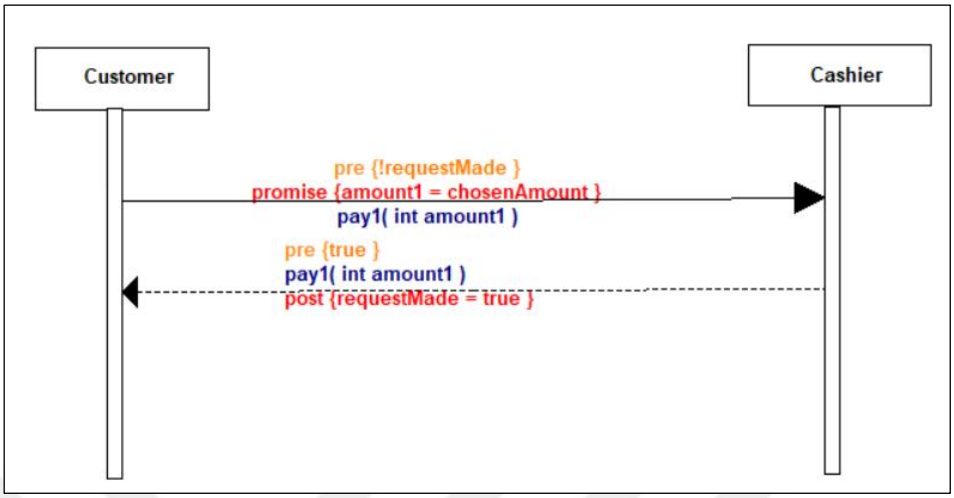


Şekil 7.3: *Cashier* bileşeninin verileri ve özellikleri.



Şekil 7.4: *Pump* bileşeninin verileri ve özellikleri.

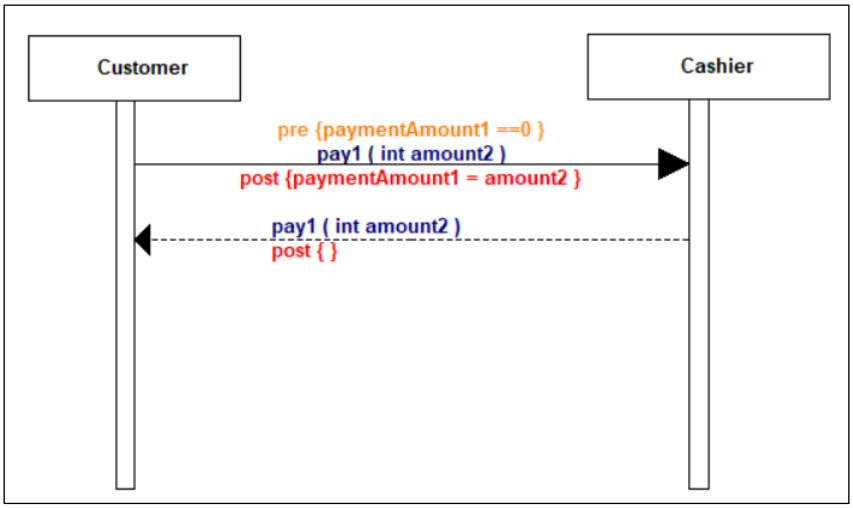
Customer'ın *Cashier* ile ilişkisi, *Customer*'ın "*pay*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.5'de gösteriliyor.



Şekil 7.5: *Customer*'ın *Cashier*'a bağlandığı istemci portun (*required port*) özellikleri.

Öncelikle, *Customer*'ın *Cashier*'den istekte bulunabilmesi için gerekli ön şartı (*!requestMade*) sağlaması gerekiyor. Bu ön şart sağlandığında, ilgili metodun parametresine atama gerçekleştirilir (*amount1=chosenAmount*). Daha sonra *Cashier*'e metot çağrısı yapılır. *Cashier*'den alınan cevapta (kesikli ok ile gösterilen), ön şart (*true*) kontrol edilir, eğer bu şart sağlanıyorsa, son koşula (*requestMade=true*) göre atama yapılır.

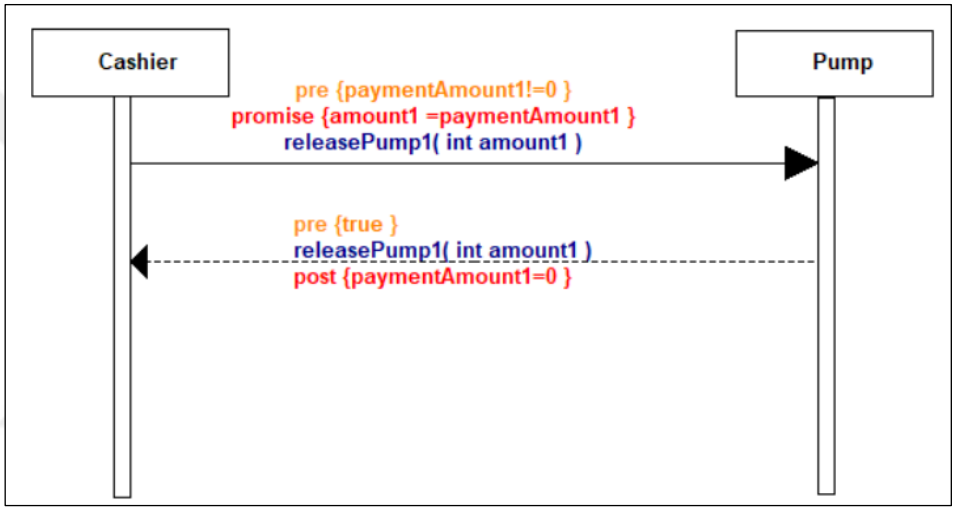
Cashier'in *Customer* ile ilişkisi, *Cashier*'in "*customer*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.6'da gösteriliyor.



Şekil 7.6: *Cashier*'in *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri.

Customer'dan *Cashier*'e istek geldiğinde önce ön şart (*paymentAmount1==0*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*paymentAmout1=amount2*) atama yapılır. *Customer*'a, (kesikli ok ile gösterilen) *pay1* metot çağrısı ile dönüş değeri yollanır. Burada dönüş değeri olmadığı için boş bırakılmıştır.

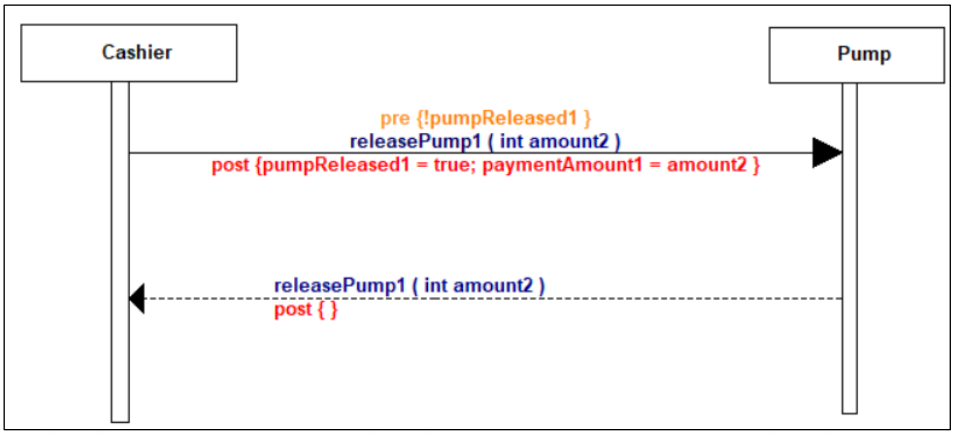
Cashier'in *Pump* ile ilişkisi, kasiyerin "releasePump" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.7'de gösteriliyor.



Şekil 7.7: *Cashier*'in *Pump*'a bağlandığı istemci portun (*required port*) özellikleri.

Cashier'in *Pump*'dan istekte bulunabilmesi için gerekli ön şartı (*paymentAmount1!=0*) sağlaması gerekiyor. Bu ön şart sağlandığında, ilgili metodun parametresine atama gerçekleştirilir (*amount1=paymentAmount1*). Daha sonra *Cashier*'e metot çağrısı yapılır. Cashier'den alınan cevapta (kesikli ok ile gösterilen), ön şart (*true*) kontrol edilir, eğer bu şart sağlanıyorsa, son koşula (*paymentAmount1=0*) göre atama yapılır.

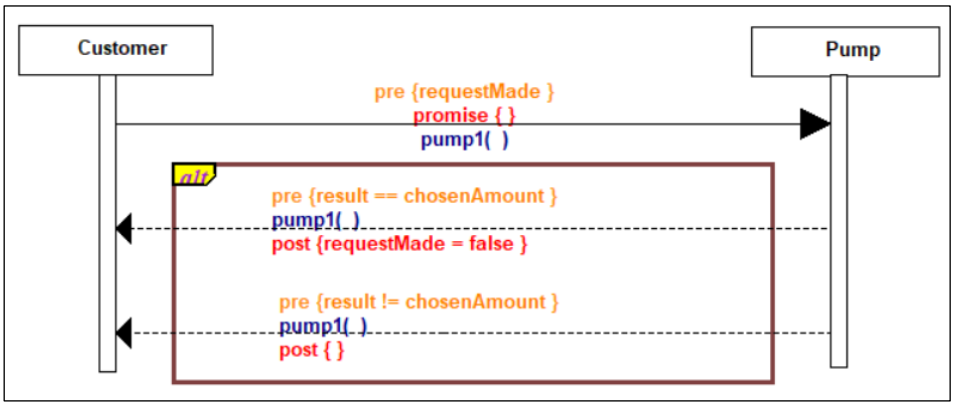
*Pump"*ın *Cashier* ile ilişkisi, *Pump*'ın "*toPump*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.8'de gösteriliyor.



Şekil 7.8: *Pump*'ın *Cashier*'e bağlandığı sağlayıcı portunun (*provided port*) özellikleri.

Cashier'den *Pump*'a istek geldiğinde önce ön şart (*!pumpReleased1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre *(pumpReleased1=true;paymentAmout1=amount2*) atama yapılır. *Customer*'a, (kesikli ok ile gösterilen) *releasePump1* metot çağrısı ile dönüş değeri yollanır. Burada dönüş değeri olmadığı için boş bırakılmıştır.

Custmer'ın *Pump* ile ilişkisi, *Customer*'ın "*oil*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.9'da gösteriliyor.

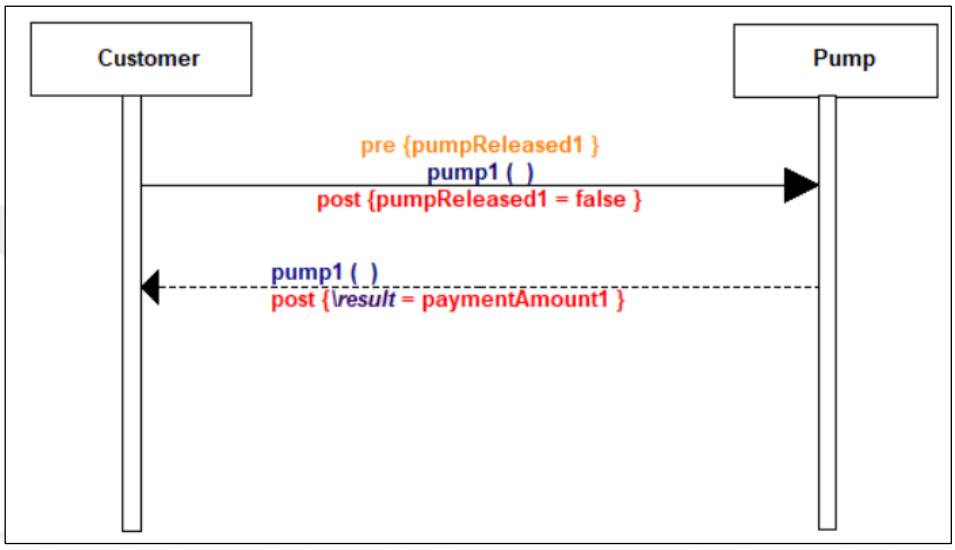


Şekil 7.9: *Customer*'ın *Pump*'a bağlandığı istemci portun (*required port*) özellikleri.

Customer'ın *Pump*'dan istekte bulunabilmesi için gerekli ön şartı (*requestMade*) sağlaması gerekiyor. Bu ön şart sağlandığında, ilgili *pump1* metot çağrısı *Pump*'a yapılır. *Pump*'tan alınan cevapta (kesikli ok ile gösterilen), ön şartlarının sağlayıp sağlamadığı kontrol edilir. Eğer

(*result==chosenAmount*) şartı sağlanıyorsa, son koşula (*requestMade=false*) göre atama yapılır; eğer (*result!=chosenAmount*) şartı sağlanıyorsa son koşulda bir atama yapılmaz.

*Pump"*ın *Customer* ile ilişkisi, *Pump*'ın "*gas*" isimli sağlayıcı porta sağ tıklama ile açılan diyagram Şekil 7.10'da gösteriliyor.

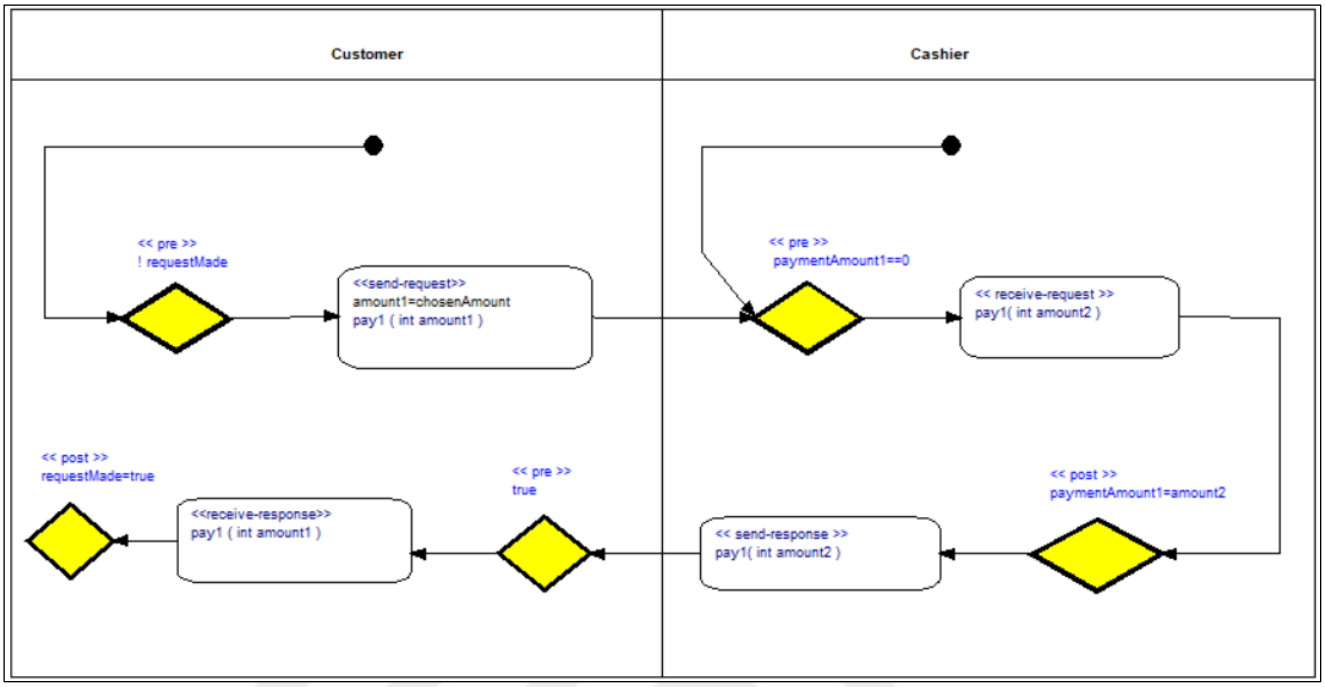


Şekil 7.10: *Pump*'ın *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri.

Customer'dan *Pump*'a istek geldiğinde önce ön şart (*pumpReleased1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre *(pumpReleased1=false*) atama yapılır. *Customer*'a, (kesikli ok ile gösterilen) *pump1* metot çağrısı ile dönüş değeri (*paymentAmount1*) yollanır.

7.2 GAZ İSTASYONU MODELİ (AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ)

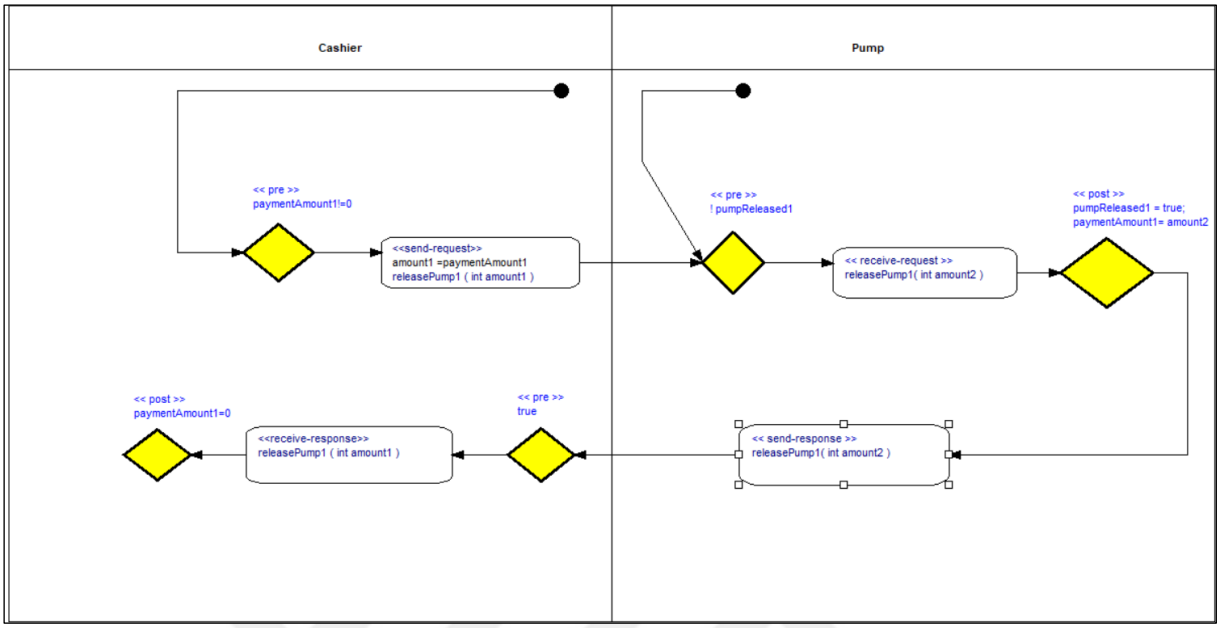
Şekil 7.1'de gösterilen gaz istasyonu modelinin aktivite diyagramı ile genişletilmiş şeklinde bileşenlerinin data verileri ve özellikleri de Şekil 7.2, Şekil 7.3 ve Şekil 7.4'deki gibi aynıdır. *Customer* ile *Cashier* arasındaki ilişki, *Customer*'ın "*pay*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.11'de gösteriliyor.



Şekil 7.11: *Customer-Cashier* aktivite diyagramı.

Şekil 7.11'de gösterilen *Customer-Cashier* aktivite diyagramında, *Customer*'ın *Cashier*'a *pay1* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!requestMade*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *pay1* metot çağrı aktivitesi gerçekleştirilir.Bu aktivitesin içinde metot parametresine yapılan atamada (*amount1=chosenAmount*) dahildir. *pay1* metodu, *Cashier* bileşenin sağlayıcı porta (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*paymentAmount1==0*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *pay1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*paymentAmount1=amount2*) sağlanır ve (*send-response*) *pay1* metot çağrı aktivitesi gerçekleştirilerek, *Customer* bileşenin istemci porta (*required port*) cevap yollanır. *Customer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *pay1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*requestMade=true*) sağlar.

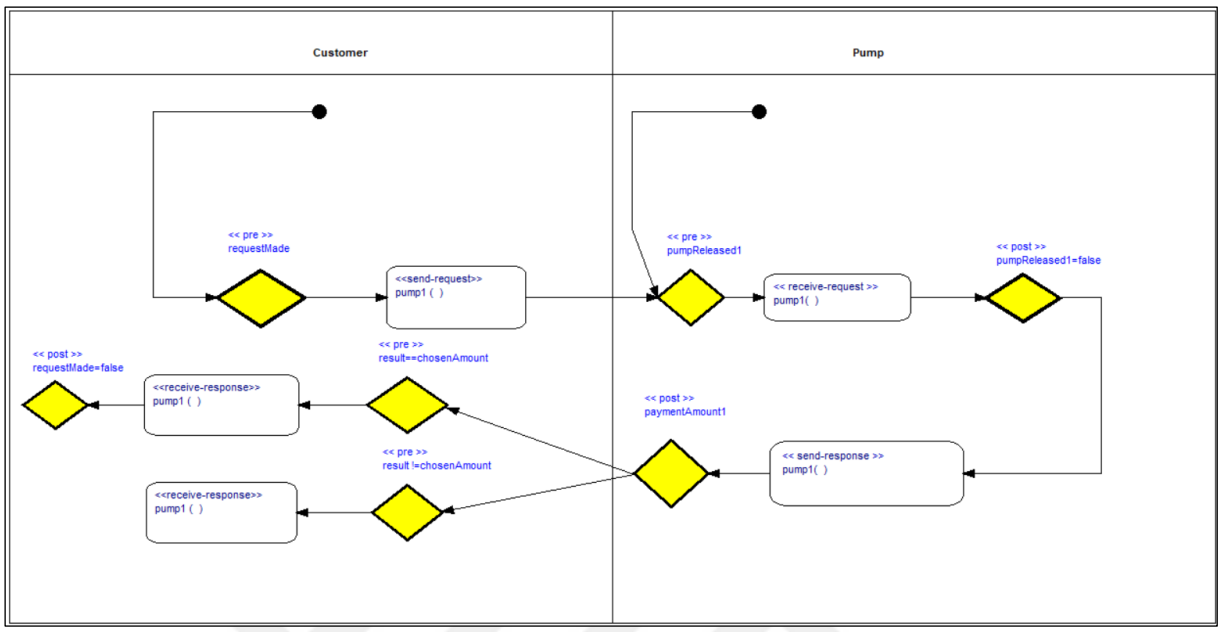
Cashier ile *Pump* arasındaki ilişki, *Cashier*'in "*releasePump*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.12'de gösteriliyor.



Şekil 7.12: *Cashier-Pump* aktivite diyagramı.

Şekil 7.12'de gösterilen *Cashier-Pump* aktivite diyagramında, *Cashier*'in *Pump*'a *releasePump1* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*paymentAmount1!=0*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *releasePump1* metot çağrı aktivitesi gerçekleştirilir.Bu aktivitesin içinde metot parametresine yapılan atamada (*amount1=paymentAmount1*) dahildir. *releasePump1* metodu, *Pump* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!pumpReleased1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *releasePump1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*pumpReleased1=true* ; *paymentAmount1=amount2*) sağlanır ve (*send-response*) *releasePump1* metot çağrı aktivitesi gerçekleştirilerek, *Cashier* bileşenin istemci portuna (*required port*) cevap yollanır. *Cashier*, cevabı ön şartsız (*true*) olarak (*receive-response*) *releasePump1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*paymentAmount1=0*) sağlar.

Customer ile *Pump* arasındaki ilişki, *Customer*'ın "*oil*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.13'de gösteriliyor.



Şekil 7.13: *Customer-Pump* aktivite diyagramı.

Şekil 7.13'de gösterilen *Customer-Pump* aktivite diyagramında, *Customer*'ın *Pump*'a *pump1* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*requestMade*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *pump1* metot çağrı aktivitesi gerçekleştirilir. *pump1* metodu, *Pump* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*pumpReleased1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *pump1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*pumpReleased1=false)* sağlanır ve (*send-response*) *pump1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *Customer* bileşenin istemci porta (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=paymentAmount1*) belirtir. *Customer,* cevabı alınca ön/sonraki koşul sembolündeki ön şartlara göre dönüş değerini (*result*) değerlendirir. Eğer (result==chosenAmount) ön şartını sağlarsa (*true*) olarak (*receive-response*) *releasePump1* metot çağrı aktivitesi gerçekleşir ve ön/sonraki koşul sembolündeki son şartı (*requestMade=false*) sağlanır.

Gaz istasyon modelinin, MetaEdit+ editörü dönüştürücüsünden elde edilen ProMeLa dosyası (gas1.pml) Liste 7.1'de gösterilmektedir.

Liste 7.1: gas1.pml dosyası.

```
1  mtype = {pay1,pump1,releasePump1}
2  chan ch_pay_REQ= [1] of{ mtype ,int };
3  chan ch_oil_REQ= [1] of{ mtype  };
4  chan ch_releasePump_REQ= [1] of{ mtype ,int };
5
6  chan ch_pay_RES= [1] of{ mtype, int} ;
7  chan ch_oil_RES= [1] of{ mtype, int} ;
8  chan ch_releasePump_RES= [1] of{ mtype, int} ;
9
10 proctype Customer() {
11   bool requestMade=false;
12   int chosenAmount=550;
13   int amount1;
14   int result;
15   int responseWaiting= false;
16
17   do
18   ::!requestMade && responseWaiting==false ->
19      responseWaiting=true ;
20      amount1 = chosenAmount
21      ch_pay_REQ !pay1,amount1
22   ::ch_pay_RES ?pay1,result->
23     if
24     ::(true)->requestMade = true;
25     ::!( true )->assert(false);
26     fi ;
27     responseWaiting=false ;
28   ::requestMade && responseWaiting==false ->
29     responseWaiting=true ;
30     ch_oil_REQ !pump1;
31   ::ch_oil_RES ?pump1,result->
32     if
33     ::(result == chosenAmount)->requestMade = false;
34     ::(result != chosenAmount)->;
35     ::!( result == chosenAmount || result != chosenAmount )->assert(false);
36     fi ;
37     responseWaiting=false ;
38   od
39 }
40 proctype Cashier() {
41   int paymentAmount1=0;
42   int amount1;
43   int amount2;
44   int result;
45   int responseWaiting= false;
46
47   do
48   ::paymentAmount1!=0 && responseWaiting==false ->
49     responseWaiting=true ;
50     amount1 =paymentAmount1
51     ch_releasePump_REQ !releasePump1,amount1
52   ::ch_releasePump_RES ?releasePump1,result->
53     if
54     ::(true)->paymentAmount1=0;
55     ::!( true )->assert(false);
56     fi ;
57     responseWaiting=false ;
58   ::ch_pay_REQ ?pay1,amount2->
59     if
60     ::(paymentAmount1 ==0)->
61       paymentAmount1 = amount2
62       ch_pay_RES !pay1,0;
63     ::!( (paymentAmount1 ==0)  )->
64       ch_pay_REQ !pay1,amount2
```

Liste 7.1: gas1.pml dosyası (devam).

```
65     fi
66   od
67 }
68 proctype Pump() {
69   bool pumpReleased1=false;
70   int paymentAmount1=0;
71   int amount2;
72   int result;
73
74   do
75   ::ch_oil_REQ ?pump1->
76     if
77     ::(pumpReleased1)->
78       pumpReleased1 = false
79       ch_oil_RES !pump1,paymentAmount1;
80     ::!( (pumpReleased1)  )->
81       ch_oil_REQ !pump1;
82     fi
83   ::ch_releasePump_REQ ?releasePump1,amount2->
84     if
85     ::(!pumpReleased1)->
86       pumpReleased1 = true; paymentAmount1 = amount2
87       ch_releasePump_RES !releasePump1,0;
88     ::!( (!pumpReleased1)  )->
89       ch_releasePump_REQ !releasePump1,amount2
90     fi
91   od
92 }
93 active proctype configuration()
94 {
95   atomic {
96    run Customer();
97    run Cashier();
98    run Pump();
99  }
100  }
101
```

Gaz istasyonu modelinin, MetaEdit+ editörü dönüştürücüsünden elde edilen java dosyası (Configuration.java) Liste 7.2'de gösterilmektedir.

Liste 7.2: Configuration.java dosyası.

```
1       public class Configuration {
2               Customer customerIns;
3               Cashier cashierIns;
4               Pump pumpIns;
5
6               CustomerCashier_pay_Adapter customercashier_pay_AdapterIns;
7
8               CustomerPump_oil_Adapter customerpump_oil_AdapterIns;
9
10              CashierPump_releasePump_Adapter cashierpump_releasePump_AdapterIns;
11
12              public Configuration() {
13                      customerIns = new Customer();
```

Liste 7.2: Configuration.java dosyası (devam).

```
14                      cashierIns = new Cashier();
15                      pumpIns = new Pump();
16                      try {
17                              customercashier_pay_AdapterIns =
18                                      new CustomerCashier_pay_Adapter(cashierIns);
19                              customerIns.customercashier_pay_InterfaceIns =
20                                       customercashier_pay_AdapterIns;
21
22                      } catch (Exception e) {
23                              e.printStackTrace();
24                      }
25                      try {
26                              customerpump_oil_AdapterIns =
27                                      new CustomerPump_oil_Adapter(pumpIns);
28                              customerIns.customerpump_oil_InterfaceIns =
29                                      customerpump_oil_AdapterIns;
30
31                      } catch (Exception e) {
32                              e.printStackTrace();
33                      }
34                      try {
35                              cashierpump_releasePump_AdapterIns =
36                                      new CashierPump_releasePump_Adapter(pumpIns);
37                              cashierIns.cashierpump_releasePump_InterfaceIns =
38                                      cashierpump_releasePump_AdapterIns;
39
40                      } catch (Exception e) {
41                              e.printStackTrace();
42                      }
43                      customerIns.start();
44                      cashierIns.start();
45
46              }
47
48              public static void main(String args[]) throws Exception {
49                      new Configuration();
50              }
51      }
52
53      interface CustomerCashier_pay_Interface {
54              public void pay1(int amount1) throws Exception;
55      }
56
57      interface CustomerPump_oil_Interface {
58              public int pump1() throws Exception;
59      }
60
61      interface CashierPump_releasePump_Interface {
62              public void releasePump1(int amount1) throws Exception;
63      }
64
65      class CustomerCashier_pay_Adapter implements CustomerCashier_pay_Interface {
66              Cashier cashier_ins;
67
68              public CustomerCashier_pay_Adapter(Cashier cashier_arg) throws Exception {
69                      cashier_ins = cashier_arg;
70              }
71
72              public void pay1(int amount1) {
73                      try {
74                              cashier_ins.pay1(amount1);
75                      } catch (Exception e) {
76                              e.printStackTrace();
77                      }
```

Liste 7.2: Configuration.java dosyası (devam).

```
78              }
79      }
80
81      class CustomerPump_oil_Adapter implements CustomerPump_oil_Interface {
82              Pump pump_ins;
83
84              public CustomerPump_oil_Adapter(Pump pump_arg) throws Exception {
85                      pump_ins = pump_arg;
86              }
87
88              public int pump1() throws Exception {
89                      return pump_ins.pump1();
90              }
91      }
92
93      class CashierPump_releasePump_Adapter implements CashierPump_releasePump_Interface {
94              Pump pump_ins;
95
96              public CashierPump_releasePump_Adapter(Pump pump_arg) throws Exception {
97                      pump_ins = pump_arg;
98              }
99
100             public void releasePump1(int amount1) {
101                     try {
102                             pump_ins.releasePump1(amount1);
103                     } catch (Exception e) {
104                             e.printStackTrace();
105                     }
106             }
107     }
108
109     class Customer extends Thread {
110             boolean requestMade = false;
111             int chosenAmount = 550;
112             private int result = 0;
113             CustomerCashier_pay_Interface customercashier_pay_InterfaceIns;
114             CustomerPump_oil_Interface customerpump_oil_InterfaceIns;
115             private boolean responseWaiting = false;
116
117             public void pay1(int amount1) throws Exception {
118                     if (!requestMade && !responseWaiting) {
119                             responseWaiting = true;
120                             amount1 = chosenAmount;
121                             customercashier_pay_InterfaceIns.pay1(amount1);
122                     }
123                     if (true) {
124                             requestMade = true;
125                     }
126                     responseWaiting = false;
127             }
128
129             public int pump1() throws Exception {
130                     if (requestMade && !responseWaiting) {
131                             responseWaiting = true;
132                             result = customerpump_oil_InterfaceIns.pump1();
133                     }
134                     if (result == chosenAmount) {
135                             requestMade = false;
136                     }
137                     if (result != chosenAmount) {
138
139                     }
```

Liste 7.2: Configuration.java dosyası (devam).

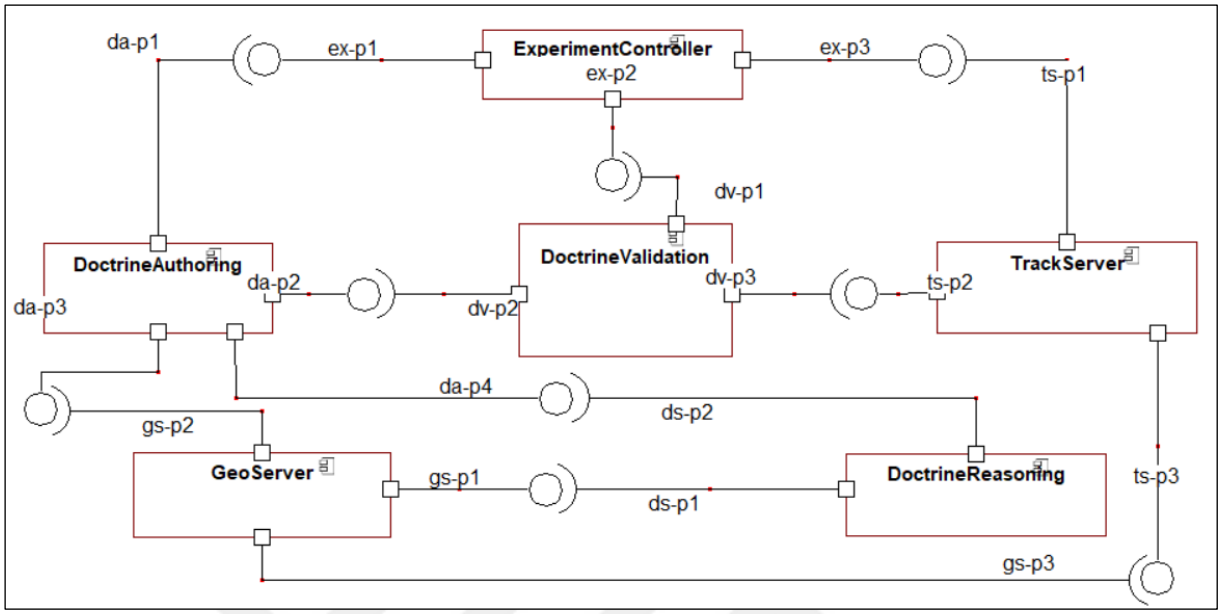
```
140                     responseWaiting = false;
141                     return result;
142             }
143
144             public void run() {
145                     while (true) {
146                             try {
147                                     pay1(0);
148                             } catch (Exception e) {
149                                     e.printStackTrace();
150                             }
151                             try {
152                                     pump1();
153                             } catch (Exception e) {
154                                     e.printStackTrace();
155                             }
156                     }
157             }
158     }
159
160     class Cashier extends Thread {
161             int paymentAmount1 = 0;
162             CashierPump_releasePump_Interface cashierpump_releasePump_InterfaceIns;
163             private boolean responseWaiting = false;
164
165             public void releasePump1(int amount1) throws Exception {
166                     if (paymentAmount1 != 0 && !responseWaiting) {
167                             responseWaiting = true;
168                             amount1 = paymentAmount1;
169                             cashierpump_releasePump_InterfaceIns.releasePump1(amount1);
170                     }
171                     if (true) {
172                             paymentAmount1 = 0;
173                     }
174                     responseWaiting = false;
175             }
176
177             public void pay1(int amount2) throws Exception {
178                     if (paymentAmount1 == 0) {
179                             paymentAmount1 = amount2;
180                     }
181             }
182
183             public void run() {
184                     while (true) {
185                             try {
186                                     releasePump1(0);
187                             } catch (Exception e) {
188                                     e.printStackTrace();
189                             }
190                     }
191             }
192     }
193
194     class Pump {
195             boolean pumpReleased1 = false;
196             int paymentAmount1 = 0;
197             private int result = 0;
198
199             public int pump1() throws Exception {
200                     if (pumpReleased1) {
201                             pumpReleased1 = false;
```

Liste 7.2: Configuration.java dosyası (devam).

```
202                             result = paymentAmount1;
203                     }
204                     return result;
205             }
206
207             public void releasePump1(int amount2) throws Exception {
208                     if (!pumpReleased1) {
209                             pumpReleased1 = true;
210                             paymentAmount1 = amount2;
211                     }
212             }
213     }
214
```

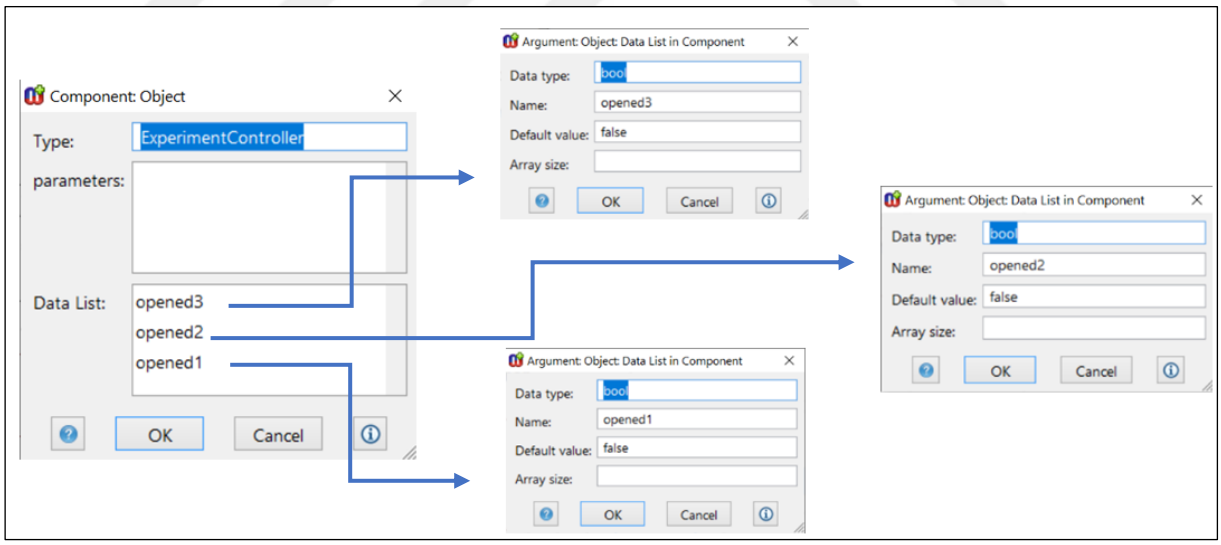
7.3 AEGİS VAKA ANALİZİ

Aegis muharebe sisteminde [42] (Şekil 7.14) *ExperimentControl* bileşeni sistem için gerekli bilgileri toplayıp hareketli yabancı unsurlarının konumlarını *TrackServer* bileşenine, *DoctrineAuthoring* bileşenine ise görev ve uygulama ile ilgili kuralları aktarıyor. *GeoServer* bileşeni, *DoctrineAuthoring* bileşeninden görev ve uygulama kuralarını; *TrackServer*'dan konum bilgilerini alarak, yabancı unsurların hangi geometrik alanda olduğunu hesabını yapıyor ve *DoctrineReasoning* bileşenine aktarıyor. *DoctrineReasoning* bileşeni de hangi aksiyonun alınacağının kararını veriyor.

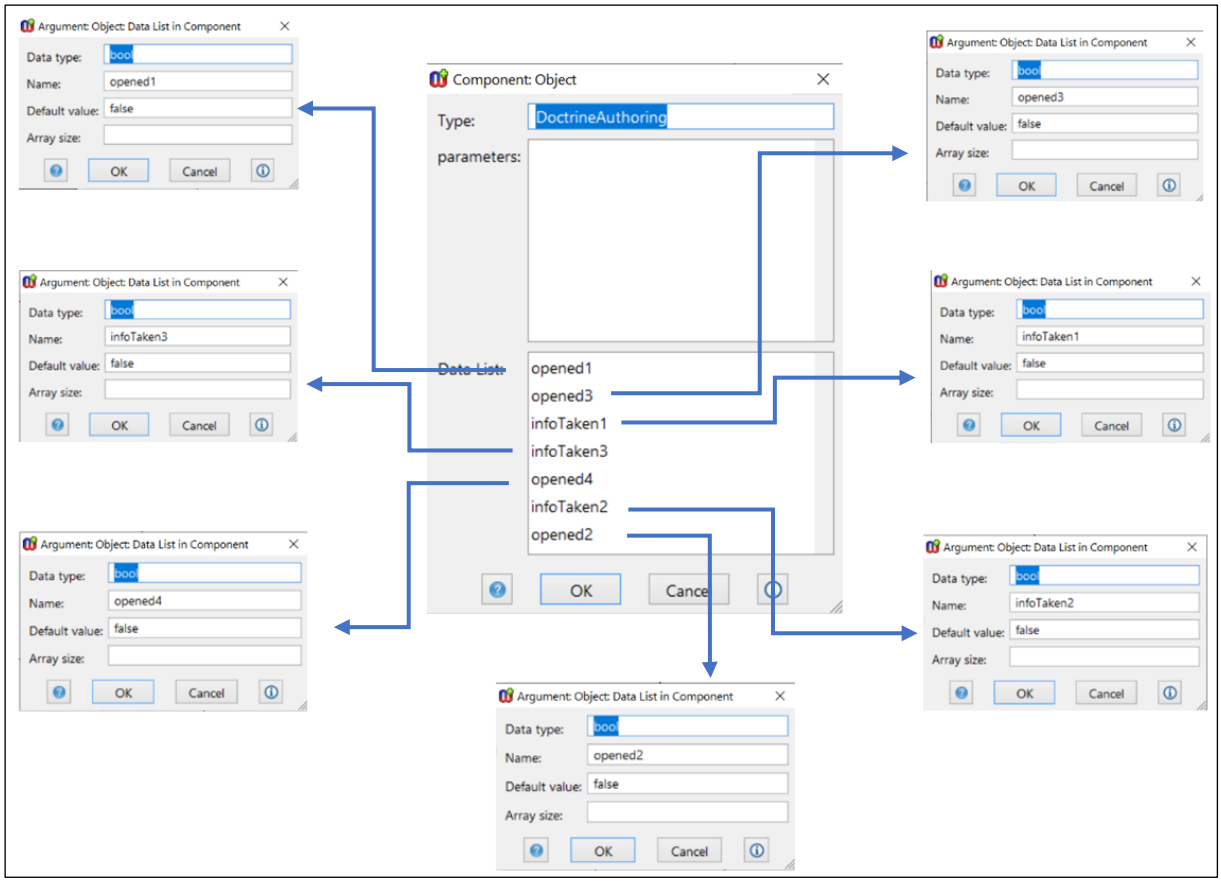


Şekil 7.14: SAwUML'de oluşturulan aegis modeli.

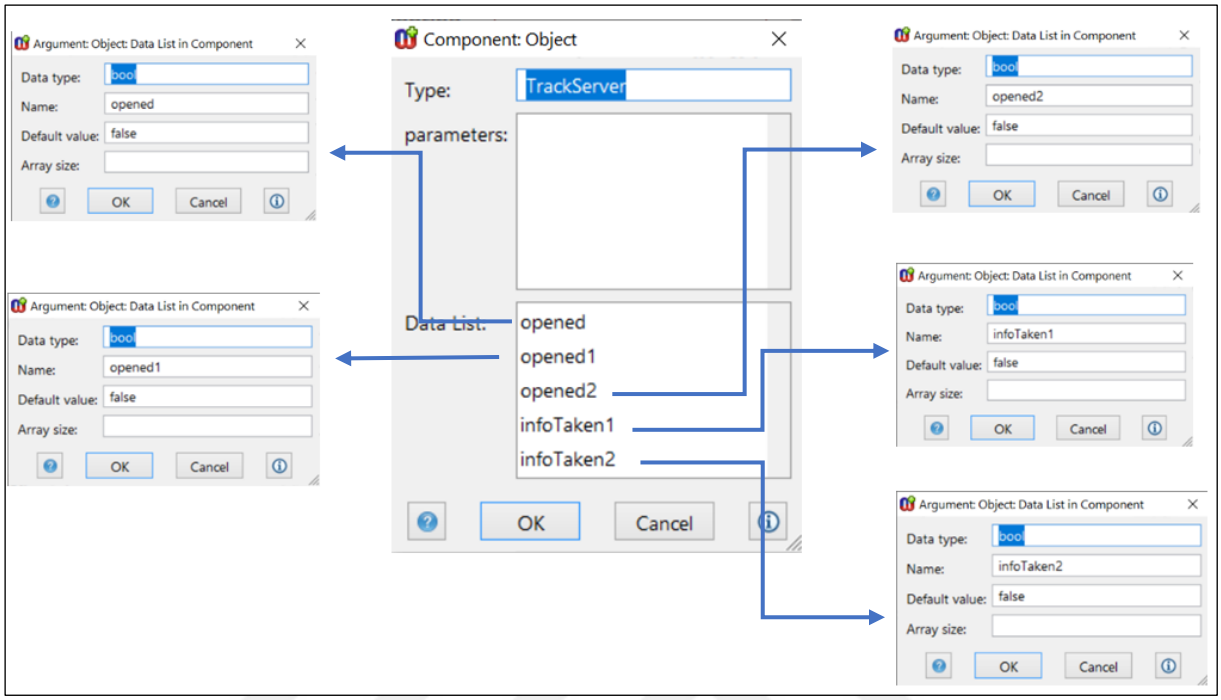
Aegis muharebe sistemi bileşenlerinin verileri (data) ve özellikleri aşağıda gösterilmiştir.



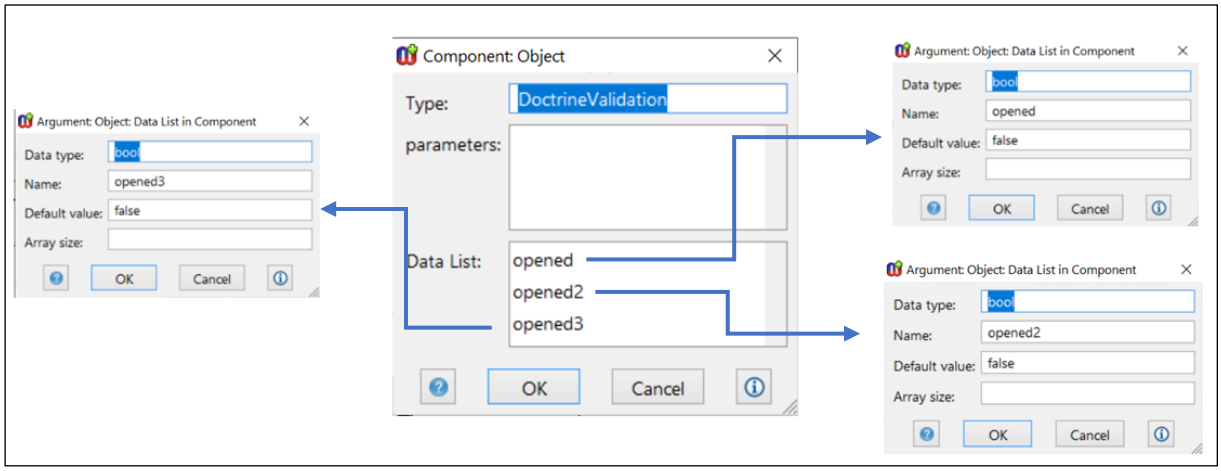
Şekil 7.15: *ExperimentController* bileşeninin verileri ve özellikleri.



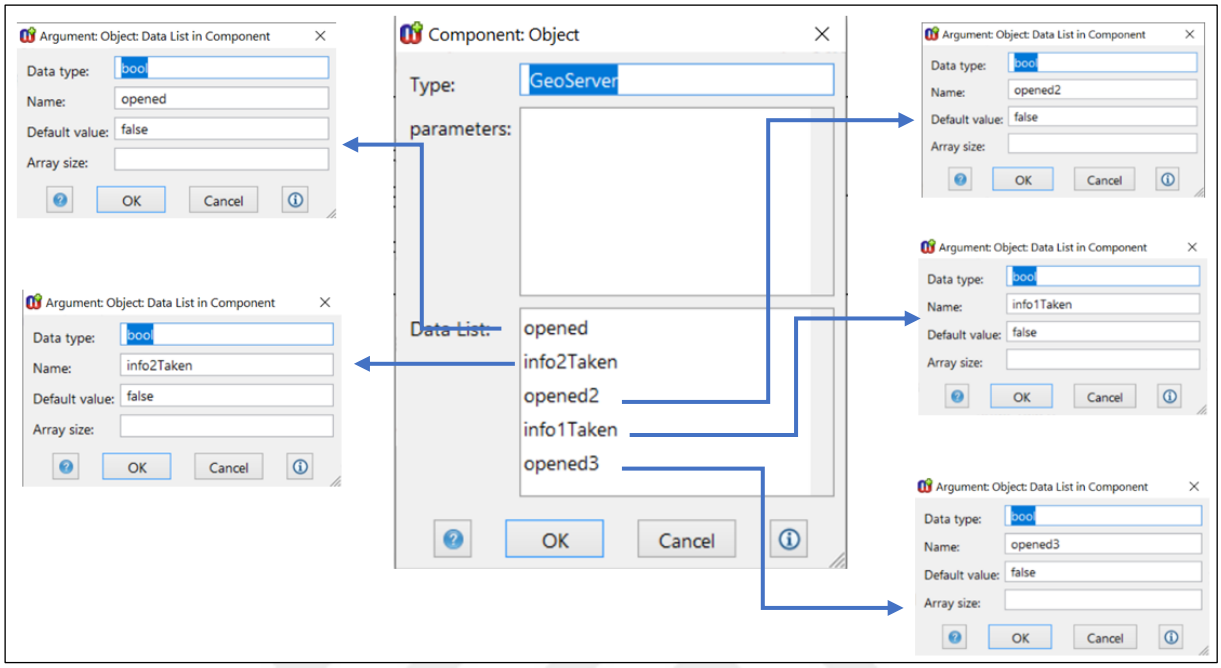
Şekil 7.16: *DoctrineAuthoring* bileşeninin verileri ve özellikleri.



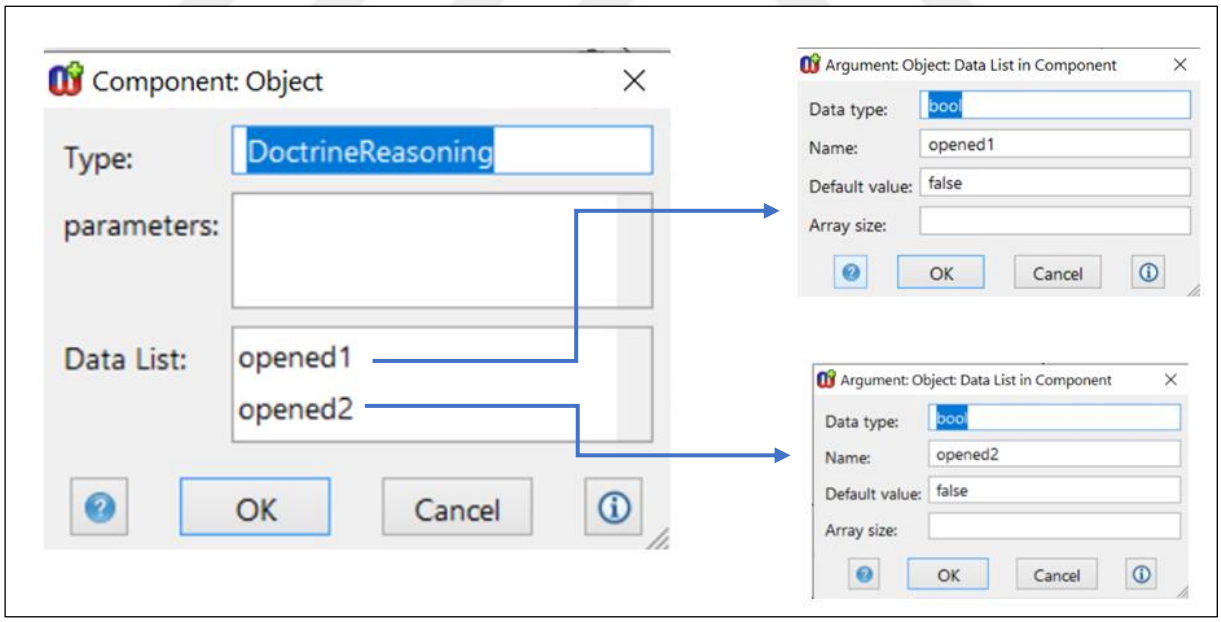
Şekil 7.17: *TrackServer* bileşeninin verileri ve özellikleri.



Şekil 7.18: *DoctrineValidation* bileşeninin verileri ve özellikleri.

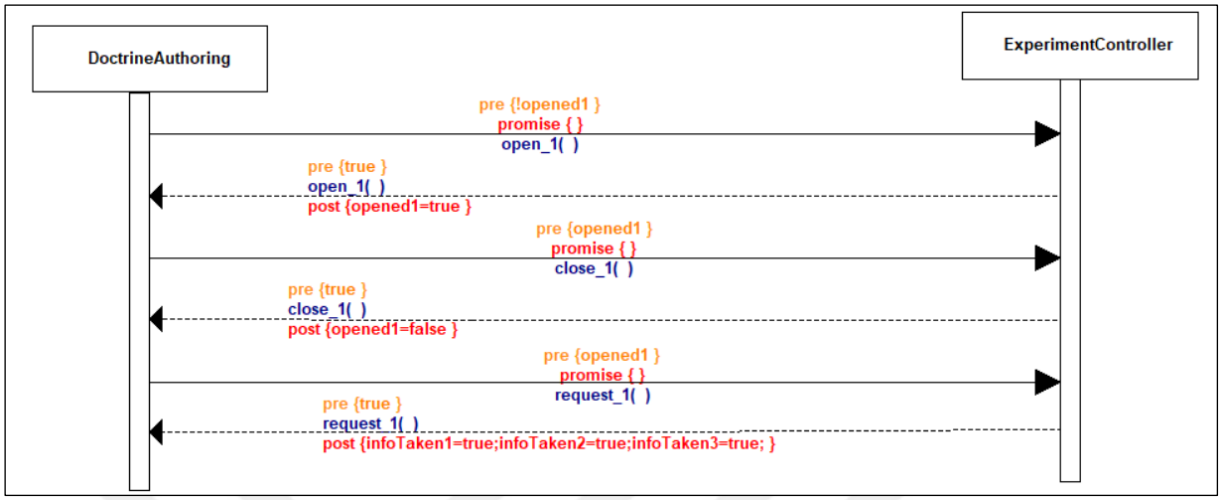


Şekil 7.19: *GeoServer* bileşeninin verileri ve özellikleri.



Şekil 7.20: *DoctrineReasoning* bileşeninin verileri ve özellikleri.

DoctrineAuthoring'in *ExperimentController* ile ilişkisi, *DoctrineAuthoring*'in "*da-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.21'de gösteriliyor.



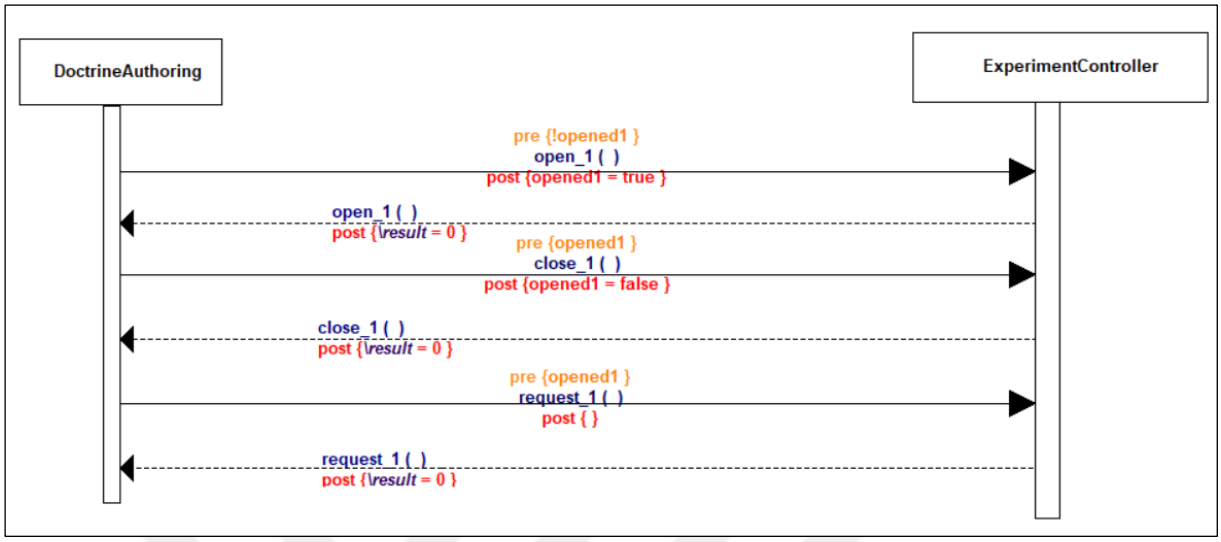
Şekil 7.21: *DoctrineAuthoring*'in *ExperimentController*'a bağlanan istemci port (*required port*) özellikleri.

DoctrineAuthoring'in *ExperimentController*'den *open_1* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *open_1* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened1=true*) göre atama yapılır.

DoctrineAuthoring'in *ExperimentController*'den *close_1* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *close_1* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened1=false*) göre atama yapılır.

DoctrineAuthoring'in *ExperimentController*'den *request_1* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *request_1* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*infoTaken1=true;infoTaken2=true;infoTaken3=true;*) göre atama yapılır.

ExperimentController'ın *DoctrineAuthoring* ile ilişkisi, *ExperimentController*'ın "*ex-p1*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.22'de gösteriliyor.



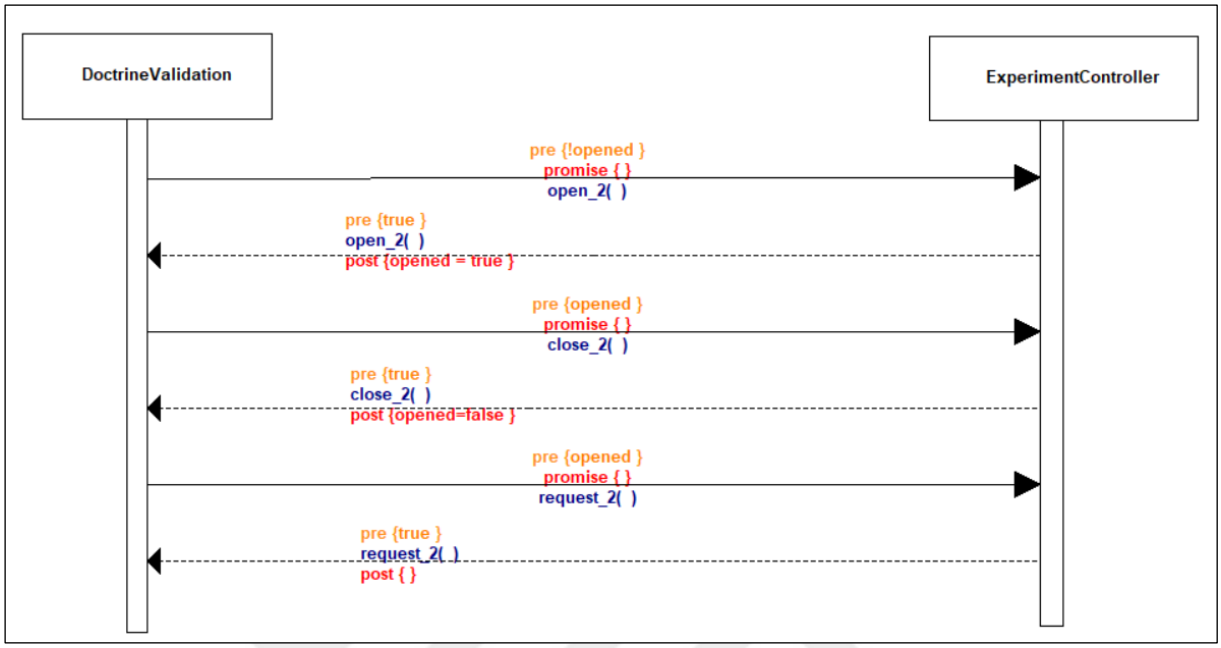
Şekil 7.22: *ExperimentController*'ın *DoctrineAuthoring*'a bağlanan sağlayıcı bağlantı (*provided port*) özellikleri.

DoctrineAuthoring'dan *ExperimentController*'a *open_1* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened1=true*) atama yapılır. *DoctrineAuthoring*'a (kesikli ok ile gösterilen) *open_1* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineAuthoring'dan *ExperimentController*'a *close_1* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened1=false*) atama yapılır. *DoctrineAuthoring*'a (kesikli ok ile gösterilen) *close_1* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineAuthoring'dan *ExperimentController*'a *request_1* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened1*) kontrol edilir. Ön şart sağlanıyorsa, *DoctrineAuthoring*'a (kesikli ok ile gösterilen) *request_1* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'in *ExperimentController* ile ilişkisi, *DoctrineValidation*'ın "*dv-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.23'de gösteriliyor.



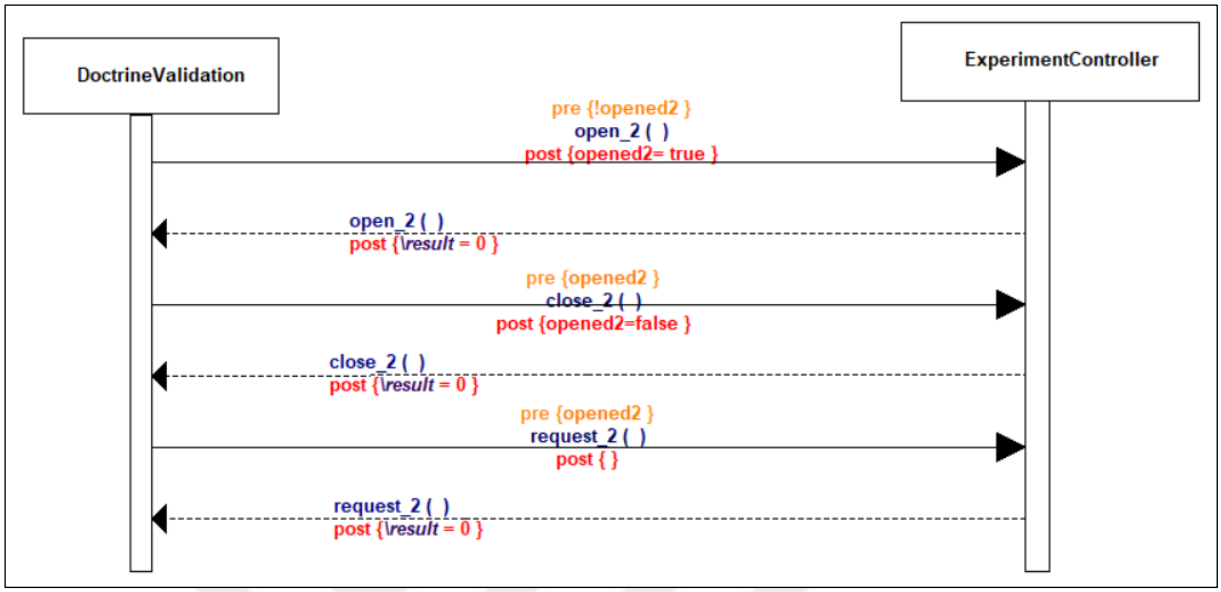
Şekil 7.23: *DoctrineValidation*'ın *ExperimentController*'a bağlanan istemci portun (*required port*) özellikleri.

DoctrineValidation'ın *ExperimentController*'den *open_2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *open_2* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened=true*) göre atama yapılır.

DoctrineValidation'ın *ExperimentController*'den *close_2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *close_2* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened=false*) göre atama yapılır.

DoctrineValidation'ın *ExperimentController*'den *request_2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *request_2* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak kabul edilir.

ExperimentController'ın *DoctrineValidation* ile ilişkisi, *ExperimentController*'ın "*ex-p2*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.24'de gösteriliyor.



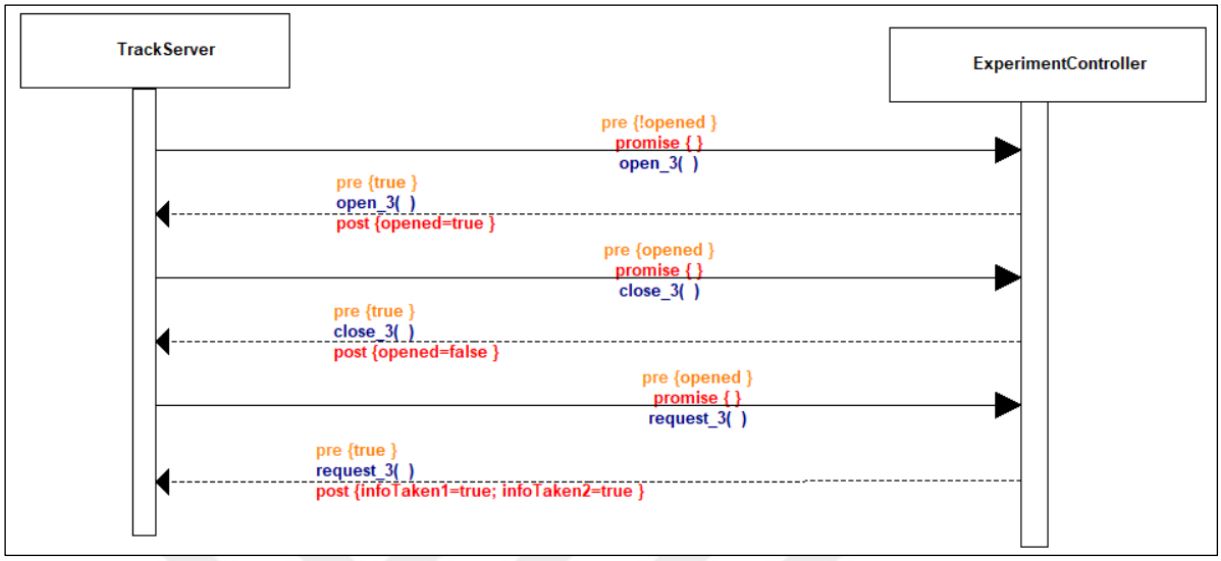
Şekil 7.24: *ExperimentController*'ın *DoctrineValidation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

DoctrineValidation'dan *ExperimentController*'a *open_2* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened2*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened2=true*) atama yapılır. *DoctrineValidation*'a (kesikli ok ile gösterilen) *open_2* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'dan *ExperimentController*'a *close_2* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened2*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened2=false*) atama yapılır. *DoctrineValidation*'a (kesikli ok ile gösterilen) *close_2* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'dan *ExperimentController*'a *request_2* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened2*) kontrol edilir. Ön şart sağlanıyorsa, *DoctrineValidation*'a (kesikli ok ile gösterilen) *request_2* metot çağrısı ile dönüş değeri (0) yollanır.

TrackServer'ın *ExperimentController* ile ilişkisi, *DoctrineValidation*'ın "*ts-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.25'de gösteriliyor.



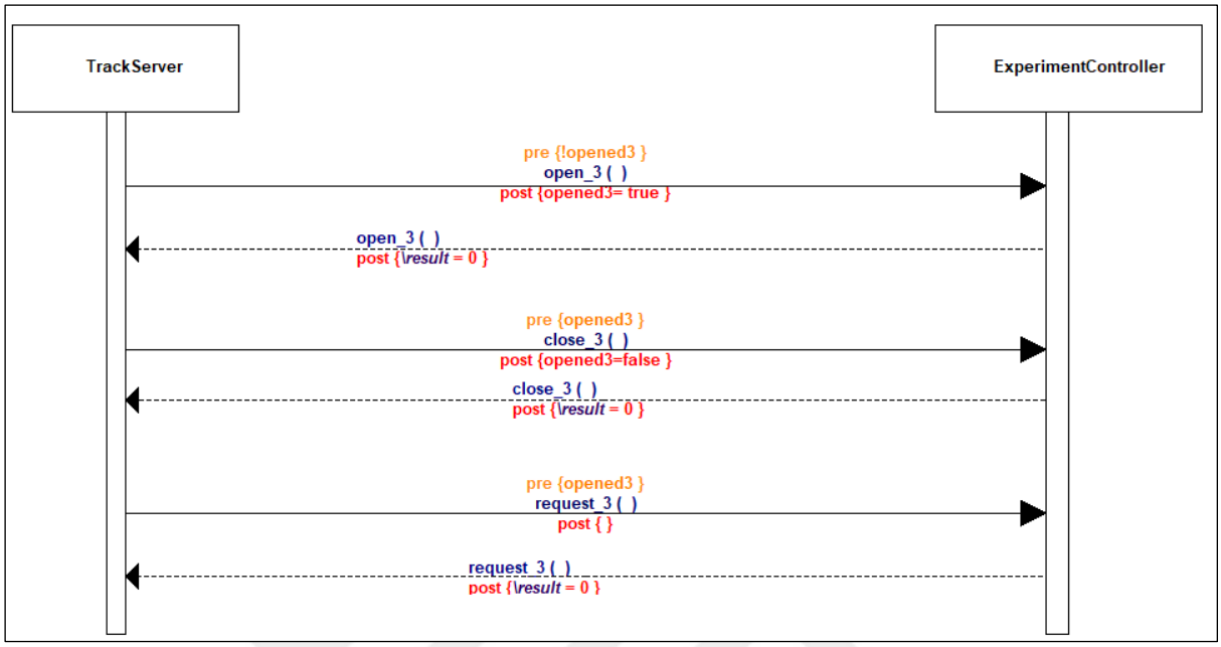
Şekil 7.25: *TrackServer*'ın *ExperimentController*'e bağlanan istemci portun (*required port*) özellikleri.

TrackServer'ın *ExperimentController*'den *open_3* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *open_3* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen), ön şart (*true*) kontrol edilir, eğer bu şart sağlanıyorsa, son koşula (*opened=true*) göre atama yapılır.

TrackServer'ın *ExperimentController*'den *close_3* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *close_3* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened=false*) göre atama yapılır.

TrackServer'ın *ExperimentController*'den *request_3* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *ExperimentController*'a *request_3* metot çağrısı yapılır. *ExperimentController*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*infoTaken1=true; infoTaken2=true;*) göre atama yapılır.

ExperimentController'ın *TrackServer* ile ilişkisi, *ExperimentController*'ın "*ex-p3*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.26'da gösteriliyor.



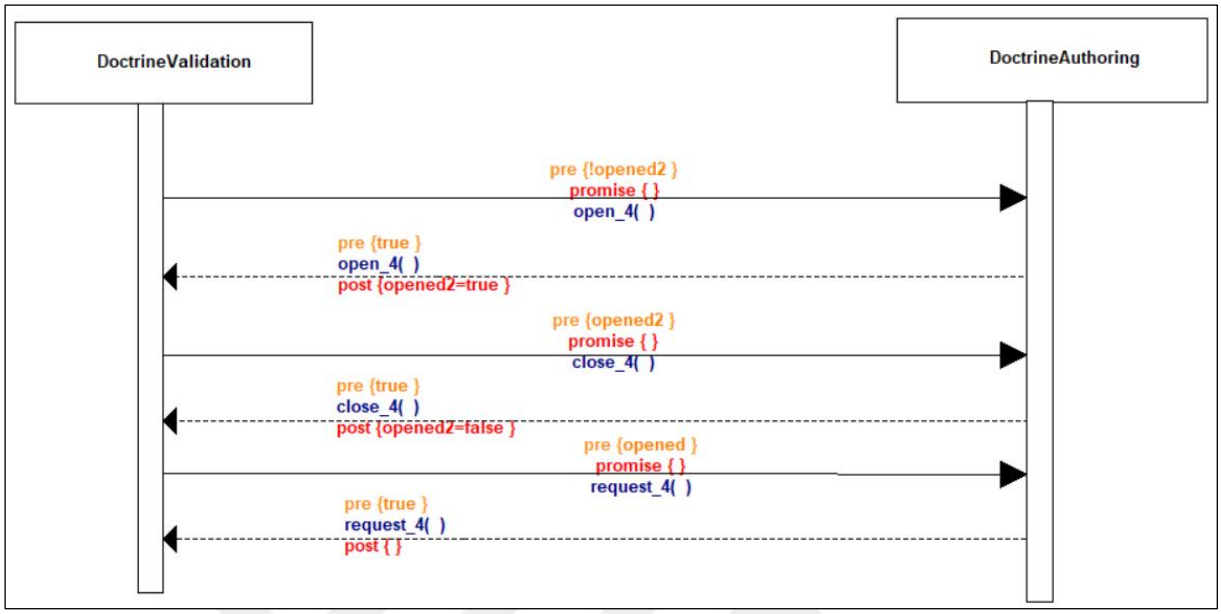
Şekil 7.26: *ExperimentController*'ın *TrackServer*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

TrackServer'dan *ExperimentController*'a *open_3* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened3*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened3=true*) atama yapılır. *DoctrineAuthoring*'a (kesikli ok ile gösterilen) *open_3* metot çağrısı ile dönüş değeri (0) yollanır.

TrackServer'dan *ExperimentController*'a *close_3* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened3*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened3=false*) atama yapılır. *DoctrineAuthoring*'a (kesikli ok ile gösterilen) *close_3* metot çağrısı ile dönüş değeri (0) yollanır.

TrackServer'dan *ExperimentController*'a *request_3* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened3*) kontrol edilir. Ön şart sağlanıyorsa, *DoctrineAuthoring*'a (kesikli ok ile gösterilen) *request_3* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'ın *DoctrineAuthoring* ile ilişkisi, *DoctrineValidation*'ın "*dv-p2*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.27'de gösteriliyor.



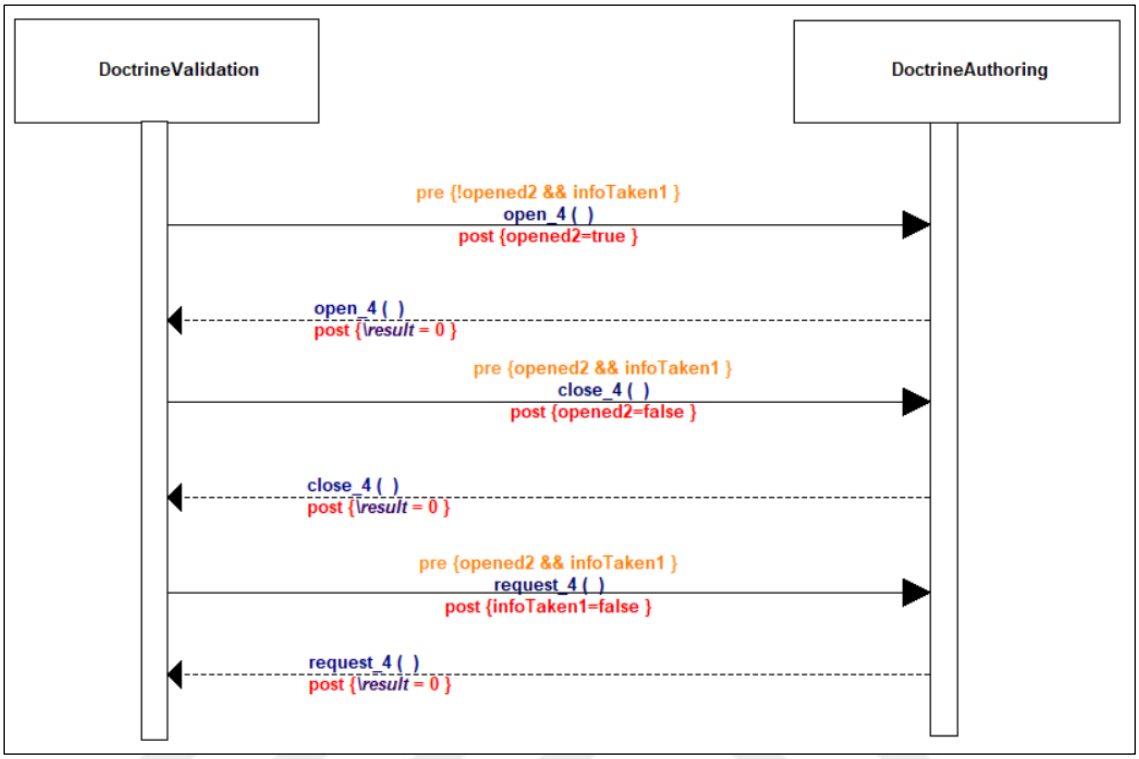
Şekil 7.27: *DoctrineValidation*'ın *DoctrineAuthoring*'a bağlanan istemci portun (*required port*) özellikleri.

DoctrineValidation'ın *DoctrineAuthoring*'den *open_4* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctrineAuthoring*'a *open_4* metot çağrısı yapılır. *DoctrineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened2=true*) göre atama yapılır.

DoctrineValidation'ın *DoctrineAuthoring*'den *close_4* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctrineAuthoring*'a *close_4* metot çağrısı yapılır. *DoctrineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened2=false*) göre atama yapılır.

DoctrineValidation'ın *DoctrineAuthoring*'den *request_3* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctrineAuthoring*'a *request_4* metot çağrısı yapılır. *DoctrineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak kabul edilir.

DoctrineAuthoring'in *DoctrineValidation* ile ilişkisi, *DoctrineAuthoring*'in "*da-p2*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.28'de gösteriliyor.



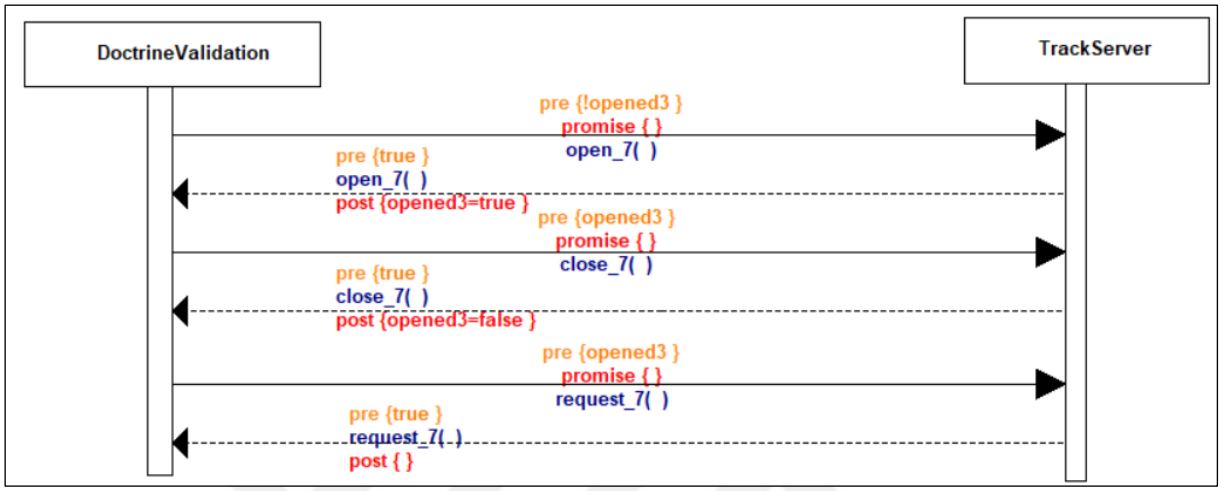
Şekil 7.28: *DoctrineAuthoring*'in *DoctrineValidation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

DoctrineValidation'dan *DoctrineAuthoring*'a *open_4* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened2 && infoTaken1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened2=true*) atama yapılır *DoctrineValidation*'a (kesikli ok ile gösterilen) *open_4* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'dan *DoctrineAuthoring*'a *close_4* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened2 && infoTaken1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened2=false*) atama yapılır. *DoctrineValidation*'a (kesikli ok ile gösterilen) *close_4* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'dan *DoctrineAuthoring*'a *request_4* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened2 && infoTaken1*) kontrol edilir. Ön şart sağlanıyorsa, *DoctrineValidation*'a (kesikli ok ile gösterilen) *request_4* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'ın *TrackServer* ile ilişkisi, *TrackServer*'ın "*dv-p3*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagramda Şekil 7.29'da gösteriliyor.



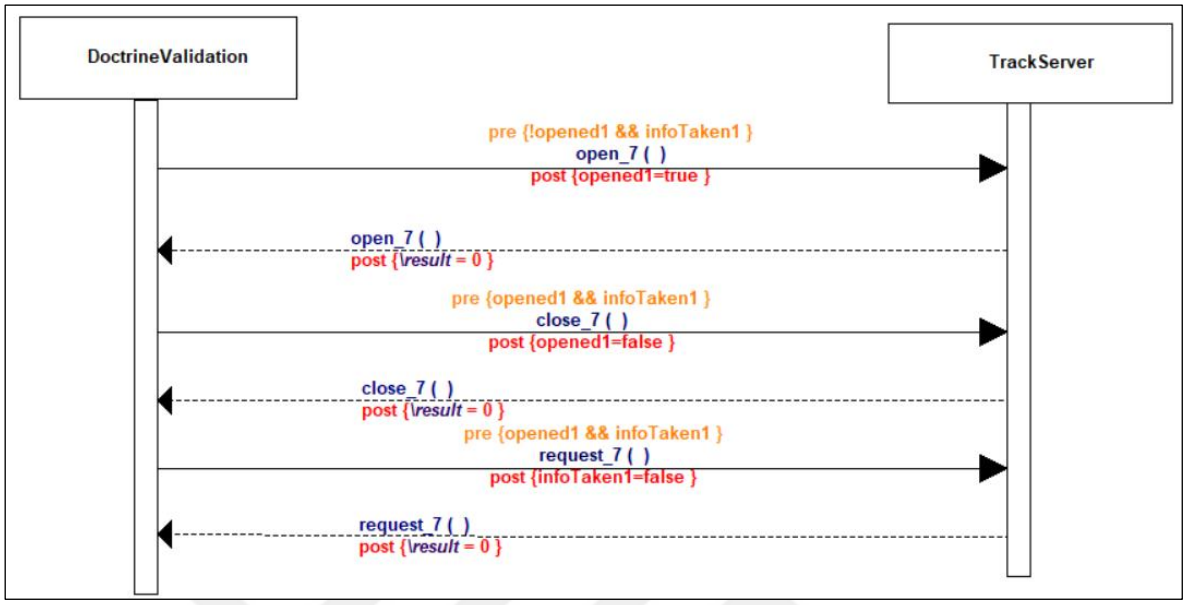
Şekil 7.29: *DoctrineValidation*'ın *TrackServer*'a bağlanan istemci portun (*required port*) özellikleri.

DoctrineValidation'ın *TrackServer*'den *open_7* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened3*) sağlaması gerekiyor. Bu ön şart sağlandığında *TrackServer*'a *open_7* metot çağrısı yapılır. *TrackServer*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened3=true*) göre atama yapılır.

DoctrineValidation'ın *TrackServer*'den *close_7* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened3*) sağlaması gerekiyor. Bu ön şart sağlandığında *TrackServer*'a *close_7* metot çağrısı yapılır. *TrackServer*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened3=false*) göre atama yapılır.

DoctrineValidation'ın *TrackServer*'den *request_7* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *TrackServer*'a *request_7* metot çağrısı yapılır. *TrackServer*'dan alınan cevapta (kesikli ok ile gösterilen), ön şartsız (*true*) olarak kabul edilir.

Trackserver'in *DoctrineValidation* ile ilişkisi, *TrackServer*'in "*ts-p2*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.30'da gösteriliyor.



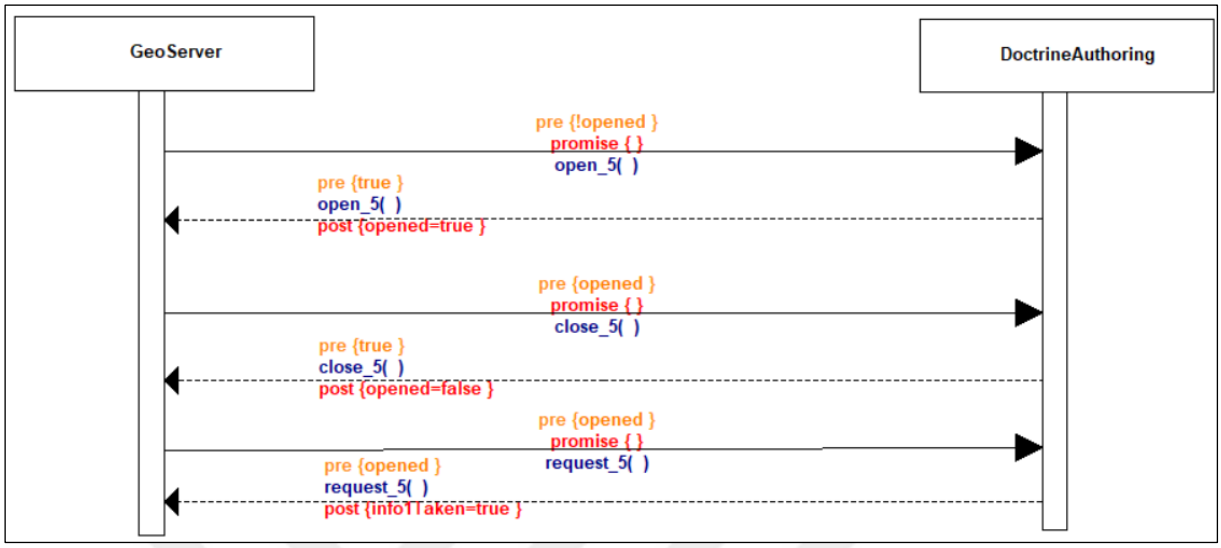
Şekil 7.30: *TrackServer*'in *DoctrineValidation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

DoctrineValidation'dan *TrackServer*'a *open_7* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened1 && infoTaken1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened1=true*) atama yapılır *DoctrineValidation*'a (kesikli ok ile gösterilen) *open_7* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'dan *TrackServer*'a *close_7* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened1 && infoTaken1*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened1=false*) atama yapılır. *DoctrineValidation*'a (kesikli ok ile gösterilen) *close_7* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineValidation'dan *TrackServer*'a *request_7* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened1 && infoTaken1*) kontrol edilir. Ön şart sağlanıyorsa, *DoctrineValidation*'a (kesikli ok ile gösterilen) *request_7* metot çağrısı ile dönüş değeri (0) yollanır.

GeoServer'ın DoctrineAuthoring ile ilişkisi, *GeoServer*'ın "*gs-p2*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.31'de gösteriliyor.



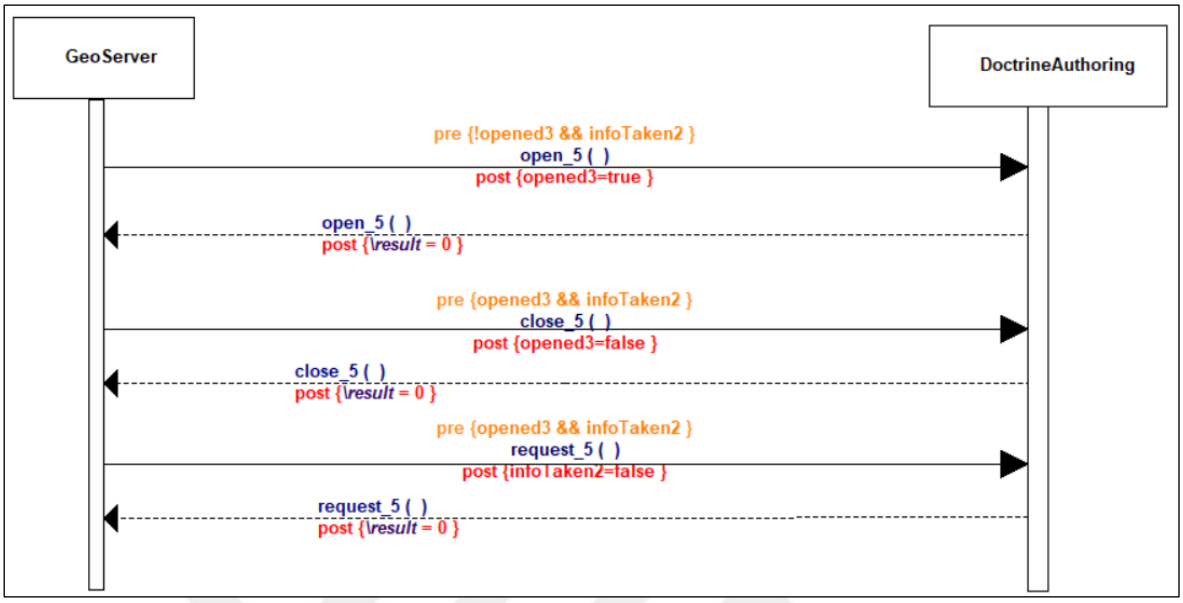
Şekil 7.31: GeoServer'ın *DoctrineAuthoring*'a bağlanan istemci portun (*required port*) özellikleri.

GeoServer'ın *DoctorineAuthoring*'den *open_5* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctrineAuthoring*'e *open_5* metot çağrısı yapılır. *DoctrineAuthoring*'den alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened=true*) göre atama yapılır.

GeoServer'ın *DoctorineAuthoring*'den *close_5* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened3*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctorineAuthoring*'a *close_5* metot çağrısı yapılır. *DoctorineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened=false*) göre atama yapılır.

GeoServer'ın *DoctorineAuthoring*'den *request_5* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctorineAuthoring*'a *request_5* metot çağrısı yapılır. *DoctorineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen), ön şartsız (*true*) olarak son koşula (*info1Taken=true*) göre atama yapılır.

DoctorineAuthoring'in *GeoServer* ile ilişkisi, *DoctorineAuthoring*'in "*da-p3*" isimli sağlayıcı porta (provide port) sağ tıklama ile açılan diyagram Şekil 7.32'de gösteriliyor.



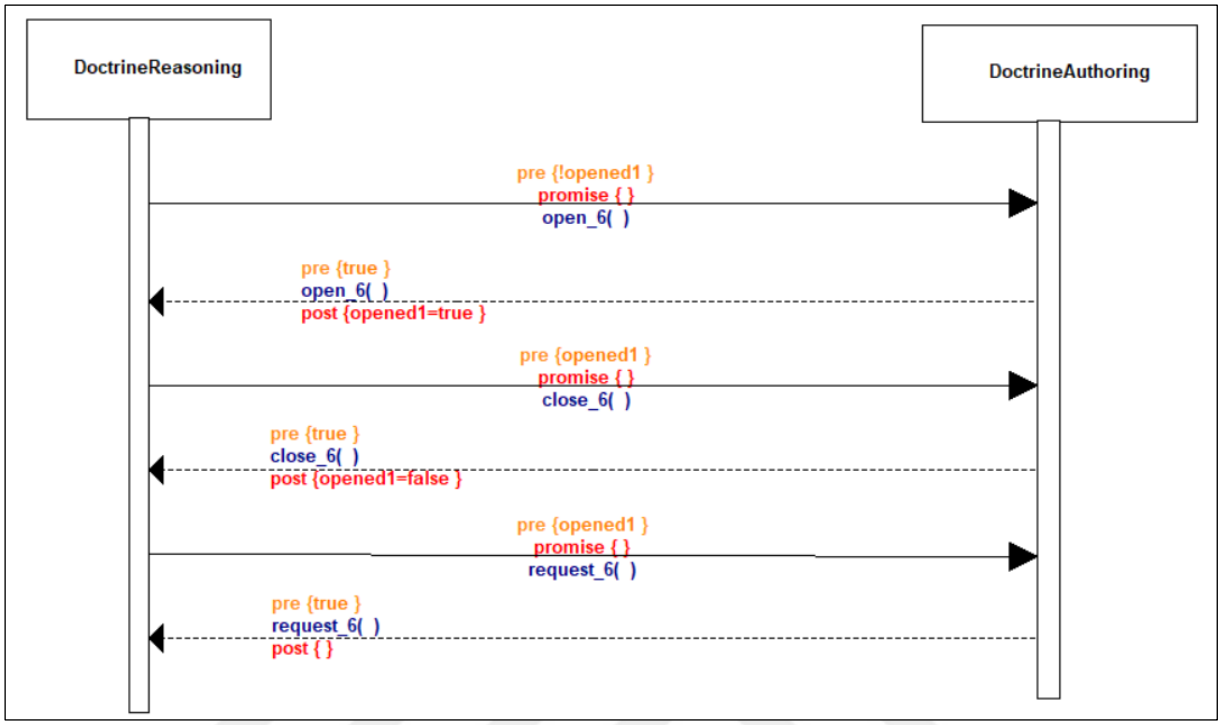
Şekil 7.32: *DoctorineAuthoring*'in *GeoServer*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

GeoServer'dan *DoctorineAuthoring*'a *open_5* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened3 && infoTaken2*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened3=true*) atama yapılır *GeoServer*'a (kesikli ok ile gösterilen) *open_5* metot çağrısı ile dönüş değeri (0) yollanır.

GeoServer'dan *DoctorineAuthoring*'a *close_5* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened3 && infoTaken2*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened3=false*) atama yapılır. *GeoServer*'a (kesikli ok ile gösterilen) *close_5* metot çağrısı ile dönüş değeri (0) yollanır.

GeoServer'dan *DoctorineAuthoring*'a *request_5* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened3 && infoTaken2*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*infoTaken2=false*) atama yapılır. *GeoServer*'a (kesikli ok ile gösterilen) *request_5* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineReasoning'in *DoctrineAuthoring* ile ilişkisi, *DoctrineReasoning*'in "*ds-p2*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram şekil 7.33'de gösteriliyor.



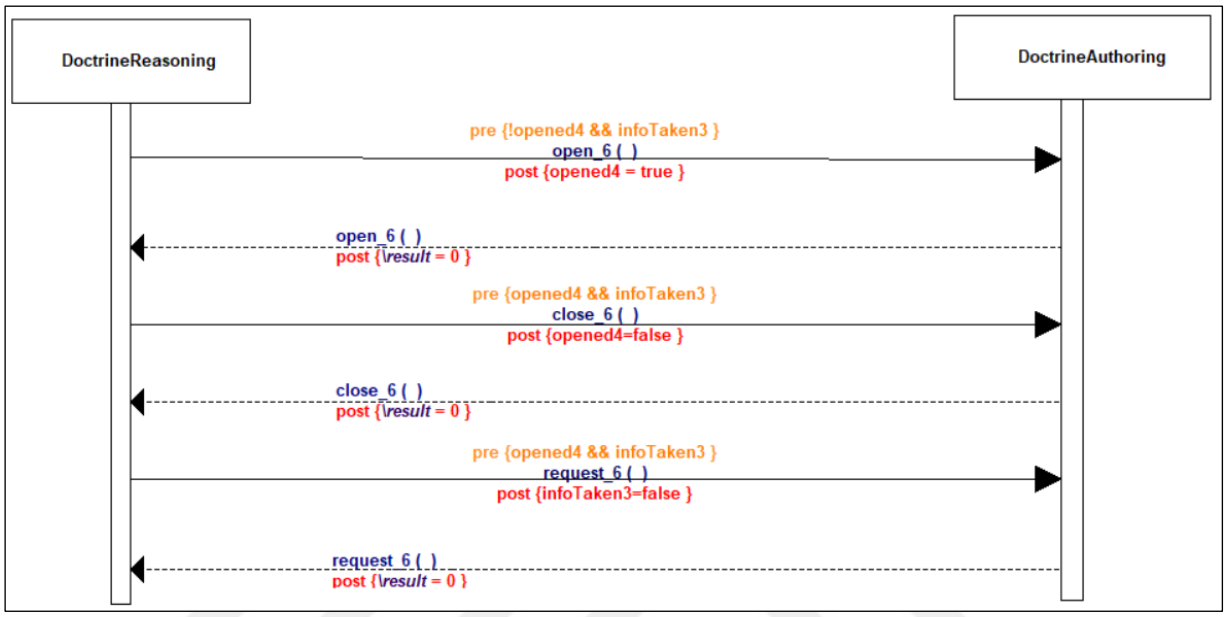
Şekil 7.33: *DoctrineReasoning*'ın *DoctrineAuthoring*'a baglanan istemci portun (*required port*) özellikleri.

DoctrineReasoning'in *DoctorineAuthoring*'den *open_6* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctrineAuthoring*'e *open_6* metot çağrısı yapılır. *DoctrineAuthoring*'den alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened1=true*) göre atama yapılır.

DoctrineReasoning'in *DoctorineAuthoring*'den *close_6* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctorineAuthoring*'a *close_6* metot çağrısı yapılır. *DoctorineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened1=false*) göre atama yapılır.

DoctrineReasoning'in *DoctorineAuthoring*'den *request_6* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened1*) sağlaması gerekiyor. Bu ön şart sağlandığında *DoctorineAuthoring*'a *request_6* metot çağrısı yapılır. *DoctorineAuthoring*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak kabul edilir.

DoctorineAuthoring'in *DoctrineReasoning* ile ilişkisi, *DoctorineAuthoring*'in "*da-p4*" isimli d porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.34'de gösteriliyor.



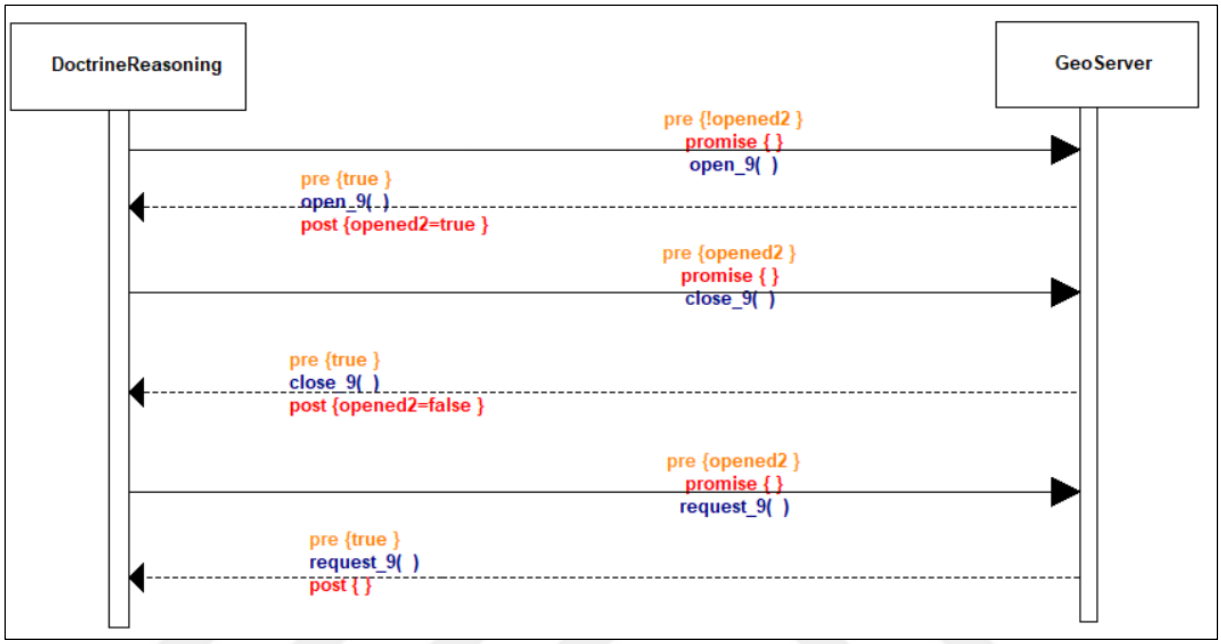
Şekil 7.34: *DoctorineAuthoring*'in *DoctrineReasoning*'e bağlanan sağlayıcı portun (*provided port*) özellikleri.

DoctrineReasoning'den *DoctorineAuthoring*'a *open_6* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened4 && infoTaken3*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened4=true*) atama yapılır *DoctrineReasoning*'a (kesikli ok ile gösterilen) *open_6* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineReasoning'den *DoctorineAuthoring*'a *close_6* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened4 && infoTaken3*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened4=false*) atama yapılır. *DoctrineReasoning*'a (kesikli ok ile gösterilen) *close_6* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineReasoning'den *DoctorineAuthoring*'a *request_5* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened4 && infoTaken3*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*infoTaken3=false*) atama yapılır *DoctrineReasoning*'a (kesikli ok ile gösterilen) *request_6* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineReasoning'in *GeoServer* ile ilişkisi, *DoctrineReasoning*'in "*ds-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.35'de gösteriliyor.



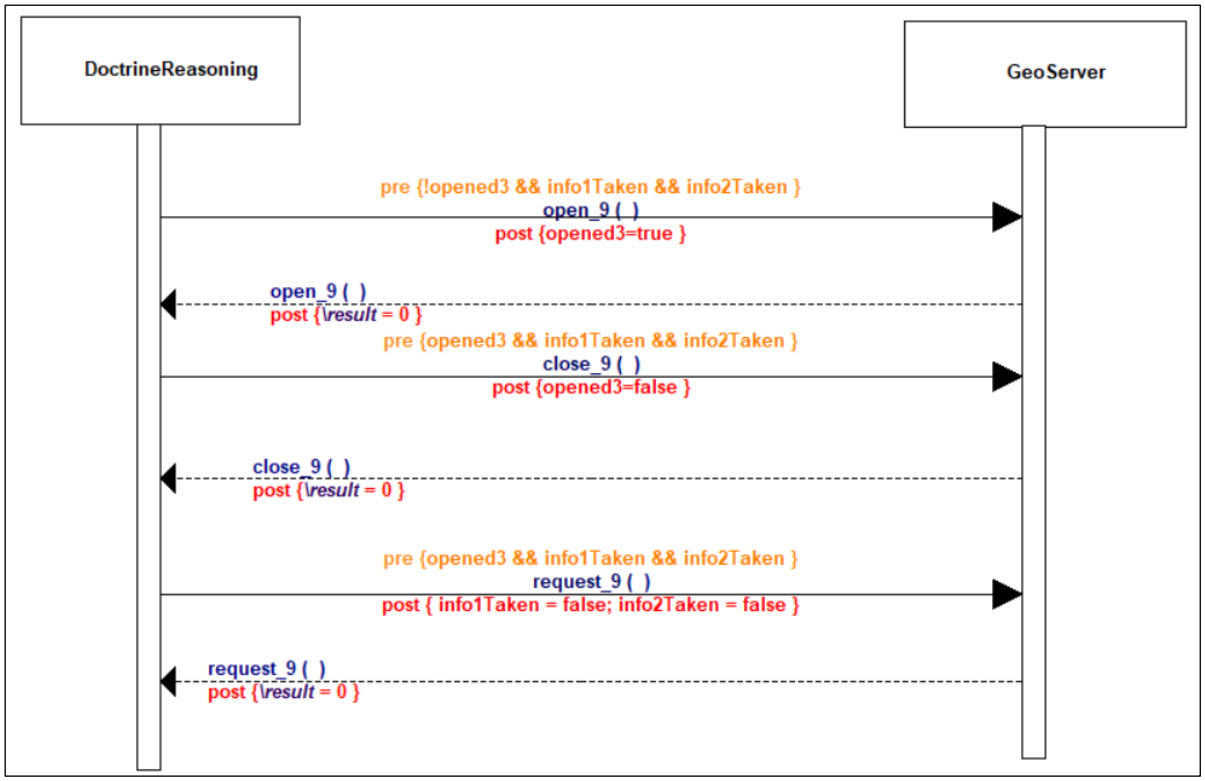
Şekil 7.35: *DoctrineReasoning*'ın *GeoServer*'a bağlanan istemci portun (*required port*) özellikleri.

DoctrineReasoning'in *GeoServer*'den *open_9* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *GeoServer*'e *open_9* metot çağrısı yapılır. *GeoServer*'den alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened2=true*) göre atama yapılır.

DoctrineReasoning'in *GeoServer*'den *close_6* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *GeoServer*'a *close_9* metot çağrısı yapılır. *GeoServer*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened2=false*) göre atama yapılır.

DoctrineReasoning'in *GeoServer*'den *request_9* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *GeoServer*'a *request_9* metot çağrısı yapılır. *GeoServer*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak kabul edilir.

GeoServer'ın *DoctrineReasoning* ile ilişkisi, *GeoServer*'ın "*gs-p1*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.36'da gösteriliyor.



Şekil 7.36: *GeoServer*'ın *DoctrineReasoning*'e bağlanan sağlayıcı portun (*provided port*) özellikleri.

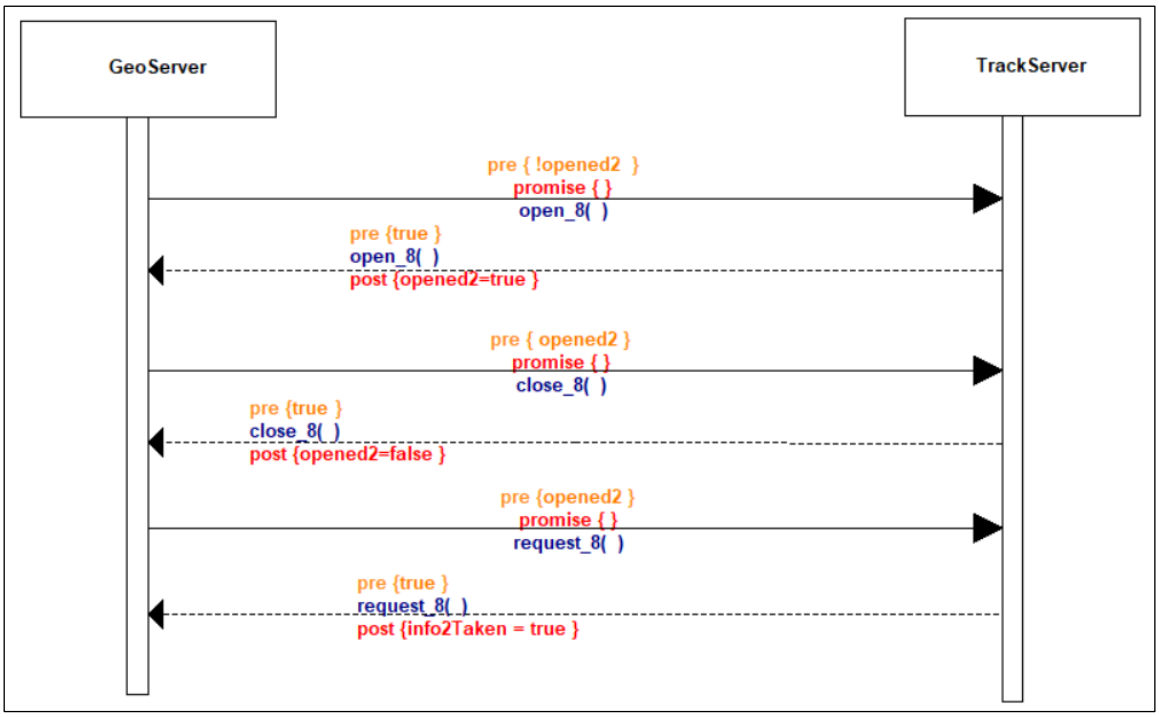
DoctrineReasoning'den *GeoServer*'a *open_9* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened3 && info1Taken && info2Taken*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened3=true*) atama yapılır *DoctrineReasoning*'a (kesikli ok ile gösterilen) *open_9* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineReasoning'den *GeoServer*'a *close_9* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened3 && info1Taken && info2Taken*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened3=false*) atama yapılır. *DoctrineReasoning*'a (kesikli ok ile gösterilen) *close_9* metot çağrısı ile dönüş değeri (0) yollanır.

DoctrineReasoning'den *GeoServer*'a *request_9* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened3 && info1Taken && info2Taken*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre

(*info1Taken=false; info2Taken=false*) atama yapılır *DoctrineReasoning*'a (kesikli ok ile gösterilen) *request_9* metot çağrısı ile dönüş değeri (0) yollanır.

GeoServer'in *TrackServer* ile ilişkisi, *GeoServer*'in "*gs-p3*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.37'de gösteriliyor.



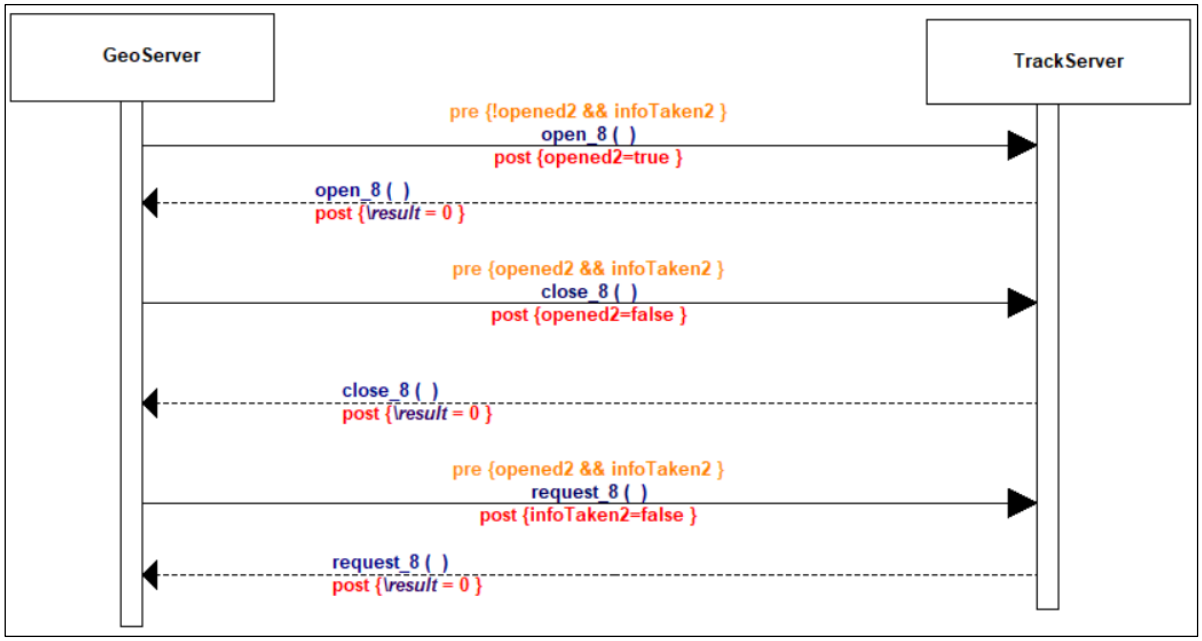
Şekil 7.37: *GeoServer*'ın *TrackServer*'a bağlanan istemci portun (*required port*) özellikleri.

GeoServer'ın *TrackServer*'den *open_8* metodu ile istekte bulunabilmesi için gerekli ön şartı (*!opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *TrackServer*'e *open_8* metot çağrısı yapılır. *TrackServer*'den alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened2=true*) göre atama yapılır.

GeoServer'ın *TrackServer*'den *close_8* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *TrackServer*'a *close_8* metot çağrısı yapılır. *TrackServer*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*opened2=false*) göre atama yapılır.

GeoServer'ın *TrackServer*'den *request_9* metodu ile istekte bulunabilmesi için gerekli ön şartı (*opened2*) sağlaması gerekiyor. Bu ön şart sağlandığında *TrackServer*'a *request_9* metot çağrısı yapılır. *TrackServer*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*info2Taken=true*) göre atama yapılır.

TrackServer'ın *GeoServer* ile ilişkisi, *TrackServer*'ın "*ts-p3*" isimli sağlayıcı porta (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.38'de gösteriliyor.



Şekil 7.38: *TrackServer*'ın *GeoServer*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

GeoServer'den *TrackServer*'a *open_8* metodu vasıtasıyla istek geldiğinde önce ön şart (*!opened2 && infoTaken2*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened2=true*) atama yapılır *GeoServer*'a (kesikli ok ile gösterilen) *open_8* metot çağrısı ile dönüş değeri (0) yollanır.

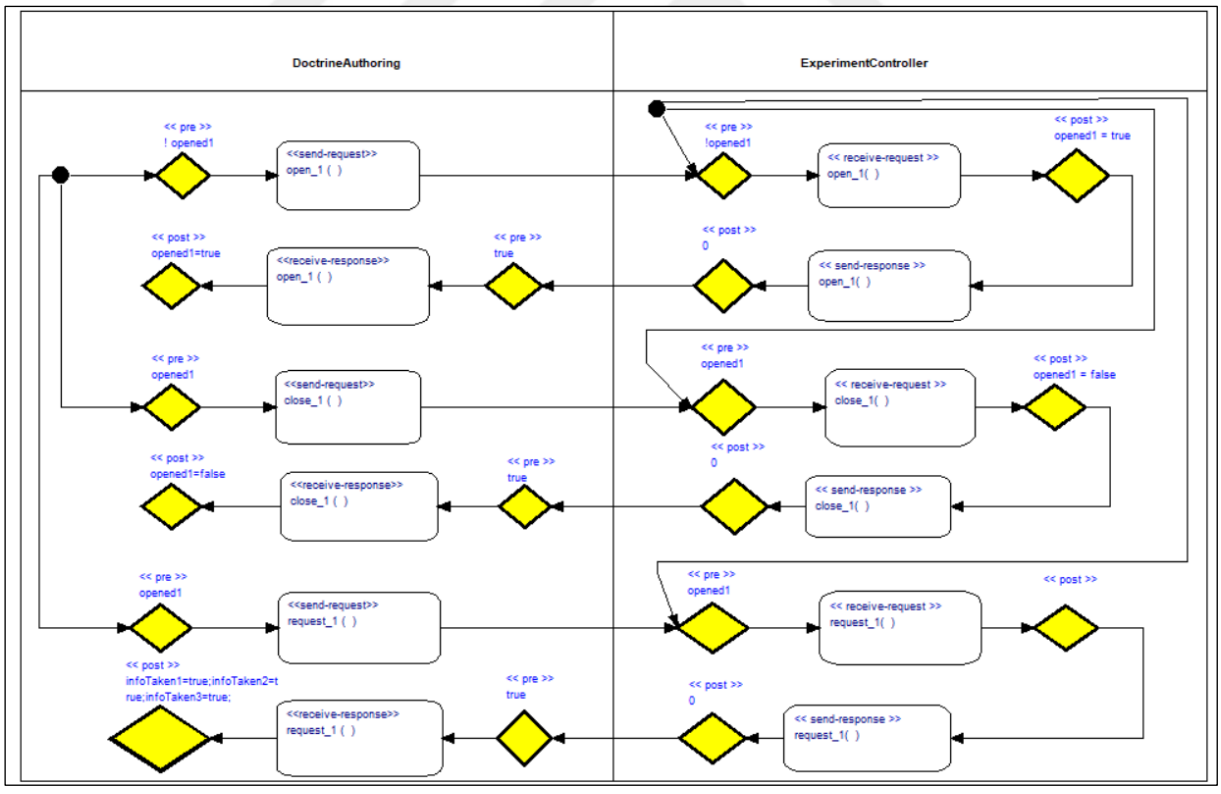
GeoServer'den *TrackServer*'a *close_8* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened2 && info2Taken*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*opened2=false*) atama yapılır. *GeoServer*'a (kesikli ok ile gösterilen) *close_8* metot çağrısı ile dönüş değeri (0) yollanır.

GeoServer'den *TrackServer*'a *request_8* metodu vasıtasıyla istek geldiğinde önce ön şart (*opened2 && info2Taken*) kontrol edilir. Ön şart sağlanıyorsa, son şarta göre (*info2Taken=false*) atama yapılır *GeoServer*'a (kesikli ok ile gösterilen) *request_8* metot çağrısı ile dönüş değeri (0) yollanır.

7.4 AEGİS VAKA ANALİZİ (AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ)

Şekil 7.14'de gösterilen aegis muharebe sistem modelinin aktivite diyagramı ile genişletilmiş şeklinde bileşenlerinin data verileri ve özellikleri Şekil 7.15,Şekil 7.16, Şekil 7.17, Şekil 7.18, Şekil 7.19 ve Şekil 7.20'deki gibi aynıdır.

DoctrineAuthoring ile *ExperimentController* arasındaki ilişki, *DoctrineAuthoring*'ın "*da-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.39'da gösteriliyor.



Şekil 7.39: *DoctrineAuthoring-ExperimentController* aktivite diyagramı.

Şekil 7.39'da gösterilen *DoctrineAuthoring-ExperimentController* aktivite diyagramında, *DoctrineAuthoring*'ın *ExperimentController*'a *open_1* metot çağrısı yapmadan önce ön/sonraki

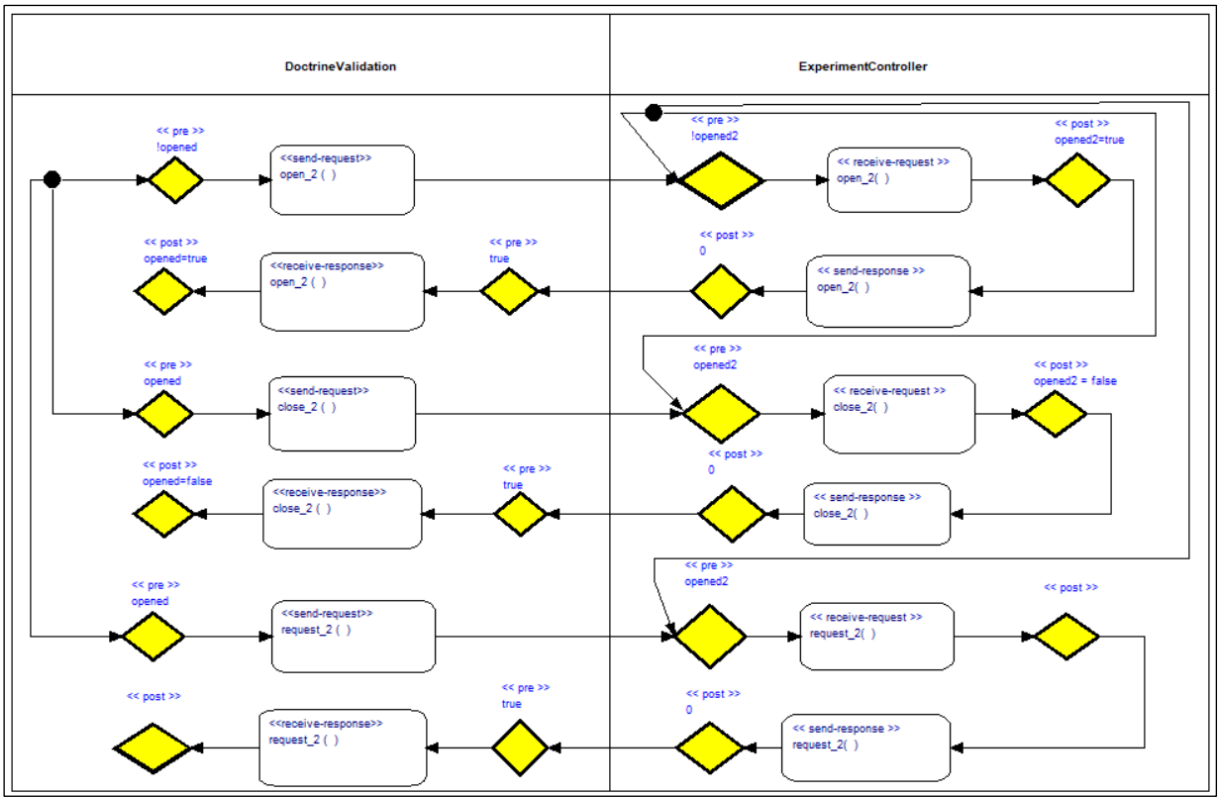
koşul sembolündeki ön şartı (*!opened1*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_1* metot çağrı aktivitesi gerçekleştirilir. *open_1* metodu, *ExperimentController* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened1=true*) sağlanır ve (*send-response*) *open_1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineAuthoring* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineAuthoring*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened1=true*) sağlar.

DoctrineAuthoring'ın *ExperimentController*'a *close_1* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened1*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_1* metot çağrı aktivitesi gerçekleştirilir. *close_1* metodu, *ExperimentController* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened1=false*) sağlanır ve (*send-response*) *close_1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineAuthoring* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineAuthoring*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened1=false*) sağlar.

DoctrineAuthoring'ın *ExperimentController*'a *request_1* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened1*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_1* metot çağrı aktivitesi gerçekleştirilir. *request_1* metodu, *ExperimentController* bileşenin sağlayıcı porta (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_1* metot çağrı aktivitesi gerçekleştirilir. Sonra (*send-response*) *request_1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineAuthoring* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir.

DoctrineAuthoring, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*infoTaken1=true; infoTaken2=true; infoTaken3=true*) sağlar.

DoctrineValidation ile *ExperimentController* arasındaki ilişki, *DoctrineValidation*'ın "*dv-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.40'da gösteriliyor.



Şekil 7.40: *DoctrineValidation-ExperimentController* aktivite diyagramı.

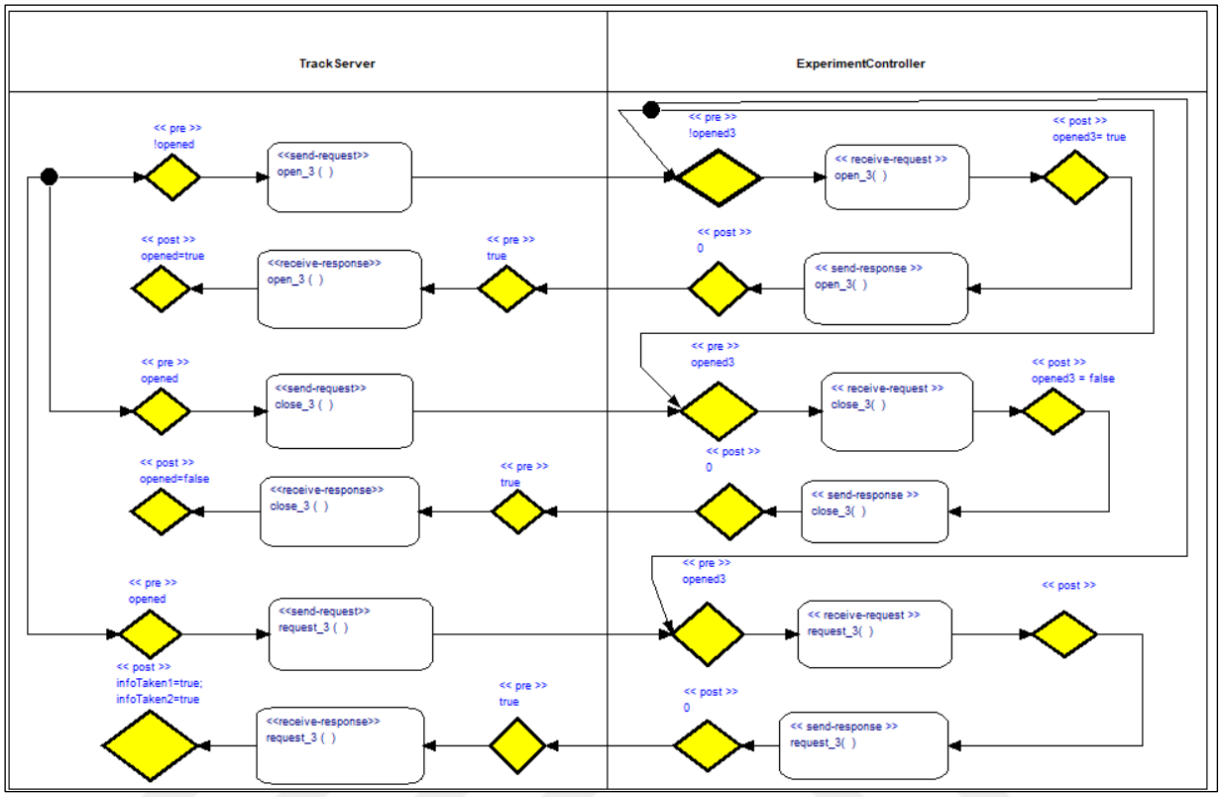
Şekil 7.40'da gösterilen *DoctrineValidation-ExperimentController* aktivite diyagramında, *DoctrineValidation*'ın *ExperimentController*'a *open_2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_2* metot çağrı aktivitesi gerçekleştirilir. *open_2* metodu, *ExperimentController* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened2=true*) sağlanır ve (*send-response*) *open_2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation*

bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened=true*) sağlar.

DoctrineValidation'ın *ExperimentController*'a *close_2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_2* metot çağrı aktivitesi gerçekleştirilir. *close_2* metodu, *ExperimentController* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_2* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened2=false*) sağlanır ve (*send-response*) *close_2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened=false*) sağlar.

DoctrineValidation'ın *ExperimentController*'a *request_2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_2* metot çağrı aktivitesi gerçekleştirilir. *request_2* metodu, *ExperimentController* bileşenin sağlayıcı porta (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_2* metot çağrı aktivitesi gerçekleştirilir. Sonra (*send-response*) *request_2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_1* metot çağrı aktivitesi ile alır.

TrackServer ile *ExperimentController* arasındaki ilişki, *DoctrineAuthoring*'ın "*ts-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.41'de gösteriliyor.



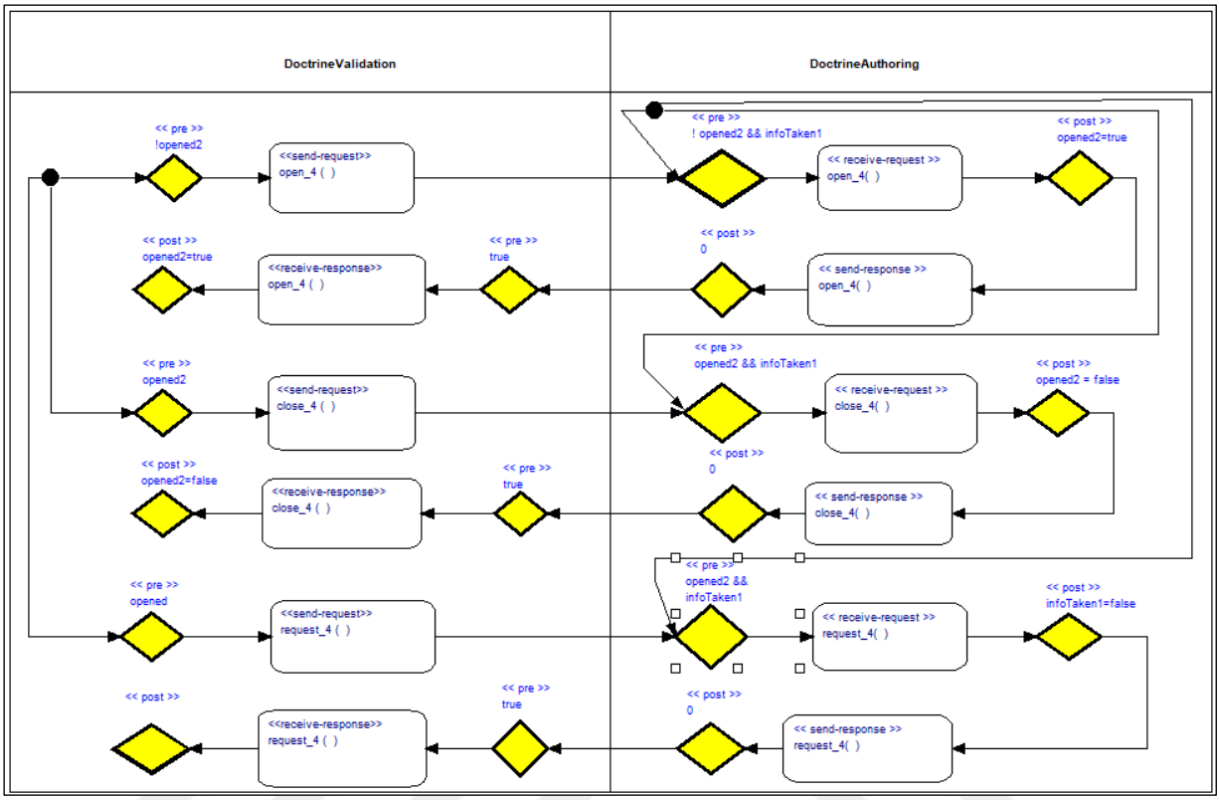
Şekil 7.41: *TrackServer-ExperimentController* aktivite diyagramı.

Şekil 7.41'de gösterilen *TrackServer-ExperimentController* aktivite diyagramında, *TrackServer*'ın *ExperimentController*'a *open_3* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_3* metot çağrı aktivitesi gerçekleştirilir. *open_3* metodu, *ExperimentController* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened3*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_3* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened3=true*) sağlanır ve (*send-response*) *open_3* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *TrackServer* bileşenin istemci porta (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *TrackServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_3* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened=true*) sağlar.

TrackServer'ın *ExperimentController*'a *close_3* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_3* metot çağrı aktivitesi gerçekleştirilir. *close_3* metodu, *ExperimentController* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened3*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_3* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened3=false*) sağlanır ve (*send-response*) *close_3* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *TrackServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *TrackServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_3* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened=false*) sağlar.

TrackServer'ın *ExperimentController*'a *request_3* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_3* metot çağrı aktivitesi gerçekleştirilir. *request_3* metodu, *ExperimentController* bileşenin sağlayıcı porta (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened3*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_3* metot çağrı aktivitesi gerçekleştirilir. Sonra (*send-response*) *request_3* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *TrackServer* bileşenin istemci porta (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *TrackServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_3* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*infoTaken1=true; infoTaken2=true;*) sağlar.

DoctrineValidation ile *DoctrineAuthoring* arasındaki ilişki, *DoctrineValidation*'ın "*dv-p2*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.42'de gösteriliyor.



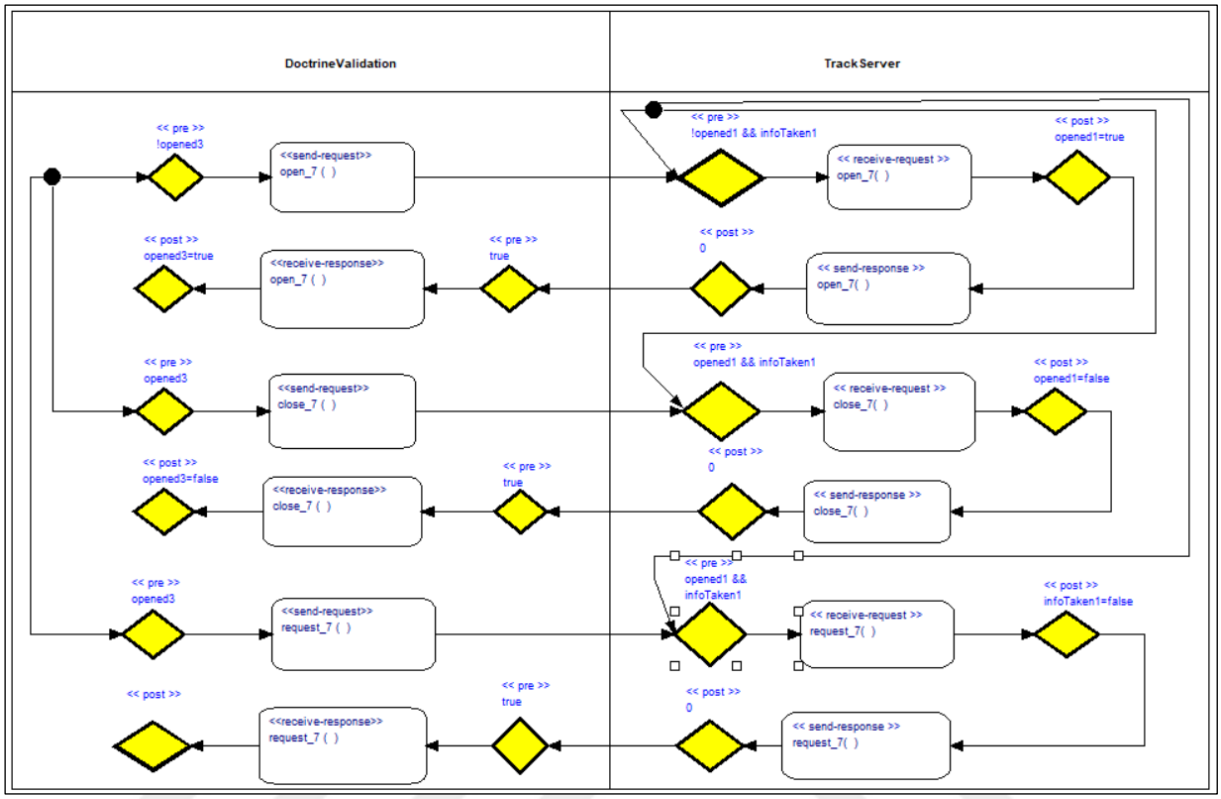
Şekil 7.42: *DoctrineValidation-DoctrineAuthoring* aktivite diyagramı.

Şekil 7.42'de gösterilen *DoctrineValidation-DoctrineAuthoring* aktivite diyagramında, *DoctrineValidation*'ın *DoctrineAuthoring*'e *open_4* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_4* metot çağrı aktivitesi gerçekleştirilir. *open_4* metodu, *DoctrineAuthoring* bileşenin sağlayıcı porta (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened2 && infoTaken1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_4* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened2=true*) sağlanır ve (*send-response*) *open_4* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_4* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened2=true*) sağlar.

DoctrineValidation'ın *DoctrineAuthoring*'e *close_4* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_4* metot çağrı aktivitesi gerçekleştirilir. *close_4* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened2 && infoTaken1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_4* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened2=false*) sağlanır ve (*send-response*) *close_4* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_4* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened2=false*) sağlar.

DoctrineValidation'ın *DoctrineAuthoring*'e *request_4* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_4* metot çağrı aktivitesi gerçekleştirilir. *request_4* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened2 && infoTaken1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_4* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*infoTaken1=false*) sağlanır ve (*send-response*) *request_4* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_4* metot çağrı aktivitesi ile alır.

DoctrineValidation ile *TrackServer* arasındaki ilişki, *DoctrineValidation*'ın "*dv-p3*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.43'de gösteriliyor.



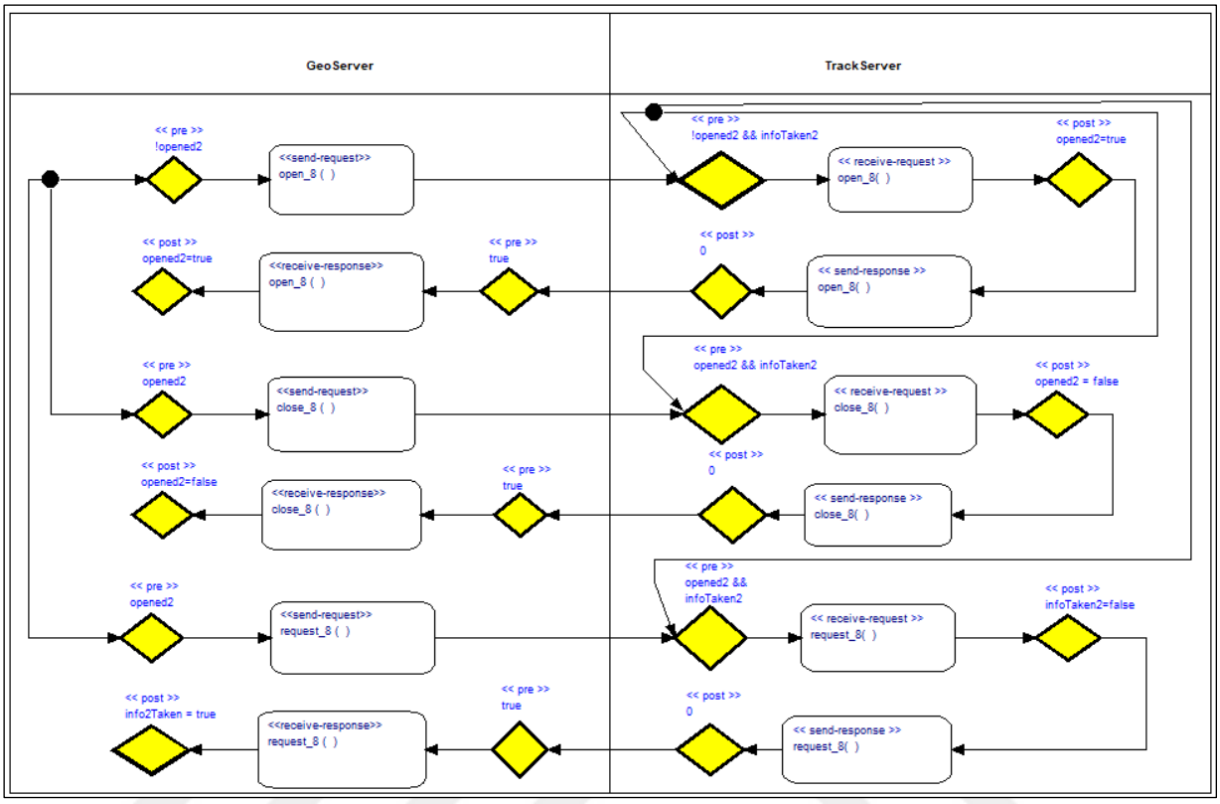
Şekil 7.43: *DoctrineValidation-TrackServer* aktivite diyagramı.

Şekil 7.43'de gösterilen *DoctrineValidation-TrackServer* aktivite diyagramında, *DoctrineValidation*'ın *TrackServer*'a *open_7* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened3*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_7* metot çağrı aktivitesi gerçekleştirilir. *open_7* metodu, *TrackServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened1 && infoTaken1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_7* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened1=true*) sağlanır ve (*send-response*) *open_7* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci bağlantı portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_7* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened3=true*) sağlar.

DoctrineValidation'ın *TrackServer*'a *close_7* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened3*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_7* metot çağrı aktivitesi gerçekleştirilir. *close_7* metodu, *TrackServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened1 && infoTaken1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_7* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened1=false*) sağlanır ve (*send-response*) *close_7* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_7* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened3=false*) sağlar.

DoctrineValidation'ın *TrackServer*'a *request_7* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened3*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_7* metot çağrı aktivitesi gerçekleştirilir. *request_7* metodu, *TrackServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened1 && infoTaken1*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_7* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*infoTaken1=false*) sağlanır ve (*send-response*) *request_7* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineValidation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineValidation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_7* metot çağrı aktivitesi ile alır.

Geoserver ile *TrackServer* arasındaki ilişki, *GeoServer*'ın "*gs-p3*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.44'de gösteriliyor.



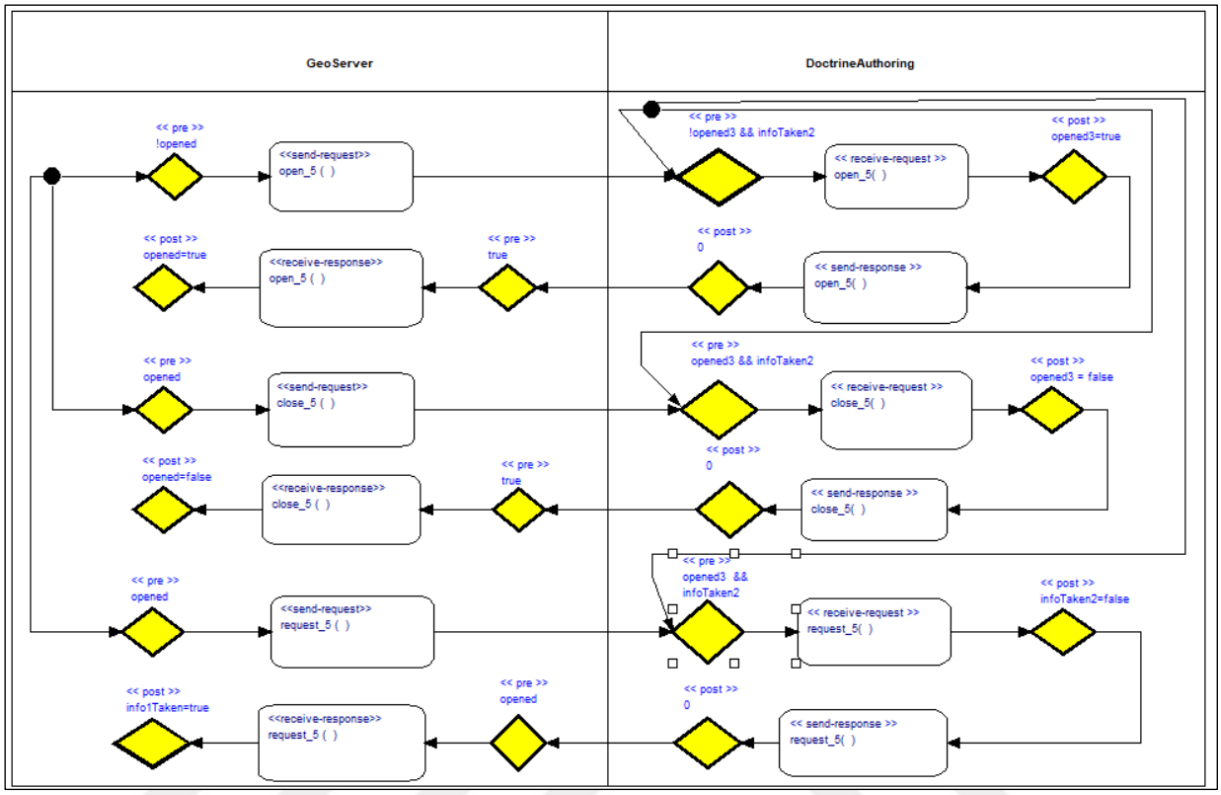
Şekil 7.44: *GeoServer-TrackServer* aktivite diyagramı.

Şekil 7.44'de gösterilen *GeoServer-TrackServer* aktivite diyagramında, *GeoServer*'ın *TrackServer*'a *open_8* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_8* metot çağrı aktivitesi gerçekleştirilir. *open_8* metodu, *TrackServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened2 && infoTaken2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_8* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened2=true*) sağlanır ve (*send-response*) *open_8* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *GeoServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *GeoServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_8* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened2=true*) sağlar.

GeoServer'ın *TrackServer*'a *close_8* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_8* metot çağrı aktivitesi gerçekleştirilir. *close_8* metodu, *TrackServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened2 && infoTaken2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_8* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened2=false*) sağlanır ve (*send-response*) *close_8* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *GeoServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *GeoServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_8* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened2=false*) sağlar.

GeoServer'ın *TrackServer*'a *request_8* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_8* metot çağrı aktivitesi gerçekleştirilir. *request_8* metodu, *TrackServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened2 && infoTaken2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_8* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*infoTaken2=false*) sağlanır ve (*send-response*) *request_8* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *GeoServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *GeoServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_8* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*info2Taken=true*) sağlar.

Geoserver ile *DoctrineAuthoring* arasındaki ilişki, *GeoServer*'ın "*gs-p2*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.45'de gösteriliyor.



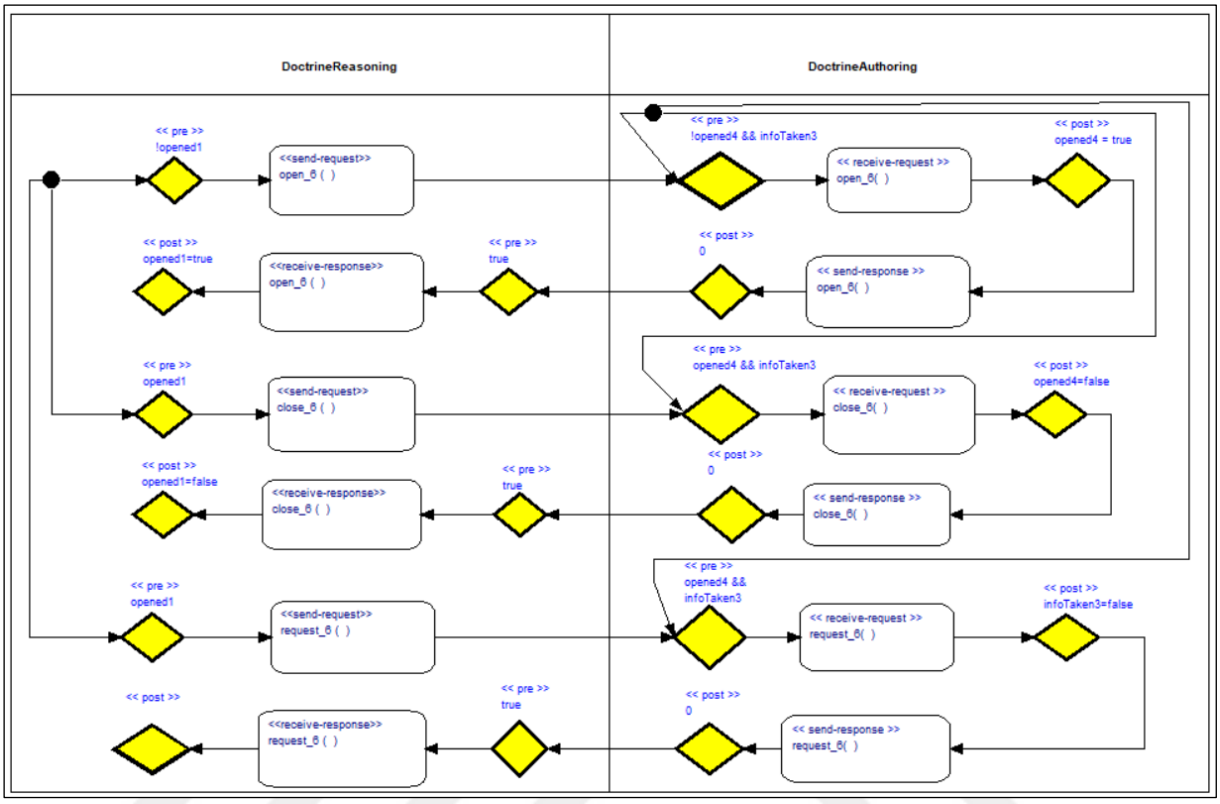
Şekil 7.45: *GeoServer-DoctrineAuthoring* aktivite diyagramı.

Şekil 7.45'de gösterilen *GeoServer-DoctrineAuthoring* aktivite diyagramında, *GeoServer*'ın *DoctrineAuthoring*'e *open_5* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_5* metot çağrı aktivitesi gerçekleştirilir. *open_5* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened3 && infoTaken2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_5* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened3=true*) sağlanır ve (*send-response*) *open_5* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *GeoServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *GeoServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_8* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened=true*) sağlar.

GeoServer'ın *DoctrineAuthoring*'e *close_5* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_5* metot çağrı aktivitesi gerçekleştirilir. *close_5* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened3 && infoTaken2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_5* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened3=false*) sağlanır ve (*send-response*) *close_8* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *GeoServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *GeoServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_5* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened=false*) sağlar.

GeoServer'ın *DoctrineAuthoring*'e *request_5* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_5* metot çağrı aktivitesi gerçekleştirilir. *request_5* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened3 && infoTaken2*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_5* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*infoTaken2=false*) sağlanır ve (*send-response*) *request_5* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *GeoServer* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *GeoServer*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_5* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*info1Taken=true*) sağlar.

DoctrineReasoning ile *DoctrineAuthoring* arasındaki ilişki, *DoctrineReasoning*'in "*ds-p2*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.46'da gösteriliyor.



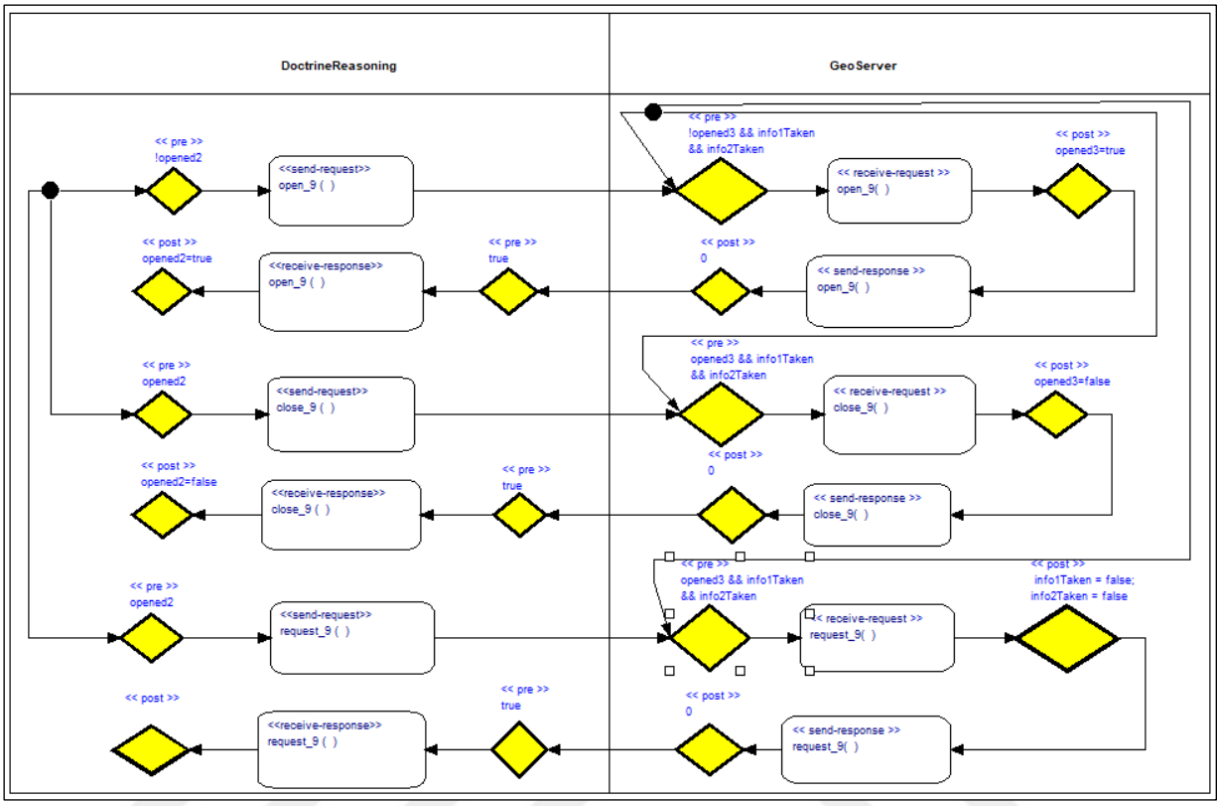
Şekil 7.46: *DoctrineReasoning-DoctrineAuthoring* aktivite diyagramı.

Şekil 7.46'da gösterilen *DoctrineReasoning-DoctrineAuthoring* aktivite diyagramında, *DoctrineReasoning*'ın *DoctrineAuthoring*'e *open_6* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened1*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_6* metot çağrı aktivitesi gerçekleştirilir. *open_6* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened4 && infoTaken3*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_6* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened4=true*) sağlanır ve (*send-response*) *open_6* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineReasoning* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineReasoning*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_6* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened1=true*) sağlar.

DoctrineReasoning'ın *DoctrineAuthoring*'e *close_6* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened1*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_6* metot çağrı aktivitesi gerçekleştirilir. *close_6* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened4 && infoTaken3*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_6* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened4=false*) sağlanır ve (*send-response*) *close_6* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineReasoning* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineReasoning*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_6* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened1=false*) sağlar.

DoctrineReasoning'ın *DoctrineAuthoring*'e *request_6* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened1*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_6* metot çağrı aktivitesi gerçekleştirilir. *request_6* metodu, *DoctrineAuthoring* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened4 && infoTaken3*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_6* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*infoTaken3=false*) sağlanır ve (*send-response*) *request_6* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineReasoning* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineReasoning*, cevabı ön şartsız (*true*) olarak (*receive-response*) *request_6* metot çağrı aktivitesi ile alır.

DoctrineReasoning ile *GeoServer* arasındaki ilişki, *DoctrineReasoning*'in "*ds-p1*" isimli istemci porta (*required port*) sağ tıklama ile açılan diyagram Şekil 7.47'de gösteriliyor.



Şekil 7.47: *DoctrineReasoning-GeoServer* aktivite diyagramı.

Şekil 7.47'de gösterilen *DoctrineReasoning-GeoServer* aktivite diyagramında, *DoctrineReasoning*'ın *GeoServer*'a *open_9* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*!opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *open_9* metot çağrı aktivitesi gerçekleştirilir. *open_9* metodu, *GeoServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*!opened3 && info1Taken && info2Taken*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *open_9* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened3=true*) sağlanır ve (*send-response*) *open_9* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineReasoning* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=0*) belirtir. *DoctrineReasoning*, cevabı ön şartsız (*true*) olarak (*receive-response*) *open_9* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened2=true*) sağlar.

DoctrineReasoning'ın *Geoserver*'a *close_9* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *close_9* metot çağrı aktivitesi gerçekleştirilir. *close_6* metodu, *GeoServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened3 && info1Taken && info2Taken*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *close_9* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*opened3=false*) sağlanır ve (*send-response*) *close_9* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineReasoning* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineReasoning*, cevabı ön şartsız (*true*) olarak (*receive-response*) *close_9* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*opened2=false*) sağlar.

DoctrineReasoning'ın *GeoServer*'a *request_9* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*opened2*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *request_9* metot çağrı aktivitesi gerçekleştirilir. *request_9* metodu, *GeoServer* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartı (*opened3 && info1Taken && info2Taken*) kontrol edilir. Bu şart sağlanıyorsa, (*receive-request*) *request_9* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*info1Taken=false; info2Taken=false;*) sağlanır ve (*send-response*) *request_9* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DoctrineReasoning* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=0*) belirtir. *DoctrineReasoning,* cevabı ön şartsız (*true*) olarak (*receive-response*) *request_9* metot çağrı aktivitesi ile alır.

Aegis muhabere sistem modelinin, MetaEdit+ editörü dönüştürücüsünden elde edilen ProMeLa dosyası (aeg.pml) Liste 7.3'de gösterilmektedir.

Liste 7.3: aeg.pml dosyası.

```
1       mtype = {close_1,request_1,open_1,open_2,request_2,close_2,close_3,request_3,
2                open_3,open_4,close_4,request_4,open_9,request_9,close_9,close_6,
3               request_6,open_6,open_8,close_8,request_8,open_7,request_7,close_7,
4               request_5,close_5,open_5}
5
6       chan ch_daPort1_REQ= [1] of{ mtype  };
```

Liste 7.3: aeg.pml dosyası (devam).

```
7       chan ch_dvPort1_REQ= [1] of{ mtype  };
8       chan ch_tsPort1_REQ= [1] of{ mtype  };
9       chan ch_dvPort2_REQ= [1] of{ mtype  };
10      chan ch_dsPort1_REQ= [1] of{ mtype  };
11      chan ch_dsPort2_REQ= [1] of{ mtype  };
12      chan ch_gsPort3_REQ= [1] of{ mtype  };
13      chan ch_dvPort3_REQ= [1] of{ mtype  };
14      chan ch_gsPort2_REQ= [1] of{ mtype  };
15
16      chan ch_daPort1_RES= [1] of{ mtype, int} ;
17      chan ch_dvPort1_RES= [1] of{ mtype, int} ;
18      chan ch_tsPort1_RES= [1] of{ mtype, int} ;
19      chan ch_dvPort2_RES= [1] of{ mtype, int} ;
20      chan ch_dsPort1_RES= [1] of{ mtype, int} ;
21      chan ch_dsPort2_RES= [1] of{ mtype, int} ;
22      chan ch_gsPort3_RES= [1] of{ mtype, int} ;
23      chan ch_dvPort3_RES= [1] of{ mtype, int} ;
24      chan ch_gsPort2_RES= [1] of{ mtype, int} ;
25
26
27      proctype ExperimentController() {
28       bool opened3=false;
29       bool opened2=false;
30       bool opened1=false;
31       int result;
32
33       do
34       ::ch_daPort1_REQ ?open_1->
35         if
36         ::(!opened1)->
37              opened1 = true
38              ch_daPort1_RES !open_1,0;
39         ::!( (!opened1)  )->
40              ch_daPort1_REQ !open_1;
41         fi
42       ::ch_daPort1_REQ ?close_1->
43         if
44         ::(opened1)->
45              opened1 = false
46              ch_daPort1_RES !close_1,0;
47         ::!( (opened1)  )->
48              ch_daPort1_REQ !close_1;
49         fi
50       ::ch_daPort1_REQ ?request_1->
51         if
52         ::(opened1)->
53              ch_daPort1_RES !request_1,0;
54         ::!( (opened1)  )->
55              ch_daPort1_REQ !request_1;
56         fi
57       ::ch_dvPort1_REQ ?request_2->
58         if
59         ::(opened2)->
60              ch_dvPort1_RES !request_2,0;
61         ::!( (opened2)  )->
62              ch_dvPort1_REQ !request_2;
63         fi
64       ::ch_dvPort1_REQ ?close_2->
65         if
66         ::(opened2)->
67              opened2=false
68              ch_dvPort1_RES !close_2,0;
69         ::!( (opened2)  )->
70              ch_dvPort1_REQ !close_2;
```

Liste 7.3: aeg.pml dosyası (devam).

```
71        fi
72      ::ch_dvPort1_REQ ?open_2->
73        if
74        ::(!opened2)->
75              opened2= true
76              ch_dvPort1_RES !open_2,0;
77        ::!( (!opened2)  )->
78              ch_dvPort1_REQ !open_2;
79        fi
80      ::ch_tsPort1_REQ ?request_3->
81        if
82        ::(opened3)->
83              ch_tsPort1_RES !request_3,0;
84        ::!( (opened3)  )->
85              ch_tsPort1_REQ !request_3;
86        fi
87      ::ch_tsPort1_REQ ?close_3->
88        if
89        ::(opened3)->
90              opened3=false
91              ch_tsPort1_RES !close_3,0;
92        ::!( (opened3)  )->
93              ch_tsPort1_REQ !close_3;
94        fi
95      ::ch_tsPort1_REQ ?open_3->
96        if
97        ::(!opened3)->
98              opened3= true
99              ch_tsPort1_RES !open_3,0;
100       ::!( (!opened3)  )->
101             ch_tsPort1_REQ !open_3;
102       fi
103     od
104   }
105   proctype DoctrineAuthoring() {
106     bool opened1=false;
107     bool opened3=false;
108     bool infoTaken1=false;
109     bool infoTaken3=false;
110     bool opened4=false;
111     bool infoTaken2=false;
112     bool opened2=false;
113     int result;
114     int responseWaiting= false;
115
116     do
117     ::opened1 && responseWaiting==false ->
118       responseWaiting=true ;
119       ch_daPort1_REQ !close_1;
120     ::ch_daPort1_RES ?close_1,result->
121       if
122       ::(true)->opened1=false;
123       ::!( true )->assert(false);
124       fi ;
125       responseWaiting=false ;
126     ::opened1 && responseWaiting==false ->
127       responseWaiting=true ;
128       ch_daPort1_REQ !request_1;
129     ::ch_daPort1_RES ?request_1,result->
130       if
131       ::(true)->infoTaken1=true;infoTaken2=true;infoTaken3=true;;
132       ::!( true )->assert(false);
133       fi ;
134       responseWaiting=false ;
```

Liste 7.3: aeg.pml dosyası (devam).

```
135     ::!opened1 && responseWaiting==false ->
136       responseWaiting=true ;
137       ch_daPort1_REQ !open_1;
138     ::ch_daPort1_RES ?open_1,result->
139       if
140       ::(true)->opened1=true;
141       ::!( true )->assert(false);
142       fi ;
143       responseWaiting=false ;
144     ::ch_dvPort2_REQ ?close_4->
145       if
146       ::(opened2 && infoTaken1)->
147            opened2=false
148            ch_dvPort2_RES !close_4,0;
149       ::!( (opened2 && infoTaken1)  )->
150            ch_dvPort2_REQ !close_4;
151       fi
152     ::ch_dvPort2_REQ ?request_4->
153       if
154       ::(opened2 && infoTaken1)->
155            infoTaken1=false
156            ch_dvPort2_RES !request_4,0;
157       ::!( (opened2 && infoTaken1)  )->
158            ch_dvPort2_REQ !request_4;
159       fi
160     ::ch_dvPort2_REQ ?open_4->
161       if
162       ::(!opened2 && infoTaken1)->
163            opened2=true
164            ch_dvPort2_RES !open_4,0;
165       ::!( (!opened2 && infoTaken1)  )->
166            ch_dvPort2_REQ !open_4;
167       fi
168     ::ch_gsPort2_REQ ?request_5->
169       if
170       ::(opened3 && infoTaken2)->
171            infoTaken2=false
172            ch_gsPort2_RES !request_5,0;
173       ::!( (opened3 && infoTaken2)  )->
174            ch_gsPort2_REQ !request_5;
175       fi
176     ::ch_gsPort2_REQ ?close_5->
177       if
178       ::(opened3 && infoTaken2)->
179            opened3=false
180            ch_gsPort2_RES !close_5,0;
181       ::!( (opened3 && infoTaken2)  )->
182            ch_gsPort2_REQ !close_5;
183       fi
184     ::ch_gsPort2_REQ ?open_5->
185       if
186       ::(!opened3 && infoTaken2)->
187            opened3=true
188            ch_gsPort2_RES !open_5,0;
189       ::!( (!opened3 && infoTaken2)  )->
190            ch_gsPort2_REQ !open_5;
191       fi
192     ::ch_dsPort2_REQ ?open_6->
193       if
194       ::(!opened4 && infoTaken3)->
195            opened4 = true
196            ch_dsPort2_RES !open_6,0;
197       ::!( (!opened4 && infoTaken3)  )->
```

Liste 7.3: aeg.pml dosyası (devam).

```
198             ch_dsPort2_REQ !open_6;
199       fi
200     ::ch_dsPort2_REQ ?close_6->
201       if
202       ::(opened4 && infoTaken3)->
203             opened4=false
204             ch_dsPort2_RES !close_6,0;
205       ::!( (opened4 && infoTaken3)  )->
206             ch_dsPort2_REQ !close_6;
207       fi
208     ::ch_dsPort2_REQ ?request_6->
209       if
210       ::(opened4 && infoTaken3)->
211             infoTaken3=false
212             ch_dsPort2_RES !request_6,0;
213       ::!( (opened4 && infoTaken3)  )->
214             ch_dsPort2_REQ !request_6;
215       fi
216     od
217    }
218    proctype DoctrineValidation() {
219     bool opened=false;
220     bool opened2=false;
221     bool opened3=false;
222     int result;
223     int responseWaiting= false;
224
225     do
226     ::!opened && responseWaiting==false ->
227       responseWaiting=true ;
228       ch_dvPort1_REQ !open_2;
229     ::ch_dvPort1_RES ?open_2,result->
230       if
231       ::(true)->opened = true;
232       ::!( true )->assert(false);
233       fi ;
234       responseWaiting=false ;
235     ::opened && responseWaiting==false ->
236       responseWaiting=true ;
237       ch_dvPort1_REQ !request_2;
238     ::ch_dvPort1_RES ?request_2,result->
239       if
240       ::(true)->;
241       ::!( true )->assert(false);
242       fi ;
243       responseWaiting=false ;
244     ::opened && responseWaiting==false ->
245       responseWaiting=true ;
246       ch_dvPort1_REQ !close_2;
247     ::ch_dvPort1_RES ?close_2,result->
248       if
249       ::(true)->opened=false;
250       ::!( true )->assert(false);
251       fi ;
252       responseWaiting=false ;
253     ::!opened2 && responseWaiting==false ->
254       responseWaiting=true ;
255       ch_dvPort2_REQ !open_4;
256     ::ch_dvPort2_RES ?open_4,result->
257       if
258       ::(true)->opened2=true;
259       ::!( true )->assert(false);
260       fi ;
```

Liste 7.3: aeg.pml dosyası (devam).

```
261       responseWaiting=false ;
262     ::opened2 && responseWaiting==false ->
263       responseWaiting=true ;
264       ch_dvPort2_REQ !close_4;
265     ::ch_dvPort2_RES ?close_4,result->
266       if
267       ::(true)->opened2=false;
268       ::!( true )->assert(false);
269       fi ;
270       responseWaiting=false ;
271     ::opened && responseWaiting==false ->
272       responseWaiting=true ;
273       ch_dvPort2_REQ !request_4;
274     ::ch_dvPort2_RES ?request_4,result->
275       if
276       ::(true)->;
277       ::!( true )->assert(false);
278       fi ;
279       responseWaiting=false ;
280     ::!opened3 && responseWaiting==false ->
281       responseWaiting=true ;
282       ch_dvPort3_REQ !open_7;
283     ::ch_dvPort3_RES ?open_7,result->
284       if
285       ::(true)->opened3=true;
286       ::!( true )->assert(false);
287       fi ;
288       responseWaiting=false ;
289     ::opened3 && responseWaiting==false ->
290       responseWaiting=true ;
291       ch_dvPort3_REQ !request_7;
292     ::ch_dvPort3_RES ?request_7,result->
293       if
294       ::(true)->;
295       ::!( true )->assert(false);
296       fi ;
297       responseWaiting=false ;
298     ::opened3 && responseWaiting==false ->
299       responseWaiting=true ;
300       ch_dvPort3_REQ !close_7;
301     ::ch_dvPort3_RES ?close_7,result->
302       if
303       ::(true)->opened3=false;
304       ::!( true )->assert(false);
305       fi ;
306       responseWaiting=false ;
307     od
308    }
309    proctype TrackServer() {
310     bool opened=false;
311     bool opened1=false;
312     bool opened2=false;
313     bool infoTaken1=false;
314     bool infoTaken2=false;
315     int result;
316     int responseWaiting= false;
317
318     do
319     ::opened && responseWaiting==false ->
320       responseWaiting=true ;
321       ch_tsPort1_REQ !close_3;
322     ::ch_tsPort1_RES ?close_3,result->
323       if
```

Liste 7.3: aeg.pml dosyası (devam).

```
324       ::(true)->opened=false;
325       ::!( true )->assert(false);
326       fi ;
327       responseWaiting=false ;
328     ::opened && responseWaiting==false ->
329       responseWaiting=true ;
330       ch_tsPort1_REQ !request_3;
331     ::ch_tsPort1_RES ?request_3,result->
332       if
333       ::(true)->infoTaken1=true; infoTaken2=true;
334       ::!( true )->assert(false);
335       fi ;
336       responseWaiting=false ;
337     ::!opened && responseWaiting==false ->
338       responseWaiting=true ;
339       ch_tsPort1_REQ !open_3;
340     ::ch_tsPort1_RES ?open_3,result->
341       if
342       ::(true)->opened=true;
343       ::!( true )->assert(false);
344       fi ;
345       responseWaiting=false ;
346     ::ch_gsPort3_REQ ?request_8->
347       if
348       ::(opened2 && infoTaken2)->
349            infoTaken2=false
350            ch_gsPort3_RES !request_8,0;
351       ::!( (opened2 && infoTaken2)  )->
352            ch_gsPort3_REQ !request_8;
353       fi
354     ::ch_gsPort3_REQ ?close_8->
355       if
356       ::(opened2 && infoTaken2)->
357            opened2=false
358            ch_gsPort3_RES !close_8,0;
359       ::!( (opened2 && infoTaken2)  )->
360            ch_gsPort3_REQ !close_8;
361       fi
362     ::ch_gsPort3_REQ ?open_8->
363       if
364       ::(!opened2 && infoTaken2)->
365            opened2=true
366            ch_gsPort3_RES !open_8,0;
367       ::!( (!opened2 && infoTaken2)  )->
368            ch_gsPort3_REQ !open_8;
369       fi
370     ::ch_dvPort3_REQ ?open_7->
371       if
372       ::(!opened1 && infoTaken1)->
373            opened1=true
374            ch_dvPort3_RES !open_7,0;
375       ::!( (!opened1 && infoTaken1)  )->
376            ch_dvPort3_REQ !open_7;
377       fi
378     ::ch_dvPort3_REQ ?request_7->
379       if
380       ::(opened1 && infoTaken1)->
381            infoTaken1=false
382            ch_dvPort3_RES !request_7,0;
383       ::!( (opened1 && infoTaken1)  )->
384            ch_dvPort3_REQ !request_7;
385       fi
386     ::ch_dvPort3_REQ ?close_7->
```

Liste 7.3: aeg.pml dosyası (devam).

```
387       if
388       ::(opened1 && infoTaken1)->
389            opened1=false
390            ch_dvPort3_RES !close_7,0;
391       ::!( (opened1 && infoTaken1)  )->
392            ch_dvPort3_REQ !close_7;
393       fi
394     od
395    }
396    proctype GeoServer() {
397     bool opened=false;
398     bool info2Taken=false;
399     bool opened2=false;
400     bool info1Taken=false;
401     bool opened3=false;
402     int result;
403     int responseWaiting= false;
404
405     do
406     ::!opened2  && responseWaiting==false ->
407       responseWaiting=true ;
408       ch_gsPort3_REQ !open_8;
409     ::ch_gsPort3_RES ?open_8,result->
410       if
411       ::(true)->opened2=true;
412       ::!( true )->assert(false);
413       fi ;
414       responseWaiting=false ;
415     ::opened2 && responseWaiting==false ->
416       responseWaiting=true ;
417       ch_gsPort3_REQ !close_8;
418     ::ch_gsPort3_RES ?close_8,result->
419       if
420       ::(true)->opened2=false;
421       ::!( true )->assert(false);
422       fi ;
423       responseWaiting=false ;
424     ::opened2 && responseWaiting==false ->
425       responseWaiting=true ;
426       ch_gsPort3_REQ !request_8;
427     ::ch_gsPort3_RES ?request_8,result->
428       if
429       ::(true)->info2Taken = true;
430       ::!( true )->assert(false);
431       fi ;
432       responseWaiting=false ;
433     ::opened && responseWaiting==false ->
434       responseWaiting=true ;
435       ch_gsPort2_REQ !request_5;
436     ::ch_gsPort2_RES ?request_5,result->
437       if
438       ::(opened)->info1Taken=true;
439       ::!( opened )->assert(false);
440       fi ;
441       responseWaiting=false ;
442     ::opened && responseWaiting==false ->
443       responseWaiting=true ;
444       ch_gsPort2_REQ !close_5;
445     ::ch_gsPort2_RES ?close_5,result->
446       if
447       ::(true)->opened=false;
448       ::!( true )->assert(false);
449       fi ;
```

Liste 7.3: aeg.pml dosyası (devam).

```
450       responseWaiting=false ;
451     ::!opened && responseWaiting==false ->
452           responseWaiting=true ;
453           ch_gsPort2_REQ !open_5;
454     ::ch_gsPort2_RES ?open_5,result->
455       if
456       ::(true)->opened=true;
457       ::!( true )->assert(false);
458       fi ;
459       responseWaiting=false ;
460     ::ch_dsPort1_REQ ?close_9->
461       if
462       ::(opened3 && info1Taken && info2Taken)->
463            opened3=false
464            ch_dsPort1_RES !close_9,0;
465       ::!( (opened3 && info1Taken && info2Taken)  )->
466            ch_dsPort1_REQ !close_9;
467       fi
468     ::ch_dsPort1_REQ ?request_9->
469       if
470       ::(opened3 && info1Taken && info2Taken)->
471            info1Taken = false; info2Taken = false
472            ch_dsPort1_RES !request_9,0;
473       ::!( (opened3 && info1Taken && info2Taken)  )->
474            ch_dsPort1_REQ !request_9;
475       fi
476     ::ch_dsPort1_REQ ?open_9->
477       if
478       ::(!opened3 && info1Taken && info2Taken)->
479            opened3=true
480            ch_dsPort1_RES !open_9,0;
481       ::!( (!opened3 && info1Taken && info2Taken)  )->
482            ch_dsPort1_REQ !open_9;
483       fi
484     od
485    }
486    proctype DoctrineReasoning() {
487     bool opened1=false;
488     bool opened2=false;
489     int result;
490     int responseWaiting= false;
491
492     do
493     ::!opened2 && responseWaiting==false ->
494       responseWaiting=true ;
495       ch_dsPort1_REQ !open_9;
496     ::ch_dsPort1_RES ?open_9,result->
497       if
498       ::(true)->opened2=true;
499       ::!( true )->assert(false);
500       fi ;
501       responseWaiting=false ;
502     ::opened2 && responseWaiting==false ->
503       responseWaiting=true ;
504       ch_dsPort1_REQ !request_9;
505     ::ch_dsPort1_RES ?request_9,result->
506       if
507       ::(true)->;
508       ::!( true )->assert(false);
509       fi ;
510       responseWaiting=false ;
511     ::opened2 && responseWaiting==false ->
512       responseWaiting=true ;
```

Liste 7.3: aeg.pml dosyası (devam).

```
513       ch_dsPort1_REQ !close_9;
514     ::ch_dsPort1_RES ?close_9,result->
515       if
516       ::(true)->opened2=false;
517       ::!( true )->assert(false);
518       fi ;
519       responseWaiting=false ;
520     ::opened1 && responseWaiting==false ->
521       responseWaiting=true ;
522       ch_dsPort2_REQ !close_6;
523     ::ch_dsPort2_RES ?close_6,result->
524       if
525       ::(true)->opened1=false;
526       ::!( true )->assert(false);
527       fi ;
528       responseWaiting=false ;
529     ::opened1 && responseWaiting==false ->
530       responseWaiting=true ;
531       ch_dsPort2_REQ !request_6;
532     ::ch_dsPort2_RES ?request_6,result->
533       if
534       ::(true)->;
535       ::!( true )->assert(false);
536       fi ;
537       responseWaiting=false ;
538     ::!opened1 && responseWaiting==false ->
539       responseWaiting=true ;
540       ch_dsPort2_REQ !open_6;
541     ::ch_dsPort2_RES ?open_6,result->
542       if
543       ::(true)->opened1=true;
544       ::!( true )->assert(false);
545       fi ;
546       responseWaiting=false ;
547     od
548    }
549    active proctype configuration()
550    {
551      atomic {
552            run ExperimentController();
553            run DoctrineAuthoring();
554            run DoctrineValidation();
555            run TrackServer();
556            run GeoServer();
557            run DoctrineReasoning();
558     }
559    }
560
```

Aegis muhabere sistem modelinin, MetaEdit+ editörü dönüştürücüsünden elde edilen java dosyası (Configuration.java) Liste 7.4'de gösterilmektedir.

Liste 7.4: Configuration.java dosyası.

```
1   public class Configuration {
2       ExperimentController experimentcontrollerIns;
3       DoctrineAuthoring doctrineauthoringIns;
4       DoctrineValidation doctrinevalidationIns;
5       TrackServer trackserverIns;
6       GeoServer geoserverIns;
7       DoctrineReasoning doctrinereasoningIns;
8
9       DoctrineAuthoringExperimentController_daPort1_Adapter
          doctrineauthoringexperimentcontroller_daPort1_AdapterIns;
10
11      DoctrineValidationExperimentController_dvPort1_Adapter
          doctrinevalidationexperimentcontroller_dvPort1_AdapterIns;
12
13      TrackServerExperimentController_tsPort1_Adapter
          trackserverexperimentcontroller_tsPort1_AdapterIns;
14
15      DoctrineValidationDoctrineAuthoring_dvPort2_Adapter
          doctrinevalidationdoctrineauthoring_dvPort2_AdapterIns;
16
17      DoctrineReasoningGeoServer_dsPort1_Adapter
          doctrinereasoninggeoserver_dsPort1_AdapterIns;
18
19      DoctrineReasoningDoctrineAuthoring_dsPort2_Adapter
          doctrinereasoningdoctrineauthoring_dsPort2_AdapterIns;
20
21      GeoServerTrackServer_gsPort3_Adapter geoservertrackserver_gsPort3_AdapterIns;
22
23      DoctrineValidationTrackServer_dvPort3_Adapter
          doctrinevalidationtrackserver_dvPort3_AdapterIns;
24
25      GeoServerDoctrineAuthoring_gsPort2_Adapter
           geoserverdoctrineauthoring_gsPort2_AdapterIns;
26
27      public Configuration() {
28        experimentcontrollerIns = new ExperimentController();
29        doctrineauthoringIns = new DoctrineAuthoring();
30        doctrinevalidationIns = new DoctrineValidation();
31        trackserverIns = new TrackServer();
32        geoserverIns = new GeoServer();
33        doctrinereasoningIns = new DoctrineReasoning();
34        try {
35          doctrineauthoringexperimentcontroller_daPort1_AdapterIns = new
36           DoctrineAuthoringExperimentController_daPort1_Adapter(experimentcontrollerIns);
37          doctrineauthoringIns.doctrineauthoringexperimentcontroller_daPort1_InterfaceIns
38            =  doctrineauthoringexperimentcontroller_daPort1_AdapterIns;
39
40         } catch (Exception e) {
41                    e.printStackTrace();
42         }
43         try {
44          doctrinevalidationexperimentcontroller_dvPort1_AdapterIns = new
45          DoctrineValidationExperimentController_dvPort1_Adapter(experimentcontrollerIns);
46         doctrinevalidationIns.doctrinevalidationexperimentcontroller_dvPort1_InterfaceIns
47           = doctrinevalidationexperimentcontroller_dvPort1_AdapterIns;
48
49         } catch (Exception e) {
50                    e.printStackTrace();
51         }
52         try {
53           trackserverexperimentcontroller_tsPort1_AdapterIns = new
54             TrackServerExperimentController_tsPort1_Adapter(experimentcontrollerIns);
55           trackserverIns.trackserverexperimentcontroller_tsPort1_InterfaceIns =
56             trackserverexperimentcontroller_tsPort1_AdapterIns;
```

Liste 7.4: Configuration.java dosyası(devam).

```
57
58          } catch (Exception e) {
59                    e.printStackTrace();
60          }
61          try {
62           doctrinevalidationdoctrineauthoring_dvPort2_AdapterIns = new
63            DoctrineValidationDoctrineAuthoring_dvPort2_Adapter(doctrineauthoringIns);
64           doctrinevalidationIns.doctrinevalidationdoctrineauthoring_dvPort2_Inter
65               = doctrinevalidationdoctrineauthoring_dvPort2_AdapterIns;
66
67           } catch (Exception e) {
68                    e.printStackTrace();
69           }
70           try {
71             doctrinereasoninggeoserver_dsPort1_AdapterIns =
72               new DoctrineReasoningGeoServer_dsPort1_Adapter(geoserverIns);
73             doctrinereasoningIns.doctrinereasoninggeoserver_dsPort1_InterfaceIns =
74                  doctrinereasoninggeoserver_dsPort1_AdapterIns;
75
76            } catch (Exception e) {
77                    e.printStackTrace();
78            }
79            try {
80              doctrinereasoningdoctrineauthoring_dsPort2_AdapterIns = new
81                DoctrineReasoningDoctrineAuthoring_dsPort2_Adapter(doctrineauthoringIns);
82             doctrinereasoningIns.doctrinereasoningdoctrineauthoring_dsPort2_InterfaceIns
83               = doctrinereasoningdoctrineauthoring_dsPort2_AdapterIns;
84
85            } catch (Exception e) {
86                    e.printStackTrace();
87            }
88            try {
89               geoservertrackserver_gsPort3_AdapterIns =
90                 new GeoServerTrackServer_gsPort3_Adapter(trackserverIns);
91               geoserverIns.geoservertrackserver_gsPort3_InterfaceIns =
92                 geoservertrackserver_gsPort3_AdapterIns;
93
94             } catch (Exception e) {
95                    e.printStackTrace();
96            }
97            try {
98               doctrinevalidationtrackserver_dvPort3_AdapterIns = new
99                     DoctrineValidationTrackServer_dvPort3_Adapter(trackserverIns);
100            doctrinevalidationIns.doctrinevalidationtrackserver_dvPort3_InterfaceIns
101                = doctrinevalidationtrackserver_dvPort3_AdapterIns;
102
103           } catch (Exception e) {
104                   e.printStackTrace();
105           }
106           try {
107              geoserverdoctrineauthoring_gsPort2_AdapterIns =
108                new GeoServerDoctrineAuthoring_gsPort2_Adapter(doctrineauthoringIns);
109              geoserverIns.geoserverdoctrineauthoring_gsPort2_InterfaceIns =
110               geoserverdoctrineauthoring_gsPort2_AdapterIns;
111
112           } catch (Exception e) {
113                   e.printStackTrace();
114           }
115            doctrineauthoringIns.start();
116            doctrinevalidationIns.start();
117            trackserverIns.start();
118            geoserverIns.start();
119            doctrinereasoningIns.start();
120    }
```

Liste 7.4: Configuration.java dosyası (devam).

```
121
122     public static void main(String args[]) throws Exception {
123             new Configuration();
124     }
125   }
126
127   interface DoctrineAuthoringExperimentController_daPort1_Interface {
128     public void close_1() throws Exception;
129     public void request_1() throws Exception;
130     public void open_1() throws Exception;
131   }
132
133   interface DoctrineValidationExperimentController_dvPort1_Interface {
134     public void open_2() throws Exception;
135     public void request_2() throws Exception;
136     public void close_2() throws Exception;
137   }
138
139   interface TrackServerExperimentController_tsPort1_Interface {
140     public void close_3() throws Exception;
141     public void request_3() throws Exception;
142     public void open_3() throws Exception;
143   }
144
145   interface DoctrineValidationDoctrineAuthoring_dvPort2_Interface {
146     public void open_4() throws Exception;
147     public void close_4() throws Exception;
148     public void request_4() throws Exception;
149   }
150
151   interface DoctrineReasoningGeoServer_dsPort1_Interface {
152     public void open_9() throws Exception;
153     public void request_9() throws Exception;
154     public void close_9() throws Exception;
155   }
156
157   interface DoctrineReasoningDoctrineAuthoring_dsPort2_Interface {
158     public void close_6() throws Exception;
159      public void request_6() throws Exception;
160     public void open_6() throws Exception;
161   }
162
163   interface GeoServerTrackServer_gsPort3_Interface {
164     public void open_8() throws Exception;
165     public void close_8() throws Exception;
166     public void request_8() throws Exception;
167   }
168
169   interface DoctrineValidationTrackServer_dvPort3_Interface {
170     public void open_7() throws Exception;
171     public void request_7() throws Exception;
172     public void close_7() throws Exception;
173   }
174
175   interface GeoServerDoctrineAuthoring_gsPort2_Interface {
176     public void request_5() throws Exception;
177     public void close_5() throws Exception;
178     public void open_5() throws Exception;
179   }
180
181   class DoctrineAuthoringExperimentController_daPort1_Adapter
182     implements DoctrineAuthoringExperimentController_daPort1_Interface {
183      ExperimentController experimentcontroller_ins;
184
```

Liste 7.4: Configuration.java dosyası (devam).

```
185     public DoctrineAuthoringExperimentController_daPort1_Adapter(
           ExperimentController experimentcontroller_arg)
186        throws Exception {
187            experimentcontroller_ins = experimentcontroller_arg;
188     }
189
190     public void close_1() {
191            try {
192                   experimentcontroller_ins.close_1();
193            } catch (Exception e) {
194                   e.printStackTrace();
195            }
196     }
197
198     public void request_1() {
199            try {
200                   experimentcontroller_ins.request_1();
201            } catch (Exception e) {
202                   e.printStackTrace();
203            }
204     }
205
206     public void open_1() {
207            try {
208                   experimentcontroller_ins.open_1();
209            } catch (Exception e) {
210                   e.printStackTrace();
211            }
212     }
213  }
214
215  class DoctrineValidationExperimentController_dvPort1_Adapter
216            implements DoctrineValidationExperimentController_dvPort1_Interface {
217     ExperimentController experimentcontroller_ins;
218
219     public DoctrineValidationExperimentController_dvPort1_Adapter(
           ExperimentController experimentcontroller_arg)
220        throws Exception {
221            experimentcontroller_ins = experimentcontroller_arg;
222     }
223
224     public void open_2() {
225            try {
226                   experimentcontroller_ins.open_2();
227            } catch (Exception e) {
228                   e.printStackTrace();
229            }
230     }
231
232     public void request_2() {
233            try {
234                   experimentcontroller_ins.request_2();
235            } catch (Exception e) {
236                   e.printStackTrace();
237            }
238     }
239
240     public void close_2() {
241            try {
242                   experimentcontroller_ins.close_2();
243            } catch (Exception e) {
244                   e.printStackTrace();
245            }
246     }
```

Liste 7.4: Configuration.java dosyası (devam).

```
247 }
248
249 class TrackServerExperimentController_tsPort1_Adapter implements
       TrackServerExperimentController_tsPort1_Interface {
250   ExperimentController experimentcontroller_ins;
251
252   public TrackServerExperimentController_tsPort1_Adapter(
          ExperimentController experimentcontroller_arg)
253       throws Exception {
254           experimentcontroller_ins = experimentcontroller_arg;
255   }
256
257   public void close_3() {
258           try {
259                   experimentcontroller_ins.close_3();
260           } catch (Exception e) {
261                   e.printStackTrace();
262           }
263   }
264
265   public void request_3() {
266           try {
267                   experimentcontroller_ins.request_3();
268           } catch (Exception e) {
269                   e.printStackTrace();
270           }
271   }
272
273   public void open_3() {
274           try {
275                   experimentcontroller_ins.open_3();
276           } catch (Exception e) {
277                   e.printStackTrace();
278           }
279   }
280 }
281
282 class DoctrineValidationDoctrineAuthoring_dvPort2_Adapter
283           implements DoctrineValidationDoctrineAuthoring_dvPort2_Interface {
284   DoctrineAuthoring doctrineauthoring_ins;
285
286   public DoctrineValidationDoctrineAuthoring_dvPort2_Adapter(
         DoctrineAuthoring doctrineauthoring_arg)
287       throws Exception {
288           doctrineauthoring_ins = doctrineauthoring_arg;
289   }
290
291   public void open_4() {
292           try {
293                   doctrineauthoring_ins.open_4();
294           } catch (Exception e) {
295                   e.printStackTrace();
296           }
297   }
298
299   public void close_4() {
300           try {
301                   doctrineauthoring_ins.close_4();
302           } catch (Exception e) {
303                   e.printStackTrace();
304           }
305   }
306
307   public void request_4() {
```

Liste 7.4: Configuration.java dosyası (devam).

```
308             try {
309                     doctrineauthoring_ins.request_4();
310             } catch (Exception e) {
311                     e.printStackTrace();
312             }
313     }
314 }
315
316 class DoctrineReasoningGeoServer_dsPort1_Adapter implements
      DoctrineReasoningGeoServer_dsPort1_Interface {
317     GeoServer geoserver_ins;
318
319     public DoctrineReasoningGeoServer_dsPort1_Adapter(
         GeoServer geoserver_arg) throws Exception {
320             geoserver_ins = geoserver_arg;
321     }
322
323     public void open_9() {
324             try {
325                     geoserver_ins.open_9();
326             } catch (Exception e) {
327                     e.printStackTrace();
328             }
329     }
330
331     public void request_9() {
332             try {
333                     geoserver_ins.request_9();
334             } catch (Exception e) {
335                     e.printStackTrace();
336             }
337     }
338
339     public void close_9() {
340             try {
341                     geoserver_ins.close_9();
342             } catch (Exception e) {
343                     e.printStackTrace();
344             }
345     }
346 }
347
348 class DoctrineReasoningDoctrineAuthoring_dsPort2_Adapter
349             implements DoctrineReasoningDoctrineAuthoring_dsPort2_Interface {
350     DoctrineAuthoring doctrineauthoring_ins;
351
352     public DoctrineReasoningDoctrineAuthoring_dsPort2_Adapter(
        DoctrineAuthoring doctrineauthoring_arg)
353       throws Exception {
354             doctrineauthoring_ins = doctrineauthoring_arg;
355     }
356
357     public void close_6() {
358             try {
359                     doctrineauthoring_ins.close_6();
360             } catch (Exception e) {
361                     e.printStackTrace();
362             }
363     }
364
365     public void request_6() {
366             try {
367                     doctrineauthoring_ins.request_6();
368             } catch (Exception e) {
```

Liste 7.4: Configuration.java dosyası (devam).

```
369                     e.printStackTrace();
370             }
371     }
372
373     public void open_6() {
374             try {
375                     doctrineauthoring_ins.open_6();
376             } catch (Exception e) {
377                     e.printStackTrace();
378             }
379     }
380 }
381
382 class GeoServerTrackServer_gsPort3_Adapter implements
        GeoServerTrackServer_gsPort3_Interface {
383     TrackServer trackserver_ins;
384
385     public GeoServerTrackServer_gsPort3_Adapter(TrackServer trackserver_arg)
        throws Exception {
386             trackserver_ins = trackserver_arg;
387     }
388
389     public void open_8() {
390             try {
391                     trackserver_ins.open_8();
392             } catch (Exception e) {
393                     e.printStackTrace();
394             }
395     }
396
397     public void close_8() {
398             try {
399                     trackserver_ins.close_8();
400             } catch (Exception e) {
401                     e.printStackTrace();
402             }
403     }
404
405     public void request_8() {
406             try {
407                     trackserver_ins.request_8();
408             } catch (Exception e) {
409                     e.printStackTrace();
410             }
411     }
412 }
413
414 class DoctrineValidationTrackServer_dvPort3_Adapter implements
        DoctrineValidationTrackServer_dvPort3_Interface {
415     TrackServer trackserver_ins;
416
417     public DoctrineValidationTrackServer_dvPort3_Adapter(TrackServer trackserver_arg)
        throws Exception {
418             trackserver_ins = trackserver_arg;
419     }
420
421     public void open_7() {
422             try {
423                     trackserver_ins.open_7();
424             } catch (Exception e) {
425                     e.printStackTrace();
426             }
427     }
428
```

Liste 7.4: Configuration.java dosyası (devam).

```
429     public void request_7() {
430             try {
431                     trackserver_ins.request_7();
432             } catch (Exception e) {
433                     e.printStackTrace();
434             }
435     }
436
437     public void close_7() {
438             try {
439                     trackserver_ins.close_7();
440             } catch (Exception e) {
441                     e.printStackTrace();
442             }
443     }
444 }
445
446 class GeoServerDoctrineAuthoring_gsPort2_Adapter implements
        GeoServerDoctrineAuthoring_gsPort2_Interface {
447     DoctrineAuthoring doctrineauthoring_ins;
448
449     public GeoServerDoctrineAuthoring_gsPort2_Adapter(DoctrineAuthoring
                                                        doctrineauthoring_arg)
          throws Exception {
450             doctrineauthoring_ins = doctrineauthoring_arg;
451     }
452
453     public void request_5() {
454             try {
455                     doctrineauthoring_ins.request_5();
456             } catch (Exception e) {
457                     e.printStackTrace();
458             }
459     }
460
461     public void close_5() {
462             try {
463                     doctrineauthoring_ins.close_5();
464             } catch (Exception e) {
465                     e.printStackTrace();
466             }
467     }
468
469     public void open_5() {
470             try {
471                     doctrineauthoring_ins.open_5();
472             } catch (Exception e) {
473                     e.printStackTrace();
474             }
475     }
476 }
477
478 class ExperimentController {
479     boolean opened3 = false;
480     boolean opened2 = false;
481     boolean opened1 = false;
482     private int result = 0;
483
484     public int open_1() throws Exception {
485             if (!opened1) {
486                     opened1 = true;
487                     result = 0;
488                     result = 0;
489                     result = 0;
```

Liste 7.4: Configuration.java dosyası (devam).

```
490             }
491             return result;
492     }
493
494     public int close_1() throws Exception {
495             if (opened1) {
496                     opened1 = false;
497                     result = 0;
498                     result = 0;
499                     result = 0;
500             }
501             return result;
502     }
503
504     public int request_1() throws Exception {
505             if (opened1) {
506                     ;
507                     result = 0;
508                     result = 0;
509                     result = 0;
510             }
511             return result;
512     }
513
514     public int request_2() throws Exception {
515             if (opened2) {
516                     ;
517                     result = 0;
518                     result = 0;
519                     result = 0;
520             }
521             return result;
522     }
523
524     public int close_2() throws Exception {
525             if (opened2) {
526                     opened2 = false;
527                     result = 0;
528                     result = 0;
529                     result = 0;
530             }
531             return result;
532     }
533
534     public int open_2() throws Exception {
535             if (!opened2) {
536                     opened2 = true;
537                     result = 0;
538                     result = 0;
539                     result = 0;
540             }
541             return result;
542     }
543
544     public int request_3() throws Exception {
545             if (opened3) {
546                     ;
547                     result = 0;
548                     result = 0;
549                     result = 0;
550             }
551             return result;
552     }
553
```

Liste 7.4: Configuration.java dosyası (devam).

```
554     public int close_3() throws Exception {
555             if (opened3) {
556                     opened3 = false;
557                     result = 0;
558                     result = 0;
559                     result = 0;
560             }
561             return result;
562     }
563
564     public int open_3() throws Exception {
565             if (!opened3) {
566                     opened3 = true;
567                     result = 0;
568                     result = 0;
569                     result = 0;
570             }
571             return result;
572     }
573 }
574
575 class DoctrineAuthoring extends Thread {
576     boolean opened1 = false;
577     boolean opened3 = false;
578     boolean infoTaken1 = false;
579     boolean infoTaken3 = false;
580     boolean opened4 = false;
581     boolean infoTaken2 = false;
582     boolean opened2 = false;
583     private int result = 0;
584     DoctrineAuthoringExperimentController_daPort1_Interface
          doctrineauthoringexperimentcontroller_daPort1_InterfaceIns;
585     private boolean responseWaiting = false;
586
587     public void close_1() throws Exception {
588             if (opened1 && !responseWaiting) {
589                     responseWaiting = true;
590                     doctrineauthoringexperimentcontroller_daPort1_InterfaceIns.close_1();
591             }
592             if (true) {
593                     opened1 = false;
594             }
595             responseWaiting = false;
596     }
597
598     public void request_1() throws Exception {
599             if (opened1 && !responseWaiting) {
600                     responseWaiting = true;
601
      doctrineauthoringexperimentcontroller_daPort1_InterfaceIns.request_1();
602             }
603             if (true) {
604                     infoTaken1 = true;
605                     infoTaken2 = true;
606                     infoTaken3 = true;
607                     ;
608             }
609             responseWaiting = false;
610     }
611
612     public void open_1() throws Exception {
613             if (!opened1 && !responseWaiting) {
614                     responseWaiting = true;
```

Liste 7.4: Configuration.java dosyası (devam).

```
615                     doctrineauthoringexperimentcontroller_daPort1_InterfaceIns.open_1();
616             }
617             if (true) {
618                     opened1 = true;
619             }
620             responseWaiting = false;
621     }
622
623
624     public int close_4() throws Exception {
625             if (opened2 && infoTaken1) {
626                     opened2 = false;
627                     result = 0;
628                     result = 0;
629                     result = 0;
630             }
631             return result;
632     }
633
634     public int request_4() throws Exception {
635             if (opened2 && infoTaken1) {
636                     infoTaken1 = false;
637                     result = 0;
638                     result = 0;
639                     result = 0;
640             }
641             return result;
642     }
643
644     public int open_4() throws Exception {
645             if (!opened2 && infoTaken1) {
646                     opened2 = true;
647                     result = 0;
648                     result = 0;
649                     result = 0;
650             }
651             return result;
652     }
653
654     public int request_5() throws Exception {
655             if (opened3 && infoTaken2) {
656                     infoTaken2 = false;
657                     result = 0;
658                     result = 0;
659                     result = 0;
660             }
661             return result;
662     }
663
664     public int close_5() throws Exception {
665             if (opened3 && infoTaken2) {
666                     opened3 = false;
667                     result = 0;
668                     result = 0;
669                     result = 0;
670             }
671             return result;
672     }
673
674     public int open_5() throws Exception {
675             if (!opened3 && infoTaken2) {
676                     opened3 = true;
677                     result = 0;
678                     result = 0;
```

Liste 7.4: Configuration.java dosyası (devam).

```
679                     result = 0;
680             }
681             return result;
682     }
683
684     public int open_6() throws Exception {
685             if (!opened4 && infoTaken3) {
686                     opened4 = true;
687                     result = 0;
688                     result = 0;
689                     result = 0;
690             }
691             return result;
692     }
693
694     public int close_6() throws Exception {
695             if (opened4 && infoTaken3) {
696                     opened4 = false;
697                     result = 0;
698                     result = 0;
699                     result = 0;
700             }
701             return result;
702     }
703
704     public int request_6() throws Exception {
705             if (opened4 && infoTaken3) {
706                     infoTaken3 = false;
707                     result = 0;
708                     result = 0;
709                     result = 0;
710             }
711             return result;
712     }
713
714     public void run() {
715             while (true) {
716                     try {
717                             close_1();
718                             request_1();
719                             open_1();
720                     } catch (Exception e) {
721                             e.printStackTrace();
722                     }
723             }
724     }
725 }
726
727 class DoctrineValidation extends Thread {
728     boolean opened = false;
729     boolean opened2 = false;
730     boolean opened3 = false;
731     DoctrineValidationExperimentController_dvPort1_Interface
732       doctrinevalidationexperimentcontroller_dvPort1_InterfaceIns;
733
734     DoctrineValidationDoctrineAuthoring_dvPort2_Interface
735       doctrinevalidationdoctrineauthoring_dvPort2_InterfaceIns;
736
737     DoctrineValidationTrackServer_dvPort3_Interface
738       doctrinevalidationtrackserver_dvPort3_InterfaceIns;
739
740     private boolean responseWaiting = false;
741
742     public void open_2() throws Exception {
```

Liste 7.4: Configuration.java dosyası (devam).

```
743             if (!opened && !responseWaiting) {
744                     responseWaiting = true;
745                     doctrinevalidationexperimentcontroller_dvPort1_InterfaceIns.open_2();
746             }
747             if (true) {
748                     opened = true;
749             }
750             responseWaiting = false;
751     }
752
753     public void request_2() throws Exception {
754             if (opened && !responseWaiting) {
755                 responseWaiting = true;
756                 doctrinevalidationexperimentcontroller_dvPort1_InterfaceIns.request_2();
757             }
758             if (true) {
759
760             }
761             responseWaiting = false;
762     }
763
764     public void close_2() throws Exception {
765             if (opened && !responseWaiting) {
766                     responseWaiting = true;
767
        doctrinevalidationexperimentcontroller_dvPort1_InterfaceIns.close_2();
768             }
769             if (true) {
770                     opened = false;
771             }
772             responseWaiting = false;
773     }
774
775     public void open_4() throws Exception {
776             if (!opened2 && !responseWaiting) {
777                     responseWaiting = true;
778                     doctrinevalidationdoctrineauthoring_dvPort2_InterfaceIns.open_4();
779             }
780             if (true) {
781                     opened2 = true;
782             }
783             responseWaiting = false;
784     }
785
786     public void close_4() throws Exception {
787             if (opened2 && !responseWaiting) {
788                     responseWaiting = true;
789                     doctrinevalidationdoctrineauthoring_dvPort2_InterfaceIns.close_4();
790             }
791             if (true) {
792                     opened2 = false;
793             }
794             responseWaiting = false;
795     }
796
797     public void request_4() throws Exception {
798             if (opened && !responseWaiting) {
799                     responseWaiting = true;
800                     doctrinevalidationdoctrineauthoring_dvPort2_InterfaceIns.request_4();
801             }
802             if (true) {
803
804             }
805             responseWaiting = false;
```

Liste 7.4: Configuration.java dosyası (devam).

```
806     }
807
808     public void open_7() throws Exception {
809             if (!opened3 && !responseWaiting) {
810                     responseWaiting = true;
811                     doctrinevalidationtrackserver_dvPort3_InterfaceIns.open_7();
812             }
813             if (true) {
814                     opened3 = true;
815             }
816             responseWaiting = false;
817     }
818
819     public void request_7() throws Exception {
820             if (opened3 && !responseWaiting) {
821                     responseWaiting = true;
822                     doctrinevalidationtrackserver_dvPort3_InterfaceIns.request_7();
823             }
824             if (true) {
825
826             }
827             responseWaiting = false;
828     }
829
830     public void close_7() throws Exception {
831             if (opened3 && !responseWaiting) {
832                     responseWaiting = true;
833                     doctrinevalidationtrackserver_dvPort3_InterfaceIns.close_7();
834             }
835             if (true) {
836                     opened3 = false;
837             }
838             responseWaiting = false;
839     }
840
841
842     public void run() {
843             while (true) {
844                     try {
845                             open_2();
846                             request_2();
847                             close_2();
848                     } catch (Exception e) {
849                             e.printStackTrace();
850                     }
851                     try {
852                             open_4();
853                             close_4();
854                             request_4();
855                     } catch (Exception e) {
856                             e.printStackTrace();
857                     }
858                     try {
859                             open_7();
860                             request_7();
861                             close_7();
862                     } catch (Exception e) {
863                             e.printStackTrace();
864                     }
865             }
866     }
867 }
868
869 class TrackServer extends Thread {
```

Liste 7.4: Configuration.java dosyası (devam).

```
870     boolean opened = false;
871     boolean opened1 = false;
872     boolean opened2 = false;
873     boolean infoTaken1 = false;
874     boolean infoTaken2 = false;
875     private int result = 0;
876     TrackServerExperimentController_tsPort1_Interface
            trackserverexperimentcontroller_tsPort1_InterfaceIns;
877     private boolean responseWaiting = false;
878
879     public void close_3() throws Exception {
880             if (opened && !responseWaiting) {
881                     responseWaiting = true;
882                     trackserverexperimentcontroller_tsPort1_InterfaceIns.close_3();
883             }
884             if (true) {
885                     opened = false;
886             }
887             responseWaiting = false;
888     }
889
890     public void request_3() throws Exception {
891             if (opened && !responseWaiting) {
892                     responseWaiting = true;
893                     trackserverexperimentcontroller_tsPort1_InterfaceIns.request_3();
894             }
895             if (true) {
896                     infoTaken1 = true;
897                     infoTaken2 = true;
898             }
899             responseWaiting = false;
900     }
901
902     public void open_3() throws Exception {
903             if (!opened && !responseWaiting) {
904                     responseWaiting = true;
905                     trackserverexperimentcontroller_tsPort1_InterfaceIns.open_3();
906             }
907             if (true) {
908                     opened = true;
909             }
910             responseWaiting = false;
911     }
912
913
914     public int request_8() throws Exception {
915             if (opened2 && infoTaken2) {
916                     infoTaken2 = false;
917                     result = 0;
918                     result = 0;
919                     result = 0;
920             }
921             return result;
922     }
923
924     public int close_8() throws Exception {
925             if (opened2 && infoTaken2) {
926                     opened2 = false;
927                     result = 0;
928                     result = 0;
929                     result = 0;
930             }
931             return result;
932     }
```

Liste 7.4: Configuration.java dosyası (devam).

```
933
934     public int open_8() throws Exception {
935             if (!opened2 && infoTaken2) {
936                     opened2 = true;
937                     result = 0;
938                     result = 0;
939                     result = 0;
940             }
941             return result;
942     }
943
944     public int open_7() throws Exception {
945             if (!opened1 && infoTaken1) {
946                     opened1 = true;
947                     result = 0;
948                     result = 0;
949                     result = 0;
950             }
951             return result;
952     }
953
954     public int request_7() throws Exception {
955             if (opened1 && infoTaken1) {
956                     infoTaken1 = false;
957                     result = 0;
958                     result = 0;
959                     result = 0;
960             }
961             return result;
962     }
963
964     public int close_7() throws Exception {
965             if (opened1 && infoTaken1) {
966                     opened1 = false;
967                     result = 0;
968                     result = 0;
969                     result = 0;
970             }
971             return result;
972     }
973
974     public void run() {
975             while (true) {
976                     try {
977                             close_3();
978                             request_3();
979                             open_3();
980                     } catch (Exception e) {
981                             e.printStackTrace();
982                     }
983             }
984     }
985 }
986
987 class GeoServer extends Thread {
988     boolean opened = false;
989     boolean info2Taken = false;
990     boolean opened2 = false;
991     boolean info1Taken = false;
992     boolean opened3 = false;
993     private int result = 0;
994     GeoServerTrackServer_gsPort3_Interface geoservertrackserver_gsPort3_InterfaceIns;
995     GeoServerDoctrineAuthoring_gsPort2_Interface
          geoserverdoctrineauthoring_gsPort2_InterfaceIns;
```

Liste 7.4: Configuration.java dosyası (devam).

```
996     private boolean responseWaiting = false;
997
998     public void open_8() throws Exception {
999             if (!opened2 && !responseWaiting) {
1000                    responseWaiting = true;
1001                    geoservertrackserver_gsPort3_InterfaceIns.open_8();
1002            }
1003            if (true) {
1004                    opened2 = true;
1005            }
1006            responseWaiting = false;
1007    }
1008
1009    public void close_8() throws Exception {
1010            if (opened2 && !responseWaiting) {
1011                    responseWaiting = true;
1012                    geoservertrackserver_gsPort3_InterfaceIns.close_8();
1013            }
1014            if (true) {
1015                    opened2 = false;
1016            }
1017            responseWaiting = false;
1018    }
1019
1020    public void request_8() throws Exception {
1021            if (opened2 && !responseWaiting) {
1022                    responseWaiting = true;
1023                    geoservertrackserver_gsPort3_InterfaceIns.request_8();
1024            }
1025            if (true) {
1026                    info2Taken = true;
1027            }
1028            responseWaiting = false;
1029    }
1030
1031    public void request_5() throws Exception {
1032            if (opened && !responseWaiting) {
1033                    responseWaiting = true;
1034                    geoserverdoctrineauthoring_gsPort2_InterfaceIns.request_5();
1035            }
1036            if (opened) {
1037                    info1Taken = true;
1038            }
1039            responseWaiting = false;
1040    }
1041
1042    public void close_5() throws Exception {
1043            if (opened && !responseWaiting) {
1044                    responseWaiting = true;
1045                    geoserverdoctrineauthoring_gsPort2_InterfaceIns.close_5();
1046            }
1047            if (true) {
1048                    opened = false;
1049            }
1050            responseWaiting = false;
1051    }
1052
1053    public void open_5() throws Exception {
1054            if (!opened && !responseWaiting) {
1055                    responseWaiting = true;
1056                    geoserverdoctrineauthoring_gsPort2_InterfaceIns.open_5();
1057            }
1058            if (true) {
1059                    opened = true;
```

Liste 7.4: Configuration.java dosyası (devam).

```
1060            }
1061            responseWaiting = false;
1062    }
1063
1064    public int close_9() throws Exception {
1065            if (opened3 && info1Taken && info2Taken) {
1066                    opened3 = false;
1067                    result = 0;
1068                    result = 0;
1069                    result = 0;
1070            }
1071            return result;
1072    }
1073
1074    public int request_9() throws Exception {
1075            if (opened3 && info1Taken && info2Taken) {
1076                    info1Taken = false;
1077                    info2Taken = false;
1078                    result = 0;
1079                    result = 0;
1080                    result = 0;
1081            }
1082            return result;
1083    }
1084
1085    public int open_9() throws Exception {
1086            if (!opened3 && info1Taken && info2Taken) {
1087                    opened3 = true;
1088                    result = 0;
1089                    result = 0;
1090                    result = 0;
1091            }
1092            return result;
1093    }
1094
1095    public void run() {
1096            while (true) {
1097                    try {
1098                            open_8();
1099                            close_8();
1100                            request_8();
1101                    } catch (Exception e) {
1102                            e.printStackTrace();
1103                    }
1104                    try {
1105                            request_5();
1106                            close_5();
1107                            open_5();
1108                    } catch (Exception e) {
1109                            e.printStackTrace();
1110                    }
1111            }
1112    }
1113}
1114
1115 class DoctrineReasoning extends Thread {
1116    boolean opened1 = false;
1117    boolean opened2 = false;
1118    DoctrineReasoningGeoServer_dsPort1_Interface
1119      doctrinereasoninggeoserver_dsPort1_InterfaceIns;
1120    DoctrineReasoningDoctrineAuthoring_dsPort2_Interface
1121      doctrinereasoningdoctrineauthoring_dsPort2_InterfaceIns;
1122    private boolean responseWaiting = false;
1123
```

Liste 7.4: Configuration.java dosyası (devam).

```
1124    public void open_9() throws Exception {
1125            if (!opened2 && !responseWaiting) {
1126                    responseWaiting = true;
1127                    doctrinereasoninggeoserver_dsPort1_InterfaceIns.open_9();
1128            }
1129            if (true) {
1130                    opened2 = true;
1131            }
1132            responseWaiting = false;
1133    }
1134
1135    public void request_9() throws Exception {
1136            if (opened2 && !responseWaiting) {
1137                    responseWaiting = true;
1138                    doctrinereasoninggeoserver_dsPort1_InterfaceIns.request_9();
1139            }
1140            if (true) {
1141
1142            }
1143            responseWaiting = false;
1144    }
1145
1146    public void close_9() throws Exception {
1147            if (opened2 && !responseWaiting) {
1148                    responseWaiting = true;
1149                    doctrinereasoninggeoserver_dsPort1_InterfaceIns.close_9();
1150            }
1151            if (true) {
1152                    opened2 = false;
1153            }
1154            responseWaiting = false;
1155    }
1156
1157    public void close_6() throws Exception {
1158            if (opened1 && !responseWaiting) {
1159                    responseWaiting = true;
1160                    doctrinereasoningdoctrineauthoring_dsPort2_InterfaceIns.close_6();
1161            }
1162            if (true) {
1163                    opened1 = false;
1164            }
1165            responseWaiting = false;
1166    }
1167
1168    public void request_6() throws Exception {
1169            if (opened1 && !responseWaiting) {
1170                    responseWaiting = true;
1171                    doctrinereasoningdoctrineauthoring_dsPort2_InterfaceIns.request_6();
1172            }
1173            if (true) {
1174
1175            }
1176            responseWaiting = false;
1177    }
1178
1179    public void open_6() throws Exception {
1180            if (!opened1 && !responseWaiting) {
1181                    responseWaiting = true;
1182                    doctrinereasoningdoctrineauthoring_dsPort2_InterfaceIns.open_6();
1183            }
1184            if (true) {
1185                    opened1 = true;
1186            }
1187            responseWaiting = false;
```

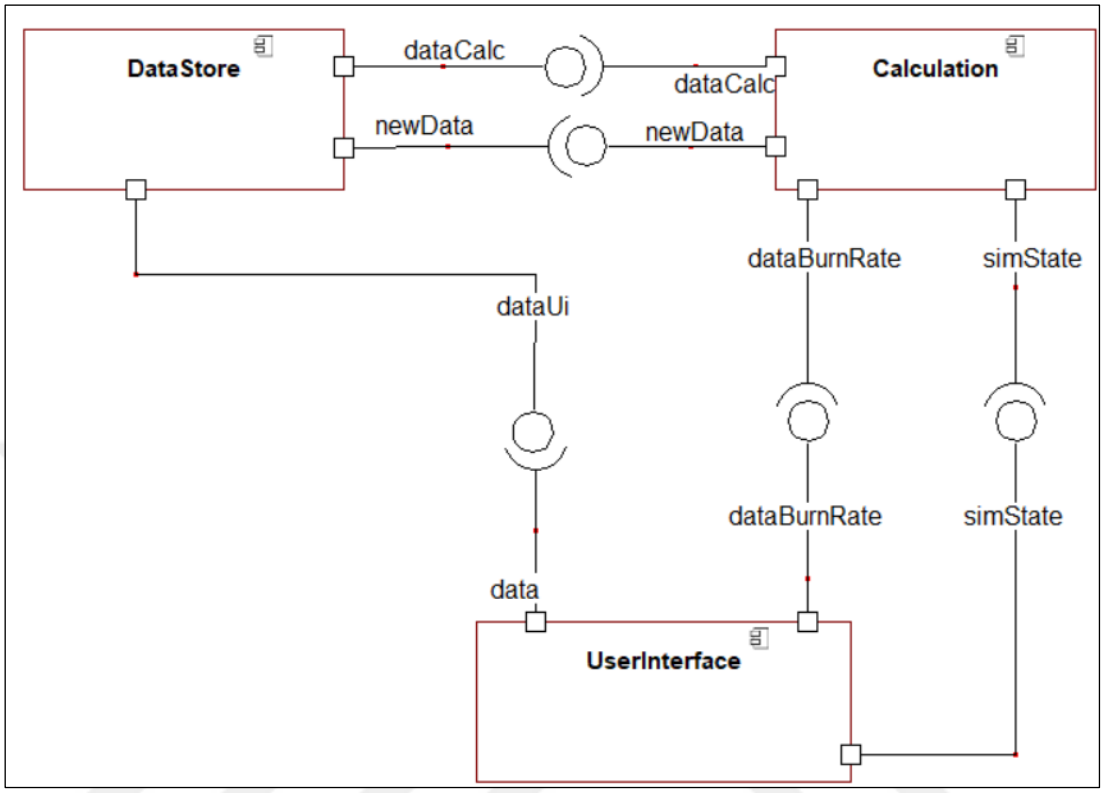
Liste 7.4: Configuration.java dosyası (devam).

```
1188    }
1189
1190    public void run() {
1191            while (true) {
1192                    try {
1193                            open_9();
1194                            request_9();
1195                            close_9();
1196                    } catch (Exception e) {
1197                            e.printStackTrace();
1198                    }
1199                    try {
1200                            close_6();
1201                            request_6();
1202                            open_6();
1203                    } catch (Exception e) {
1204                            e.printStackTrace();
1205                    }
1206            }
1207    }
1208}
1209
```

7.5 AYA İNİŞ VAKA ANALİZİ

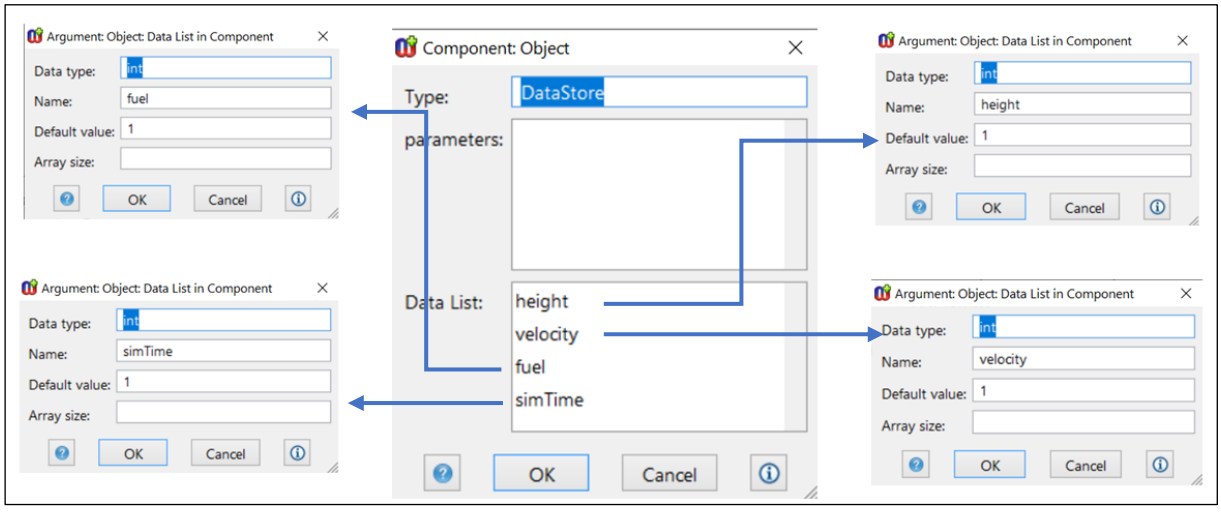
Aya iniş modeli [43] üç bileşenden oluşuyor. Bunlar sırasıyla *"UserInterface"*, *"DataStore"* *"Calculation"* isimli bileşenlerdir. *"DataStore"* bileşeni, hız (velocity), yükseklik (height), yakıt (fuel) ve simülasyon zamanı (simTime) gibi verileri muhafaza edip, *"UserInterface"* ve *"Calculation"* bileşenlerine aktarmayı sağlamaktadır. *"Calculation"* bileşeni, *"DataStore"* bileşeninden aldığı hız, yükseklik, yakıt ve simülasyon zamanını verilerini; yakıt hızı miktarına (burnRate) göre güncelleyip *"DataStore"* bileşenine aktarır. *"UserInterface"* bileşeni, yakıt hızı miktarını *"Calculation"* bileşenine iletirken, simülatörün statüsü (currentSimState) ile ilgili verisi *"Calculation"* tarafından *"UserInterface"* bileşenine gelen istek sonrası güncellenmektedir.

.

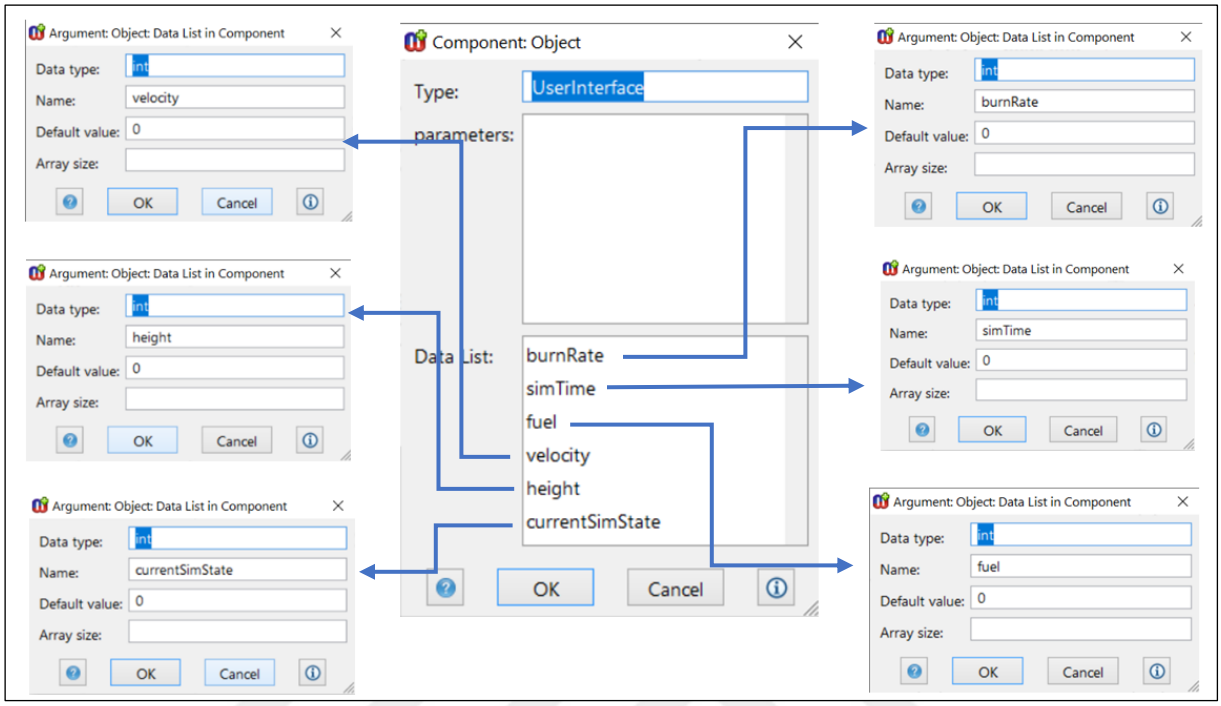


Şekil 7.48: SAwUML'de oluşturulan Aya İniş modeli.

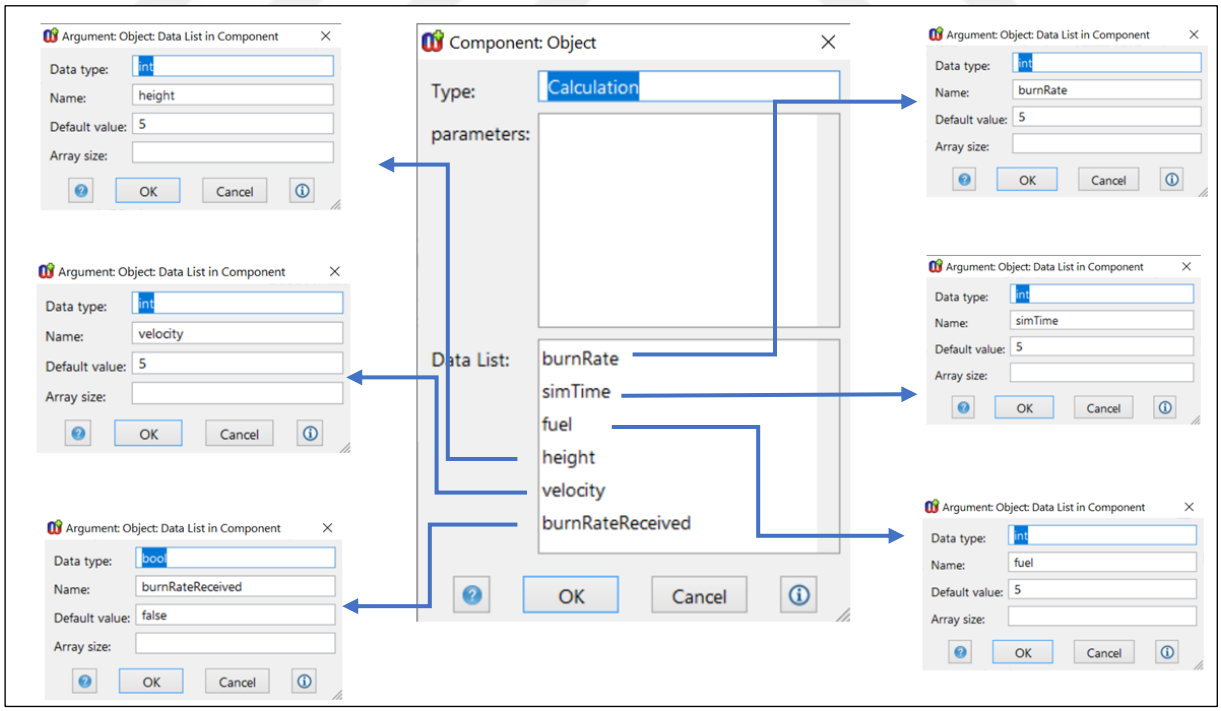
Bileşenlerinin verileri (data) ve özellikleri aşağıdaki şekillerde gösterilmiştir.



Şekil 7.49: *DataStore* bileşeninin verileri ve özellikleri.

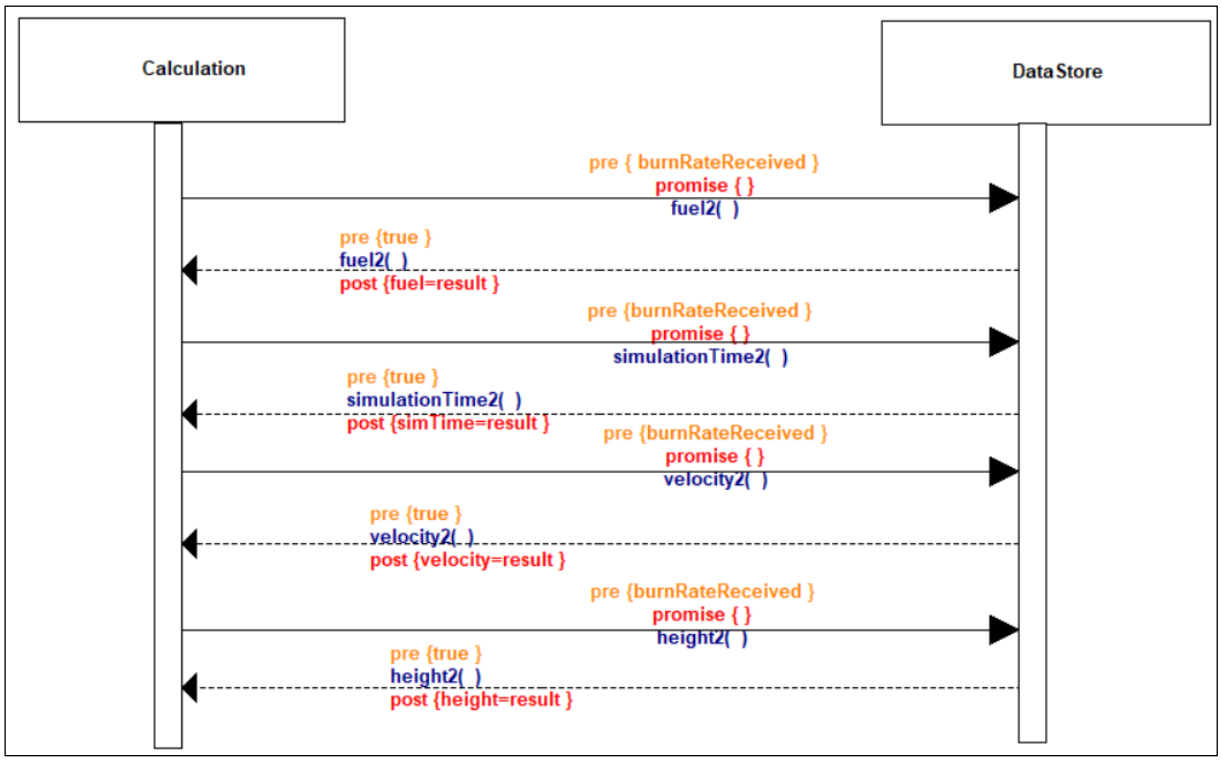


Şekil 7.50: *Calculation* bileşeninin verileri ve özellikleri.



Şekil 7.51: *UserInterface* bileşeninin verileri ve özellikleri.

Calculation'in *DataStore* ile ilişkisi, *Calculation*'ın "*dataCalc*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.52'de gösteriliyor.



Şekil 7.52: *Calculation*'ın *DataStore*'a bağlanan istemci portun (*required port*) özellikleri.

Calculation'ın *DataStore*'dan *fuel2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*burnRateReceived*) sağlaması gerekiyor. Bu ön şart sağlandığında *DataStore*'a *fuel2* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*fuel=result*) göre atama yapılır.

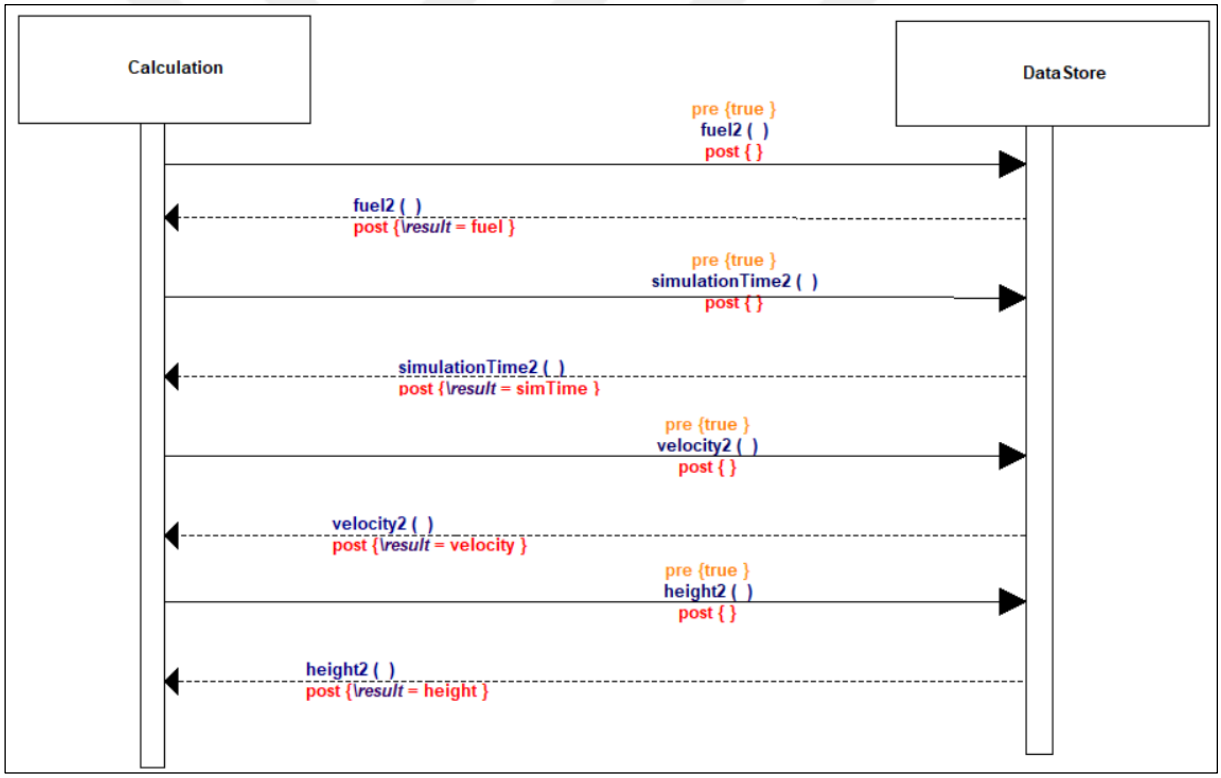
Calculation'ın *DataStore*'dan *simulationTime2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*burnRateReceived*) sağlaması gerekiyor. Bu ön şart sağlandığında *DataStore*'a *simulationTime2* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*simTime=result*) göre atama yapılır.

Calculation'ın *DataStore*'dan *velocity2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*burnRateReceived*) sağlaması gerekiyor. Bu ön şart sağlandığında *DataStore*'a *velocity2* metot

çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*velocity=result*) göre atama yapılır.

Calculation'ın *DataStore*'dan *height2* metodu ile istekte bulunabilmesi için gerekli ön şartı (*burnRateReceived*) sağlaması gerekiyor. Bu ön şart sağlandığında *DataStore*'a *height2* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*height=result*) göre atama yapılır.

DataStore'un *Calculation* ile ilişkisi, *DataStore*'un "*dataCalc*" isimli sağlayıcı portuna (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.53'de gösteriliyor.



Şekil 7.53: *DataStore*'un *Calculation*'a bağlanan sağlayıcı portun (*provide port*) özellikleri.

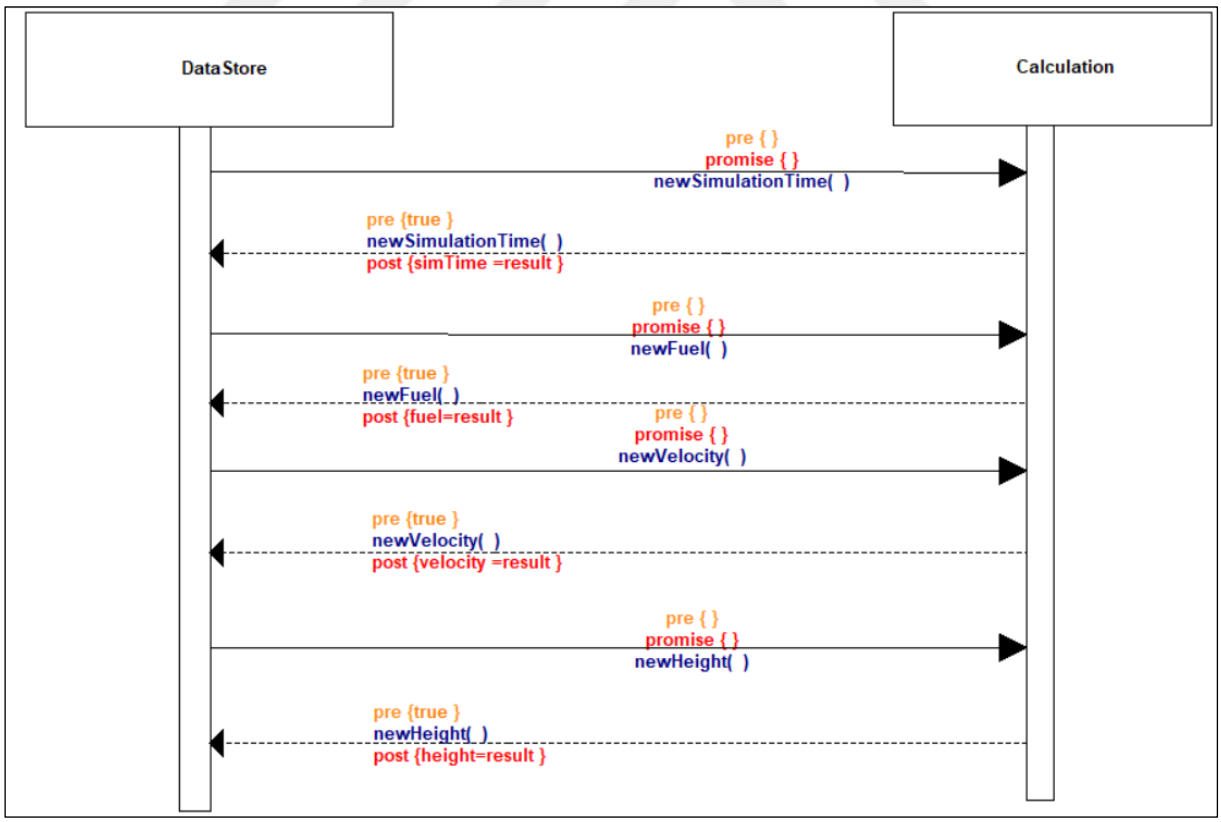
Calculation'dan *DataStore*'a *fuel2* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *Calculation*'a (kesikli ok ile gösterilen) *fuel2* metot çağrısı ile dönüş değeri (*fuel*) yollanır.

Calculation'dan *DataStore*'a *simulationTime2* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *Calculation*'a (kesikli ok ile gösterilen) *simulationTime2* metot çağrısı ile dönüş değeri (*simTime*) yollanır.

Calculation'dan *DataStore*'a *velocity2* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *Calculation*'a (kesikli ok ile gösterilen) *velocity2* metot çağrısı ile dönüş değeri (*velocity*) yollanır.

Calculation'dan *DataStore*'a *height2* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *Calculation*'a (kesikli ok ile gösterilen) *height2* metot çağrısı ile dönüş değeri (*height*) yollanır.

DataStore'un Calculation ile ilişkisi, *DataStore*'un "new*Data*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.54'de gösteriliyor.



Şekil 7.54: *DataStore*'ın *Calculation*'a bağlanan istemci portun (*required port*) özellikleri.

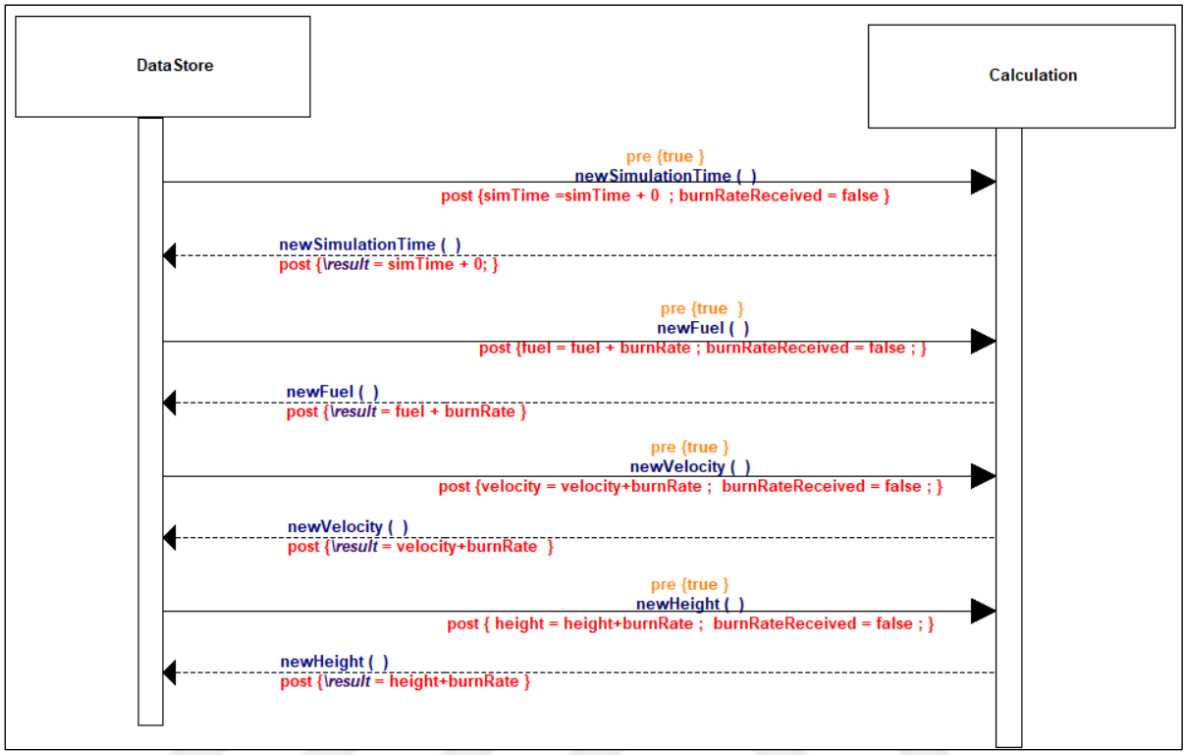
DataStore'dan ön şartsız *Calculation*'a *newSimulationTime* metot çağrısı yapılır. *Calculation*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*simTime=result*) göre atama yapılır.

DataStore'dan ön şartsız *Calculation*'a *newFuel* metot çağrısı yapılır. *Calculation*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*fuel=result*) göre atama yapılır.

DataStore'dan ön şartsız *Calculation*'a *newVelocity* metot çağrısı yapılır. *Calculation*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*velocity=result*) göre atama yapılır.

DataStore'dan ön şartsız *Calculation*'a *newHeight* metot çağrısı yapılır. *Calculation*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*height=result*) göre atama yapılır.

Calculation'un *DataStore* ile ilişkisi, *Calculation*'nın "*newData*" isimli sağlayıcı portuna (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.55'de gösteriliyor.



Şekil 7.55: *Calculation*'nın *DataStore*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

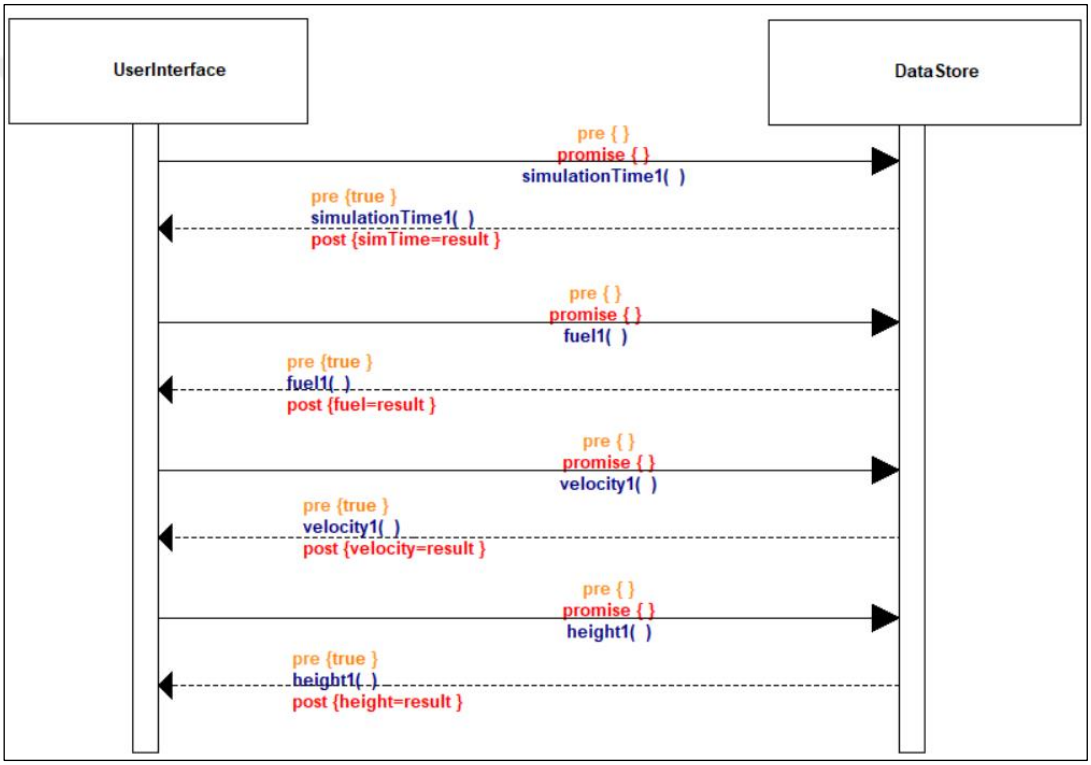
*DataStore'*dan *Calculation*'a *newSimulationTime* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. Son şarta göre (*simTime=simTime+0; burnRateReceived=false;*) atama yapılır. *DataStore*'a (kesikli ok ile gösterilen) *newSimulationTime* metot çağrısı ile dönüş değeri (*simTime + 0*) yollanır.

*DataStore'*dan *Calculation*'a *newFuel* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. Son şarta göre (*fuel=fule+burnRate; burnRateReceived=false;*) atama yapılır. *DataStore*'a (kesikli ok ile gösterilen) *newFuel* metot çağrısı ile dönüş değeri (*fule+burnRate*) yollanır.

*DataStore'*dan *Calculation*'a *newVelocity* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. Son şarta göre (*velocity=velocity+burnRate; burnRateReceived=false;*) atama yapılır. *DataStore*'a (kesikli ok ile gösterilen) *newVelocity* metot çağrısı ile dönüş değeri (*velocity+burnRate*) yollanır.

DataStore'dan *Calculation*'a *newHeight* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. Son şarta göre (*height=height+burnRate; burnRateReceived=false;*) atama yapılır. *DataStore*'a (kesikli ok ile gösterilen) *newHeight* metot çağrısı ile dönüş değeri (*height+burnRate*) yollanır.

UserInterface'in *DataStore* ile ilişkisi, *UserInterface*'in "*data*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagramda Şekil 7.56'da gösteriliyor.



Şekil 7.56: *UserInterface*'in *DataStore*'a bağlanan istemci portun (*required port*) özellikleri.

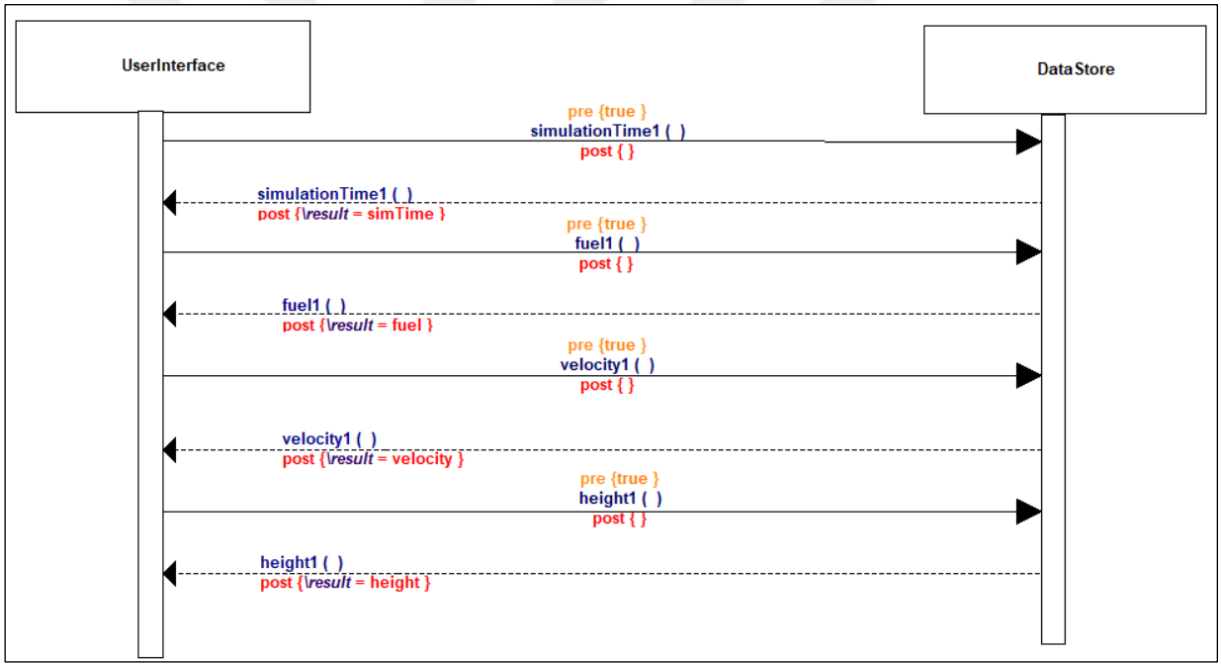
UserInterface'den ön şartsız *DataStore*'a *simulationTime1* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*simTime=result*) göre atama yapılır.

UserInterface'den ön şartsız *DataStore*'a *fuel1* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*fuel=result*) göre atama yapılır.

UserInterface'den ön şartsız *DataStore*'a *velocity1* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*velocity=result*) göre atama yapılır.

UserInterface'den ön şartsız *DataStore*'a *height1* metot çağrısı yapılır. *DataStore*'dan alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*height=result*) göre atama yapılır.

Datastore'un *UserInterface* ile ilişkisi, *DataStore*'un "*dataUi*" isimli sağlayıcı portuna (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.57'de gösteriliyor.



Şekil 7.57: *DataStore*'un *UserInterface*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

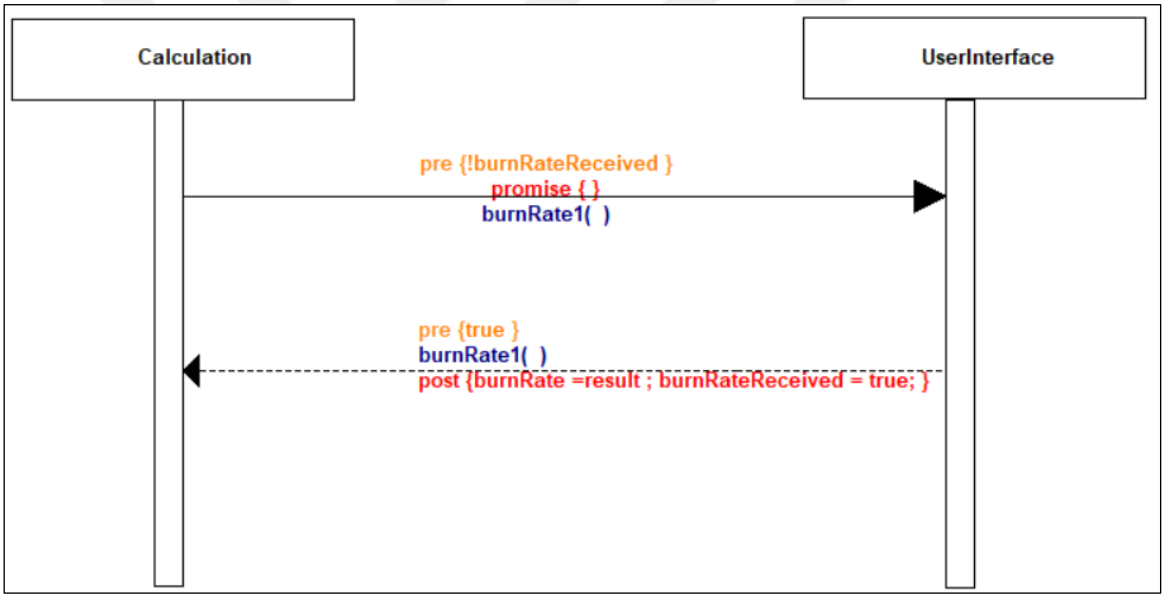
*UserInterface'*den *DataStore*'a *simulationTime1* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *UserInterface*'e (kesikli ok ile gösterilen) *simulationTime* metot çağrısı ile dönüş değeri (*simTime*) yollanır.

*UserInterface'*den *DataStore*'a *fuel1* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *UserInterface*'e (kesikli ok ile gösterilen) *fuel1* metot çağrısı ile dönüş değeri (*fuel*) yollanır.

*UserInterface'*den *DataStore*'a *velocity1* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *UserInterface*'e (kesikli ok ile gösterilen) *velocity1* metot çağrısı ile dönüş değeri (*velocity*) yollanır.

*UserInterface'*den *DataStore*'a *height1* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *UserInterface*'e (kesikli ok ile gösterilen) *height1* metot çağrısı ile dönüş değeri (*height*) yollanır.

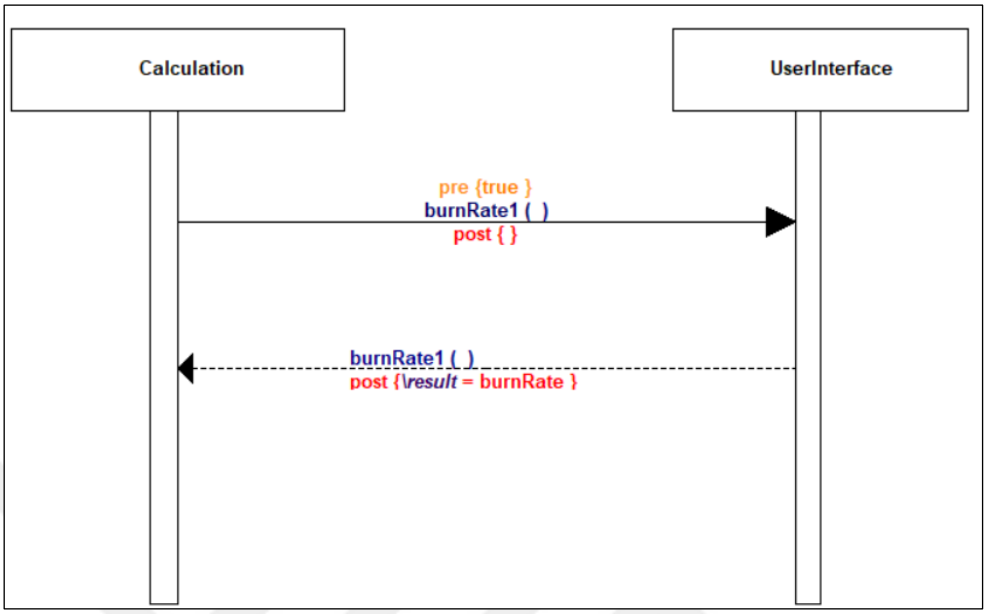
Calculation'ın *UserInterface* ile ilişkisi, *UserInterface*'in "*dataBurnRate*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.58'de gösteriliyor.



Şekil 7.58: *Calculate'*in *UserInterface*'e bağlanan istemci portun (*required port*) özellikleri.

Calculation'ın *UserInterface*'den *burnRate1* metodu ile istekte bulunabilmesi için gerekli ön şartı (!*burnRateReceived*) sağlaması gerekiyor. Bu ön şart sağlandığında *DataStore*'a *burnRate1* metot çağrısı yapılır. *UserInterface*'den alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak son koşula (*burnrate=result; burnRateReceived=true;*) göre atama yapılır.

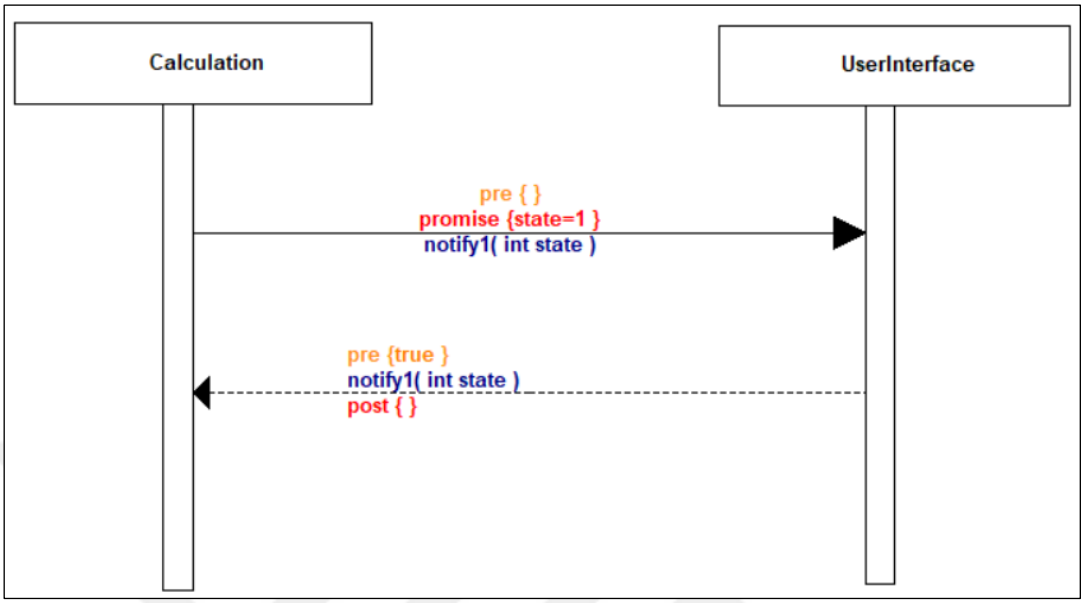
UserInterface'in Calculation ile ilişkisi, *UserInterface*'in "*dataBurnRate*" isimli sağlayıcı portuna (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.59'da gösteriliyor.



Şekil 7.59: *UserInterface*'in *Calculation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

*Calculation'*dan *UserInterface*'a *burnRate1* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. *Calculation*'a (kesikli ok ile gösterilen) *burnRate1* metot çağrısı ile dönüş değeri (*burnRate*) yollanır.

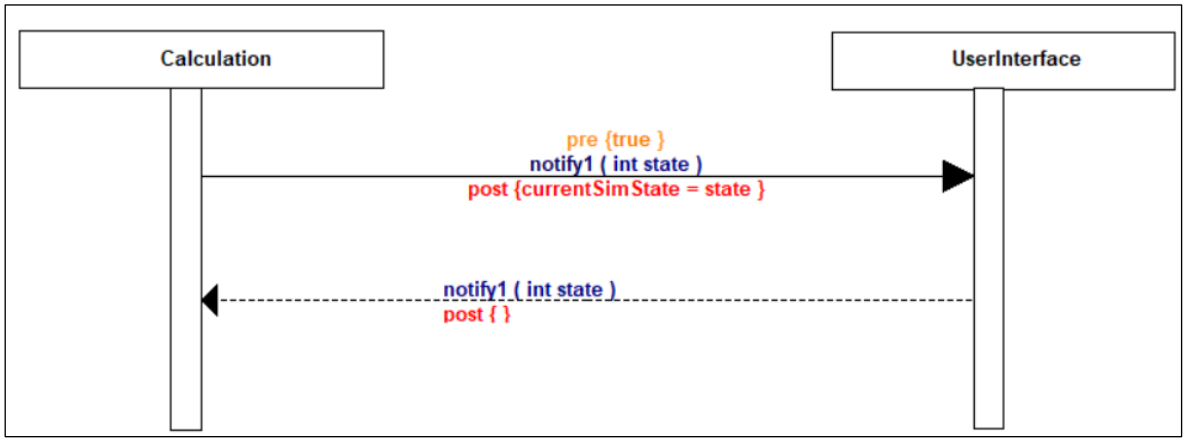
Calculation'ın *UserInterface* ile ilişkisi, *UserInterface*'in "*simState*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.60'da gösteriliyor.



Şekil 7.60: *Calculate'*in *UserInterface*'e bağlanan istemci portun (*required port*) özellikleri.

Calculation'ın *UserInterface*'den *notify1* metodu ile istekte bulunabilmesi için gerekli ön şart aranmaksızın ilgili metodun parameteresine atama gerçekleştirilir (*state=1*). Daha sonra *UserInterface*'e metot çağrısı yapılır. *UserInterface*'den alınan cevapta (kesikli ok ile gösterilen) ön şartsız (*true*) olarak kabul edilir.

UserInterface'in Calculation ile ilişkisi, *UserInterface*'in "*simState*" isimli sağlayıcı portuna (*provided port*) sağ tıklama ile açılan diyagram Şekil 7.61'de gösteriliyor.

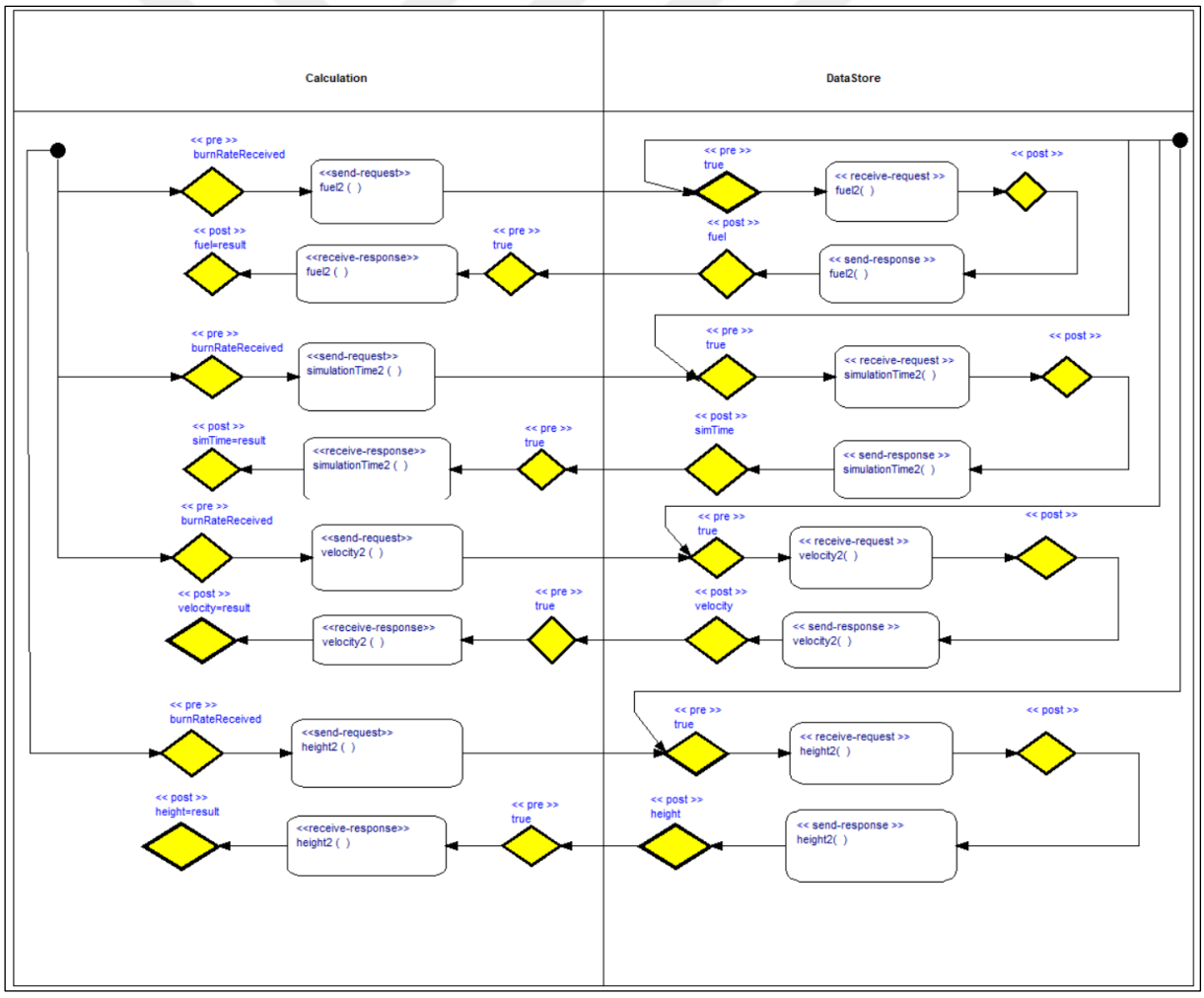


Şekil 7.61: *UserInterface*'in *Calculation*'a bağlanan sağlayıcı portun (*provided port*) özellikleri.

*Calculation'*dan *UserInterface*'a *notify1* metodu vasıtasıyla istek geldiğinde ön şartsız (*true*) olarak kabul edilir. Son şarta göre (*currentSimState=state;*) atama yapılır *Calculation*'a (kesikli ok ile gösterilen) *notify1* metot çağrısı ile dönüş değersiz yollanır.

7.6 AYA İNİŞ VAKA ANALİZİ (AKTİVİTE DİYAGRAMI İLE GENİŞLETİLMİŞ)

Şekil 7.48'de gösterilen aya iniş modelinin aktivite diyagramı ile genişletilmiş şeklinde bileşenlerinin data verileri ve özellikleri Şekil 7.49, Şekil 7.50 ve Şekil 7.51'deki gibi aynıdır. *Calculation* ile *DataStore* arasındaki ilişki, *Calculation*'nın "*dataCalc*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.62'de gösteriliyor.



Şekil 7.62: *Calculation - DataStore* aktivite diyagramı.

Şekil 7.62'de gösterilen *Calculation-DataStore* aktivite diyagramında, *Calculation*'nın *DataStore*'a *fuel2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*burnRateReceived*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *fuel2* metot çağrı aktivitesi gerçekleştirilir. *fuel2* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *fuel2* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *fuel2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *Calculation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=fuel*) belirtir. *Calculation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *fuel2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*fuel=result*) sağlar.

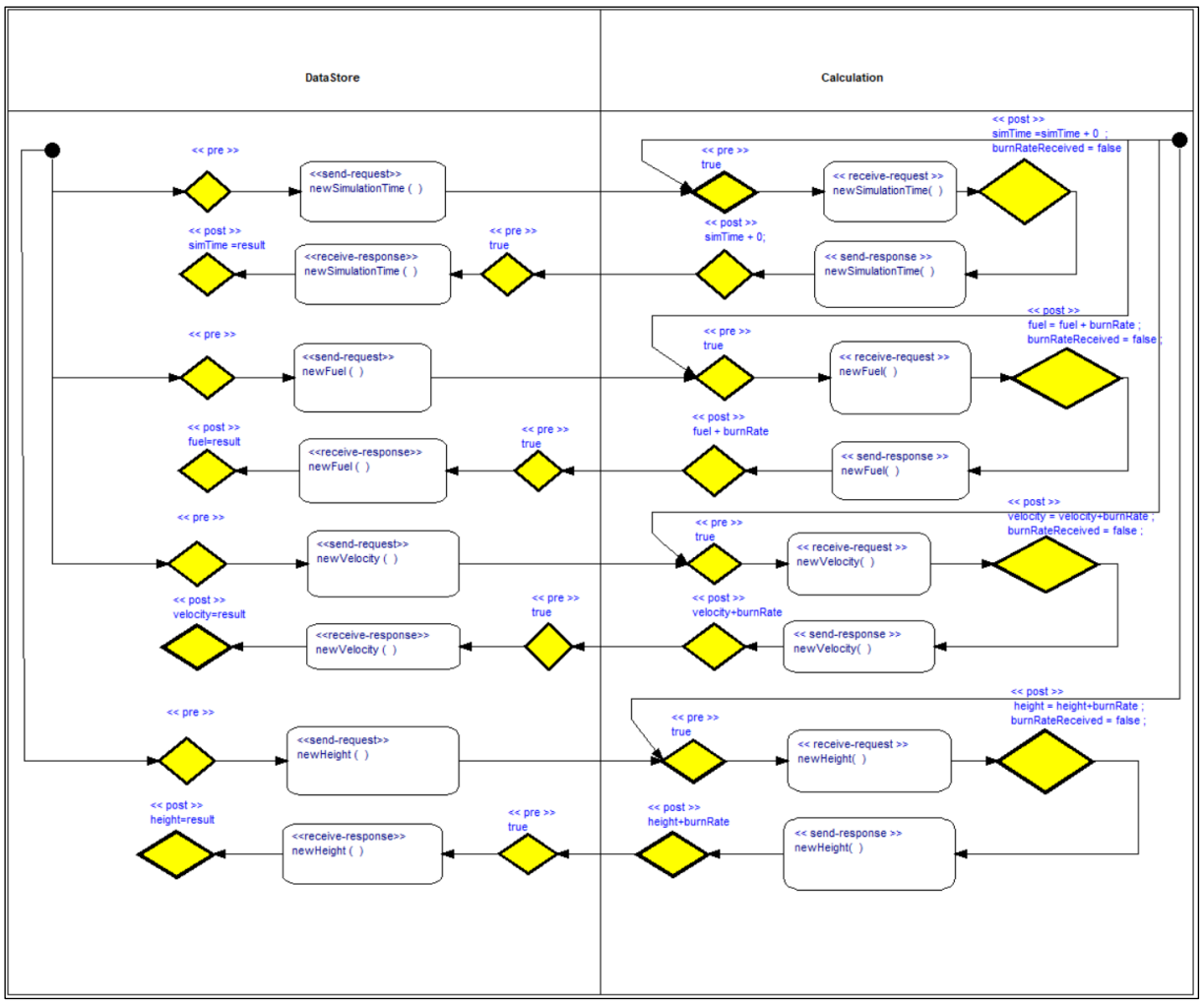
Calculation'nın *DataStore*'a *simulationTime2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*burnRateReceived*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *simulationTime2* metot çağrı aktivitesi gerçekleştirilir. *simulationTime2* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *simulationTime2* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *simulationTime2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *Calculation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=simTime*) belirtir. *Calculation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *simulationTime2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*simTime=result*) sağlar.

Calculation'nın *DataStore*'a *simulationTime2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*burnRateReceived*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *velocity2* metot çağrı aktivitesi gerçekleştirilir. *velocity2* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *velocity2* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *velocity2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *Calculation* bileşenin istemci portuna (*required port*) cevap olarak yollanır.

Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=velocity*) belirtir. *Calculation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *velocity2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*velocity=result*) sağlar.

Calculation'nın *DataStore*'a *simulationTime2* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (*burnRateReceived*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *height2* metot çağrı aktivitesi gerçekleştirilir. *height2* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *height2* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *height2* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *Calculation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=height*) belirtir. *Calculation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *height2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*height=result*) sağlar.

DataStore ile Calculation arasındaki ilişki, *DataStore*'un "*newData*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.63'de gösteriliyor.



Şekil 7.63: *DataStore - Calculation* aktivite diyagramı.

Şekil 7.63'de gösterilen *DataStore-Calculation* aktivite diyagramında, *DataStore*'un *Calculation*'a *newSimulationTime* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *newSimulationTime* metodu, *Calculation* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *newSimulationTime* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*simTime=simTime+0; burnRateReceived=false*) sağlanır ve (*send-response*) *newSimulationTime* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DataStore* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=simTime+0;*) belirtir. *DataStore*, cevabı

ön şartsız (*true*) olarak (*receive-response*) *newSimulationTime* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*simTime=result*) sağlar.

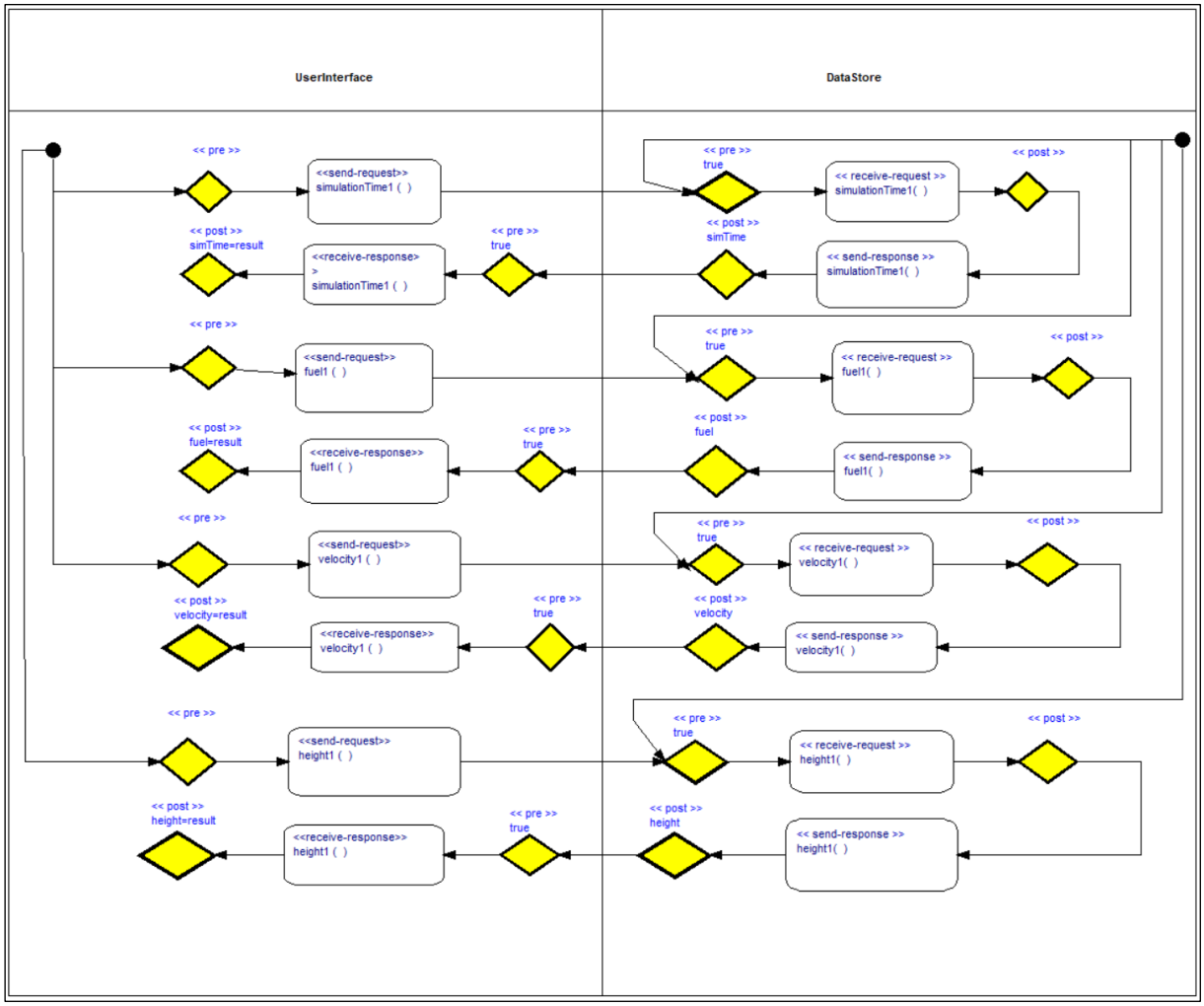
DataStore'un *Calculation*'a *newFuel* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *newFuel* metodu, *Calculation* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *newFuel* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*fuel=fuel+burnRate; burnRateReceived=false*) sağlanır ve (*send-response*) *newFuel* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DataStore* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=fuel+burnRate;*) belirtir. *DataStore*, cevabı ön şartsız (*true*) olarak (*receive-response*) *newFuel* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*fuel=result*) sağlar.

DataStore'un *Calculation*'a *newVelocity* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *newVelocity* metodu, *Calculation* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *newVelocity* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*velocity=velocity+burnRate; burnRateReceived=false*) sağlanır ve (*send-response*) *newVelocity* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DataStore* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=velocity+burnRate;*) belirtir. *DataStore*, cevabı ön şartsız (*true*) olarak (*receive-response*) *newVelocity* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*velocity=result*) sağlar.

DataStore'un *Calculation*'a *newHeight* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *newHeight* metodu, *Calculation* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *newHeight* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*height=height+burnRate; burnRateReceived=false*) sağlanır ve (*send-response*) *newHeight* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *DataStore* bileşenin istemci

portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=height+burnRate;*) belirtir. *DataStore*, cevabı ön şartsız (*true*) olarak (*receive-response*) *newHeight* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*height=result*) sağlar.

UserInterface ile DataStore arasındaki ilişki, *UserInterface*'in "*data*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.64'de gösteriliyor.



Şekil 7.64: *UserInterface - DataStore* aktivite diyagramı.

Şekil 7.64'de gösterilen *UserInterface-DataStore* aktivite diyagramında, *UserInterface*'in *DataStore*'a *simulationTime1* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *simulationTime1* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*)

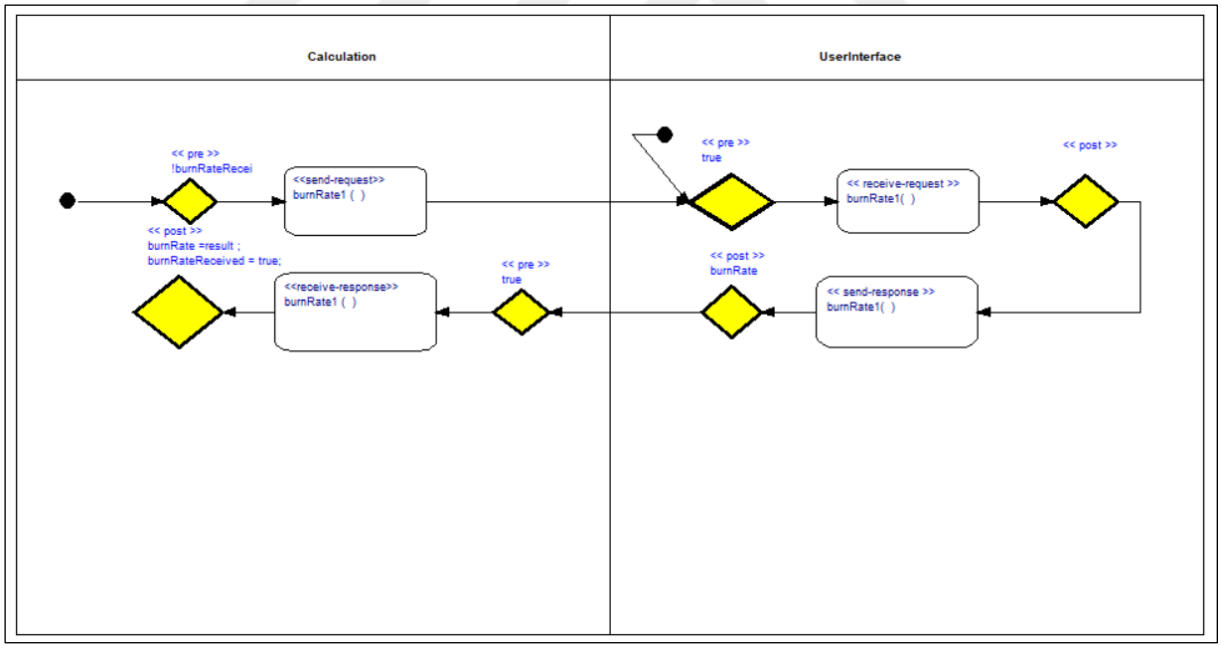
ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *newSimulationTime* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *simulationTime1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *simulationTime1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *UserInterface* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=simTime*) belirtir. *UserInterface*, cevabı ön şartsız (*true*) olarak (*receive-response*) *simulationTime1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*simTime=result*) sağlar.

UserInterface'in *DataStore*'a *fuel1* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *fuel1* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *fuel1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *fuel1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *fuel1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *UserInterface* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=fuel*) belirtir. *UserInterface*, cevabı ön şartsız (*true*) olarak (*receive-response*) *fuel1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*fuel=result*) sağlar.

UserInterface'in *DataStore*'a *velocity1* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *velocity1* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *velocity1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *velocity1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *velocity1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *UserInterface* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=velocity*) belirtir. *UserInterface*, cevabı ön şartsız (*true*) olarak (*receive-response*) *velocity1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*velocity=result*) sağlar.

UserInterface'in *DataStore*'a *height1* metot çağrısı ön/sonraki koşul sembolündeki ön şartsız gerçekleştirilir. *height1* metodu, *DataStore* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *height1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *height1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *height1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *UserInterface* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*result=height*) belirtir. *UserInterface*, cevabı ön şartsız (*true*) olarak (*receive-response*) *velocity1* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*height=result*) sağlar.

Calculation ile UserInterface arasındaki ilişki, *Calculation*'nın "*dataBurnRate*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.65'de gösteriliyor.

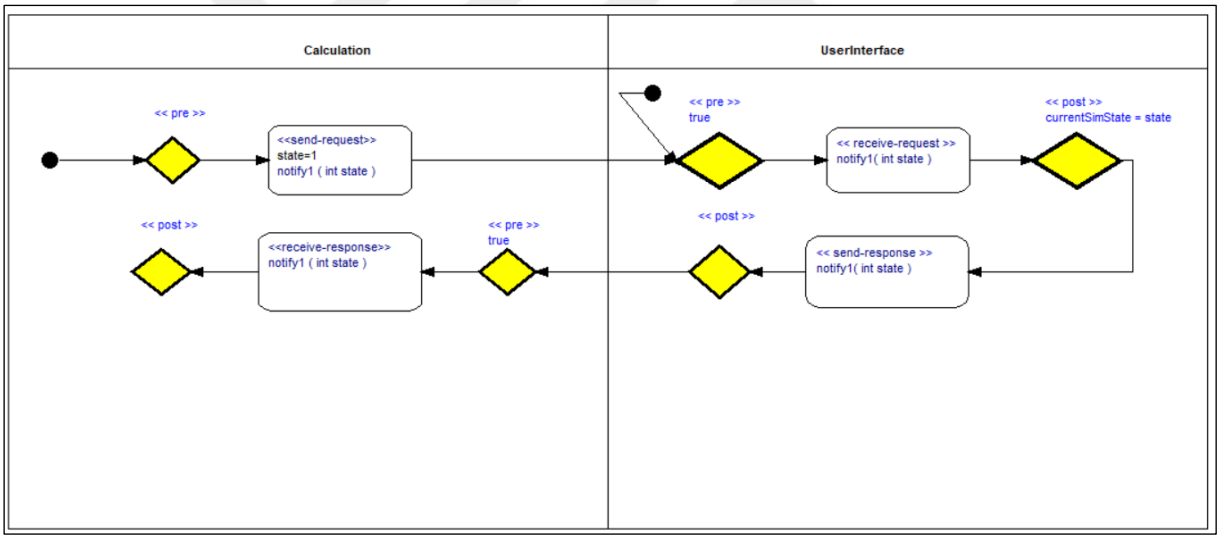


Şekil 7.65: *Calculation - UserInterface* aktivite diyagramı.

Şekil 7.65'de gösterilen *Calculation-UserInterface* aktivite diyagramında, *Calculation*'nın *UserInterface*'e *burnRate1* metot çağrısı yapmadan önce ön/sonraki koşul sembolündeki ön şartı (!*burnRateReceived*) kontrol edilir. Eğer ön koşul şartı sağlanırsa *burnRate1* metot çağrı aktivitesi gerçekleştirilir. *burnRate1* metodu, *UserInterface* bileşenin sağlayıcı portuna

(*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *burnRate1* metot çağrı aktivitesi gerçekleştirilir. Ön/sonraki koşul sembolündeki son şartı olmadığından (*send-response*) *burnRate1* metot çağrı aktivitesi gerçekleştirilerek, dönüş değerli olarak *Calculation* bileşenin istemci portuna (*required port*) cevap olarak yollanır. Buradaki ön/sonraki koşul sembolündeki son şartı dönüş değerini (*\result=burnRate*) belirtir. *Calculation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *fuel2* metot çağrı aktivitesi ile alır ve ön/sonraki koşul sembolündeki son şartı (*burnRate=result; burnRateReceived=true;*) sağlar.

Calculation ile UserInterface arasındaki ilişki, *Calculation*'nın "*simState*" isimli istemci portuna (*required port*) sağ tıklama ile açılan diyagram Şekil 7.66'da gösteriliyor.



Şekil 7.66: *Calculation - UserInterface* aktivite diyagramı.

Şekil 7.66'da gösterilen *Calculation-UserInterface* aktivite diyagramında, *Calculation*'nın *UserInterface*'e *notify1* metot çağrısı ön şartsız, metot parametresine yapılan atamayla (*state=1*) gerçekleştirilir. *notify1* metodu, *UserInterface* bileşenin sağlayıcı portuna (*provided port*) ulaşınca, ön/sonraki koşul sembolündeki ön şartsız (*true*) (*receive-request*) *notify1* metot çağrı aktivitesi gerçekleştirilir. Sonra, ön/sonraki koşul sembolündeki son şartı (*currentSimState=state;*) sağlanır ve (*send-response*) *newSimulationTime* metot çağrı aktivitesi gerçekleştirilerek dönüş değeri olmadan *Calculation* bileşenin istemci portuna (*required port*)

cevap olarak yollanır. *Calculation*, cevabı ön şartsız (*true*) olarak (*receive-response*) *notify1* metot çağrı aktivitesi ile alır.

Ay iniş modelinin, MetaEdit+ editörü dönüştürücüsünden elde edilen ProMeLa dosyası (lunar.pml) Liste 7.5'te gösterilmektedir.

Liste 7.5: lunar.pml dosyası.

```
1       mtype =
{height1,fuel1,simulationTime1,velocity1,notify1,burnRate1,fuel2,simulationTime2,
2                velocity2,height2,newSimulationTime,newHeight,newFuel,newVelocity}
3       chan ch_data_REQ= [1] of{ mtype  };
4       chan ch_simState_REQ= [1] of{ mtype ,int };
5       chan ch_dataBurnRate_REQ= [1] of{ mtype  };
6       chan ch_dataCalc_REQ= [1] of{ mtype  };
7       chan ch_newData_REQ= [1] of{ mtype  };
8
9       chan ch_data_RES= [1] of{ mtype, int} ;
10      chan ch_simState_RES= [1] of{ mtype, int} ;
11      chan ch_dataBurnRate_RES= [1] of{ mtype, int} ;
12      chan ch_dataCalc_RES= [1] of{ mtype, int} ;
13      chan ch_newData_RES= [1] of{ mtype, int} ;
14
15      proctype DataStore() {
16       int height=1;
17       int velocity=1;
18       int fuel=1;
19       int simTime=1;
20       int result;
21       int responseWaiting= false;
22
23       do
24       :: responseWaiting==false ->
25             responseWaiting=true;
26             ch_newData_REQ !newSimulationTime;
27       ::ch_newData_RES ?newSimulationTime,result->
28         if
29         ::(true)->simTime =result;
30         ::!( true )->assert(false);
31         fi ;
32         responseWaiting=false ;
33       :: responseWaiting==false ->
34             responseWaiting=true;
35             ch_newData_REQ !newHeight;
36       ::ch_newData_RES ?newHeight,result->
37         if
38         ::(true)->height=result;
39         ::!( true )->assert(false);
40         fi ;
41         responseWaiting=false ;
42       ::responseWaiting==false ->
43         responseWaiting=true;
44         ch_newData_REQ !newFuel;
45       ::ch_newData_RES ?newFuel,result->
46         if
47         ::(true)->fuel=result;
48         ::!( true )->assert(false);
49         fi ;
50         responseWaiting=false ;
51       ::responseWaiting==false ->
52         responseWaiting=true;
```

Liste 7.5: lunar.pml dosyası (devam).

```
53          ch_newData_REQ !newVelocity;
54        ::ch_newData_RES ?newVelocity,result->
55          if
56          ::(true)->velocity =result;
57          ::!( true )->assert(false);
58          fi ;
59         responseWaiting=false ;
60        ::ch_data_REQ ?velocity1->
61          if
62          ::(true)->
63               ch_data_RES !velocity1,velocity;
64          ::!( (true)  )->
65               ch_data_REQ !velocity1;
66          fi
67        ::ch_data_REQ ?fuel1->
68          if
69          ::(true)->
70               ch_data_RES !fuel1,fuel;
71          ::!( (true)  )->
72               ch_data_REQ !fuel1;
73          fi
74        ::ch_data_REQ ?simulationTime1->
75          if
76          ::(true)->
77               ch_data_RES !simulationTime1,simTime;
78          ::!( (true)  )->
79               ch_data_REQ !simulationTime1;
80          fi
81        ::ch_data_REQ ?height1->
82          if
83          ::(true)->
84               ch_data_RES !height1,height;
85          ::!( (true)  )->
86               ch_data_REQ !height1;
87          fi
88        ::ch_dataCalc_REQ ?fuel2->
89          if
90          ::(true)->
91               ch_dataCalc_RES !fuel2,fuel;
92          ::!( (true)  )->
93               ch_dataCalc_REQ !fuel2;
94          fi
95        ::ch_dataCalc_REQ ?height2->
96          if
97          ::(true)->
98               ch_dataCalc_RES !height2,height;
99          ::!( (true)  )->
100              ch_dataCalc_REQ !height2;
101         fi
102       ::ch_dataCalc_REQ ?simulationTime2->
103         if
104         ::(true)->
105              ch_dataCalc_RES !simulationTime2,simTime;
106         ::!( (true)  )->
107              ch_dataCalc_REQ !simulationTime2;
108         fi
109       ::ch_dataCalc_REQ ?velocity2->
110         if
111         ::(true)->
112              ch_dataCalc_RES !velocity2,velocity;
113         ::!( (true)  )->
114              ch_dataCalc_REQ !velocity2;
115         fi
116       od
```

Liste 7.5: lunar.pml dosyası (devam).

```
117     }
118     proctype Calculation() {
119      int burnRate=5;
120      int simTime=5;
121      int fuel=5;
122      int height=5;
123      int velocity=5;
124      bool burnRateReceived=false;
125      int state;
126      int result;
127      int responseWaiting= false;
128
129      do
130      :: responseWaiting==false ->
131            responseWaiting=true;
132            state=1
133            ch_simState_REQ !notify1,state
134      ::ch_simState_RES ?notify1,result->
135        if
136        ::(true)->;
137        ::!( true )->assert(false);
138        fi ;
139        responseWaiting=false ;
140      ::!burnRateReceived && responseWaiting==false ->
141            responseWaiting=true ;
142            ch_dataBurnRate_REQ !burnRate1;
143      ::ch_dataBurnRate_RES ?burnRate1,result->
144        if
145        ::(true)->burnRate =result ; burnRateReceived = true;;
146        ::!( true )->assert(false);
147        fi ;
148        responseWaiting=false ;
149      :: burnRateReceived && responseWaiting==false ->
150            responseWaiting=true ;
151            ch_dataCalc_REQ !fuel2;
152      ::ch_dataCalc_RES ?fuel2,result->
153        if
154        ::(true)->fuel=result;
155        ::!( true )->assert(false);
156        fi ;
157        responseWaiting=false ;
158      ::burnRateReceived && responseWaiting==false ->
159        responseWaiting=true ;
160        ch_dataCalc_REQ !simulationTime2;
161      ::ch_dataCalc_RES ?simulationTime2,result->
162        if
163        ::(true)->simTime=result;
164        ::!( true )->assert(false);
165        fi ;
166        responseWaiting=false ;
167      ::burnRateReceived && responseWaiting==false ->
168        responseWaiting=true ;
169        ch_dataCalc_REQ !velocity2;
170      ::ch_dataCalc_RES ?velocity2,result->
171        if
172        ::(true)->velocity=result;
173        ::!( true )->assert(false);
174        fi ;
175        responseWaiting=false ;
176      ::burnRateReceived && responseWaiting==false ->
177        responseWaiting=true ;
178        ch_dataCalc_REQ !height2;
179      ::ch_dataCalc_RES ?height2,result->
180        if
```

Liste 7.5: lunar.pml dosyası (devam).

```
181       ::(true)->height=result;
182       ::!( true )->assert(false);
183       fi ;
184       responseWaiting=false ;
185      ::ch_newData_REQ ?newHeight->
186           if
187           ::(true)->
188             height = height+burnRate ;  burnRateReceived = false ;
189             ch_newData_RES !newHeight,height+burnRate;
190           ::!( (true)  )->
191             ch_newData_REQ !newHeight;
192           fi
193      ::ch_newData_REQ ?newFuel->
194           if
195           ::(true )->
196             fuel = fuel + burnRate ; burnRateReceived = false ;
197             ch_newData_RES !newFuel,fuel + burnRate;
198           ::!( (true )  )->
199             ch_newData_REQ !newFuel;
200           fi
201      ::ch_newData_REQ ?newSimulationTime->
202           if
203           ::(true)->
204             simTime =simTime + 0  ; burnRateReceived = false
205             ch_newData_RES !newSimulationTime,simTime + 0;;
206           ::!( (true)  )->
207             ch_newData_REQ !newSimulationTime;
208           fi
209      ::ch_newData_REQ ?newVelocity->
210           if
211           ::(true)->
212             velocity = velocity+burnRate ;  burnRateReceived = false ;
213             ch_newData_RES !newVelocity,velocity+burnRate ;
214           ::!( (true)  )->
215             ch_newData_REQ !newVelocity;
216           fi
217      od
218    }
219    proctype UserInterface() {
220     int burnRate=0;
221     int simTime=0;
222     int fuel=0;
223     int velocity=0;
224     int height=0;
225     int currentSimState=0;
226     int state;
227     int result;
228     int responseWaiting= false;
229     do
230     ::responseWaiting==false ->
231       responseWaiting=true;
232       ch_data_REQ !height1;
233     ::ch_data_RES ?height1,result->
234       if
235       ::(true)->height=result;
236       ::!( true )->assert(false);
237       fi ;
238       responseWaiting=false ;
239     :: responseWaiting==false ->
240           responseWaiting=true;
241           ch_data_REQ !fuel1;
242     ::ch_data_RES ?fuel1,result->
243       if
244       ::(true)->fuel=result;
```

Liste 7.5: lunar.pml dosyası (devam).

```
245       ::!( true )->assert(false);
246       fi ;
247       responseWaiting=false ;
248     ::responseWaiting==false ->
249       responseWaiting=true;
250       ch_data_REQ !simulationTime1;
251     ::ch_data_RES ?simulationTime1,result->
252       if
253       ::(true)->simTime=result;
254       ::!( true )->assert(false);
255       fi ;
256       responseWaiting=false ;
257     ::responseWaiting==false ->
258       responseWaiting=true;
259       ch_data_REQ !velocity1;
260     ::ch_data_RES ?velocity1,result->
261       if
262       ::(true)->velocity=result;
263       ::!( true )->assert(false);
264       fi ;
265       responseWaiting=false ;
266     ::ch_dataBurnRate_REQ ?burnRate1->
267       if
268       ::(true)->
269            ch_dataBurnRate_RES !burnRate1,burnRate;
270       ::!( (true)  )->
271            ch_dataBurnRate_REQ !burnRate1;
272       fi
273     ::ch_simState_REQ ?notify1,state->
274       if
275       ::(true)->
276            currentSimState = state
277            ch_simState_RES !notify1,0;
278       ::!( (true)  )->
279            ch_simState_REQ !notify1,state
280       fi
281     od
282    }
283    active proctype configuration()
284    {
285      atomic {
286       run DataStore();
287       run Calculation();
288       run UserInterface();
289    }
290
291    }
292
```

Aya iniş modelinin, MetaEdit+ editörü dönüştürücüsünden elde edilen java dosyası (Configuration.java) Liste 7.6'da gösterilmektedir.

Liste 7.6: Configuration.java dosyası.

```
1  public class Configuration {
2       DataStore datastoreIns;
3       Calculation calculationIns;
4       UserInterface userinterfaceIns;
5
6       UserInterfaceDataStore_data_Adapter userinterfacedatastore_data_AdapterIns;
7
8       CalculationUserInterface_simState_Adapter
            calculationuserinterface_simState_AdapterIns;
9
10      CalculationUserInterface_dataBurnRate_Adapter
             calculationuserinterface_dataBurnRate_AdapterIns;
11
12      CalculationDataStore_dataCalc_Adapter calculationdatastore_dataCalc_AdapterIns;
13
14      DataStoreCalculation_newData_Adapter datastorecalculation_newData_AdapterIns;
15
16      public Configuration() {
17        datastoreIns = new DataStore();
18        calculationIns = new Calculation();
19        userinterfaceIns = new UserInterface();
20        try {
21             userinterfacedatastore_data_AdapterIns =
22                    new UserInterfaceDataStore_data_Adapter(datastoreIns);
23             userinterfaceIns.userinterfacedatastore_data_InterfaceIns =
24                    userinterfacedatastore_data_AdapterIns;
25
26        } catch (Exception e) {
27                    e.printStackTrace();
28        }
29        try {
30          calculationuserinterface_simState_AdapterIns =
31                new CalculationUserInterface_simState_Adapter(userinterfaceIns);
32             calculationIns.calculationuserinterface_simState_InterfaceIns =
33                calculationuserinterface_simState_AdapterIns;
34
35         } catch (Exception e) {
36                    e.printStackTrace();
37         }
38         try {
39              calculationuserinterface_dataBurnRate_AdapterIns =
40                 new CalculationUserInterface_dataBurnRate_Adapter(userinterfaceIns);
41              calculationIns.calculationuserinterface_dataBurnRate_InterfaceIns =
42                    calculationuserinterface_dataBurnRate_AdapterIns;
43
44         } catch (Exception e) {
45                    e.printStackTrace();
46         }
47         try {
48               calculationdatastore_dataCalc_AdapterIns =
49              new CalculationDataStore_dataCalc_Adapter(datastoreIns);
50               calculationIns.calculationdatastore_dataCalc_InterfaceIns =
51              calculationdatastore_dataCalc_AdapterIns;
52
53         } catch (Exception e) {
54                    e.printStackTrace();
55         }
56         try {
57               datastorecalculation_newData_AdapterIns =
58                     new DataStoreCalculation_newData_Adapter(calculationIns);
59              datastoreIns.datastorecalculation_newData_InterfaceIns =
60                    datastorecalculation_newData_AdapterIns;
61
62         } catch (Exception e) {
```

Liste 7.6: Configuration.java dosyası (devam).

```
63                      e.printStackTrace();
64          }
65          datastoreIns.start();
66          calculationIns.start();
67          userinterfaceIns.start();
68
69      }
70
71      public static void main(String args[]) throws Exception {
72              new Configuration();
73      }
74  }
75
76  interface UserInterfaceDataStore_data_Interface {
77      public void height1() throws Exception;
78
79      public void fuel1() throws Exception;
80
81      public void simulationTime1() throws Exception;
82
83      public void velocity1() throws Exception;
84  }
85
86  interface CalculationUserInterface_simState_Interface {
87      public void notify1(int state) throws Exception;
88  }
89
90  interface CalculationUserInterface_dataBurnRate_Interface {
91      public void burnRate1() throws Exception;
92  }
93
94  interface CalculationDataStore_dataCalc_Interface {
95       public void fuel2() throws Exception;
96
97       public void simulationTime2() throws Exception;
98
99       public void velocity2() throws Exception;
100
101   public void height2() throws Exception;
102 }
103
104 interface DataStoreCalculation_newData_Interface {
105       public void newSimulationTime() throws Exception;
106
107       public void newHeight() throws Exception;
108
109       public void newFuel() throws Exception;
110
111       public void newVelocity() throws Exception;
112 }
113
114 class UserInterfaceDataStore_data_Adapter implements
UserInterfaceDataStore_data_Interface {
115      DataStore datastore_ins;
116
117      public UserInterfaceDataStore_data_Adapter(DataStore datastore_arg) throws
Exception {
118               datastore_ins = datastore_arg;
119      }
120
121      public void height1() {
122              try {
123                      datastore_ins.height1();
124              } catch (Exception e) {
```

Liste 7.6: Configuration.java dosyası (devam).

```
125                     e.printStackTrace();
126             }
127     }
128
129     public void fuel1() {
130             try {
131                     datastore_ins.fuel1();
132             } catch (Exception e) {
133                     e.printStackTrace();
134             }
135     }
136
137     public void simulationTime1() {
138             try {
139                     datastore_ins.simulationTime1();
140             } catch (Exception e) {
141                     e.printStackTrace();
142             }
143     }
144
145     public void velocity1() {
146             try {
147                     datastore_ins.velocity1();
148             } catch (Exception e) {
149                     e.printStackTrace();
150             }
151     }
152 }
153
154 class CalculationUserInterface_simState_Adapter
      implements CalculationUserInterface_simState_Interface {
155     UserInterface userinterface_ins;
156
157     public CalculationUserInterface_simState_Adapter(UserInterface userinterface_arg)
        throws Exception {
158         userinterface_ins = userinterface_arg;
159     }
160
161     public void notify1(int state) {
162             try {
163                     userinterface_ins.notify1(state);
164             } catch (Exception e) {
165                     e.printStackTrace();
166             }
167     }
168 }
169
170 class CalculationUserInterface_dataBurnRate_Adapter
      implements CalculationUserInterface_dataBurnRate_Interface {
171     UserInterface userinterface_ins;
172
173     public CalculationUserInterface_dataBurnRate_Adapter(UserInterface
            userinterface_arg) throws Exception {
174         userinterface_ins = userinterface_arg;
175     }
176
177     public void burnRate1() {
178             try {
179                     userinterface_ins.burnRate1();
180             } catch (Exception e) {
181                     e.printStackTrace();
182             }
183     }
184 }
```

Liste 7.6: Configuration.java dosyası (devam).

```
185
186 class CalculationDataStore_dataCalc_Adapter
      implements CalculationDataStore_dataCalc_Interface {
187     DataStore datastore_ins;
188
189     public CalculationDataStore_dataCalc_Adapter(DataStore datastore_arg) throws
            Exception {
190           datastore_ins = datastore_arg;
191     }
192
193     public void fuel2() {
194           try {
195                 datastore_ins.fuel2();
196           } catch (Exception e) {
197                 e.printStackTrace();
198           }
199     }
200
201     public void simulationTime2() {
202           try {
203                 datastore_ins.simulationTime2();
204           } catch (Exception e) {
205                 e.printStackTrace();
206           }
207     }
208
209     public void velocity2() {
210           try {
211                 datastore_ins.velocity2();
212           } catch (Exception e) {
213                 e.printStackTrace();
214           }
215     }
216
217     public void height2() {
218           try {
219                 datastore_ins.height2();
220           } catch (Exception e) {
221                 e.printStackTrace();
222           }
223     }
224 }
225
226 class DataStoreCalculation_newData_Adapter
      implements DataStoreCalculation_newData_Interface {
227     Calculation calculation_ins;
228
229     public DataStoreCalculation_newData_Adapter(Calculation calculation_arg)
        throws Exception {
230           calculation_ins = calculation_arg;
231     }
232
233     public void newSimulationTime() {
234           try {
235                 calculation_ins.newSimulationTime();
236           } catch (Exception e) {
237                 e.printStackTrace();
238           }
239     }
240
241     public void newHeight() {
242           try {
243                 calculation_ins.newHeight();
244           } catch (Exception e) {
```

Liste 7.6: Configuration.java dosyası (devam).

```
245                     e.printStackTrace();
246             }
247     }
248
249     public void newFuel() {
250             try {
251                     calculation_ins.newFuel();
252             } catch (Exception e) {
253                     e.printStackTrace();
254             }
255     }
256
257     public void newVelocity() {
258             try {
259                     calculation_ins.newVelocity();
260             } catch (Exception e) {
261                     e.printStackTrace();
262             }
263     }
264 }
265
266 class DataStore extends Thread {
267     int height = 1;
268     int velocity = 1;
269     int fuel = 1;
270     int simTime = 1;
271     private int result = 0;
272     DataStoreCalculation_newData_Interface datastorecalculation_newData_InterfaceIns;
273     private boolean responseWaiting = false;
274
275     public void newSimulationTime() throws Exception {
276             if (!responseWaiting) {
277                     responseWaiting = true;
278                     datastorecalculation_newData_InterfaceIns.newSimulationTime();
279             }
280             if (true) {
281                     simTime = result;
282             }
283             responseWaiting = false;
284     }
285
286     public void newHeight() throws Exception {
287             if (!responseWaiting) {
288                     responseWaiting = true;
289                     datastorecalculation_newData_InterfaceIns.newHeight();
290             }
291             if (true) {
292                     height = result;
293             }
294             responseWaiting = false;
295     }
296
297     public void newFuel() throws Exception {
298             if (!responseWaiting) {
299                     responseWaiting = true;
300                     datastorecalculation_newData_InterfaceIns.newFuel();
301             }
302             if (true) {
303                     fuel = result;
304             }
305             responseWaiting = false;
306     }
307
308     public void newVelocity() throws Exception {
```

Liste 7.6: Configuration.java dosyası (devam).

```
309             if (!responseWaiting) {
310                     responseWaiting = true;
311                     datastorecalculation_newData_InterfaceIns.newVelocity();
312             }
313             if (true) {
314                     velocity = result;
315             }
316             responseWaiting = false;
317     }
318
319
320     public int velocity1() throws Exception {
321             if (true) {
322                     ;
323                     result = simTime;
324                     result = velocity;
325                     result = height;
326                     result = fuel;
327             }
328             return result;
329     }
330
331     public int fuel1() throws Exception {
332             if (true) {
333                     ;
334                     result = simTime;
335                     result = velocity;
336                     result = height;
337                     result = fuel;
338             }
339             return result;
340     }
341
342     public int simulationTime1() throws Exception {
343             if (true) {
344                     ;
345                     result = simTime;
346                     result = velocity;
347                     result = height;
348                     result = fuel;
349             }
350             return result;
351     }
352
353     public int height1() throws Exception {
354             if (true) {
355                     ;
356                     result = simTime;
357                     result = velocity;
358                     result = height;
359                     result = fuel;
360             }
361             return result;
362     }
363
364     public int fuel2() throws Exception {
365             if (true) {
366                     ;
367                     result = height;
368                     result = velocity;
369                     result = simTime;
370                     result = fuel;
371             }
372             return result;
```

Liste 7.6: Configuration.java dosyası (devam).

```
373     }
374
375     public int height2() throws Exception {
376           if (true) {
377                 ;
378                 result = height;
379                 result = velocity;
380                 result = simTime;
381                 result = fuel;
382           }
383           return result;
384     }
385
386     public int simulationTime2() throws Exception {
387           if (true) {
388                 ;
389                 result = height;
390                 result = velocity;
391                 result = simTime;
392                 result = fuel;
393           }
394           return result;
395     }
396
397     public int velocity2() throws Exception {
398           if (true) {
399                 ;
400                 result = height;
401                 result = velocity;
402                 result = simTime;
403                 result = fuel;
404           }
405           return result;
406     }
407
408     public void run() {
409           while (true) {
410                 try {
411                       newSimulationTime();
412                       newHeight();
413                       newFuel();
414                       newVelocity();
415                 } catch (Exception e) {
416                       e.printStackTrace();
417                 }
418           }
419     }
420 }
421
422 class Calculation extends Thread {
423     int burnRate = 5;
424     int simTime = 5;
425     int fuel = 5;
426     int height = 5;
427     int velocity = 5;
428     boolean burnRateReceived = false;
429     private int result = 0;
430     CalculationUserInterface_simState_Interface
                  calculationuserinterface_simState_InterfaceIns;
431     CalculationUserInterface_dataBurnRate_Interface
                  calculationuserinterface_dataBurnRate_InterfaceIns;
432     CalculationDataStore_dataCalc_Interface
                  calculationdatastore_dataCalc_InterfaceIns;
433     private boolean responseWaiting = false;
```

Liste 7.6: Configuration.java dosyası (devam).

```
434
435     public void notify1(int state) throws Exception {
436           if (!responseWaiting) {
437                 responseWaiting = true;
438                 state = 1;
439                 calculationuserinterface_simState_InterfaceIns.notify1(state);
440           }
441           if (true) {
442
443           }
444           responseWaiting = false;
445     }
446
447     public void burnRate1() throws Exception {
448           if (!burnRateReceived && !responseWaiting) {
449                 responseWaiting = true;
450                 calculationuserinterface_dataBurnRate_InterfaceIns.burnRate1();
451           }
452           if (true) {
453                 burnRate = result;
454                 burnRateReceived = true;
455                 ;
456           }
457           responseWaiting = false;
458     }
459
460     public void fuel2() throws Exception {
461           if (burnRateReceived && !responseWaiting) {
462                 responseWaiting = true;
463                 calculationdatastore_dataCalc_InterfaceIns.fuel2();
464           }
465           if (true) {
466                 fuel = result;
467           }
468           responseWaiting = false;
469     }
470
471     public void simulationTime2() throws Exception {
472           if (burnRateReceived && !responseWaiting) {
473                 responseWaiting = true;
474                 calculationdatastore_dataCalc_InterfaceIns.simulationTime2();
475           }
476           if (true) {
477                 simTime = result;
478           }
479           responseWaiting = false;
480     }
481
482     public void velocity2() throws Exception {
483           if (burnRateReceived && !responseWaiting) {
484                 responseWaiting = true;
485                 calculationdatastore_dataCalc_InterfaceIns.velocity2();
486           }
487           if (true) {
488                 velocity = result;
489           }
490           responseWaiting = false;
491     }
492
493     public void height2() throws Exception {
494           if (burnRateReceived && !responseWaiting) {
495                 responseWaiting = true;
496                 calculationdatastore_dataCalc_InterfaceIns.height2();
497           }
```

Liste 7.6: Configuration.java dosyası (devam).

```
498             if (true) {
499                     height = result;
500             }
501             responseWaiting = false;
502     }
503
504     public int newHeight() throws Exception {
505             if (true) {
506                     height = height + burnRate;
507                     burnRateReceived = false;
508                     ;
509                     result = simTime + 0;
510                     ;
511                     result = velocity + burnRate;
512                     result = height + burnRate;
513                     result = fuel + burnRate;
514             }
515             return result;
516     }
517
518     public int newFuel() throws Exception {
519             if (true) {
520                     fuel = fuel + burnRate;
521                     burnRateReceived = false;
522                     ;
523                     result = simTime + 0;
524                     ;
525                     result = velocity + burnRate;
526                     result = height + burnRate;
527                     result = fuel + burnRate;
528             }
529             return result;
530     }
531
532     public int newSimulationTime() throws Exception {
533             if (true) {
534                     simTime = simTime + 0;
535                     burnRateReceived = false;
536                     result = simTime + 0;
537                     ;
538                     result = velocity + burnRate;
539                     result = height + burnRate;
540                     result = fuel + burnRate;
541             }
542             return result;
543     }
544
545     public int newVelocity() throws Exception {
546             if (true) {
547                     velocity = velocity + burnRate;
548                     burnRateReceived = false;
549                     ;
550                     result = simTime + 0;
551                     ;
552                     result = velocity + burnRate;
553                     result = height + burnRate;
554                     result = fuel + burnRate;
555             }
556             return result;
557     }
558
559     public void run() {
560             while (true) {
561                     try {
```

Liste 7.6: Configuration.java dosyası (devam).

```
562                         notify1(0);
563                   } catch (Exception e) {
564                         e.printStackTrace();
565                   }
566                   try {
567                         burnRate1();
568                   } catch (Exception e) {
569                         e.printStackTrace();
570                   }
571                   try {
572                         fuel2();
573                         simulationTime2();
574                         velocity2();
575                         height2();
576                   } catch (Exception e) {
577                         e.printStackTrace();
578                   }
579             }
580       }
581 }
582
583 class UserInterface extends Thread {
584       int burnRate = 0;
585       int simTime = 0;
586       int fuel = 0;
587       int velocity = 0;
588       int height = 0;
589       int currentSimState = 0;
590       private int result = 0;
591       UserInterfaceDataStore_data_Interface userinterfacedatastore_data_InterfaceIns;
592       private boolean responseWaiting = false;
593
594       public void height1() throws Exception {
595             if (!responseWaiting) {
596                   responseWaiting = true;
597                   userinterfacedatastore_data_InterfaceIns.height1();
598             }
599             if (true) {
600                   height = result;
601             }
602             responseWaiting = false;
603       }
604
605       public void fuel1() throws Exception {
606             if (!responseWaiting) {
607                   responseWaiting = true;
608                   userinterfacedatastore_data_InterfaceIns.fuel1();
609             }
610             if (true) {
611                   fuel = result;
612             }
613             responseWaiting = false;
614       }
615
616       public void simulationTime1() throws Exception {
617             if (!responseWaiting) {
618                   responseWaiting = true;
619                   userinterfacedatastore_data_InterfaceIns.simulationTime1();
620             }
621             if (true) {
622                   simTime = result;
623             }
624             responseWaiting = false;
625       }
```

Liste 7.6: Configuration.java dosyası (devam).

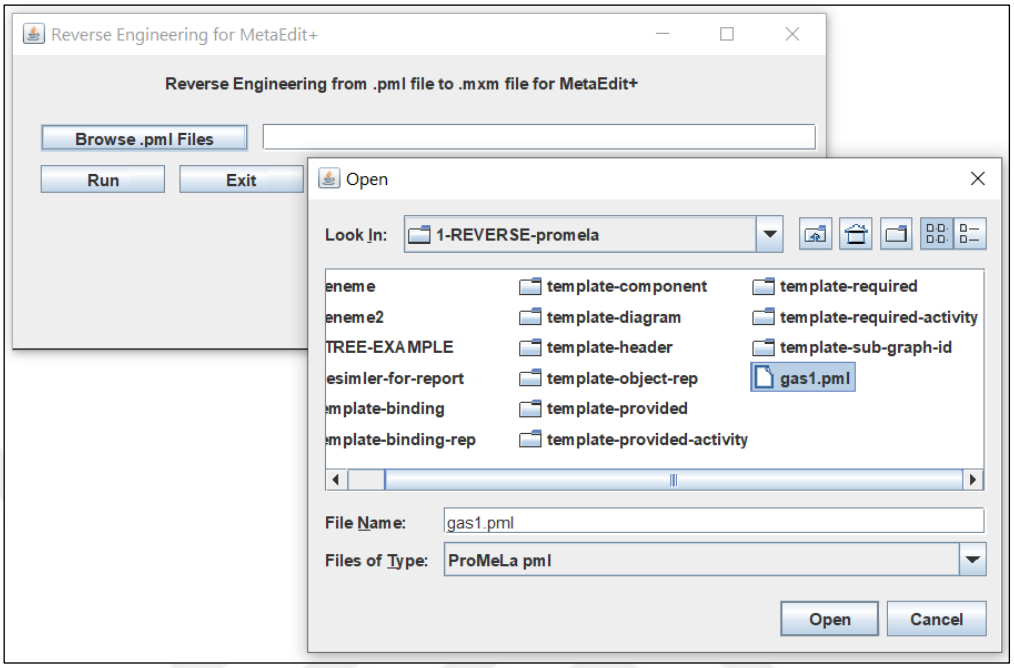
```
626
627     public void velocity1() throws Exception {
628             if (!responseWaiting) {
629                     responseWaiting = true;
630                     userinterfacedatastore_data_InterfaceIns.velocity1();
631             }
632             if (true) {
633                     velocity = result;
634             }
635             responseWaiting = false;
636     }
637
638     public int burnRate1() throws Exception {
639             if (true) {
640                     ;
641                     result = burnRate;
642             }
643             return result;
644     }
645
646     public void notify1(int state) throws Exception {
647             if (true) {
648                     currentSimState = state;
649             }
650     }
651
652     public void run() {
653             while (true) {
654                     try {
655                             height1();
656                             fuel1();
657                             simulationTime1();
658                             velocity1();
659                     } catch (Exception e) {
660                             e.printStackTrace();
661                     }
662             }
663     }
664 }
665
```

8. TERSİNE MÜHENDİSLİK İLE SAwUML OLUŞTURMA

Önceden MetaEdit+ editöründeki dönüştürücüyü kullanılarak oluşturulmuş bir .pml dosyasından, tersine dönüştürücüsü programı kullanarak .mxm uzantılı bir dosya elde edilir. Bu .mxm uzantılı dosya tekrar MetaEdit+ editörüne aktarılarak ilgili SAwUML modeli veya aktivite diyagramı ile genişletilmiş SAwUML modeli oluşturulmuş olur.

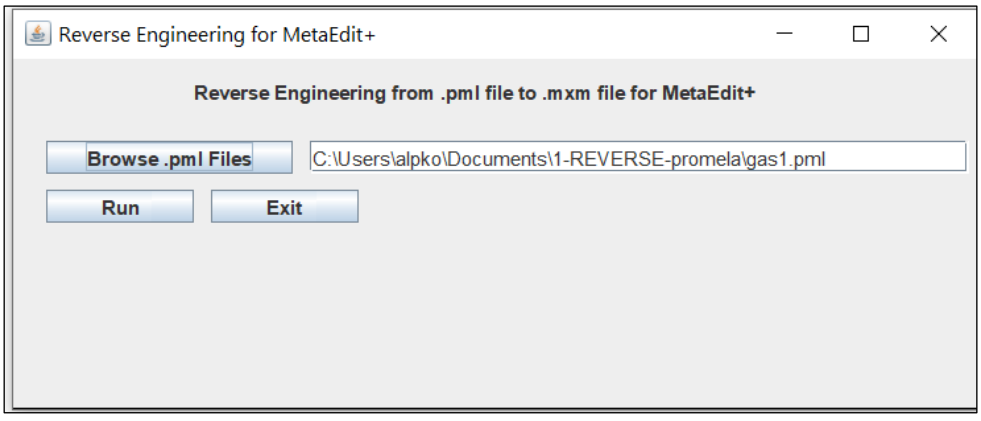
MetaEdit+ editöründe oluşturulan herhangi bir modelinden elde edilen .mxm uzantılı (MetaEdit+ deki .xml dosyasının karşılığı) dosyasında .xml uzantılı şablon dosyalar elde edilir. Java programlama dilinde geliştirilen tersine dönüştürücü programı çalıştırıldığında önce ilgili .pml uzantılı dosyayı satır satır okunur ve bilenşenler, bileşenlerin spesifikasyonları (dataları, metotları, metot parametreleri, ön/son şartlar), bileşenlerin bağlantı türleri (gerekli veya sağlayıcı bağlantı türleri), bağlandığı diğer bileşenler gibi verileri SQLite kullanılarak oluşturulan veri tabanına yazdırılır. Daha sonra, bu veriler veri tabanından okunup şablonların kopyalarının ilgili yerlerine yazdırılır ve verilerle yazdırılmış bu kopya şablonlar birleştirilerek yeni bir .mxm uzantılı dosya oluşturulur. Bu yeni .mxm uzantılı dosya MetaEdit+ editörüne yüklendiğinde, ilgili .pml dosyasından bu şekilde, bir aktivite diyagramı ile genişletilmiş SAwUML veya SAwUML modeli oluşturulmuş olur.

Tersine dönüştürücü programı çalıştırıldığında önce .pml programının seçilebileceği grafiksel kullanıcı arayüzü (GUI) çıkar (Şekil 8.1).



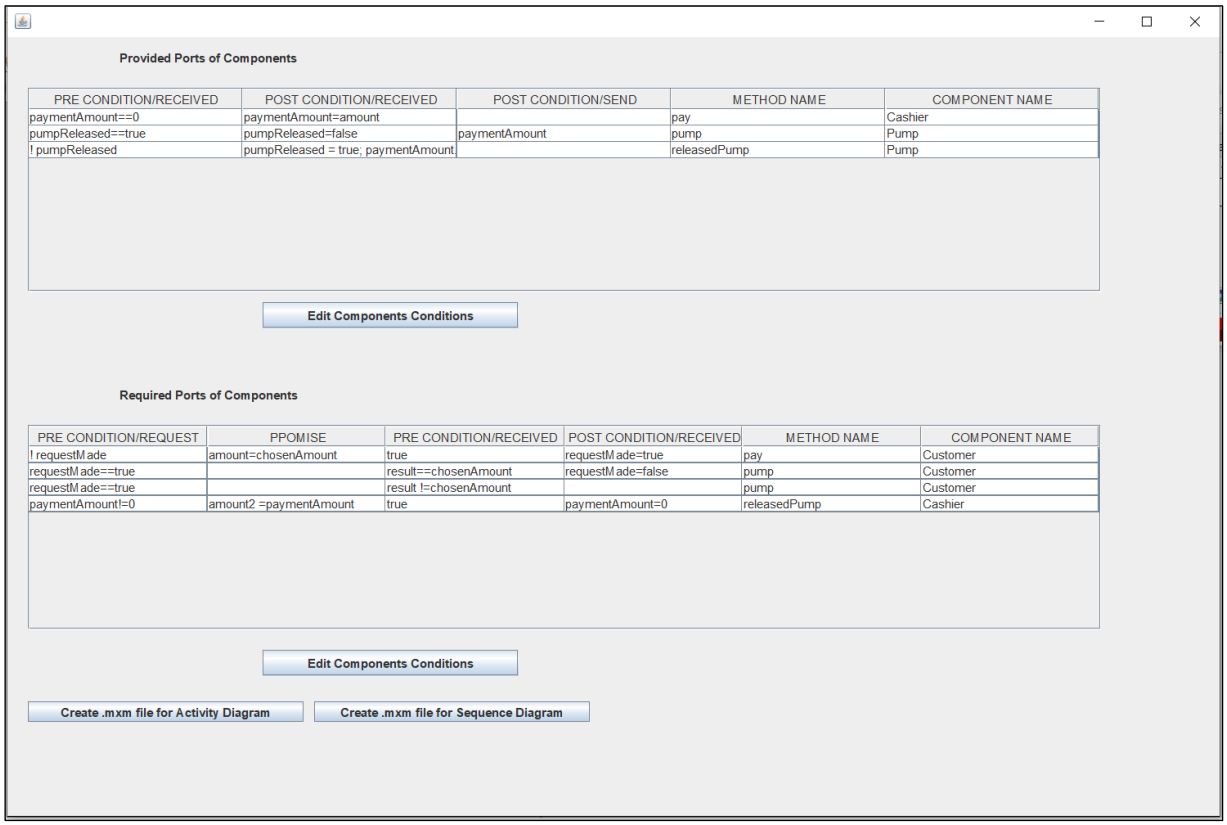
Şekil 8.1: Tersine model için tek müşterili gaz istasyon modelinin .pml dosyasının seçilimi.

Şekil 8.1'de gösterildiği gibi .pml uzantılı dosya seçildikten sonra kullanıcı "run" komutuna bastığı zaman ilgili .pml dosyasındaki tanımlı her bir bileşen ve o bileşenlerin sahip oldukları veriler ile özellikleri (eğer gerekli bağlantı noktaları varsa: ön şart, "promise" ve son şartları; eğer sağlayıcı bağlantı noktaları varsa: istek alma ön/son şartları ile varsa dönüş değeri) belirlenir (Şekil 8.2).



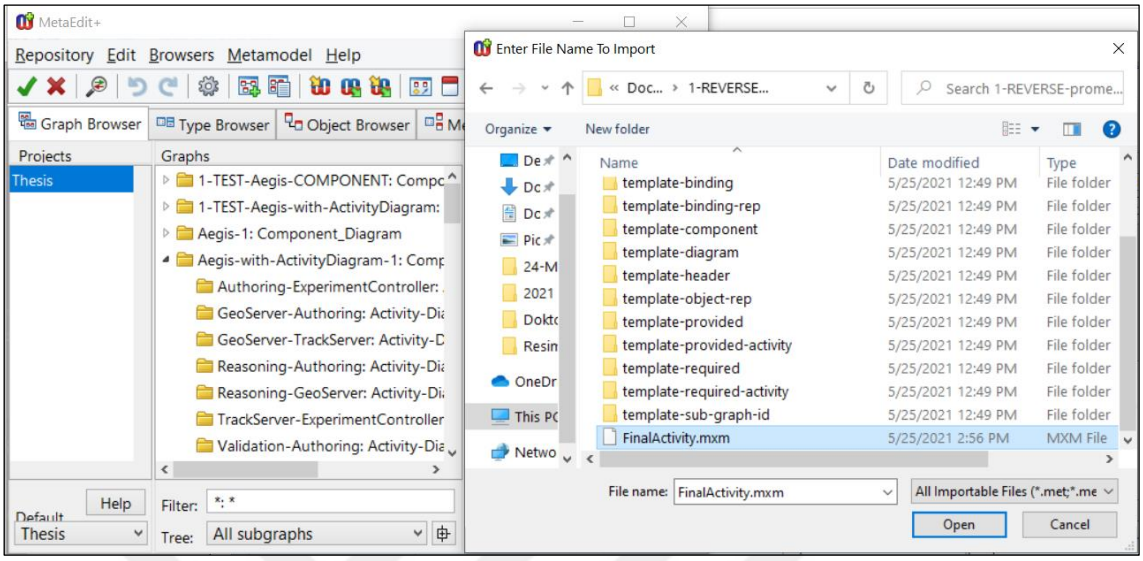
Şekil 8.2: Seçilen .pml uzantılı dosyanın çalıştırılması.

Kullanıcı, Şekil 8.3'de görüldüğü üzere SAwUML modeli (sıralama diyagramı olan model) veya aktivite diyagramı ile genişletilmiş SAwUML modeli oluşturmak için "Create .mxm file for Activity Diagram" veya "Create .mxm file for Sequence Diagram" isimli butonlardan birine basar.



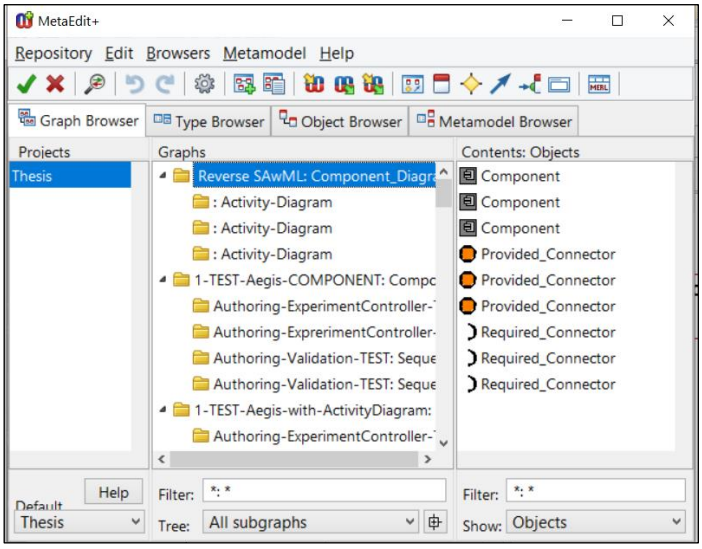
Şekil 8.3: Seçilen bir müşterili gaz istasyon modeline ait .pml uzantılı dosyanın çalıştırılması sonucu elde edilen bileşenler ve o bileşenlere ait özellikler.

Bu işlemin sonucu olarak Şekil 8.4'de görüldüğü gibi .mxm uzantılı dosya oluşur; ve bu dosya MetaEdit+ editörüne yüklendiğinde .pml dosyasından elde edilen model hazır olmuş olur.



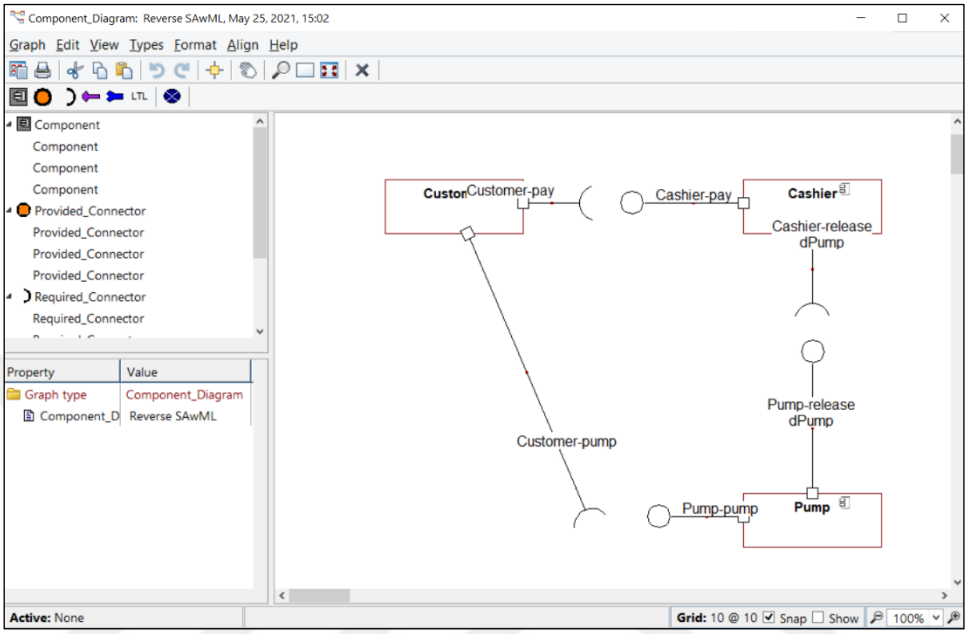
Şekil 8.4: .mxm uzantılı dosyanın elde edilmesi ve MetaEdit+ editörüne yüklenmesi.

"Create .mxm file for Activity Diagram" butonuna basılmasıyla elde edilen FinalActivity.mxm dosyasının (Şekil 8.4), MetaEdit+ editörüne yüklenmesi sonucu elde edilen modeli Şekil 8.5'da gösteriliyor.

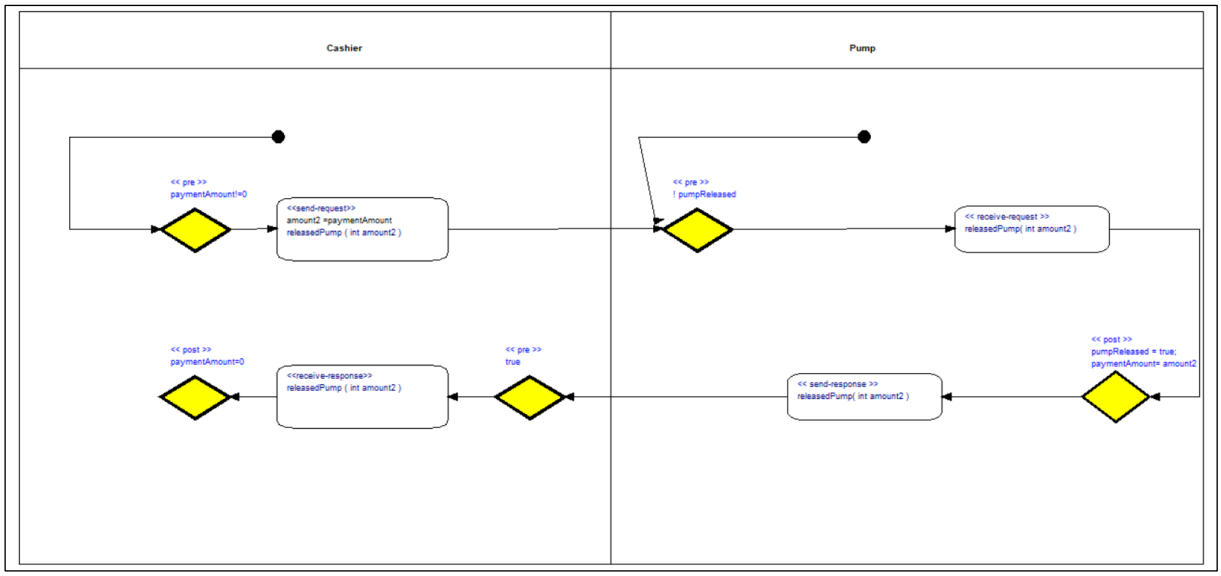


Şekil 8.5: Tersine programlama ile elde edilmiş, aktivite diyagramı ile genişletilmiş tek müşterili gaz istasyonu SAwUML modeli.

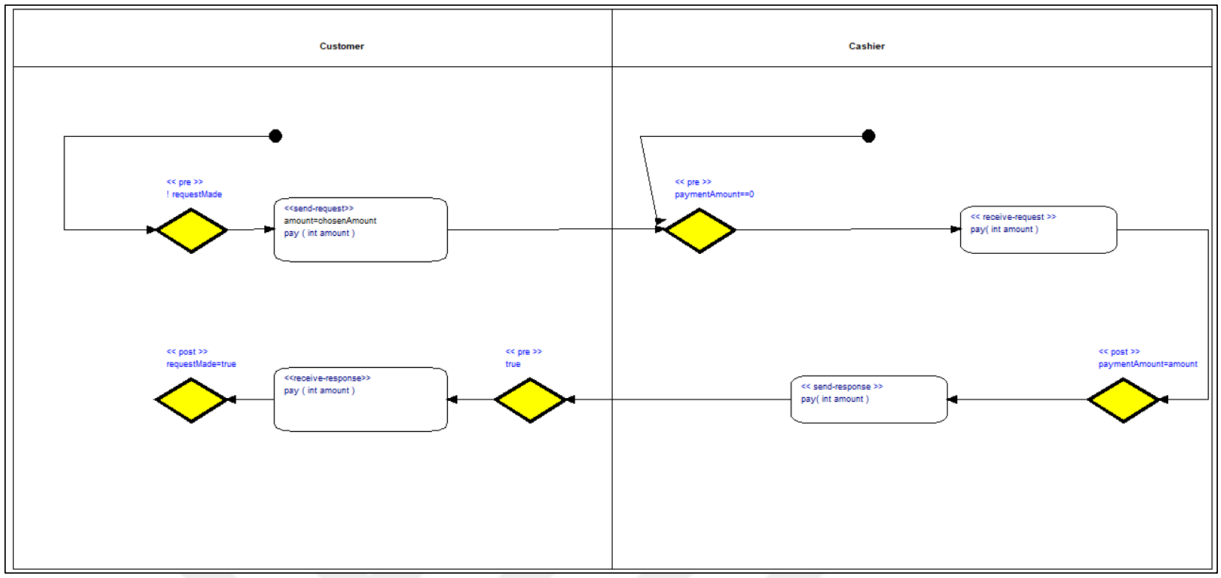
Tersine mühendislik ile elde edilmiş aktivite diyagramı ile genişletilmiş tek müşterili gaz istasyon SAwUML modelinin diyagramları aşağıda gösterilmektedir.



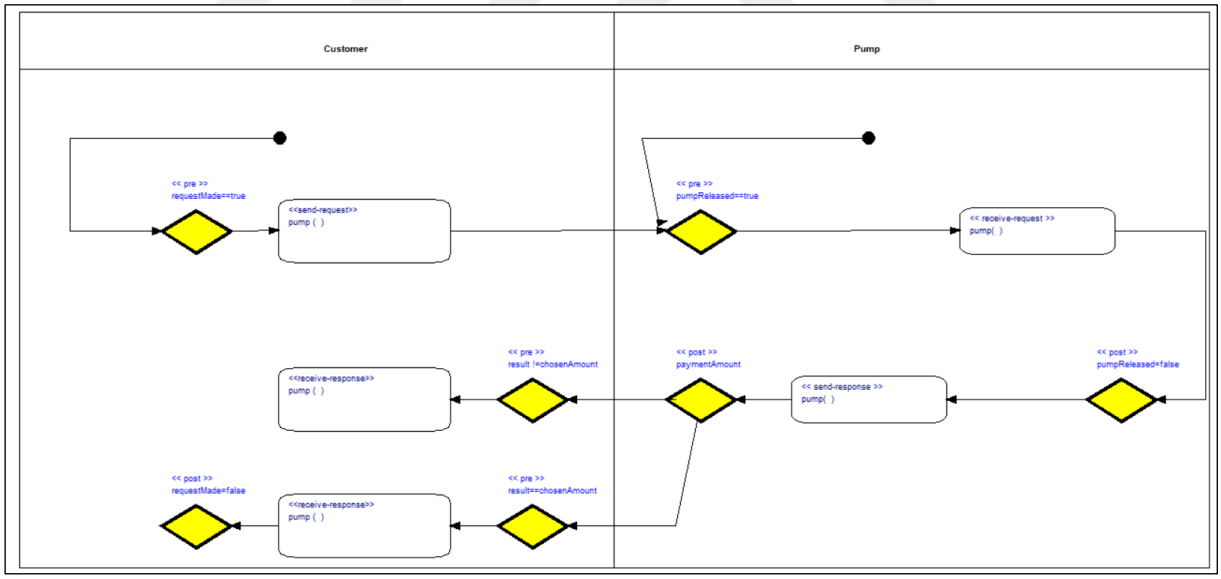
Şekil 8.6: Aktivite diyagramı ile genişletilmiş tek müşterili gaz istasyonu SAwUML modelinin ana grafiği.



Şekil 8.7: Tek müşterili gaz istasyonu SAwUML modelinin Cashier-Pump aktivite diyagramını gösteren alt grafiği.

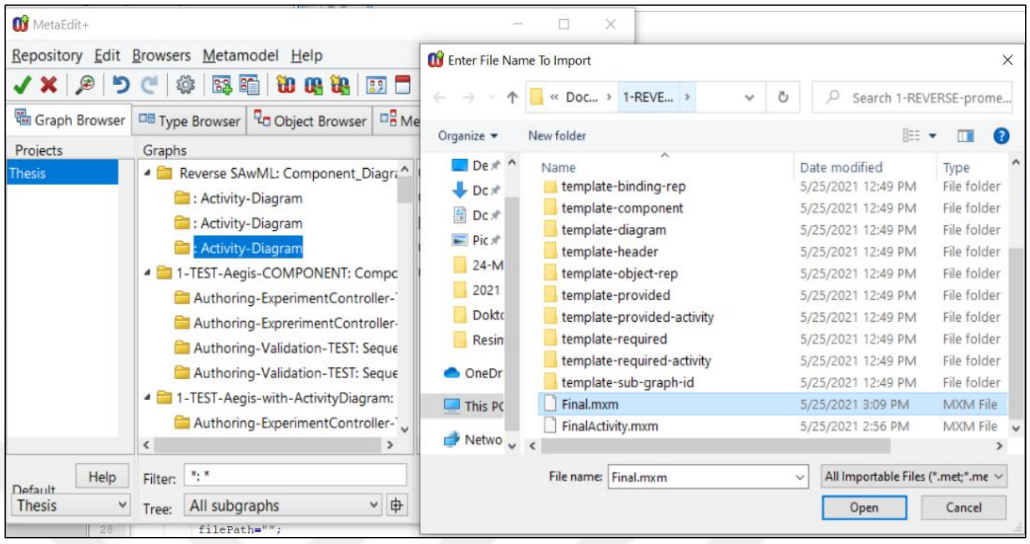


Şekil 8.8: Tek müşterili gaz istasyonu SAwUML modelinin Customer-Cashier aktivite diyagramını gösteren alt grafiği.

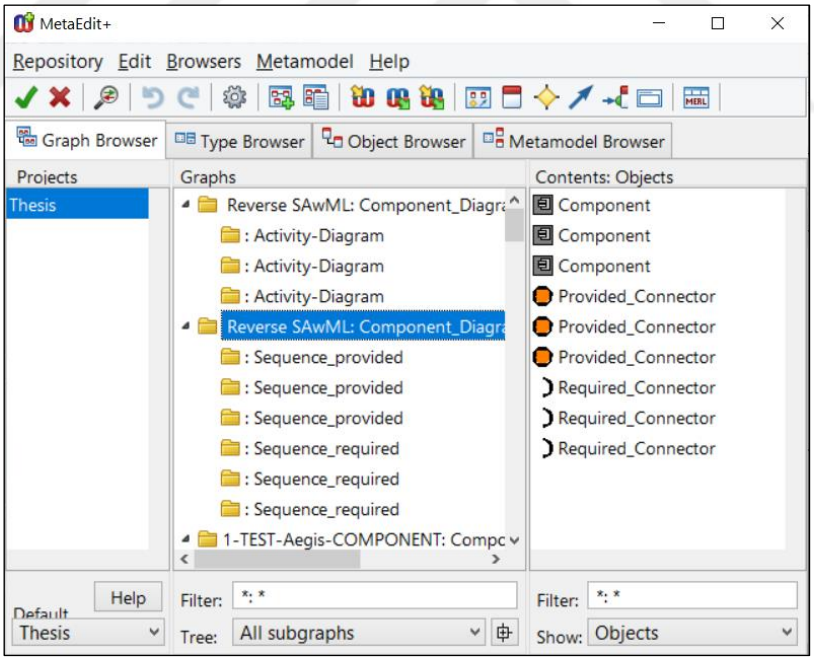


Şekil 8.9: Tek müşterili gaz istasyonu SAwUML modelinin Customer-Pump aktivite diyagramını gösteren alt grafiği.

"Create .mxm file for Sequence Diagram" butonuna basılmasıyla elde edilen Final.mxm dosyasının (Şekil 8.10), MetaEdit+ editörüne yüklenmesi sonucu elde edilen model Şekil 8.11'de gösteriliyor.

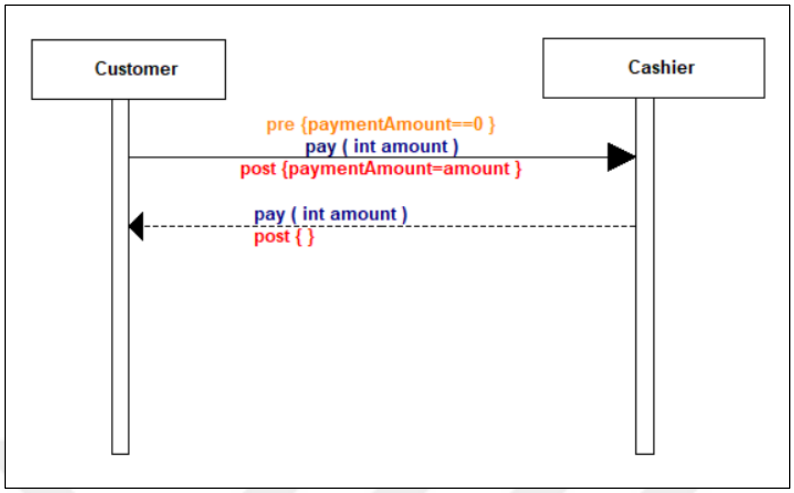


Şekil 8.10: Final.mxm uzantılı dosyanın elde edilmesi ve MetaEdit+ editörüne yüklenmesi.

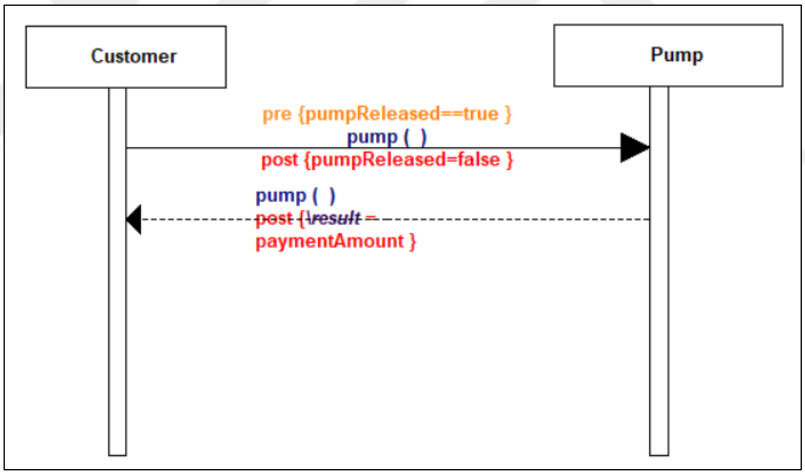


Şekil 8.11: Tersine programlama ile elde edilmiş, tek müşterili gaz istasyonu SAwUML modeli.

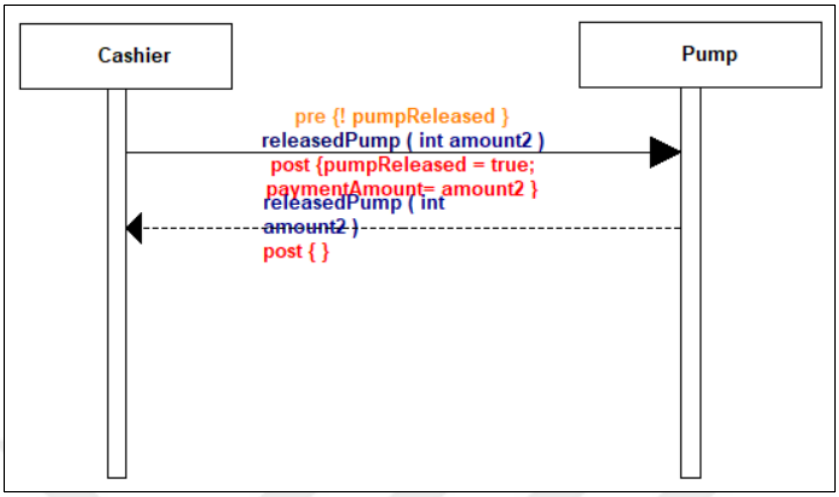
Tek müşterili gaz istasyonu SAwUML modelinin ana grafiği Şekil 8.6'da gösterilen diyagramla aynıdır. Bu modelin alt diyagramları aşağıda gösterilmiştir.



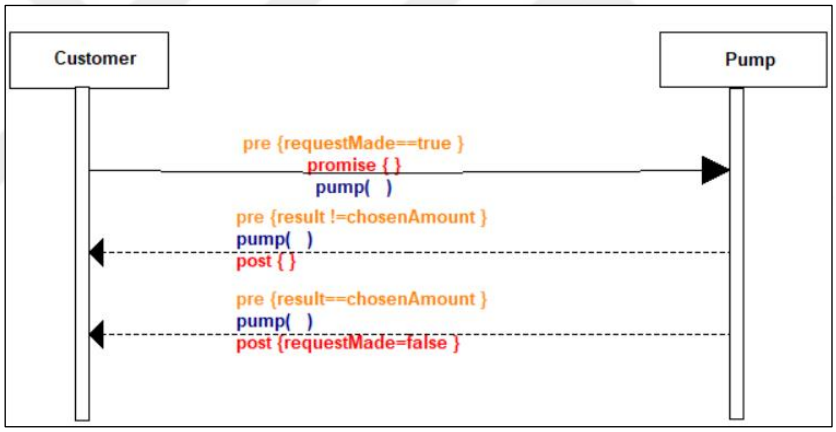
Şekil 8.12: *Cashier*'in *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri.



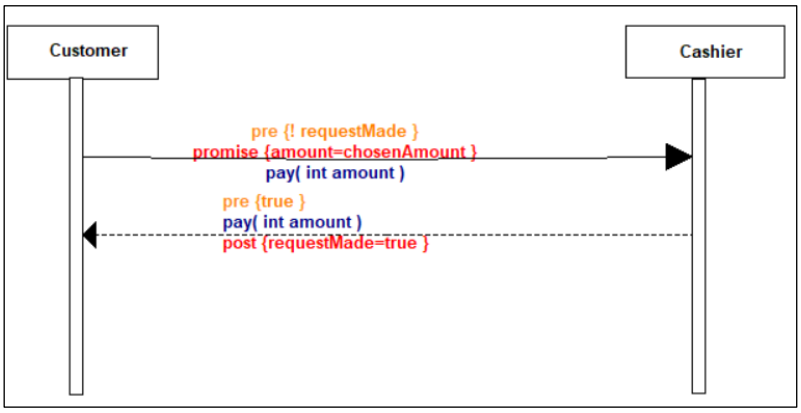
Şekil 8.13: *Pump*'ın *Customer*'a bağlandığı sağlayıcı portun (*provided port*) özellikleri.



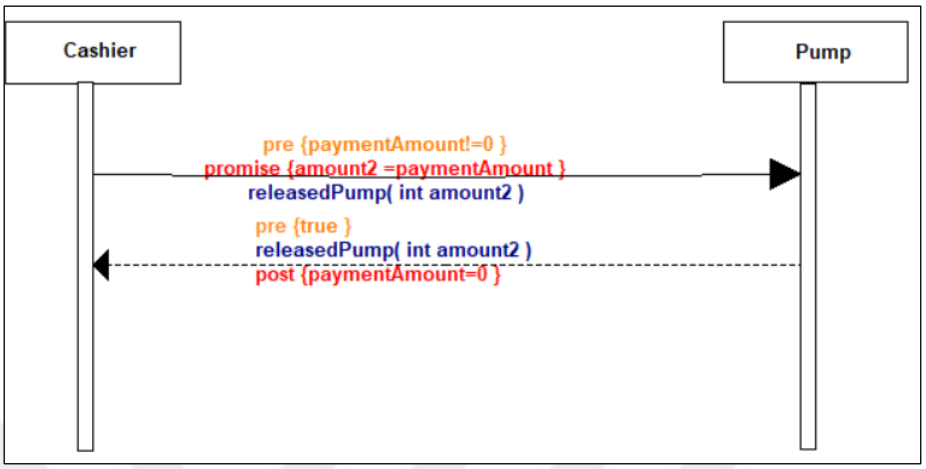
Şekil 8.14: *Pump*'ın *Cashier*'e bağlandığı sağlayıcı portun (*provided port*) özellikleri.



Şekil 8.15: *Customer*'ın *Pump*'a bağlandığı istemci portun (*required port*) özellikleri.



Şekil 8.16: *Customer*'ın *Cashier*'a bağlandığı istemci portun (*required port*) özellikleri.



Şekil 8.17: *Cashier*'in *Pump*'a bağlandığı istemci portun (*required port*) özellikleri.

9. DEĞERLENDİRME

Vaka analizlerinde oluşturulan ProMeLa dosyaları, SPIN model denetleyicinde kontrol edilmiş ve Tablo 9.1 ‘de gösterilen sonuçlar bulunmuştur. Model denetlemede, 2.2 GHz Intel Core i7-8750H, 16GB RAM2i olan ve Windows 10 Home işletim sistemli bilgisayar ile 6.4.5 SPIN versiyonu kullanılmıştır. Gaz istasyon modeli için *spin -a gas1.pml,* aya iniş modeli için *spin -a lunar.pml,* aegis muhabere sistem modeli için *spin -a aeg.pml* olmak üzere; daha sonra gaz istasyon modeli için *gcc -o pan pan.c,* ve *pan* komutları; aya iniş ve aegis muharebe sistem modelleri için *gcc -DBITSTATE -o pan pan.c ve pan -m5000000* komutları kullanılmıştır.

Tablo 9.1: SPIN Model Denetleyicisi Sonuçları.

Vaka Analizleri	**Durum-vektörü (State-vector) (bytes)**	**Durumlar (States)**		**Hafıza (Memory) (Mb)**	**Zaman (Time) (sn)**
		Kaydedilmiş (Stored)	**Eşlenmiş (Matched)**		
Gaz İstasyon Modeli	144	101	59	64.539	0
Aya İniş Modeli	216	53865248	53776075	309.958	139
Aegis Muharebe Sistem Modeli	244	63315191	102230970	1463.278	219

9.1 BİÇİMSEL SINAMA (FORMAL VERİFİCATION)

Tablo 9.1’de vaka analizleri için yapılan SPIN model denetleyicisi sonuçları gösteriliyor. SPIN model denetleyicisinde bir model test edilirken kilitlenme (*deadlock*) olduğu zaman, geçersiz durum hatası (*invalid end state error*) oluşur. Kilitlenme (*deadlock*), modelde çalışan her bir süreçin (*process*), kendi kodunun sonuna erişemediğini gösterir. Örneğin kilitlenme (*deadlock*), Gaz İstasyon Modeli örneğinde *Customer* bileşenin *pay* isimli portundaki *requestMade* datası *true* yapılırsa oluşur. Bu durumda, *Customer* bileşeninin hem *pay* isimli portu hem de *oil* isimli portunun ön şartları aynı olur. *Customer* bileşeninin, *Cashier* bileşeninden ödeme (*pay1*) veya *Pump* bileşeninden yakıt alınması (*pump1*) cevabını beklemesi, *Cashier* bileşeninin *Customer*

bileşeninden ödeme (*pay1*) isteğini; *Pump* bileşeninden yakıt verilme onayı cevabının (*releasePump1*) beklemesi; *Pump* bileşenin *Customer* bileşeninden yakıt alınma (*pump1*) isteğini ve *Cashier* bileşeninden yakıt verilme onayı (*releasePump1*) isteğinin beklemesi süresiz oluşacağı için kilitlenme (*deadlock*) oluşur.

Şekil 9.1'de görüldüğü gibi *Customer* bileşenin *pay* isimli portundaki r*equestMade* datası, *true* yapıldığında, SPIN model denetleyicisinde oluşan kilitlenme (*deadlock*) hatası gösterilmiştir.

```
1   ..........................
2        1:    proc  0 (configuration) gas1.pml:98 (state 4)    [(run Customer())]
3        2:    proc  0 (configuration) gas1.pml:99 (state 2)    [(run Cashier())]
4        3:    proc  0 (configuration) gas1.pml:100 (state 3)   [(run Pump())]
5      spin: trail ends after 3 steps
6      #processes 4:
7        3:    proc 0 (configuration)  gas1.pml:102 (state  5)
8                                    -end-
9        3:    proc 1 (Customer)  gas1.pml:20 (state 25) (invalid end state)
10                                   ((requestMade&&(responseWaiting==0)))
11                                   ch_pay_RES?pay1,result
12                                   ((requestMade&&(responseWaiting==0)))
13                                   ch_oil_RES?pump1,result
14       3:    proc 2 (Cashier)  gas1.pml:50 (state 21) (invalid end state)
15                                   (((paymentAmount1!=0)&&(responseWaiting==0)))
16                                   ch_releasePump_RES?releasePump1,result
17                                   ch_pay_REQ?pay1,amount2
18       3:    proc 3 (Pump)  gas1.pml:77 (state 18) (invalid end state)
19                                   ch_oil_REQ?pump1
20                                   ch_releasePump_REQ?releasePump1,amount2
21
```

Şekil 9.1: gas1.pml de SPIN model denetleyicide pan -r komutu kullanılarak tespit edilen kilitlenme (*deadlock*) hatasının gösterimi.

Modelde, eğer sağlayıcı portundan (*provided port*), gerekli portuna (*required port*) gelen yanıt, davranış özelliklerinden birisini içermiyorsa eksiklik hatası (*incompleteness error*) olur. Örneğin, gaz istasyonu modelinde *Customer* bileşeninin *Pump* bileşeninden gelen yanıt davranış özellikleri *result* değişkeninin *chosenAmount* değişkenine eşit olması veya olmaması durumudur. Eğer bu durumlardan birinin mevcut olmaması halinde (örneğin, *result == chosenAmount*), SPIN model denetleyici tarafından "*assertion violation error"* şeklinde ifade edilen eksiklik hatasını (*incompleteness error*) gösterir (Şekil 9.2 ve Şekil 9.3). Şekil 9.2'de gösterilen birinci satırdaki "*invalid end state*" ifadesi "*assertion violation error"* ifadesini belirtmektedir.

```
pan:1: invalid end state (at depth 22)
pan: wrote gas1.pml.trail

(Spin Version 6.4.5 -- 1 January 2016)
Warning: Search not completed
                + Partial Order Reduction

Full statespace search for:
                never claim             - (none specified)
                assertion violations    +
                acceptance   cycles     - (not selected)
                invalid end states      +

State-vector 144 byte, depth reached 23, errors: 1
           22 states, stored
                0 states, matched
           22 transitions (= stored+matched)
                2 atomic steps
hash conflicts:         0 (resolved)

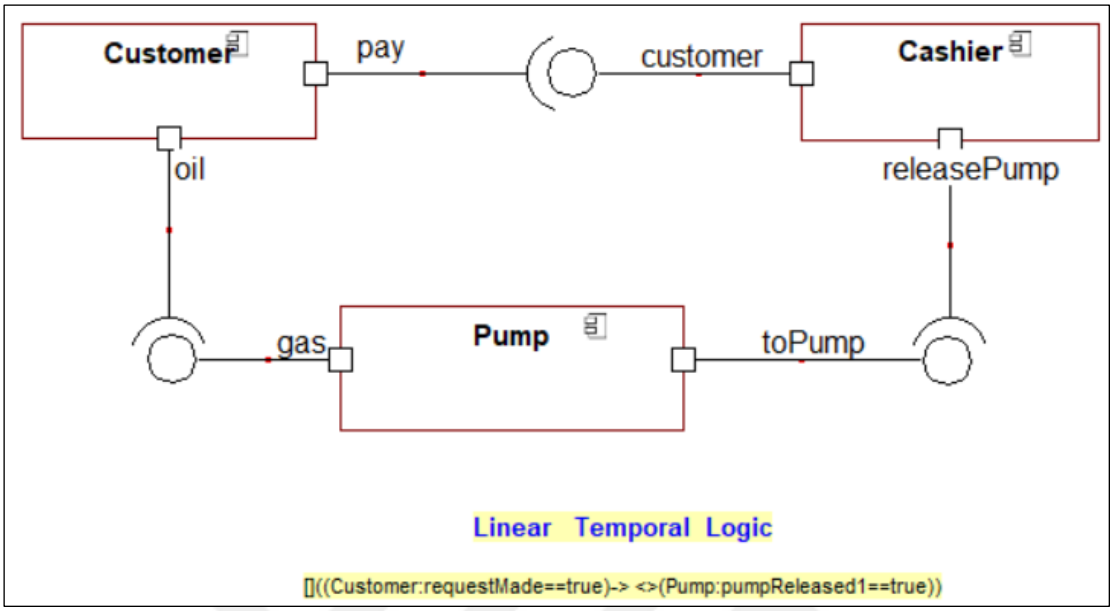
Stats on memory usage (in Megabytes):
        0.003       equivalent memory usage for states (stored*(State-vector + overhead))
        0.286       actual memory usage for states
   64.000       memory used for hash table (-w24)
        0.343       memory used for DFS stack (-m10000)
   64.539       total actual memory usage
```

Şekil 9.2: Gaz istasyon modelinde, SPIN model denetleyicisi tarafından pan komutu kullanılarak tespit edilen eksiklik hatasının (*incompleteness error*) gösterimi.

```
 ..................................
spin: trail ends after 23 steps
#processes 4:
 23:    proc 0 (configuration)  gas1.pml:102 (state  5)
                                -end-
 23:    proc 1 (Customer)  gas1.pml:36 (state 20) (invalid end state)
                                ((result!=chosenAmount))
                                (!(((result==chosenAmount)||(result!=chosenAmount))))
 23:    proc 2 (Cashier)  gas1.pml:50 (state 21) (invalid end state)
                                (((paymentAmount1!=0)&&(responseWaiting==0)))
                                ch_releasePump_RES?releasePump1,result
                                ch_pay_REQ?pay1,amount2
 23:    proc 3 (Pump)  gas1.pml:77 (state 18) (invalid end state)
                                ch_oil_REQ?pump1
                                ch_releasePump_REQ?releasePump1,amount21
```

Şekil 9.3: Gaz istasyon modelinde, SPIN model denetleyicisi tarafından pan -r komutu kullanılarak tespit edilen eksiklik hatasının (*incompleteness error*) detaylı gösterimi.

Kullanıcı editöründeki LTL ikonu kullanarak modele ilgili LTL özelliği Şekil 9.4' deki gibi ekleyebilir.



Şekil 9.4: Kullanıcı tarafından gaz istasyon modelinde eklenen LTL özelliği.

Model, dönüştürücüyle ProMeLa dilinde oluşturulan .pml dosyasına dönüştürüldüğünde editörde oluşturulan LTL özelliği de oluşturulan .pml dosyasına eklenmiş olur. Spin model denetleyicinde bu özellikler kontrol edilir. Şekil 9.5'de ve Şekil 9.6'da örnek olarak kullanıcı tarafından oluşturulan LTL özelliklerinin SPIN model denetleyicindeki sonuçları gösteriliyor. LTL özelliği hata oluşturduğunda SPIN model denetleyicisinde "*assertion violation error*" ifadesini olmaktadır.

```
1       >spin -a gas1.pml
2       'gcc-4' is not recognized as an internal or external command,
3       operable program or batch file.
4       'gcc-3' is not recognized as an internal or external command,
5       operable program or batch file.
6       ltl f1: [] ((! ((Customer:requestMade==1))) || (<> ((Pump:pumpReleased1==1))))
7
8       >gcc -DNOREDUCE -o pan pan.c
9
10      >pan
11      warning: never claim + accept labels requires -a flag to fully verify
12
13      (Spin Version 6.4.5 -- 1 January 2016)
14
15      Full statespace search for:
16                      never claim             + (f1)
17                      assertion violations    + (if within scope of claim)
18                      acceptance   cycles     - (not selected)
19                      invalid end states      - (disabled by never claim)
20
21      State-vector 152 byte, depth reached 159, errors: 0
22      ------------------------------------------------
```

Şekil 9.5: Gaz istasyon modelinde, hata vermeyen LTL özelliği.

```
1       >spin -a gas1.pml
2       'gcc-4' is not recognized as an internal or external command,
3       operable program or batch file.
4       'gcc-3' is not recognized as an internal or external command,
5       operable program or batch file.
6       ltl f1: [] ((! ((Customer:requestMade==1))) || ([] ((Pump:pumpReleased1==1))))
7
8       >gcc -DNOREDUCE -o pan pan.c
9
10      >pan
11      warning: never claim + accept labels requires -a flag to fully verify
12      pan:1: assertion violated  !( !((Pump.pumpReleased1==1))) (at depth 44)
13      pan: wrote gas1.pml.trail
14
15      (Spin Version 6.4.5 -- 1 January 2016)
16      Warning: Search not completed
17
18      Full statespace search for:
19                      never claim             + (f1)
20                      assertion violations    + (if within scope of claim)
21                      acceptance   cycles     - (not selected)
22                      invalid end states      - (disabled by never claim)
23
24      State-vector 152 byte, depth reached 44, errors: 1
25      ....................................................
```

Şekil 9.6: Gaz istasyon modelinde, hata veren bir LTL özelliği.

10. TARTIŞMA

Benzer çalışmalar bölümünde görüldüğü gibi, SAwUML mevcut çalışmalar içinde yapısal tasarım tercihlerinde bileşen diyagramı; davranışsal tasarım kararlarında sıralama (*sequence*) veya aktivite (*activity*) diyagramı kullanan tek modelleme dilidir.

Yapılan çalışmalarda yapısal tasarım için genelde sınıf (*class*) diyagramı tercih edilmiştir. Sınıf (*class*) diyagramı sistemin bileşenlerini göstermekle birlikte, bileşenler arasındaki iletişimi arayüzler vasıtasıyla (örneğin, istemci veya sağlayıcı portlar) olarak gösterilmemektedir. Sınıf (*class*) diyagramında bileşenlerin yaptıkları metot çağrıları net olarak ifade edilmemektedir. SAwUML'de ise kullanılan bileşen (*component*) diyagramı sayesinde istemci (*required*) port veya sağlayıcı (*provided*) port ayrımı yapılarak arayüzler net bir şekilde gösterilebilmekte, bileşenler arasındaki iletişim (senkron veya asenkron) belirlenebilmektedir. Arayüzlere sağ tıklayarak açılan alt editörlerde (tercihe göre) oluşturulmuş sıralama veya aktivite diyagramlarında bileşenlerin birbirleri ile yaptıkları iletişimi (mesaj gönderme, mesaj alma) gösterilmektedir.

Genelde yapılan çalışmalarda durum (*state-machine*) diyagramı kullanılmıştır. Durum diyagramları, sistem içindeki bileşenlerin meydana gelen olaylar ile (örneğin, metot çağrısı) durumları arasındaki geçişleri ifade eder. Karmaşık sistemleri, durum (*state machine*) diyagramları ile göstermek her zaman pratik olmamaktadır (çok sayıda durum ve geçiş olması durumunda). Ozkaya [12] çalışmasında durum (*state-machine*) diyagramları en az kullanılan davranışsal diyagram olduğu görülmüştür. Buna karşılık SAwUML'de davranışsal kararlar port metot düzeyinde alınır. Her bir port metot davranışı portun tipine göre (istemci veya sağlayıcı) semantiği belirli mesaj gönderme ve alma özellikler ile belirtilmiştir. Port metot davranışlarında, sonuç (*result*) ve parametre atanması da belirtilmiştir. Durum (state-machine) diyagramları, SAwUML'deki port davranışlarını ifade etmek için iyi bir tercih olmamaktadır.

Kontrata dayalı tasarım ile kullanıcı, metot gönderilmesi ve alınmasında için ön koşul şartını, metot alındıktan sonra son koşul şartını, metot gönderilmeden parametreye atama yapılması ile metot alındıktan sonra sonuç (*result*) atamasının yapılmasını belirtir. X-Man, bileşen davranışlarını belirlemek için kontrata dayalı tasarımı (*design by contract*) destekleyen başka

bir mimari dildir. X-Man, bileşen arayüzlerine eklenen ön ve son koşullar açısından kontratları kullanır ve arayüz hizmetlerinin ön koşulları karşılayan giriş parametreleriyle çağrılmasını sağlar ve son koşulu karşılayan sonuçları üretir. SAwUML'nin kontratlarından farklı olarak, X-Man'in kontratları, bileşen durumu üzerinden tanımlanamaz ve verilen bazı ön koşullarla değiştirilemez. SAwUML'de kullanıcılar pratik bir biçimde kontratları değiştirebilir alternatif kontratlar ekleyebilir.

SAwUML'de diğer birçok mimari dillerinde olduğu gibi biçimsel temeli olarak ProMeLa'yı kullanıyor. ProMeLa'nın notasyon seti C programlama dilini andırdığı için kullanıcı açısından öğrenilmesini ve kullanılmasını kolaydır. ProMeLa dilinde kullanılan değişken bildirimi, veri atama, eğer-aksi takdirde deyimleri (*if-else statements*), döngüler, kullanıcı açısından oldukça benzer gelmektedir. ProMeLa ayrıca, 1991'den beri aktif olarak araştırılan ve kullanımı ücretsiz olan ve son derece güçlü olan SPIN model denetleyicisi ile desteklenmektedir.

TTool gibi çok gelişmiş bir araç setine sahip olmamakla beraber, SAwUML, MetaEdit+ araç setini kullanarak geliştirdiği editör kullanıcılar için oldukça pratik ve kullanımı kolaydır.

ProMeLa kodu üzerinde yapılan değişikliklerin model seviyesinde tasarım kararlarını ihlal edip etmediğini SAWUML notasyon kümesi ile daha net olarak görülebilmesi nedeniyle, tersine mühendislik kullanılarak, ProMeLa kodunda, SAwUML model elde edilir.

11. SONUÇ

Bu çalışmada SAwUML isimli yeni bir mimari tanımlama dili geliştirilmiştir. Tasarımcıların, genelde yazılım mimarisi modelleri geliştirirken kullandıkları UML modelleme dili, SAwUML mimari tanımlama dilinin de esasını teşkil etmiştir. UML'deki yapısal ve davranışsal modelleri tek bir çatı altında toplayan SAwUML kullanıcıya, yazılım mimarisi modellemede kolaylık sağlamayı amaçlamıştır. SAwUML, yapısal modelde UML'de kullanılan bileşen diyagramdan (*component diagram*); davranışsal model için ise UML de kullanılan aktivite (*activity diagram*) ve sıralama (*sequence diagram*) diyagramlarından esinlenmiştir. Yapısal model için kullanılan bileşen diyagramı sayesinde bileşenlerin birbirleri ile farklı arayüz ile bağlanabilmeleri olanağını sağlamıştır. Davranışsal modellemede kullanılması içinse kullanıcıya sıralama veya aktivite diyagramı olmak üzere iki farklı alternatif sunulmuştur. SAwUML mimari tanımlama dili grafik tabanlı bir dil olarak tasarlandığı için bir grafik modelleme editörüne ihtiyaç duyar. Bu amaçla dilin geliştirilip kullanılması için MetaEdit+ isimli modelleme editöründen faydalanılmıştır. Yapısal model için geliştiren grafikteki her bir bileşenin arayüzünden o bileşen ve onun bağlantılı olduğu bileşenin davranışsal model diyagramına erişilir; böylece her iki model yapısı (yapısal ve davranışsal) bir arada gösterilmiş olur. Bileşenler arasındaki iletişimler için kontrata dayalı tasarım (*design by contract*) yaklaşımı kullanılmıştır. Bu tasarım metodu, SAwUML'de kullanılırken ön koşul şartı korunurken son koşul şartı yerine veri ataması özelliği getirilmiştir. Bu tasarım metodu, modelin biçimsel (*formal*) analizinin yapılmasını kolaylaştırmıştır. MetaEdit + editöründe ProMeLa ve java dilinde dönüştürücüler geliştirilmiştir. Tasarımcı, geliştirdiği modelin dönüştürücü sayesinde ProMeLa dilinde oluşturulmuş dosyasını oluşturabilir. Daha sonra bu dosyayı ProMeL tabanlı SPIN model denetleyicisinde kilitlenme (*deadlock*), eksiklik (*incompleteness*) gibi önceden tanımlanmış özellikleri veya kullanıcı tanımlı doğrusal zaman mantık (LTL) özelliklerini kullanarak test edebilir. Tasarımcı, aynı zamanda java dönüştürücüsünü kullanarak geliştirdiği modelden java iskelet kodu üretebilir ve bu koda yeni modüller ekleyerek mevcut kodu geliştirebilir.

SAwUML modelinden, MetaEdit+ editöründeki dönüştürücü sayesinde elde edilen bir ProMeLa dosyasından tersine mühendislik kullanılarak tekrar SAwUML modeli elde ediliyor. Tasarımcı, tersine mühendislik programı sayesinde alternatif davranışsal model (sıralı

diyagramdan aktivite diyagramı veya tam tersi) oluşturabiliyor. Ayrıca, tersine mühendislik programı ile tasarlanmış modelin ön/son koşullarında değişiklik yapılabilir. Tersine mühendislik programının bu avantajları ile, özellikle detaylı SAwUML modellerinde yapılabilecek ön/son şart değişiklikleri veya alternatif davranışsal model oluşturmada, oluşturulan modelden java kodunu oluşturabilmesi, tasarımcıya zaman ve emek açısından kolaylık sağlamış olur. ProMeLa modeli üzerinde yapılan değişikliklerin (örneğin, kontratların değiştirilmesi, farklı veri değişkeni kullanılması gibi) model seviyesinde tasarım kararlarını ihlal edip etmediğini belirlemeye çalışmak için de kullanabilir.

REFERANSLAR

[1] P. Clements, D. Garlan, R. Little, R. Nord and J. Stafford, (2003). "Documenting software architectures: views and beyond", 25th International Conference on Software Engineering, 2003. Proceedings, pp. 740-741

[2] Medvidovic, N., & Taylor, R. N. (2010). "Software architecture: foundations, theory, and practice", ACM/IEEE 32nd International Conference on Software Engineering, vol. 2, pp. 471-472.

[3] Garlan, D. & Shaw, M. (1993). "An introduction to software architecture", In Advances in software engineering and knowledge engineering. World Scientific, pp. 1–39.

[4] Ozkaya, M. (2018). "The analysis of architectural languages for the needs of practitioners", Software: Practice and Experience, vol.48, no. 5, pp. 985-1018.

[5] Medvidovic, N., & Taylor, R. N. (2000). "A classification and comparison framework for software architecture description languages", IEEE Transactions on software engineering, vol. 26, no. 1, pp. 70-93.

[6] Clements, P. C. (1996). "A survey of architecture description languages", Proceedings of the 8th international workshop on software specification and design, pp. 16-25.

[7] Malavolta, I., Lago, P., Muccini, H., Pelliccione, P., & Tang, A. (2012). "What industry needs from architectural languages: A survey", IEEE Transactions on Software Engineering, vol. 39, no. 6, pp. 869-891.

[8] Object Management Group (2015). OMG unified modeling language secification –version 2.5. [Online]. Available: http://www.omg.org/spec/UML/2.5/.

[9] Reggio, G., Leotta, M., & Ricca, F. (2014). "Who knows/uses what of the UML: A personal opinion survey", International Conference on Model Driven Engineering Languages and Systems, Springer, Cham, pp. 149-165.

[10] Grossman, M., Aronson, J. E., & McCarthy, R. V. (2005). “Does UML make the grade? Insights from the software development community”, Information and Software Technology, vol. 47, no. 6, pp. 383-397.

[11] Zeichick, A. (2002). “Modeling usage low; Developers confused about UML 2.0, MDA”, SD Times.

[12] Ozkaya, M. (2018). “Do the informal & formal software modeling notations satisfy practitioners for software architecture modeling?”, Information and Software Technology, vol. 95, pp. 15-33.

[13] Ozkaya, M., & Kose, M. A. (2018). “SAwUML–UML-based, contractual software architectures and their formal analysis using SPIN”, Computer Languages, Systems & Structures, vol. 54, pp. 71-94.

[14] Alhumaidan, F., & Zafar, N. A. (2014). “Possible improvements in UML behavior diagrams”, International Conference on Computational Science and Computational Intelligence, vol. 2, pp. 173-178.

[15] Meyer, B. (1992). “Applying 'design by contract' ”, Computer, vol. 25, no. 10, pp. 40-51.

[16] Hoare, C. A. R. (1969). “An axiomatic basis for computer programming”, Communications of the ACM, vol. 12, no. 10, pp. 576-580.

[17 Holzmann, G. J. (2004). “The SPIN Model Checker-Primer and Reference Manual”, Addison-Wesley.

[18] Lago, P., Malavolta, I., Muccini, H., Pelliccione, P., & Tang, A. (2014). “The road ahead for architectural languages”, IEEE Software, vol. 32, no.1, pp. 98-105.

[19] Poizat, P., & Royer, J. C. (2006). “A formal architectural description language based on symbolic transition systems and modal logic”, Journal of Universal Computer Science, vol. 12, no. 12, pp. 1741-1782.

[20] Jahanian, F., & Mok, A. K. (1994). “Modechart: A specification language for real-time systems”, IEEE Transactions on Software engineering, vol. 20, no. 12, pp. 933-947.

[21] Harel, D. (1987). “Statecharts: A visual formalism for complex systems”, Science of computer programming, vol. 8, no. 3, pp. 231-274.

[22] He, N., Kroening, D., Wahl, T., Lau, K. K., Taweel, F., Tran, C., Rümmer, P., & Sharma, S. (2012). “Component-based design and verification in X-MAN”, Embedded Real Time Software and Systems.

[23] Mohammad, M., & Alagar, V. (2011). “A formal approach for the specification and verification of trustworthy component-based systems”, Journal of Systems and Software, vol. 84, no. 1, pp. 77-104.

[24] Burger, E. J. (2013, June). “Flexible views for view-based model-driven development”, Proceedings of the 18th international doctoral symposium on Components and architecture, pp. 25-30.

[25] Larsen, K. G., Pettersson, P., & Yi, W. (1997). “UPPAAL in a nutshell”, International journal on software tools for technology transfer, vol. 1, no. 1, pp. 134-152.

[26] Pataricza, A., Majzik, I., Huszerl, G., & Várnai, G. (2003). “UML-based design and formal analysis of a safety-critical railway control software modüle”, Safety, vol. 48, pp. 48V.

[27] Cabot, J., Clarisó, R., & Riera, D. (2014). “On the verification of UML/OCL class diagrams using constraint programming”, Journal of Systems and Software, vol. 93, pp. 1-23.

[28] Apvrille, L., Courtiat, J. P., Lohr, C., & de Saqui-Sannes, P. (2004). “TURTLE: A real-time UML profile supported by a formal validation toolkit”, IEEE transactions on Software Engineering, vol. 30, no. 7, pp. 473-487.

[29] Courtiat, J. P., Santos, C. A., Lohr, C., & Outtaj, B. (2000). “Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique”, Computer Communications, vol. 23, no. 12, pp. 1104-1123.

[30] Kamel, M., & Leue, S. (2000). “Vip: A visual editor and compiler for v-promela”, International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, Berlin, Heidelberg, pp. 471-486.

[31] Selic, B. (1998). “Using UML for modeling complex real-time systems”, Languages, compilers, and tools for embedded systems, Springer, Berlin, Heidelberg, pp. 250-260.

[32] Bose, P. (1999). “Automated translation of UML models of architectures for verification and simulation using SPIN”, 14th IEEE International Conference on Automated Software Engineering, pp. 102-109.

[33] Siveroni, I., Zisman, A., & Spanoudakis, G. (2008). “Property specification and static verification of UML models”, 2008 Third International Conference on Availability, Reliability and Security, pp. 96-103.

[34] McUmber, W. E., & Cheng, B. H. (2001). “A general framework for formalizing UML with formal languages”, Software Engineering, International Conference on, pp. 433-442.

[35] Campbell, L. A., Cheng, B. H., McUmber, W. E., & Stirewalt, R. K. (2002). “Automatically detecting and visualising errors in UML diagrams”, Requirements Engineering, vol. *7*, no. 4, pp. 264-287.

[36] Beek, M. H. T., Gnesi, S., & Mazzanti, F. (2015). “From EU projects to a family of model checkers”, Software, Services, and Systems, Springer, Cham, pp. 312-328.

[37] Ter Beek, M. H., Fantechi, A., Gnesi, S., & Mazzanti, F. (2011). “A state/event-based model-checking approach for the analysis of abstract system properties”, Science of Computer Programming, vol. 76, no. 2, pp. 119-135.

[38] Lilius, J., & Paltor, I. P. (1999). “vUML: A tool for verifying UML models”, 14th IEEE international conference on automated software engineering, pp. 255-258.

[39] Kelly, S., Lyytinen, K., & Rossi, M. (1996). “Metaedit+ a fully configurable multi-user and multi-tool case and came environment”, International Conference on Advanced Information Systems Engineering, Springer, Berlin, Heidelberg, pp. 1-21.

[40] Kose, M. A., & Ozkaya, M. (2020). “Towards extending uml’s activity diagram for the architectural modeling, analysis, and implementation”, 15th conference on computer science and information systems (fedcsis), pp. 639-648.

[41] Naumovich, G., Avrunin, G. S., Clarke, L. A., & Osterweil, L. J. (1997). “Applying static analysis to software architectures”, Proceedings of the 6th European SOFTWARE ENGINEERING conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering, pp. 77-93.

[42] Allen, R., & Garlan, D. (1996). “A case study in architectural modeling: the AEGIS system”, Proceedings of the 8th International Workshop on Software Specification and Design, pp. 6-15.

[43] Medvidovic, N., & Taylor, R. N. (2010). “Software architecture: foundations, theory, and practice”, ACM/IEEE 32nd International Conference on Software Engineering, vol. 2, pp. 471-472.

EK A

SIRALAMA DİYAGRAMINDAN PROMELA DÖNÜŞTÜRÜCÜ KODU

SAwUML sıralamalı diyagramından (*sequence diagram*), ProMeLa dosyası elde etmek için MetaEdit+ editöründe MERL dilinde yazılan dönüştürücüsü aşağıda gösterilmektedir.

A.1: SAwUML_Sequence-toProMeLa.merl dosyası.

```
1  Report 'SAwUML_Sequence-to-ProMeLa'
2
3
4   to '%firstLine
5  /\s*(\S*)\s*=\s*(\S*)\s*/ $$1'
6  endto
7
8  to '%ArgumentwithType
9  /(.*)\s*\((.*)\)/ $$2'
10 endto
11
12 to '%firstArgument
13 /\s*(\S*)\s*(\S*)\s*/ $$2'
14 endto
15 newline;
16
17 to '%firstName
18 /(.*)\s*\((.*)\)/ $$1'
19 endto
20
21
22 $count_mytype='0';
23 foreach .Required_Connector{dowhile ~Required Interface { } @id=$count_mytype++; }
newline;
24
25 $re_count='0';
26 'mtype = {'
27      foreach .Required_Connector
28      {
29          do decompositions
30          {
31              foreach .LifeLine
32              {
33                   dowhile ~Message from>Sync_Call_required
34                   {
35                      dowhile:Service Name-2 { :Method\/Event  ',' } ','
36                   }
37               }
38           }
39
40          @id=$re_count++;
```

A.1: SAwUML_Sequence-toProMeLa.merl dosyası (devam).

```
41
42          if ($re_count<>$count_mytype) then
43              ','
44          endif
45      }
46   '}' newline;
47
48 foreach .Required_Connector {
49     'chan ch_'~Required Interface:Name'_REQ = ['~Required interface:buffer size'] of{
mtype '
50      $countMax='0';
51      $localMax='0';
52      do decompositions
53      {
54         foreach .LifeLine
55         {
56           if ( ~Message from>Sync_Call_required:Service Name-2:Parameters:Data type<>'')
then
57                      ','
58           endif
59         }
60         foreach .LifeLine
61         {
62             do ~Message from>Sync_Call_required
63             {
64                 dowhile:Service Name-2
65                 {
66                    $localMax='0';
67                    dowhile :Parameters
68                    {
69                        @dummy=$localMax++
70                        :Data type ',';
71                     }
72                     if ( $countMax< $localMax) then
73                          $countMax=$localMax;
74                     endif
75                  }
76                }
77             }
78        }
79       $countloop5='0';
80
81      ' };' newline;
82   } ; newline;
83
84   foreach .Required_Connector {'chan ch_'~Required Interface:Name'_RES= ['~Required
       interface:buffer size'] of{ mtype, int} ;' newline;}; newline;
85
86   $graphName=id newline ; /* name of the graph*/
87
88   foreach .Component
89   {
90
91     $ComponentName=:Type;
92     'proctype ' :Type '('dowhile:parameters{ :Data type ' ' :Parameter name ';'} ') {'
       newline;
93
94      ' '  do:Data List{ :Data type ' ' :Name; if ( :Array size<>'') then '[' :Array
                size;']' endif'=':Default value ';'; newline; }
95
```

A.1: SAwUML_Sequence-toProMeLa.merl dosyası (devam).

```
96      foreach .Required_Connector
97      {
98         $dummyServiceName='';
99         $ServiceNameControl='';
100        do decompositions
101        {
102           foreach .LifeLine
103           {
104               if ($ComponentName=:Class name) then
105                   do ~Message from>Sync_Call_required
106                   {
107                      $ServiceNameControl=:Service Name-2 ;
108                      dowhile:Service Name-2
109                      {
110                         if (:Parameters)  then
111                             do :Parameters
112                             {
113                                 :Data type ' ' :Parameter ';' newline;
114                             }
115                         endif
116                      }
117                   }
118               endif
119           }
120        }
121     }
122
123     foreach .Provided_Connector
124     {
125         $dummyServiceName='';
126        $ServiceNameControl='';
127        do decompositions
128        {
129           foreach .LifeLine
130           {
131              if ($ComponentName=:Class name) then
132
133                 do ~sync_Message>Sync_Call
134                 {
135                     $ServiceNameControl=:Service Name-2 ;
136                        dowhile:Service Name-2
137                        {
138                           if (:Parameters)  then
139                               do :Parameters
140                               {
141                                   :Data type ' ' :Parameter ';' newline;
142                               }
143                           endif
144                        }
145                 }
146              endif
147           }
148        }
149
150     }
151
152     if ( ~Component>Interface~Required interface.Required_Connector)  then
153           'int result; ' newline;
154     endif
```

A.1: SAwUML_Sequence-toProMeLa.merl dosyası (devam).

```
155
156      if (~Component>Interface~Required interface.Required_Connector) then
157            'int responseWaiting= false; ' newline;
158      endif
159
160      newline;
161      'do ' newline;
162
163       /***** IF PORT IS REQUIRED  ************/
164       foreach .Required_Connector
165       {
166         $count_req='1';
167         if  ( $ComponentName= ~Required interface>Interface~Component.Component:Type )
then
168
169              do decompositions
170              {
171                  foreach .LifeLine
172                  {
173                      if ($ComponentName=:Class name) then
174
175                         do ~Message from>Sync_Call_required
176                         {
177                             dowhile:Service Name-2
178                             {
179                                   $req_iteration=$count_req++;
180                             }
181                         }
182
183                      endif
184                  }
185               }
186
187               $countReq='0';
188               $alp_temp= ~Required Interface:Name;
189               $InterfaceName_rq= ~Required Interface:Name;
190
191               $countMax='0';
192               $localMax='0';
193
194               do decompositions
195               {
196                   foreach .LifeLine
197                   {
198                      if ($ComponentName=:Class name) then
199
200                         do ~Message from>Sync_Call_required
201                         {
202                            dowhile:Service Name-2
203                            {
204                                $localMax='0';
205                                dowhile :Parameters
206                                {
207                                    @dummy=$localMax++
208                                }
209                                if ( $countMax< $localMax) then
210                                    $countMax=$localMax;
211                                endif
212
213                            }
```

A.1: SAwUML_Sequence-toProMeLa.merl dosyası (devam).

```
214                         }
215
216                     endif
217                 }
218             }
219
220             _RequiredLoop();
221
222          endif
223
224
225       }
226       /***** END OF IF PORT IS REQUIRED  ******/
227
228       /***** IF PORT IS PROVIDED  ************/
229       newline;
230
231       foreach .Provided_Connector
232       {
233          $count_req='1';
234
235          if ($ComponentName = ~Provided interface>Interface~Component.Component:Type)
then
236
237              $dummyServiceName='';
238              $ServiceNameControl='';
239              $MetotName_temp='';
240
241              do decompositions
242              {
243                 foreach .LifeLine
244                 {
245                     if ($ComponentName=:Class name) then
246
247                         do ~sync_Message>Sync_Call
248                         {
249                            $ServiceNameControl=:Service Name-2 ;
250                            dowhile:Service Name-2
251                            {
252                               if ($ServiceNameControl<> $dummyServiceName ) then
253                                   $req_iteration=$count_req++;
254                                   $MetotName_temp=$ServiceNameControl;
255                               endif
256                               $dummyServiceName=$ServiceNameControl;
257                            }
258                         }
259
260                     endif
261                 }
262              }
263
264             do graphs
265             {
266               if id= $graphName  then
267                   foreach .Required_Connector
268                   {
269                     $RequiredInterfaceName_temp= ~Required Interface:Name;
270
271                     do decompositions
```

A.1: SAwUML_Sequence-toProMeLa.merl dosyası (devam).

```
272                     {
273                        foreach .LifeLine
274                        {
275                           do ~Message from>Sync_Call_required
276                           {
277                               if ( :Service Name-2 = $MetotName_temp ) then
278
279                                       $RequiredInterfaceName=
                                                $RequiredInterfaceName_temp;
280                                   endif
281                             }
282                           }
283                        }
284                     }
285                  endif
286               }
287               $countReq='0';
288
289               $countMax='0';
290               $localMax='0';
291
292               do decompositions
293               {
294                  foreach .LifeLine
295                  {
296                     if ($ComponentName=:Class name) then
297                          do ~sync_Message>Sync_Call
298                          {
299                                dowhile:Service Name-2
300                                {
301                                    $localMax='0';
302                                    dowhile :Parameters
303                                    {
304                                        @dummy=$localMax++
305                                    }
306                                    if ( $countMax< $localMax) then
307                                          $countMax=$localMax;
308                                    endif
309                                 }
310                          }
311                     endif
312                  }
313                }
314
315               _ProvidedLoop();
316
317          endif
318        }
319          /***** END OF IF PORT IS PROVIDED  ******/
320
321        'od ' newline;
322      '}' newline ;
323
324    }
325
326 'active proctype configuration()' newline;
327 '{' newline;
328        'atomic {' newline
329         foreach .Component { 'run ' :Type '('  dowhile:parameters{ :Value ',',} ');'
newline}
```

A.1: SAwUML_Sequence-toProMeLa.merl dosyası (devam).

```
330       ' }' newline ;
331
332       foreach .LTL
333       {
334          $countLTL='1';
335          do:linearTempLog
336          {
337            '/* ltl f'$countLTL' { ' :Name_Logic ' }' '*/' newline;
338             @dummy=$countLTL++;
339          }
340       }
341
342    '  }' newline;
343
344 endreport
```

A.2: _RequiredLoop.merl dosyası.

```
1  _RequiredLoop()
2
3   to '%firstName
4  /(.*)\s+(.*\s+.*)/ $$1'
5  endto
6
7  if ($countReq++ < $req_iteration)  then
8
9    $count_req='0';
10
11   do decompositions
12   {
13        foreach .LifeLine
14        {
15            if ($ComponentName=:Class name) then
16
17               do ~Message from>Sync_Call_required
18               {
19                              dowhile:Service Name-2
20                    {
21                        $dummy=$count_req++;
22
23                        if ($countReq=$count_req) then
24                            $MetotName_rq = :Method\/Event;
25                        endif
26                    }
27               }
28
29            endif
30         }
31      }
32
33    if    (  decompositions ) then
34          /*do nothing */
35
36    else
37        '::true && responseWaiting==false -> ' newline;
38        ' responseWaiting=true' newline;
39        ' ch_'$InterfaceName_rq'_REQ !'$MetotName_rq ';'; newline;
```

A.2: _RequiredLoop.merl dosyası (devam).

```
40        ' ::ch_'$InterfaceName_rq'_RES ?'$MetotName_rq',result->'; newline
41              '   if' newline;
42              '   ::(true)-> ;' newline;
43              '   :: result == -1-> assert(false);' newline;
44        '   fi ;' newline;
45        ' responseWaiting=false ; ' newline;
46     endif
47
48   do decompositions
49   {
50      foreach .LifeLine
51      {
52
53        if ($ComponentName=:Class name) then
54
55          do ~Message from>Sync_Call_required
56          {
57             if ( $MetotName_rq = :Service Name-2:Method\/Event ) then
58
59               if ( :pre Condition ) then
60                '::':pre Condition ' && responseWaiting==false ->' newline;
61                '   responseWaiting=true ;' newline;
62
63               else
64                ':: responseWaiting==false -> ' newline;
65                       '   responseWaiting=true;' newline;
66               endif
67               :promise ' ' newline;
68
69               $countloop9='1';
70
71               'ch_'$InterfaceName_rq'_REQ !'$MetotName_rq
72
73             if ( $countMax>'0') then
74
75              ',' dowhile:Service Name-2:Parameters{@dummy=$countloop9++;:Parameter ','}
'';
76             endif
77             if ( $countMax='0') then
78
79               ';'; newline;
80             endif
81          endif
82          }
83
84       endif
85      }
86    }
87
88   newline;
89   do decompositions
90   {
91      foreach .LifeLine
92      {
93
94         if ($ComponentName=:Class name) then
95
96           if ( ~result_Message >Result_required) then
97                '::ch_'$InterfaceName_rq'_RES ?'$MetotName_rq',result->'; newline
```

A.2: _RequiredLoop.merl dosyası (devam).

```
98                  'if' newline;
99              endif
100
101             $countResultMessage='0';
102             do ~result_Message >Result_required
103             {
104                if ( $MetotName_rq = :Service Name-2:Method\/Event) then
105                   @dummy=$countResultMessage++;
106
107                ':(':pre Condition')->':post Condition';' newline;
108
109                endif
110             }
111             if ( ~result_Message >Result_required) then
112
113                $testCountResultMessage='0';
114                '::!( '
115                dowhile ~result_Message >Result_required
116                {
117                   if ( $MetotName_rq = :Service Name-2:Method\/Event ) then
118                      @dummy=$testCountResultMessage++;
119                      :pre Condition
120                      if ( $testCountResultMessage<> $countResultMessage ) then
121                         if (:pre Condition<>'') then
122                            ' || '
123                         endif
124                      endif
125                   endif
126                }
127                ' )->assert(false); ' newline;
128
129                'fi ;' newline;
130                'responseWaiting=false ; ' newline;
131             endif
132          endif
133       }
134    }
135   _RequiredLoop()
136 endif
```

A.3: _ProvidedLoop.merl dosyası.

```
1  _ProvidedLoop()
2   $inner3='0';
3
4   if ($countReq++ < $req_iteration)  then
5       $count_req='0';
6
7       do decompositions
8       {
9           foreach .LifeLine
10          {
11             if ($ComponentName=:Class name) then
12
13                do ~sync_Message>Sync_Call
14                {
```

A.3: _ProvidedLoop.merl dosyası (devam).

```
15                  dowhile:Service Name-2
16                  {
17                     $dummy=$count_req++;
18                     if ($countReq=$count_req) then
19                          $MetotName_pr = :Method\/Event;
20                     endif
21
22                    }
23                 }
24
25              endif
26           }
27        }
28
29        do decompositions
30        {
31           foreach .LifeLine
32           {
33              if ($ComponentName=:Class name) then
34
35                do ~Message result>Result
36                {
37                     if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
38                         @id=$inner3++;
39                     endif
40                }
41
42                if ($inner3 ='0') then
43                    do ~sync_Message>Sync_Call
44                    {
45                       if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
46                            @id=$inner3++;
47                       endif
48                    }
49                endif
50
51              endif
52           }
53        }
54
55        $outer='1';
56        $countloop4='0';
57        _SubProvidedLoop-1();
58
59        $count4='1';
60        do decompositions
61        {
62
63           foreach .LifeLine
64           {
65
66             if ($ComponentName=:Class name) then
67
68                if ( ~sync_Message>Sync_Call ) then
69                   do ~sync_Message>Sync_Call
70                   {
71                     if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
72                         $out=$count4++ ;
73                      endif
```

A.3: _ProvidedLoop.merl dosyası (devam).

```
74                      }
75
76                     $count4='1';
77
78                     ' ::!( ' ;
79                    do ~sync_Message>Sync_Call
80                    {
81
82                        if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
83                             $dummy=$count4++;
84                             if ( $out='1' ) then
85                                '(':pre Condition  ') ';
86                             endif
87                            if ( $dummy< $out ) then
88                               '(':pre Condition  ') || ';
89                            endif
90                            if ($dummy = $out) and ($out>'1') then
91                               '(':pre Condition  ') ';
92                            endif
93
94                         endif
95
96                    }
97
98                    ' )->' newline;
99
100
101                   $countloop10='0';
102
103                   'ch_'$RequiredInterfaceName'_REQ !' $MetotName_pr
104                    if ( $countMax>'0') then
105                       do ~sync_Message>Sync_Call
106                       {
107                          if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
108                              dowhile:Service Name-2:Parameters{@dummy=$countloop10++;
','
                                    :Parameter } '';
109                          endif
110                       }
111                    endif
112                    if ( $countMax='0') then
113                        ';'; newline;
114                        'fi'; newline;
115
116                    endif
117
118                    $testdummy='';
119                    do ~sync_Message>Sync_Call
120                    {
121                      if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
122
123                           if ( $testdummy <> :Service Name-2:Method\/Event ) then
124                                newline;
125                                $testdummy=:Service Name-2:Method\/Event;
126                           endif
127                        endif
128                      }
129
130                  endif
```

A.3: _ProvidedLoop..merl dosyası (devam).

```
131
132             endif
133            }
134          }
135
136          $commaCheck='0';
137          _SubProvidedLoop-2();
138
139         newline;
140         if (  decompositions ) then
141              /*do nothing */
142         else
143          ' ::!( ' ;
144            '( true ) ';
145          ' )->' newline;
146              'ch_'$RequiredInterfaceName'_REQ !' $MetotName_pr ';'; newline;
147         'fi' newline;
148
149         endif
150
151        _ProvidedLoop();
152
153    endif
154
```

A.4: _SubProvidedLoop-1.merl dosyası.

```
1    _SubProvidedLoop-1()
2
3     $inner='1';
4     $inner2='1';
5
6     if ($countloop4++ <= $inner3)  then
7
8       if (  decompositions ) then
9           /*do nothing */
10      else
11
12         if ($countloop4++ = '1' ) then
13              '::ch_'$RequiredInterfaceName'_REQ ?'$MetotName_pr'->' newline;
14              'if' newline;
15         endif
16         '::(true )->' newline;
17
18         '::ch_'$RequiredInterfaceName'_RES !' $MetotName_pr ',result;'; newline
19
20
21
22      endif
23
24      do decompositions
25      {
26          foreach .LifeLine
27          {
28              if ($ComponentName=:Class name) then
```

A.4: _SubProvidedLoop-1.merl dosyası (devam).

```
29
30                if ( ~sync_Message>Sync_Call ) then
31                     if ($countloop4 = '1' ) then
32                         $countloop11='1';
33                          '::ch_'$RequiredInterfaceName'_REQ ?' $MetotName_pr
34                           if ( $countMax>'0') then
35
36                              do ~sync_Message>Sync_Call
37                              {
38                                if ( $MetotName_pr = :Service Name-2:Method\/Event )
                                   then
39                                   dowhile:Service Name-2:Parameters
                                           {@dummy=$countloop11++;',' :Parameter ','} '';
40
41                                endif
42                              }
43                           endif
44                           if ( $countMax='0') then
45                                  '->'; newline;
46                                  'if' newline;
47                           endif
48                           if ( $countMax>'0' and $countloop4 = $inner3) then
49                                  '->'; newline;
50                                  'if' newline;
51                           endif
52
53
54                      endif
55               endif
56            endif
57         }
58     }
59     $commaCheck='0';
60
61     _SubProvidedLoop-1();
62     newline;
63     do decompositions
64     {
65        foreach .LifeLine
66        {
67           if ($ComponentName=:Class name) then
68               do ~sync_Message>Sync_Call
69               {
70                    if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
71                           if ($inner = $outer) then
72                                  '::(':pre Condition ')->' newline;
73                                  :post Condition;  newline;
74                           endif
75                           @id=$inner++;
76                     endif
77               }
78               do ~Message result>Result
79               {
80                    if ( $MetotName_pr = :Service Name-2:Method\/Event ) then
81
82                      if ($inner2 = $outer) then
83                           if (:post Condition) then
84                              'ch_'$RequiredInterfaceName'_RES !' $MetotName_pr ','
```

A.4: _SubProvidedLoop-1.merl dosyası (devam).

```
                                    :post Condition ';'; newline;

                                else
                                 'ch_'$RequiredInterfaceName'_RES !' $MetotName_pr ',0;';
newline;
                                endif
                              endif
                              @id2=$inner2++;
                            endif
                        }
                        @id3=$outer++;

                endif
            }
        }

          _SubProvidedLoop-1();

      endif
```

A.5: _SubProvidedLoop-2.merl dosyası.

```
_SubProvidedLoop-2()

if ($countloop10 <= $countMax and $countMax >'0')  then

        if ( $countloop10< $countMax) then
            '-1,';
            $commaCheck='1';
        endif
        if ( $countloop10= $countMax and $commaCheck='1') then
            '-1' newline;
            'fi ' newline;

        endif
        if ( $countloop10= $countMax and $commaCheck='0') then
            newline;
            'fi ' newline;

        endif

   @dummy=$countloop10++;
   _SubProvidedLoop-2();
endif
```

EK B

AKTİVİTE DİYAGRAMINDAN PROMELA DÖNÜŞTÜRÜCÜ KODU

SAwUML aktivite diyagramından (*activity diagram*), ProMeLa dosyası elde etmek için MetaEdit+ editöründe MERL dilinde yazılan dönüştürücüsü aşağıda gösterilmektedir.

B.1: SAwUML_Activity-to-ProMeLa.merl dosyası.

```
1  Report 'SAwUML_Activity-to-ProMeLa'
2
3     $count_mytype='0';
4     foreach .Required_Connector{dowhile ~Required Interface { } @id=$count_mytype++; }
newline;
5
6     $re_count='0';
7
8      'mtype = {'
9        dowhile .Required_Connector
10       {
11
12          do decompositions
13          {
14              dowhile .ActivityRequest1
15              {
16                 :Service Name:Method\/Event ','
17               }
18                    }
19           ','
20       }
21    '}' newline;
22
23   foreach .Required_Connector
24   {
25         'chan ch_'~Required Interface:Name'_REQ = ['~Required interface:buffer size'] of
{
           mtype '
26         @comma='false';
27         do decompositions
28         {
29             foreach .ActivityRequest1
30             {
31                if ( :Service Name:Parameters<>'') then
32                     @comma='true';
33                endif
34             }
35             if (@comma='true') then
36                 ',';
37                 dowhile .ActivityRequest1
38                 {
39                    dowhile :Service Name
40                    {
41                        dowhile :Parameters
42                        {
43                           :Data type  ','
44                        }
45                     }
46                     ','
```

B.1: SAwUML_Activity-to-ProMeLa.merl dosyası (devam).

```
47                  }
48              endif
49          }
50          ' } ' newline;
51
52  }
53  foreach .Required_Connector
54  {
55     'chan ch_'~Required Interface:Name'_RES= ['~Required interface:buffer size'] of{
          mtype,  int} ;' newline;
56  }
57  newline;
58  foreach .Component
59  {
60     $componentName=:Type;
61     newline;
62    'proctype ' :Type '(' dowhile :parameters{ :Data type ' ' :Parameter name';'} ') {'
      newline;
63
64     do: Data List{ :Data type ' ' :Name; if ( :Array size<>'') then '[' :Array size;']'
                  Endif '=':Default value ';'; newline; }
65
66     if ( .Required_Connector) then
             'int responseWaiting= false; ' newline;
             'int result; ' newline;
67     endif
68     foreach .Required_Connector
69     {
70        do decompositions
71        {
72           if ( $componentName = .Lane_Partition:Required Component Name;) then
73               foreach .ActivityRequest1
74               {
75                  if (:Activity_streotype = 'send-request') then
76                      dowhile: Service Name
77                      {
78                         do: Parameters { :Data type '  ' :Parameter ';'; newline; }
79
80                      }
81
82                   endif
83               }
84
85            endif
86         }
87      }
88
89      foreach .Required_Connector
90      {
91          do decompositions
92          {
93             if ( $componentName = .Lane_Partition:Provided Component Name;) then
94                foreach .ActivitySend1
95                {
96                    $streotype=:Activity_streotype ;
97                    if ( $streotype = 'receive-request' ) then
98
99                       dowhile:Service Name
100                      {
101
102
```

B.1: SAwUML_Activity-to-ProMeLa.merl dosyası (devam).

```
103                         do:Parameters {:Data type '  ' :Parameter ';'; newline; }
104                     }
105                 endif
106             }
107
108         endif
109     }
110 }
111 'do ' newline;
112
113 foreach .Required_Connector
114 {
115    $nameChannel= ~Required Interface:Name;
116    do decompositions
117    {
118       if ( $componentName = .Lane_Partition:Required Component Name;) then
119
120          do .Start~From>Activity-Message~To.Activity_Diamond~From>Activity-
               Message~From>Activity-Message.ActivityRequest1
121          {
122            do  ~To>Activity-Message~From.Activity_Diamond
123            {
124               ' :: '
125               if (~To>Activity-Message~From.Start) then
126                    $isPreCond=:pre_or_post;
127                    $preCond=:condition;
128
129                    if ( $isPreCond ='pre' and $preCond<>'' ) then
130                           $preCond  '  && ';
131                    endif
132                endif
133                ' !responseWaiting -> ' newline;
134
135             }
136             $streotype =:Activity_streotype;
137             if ( $streotype= 'send-request') then
138                 'responseWaiting=true;' newline;
139                  if (:promise <>'') then
140                      :promise ';' newline;
141                  endif
142             endif
143
144             'ch_'$nameChannel'_REQ!'
145
146             if (:Activity_streotype = 'send-request') then
147                  dowhile :Service Name
148                  {
149                     id do :Parameters { if (:Parameter <>'') then ',' endif
                          :Parameter';'; newline; }
150                  }
151
152             endif
153
154             newline;
155             '::  ch_'$nameChannel'_RES?'
156
157             if (:Activity_streotype = 'send-request') then
158                    do :Service Name { id ',result -> ' }
159             endif
```

B.1: SAwUML_Activity-to-ProMeLa.merl dosyası (devam).

```
160
161                 newline;
162
163                 @ActivityName=:Service Name;:Method\/Event;
164                 'if ' newline;
165
166                 foreach .ActivityRequest2; where :Service
                        Name;:Method\/Event=@ActivityName
167                 {
168
169                    $streotype =:Activity_streotype ;
170                    if (  $streotype ='receive-response') then
171
172                       do ~To>Activity-Message~From.Activity_Diamond
173                       {
174                          ':: ( ' :condition ' )-> '
175                       }
176                       if not (~From>Activity-Message~To.Activity_Diamond) then
177                          ';'
178                       endif
179                       do  ~From>Activity-Message~To.Activity_Diamond
180                       {
181                           if (:condition <>'' ) then
182                               :condition ';'
183                           endif
184                        }
185                       newline;
186
187                    endif
188
189                 }
190                 ':: !  ('
191                 $count='0';
192                 foreach .ActivityRequest2; where :Service
                     Name;:Method\/Event=@ActivityName
193                 {
194                    if ( :Activity_streotype ='receive-response') then
195                          @dummy=$count++;
196                    endif
197                 }
198
199                 $count2='0';
200                 foreach .ActivityRequest2; where :Service
                       Name;:Method\/Event=@ActivityName
201                 {
202                    $streotype =:Activity_streotype ;
203                     if ( $streotype ='receive-response') then
204                           @dummy=$count2++;
205                           do ~To>Activity-Message~From.Activity_Diamond
206                            {
207                               :condition
208                           }
209                           if ( $count2 < $count) then
210                              ' ||'
211                           endif
212                     endif
213                  }
214                  ' )-> assert(false);' newline;
215                  'fi' newline;
```

B.1: SAwUML_Activity-to-ProMeLa.merl dosyası (devam).

```
216                    'responseWaiting = false; ' newline;
217
218
219                 }
220
221              endif
222           }
223        }
224
225        foreach .Required_Connector
226        {
227           $channelNameTest= ~Required interface>Interface~Component.Component:Type;
228           $channelNameTemp= ~Required Interface:Name;
229
230           do decompositions
231           {
232              if ( $channelNameTest = .Lane_Partition:Required Component Name;) and
                   ( $componentName = .Lane_Partition:Provided Component Name;) then
233                 $nameChannel = $channelNameTemp;
234
235                 foreach .ActivitySend1
236                 {
237
238                   $streotype =:Activity_streotype ;
239                   if ( $streotype ='receive-request') then
240                      ':: ch_'$nameChannel'_REQ ?'
241                       dowhile :Service Name
242                       {
243                         id do:Parameters { if (:Parameter <>'') then ',' endif
                             :Parameter ;}
244                       }
245                       ' -> '  newline;
246                       ' if ' newline;
247                       '   :: ( ' ~To>Activity-Message~From.Activity_Diamond:condition
                        ' )->'  newline;
248                       if ( ~From>Activity-Message~To.Activity_Diamond:condition <>'')
                          then
249                          ~From>Activity-Message~To.Activity_Diamond:condition '; '
                           newline;
250                       endif
251                       '  ch_'$nameChannel'_RES ! '
252
253                       dowhile :Service Name { id ; }
254                       ','
255                       $postResult='';
256                       if ( ~From>Activity-Message~To.Activity_Diamond~From>Activity-
Message~To.ActivitySend2~From>Activity-Message~To.Activity_Diamond:pre_or_post ='post')
                           then
257                          $postResult = ~From>Activity-
Message~To.Activity_Diamond~From>Activity-Message~To.ActivitySend2~From>Activity-
Message~To.Activity_Diamond:condition;
258                    endif
259
260                    if ( $postResult <> '') then
261                         $postResult ';'
262                    else
263                         '0' ';';
264                    endif
265                    newline;
```

B.1: SAwUML_Activity-to-ProMeLa.merl dosyası (devam).

```
266                         '  ::  !( ' ~To>Activity-Message~From.Activity_Diamond:condition
' )->'
                            newline;
267                         '      ch_'$nameChannel'_REQ ! '
268                      endif
269                      if ( $streotype ='receive-request') then
270                         dowhile :Service Name
271                         {
272                             id
273                              do :Parameters
274                              {
275                                 if (:Parameter <>'') then
276                                     ','
277                                 endif
278                                 :Parameter ;
279                               }
280                           }
281                       endif
282                       newline;
283                      'fi ; ' newline;
284                  }
285
286               endif
287            }
288
289         }
290
291         ' od ' newline;
292         '}' newline;
293
294   }
295   'active proctype configuration()' newline;
296   '{' newline;
297   'atomic {' newline
298    foreach .Component
299    {
300      'run ' :Type '('  dowhile :parameters { :Value ','} ');' newline;
301    }
302    ' }' newline ;
303
304    foreach .LTL
305    {
306        $countLTL='1';
307        do :linearTempLog
308        {
309           '/* ltl f'$countLTL' { ' :Name_Logic ' }' '*/' newline;
310            @dummy=$countLTL++;
311        }
312     }
313     ' }'
314  newline;
315 endreport
```

EK C

SIRALAMA DİYAGRAMINDAN JAVA DÖNÜŞTÜRÜCÜ KODU

SAwUML sıralamalı diyagramından (*sequence diagram*), java dosyası elde etmek için MetaEdit+ editöründe MERL dilinde yazılan dönüştürücüsü aşağıda gösterilmektedir.

C.1: SAwUML_Sequence-to-Java.merl dosyası.

```
1  Report 'SAwUML_Sequence-to-Java'
2
3       to
4          '%lower' newline; 'A-Z a-z'
5       endto
6
7       to '%resultTest
8       /(result)\s(.*)/ $$1
9       /(.*)(result)(.*)/ $$2'
10      endto
11
12      to '%equalTest
13      /=/ $=='
14      endto
15
16      to '%andTest
17      /;/ $&&'
18      endto
19
20   'public class Configuration {' newline;
21
22      foreach .Component
23      {
24        '    ' :Type ' ' :Type%lower'Ins' ';' newline;
25      }
26      newline;
27      foreach .Required_Connector
28      {
29          $alp1=:Type;
30          $portName = ~Required interface:Name;
31          do decompositions
32          {
33             do .LifeLine~Message from>Sync_Call_required~sync_Message.LifeLine
34             {
35                 $postName=:Class name;
36             }
37             do .LifeLine~Message result>Result_required~result_Message.LifeLine:Class
name;
38             {
39                $preName=:Class name ;
40             }
41             '    ' $preName$postName'_'$portName'_Adapter' '    '
               $preName%lower$postName%lower'_'$portName'_AdapterIns' ';' newline;
42
43             newline;
44           }
45
46      }
47      newline;
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
48      '  public Configuration() {' newline;
49      foreach .Component
50      {
51         '   '  :Type%lower'Ins' ' = new ' :Type '(' dowhile :parameters{  :Value ' ,' ;}
');'
           newline;
52      }
53
54      foreach .Required_Connector
55      {
56         $link=>Interface~Required interface:Name;
57         $portName= ~Required interface:Name;
58         'try {' newline;
59         do decompositions
60         {
61
62              do .LifeLine~Message from>Sync_Call_required~sync_Message.LifeLine
63              {
64                 $postName=:Class name;
65              }
66              do .LifeLine~Message result>Result_required~result_Message.LifeLine:Class
name;
67              {
68                  $preName=:Class name ;
69              }
70              $preName%lower$postName%lower'_'$portName'_AdapterIns' ' =  new '
                 $preName$postName'_'$portName'_Adapter ( ' $postName%lower'Ins ) ;' ;
                 newline;
71              $preName%lower'Ins.'
$preName%lower$postName%lower'_'$portName'_InterfaceIns =
                 '$preName%lower$postName%lower'_'$portName'_AdapterIns';  ' ; ' newline;
72              newline;
73
74         }
75         '}' newline;
76         'catch (Exception e){' newline;
77         '  e.printStackTrace();' newline;
78         '}' newline;
79
80      }
81
82      foreach.Component
83      {
84         if (~Component>Interface~Required interface:Name<>'') then
85                :Type%lower'Ins.start() ;'  newline;
86         endif
87
88      }
89      newline;
90      '} '  newline;
91
92      '  public static void main(String args[]) throws Exception {' newline;
93            '     new Configuration();' newline
94      '  }'
95      newline;
96      '}' newline
97
98      foreach .Required_Connector
99      {
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
100         $link=>Interface~Required interface:Name;
101         $portName= ~Required interface:Name;
102         do decompositions
103         {
104            do .LifeLine~Message from>Sync_Call_required~sync_Message.LifeLine
105            {
106                $postNameClass=:Class name;
107            }
108            do .LifeLine~Message result>Result_required~result_Message.LifeLine:Class
name;
109            {
110                $preNameClass=:Class name ;
111            }
112         }
113
114
115         ' interface  '  $preNameClass$postNameClass'_'$portName'_Interface {' newline;
116         $count='1';
117         do decompositions
118         {
119            $preName2='';
120            do .LifeLine~Message from>Sync_Call_required
121            {
122               $serviceName_fw=:Service Name-2;
123
124               do ~sync_Message.LifeLine~Message result>Result_required;
                      where:Service Name-2=$serviceName_fw
125               {
126                    $preName2= :pre Condition;
127               }
128
129               if ( $preName2%resultTest='result') then
130                    $returnValue='int';
131               else
132                   $returnValue='void';
133                endif
134
135                do :Service Name-2
136                {
137                     ' ' 'public ' $returnValue ' ' :Method\/Event '('
138                   dowhile :Parameters
139                   {
140                     :Data type ' ' :Parameter ','
141                   }
142                   ')  throws Exception;' newline;
143
144                }
145
146             }
147             '}' newline;
148
149          }
150
151    }
152
153    foreach .Required_Connector
154    {
155        $nameLink= ~Required interface:Name ;
156        $portName= ~Required interface:Name;
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
157         do decompositions
158         {
159            do .LifeLine~Message from>Sync_Call_required~sync_Message.LifeLine
160            {
161                $postNameClass=:Class name;
162            }
163           do .LifeLine~Message result>Result_required~result_Message.LifeLine;
164            {
165                $preNameClass=:Class name ;
166            }
167
168          }
169
170         $link=>Interface~Required interface:Name;
171         do decompositions
172         {
173             do .LifeLine~Message result>Result_required
174            {
175               $preName=:pre Condition ;
176               if ($preName%resultTest='result') then
177                   $returnValue='int';
178               else
179                   $returnValue='void';
180               endif
181             }
182
183         }
184         ' class  '   $preNameClass$postNameClass'_'$portName'_Adapter' '  implements  '
           $preNameClass$postNameClass'_'$portName'_Interface { ' newline;
185         '     ' $postNameClass '  ' $postNameClass%lower '_ins ;' newline;
186         '   ' 'public  '   $preNameClass$postNameClass'_'$portName'_Adapter' ' ( '
          $postNameClass '  '$postNameClass%lower '_arg )  throws Exception' newline;
187         '     {' newline;
188         '         '$postNameClass%lower'_ins = '$postNameClass%lower'_arg ; ' newline;
189         '     }' newline;
190
191         $count='1';
192         do decompositions
193         {
194            do .LifeLine~Message from>Sync_Call_required
195            {
196               $serviceName_fw=:Service Name-2;
197
198               do ~sync_Message.LifeLine~Message result>Result_required;
                  where:Service Name- 2=$serviceName_fw
199               {
200                   $preName2= :pre Condition;
201               }
202
203               if ( $preName2%resultTest='result') then
204                     $returnValue='int';
205               else
206                     $returnValue='void';
207               endif
208               '      public   ' $returnValue ' '
209               do :Service Name-2
210               {
211                  :Method\/Event ' ( ';
212                  dowhile :Parameters
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
213                     {
214                        :Data type; '  '
215                        :Parameter ',' ;
216                     }
217
218                     ' )' ;
219                  }
220                  if ($returnValue='int') then
221                       ' throws  Exception  {';
222                       newline;
223                       'return   '  $postNameClass%lower'_ins.'
224                       do :Service Name-2; where $serviceName_fw=:Method\/Event
225                       {
226                            :Method\/Event ' ( ';
227                            dowhile :Parameters
228                            {
229                               :Parameter ',' ;
230                            }
231                            ' ) ;'  newline;
232                        }
233                        '    }  ' newline;
234                   else
235                        '   {' newline;
236                        '        try { ' newline;
237                        '            ' $postNameClass%lower'_ins.'
238
239                        do :Service Name-2; where $serviceName_fw=:Method\/Event
240                        {
241                             :Method\/Event ' ( ';
242                             dowhile :Parameters
243                             {
244                                :Parameter ',' ;
245                             }
246                             ') ;'  newline;
247                         }
248
249                         '        } ' newline;
250                         '        catch (Exception e) ' newline;
251                         '        {  ' newline;
252                         '        e.printStackTrace(); ' newline;
253                         '        }  '  newline;
254                         '    }   ' newline;
255                    endif
256                  }
257
258           }
259
260       ' } ' newline;
261
262
263    }
264
265    foreach .Component
266    {
267        $thread ='false';
268        ' class  ' :Type  '  ';
269        do  ~Component>Interface~Required interface
270        {
271          if (:Name<>'') then
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
272             $thread ='true';
273          endif
274       }
275       if ( $thread='true') then
276         'extends Thread '
277       endif
278       '{'  newline;
279       do :Data List
280       {
281          if (:Data type = 'bool') then
282              :Data type'ean  ' :Name; if ( :Array size<>'') then '[' :Array size;']'
                   endif ' = ':Default value ' ;' newline;
283          else
284             :Data type' ' :Name; if ( :Array size<>'') then '[' :Array size;']'
                   endif ' = ':Default value ' ;' newline;
285          endif
286        }
287
288        $result='false';
289
290       if (:parameters<>'') then
291           do :parameters {  :Data type;'  ':Parameter name; ' ;' newline; } newline;
292           :Type ' ('dowhile:parameters{  :Data type;'  ':Parameter name; ' ,' ;} ') {'
               newline;
293           do :parameters{ 'this.':Parameter name;'=':Parameter name; ' ; ' newline; };
294           '}' newline;
295       endif
296
297       do ~Component>Interface~Required interface.Required_Connector
298       {
299          do decompositions
300          {
301                 do ~sync_Message.LifeLine~Message result>Result_required
302                 {
303                     $preName= :pre Condition
304                     if ($preName%resultTest='result') then
305                            $result='true';
306                     endif
307                 }
308              }
309           }
310
311          do ~Component>Interface~Provided interface.Provided_Connector
312          {
313               do decompositions
314               {
315                   do ~sync_Message.LifeLine~Message result>Result
316                   {
317                      $preName= :post Condition
318                       if ($preName <> '') then
319                           $result='true';
320                       endif
321                    }
322               }
323             }
324             if ($result='true') then
325              'private  int result = 0 ; ' newline;
326             endif
327
328
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
329       do ~Component>Interface~Required interface.Required_Connector
330       {
331           $portName = ~Required interface:Name;
332           do decompositions
333               {
334                   do .LifeLine~Message from>Sync_Call_required~sync_Message.LifeLine
335                   {
336                        $postNameClass=:Class name;
337                   }
338                   do .LifeLine~Message result>Result_required~result_Message.LifeLine;
339                   {
340                        $preNameClass=:Class name ;
341                   }
342               }
343               '   ' $preNameClass$postNameClass'_'$portName'_Interface' '   '
                  $preNameClass%lower$postNameClass%lower'_'$portName'_InterfaceIns' ' ;'
                  newline;
344        }
345
346        if ( ~Component>Interface~Required interface ) then
347            'private   boolean responseWaiting= false;' newline;
348        endif
349
350
351       do ~Component>Interface~Required interface.Required_Connector
352       {
353           $LinkInterface= ~Required interface:Name
354           $portName= ~Required interface:Name;
355           do decompositions
356               {
357                  do .LifeLine~Message from>Sync_Call_required~sync_Message.LifeLine
358                  {
359                      $postNameClass=:Class name;
360                  }
361                  do .LifeLine~Message result>Result_required~result_Message.LifeLine;
362                  {
363                      $preNameClass=:Class name ;
364                  }
365               }
366              do decompositions
367              {
368                  do .LifeLine~Message from>Sync_Call_required
369                  {
370                      $metotName=:Service Name-2
371                      dowhile :Service Name-2 { $parameters= :Parameters  ',' }
372                      $preCondition_sync= :pre Condition   newline;
373                      $promise_sync= :promise  newline;
374
375                      do ~sync_Message.LifeLine~Message result>Result_required
376                      {
377                           if ( $metotName=:Service Name-2 ) then
378                               $preName= :pre Condition
379                               if ($preName%resultTest='result') then
380                                    $returnValue='int';
381                               else
382                                    $returnValue='void';
383                               endif
384                           endif
385                    }
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
386
387                     if ($parameters='') then
388                         'public  ' $returnValue ' '$metotName '()  throws Exception
                            {'
                         newline;
389                     else
390                         'public  ' $returnValue ' '$metotName ' ('$parameters' )
                             throws Exception {' newline;
391                     endif
392                     if ( $preCondition_sync<>'') then
393                           ' if ( ' $preCondition_sync '&& !responseWaiting )  {'
                    newline;
394                     else
395                           ' if ( !responseWaiting )  {'   newline;
396                     endif
397                     '     responseWaiting=true; ' newline;
398                     if ( $promise_sync <>'') then
399                          '    ' $promise_sync ';'   newline;
400                     endif
401                     if ( $promise_sync ='') then
402                         /***** do nothing ***/
403                     endif
404                     if ($returnValue='int') then
405                           'result= '  ;
406                     endif
407                     $preNameClass%lower$postNameClass%lower'_'

                            $portName'_InterfaceIns.'$metotName' ('
408                      do :Service Name-2
409                      {
410                          dowhile  :Parameters
411                          {
412                                   :Parameter ','
413                          }
414                       }
415                      ' ) ;' newline;
416                       '}' newline;
417
418                     do ~sync_Message.LifeLine~Message result>Result_required
419                     {
420                        if ( $metotName=:Service Name-2 ) then
421                            ' if (':pre Condition ')'
422                            ' { ' newline;
423                            if ( :post Condition <>'') then
424                                 '    ' :post Condition ';'  newline;
425                            else
426                                 '    // do nothing  '  newline;
427                            endif
428                            ' }' newline;
429                        endif
430                     }
431                     '        responseWaiting=false; ' newline;
432                     if ($returnValue='int') then
433                          'return  result;  '  newline;
434                     endif
435                     '   }      //req endof' newline;
436
437                 }
438
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
439                 }
440         }
441         newline;
442
443         do ~Component>Interface~Provided interface.Provided_Connector
444         {
445             $LinkInterface= ~Required interface:Name
446             do decompositions
447                 {
448                     do .LifeLine~Message from>Sync_Call~sync_Message.LifeLine
449                      {
450                           $postNameClass=:Class name;
451                      }
452                      do .LifeLine~Message result>Result~result_Message.LifeLine;
453                      {
454                           $preNameClass=:Class name ;
455                      }
456                 }
457                 newline;
458                 do decompositions
459                 {
460                     do .LifeLine~Message from>Sync_Call
461                     {
462                         $metotName=:Service Name-2
463                         dowhile :Service Name-2 { $parameters= :Parameters ',' }
464                         $preCondition_sync= :pre Condition   newline;
465                         $promise_sync= :post Condition  newline;
466
467                         do ~sync_Message.LifeLine~Message result>Result
468                         {
469                              if ( $metotName=:Service Name-2 ) then
470                                 $preName= :post Condition
471                                 if ($preName <> '') then
472                                      $returnValue='int';
473                                 else
474                                     $returnValue='void';
475                                 endif
476                              endif
477                         }
478
479                         if ($parameters='') then
480                             'public  ' $returnValue ' '$metotName ' () throws Exception
{'
481                             newline;
482                         else
                                'public  ' $returnValue ' '$metotName ' ('$parameters')
                                throws Exception {' newline;
483                         endif
484
485                         ' if ( ' $preCondition_sync ') {'
486
487                         '    ' $promise_sync ';'
488
489                         do ~sync_Message.LifeLine~Message result>Result
490                         {
491                            if (:post Condition <> '') then
492                                'result=' :post Condition ';'  newline;
493                            else
494                                '   // do nothing '   newline;
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
495                         endif
496                       }
497                       ' } ' newline;
498
499                       if ($returnValue='int') then
500                           'return  result;  '  newline;
501                       endif
502
503                       '   }' '  //provided  endof ' newline;
504                     }
505
506               }
507     }
508
509     if (~Component>Interface~Required interface.Required_Connector) then
510
511             '    public void run() ' newline;
512             '              { ' newline;
513             '     while(true) { ' newline;
514             do ~Component>Interface~Required interface.Required_Connector
515             {
516                     '                       try ' newline;
517                     '                       {   ' newline;
518                     do decompositions
519                     {
520                             do .LifeLine~Message from>Sync_Call_required
521                             {
522                                     :Service Name-2 '('
523                                     dowhile :Service Name-2
524                                     {
525                                             $parameters= :Parameters
526                                             @count='1';
527                                             @countCheck='1';
528                                             do :Parameters:Data type
529                                             {
530                                               @dummy=@countCheck++;
531                                             }
532                                             do :Parameters:Data type
533                                             {
534                                                if ( :Data type='int') then
535                                                      '0';
536                                                       @dummy=@count++;
537                                                endif
538                                                if ( :Data type='bool') then
539                                                      'true';
540                                                       @dummy=@count++;
541                                                endif
542                                                if (@count<@countCheck ) then
543                                                      ','
544                                                endif
545
546                                             }
547
548                                     }
549                                     ' ) ;' newline;
550                              }
551                     }
552                     '        }   ' newline;
553                     '        catch (Exception e) ' newline;
```

C.1: SAwUML_Sequence-to-Java.merl dosyası (devam).

```
554                   '         {    ' newline;
555                   '            e.printStackTrace(); ' newline;
556                   '          } ' newline;
557
558            }
559
560            '      }' newline;
561
562            ' } '  newline;
563     endif
564
565 '} '
566  newline;
567  newline;
568 }
569
570 endreport
571
```

EK D

AKTİVİTE DİYAGRAMINDAN JAVA DÖNÜŞTÜRÜCÜ KODU

SAwUML aktivite diyagramından (*activity diagram*), java dosyası elde etmek için MetaEdit+ editöründe MERL dilinde yazılan dönüştürücüsü aşağıda gösterilmektedir.

D.1: SAwUML_Activity-to-Java.merl dosyası.

```
1  Report 'SAwUML_Activity-to-Java'
2
3
4  to
5     '%lower' newline; 'A-Z a-z'
6  endto
7
8  to '%resultTest
9  /(result)\s(.*)/ $$1
10 /(.*)(result)(.*)/ $$2'
11 endto
12
13 to '%equalTest
14 /=/ $=='
15 endto
16
17 to '%andTest
18 /;/ $&&'
19 endto
20
21
22
23
24 'public class Configuration {' newline;
25
26  foreach .Component{
27        '    ' :Type ' ' :Type%lower'Ins' ';' newline;
28  }
29  newline;
30  foreach .Required_Connector
31  {
32     $alp1=:Type;
33     $portName = ~Required interface:Name;
34     do decompositions
35     {
36       $preName=.Lane_Partition:Required Component Name;
37       $postName=.Lane_Partition:Provided Component Name;
38       '    ' $preName'_to_'$postName'_'$portName'_Required' '    '
          $preName%lower'_to_'$postName%lower'_'$portName ';' newline;
39
40        newline;
41      }
42   }
43
44   newline;
45   '  public Configuration() {' newline
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
46  foreach .Component{
47    '   '  :Type%lower'Ins' ' = new ' :Type '(' dowhile :parameters{ :Value ' ,' ;} ');'
      newline;
48  }
49
50
51
52  foreach .Required_Connector
53  {
54    $link=>Interface~Required interface:Name;
55    $portName = ~Required interface:Name;
56    'try {' newline;
57    do decompositions
58    {
59       $preName=.Lane_Partition:Required Component Name;
60       $postName=.Lane_Partition:Provided Component Name;
61
62       $preName%lower'_to_'$postName%lower'_'$portName ' =  new '
         $preName'_to_'$postName'_'$portName'_Required ( ' $postName%lower'Ins ) ;' ;
         newline;
63       $preName%lower'Ins.' $preName%lower'2'$postName%lower'_'$portName'_LinkInstance =
            '$preName%lower'_to_'$postName%lower'_'$portName;  ' ; ' newline;
64          newline;
65
66       }
67       '}' newline;
68       'catch (Exception e){' newline;
69       '  e.printStackTrace();' newline;
70       '}' newline;
71
72
73  }
74
75  foreach .Component
76  {
77     if (~Component>Interface~Required interface:Name<>'') then
78          :Type%lower'Ins.start() ;'  newline;
79    endif
80
81   }
82   newline;
83   '} '  newline;
84
85   '  public static void main(String args[]) throws Exception {' newline;
86   '     new Configuration();' newline;
87   '  }'
88   newline;
89   '}' newline;
90
91   foreach .Required_Connector
92   {
93     $link=>Interface~Required interface:Name;
94     $portName= ~Required interface:Name;
95     do decompositions
96     {
97       $preNameClass=.Lane_Partition:Required Component Name;
98       $postNameClass=.Lane_Partition:Provided Component Name;
99     }
100
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
101
102     ' interface  '  $preNameClass'2'$postNameClass'_'$portName'_Link {' newline;
103
104     do decompositions {
105       foreach .ActivityRequest1
106       {
107         $metotName = :Service Name;
108
109         foreach .ActivitySend2; where :Service Name=$metotName
110         {
111           $resultCheck='';
112           if ( ~From>Activity-Message~To.Activity_Diamond:pre_or_post='post' ) then
113               $resultCheck= ~From>Activity-Message~To.Activity_Diamond:condition
114           endif
115           if ($resultCheck<>'') then
116              $returnValue='int';
117           else
118              $returnValue='void';
119           endif
120         }
121         $streotype=:Activity_streotype;
122         if ( $streotype= 'send-request') then
123
124            do :Service Name
125            {
126               ' ' 'public  ' $returnValue '  ' :Method\/Event'('
127               dowhile :Parameters{
128                 :Data type ' ' :Parameter ','
129               }
130               ')  throws Exception;' newline;
131
132            }
133
134           endif
135         }
136         '}' newline;
137    }
138
139   }
140
141   foreach .Required_Connector {
142     $nameLink= ~Required interface:Name ;
143     $portName= ~Required interface:Name;
144     do decompositions
145     {
146        $preNameClass=.Lane_Partition:Required Component Name;
147        $postNameClass=.Lane_Partition:Provided Component Name;
148     }
149
150
151    $link=>Interface~Required interface:Name;
152
153     ' class  '   $preNameClass '_to_' $postNameClass'_'$portName'_Required  implements
'
        $preNameClass'2'$postNameClass'_'$portName'_Link { ' newline;
154     '     ' $postNameClass '  ' $postNameClass '_ins ;' newline;
155     '   ' 'public  '   $preNameClass '_to_' $postNameClass'_'$portName '_Required ( '
        $postNameClass '  '$postNameClass '_arg )  throws Exception' newline;
156     '     {' newline;
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
157     '          '$postNameClass'_ins = '$postNameClass'_arg ; ' newline;
158     '      }' newline;
159
160     $count='1';
161
162     do decompositions {
163       foreach .ActivityRequest1
164       {
165         $metotName = :Service Name;
166
167         foreach .ActivitySend2; where :Service Name=$metotName
168         {
169            $resultCheck='';
170            $returnValue='';
171            if ( ~From>Activity-Message~To.Activity_Diamond:pre_or_post='post' ) then
172               $resultCheck = ~From>Activity-Message~To.Activity_Diamond:condition
173
174            endif
175            if ($resultCheck<>'') then
176               $returnValue='int';
177            else
178               $returnValue='void'  newline;
179            endif
180          }
181          '      public    ' $returnValue ' '
182          $streotype=:Activity_streotype;
183          if ( $streotype= 'send-request') then
184            do :Service Name
185            {
186               :Method\/Event'('
187               dowhile :Parameters{
188                    :Data type; ' '
189                    :Parameter ',' ;
190               }
191               ') ' ;
192            }
193           endif
194
195          newline;
196
197          if ( $returnValue = 'int') then
198
199            ' throws  Exception  {';
200            newline;
201            'return   '  $postNameClass'_ins.'
202
203            $streotype=:Activity_streotype;
204            if ( $streotype= 'send-request') then
205                do :Service Name
206                {
207                   :Method\/Event' ( ';
208                   dowhile :Parameters {
209                      :Parameter ',' ;
210                   }
211                   ') ;'  newline;
212                }
213             endif
214
215             '    }  ' newline;
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
216
217          else
218           '   {' newline;
219           '        try { ' newline;
220           '            ' $postNameClass'_ins.'
221           $streotype=:Activity_streotype;
222           if ( $streotype= 'send-request') then
223               do :Service Name
224               {
225                  :Method\/Event' ( ';
226                  dowhile :Parameters {
227                     :Parameter ',' ;
228                  }
229                  ') ;'  newline;
230               }
231           endif
232           }' newline;
233           '        catch (Exception e) ' newline;
234           '        {  ' newline;
235           '           e.printStackTrace(); ' newline;
236           '        }  '  newline;
237           '    }  ' newline;
238          endif
239
240        }
241      }
242
243
244
245      ' } ' newline;
246
247  }
248
249
250  foreach .Component
251  {
252       $className=:Type;
253       $thread ='false';
254       ' class  ' :Type  '  ';
255       do  ~Component>Interface~Required interface {
256              if (:Name<>'') then
257                  $thread ='true';
258              endif
259      }
260      if ( $thread='true') then
261        'extends Thread '
262      endif
263      '{'         newline;
264      do :Data List {
265                 if (:Data type = 'bool') then
266                  :Data type'ean  ' :Name; if ( :Array size<>'') then '['
                    :Array size;']' endif ' = ':Default value ' ;' newline;
267                 else
268                  :Data type' ' :Name; if ( :Array size<>'') then '[' :Array size;']'
                       endif ' = ':Default value ' ;' newline;
269                 endif
270      }
271
272      $result='false';
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
273
274       if (:parameters<>'') then
275         do:parameters{  :Data type;'  ':Parameter name; ' ;' newline;} newline;
276         :Type ' ('dowhile:parameters{  :Data type;'  ':Parameter name; ' ,' ;} ') {'
            newline;
277          do:parameters{ 'this.':Parameter name;'=':Parameter name; ' ; ' newline;};
278         '}' newline;
279       endif
280
281       foreach .Required_Connector {
282         $portName = ~Required interface:Name;
283         do decompositions
284         {
285           if ( ($className = .Lane_Partition:Required Component Name;) or ($className =
                .Lane_Partition:Provided Component Name;) ) then
286              foreach .ActivityRequest2
287              {
288                  $streotype=:Activity_streotype;
289                  if ( $streotype= 'receive-response') then
290                      $resultCheck= ~To>Activity-
Message~From.Activity_Diamond:condition;
291                      if ($resultCheck%resultTest='result') then
292                              $result='true';
293                       endif
294                    endif
295              }
296              foreach .ActivitySend2
297              {
298                 $streotype=:Activity_streotype;
299                 if ( $streotype= 'send-response') then
300                      $resultCheck='';
301                      if (~From>Activity-Message~To.Activity_Diamond:pre_or_post='post')
then
302                           $resultCheck= ~From>Activity-
Message~To.Activity_Diamond:condition
303                      endif
304                      if ($resultCheck<>'') then
305                          $result='true';
306                      endif
307                 endif
308               }
309            endif
310         }
311     }
312
313     if ($result='true') then
314          'private  int result = 0 ; ' newline;
315     endif
316
317
318     foreach .Required_Connector
319     {
320        $portName= ~Required interface:Name
321        do decompositions
322        {
323          if ( ($className = .Lane_Partition:Required Component Name;)  ) then
324               $preNameClass=.Lane_Partition:Required Component Name;
325               $postNameClass=.Lane_Partition:Provided Component Name;
326                '   ' $preNameClass'2'$postNameClass'_'$portName'_Link' '   '
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
                $preNameClass%lower'2'$postNameClass%lower'_'$portName'_LinkInstance ;'
                newline;
327           endif
328         }
329       }
330
331     if ( ~Component>Interface~Required interface ) then
332         'private   boolean responseWaiting= false;' newline;
333     endif
334
335     do ~Component>Interface~Required interface.Required_Connector
336     {
337         $LinkInterface= ~Required interface:Name
338         $portName= ~Required interface:Name;
339         do decompositions
340         {
341               if ( ($className = .Lane_Partition:Required Component Name;)  ) then
342                   $preNameClass=.Lane_Partition:Required Component Name;
343                   $postNameClass=.Lane_Partition:Provided Component Name;
344                   foreach .ActivityRequest1
345                   {
346                      $metotName = :Service Name;
347                      $streotype=:Activity_streotype;
348
349
350                     foreach .ActivitySend2; where :Service Name = $metotName
351                     {
352                         $resultCheck='';
353                         if(~From>Activity-
Message~To.Activity_Diamond:pre_or_post='post')then
354                            $resultCheck= ~From>Activity-
Message~To.Activity_Diamond:condition
355                         endif
356                         if($resultCheck<>'') then
357                            $returnValue='int';
358                         else
359                           $returnValue='void';
360                         endif
361                      }
362
363                      $preCondition_sync='';
364                      $promise_sync='';
365                      $parameters='';
366                      if ( $streotype= 'send-request') then
367                       $metotName=:Service Name
368                       dowhile :Service Name{$parameters= :Parameters  ',' }
369                       dowhile :Service Name{dowhile :Parameters { $params =:Parameter
',','}}
370                       $preCondition_sync=~To>Activity-
Message~From.Activity_Diamond:condition;
                          newline;
371                       $promise_sync= :promise  newline;
372                      endif
373
374
375                      if ($parameters='') then
376                         'public  '
377                         $returnValue ' '$metotName '()  throws Exception { '
                           newline;
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
378                     else
379                       'public  ' $returnValue ' '$metotName ' ('$parameters' )
                           throws Exception { '    newline;
380                     endif
381                     if ( $preCondition_sync <>'') then
382                          ' if ( ' $preCondition_sync '&& !responseWaiting )  {'
                           newline;
383                     else
384                          ' if ( !responseWaiting )  {'  newline;
385                     endif
386                     '     responseWaiting=true; ' newline;
387                     if ( $promise_sync <>'') then
388                        '    ' $promise_sync ';'   newline;
389                     endif
390                     if ( $promise_sync ='') then
391                         /***** do nothing ***/   newline;
392                     endif
393                     if ($returnValue='int') then
394                        'result= '  ;
395                     endif
396
397                     $preNameClass%lower'2'$postNameClass%lower'_'
                        $portName'_LinkInstance.'$metotName' ('
398
399                     $streotype=:Activity_streotype;
400                     if ( $streotype= 'send-request') then
401                            dowhile :Service Name { dowhile :Parameters {  :Parameter
','}  }
402                     endif
403
404
405                     ' ) ;' newline;
406                     '}' newline;
407                     foreach .ActivityRequest2 ; where :Service Name = $metotName
408                     {
409                       $streotype=:Activity_streotype;
410                       if ( $streotype= 'receive-response') then
411                          ' if ( '~To>Activity-Message~From.Activity_Diamond:condition
')'
412                          ' { ' newline;
413                          $postCondition =
                                ~From>Activity-Message~To.Activity_Diamond:condition;
414                          if ( $postCondition <>'') then
415                              '    ' $postCondition ';'
416                          endif
417                          ' }' newline;
418                        endif
419                      }
420
421                      '        responseWaiting=false; ' newline;
422                      if ($returnValue='int') then
423                          'return  result;  '  newline;
424                      endif
425                      '   }      ' newline;
426                   }
427               endif
428
429
430           }
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
431
432   }
433
434
435   foreach .Required_Connector
436   {
437     $channelNameTest= ~Required interface>Interface~Component.Component:Type;
438     $channelNameTemp= ~Required Interface:Name;
439     $LinkInterface= ~Required interface:Name
440     do decompositions
441     {
442       if ( $channelNameTest = .Lane_Partition:Required Component Name;) and
             ( $className = .Lane_Partition:Provided Component Name;) then
443           $postNameClass = .Lane_Partition:Provided Component Name;
444           $preNameClass = .Lane_Partition:Required Component Name;
445
446
447           foreach .ActivitySend1
448           {
449               $metotName = :Service Name;
450               foreach .ActivitySend2; where :Service Name=$metotName
451               {
452                 $streotype = :Activity_streotype;
453                 $postResult='';
454                 $preName='';
455                 $returnValue='';
456                 $preCondition_sync='';
457                 $promise_sync='';
458                 if ( $streotype = 'send-response') then
459                    $metotName=:Service Name ;
460                    dowhile :Service Name { $parameters= :Parameters ',' }
461                    $preName= '';
462                    if (~From>Activity-Message~To.Activity_Diamond:pre_or_post
='post')then
463                        $preName='true';
464                        if ( $preName ='true') then
465                          $postResult= ~From>Activity-
Message~To.Activity_Diamond:condition
466                        endif
467                    endif
468                    if ($postResult <> '') then
469                       $returnValue = 'int';
470                    else
471                       $returnValue= 'void';
472                    endif
473                    if ($parameters='') then
474                      'public    ' $returnValue ' '$metotName ' () throws Exception {'
                         newline;
475                    else
476                    'public  '$returnValue' '$metotName ' ('$parameters')throws
                     Exception{'  newline;
477                     endif
478                  endif
479               }
480
481               $streotype = :Activity_streotype;
482               if ( $streotype ='receive-request') then
483
484
485                 dowhile :Service Name { $parameters= :Parameters ',' }
```

D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
                    $preCondition_sync = ~To>Activity-
Message~From.Activity_Diamond:condition;
                    $promise_sync = ~From>Activity-
Message~To.Activity_Diamond:condition;
                  endif

                  ' if ( ' $preCondition_sync ') {' newline;
                  '    ' $promise_sync ';'  newline;
                  if ($preName <> '') then
                      if ( $postResul<>'') then
                          'result=' $postResult ';'
                      endif
                   endif
                  ' } ' newline;
                  if ($returnValue='int') then
                      'return  result;  '  newline;

                  endif
                  '   }'  newline;
                }
           endif

       }

    }

    if (~Component>Interface~Required interface.Required_Connector) then

      '    public void run() ' newline;
      '            { ' newline;
      '     while(true) { ' newline;
      do ~Component>Interface~Required interface.Required_Connector
      {
          '               try ' newline;
          '               {   ' newline;
         do decompositions
         {

            foreach .ActivityRequest1
            {
               $streotype=:Activity_streotype;
                if ( $streotype= 'send-request') then
                     :Service Name '(' ;
                     dowhile :Service Name
                     {
                         $parameters= :Parameters
                         @count='1';
                         @countCheck='1';
                         do :Parameters:Data type
                         {
                            @dummy=@countCheck++;
                         }
                         do :Parameters:Data type
                         {
                           if ( :Data type='int') then
                              '0';
                              @dummy=@count++;
                           endif
                            if ( :Data type='bool') then
```

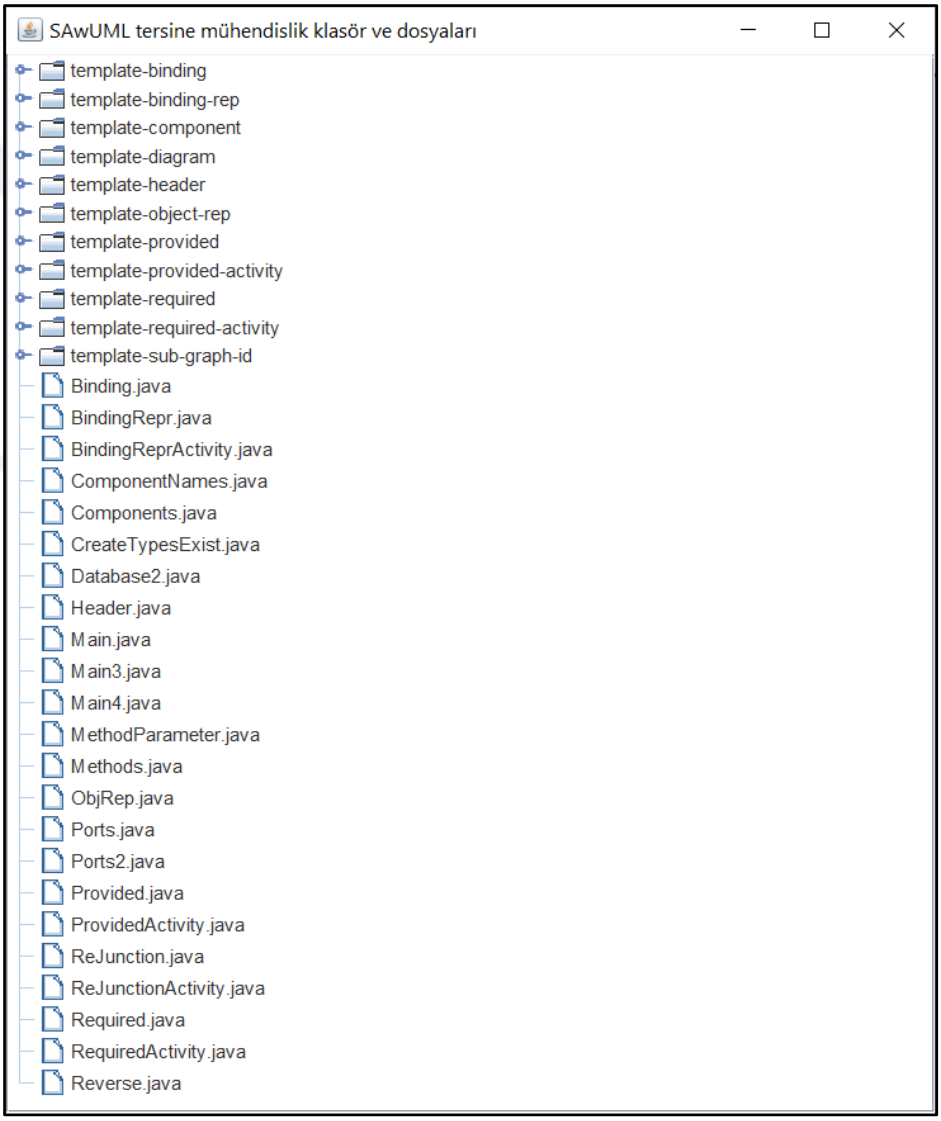
D.1: SAwUML_Activity-to-Java.merl dosyası (devam).

```
543                                 'true';
544                                 @dummy=@count++;
545                               endif
546                               if (@count<@countCheck ) then
547                                 ','
548                               endif
549                             }
550
551                           }
552                           ' ) ;' newline;
553                       endif
554                   }
555             }
556             '         }   ' newline;
557             '         catch (Exception e) ' newline;
558             '         {   'newline;
559             '            e.printStackTrace(); ' newline;
560             '          } ' newline;
561
562         }
563
564
565
566         '     }' newline;
567
568         ' } '  newline;
569       endif
570
571    '} '
572    newline;
573    newline;
574   }
575 endreport
576
```

EK E

PROMELA KODUNDAN SAwUML MODELİ ELDE ETME

MetaEdit+ editöründeki dömüştürücüden elde edilmiş bir ProMeLa dosyasından tersine mühendislikle SAwUML veya aktivite diyagramı ile genişletilmiş SAwUML modeli elde etmek için kullanılan kaynak kodu aşağıdaki Şekil E.1'de gösterilmektedir.



Şekil E.1: Tersine mühendislik için kullanılan java dosyaları ve klasörler.

E.1: Reverse.java dosyası.

```
/**   javac -classpath ".;sqlite-jdbc-3.7.2.jar" Reverse.java   **/
/**    java -classpath ".;sqlite-jdbc-3.7.2.jar" Reverse       **/
 import javax.swing.Box;
 import javax.swing.JLabel;
 import javax.swing.JOptionPane;
 import javax.swing.JPanel;
 import javax.swing.JTextField;
 import java.util.*;
 import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.filechooser.*;

public class Reverse {
 JPanel myPanel;
 JFrame f;
 String filePath;
 String data;

 Reverse()
 {
   filePath="";
   data="";
   f=new JFrame("Reverse Engineering for MetaEdit+");

   JLabel label = new JLabel("Reverse Engineering from .pml file to .mxm file for
                              MetaEdit+");
   label.setBounds(110,10,450,20);
   f.add(label);
   JButton browseButton = new JButton("Browse .pml Files");
   browseButton.setBounds(20,50,150,20);

   JTextField textField1 = new JTextField();
   textField1.setBounds(180,50,400,20);

   JButton b2 = new JButton("Run");
   b2.setBounds(20,80,90,20);

   JButton backButton = new JButton("Exit");
   backButton.setBounds(120,80,90,20);

   f.add(textField1);

   f.add(backButton);
   f.add(b2);
   f.add(browseButton);

   f.setLayout(null);
   f.setVisible(true);
   f.setSize(600,250);

   f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   backButton.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e)
    {
        f.dispose();
    }
   });
```

E.1: Reverse.java dosyası (devam).

```
59
60    b2.addActionListener(new ActionListener(){
61     public void actionPerformed(ActionEvent e)
62     {
63        if( !filePath.isEmpty())
64           new Main(filePath);
65     }
66    });
67    browseButton.addActionListener(new ActionListener(){
68       public void actionPerformed(ActionEvent e)
69       {
70          JFileChooser fileChooser = new JFileChooser();
71          FileNameExtensionFilter filter= new FileNameExtensionFilter("ProMeLa
                                                                        pml","pml");
72          fileChooser.setFileFilter(filter);
73          int result = fileChooser.showOpenDialog(f);
74          if (result == JFileChooser.APPROVE_OPTION)
75          {
76             File selectedFile = fileChooser.getSelectedFile();
77             filePath =selectedFile.getAbsolutePath();
78             data=selectedFile.getName();
79          }
80          textField1.setText(filePath);
81       }
82     });
83   }
84   public static void main(String[] args)
85   {
86      new Reverse();
87   }
88 }
```

E.2: Main.java dosyası.

```
1 public class Main {
2       Database2 databaseIns;
3       Components componentsIns;
4       MethodParameter methodParameterIns;
5       Methods methodsIns;
6       Ports portsIns;
7       Ports2 portsIns2;
8       ComponentNames componentNamesIns;
9       CreateTypesExist createTypesExistIns;
10
11      ComponentParams componentParamsIns;
12
13       public Main(String filePath)
14      {
15        databaseIns =new Database2();
16        componentsIns = new Components(filePath);
17        componentParamsIns = new ComponentParams(filePath);
18        methodParameterIns = new MethodParameter(filePath);
19        methodsIns = new Methods(filePath);
20        portsIns = new Ports(filePath);
21        portsIns2 = new Ports2(filePath);
```

E.2: Main.java dosyası (devam).

```
22        componentNamesIns = new ComponentNames(filePath);
23       createTypesExistIns = new CreateTypesExist();
24     }
25 }
26
```

E.3: Database2.java dosyası.

```
1  import java.sql.*;
2  import java.util.*;
3  import java.io.*;
4  import org.sqlite.SQLiteConfig;
5
6  public class Database2
7  {
8        Connection c = null;
9        Statement stmt = null;
10       Database2()
11       {
12         createTables();
13       }
14       public void createTables()
15       {
16          try
17          {
18
19             Class.forName("org.sqlite.JDBC");
20             SQLiteConfig config = new SQLiteConfig();
21             config.enforceForeignKeys(true);
22             c = DriverManager.getConnection( "jdbc:sqlite:ProMeLaDb.db",
                                                config.toProperties());
23
24             stmt = c.createStatement();
25             String sql=   "DROP TABLE IF EXISTS Components;"+
26                           "DROP TABLE IF EXISTS DataList;"+
27                           "DROP TABLE IF EXISTS MethodParameter;"+
28                           "DROP TABLE IF EXISTS Methods;"+
29                           "DROP TABLE IF EXISTS Ports;"+
30                           "DROP TABLE IF EXISTS CompMethod;"+
31                           "DROP TABLE IF EXISTS ReqCompMth;"+
32                           "DROP TABLE IF EXISTS ProCompMth;"+
33                           "DROP TABLE IF EXISTS ComponentNames;"+
34                          "DROP TABLE IF EXISTS ReqPostCond;"+
35                           "DROP TABLE IF EXISTS ReqPorts;"+
36                           "DROP TABLE IF EXISTS ParameterList;"+
37                           "CREATE TABLE Components  " +
38                                  "( componentName TEXT PRIMARY KEY NOT NULL UNIQUE  );
" +
39                           " CREATE TABLE DataList " +
40                                  " (dataName TEXT  NOT NULL ,  " +
41                                  " dataType TEXT  NOT NULL,  "+
42                                  " dataDefaultValue TEXT,  "+ " arraySize TEXT, " +
43                                  " componentName TEXT);  "+
44                                  " CREATE TABLE ParameterList " +
```

E.3: Database2.java dosyası (devam).

```
45                                  " (parameterName TEXT  NOT NULL ,  " +
46                                  " parameterType TEXT  NOT NULL,  "+
47                                  " parameterDefaultValue TEXT,  "+
48                                  " componentName TEXT);  "+
49                                  " CREATE TABLE MethodParameter " +
50                                  " (parameterType TEXT ,  " +
51                                  " parameterName TEXT,  "+
52                                  " componentName TEXT);  "+
53                                  " CREATE TABLE Methods " +
54                                  " (methodName TEXT ,  " +
55                                  " parameterName TEXT,  "+
56                                  " componentName TEXT);  "+
57                                  " CREATE TABLE Ports " +
58                                  " (portType TEXT NOT NULL,  " +
59                                  " preCon1 TEXT, "+
60                                  " postCon1 TEXT, "+
61                                  " preCon2 TEXT, "+
62                                  " postCon2 TEXT, "+
63                                  " methodName TEXT,"+
64                                  " compName TEXT,"+
65                                  " portName TEXT,"+
66                                  " compNameCoordinateX TEXT,"+
67                                  " compNameCoordinateY TEXT,"+
68                                  " methNameCoordinateX TEXT,"+
69                                  " methNameCoordinateY TEXT);"+
70                                  " CREATE TABLE ReqCompMth " +
71                                  " (portType TEXT NOT NULL,  " +
72                                  " methodName TEXT,"+
73                                  " portName TEXT,"+
74                                  " compName TEXT);"+
75                                  " CREATE TABLE ProCompMth " +
76                                  " (portType TEXT NOT NULL,  " +
77                                  " methodName TEXT,"+
78                                  " portName TEXT,"+
79                                  " compName TEXT);"+
80                                  " CREATE TABLE ComponentNames " +
81                                  " (componentName TEXT NOT NULL); "+
82                                  " CREATE TABLE ReqPostCond " +
83                                  " (preCon2 TEXT, "+
84                                  " postCon2 TEXT, "+
85                                  " methodName TEXT,"+
86                                  " compName TEXT);"+

87                                  " CREATE TABLE ReqPorts " +
88                                  "( preCon1 TEXT, "+
89                                  " postCon1 TEXT, "+
90                                  " methodName TEXT,"+
91                                  " compName TEXT,"+
92                                  " portName TEXT,"+
93                                  " compNameCoordinateX TEXT,"+
94                                  " compNameCoordinateY TEXT,"+
95                                  " methNameCoordinateX TEXT,"+
96                                  " methNameCoordinateY TEXT);";
97
98          stmt.executeUpdate(sql);
99          stmt.close();
100         c.close();
101
102        } catch ( Exception e ) {
```

E.3: Database2.java dosyası (devam).

```
103             System.err.println( e.getClass().getName() + ": " + e.getMessage() );
104             System.exit(0);
105         }
106         System.out.println("Table created successfully");
107       }
108
109
110 }
111
```

E.4: Component.java dosyası.

```
1  import java.io.IOException;
2  import java.io.RandomAccessFile;
3  import java.io.File;
4  import java.io.PrintWriter;
5  import java.sql.*;
6  import org.sqlite.SQLiteConfig;
7  import java.io.*;
8
9  public class Components {
10     Connection c;
11     Statement stmt;
12     String filePath;
13
14     public Components(String filePath) {
15       this.filePath = filePath;
16       writingComponents();
17     }
18
19     public void writingComponents() {
20       String componentName = "";
21       String dataName = "";
22       String dataType = "";
23       String dataDefaultValue = "";
24      String arraySize2="";
25       try {
26            RandomAccessFile javaFile = new RandomAccessFile(filePath, "r");
27            String strJava;
28            String strJava2;
29            while ((strJava = javaFile.readLine()) != null) {
30
31              if (strJava.contains("proctype") && !strJava.contains("active")) {
32                 componentName = strJava.trim();
33                 componentName = componentName.substring(0, componentName.indexOf("("));
34                 componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype", "");
35                 while ((strJava2 = javaFile.readLine()) != null) {
36                   if (strJava2.contains("do"))
37                       break;
38                   if (strJava2.contains("bool") && strJava2.contains("=")
39                        && !strJava2.contains("responseWaiting")) {
40                       strJava2 = strJava2.trim();
41                    strJava2 = strJava2.replaceAll(";", "");
42                    dataType = "bool";
```

E.4: Component.java dosyası (devam).

```
43                      strJava2 = strJava2.replaceAll("bool", "");
44                      String[] tempJavaLine = strJava2.split("=", 0);
45                      if (tempJavaLine[0].contains("[") && tempJavaLine[0].contains("]"))
46                      {
47                        String[] arraySize=tempJavaLine[0].split("\\[", 0);
48                        dataName=arraySize[0];
49                        arraySize2=arraySize[1].replaceAll("]", "");
50                      }
51                      else
52                       dataName = tempJavaLine[0];
53                      dataDefaultValue = tempJavaLine[1];
54
55                      c = null;
56                      stmt = null;
57                      try {
58                            Class.forName("org.sqlite.JDBC");
59                            SQLiteConfig config = new SQLiteConfig();
60                            config.enforceForeignKeys(true);
61
62                            c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                  config.toProperties());
63
64                            stmt = c.createStatement();
65
66                            dataType = dataType.trim();
67                            dataName = dataName.trim();
68                            dataDefaultValue = dataDefaultValue.trim();
69                            componentName = componentName.trim();
70                            String sql = "INSERT INTO DataList VALUES('" + dataName +
71                                    "','" +  dataType + "','"+ dataDefaultValue + "','"
72                                            + arraySize2 + "','" + componentName + "'); ";
73                            stmt.executeUpdate(sql);
74                            stmt.close();
75                            c.close();
76
77                     } catch (Exception err) {
78                            System.err.println(err.getClass().getName() + ": " +
                                          err.getMessage());
79                     }
80
81                  }
82                  if (strJava2.contains("int") && strJava2.contains("=")
83                      && !strJava2.contains("responseWaiting")) {
84                       strJava2 = strJava2.trim();
85                       strJava2 = strJava2.replaceAll(";", "");
86                       dataType = "int";
87                       strJava2 = strJava2.replaceAll("int", "");
88                       String[] tempJavaLine = strJava2.split("=", 0);
89                       if (tempJavaLine[0].contains("[")&& tempJavaLine[0].contains("]"))
90                       {
91                          String[] arraySize=tempJavaLine[0].split("\\[", 0);
92                          dataName=arraySize[0];
93                          arraySize2=arraySize[1].replaceAll("]", "");
94                        }
95                       else
96                          dataName = tempJavaLine[0];
97                       dataDefaultValue = tempJavaLine[1];
98
99                         c = null;
100                        stmt = null;
```

E.4: Component.java dosyası (devam).

```
                        try {
                              Class.forName("org.sqlite.JDBC");
                              SQLiteConfig config = new SQLiteConfig();
                              config.enforceForeignKeys(true);
                              c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                  config.toProperties());
                              stmt = c.createStatement();
                              dataType = dataType.trim();
                              dataName = dataName.trim();
                              dataDefaultValue = dataDefaultValue.trim();
                              componentName = componentName.trim();
                              String sql = "INSERT INTO DataList VALUES('" + dataName +
                                     "','" +  dataType + "','"+ dataDefaultValue + "','" +
                                   arraySize2 + "','"+ componentName + "'); ";
                              stmt.executeUpdate(sql);
                              stmt.close();
                              c.close();

                        } catch (Exception err) {
                            System.err.println(err.getClass().getName() + ": " +
                                               err.getMessage());
                        }
                    }
                 }
              }
           }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

E.5: MethodParameter.java dosyası

```
import java.io.IOException;
import java.io.RandomAccessFile;
import java.io.File;
import java.io.PrintWriter;
import java.sql.*;
import org.sqlite.SQLiteConfig;
import java.io.*;

public class MethodParameter {
    Connection c;
    Statement stmt;
    String filePath;

    public MethodParameter(String filePath)
    {
      this.filePath = filePath;
      writingMethodParameter();
    }

```

E.5: MethodParameter.java dosyası (devam).

```
20      public void writingMethodParameter()
21      {
22        String componentName = "";
23        String methodName = "";
24        String parameterType = null;
25        String parameterName = null;
26        try
27        {
28          RandomAccessFile javaFile = new RandomAccessFile(filePath, "r");
29          String strJava;
30          String strJava2;
31          while ((strJava = javaFile.readLine()) != null)
32          {
33            if (strJava.contains("proctype") && !strJava.contains("active"))
34            {
35                componentName = strJava.trim();
36                componentName = componentName.substring(0, componentName.indexOf("("));
37                componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype", "");
38                while ((strJava2 = javaFile.readLine()) != null) {
39                    if (strJava2.contains("do"))
40                        break;
41                    if (strJava2.contains("bool") && !strJava2.contains("=") &&
                          !strJava2.contains("responseWaiting") &&
!strJava2.contains("result"))
42                    {
43                        strJava2 = strJava2.trim();
44                        strJava2 = strJava2.replaceAll(";", "");
45                        parameterType = "bool";
46                        strJava2 = strJava2.replaceAll("bool", "");
47                        String[] tempJavaLine = strJava2.split("=", 0);
48                        parameterName = tempJavaLine[0];
49
50                        c = null;
51                        stmt = null;
52                        try

53                        {
54                              Class.forName("org.sqlite.JDBC");
55                              SQLiteConfig config = new SQLiteConfig();
56                              config.enforceForeignKeys(true);
57
58                              c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                                                config.toProperties());
59                              stmt = c.createStatement();
60
61                              parameterType = parameterType.trim();
62                              parameterName = parameterName.trim();
63                              componentName = componentName.trim();
64
65                              if (parameterType.isEmpty())
66                              {
67                                  parameterType=null;
68                                  System.out.println(" parameterType is NULL");
69                              }
70                              if (!parameterType.isEmpty())
71                              {
72                                 //parameterType=null;
73                                 System.out.println(" parameterType is NOT NULL");
74                              }
```

E.5: MethodParameter.java dosyası (devam).

```
75                              if (parameterName.isEmpty())
76                                  parameterName=null;
77
78                              String sql = "INSERT INTO MethodParameter VALUES('" +
parameterType
                                          + "','"+ parameterName + "','" + componentName +
"'); ";
79
80                              stmt.executeUpdate(sql);
81                              stmt.close();
82                              c.close();
83
84                              }
85                              catch (Exception err)
86                              {
87                                System.err.println(err.getClass().getName() + ": " +
                                                   err.getMessage());
88                              }
89                      }
90                      if (strJava2.contains("int") && !strJava2.contains("=") &&
                            !strJava2.contains("responseWaiting") &&
!strJava2.contains("result"))
91                      {
92                              strJava2 = strJava2.trim();
93                              strJava2 = strJava2.replaceAll(";", "");
94                              parameterType = "int";
95                              strJava2 = strJava2.replaceAll("int", "");
96                              String[] tempJavaLine = strJava2.split("=", 0);
97                              parameterName = tempJavaLine[0];
98
99                              c = null;
100                             stmt = null;
101                             try
102                             {
103                               Class.forName("org.sqlite.JDBC");
104                               SQLiteConfig config = new SQLiteConfig();
105                               config.enforceForeignKeys(true);
106
107                               c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                                                 config.toProperties());
108                                stmt = c.createStatement();
109                                parameterType = parameterType.trim();
110                               parameterName = parameterName.trim();
111                               componentName = componentName.trim();
112                               String sql = "INSERT INTO MethodParameter VALUES('" +
                                             parameterType + "','" + parameterName + "','"
+
                                             componentName + "'); ";
113                              stmt.executeUpdate(sql);
114                              stmt.close();
115                               c.close();
116                             }
117                             catch (Exception err)
118                             {
119                               System.err.println(err.getClass().getName() + ": " +
                                                   err.getMessage());
120                             }
121                     }
122             }
```

E.5: MethodParameter.java dosyası (devam).

```
123           }
124        }
125       }
126       catch (IOException e)
127       {
128            e.printStackTrace();
129       }
130    }
131}
```

E.6: Methods.java dosyası

```
1  import java.io.IOException;
2  import java.io.RandomAccessFile;
3  import java.io.File;
4  import java.io.PrintWriter;
5  import java.sql.*;
6  import org.sqlite.SQLiteConfig;
7  import java.io.*;
8
9  public class Methods
10 {
11   Connection c;
12   Statement stmt;
13   String filePath;
14
15   public Methods(String filePath) {
16     this.filePath = filePath;
17     writingComponents();
18   }
19
20   public void writingComponents()
21   {
22     String componentName = "";
23     String methodName = "";
24     String parameterName = "";
25     String portType = "";
26     try
27     {
28       RandomAccessFile javaFile = new RandomAccessFile(filePath, "r");
29       String strJava;
30       String strJava2;
31       while ((strJava = javaFile.readLine()) != null)
32       {
33            if (strJava.contains("proctype") && !strJava.contains("active"))
34            {
35              componentName = strJava.trim();
36              componentName = componentName.substring(0, componentName.indexOf("("));
37              componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype", "");
38              while ((strJava2 = javaFile.readLine()) != null)
39              {
40                   if (strJava2.contains("od"))
41                   {
42                     methodName = "";
```

E.6: Methods.java dosyası (devam).

```
43                        parameterName = "";
44                        break;
45                      }
46                      if (strJava2.contains("REQ"))
47                      {
48                        strJava2 = strJava2.trim();
49                        strJava2 = strJava2.replaceAll(";|:|-|>|\\?|!", "");
50                        String tempJavaLine[] = null;
51                        tempJavaLine = strJava2.split("REQ", 0);
52                        String strTemp = tempJavaLine[1];
53                        String[] tempJavaLine2 = strTemp.split(",", 0);
54                        if (tempJavaLine2.length == 1)
55                            parameterName = "";
56
57                        tempJavaLine2[0] = tempJavaLine2[0].trim();
58                        methodName = methodName.trim();
59
60                        if (!methodName.equals(tempJavaLine2[0]))
61                        {
62                              methodName = tempJavaLine2[0];
63                              if (tempJavaLine2.length == 1)
64                              {
65                                c = null;
66                                stmt = null;
67                                try
68                                {
69                                      Class.forName("org.sqlite.JDBC");
70                                      SQLiteConfig config = new SQLiteConfig();
71                                      config.enforceForeignKeys(true);
72                                      c =
DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
73
config.toProperties());
74                                      stmt = c.createStatement();
75
76                                      methodName = methodName.trim();
77                                      parameterName = parameterName.trim();
78                                      componentName = componentName.trim();
79
80                                      if (parameterName.isEmpty())
81                                          parameterName=null;
82
83                                      String sql = "INSERT INTO Methods VALUES('" +
methodName +
                                                   "','" + parameterName+ "','" +
componentName +
                                                   "'); ";
84
85                                      stmt.executeUpdate(sql);
86                                      stmt.close();
87                                      c.close();
88                                }
89                                catch (Exception err)
90                                {
91                                      System.err.println(err.getClass().getName() + ": " +
                                                   err.getMessage());
92                                }
93                              }
94                              for (int i = 1;i< tempJavaLine2.length;i++)
```

E.6: Methods.java dosyası (devam).

```
95                              {
96                                      c = null;
97                                      stmt = null;
98                                      try
99                                      {
100                                             Class.forName("org.sqlite.JDBC");
101                                             SQLiteConfig config = new SQLiteConfig();
102                                             config.enforceForeignKeys(true);
103
104                                             c = DriverManager.getConnection (

"jdbc:sqlite:ProMeLaDb.db",config.toProperties());
105
106                                             stmt = c.createStatement();
107                                             String sql = "";
108
109                                             methodName = methodName.trim();
110                                             parameterName = parameterName.trim();
111                                             componentName = componentName.trim();
112                                             tempJavaLine2[i] = tempJavaLine2[i].trim();
113                                             if (!(tempJavaLine2[i].equals("result")))
114                                                sql = "INSERT INTO Methods VALUES('" +
methodName
                                                        + "','" + tempJavaLine2[i]+ "','" +
                                                        componentName + "'); ";
115
116                                             stmt.executeUpdate(sql);
117                                             stmt.close();
118                                             c.close();
119
120                                     } catch (Exception err)
121                                     {
122                                        System.err.println(err.getClass().getName() + ":"+
                                                        err.getMessage());
123                                     }
124                             }
125                     }
126                 }
127             }
128         }
129       }
130     }
131     catch (IOException e)
132     {
133             e.printStackTrace();
134     }
135  }
136
137 }
138
```

E.7: Ports.java dosyası.

```
1  import java.io.IOException;
2  import java.io.RandomAccessFile;
3  import java.io.File;
4  import java.io.PrintWriter;
5  import java.sql.*;
6  import org.sqlite.SQLiteConfig;
7  import java.io.*;
8
9  public class Ports {
10     Connection c;
11     Statement stmt;
12     String filePath;
13
14     public Ports(String filePath) {
15       this.filePath = filePath;
16       writingPorts();
17     }
18
19     public void writingPorts() {
20      String componentName = "";
21      String portsType = "";
22      String preCon1 = "";
23      String postCon1 = "";
24      String preCon2 = "";
25      String postCon2 = "";
26
27      String methodName = "";
28      String portName = "";
29
30      try
31      {
32        RandomAccessFile javaFile = new RandomAccessFile(filePath, "r");
33        String strJava;
34        String strJava2;
35        String strJava3;
36        while ((strJava = javaFile.readLine()) != null)
37        {
38          if (strJava.contains("proctype") && !strJava.contains("active"))
39          {
40              componentName = strJava.trim();
41              componentName = componentName.substring(0, componentName.indexOf("("));
42              componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype", "");
43              while ((strJava2 = javaFile.readLine()) != null)
44              {
45                 if (strJava2.contains("od"))
46                    break;
47                 if (strJava2.contains("responseWaiting") && strJava2.contains("->"))
48                 {
49                    portsType = "Required";
50                    strJava2 = strJava2.trim();
51                    strJava2 = strJava2.replaceAll(":|-|>", "");
52                    String[] tempJavaLine = strJava2.split("&&", 0);
53                    preCon1 = tempJavaLine[0];
54                    preCon1=preCon1.trim();
55                    if (preCon1.equals("responseWaiting==false") ||
                          preCon1.equals("!responseWaiting"))
56                       preCon1="";
57                    while ((strJava3 = javaFile.readLine()) != null)
58                    {
```

E.7: Ports.java dosyası (devam)

```
59                      if (strJava3.contains("fi"))
60                      {
61                        portsType = "";
62                        preCon1 = "";
63                        postCon1 = "";
64                        preCon2 = "";
65                        postCon2 = "";
66                        methodName = "";
67                        break;
68                       }
69                       if (!strJava3.contains("::") && strJava3.contains("=")
70                             && !strJava3.contains("responseWaiting"))
71                       {
72                         postCon1 = postCon1 + strJava3.trim();
73                         postCon1 = postCon1.replaceAll("(.*);", "$1");
74                       }
75                       if (strJava3.contains("REQ") && strJava3.contains("!"))
76                       {
77                          //extracting portName
78                          String strJava6 = strJava3.trim();
79                          strJava6 = strJava6.replaceAll(";", "");
80                          strJava6 = strJava6.replaceAll("(.*)_REQ!(.*)", "$1");
81                          String[] tempJavaLine3 = strJava6.split("_", 0);
82                          portName = tempJavaLine3[1];
83
84                         // end of extracting port Name
85                         String strJava4 = strJava3.trim();
86                         strJava4 = strJava4.replaceAll(";", "");
87                         strJava4 = strJava4.replaceAll("(.*)!(.*)", "$2");
88                         String[] tempJavaLine2 = strJava4.split(",", 0);
89                         methodName = tempJavaLine2[0];
90
91                        c = null;
92                        stmt = null;
93                        try
94                        {
95                           Class.forName("org.sqlite.JDBC");
96                           SQLiteConfig config = new SQLiteConfig();
97                           config.enforceForeignKeys(true);
98
99                           c = DriverManager.getConnection(
                                  "jdbc:sqlite:ProMeLaDb.db",config.toProperties());
100                          stmt = c.createStatement();
101                          preCon1 = preCon1.trim();
102                          postCon1 = postCon1.trim();
103                          methodName = methodName.trim();
104                          componentName = componentName.trim();
105                          portName = portName.trim();
106
107                          String sql = "INSERT INTO ReqPorts VALUES('" + preCon1 +
"','" +
                                              postCon1+ "','" + methodName + "','" +
componentName
                                              + "','" + portName + "',"+ "'','','','');";
108                          stmt.executeUpdate(sql);
109                          stmt.close();
110                          c.close();
111                        }
112                        catch (Exception err)
```

E.7: Ports.java dosyası (devam).

```
113                         {
114                           System.err.println(err.getClass().getName() + ": " +
                                                  err.getMessage());
115                       }
116                     }
117                     if (strJava3.contains("::") && !strJava3.contains("assert") &&
                            strJava3.contains("->")&& !strJava3.contains("REQ") &&
                            !strJava3.contains("RES"))
118                     {
119                        strJava3 = strJava3.trim();
120                        strJava3 = strJava3.replaceAll("(.*);", "$1");
121                        strJava3 = strJava3.replaceAll(":|-|>|\\(", "");
122                        String[] tempJavaLine2 = strJava3.split("\\)", 0);
123                        for (int i = 0; i < tempJavaLine2.length; i++)
124                        {
125                          if (i == 0)
126                             preCon2 = tempJavaLine2[i];
127                          else
128                             postCon2 = tempJavaLine2[i];
129                        }
130                        c = null;
131                        stmt = null;
132                        try
133                        {
134                          Class.forName("org.sqlite.JDBC");
135                          SQLiteConfig config = new SQLiteConfig();
136                          config.enforceForeignKeys(true);
137
138                          c = DriverManager.getConnection(
                                   "jdbc:sqlite:ProMeLaDb.db",config.toProperties());
139                          stmt = c.createStatement();
140
141                          preCon2 = preCon2.trim();
142                          postCon2 = postCon2.trim();
143                          methodName = methodName.trim();
144                          componentName = componentName.trim();
145
146                          String sql = "INSERT INTO ReqPostCond VALUES('" + preCon2 +
"','" +
                                                 postCon2+ "','" + methodName + "','" +
componentName +
                                                 "'); ";
147
148                          stmt.executeUpdate(sql);
149                          stmt.close();
150                          c.close();
151                        } catch (Exception err)
152                        {
153                           System.err.println(err.getClass().getName() + ": " +
                                                 err.getMessage());
154                        }
155                     }
156                   }
157                 }
158                 if (strJava2.contains("REQ") && strJava2.contains("?"))
159                 {
160                   // extracting portName
161                   String strJava6 = strJava2.trim();
162                   strJava6 = strJava6.replaceAll(":,;", "");
```

E.7: Ports.java dosyası (devam).

```
163                     strJava6 = strJava6.replaceAll("(.*)_REQ!(.*)", "$1");
164                     String[] tempJavaLine4 = strJava6.split("_", 0);
165                     portName = tempJavaLine4[1];
166
167                     // end of extracting port Name
168                     portsType = "Provided";
169                     String strJava5 = strJava2.trim();
170                     strJava5 = strJava5.replaceAll(";|-|>", "");
171                     strJava5 = strJava5.replaceAll("(.*)\\?(.*)", "$2");
172
173                     String[] tempJavaLine5 = strJava5.split(",", 0);
174                     methodName = tempJavaLine5[0];
175                     while ((strJava3 = javaFile.readLine()) != null)
176                     {
177                       if (strJava3.contains("RES") && strJava3.contains("!"))
178                       {
179                         strJava3 = strJava3.trim();
180                        strJava3 = strJava3.replaceAll("(.*);", "$1");
181                        String[] tempJavaLine2 = strJava3.split("!", 0);
182                        String[] tempJavaLine3 = tempJavaLine2[1].split(",", 0);
183                        for (int i = 0; i < tempJavaLine3.length; i++)
184                        {
185                          if (i == 1 && !tempJavaLine3[1].equals("0"))
186                          {
187                            postCon2 = tempJavaLine3[1];
188                          }
189                          else
190                            postCon2 = "";
191                        }
192                        c = null;
193                        stmt = null;
194                        try
195                        {
196                          Class.forName("org.sqlite.JDBC");
197                          SQLiteConfig config = new SQLiteConfig();
198                          config.enforceForeignKeys(true);
199
200                          c = DriverManager.getConnection( "jdbc:sqlite:ProMeLaDb.db",
201                                                           config.toProperties());
202                          stmt = c.createStatement();
203
204                          portsType = portsType.trim();
205                          preCon1 = preCon1.trim();
206                          postCon1 = postCon1.trim();
207                          preCon2 = preCon2.trim();
208                          postCon2 = postCon2.trim();
209                          methodName = methodName.trim();
210                          componentName = componentName.trim();
211                          portName = portName.trim();
212
213                          String sql = "INSERT INTO Ports VALUES('" + portsType + "','" +
preCon1
                                       + "','"+ postCon1 + "','" + preCon2 + "','" +
postCon2 +
                                       "','" + methodName+ "','" + componentName + "','" +
                                       portName + "'," + "'','','','','); ";
214                          stmt.executeUpdate(sql);
215                          stmt.close();
216                          c.close();
```

E.7: Ports.java dosyası (devam).

```
                    }
                    catch (Exception err)
                    {
                      System.err.println(err.getClass().getName()
+":"+err.getMessage());
                    }
                    portsType = "";
                    preCon1 = "";
                    postCon1 = "";
                    preCon2 = "";
                    postCon2 = "";
                    break;
                  }
                  if (strJava3.contains("::") && strJava3.contains("->"))
                  {
                      String strJava4 = strJava3.trim();
                      strJava4 = strJava3.replaceAll(";|:|-|>|\\(|\\)", "");
                      preCon1 = strJava4;
                  }
                  if (!strJava3.contains("::") && !strJava3.contains("->") &&
                       !strJava3.contains("if"))
                  {
                      String strJava4 = strJava3.replaceAll("(.*);", "$1");
                      strJava4 = strJava4.trim();
                      postCon1 = postCon1 + strJava4;

                  }
                }
              }
            }
          }
        }
         }
         catch (IOException e)
         {
           e.printStackTrace();
         }
    }
}

```

E.8: Ports2.java dosyası.

```
import java.io.IOException;
import java.io.RandomAccessFile;
import java.io.File;
import java.io.PrintWriter;
import java.sql.*;
import org.sqlite.SQLiteConfig;
import java.io.*;

public class Ports2 {
    Connection c;
```

E.8: Ports2.java dosyası (devam).

```
11      Statement stmt;
12      boolean skip;
13      String filePath;
14
15      public Ports2(String filePath) {
16             this.filePath = filePath;
17             writingPorts2();
18      }
19      public void writingPorts2()
20      {
21        String componentName = "";
22        String portsType = "";
23        String methodName = "";
24        String portName = "";
25        skip = false;
26        try
27        {
28              RandomAccessFile javaFile = new RandomAccessFile(filePath, "r");
29              String strJava;
30              String strJava2;
31              String strJava3;
32              while ((strJava = javaFile.readLine()) != null)
33              {
34                if (strJava.contains("proctype") && !strJava.contains("active"))
35                {
36                      componentName = strJava.trim();
37                      componentName = componentName.substring(0,
componentName.indexOf("("));
38                      componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype",
"");
39                      while ((strJava2 = javaFile.readLine()) != null)
40                      {
41                        if (strJava2.contains("od"))
42                        {
43                              skip = false;
44                              break;
45                        }
46                        if (strJava2.contains("fi"))
47                        {
48                              skip = false;
49                        }
50                        if (strJava2.contains("::!("))
51                        {
52                              skip = true;
53                        }
54                        if (skip)
55                        {
56                              continue;
57                        }
58                        if (strJava2.contains("REQ") && strJava2.contains("!"))
59                        {
60                              // extracting portName
61                              String strJava6 = strJava2.trim();
62                              strJava6 = strJava6.replaceAll(";", "");
63                              strJava6 = strJava6.replaceAll("(.*)_REQ!(.*)", "$1");
64                              String[] tempJavaLine3 = strJava6.split("_", 0);
65                              portName = tempJavaLine3[1];
66
67                              // end of extracting port Name
```

E.8: Ports2.java dosyası (devam).

```
68                              portsType = "Required";
69                              String strJava4 = strJava2.trim();
70                              strJava4 = strJava4.replaceAll(";", "");
71                              strJava4 = strJava4.replaceAll("(.*)!(.*)", "$2");
72                              String[] tempJavaLine2 = strJava4.split(",", 0);
73                              methodName = tempJavaLine2[0];
74                              c = null;
75                              stmt = null;
76                              try
77                              {
78                                      Class.forName("org.sqlite.JDBC");
79                                      SQLiteConfig config = new SQLiteConfig();
80                                      config.enforceForeignKeys(true);
81                                      c = DriverManager.getConnection(
                                            "jdbc:sqlite:ProMeLaDb.db",config.toProperties());
82                                      stmt = c.createStatement();
83                                      portsType = portsType.trim();
84                                      methodName = methodName.trim();
85                                      componentName = componentName.trim();
86                                      String sql = "INSERT INTO ReqCompMth VALUES('" +
portsType +
                                                   "','" +methodName + "','"+ portName +
"','"  +
                                                   componentName + "'); ";
87                                      stmt.executeUpdate(sql);
88                                      stmt.close();
89                                      c.close();
90                              }
91                              catch (Exception err)
92                              {
93                                System.err.println(err.getClass().getName() + ":"+
                                                   err.getMessage());
94                              }
95                          }
96                          if (strJava2.contains("RES") && strJava2.contains("!"))
97                          {
98                              // extracting portName
99                              String strJava6 = strJava2.trim();
100                             strJava6 = strJava6.replaceAll(":,;", "");
101                             strJava6 = strJava6.replaceAll("(.*)_REQ!(.*)", "$1");
102                             String[] tempJavaLine4 = strJava6.split("_", 0);
103                             portName = tempJavaLine4[1];
104                             portsType = "Provided";
105                             String strJava4 = strJava2.trim();
106                             strJava4 = strJava4.replaceAll(";", "");
107                             strJava4 = strJava4.replaceAll("(.*)!(.*)", "$2");
108                             String[] tempJavaLine2 = strJava4.split(",", 0);
109                             methodName = tempJavaLine2[0];
110                             c = null;
111                             stmt = null;
112                             try
113                             {
114                                     Class.forName("org.sqlite.JDBC");
115                                     SQLiteConfig config = new SQLiteConfig();
116                                     config.enforceForeignKeys(true);
117
118                                     c =
DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",

config.toProperties());
```

E.8: Ports2.java dosyası (devam).

```
119                                     stmt = c.createStatement();
120
121                                     stmt = c.createStatement();
122                                     portsType = portsType.trim();
123                                     methodName = methodName.trim();
124                                     componentName = componentName.trim();
125                                     String sql = "INSERT INTO ProCompMth VALUES('" +
portsType +
                                                    "','" + methodName + "','"+ portName +
"','" +
                                                    componentName + "'); ";
126
127                                     stmt.executeUpdate(sql);
128                                     stmt.close();
129                                     c.close();
130                             }
131                             catch (Exception err)
132                             {
133                               System.err.println( err.getClass().getName()+":"
                                                    +err.getMessage());
134                             }
135                         }
136                     }
137                 }
138               }
139             }
140             catch (IOException e)
141             {
142                     e.printStackTrace();
143             }
144     }
145 }
146
```

E.9: ComponentNames.java dosyası.

```
1  import java.io.IOException;
2  import java.io.RandomAccessFile;
3
4  import java.io.File;
5  import java.io.PrintWriter;
6
7  import java.sql.*;
8  import org.sqlite.SQLiteConfig;
9  import java.io.*;
10
11 public class ComponentNames {
12      Connection c;
13      Statement stmt;
14      String filePath;
15
16      public ComponentNames(String filePath) {
17              this.filePath = filePath;
18              writingComponentNames();
```

E.9: ComponentNames.java dosyası (devam).

```
19      }
20
21      public void writingComponentNames()
22      {
23        String componentName = "";
24       try
25        {
26              RandomAccessFile javaFile = new RandomAccessFile(filePath,"r");
27              String strJava;
28              String strJava2;
29
30              while ((strJava = javaFile.readLine()) != null)
31              {
32                if (strJava.contains("proctype") && !strJava.contains("active"))
33                {
34                      componentName = strJava.trim();
35                      componentName = componentName.substring(0,
                                                    componentName.indexOf("("));
36                      componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype",
                                                          "");
37                      c = null;
38                      stmt = null;
39                      try
40                      {
41                        Class.forName("org.sqlite.JDBC");
42                        SQLiteConfig config = new SQLiteConfig();
43                        config.enforceForeignKeys(true);
44                        c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                                    config.toProperties());
45                        stmt = c.createStatement();
46                        componentName = componentName.trim();
47                        String sql = "INSERT INTO ComponentNames VALUES('" + componentName
                                                                + "'); ";
48                        stmt.executeUpdate(sql);
49                        stmt.close();
50                        c.close();
51
52                      }
53                      catch (Exception err)
54                      {
55                        System.err.println(err.getClass().getName()+":"+err.getMessage());
56                      }
57                }
58              }
59        }
60        catch (IOException e)
61        {
62              e.printStackTrace();
63        }
64      }
65  }
66
```

E.10: ComponentParams.java dosyası .

```
1  import java.io.IOException;
2  import java.io.RandomAccessFile;
3  import java.io.File;
4  import java.io.PrintWriter;
5  import java.sql.*;
6  import org.sqlite.SQLiteConfig;
7  import java.io.*;
8
9  public class ComponentParams {
10     Connection c;
11     Statement stmt;
12     String filePath;
13
14     public ComponentParams(String filePath) {
15            this.filePath = filePath;
16            writingComponentParams();
17     }
18
19     public void writingComponentParams() {
20            String componentName = "";
21            String paramName = "";
22            String paramType = "";
23            String paramDefaultValue = "";
24
25            try {
26                RandomAccessFile javaFile = new RandomAccessFile(filePath, "r");
27                String strJava;
28                String strJava2;
29                while ((strJava = javaFile.readLine()) != null)
30                {
31
32                  if (strJava.contains("proctype") && !strJava.contains("active")) {
33                      componentName = strJava.trim();
34                      componentName = componentName.substring(0,
                                                    componentName.indexOf("("));
35                      componentName = componentName.replaceAll("[^a-zA-Z0-9]|proctype",
                                                        "");
36                      componentName=componentName.trim();
37
38                     String answer = strJava.substring(strJava.indexOf("(")+1,
                                                  strJava.indexOf(")"));
39                     answer=answer.trim();
40                    if (answer.contains(";"))
41                    {
42                        String[] splitAnswer1 =answer.split(";");
43
44                       for(int i = 0; i < splitAnswer1.length; i++)
45                       {
46                          String[] splitAnswer2 =splitAnswer1[i].split("\\s+");
47                           paramType=splitAnswer2[0].trim();
48                          paramName=splitAnswer2[1].trim();
49                          RandomAccessFile javaFile2 = new RandomAccessFile(filePath,
                                                                "r");
50                          while ((strJava2 = javaFile2.readLine()) != null)
51                          {
52                            if (strJava2.contains("run") &&
                                      strJava2.contains(componentName))
53                            {
54                              String answer3 =
```

E.10: ComponentParams.java dosyası (devam).

```
                                    strJava2.substring(strJava2.indexOf("(")+1,
                                                       strJava2.indexOf(")"));
55                              String[] splitAnswer3 =answer3.split(",");
56                              paramDefaultValue = splitAnswer3[i];

57                            }
58                          }
59                          c = null;
60                          stmt = null;
61                          try {
62                               Class.forName("org.sqlite.JDBC");
63                               SQLiteConfig config = new SQLiteConfig();
64                               config.enforceForeignKeys(true);
65
66                               c = DriverManager.getConnection(
                                      "jdbc:sqlite:ProMeLaDb.db", config.toProperties());
67
68                               stmt = c.createStatement();
69
70                               paramType = paramType.trim();
71                               paramName = paramName.trim();
72                               paramDefaultValue = paramDefaultValue.trim();
73                               componentName = componentName.trim();
74                               String sql = "INSERT INTO ParameterList VALUES('" +
                                             paramName  + "','" + paramType + "','"
                                             + paramDefaultValue + "','" +
                                             componentName + "'); ";
75                               stmt.executeUpdate(sql);
76                               stmt.close();
77                               c.close();
78
79                          } catch (Exception err) {
80                               System.err.println (err.getClass().getName() + ": " +
                                                    err.getMessage());
81                          }
82                      }
83
84                  }
85                  if (!answer.contains(";") && !answer.isEmpty())
86                  {
87
88                      String[] splitAnswer3 =answer.split("\\s+");
89                      paramType=splitAnswer3[0].trim();
90                      paramName=splitAnswer3[1].trim();
91                      while ((strJava2 = javaFile.readLine()) != null)
92                      {
93                         if (strJava2.contains("run") &&
                               strJava2.contains(componentName))
94                         {
95                            paramDefaultValue = strJava2.substring(
                                                       strJava2.indexOf("(")+ 1,
                                                       strJava2.indexOf(")")).trim();
96                         }
97
98                      }
99                      c = null;
100                     stmt = null;
101                      try {
102                        Class.forName("org.sqlite.JDBC");
```

E.10: ComponentParams.java dosyası (devam).

```
103                             SQLiteConfig config = new SQLiteConfig();
104                             config.enforceForeignKeys(true);
105
106                             c = DriverManager.getConnection(
                                        "jdbc:sqlite:ProMeLaDb.db",config.toProperties());
107
108                             stmt = c.createStatement();
109
110                             paramType = paramType.trim();
111                             paramName = paramName.trim();
112                             paramDefaultValue = paramDefaultValue.trim();
113
114                             componentName = componentName.trim();
115
116                            String sql = "INSERT INTO ParameterList VALUES('" +
                                              paramName + "','" + paramType + "','"+
                                       paramDefaultValue + "','" + componentName + "'); ";
117                             stmt.executeUpdate(sql);
118                             stmt.close();
119                             c.close();
120
121                           } catch (Exception err) {
122                                 System.err.println( err.getClass().getName() + ": " +
                                                        err.getMessage());
123                           }
124                        }
125                     }
126                 }
127             } catch (IOException e) {
128                   e.printStackTrace();
129             }
130     }
131 }
```

E.11: CreateTypesExist.java dosyası.

```
1  import java.awt.event.ActionEvent;
2  import java.awt.event.ActionListener;
3  import java.awt.event.MouseAdapter;
4  import java.awt.event.MouseEvent;
5  import javax.swing.JButton;
6  import javax.swing.JFrame;
7  import javax.swing.JScrollPane;
8  import javax.swing.JTable;
9  import javax.swing.JTextField;
10 import javax.swing.table.DefaultTableModel;
11 import javax.swing.JComboBox;
12 import java.util.*;
13 import java.sql.*;
14 import org.sqlite.SQLiteConfig;
15 import javax.swing.JOptionPane;
16 import javax.swing.JLabel;
17 import java.io.*;
18
```

E.11: CreateTypesExist.java dosyası (devam).

```
19 public class CreateTypesExist {
20     JFrame frame;
21 
22     Connection c;
23     Statement stmt;
24     Object[] row3;
25     Object[] row4;
26     String patternName;
27     JScrollPane pane3;
28     JScrollPane pane4;
29     String data1, data2, data3;
30 
31     JTextField textId3;
32     int count;
33 
34     CreateTypesExist()
35     {
36       frame = new JFrame();
37       JLabel label = new JLabel("Provided Ports of Components");
38       label.setBounds(110, 10, 450, 20);
39       frame.add(label);
40       count = 0;
41       openDb3();
42       openDb2();
43 
44       JButton mxmActivityButton = new JButton("Create .mxm file for Activity Diagram");
45       mxmActivityButton.setBounds(20, 650, 270, 20);
46 
47       mxmActivityButton.addActionListener(new ActionListener()
48       {
49          public void actionPerformed(ActionEvent e) {
50            new Main4();
51          }
52       });
53 
54       frame.add(mxmActivityButton);
55 
56       JButton mxmSequenceButton = new JButton("Create .mxm file for Sequence Diagram");
57       mxmSequenceButton.setBounds(300, 650, 270, 20);
58 
59       mxmSequenceButton.addActionListener(new ActionListener()
60       {
61          public void actionPerformed(ActionEvent e) {
62            new Main3();
63          }
64       });
65 
66       frame.add(mxmSequenceButton);
67       frame.setSize(1200, 800);
68       frame.setLocationRelativeTo(null);
69       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
70       frame.setVisible(true);
71     }
72 
73     public void openDb3() {
74       System.out.println("Table created successfully");
75       JTable table3;
76       DefaultTableModel model3;
77       table3 = new JTable();
```

E.11: CreateTypesExist.java dosyası (devam).

```
78          row3 = new Object[5];
79          model3 = new DefaultTableModel();
80          model3.addColumn("PRE CONDITION/RECEIVED");
81          model3.addColumn("POST CONDITION/RECEIVED");
82          model3.addColumn("POST CONDITION/SEND");
83          model3.addColumn("METHOD NAME");
84          model3.addColumn("COMPONENT NAME");
85
86          table3.setModel(model3);
87          JButton btnUpDate3 = new JButton("Edit Components Conditions");
88          btnUpDate3.setBounds(250, 260, 250, 25);
89
90          pane3 = new JScrollPane(table3);
91          pane3.setBounds(20, 50, 1050, 200);
92          frame.setLayout(null);
93          frame.add(pane3);
94          frame.add(btnUpDate3);
95          stmt = null;
96          try
97          {
98              Class.forName("org.sqlite.JDBC");
99              SQLiteConfig config = new SQLiteConfig();
100               config.enforceForeignKeys(true);
101               c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                                  config.toProperties());
102
103               stmt = c.createStatement();
104               ResultSet rs = stmt.executeQuery("SELECT * FROM Ports;");
105
106               while (rs.next())
107               {
108                       row3[0] = rs.getString("preCon1");
109                       row3[1] = rs.getString("postCon1");
110                       row3[2] = rs.getString("postCon2");
111                       row3[3] = rs.getString("metotName");
112                       row3[4] = rs.getString("compName");
113                       model3.addRow(row3);
114               }
115               stmt.close();
116               c.close();
117
118               } catch (Exception err)
119               {
120                 System.err.println(err.getClass().getName()+":"+ err.getMessage());
121                 System.exit(0);
122               }
123
124               table3.addMouseListener(new MouseAdapter()
125               {
126                 @Override
127                 public void mouseClicked(MouseEvent e)
128                 {
129                    int i = table3.getSelectedRow();
130                    int j = table3.getSelectedColumn();
131                 }
132               });
133
134               btnUpDate3.addActionListener(new ActionListener()
135               {
```

E.11: CreateTypesExist.java dosyası (devam).

```
136             @Override
137             public void actionPerformed(ActionEvent e)
138             {
139               int i = table3.getSelectedRow();
140               int j = table3.getSelectedColumn();
141               String tempColumnValue = "";
142
143               if (j == 0)
144               {
145                   String precon1 = JOptionPane.showInputDialog(frame,"Pre
                                                          Condition/Received");
146                   tempColumnValue = precon1;
147               }
148               if (j == 1)
149               {
150                   String precon2 = JOptionPane.showInputDialog(frame,"Post
                                                          Condition/Received");
151                   tempColumnValue = precon2;
152               }
153               if (j == 2)
154               {
155                   String methodName = JOptionPane.showInputDialog(frame,"Post
                                                            Condition/Send");
156                   tempColumnValue = methodName;
157               }
158
159               String oldValue = "";
160               if (i >= 0 && j >= 0 && j < 3)
161               {
162                   oldValue = model3.getValueAt(i, j).toString();
163                   model3.setValueAt(tempColumnValue, i, j);
164               }
165               else
166               {
167                   System.out.println("Update Error");
168                }
169                c = null;
170                stmt = null;
171                try
172                {
173                     Class.forName("org.sqlite.JDBC");
174                     SQLiteConfig config = new SQLiteConfig();
175                     config.enforceForeignKeys(true);
176                     c = DriverManager.getConnection(
                              "jdbc:sqlite:ProMeLaDb.db",config.toProperties());
177                     stmt = c.createStatement();
178                     String sql = "";
179                     if (j == 0)
180                     {
181                        sql = "UPDATE Ports SET preCon1='" + tempColumnValue + "' WHERE
                             preCon1='" + oldValue + "';";
182                     }
183                     if (j == 1)
184                     {
185                        sql = "UPDATE Ports SET postCon1='" + tempColumnValue + "' WHERE
                             postCon1='" + oldValue + "';";
186                     }
187                     if (j == 2)
188                     {
```

E.11: CreateTypesExist.java dosyası (devam).

```
189                         sql = "UPDATE Ports SET postCon2='" + tempColumnValue + "' WHERE
                                  postCon2='" + oldValue + "';";
190                     }
191                     if (!sql.equals(""))
192                         stmt.executeUpdate(sql);
193                      stmt.close();
194                      c.close();
195                 }
196                 catch (Exception err)
197                 {
198                       System.err.println(err.getClass().getName() +
                                            ":"+err.getMessage());
199                       System.exit(0);
200                 }
201              }
202           });
203
204    }
205
206    public void openDb2()
207    {
208      System.out.println("Table created successfully");
209      JTable table3;
210      DefaultTableModel model3;
211
212      table3 = new JTable();
213      row4 = new Object[6];
214      model3 = new DefaultTableModel();
215      model3.addColumn("PRE CONDITION/REQUEST");
216      model3.addColumn("PPOMISE");
217      model3.addColumn("PRE CONDITION/RECEIVED");
218      model3.addColumn("POST CONDITION/RECEIVED");
219      model3.addColumn("METHOD NAME");
220      model3.addColumn("COMPONENT NAME");
221
222      table3.setModel(model3);
223      JButton btnUpDate3 = new JButton("Edit Components Conditions");
224      btnUpDate3.setBounds(250, 600, 250, 25);
225
226      pane4 = new JScrollPane(table3);
227      pane4.setBounds(20, 380, 1050, 200);
228      JLabel label = new JLabel("Required Ports of Components");
229      label.setBounds(110, 340, 450, 20);
230      frame.add(label);
231      frame.setLayout(null);
232      frame.add(pane4);
233      frame.add(btnUpDate3);
234
235      c = null;
236      stmt = null;
237      try
238      {
239           Class.forName("org.sqlite.JDBC");
240           SQLiteConfig config = new SQLiteConfig();
241           config.enforceForeignKeys(true);
242           c= DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db"
                                            ,config.toProperties());
243           stmt = c.createStatement();
244           ResultSet rs = stmt.executeQuery("SELECT * FROM ReqPorts;");
245           String tempMethodName = "";
```

E.11: CreateTypesExist.java dosyası (devam).

```
246             while (rs.next())
247             {
248               row4[0] = rs.getString("preCon1");
249               row4[1] = rs.getString("postCon1");
250               tempMethodName = rs.getString("methodName");
251               Statement stmt2 = c.createStatement();
252               ResultSet rs2 =
253               stmt2.executeQuery ("SELECT * FROM ReqPostCond WHERE
                                      methodName='"+tempMethodName + "';");
254
255               while (rs2.next())
256               {
257                 row4[2] = rs2.getString("preCon2");
258                 row4[3] = rs2.getString("postCon2");
259                 row4[4] = rs2.getString("methodName");
260                 row4[5] = rs2.getString("compName");
261                 model3.addRow(row3);
262               }
263               stmt2.close();
264             }
265             stmt.close();
266             c.close();
267
268       }
269       catch (Exception err)
270       {
271            System.err.println(err.getClass().getName()+":"+err.getMessage());
272            System.exit(0);
273       }
274        table3.addMouseListener(new MouseAdapter() {
275              @Override
276              public void mouseClicked(MouseEvent e){
277                 int i = table3.getSelectedRow();
278                 int j = table3.getSelectedColumn();
279              }
280       });
281
282       btnUpDate3.addActionListener(new ActionListener(){
283            @Override
284            public void actionPerformed(ActionEvent e)
285            {
286              int i = table3.getSelectedRow();
287              int j = table3.getSelectedColumn();
288              String tempColumnValue = "";
289
290              if (j == 0) {
291                    String precon1 = JOptionPane.showInputDialog(frame, "Pre
                                                          Condition/Request");
292                    tempColumnValue = precon1;
293              }
294              if (j == 1) {
295                    String promise = JOptionPane.showInputDialog(frame, "Promise");
296                    tempColumnValue = promise;
297              }
298              if (j == 2) {
299                    String precon2 = JOptionPane.showInputDialog(frame, "Pre
                                                        Condition/Received");
300                    tempColumnValue = precon2;
301              }
```

E.11: CreateTypesExist.java dosyası (devam).

```
302                 if (j == 3) {
303                       String postcon2 = JOptionPane.showInputDialog(frame, "Post
                                                                  Condition/Received");
304                       tempColumnValue = postcon2;
305                 }
306
307                 String oldValue = "";
308                 if (i >= 0 && j >= 0 && j < 4) {
309                    oldValue = model3.getValueAt(i, j).toString();
310                    model3.setValueAt(tempColumnValue, i, j);
311                 } else {
312                        System.out.println("Update Error");
313                 }
314
315                 c = null;
316                 stmt = null;
317
318                 try {
319                       Class.forName("org.sqlite.JDBC");
320                       SQLiteConfig config = new SQLiteConfig();
321                       config.enforceForeignKeys(true);
322
323                       c = DriverManager.getConnection(
                                "jdbc:sqlite:ProMeLaDb.db",config.toProperties());
324                       stmt = c.createStatement();
325                       String sql = "";
326
327                       if (j == 0) {
328                         sql = "UPDATE ReqPorts SET preCon1='" +tempColumnValue+"' WHERE
                                preCon1='" +oldValue + "';";
329                       }
330                       if (j == 1) {
331                         sql = "UPDATE ReqPorts SET postCon1='" +tempColumnValue+"' WHERE
                                postCon1='" +oldValue+"';";
332                       }
333                       if (j == 2) {
334                         sql = "UPDATE ReqPostCond SET preCon2='"+tempColumnValue+"' WHERE
                                preCon2='" + oldValue+ "';";
335                       }
336                       if (j == 3) {
337                         sql = "UPDATE ReqPostCond SET postCon2='"+tempColumnValue + "'
                                WHERE  postCon2='" + oldValue+ "';";
338                       }
339
340                       stmt.executeUpdate(sql);
341
342                       stmt.close();
343                       c.close();
344
345                  } catch (Exception err) {
346                      System.err.println(err.getClass().getName()+":"+err.getMessage());
347                      System.exit(0);
348                  }
349
350               }
351             });
352         }
353 }
```

E.12: Main3.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5
6  public class Main3 {
7    Header  headerIns;
8    Provided providedIns;
9    Required requiredIns;
10   Binding bindingIns;
11   ObjRep objRepIns;
12   BindingRepr bindingReprIns;
13   ReJunction rejunctionIns;
14   Connection c;
15   Statement stmt;
16   public Main3()
17   {
18    deleteFiles();
19    headerIns =new Header();
20    writingComponent();
21    providedIns = new Provided();
22    requiredIns = new Required();
23    bindingIns = new Binding();
24    objRepIns= new ObjRep();
25    bindingReprIns = new BindingRepr();
26    rejunctionIns = new ReJunction();
27   }
28   public void deleteFiles()
29   {
30     File f1 =new File("Final.mxm");
31     f1.delete();
32     f1 =new File("Final2.mxm");
33     f1.delete();
34     f1 =new File("TEST1.xml");
35     f1.delete();
36     f1 =new File("TEST2.xml");
37     f1.delete();
38     f1 =new File("TEST3.xml");
39     f1.delete();
40     f1 =new File("TEST4.xml");
41     f1.delete();
42     f1 =new File("TEST5.xml");
43     f1.delete();
44     f1 =new File("TEST6.xml");
45     f1.delete();
46     f1 =new File("TEST7.xml");
47     f1.delete();
48
49   }
50   public static void main(String[] args)
51   {
52     new Main3();
53   }
54
55   public void writingComponent()
56   {
57
58     try {
59           Class.forName("org.sqlite.JDBC");
```

E.12: Main3.java dosyası (devam).

```
60          SQLiteConfig config = new SQLiteConfig();
61          config.enforceForeignKeys(true);
62          c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                            config.toProperties());
63         stmt = c.createStatement();
64
65          ResultSet rs1 = stmt.executeQuery("SELECT componentName FROM ComponentNames
                                            ");
66          int countFinal=0;
67          while (rs1.next()) {
68              String data1a = rs1.getString("componentName");
69              Statement stmt2 = c.createStatement();
70              try
71              {
72                 RandomAccessFile objects= null;
73                 String tmpStr="";
74                 objects = new RandomAccessFile("template-component/Component.xml",
                                                  "r");
75                 File file = new File("TEST1.xml");
76                 FileWriter fr = new FileWriter(file, true);
77                 BufferedWriter br = new BufferedWriter(fr);
78                 PrintWriter pr = new PrintWriter(br);
79
80                 while ((tmpStr = objects.readLine()) != null)
81              {
82                 tmpStr = tmpStr.replaceAll("COMPONENT-NAME-1",data1a);
83                 tmpStr = tmpStr.replaceAll("COMPONENT-NAME",data1a);
84
85                 pr.println(tmpStr);
86
87              }
88              stmt2.close();
89              objects.close();
90              pr.close();
91              br.close();
92              fr.close();
93            }
94            catch (IOException e)
95            {
96              e.printStackTrace();
97            }
98
99            Statement stmt3  = c.createStatement();
100           ResultSet rs3 = stmt3.executeQuery("SELECT parameterName, parameterType,
                                              parameterDefaultValue, componentName FROM
                                              ParameterList WHERE componentName='"
                                              +data1a+"';" );
101           while (rs3.next())
102           {
103             String data1b = rs3.getString("parameterName");
104             String data2b = rs3.getString("parameterType");
105             String data3b = rs3.getString("parameterDefaultValue");
106             String data4b = rs3.getString("componentName");
107
108
109
110            try
111            {
112             RandomAccessFile objects= null;
```

E.12: Main3.java dosyası (devam).

```
113             String tmpStr="";
114             objects = new RandomAccessFile("template-component/ComponentParamList.xml"
                                               ,"r");
115             File file = new File("TEST1.xml");
116             FileWriter fr = new FileWriter(file, true);
117             BufferedWriter br = new BufferedWriter(fr);
118             PrintWriter pr = new PrintWriter(br);
119             while ((tmpStr = objects.readLine()) != null)
120             {
121               tmpStr = tmpStr.replaceAll("PARAM-TYPE",data2b);
122               tmpStr = tmpStr.replaceAll("PARAM-NAME",data1b);
123               tmpStr = tmpStr.replaceAll("PARAM-VALUE",data3b);
124               pr.println(tmpStr);
125             }
126             objects.close();
127             pr.close();
128             br.close();
129             fr.close();
130
131            }
132            catch (IOException e)
133            {
134              e.printStackTrace();
135            }
136          }
137          stmt3.close();
138
139          try
140          {
141            RandomAccessFile objects= null;
142            String tmpStr="";
143            objects = new RandomAccessFile("template-component/Component2.xml", "r");
144            File file = new File("TEST1.xml");
145            FileWriter fr = new FileWriter(file, true);
146            BufferedWriter br = new BufferedWriter(fr);
147            PrintWriter pr = new PrintWriter(br);
148
149           while ((tmpStr = objects.readLine()) != null)
150           {
151             pr.println(tmpStr);
152           }
153           objects.close();
154           pr.close();
155           br.close();
156           fr.close();
157         }
158         catch (IOException e)
159         {
160          e.printStackTrace();
161         }
162
163         ResultSet rs2 = stmt2.executeQuery("SELECT dataName,dataType,
                                               dataDefaultValue, componentName ,arraySize
                                          FROM DataList WHERE componentName='"+data1a+"';");
164         while (rs2.next())
165         {
166           String data1b = rs2.getString("dataName");
167           String data2b = rs2.getString("dataType");
168           String data3b = rs2.getString("dataDefaultValue");
```

E.12: Main3.java dosyası (devam).

```
169             String data4b = rs2.getString("componentName");
170             String data5b = rs2.getString("arraySize");
171             try
172             {
173               RandomAccessFile objects= null;
174               String tmpStr="";
175               objects = new RandomAccessFile("templatecomponent/ComponentDataList.xml"
                                                 ,"r");
176               File file = new File("TEST1.xml");
177               FileWriter fr = new FileWriter(file, true);
178               BufferedWriter br = new BufferedWriter(fr);
179               PrintWriter pr = new PrintWriter(br);
180               while ((tmpStr = objects.readLine()) != null)
181               {
182                tmpStr = tmpStr.replaceAll("DATA-TYPE",data2b);
183                tmpStr = tmpStr.replaceAll("DATA-NAME",data1b);
184                tmpStr = tmpStr.replaceAll("DEFAULT-VALUE",data3b);
185                tmpStr = tmpStr.replaceAll("<string></string>",
                                              "<string>"+data5b+"</string>");
186                pr.println(tmpStr);
187               }
188               objects.close();
189               pr.close();
190               br.close();
191              fr.close();
192             }
193             catch (IOException e)
194             {
195               e.printStackTrace();
196             }
197           }
198
199           stmt2.close();
200           try
201           {
202            RandomAccessFile objects= null;
203            String tmpStr="";
204            objects = new RandomAccessFile("template-component/ComponentEnd.xml", "r");
205            File file = new File("TEST1.xml");
206            FileWriter fr = new FileWriter(file, true);
207            BufferedWriter br = new BufferedWriter(fr);
208            PrintWriter pr = new PrintWriter(br);
209
210            while ((tmpStr = objects.readLine()) != null)
211            {
212
213               pr.println(tmpStr);
214
215            }
216            objects.close();
217            pr.close();
218            br.close();
219            fr.close();
220           }
221           catch (IOException e)
222           {
223            e.printStackTrace();
224           }
225         }
```

E.12: Main3.java dosyası (devam).

```
226         stmt.close();
227         c.close();
228       }catch (Exception err)
229       {
230          System.err.println(err.getClass().getName() + ":"+err.getMessage());
231       }
232    }
233
234 }
```

E.13: Header.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5
6  public class Header {
7
8       Connection c;
9       Statement stmt;
10
11      public Header()
12      {
13              writingHeader();
14      }
15
16      public void writingHeader()
17      {
18          c = null;
19              stmt = null;
20
21              try
22              {
23                      RandomAccessFile objects= null;
24                      String tmpStr="";
25                      objects = new RandomAccessFile("template-header/Header.xml", "r");
26                      File file = new File("TEST1.xml");
27                      FileWriter fr = new FileWriter(file, true);
28                      BufferedWriter br = new BufferedWriter(fr);
29                      PrintWriter pr = new PrintWriter(br);
30                      while ((tmpStr = objects.readLine()) != null)
31                      {
32                              pr.println(tmpStr);
33                              //writer1.write(tmpStr);
34                      }
35                              objects.close();
36                              pr.close();
37                              br.close();
38                              fr.close();
39                      }
40                      catch (IOException e)
41                      {
42                              e.printStackTrace();
43                      }
44      }
```

E.14: Provided.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5
6  public class Provided {
7    Connection c;
8    Statement stmt;
9    public Provided()
10   {
11      writingComponent();
12   }
13   public static void main(String[] args)
14   {
15     new Provided();
16   }
17   public void writingComponent()
18   {
19    try {
20          Class.forName("org.sqlite.JDBC");
21          SQLiteConfig config = new SQLiteConfig();
22          config.enforceForeignKeys(true);
23          c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                            config.toProperties());
24         stmt = c.createStatement();
25
26          ResultSet rs1 = stmt.executeQuery("SELECT * FROM Ports WHERE
                                              portType='Provided' ");
27        while (rs1.next())
28        {
29          String data1a = rs1.getString("portType");
30          String data2a = rs1.getString("preCon1");
31          String data3a = rs1.getString("postCon1");
32          String data4a = rs1.getString("preCon2");
33          String data5a = rs1.getString("postCon2");
34          String data6a = rs1.getString("methodName");
35          String data7a = rs1.getString("compName");
36          String data8a="";
37          Statement stmt2 = c.createStatement();
38
39          ResultSet rs2 = stmt2.executeQuery("SELECT compName FROM ReqPorts
                                               WHERE methodName='"+data6a+"';");
40          while (rs2.next())
41          {
42             data8a = rs2.getString("compName");
43          }
44          stmt2.close();
45          try
46          {
47            RandomAccessFile objects= null;
48            String tmpStr="";
49            objects = new RandomAccessFile("template-provided/provided-entrance.xml",
                                             "r");
50            File file = new File("TEST2.xml");
51           FileWriter fr = new FileWriter(file,true);
52            BufferedWriter br = new BufferedWriter(fr);
53            PrintWriter pr = new PrintWriter(br);
54
55            while ((tmpStr = objects.readLine()) != null)
```

E.14: Provided.java dosyası (devam).

```
56              {
57                 tmpStr = tmpStr.replaceAll("PROVIDER-OBJECT-1",data7a+"-"+
                                                 data6a+"PROVIDER-OBJECT");
58                 tmpStr = tmpStr.replaceAll("SUB-GRAPH-1",data7a+"-"+
                                                 data8a+"-"+data6a+"SUB-GRAPH");
59                 tmpStr = tmpStr.replaceAll("LIFELINE-1",data7a+ "LIFELINE-1-
                                                 "+data6a);
60                 tmpStr = tmpStr.replaceAll("COMPONENT-NAME-PROVIDED", data7a);
61                 tmpStr = tmpStr.replaceAll("LIFELINE-2",data8a+"LIFELINE-2-"+data6a);
62                 tmpStr = tmpStr.replaceAll("COMPONENT-NAME-REQUIRED",data8a);
63                 pr.println(tmpStr);
64               }
65               objects.close();
66               pr.close();
67               br.close();
68               fr.close();
69            }
70            catch (IOException e)
71            {
72              e.printStackTrace();
73            }
74            String data9a="";
75            String data10a="";
76            String data11a="";
77            String data12a="";
78            Statement stmt3 = c.createStatement();
79            ResultSet rs3 = stmt3.executeQuery("SELECT * FROM Methods WHERE
                                                 Methods.componentName='"+ data7a +
                                       "' AND Methods.methodName='" + data6a+ "'");
80            while (rs3.next())
81            {
82
83             data9a=rs3.getString("parameterName");
84             data10a = rs3.getString("methodName");
85             data9a=data9a.trim();
86             if(data9a!="")
87             {
88               Statement stmt4 = c.createStatement();
89               ResultSet rs4 = stmt4.executeQuery("SELECT * FROM MethodParameter
                                                    WHERE componentName='"+data7a +
                                                "' AND parameterName='" + data9a + "'");

90              while (rs4.next())
91              {
92                data11a=rs4.getString("parameterType");
93                data12a = rs4.getString("parameterName");
94                try
95                {
96                    RandomAccessFile objects= null;
97                    String tmpStr="";
98                   objects = new RandomAccessFile("template-provided/
                                                     provided-parameters.xml", "r");
99                    File file = new File("./deneme/Test-Params.xml");
100                   FileWriter fr = new FileWriter(file, true);
101                   BufferedWriter br = new BufferedWriter(fr);
102                   PrintWriter pr = new PrintWriter(br);
103                   while ((tmpStr = objects.readLine()) != null)
104                   {
105                     tmpStr = tmpStr.replaceAll("PARAMETER-NAME",data12a);
```

E.14: Provided.java dosyası (devam).

```
106                     tmpStr = tmpStr.replaceAll("PARAMETER-TYPE",data11a);
107                     pr.println(tmpStr);
108                   }
109                   objects.close();
110                   pr.close();
111                   br.close();
112                   fr.close();
113                 }
114                 catch (IOException e)
115                 {
116                   e.printStackTrace();
117                 }
118               }
119               stmt4.close();
120             }
121             if(data9a.isEmpty() )
122             {
123               data11a="";
124               data12a="";
125               try
126               {
127                 RandomAccessFile objects= null;
128                 String tmpStr="";
129               objects = new RandomAccessFile("template-provided/provided-parameters.xml",
                                                 "r");
130               File file = new File("deneme/Test-Params.xml");
131               FileWriter fr = new FileWriter(file, true);
132               BufferedWriter br = new BufferedWriter(fr);
133               PrintWriter pr = new PrintWriter(br);
134
135
136               while ((tmpStr = objects.readLine()) != null)
137               {
138                 tmpStr = tmpStr.replaceAll("PARAMETER-NAME",data12a);
139                 tmpStr = tmpStr.replaceAll("PARAMETER-TYPE",data11a);
140                 pr.println(tmpStr);
141
142               }
143               objects.close();
144               pr.close();
145               br.close();
146               fr.close();
147             }
148             catch (IOException e)
149             {
150              e.printStackTrace();
151             }
152           }
153         }
154         stmt3.close();
155         try
156         {
157           RandomAccessFile objects= null;
158           String tmpStr="";
159           objects = new RandomAccessFile("template-provided/
                                              provided-binding-return-method-name.xml", "r");
160           File file = new File("TEST2.xml");
161          FileWriter fr = new FileWriter(file, true);
162           BufferedWriter br = new BufferedWriter(fr);
```

E.14: Provided.java dosyası (devam).

```
163         PrintWriter pr = new PrintWriter(br);
164
165         while ((tmpStr = objects.readLine()) != null)
166         {
167             tmpStr = tmpStr.replaceAll("SUB-BINDING-1",data7a+"-"+ data8a+
                                                "-"+data6a+"SUB-BINDING");
168             tmpStr = tmpStr.replaceAll("METHOD-NAME",data6a);
169             pr.println(tmpStr);
170         }
171         objects.close();
172         pr.close();
173         br.close();
174         fr.close();
175        }
176        catch (IOException e)
177        {
178          e.printStackTrace();
179        }
180        try
181        {
182          RandomAccessFile objects= null;
183          String tmpStr="";
184          objects = new RandomAccessFile("deneme/Test-Params.xml", "rw");
185          File file = new File("TEST2.xml");
186          FileWriter fr = new FileWriter(file, true);
187          BufferedWriter br = new BufferedWriter(fr);
188          PrintWriter pr = new PrintWriter(br);
189
190          while ((tmpStr = objects.readLine()) != null)
191          {
192            pr.println(tmpStr);
193          }
194          objects.close();
195          pr.close();
196          br.close();
197          fr.close();
198       }
199       catch (IOException e)
200       {
201          e.printStackTrace();
202       }
203       try
204       {
205         RandomAccessFile objects= null;
206         String tmpStr="";
207         objects=new RandomAccessFile("template-provided/provided-binding-return-
                                        end.xml","r");
208         File file = new File("TEST2.xml");
209         FileWriter fr = new FileWriter(file, true);
210         BufferedWriter br = new BufferedWriter(fr);
211         PrintWriter pr = new PrintWriter(br);
212
213         while ((tmpStr = objects.readLine()) != null)
214         {
215             tmpStr = tmpStr.replaceAll("LIFELINE-1",data7a+"LIFELINE-1-"+data6a);
216             tmpStr = tmpStr.replaceAll("LIFELINE-2",data8a+"LIFELINE-2-"+data6a);
217             tmpStr = tmpStr.replaceAll("RETURN-VALUE",data5a);
218             pr.println(tmpStr);
219         }
```

E.14: Provided.java dosyası (devam).

```
220         objects.close();
221         pr.close();
222         br.close();
223         fr.close();
224       }
225       catch (IOException e)
226       {
227         e.printStackTrace();
228       }
229       try
230       {
231         RandomAccessFile objects= null;
232         String tmpStr="";
233         object=new RandomAccessFile("template-provided/
                                        provided-binding-forward.xml","r");
234         File file = new File("TEST2.xml");
235         FileWriter fr = new FileWriter(file, true);
236         BufferedWriter br = new BufferedWriter(fr);
237         PrintWriter pr = new PrintWriter(br);
238
239         while ((tmpStr = objects.readLine()) != null)
240         {
241            tmpStr = tmpStr.replaceAll("SUB-BINDING-2",data8a+"-"+ data7a+"-"+data6a
                                          +"SUB-BINDING");
242            tmpStr = tmpStr.replaceAll("METHOD-NAME",data6a);
243            pr.println(tmpStr);
244         }
245         objects.close();
246         pr.close();
247         br.close();
248         fr.close();
249       }
250       catch (IOException e)
251       {
252         e.printStackTrace();
253       }
254       try
255       {
256         RandomAccessFile objects= null;
257         String tmpStr="";
258         objects = new RandomAccessFile("deneme/Test-Params.xml", "rw");
259         File file = new File("TEST2.xml");
260         FileWriter fr = new FileWriter(file, true);
261         BufferedWriter br = new BufferedWriter(fr);
262         PrintWriter pr = new PrintWriter(br);
263
264         while ((tmpStr = objects.readLine()) != null)
265         {
266            pr.println(tmpStr);
267         }
268         objects.close();
269         pr.close();
270         br.close();
271         fr.close();
272        File myObj = new File("deneme/Test-Params.xml");
273         if(myObj.exists() && !myObj.isDirectory())
274         {
275
276            if (myObj.delete()) {
```

E.14: Provided.java dosyası (devam).

```
277              System.out.println("Deleted the file: " + myObj.getName());
278            }
279            else
280            {
281             System.out.println("Failed to delete the file.");
282            }
283          }
284        }
285        catch (IOException e)
286       {
287          e.printStackTrace();
288       }
289       try
290       {
291         RandomAccessFile objects= null;
292         String tmpStr="";
293         objects=new RandomAccessFile("template-provided/provided-binding-forward-
                                        end.xml","r");
294         File file = new File("TEST2.xml");
295         FileWriter fr = new FileWriter(file, true);
296         BufferedWriter br = new BufferedWriter(fr);
297         PrintWriter pr = new PrintWriter(br);
298
299         while ((tmpStr = objects.readLine()) != null)
300         {
301           tmpStr = tmpStr.replaceAll("LIFELINE-1",data7a+"LIFELINE-1-"+data6a);
302           tmpStr = tmpStr.replaceAll("LIFELINE-2",data8a+"LIFELINE-2-"+data6a);
303           tmpStr = tmpStr.replaceAll("PRE-CONDITION",data2a);
304           tmpStr = tmpStr.replaceAll("POST-CONDITION",data3a);
305
306           pr.println(tmpStr);
307
308          }
309          objects.close();
310          pr.close();
311          br.close();
312          fr.close();
313       }
314       catch (IOException e)
315       {
316         e.printStackTrace();
317       }
318       try
319       {
320         RandomAccessFile objects= null;
321         String tmpStr="";
322         objects=new RandomAccessFile("template-provided/
                                        provided-diyagram-entrance.xml","r");
323         File file = new File("TEST2.xml");
324         FileWriter fr = new FileWriter(file, true);
325         BufferedWriter br = new BufferedWriter(fr);
326         PrintWriter pr = new PrintWriter(br);
327         while ((tmpStr = objects.readLine()) != null)
328         {
329          tmpStr = tmpStr.replaceAll("SUB-GRAPH-1",data7a+"-"+ data8a+"-"+data6a+
                                          "SUB-GRAPH");
330          tmpStr = tmpStr.replaceAll("LIFELINE-2",data8a+"LIFELINE-2-"+data6a);
331          tmpStr = tmpStr.replaceAll("objcetRep-2",data8a+"-OBJECTREPR-2-"+data6a);
332
```

E.14: Provided.java dosyası (devam).

```
333         tmpStr = tmpStr.replaceAll("objcetRep-1",data7a+"-OBJECTREPR-1-"+data6a);
334         tmpStr = tmpStr.replaceAll("LIFELINE-1",data7a+"LIFELINE-1-"+data6a);
335         pr.println(tmpStr);
336        }
337        objects.close();
338       pr.close();
339       br.close();
340       fr.close();
341     }
342     catch (IOException e)
343     {
344        e.printStackTrace();
345     }
346     try
347     {
348       RandomAccessFile objects= null;
349       String tmpStr="";
350       objects=new RandomAccessFile("template-provided/
                                      provided-bindingRepr-forward.xml","r");
351       File file = new File("TEST2.xml");
352      FileWriter fr = new FileWriter(file, true);
353       BufferedWriter br = new BufferedWriter(fr);
354       PrintWriter pr = new PrintWriter(br);
355
356       while ((tmpStr = objects.readLine()) != null)
357       {
358        tmpStr = tmpStr.replaceAll("SUB-BINDING-2",data8a+"-"+ data7a+"-"+data6a+
                                      "SUB-BINDING");
359        tmpStr = tmpStr.replaceAll("LIFELINE-1-OBJECTREPR",data7a+
                                      "-OBJECTREPR-1-"+data6a);
360        tmpStr = tmpStr.replaceAll("LIFELINE-2-OBJECTREPR",data8a+
                                      "-OBJECTREPR-2-"+data6a);
361        pr.println(tmpStr);
362       }
363       objects.close();
364       pr.close();
365       br.close();
366       fr.close();
367     }
368     catch (IOException e)
369     {
370       e.printStackTrace();
371     }
372     try
373     {
374       RandomAccessFile objects= null;
375       String tmpStr="";
376       objects=new RandomAccessFile("template-provided/
                                      provided-bindingRepr-return.xml","r");
377       File file = new File("TEST2.xml");
378       FileWriter fr = new FileWriter(file, true);
379       BufferedWriter br = new BufferedWriter(fr);
380       PrintWriter pr = new PrintWriter(br);
381
382       while ((tmpStr = objects.readLine()) != null)
383       {
384         tmpStr =tmpStr.replaceAll("SUB-BINDING-1",data7a+"-"+ data8a+"-"+data6a+
                                      "SUB-BINDING");
385         tmpStr =tmpStr.replaceAll("LIFELINE-1-OBJECTREPR",data7a+
```

E.14: Provided.java dosyası (devam).

```
                                                "-OBJECTREPR-1-"+data6a);
386         tmpStr =tmpStr.replaceAll("LIFELINE-2-OBJECTREPR",data8a+
                                                "-OBJECTREPR-2-"+data6a);
387         pr.println(tmpStr);
388       }
389       objects.close();
390       pr.close();
391       br.close();
392       fr.close();
393     }
394     catch (IOException e)
395     {
396      e.printStackTrace();
397     }
398     try
399     {
400       RandomAccessFile objects= null;
401       String tmpStr="";
402       objects = new RandomAccessFile("template-provided/provided-diyagram-end.xml",
"r");
403       File file = new File("TEST2.xml");
404       FileWriter fr = new FileWriter(file, true);
405       BufferedWriter br = new BufferedWriter(fr);
406       PrintWriter pr = new PrintWriter(br);
407
408      while ((tmpStr = objects.readLine()) != null)
409      {
410        pr.println(tmpStr);
411      }
412      objects.close();
413      pr.close();
414      br.close();
415      fr.close();
416     }
417     catch (IOException e)
418     {
419       e.printStackTrace();
420     }
421    }
422    stmt.close();
423    c.close();
424   }catch (Exception err)
425   {
426     System.err.println(err.getClass().getName() + ":"+ err.getMessage());
427   }
428 }
429}
```

E.15: Required.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5  import java.nio.file.*;
6
7  public class Required {
8    Connection c;
9    Statement stmt;
10   RandomAccessFile objects = null;
11   public Required() {
12     writingComponent();
13
14    }
15
16   public static void main(String[] args) {
17     new Required();
18   }
19
20   public void writingComponent() {
21     try {
22           Class.forName("org.sqlite.JDBC");
23           SQLiteConfig config = new SQLiteConfig();
24           config.enforceForeignKeys(true);
25           c=DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                           config.toProperties());
26
27           Statement stmt5 = c.createStatement();
28           ResultSet rs5 = stmt5.executeQuery("SELECT * FROM ReqCompMth  ");
29           while (rs5.next())
30           {
31             String data1b = rs5.getString("methodName");
32             stmt = c.createStatement();
33
34             ResultSet rs1 = stmt.executeQuery("SELECT * FROM ReqPorts WHERE
                                                 methodName='" + data1b + "';");
35
36            while (rs1.next())
37            {
38              String data2a = rs1.getString("preCon1");
39              String data3a = rs1.getString("postCon1");
40              String data6a = rs1.getString("methodName");
41              String data7a = rs1.getString("compName");
42              String data8a = "";
43              Statement stmt2 = c.createStatement();
44
45              ResultSet rs2 = stmt2.executeQuery("SELECT compName FROM Ports WHERE
                                portType='Provided' AND methodName='" + data6a + "';");
46              while (rs2.next()) {
47                    data8a = rs2.getString("compName");
48              }
49              stmt2.close();
50
51              try {
52                 objects = null;
53                 String tmpStr = "";
54                 objects =new RandomAccessFile("template-required/
                                                  required-entrance.xml","r");
55                 File file = new File("TEST3.xml");
```

E.15: Required.java dosyası (devam).

```
56                  FileWriter fr = new FileWriter(file, true);
57                  BufferedWriter br = new BufferedWriter(fr);
58                  PrintWriter pr = new PrintWriter(br);
59
60                  while ((tmpStr = objects.readLine()) != null) {
61                    tmpStr = tmpStr.replaceAll("REQUIRED-OBJECT-1",data7a +
                                                "-" + data6a +  "REQUIRED-OBJECT");
62                    tmpStr = tmpStr.replaceAll("SUB-GRAPH-2",data7a + "-" + data8a +
                                                "-" + data6a + "SUB-GRAPH");
63                    tmpStr = tmpStr.replaceAll("LIFELINE-3", data8a + "LIFELINE-3-" +
                                                 data6a);
64                    tmpStr = tmpStr.replaceAll("COMPONENT-NAME-PROVIDED", data8a);
65                    tmpStr = tmpStr.replaceAll("LIFELINE-4", data7a + "LIFELINE-4-"+
                                                data6a);
66                    tmpStr = tmpStr.replaceAll("COMPONENT-NAME-REQUIRED", data7a);
67                    pr.println(tmpStr);
68                  }
69                  objects.close();
70                  pr.close();
71                  br.close();
72                  fr.close();
73              } catch (IOException e) {
74                  e.printStackTrace();
75              }
76              String data9a = "";
77              String data10a = "";
78              String data11a = "";
79              String data12a = "";
80              Statement stmt3 = c.createStatement();
81
82              ResultSet rs3 = stmt3.executeQuery("SELECT * FROM Methods WHERE
                                        Methods.componentName='" +data7a+ "' AND
                                        Methods.methodName='"+ data6a + "'");
83              while (rs3.next()) {
84                data9a = rs3.getString("parameterName");
85                data10a = rs3.getString("methodName");
86                data9a = data9a.trim();
87                if (data9a != "") {
88                  Statement stmt4 = c.createStatement();
89                  ResultSet rs4 = stmt4.executeQuery("SELECT * FROM MethodParameter
                                            WHERE componentName='"+data7a+
                                          "' AND parameterName='"+data9a+"'");
90                  while (rs4.next()) {
91                      data11a = rs4.getString("parameterType");
92                      data12a = rs4.getString("parameterName");
93                      try {
94                          RandomAccessFile objects = null;
95                          String tmpStr = "";
96                          objects = new RandomAccessFile("template-required/required-
                                                    parameters.xml", "r");
97                          File file = new File("deneme2/Test-Params.xml");
98                          FileWriter fr = new FileWriter(file, true);
99                          BufferedWriter br = new BufferedWriter(fr);
100                         PrintWriter pr = new PrintWriter(br);
101                         while ((tmpStr = objects.readLine()) != null) {
102                             tmpStr = tmpStr.replaceAll("PARAMETER-NAME",
                                                        data12a);
103                             tmpStr = tmpStr.replaceAll("PARAMETER-TYPE",
                                                        data11a);
```

E.15: Required.java dosyası (devam).

```
104                             pr.println(tmpStr);
105                           }
106                           objects.close();
107                           pr.close();
108                           br.close();
109                           fr.close();
110                         } catch (IOException e) {
111                            e.printStackTrace();
112                         }
113                       }
114                       stmt4.close();
115                     }
116                     if (data9a.isEmpty()) {
117                       data11a = "";
118                       data12a = "";
119
120                       try {
121                          objects = null;
122                          String tmpStr = "";
123                          objects = new RandomAccessFile("template-required/
                                                          required-parameters.xml", "r");
124                          File file = new File("deneme2/Test-Params.xml");
125                          FileWriter fr = new FileWriter(file, true);
126                          BufferedWriter br = new BufferedWriter(fr);
127                          PrintWriter pr = new PrintWriter(br);
128                          while ((tmpStr = objects.readLine()) != null) {
129                            tmpStr = tmpStr.replaceAll("PARAMETER-NAME", data12a);
130                            tmpStr = tmpStr.replaceAll("PARAMETER-TYPE", data11a);
131                             pr.println(tmpStr);
132                          }
133                          objects.close();
134                          pr.close();
135                          br.close();
136                          fr.close();
137                       } catch (IOException e) {
138                          e.printStackTrace();
139                       }
140                     }
141
142                 }
143                 stmt3.close();
144                 Statement stmt6 = c.createStatement();
145                 ResultSet rs6 = stmt6.executeQuery("SELECT * FROM ReqPostCond  WHERE
                                                      methodName='" + data1b + "';");
146                 int count=0;
147                 while (rs6.next()) {
148                    count++;
149                    String data4a = rs6.getString("preCon2");
150                    String data5a = rs6.getString("postCon2");
151
152                    try {
153                     objects = null;
154                     String tmpStr = "";
155                     objects = new RandomAccessFile("template-required/
                                              required-binding-return-method-name.xml","r");
156                     File file = new File("TEST3.xml");
157                     FileWriter fr = new FileWriter(file, true);
158                     BufferedWriter br = new BufferedWriter(fr);
159                     PrintWriter pr = new PrintWriter(br);
```

E.15: Required.java dosyası (devam).

```
160
161                     while ((tmpStr = objects.readLine()) != null) {
162                         if(count>1)
163                             tmpStr = tmpStr.replaceAll("SUB-BINDING-3",data7a +
                                                           "-" + data8a + "-" + data6a +
                                                           "SUB-BINDING-REQ-NEXT"+count);
164                         else
165                            tmpStr = tmpStr.replaceAll("SUB-BINDING-3",data7a + "-" +
                                                  data8a + "-" + data6a + "SUB-BINDING-REQ");
166                         tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
167                         pr.println(tmpStr);
168                     }
169                     objects.close();
170                     pr.close();
171                     br.close();
172                     fr.close();
173                 } catch (IOException e) {
174                     e.printStackTrace();
175                 }
176
177                 try {
178                  objects = null;
179                  String tmpStr = "";
180                  objects = new RandomAccessFile("deneme2/Test-Params.xml", "rw");
181                  File file = new File("TEST3.xml");
182                  FileWriter fr = new FileWriter(file, true);
183                  BufferedWriter br = new BufferedWriter(fr);
184                  PrintWriter pr = new PrintWriter(br);
185                  while ((tmpStr = objects.readLine()) != null) {
186                     pr.println(tmpStr);
187                  }
188                  objects.close();
189                  pr.close();
190                  br.close();
191                  fr.close();
192
193                 } catch (IOException e) {
194                    e.printStackTrace();
195                 }
196
197                 try {
198                  objects = null;
199                  String tmpStr = "";
200                  objects = new RandomAccessFile("template-required/
                                                  required-binding-return-end.xml", "r");
201                  File file = new File("TEST3.xml");
202                  FileWriter fr = new FileWriter(file, true);
203                  BufferedWriter br = new BufferedWriter(fr);
204                  PrintWriter pr = new PrintWriter(br);
205
206                  while ((tmpStr = objects.readLine()) != null) {
207                    tmpStr = tmpStr.replaceAll("PRE-CONDITION", data4a);
208                    tmpStr = tmpStr.replaceAll("POST-CONDITION", data5a);
209                    tmpStr = tmpStr.replaceAll("LIFELINE-3", data8a + "LIFELINE-3-
                                                      "+data6a);
210                    tmpStr = tmpStr.replaceAll("LIFELINE-4", data7a + "LIFELINE-4-
                                                      "+data6a);
211                    pr.println(tmpStr);
212                  }
```

E.15: Required.java dosyası (devam).

```
213                     objects.close();
214                     pr.close();
215                     br.close();
216                     fr.close();
217                    } catch (IOException e) {
218                      e.printStackTrace();
219                    }
220                  }
221                  stmt6.close();
222
223                  try {
224                   objects = null;
225                   String tmpStr = "";
226                   objects = new RandomAccessFile("template-required/required-binding-
                                                      forward.xml", "r");
227                   File file = new File("TEST3.xml");
228                   FileWriter fr = new FileWriter(file, true);
229                   BufferedWriter br = new BufferedWriter(fr);
230                   PrintWriter pr = new PrintWriter(br);
231
232                   while ((tmpStr = objects.readLine()) != null) {
233                      tmpStr = tmpStr.replaceAll("SUB-BINDING-4",data8a + "-" +
                                                    data7a + "-"+ data6a + "SUB-BINDING-REQ");
234                      tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
235                      pr.println(tmpStr);
236                   }
237                  objects.close();
238                  pr.close();
239                  br.close();
240                  fr.close();
241                } catch (IOException e) {
242                     e.printStackTrace();
243                }
244
245                try {
246                  objects = null;
247                  String tmpStr = "";
248                  objects = new RandomAccessFile("deneme2/Test-Params.xml", "rw");
249                  File file = new File("TEST3.xml");
250                  FileWriter fr = new FileWriter(file, true);
251                  BufferedWriter br = new BufferedWriter(fr);
252                  PrintWriter pr = new PrintWriter(br);
253
254                  while ((tmpStr = objects.readLine()) != null) {
255                      pr.println(tmpStr);
256                  }
257                  objects.close();
258                  pr.close();
259                  br.close();
260                  fr.close();
261
262                  File myObj = new File("deneme2/Test-Params.xml");
263                  if (myObj.exists() && !myObj.isDirectory()) {
264
265                    if (myObj.delete()) {
266                      System.out.println("Deleted the file: " + myObj.getName());
267                    } else {
268                      System.out.println("Failed to delete the file.");
269                    }
```

E.15: Required.java dosyası (devam).

```
270                 }
271
272
273               } catch (IOException e) {
274                   e.printStackTrace();
275               }
276
277               try {
278                 objects = null;
279                 String tmpStr = "";
280                 objects = new RandomAccessFile("template-required/
                                                    required-binding-forward-end.xml", "r");
281                 File file = new File("TEST3.xml");
282                 FileWriter fr = new FileWriter(file, true);
283                 BufferedWriter br = new BufferedWriter(fr);
284                 PrintWriter pr = new PrintWriter(br);
285
286                 while ((tmpStr = objects.readLine()) != null) {
287                     tmpStr = tmpStr.replaceAll("LIFELINE-3", data8a + "LIFELINE-3-"+
                                                      data6a);
288                     tmpStr = tmpStr.replaceAll("LIFELINE-4", data7a + "LIFELINE-4-" +
                                                      data6a);
289                     tmpStr = tmpStr.replaceAll("PRE-CONDITION", data2a);
290                     tmpStr = tmpStr.replaceAll("PROMISE", data3a);
291
292                     pr.println(tmpStr);
293                 }
294                 objects.close();
295                 pr.close();
296                 br.close();
297                 fr.close();
298               } catch (IOException e) {
299                  e.printStackTrace();
300               }
301
302               try {
303                 objects = null;
304                 String tmpStr = "";
305                  objects = new RandomAccessFile("template-required/
                                                     required-diyagram-entrance.xml","r");
306                  File file = new File("TEST3.xml");
307                  FileWriter fr = new FileWriter(file, true);
308                  BufferedWriter br = new BufferedWriter(fr);
309                  PrintWriter pr = new PrintWriter(br);
310
311                  while ((tmpStr = objects.readLine()) != null) {
312                    tmpStr = tmpStr.replaceAll("SUB-GRAPH-2",data7a + "-" + data8a +
                                                     "-" + data6a + "SUB-GRAPH");
313                    tmpStr = tmpStr.replaceAll("objcetRep-3", data8a +
                                                     "-OBJECTREPR-3-"+data6a);
314                    tmpStr = tmpStr.replaceAll("LIFELINE-3", data8a + "LIFELINE-3-"+
                                                     data6a);
315                    tmpStr = tmpStr.replaceAll("objcetRep-4", data7a +
                                                     "-OBJECTREPR-4-"+ data6a);
316                    tmpStr = tmpStr.replaceAll("LIFELINE-4", data7a + "LIFELINE-4-"+
                                                     data6a);
317                    pr.println(tmpStr);
318                   }
319                  objects.close();
```

E.15: Required.java dosyası (devam).

```
320                 pr.close();
321                 br.close();
322                 fr.close();
323               } catch (IOException e) {
324                   e.printStackTrace();
325               }
326
327               try {
328                 objects = null;
329                 String tmpStr = "";
330                 objects = new RandomAccessFile("template-required/
                                                    required-bindingRepr-forward.xml", "r");
331                 File file = new File("TEST3.xml");
332                 FileWriter fr = new FileWriter(file, true);
333                 BufferedWriter br = new BufferedWriter(fr);
334                 PrintWriter pr = new PrintWriter(br);
335
336                 while ((tmpStr = objects.readLine()) != null) {
337                     tmpStr = tmpStr.replaceAll("SUB-BINDING-4",data8a + "-" + data7a
                                                        + "-" + data6a + "SUB-BINDING-REQ");
338                     tmpStr = tmpStr.replaceAll("LIFELINE-3-OBJECTREPR", data8a +
                                                        "-OBJECTREPR-3-"+data6a);
339                     tmpStr = tmpStr.replaceAll("LIFELINE-4-OBJECTREPR", data7a +
                                                        "-OBJECTREPR-4-"+data6a);
340                     pr.println(tmpStr);
341                 }
342                 objects.close();
343                 pr.close();
344                 br.close();
345                 fr.close();
346               } catch (IOException e) {
347                  e.printStackTrace();
348               }
349               stmt6 = c.createStatement();
350               rs6 = stmt6.executeQuery("SELECT * FROM ReqPostCond  WHERE  methodName='"
                                        + data1b + "';");
351               count=0;
352               int x=295;
353               int y=220;
354               int dy=40;
355
356               int offSetY=80;
357               int dyOffSetY=60;
358               while (rs6.next())
359               {
360                 count++;
361                 if (count>1)
362                 {
363                     y=y+dy;
364                     offSetY=offSetY+dyOffSetY;
365                 }
366
367                 try {
368                  objects = null;
369                  String tmpStr = "";
370                  objects = new RandomAccessFile("template-required/required-bindingRepr-
                                                    return.xml", "r");
371                  File file = new File("TEST3.xml");
372                  FileWriter fr = new FileWriter(file, true);
```

E.15: Required.java dosyası (devam).

```
373                 BufferedWriter br = new BufferedWriter(fr);
374                 PrintWriter pr = new PrintWriter(br);
375 
376                 while ((tmpStr = objects.readLine()) != null) {
377                   if(count>1)
378                   {
379                     tmpStr = tmpStr.replaceAll("SUB-BINDING-3",data7a + "-" + data8a
                                          + "-"+ data6a + "SUB-BINDING-REQ-NEXT"+count);
380                     tmpStr = tmpStr.replaceAll("place=\"295,220\"",
                                                "place=\""+x+","+y+"\"");
381                     tmpStr = tmpStr.replaceAll("objTargetOffset=\"-10,80\"",
                                         "objTargetOffset=\""+(-10)+","+offSetY+"\"");
382                     tmpStr = tmpStr.replaceAll("objTargetOffset=\"30,80\"",
                                         "objTargetOffset=\""+30+","+offSetY+"\"");
383                   }
384                   else
385                     tmpStr = tmpStr.replaceAll("SUB-BINDING-3",data7a + "-" + data8a
                                                 + "-" + data6a + "SUB-BINDING-REQ");
386 
387                     tmpStr = tmpStr.replaceAll("LIFELINE-3-OBJECTREPR", data8a + "-
                                                OBJECTREPR-3-"+data6a);
388                     tmpStr = tmpStr.replaceAll("LIFELINE-4-OBJECTREPR", data7a + "-
                                                OBJECTREPR-4-"+data6a);
389                   pr.println(tmpStr);
390                 }
391                 objects.close();
392                 pr.close();
393                 br.close();
394                 fr.close();
395               } catch (IOException e) {
396                   e.printStackTrace();
397               }
398             }
399             stmt6.close();
400 
401             try {
402               objects = null;
403               String tmpStr = "";
404               objects=new RandomAccessFile("template-required/
                                              required-diyagram-end.xml", "r");
405               File file = new File("TEST3.xml");
406               FileWriter fr = new FileWriter(file, true);
407               BufferedWriter br = new BufferedWriter(fr);
408               PrintWriter pr = new PrintWriter(br);
409 
410               while ((tmpStr = objects.readLine()) != null) {
411                    pr.println(tmpStr);
412               }
413               objects.close();
414               pr.close();
415               br.close();
416               fr.close();
417             } catch (IOException e) {
418                   e.printStackTrace();
419             }
420           }
421           stmt.close();
422         }
423         stmt5.close();
```

E.15: Required.java dosyası (devam).

```
424         c.close();
425       } catch (Exception err) {
426           System.err.println(err.getClass().getName() + ": " + err.getMessage());
427       }
428 
429   }
430 
431 }
432 
```

E.16: ObjRep.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5  import java.nio.file.*;
6  
7  public class ObjRep {
8  
9   public ObjRep() {
10     writingComponents();
11     writingBinding1();
12     writingBinding2();
13  }
14 
15  public static void main(String[] args) {
16            new ObjRep();
17  }
18 
19  public void writingComponents()
20  {
21     Connection c;
22     Statement stmt;
23     try
24     {
25       Class.forName("org.sqlite.JDBC");
26       SQLiteConfig config = new SQLiteConfig();
27       config.enforceForeignKeys(true);
28       c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                         config.toProperties());
29       int x=200;
30       int y=200;
31       int count=0;
32       int dx= 400;
33       int dy= 350;
34       Statement stmt5 = c.createStatement();
35       Statement stmt6 = c.createStatement();
36       ResultSet rs5 = stmt5.executeQuery("SELECT * FROM ComponentNames  ");
37       while (rs5.next())
38       {
39         if (count>0)
40         {
41             if(count%2==1)
42                x=x+dx;
```

E.16: ObjRep.java dosyası (devam).

```
43              if(count%2==0)
44                  y=y+dy;
45          }
46
47          count++;
48          String data1b = rs5.getString("componentName");
49          stmt = c.createStatement();
50          try {
51              RandomAccessFile objects = null;
52              String tmpStr = "";
53              objects = new RandomAccessFile("template-object-rep/object-rep.xml", "r");
54              File file = new File("TEST5.xml");
55              FileWriter fr = new FileWriter(file, true);
56              BufferedWriter br = new BufferedWriter(fr);
57              PrintWriter pr = new PrintWriter(br);
58
59              while ((tmpStr = objects.readLine()) != null) {
60                  tmpStr = tmpStr.replaceAll("OBJREPR",data1b +"-OBJREPR");
61                  tmpStr = tmpStr.replaceAll("COMPONENT-NAME", data1b);
62              tmpStr = tmpStr.replaceAll("place=\"240,110\"",
                                           "place=\""+x+","+y+"\"");
63              pr.println(tmpStr);
64
65            }
66            objects.close();
67            pr.close();
68            br.close();
69            fr.close();
70          } catch (IOException e) {
71              e.printStackTrace();
72          }
73          String sql = "UPDATE ReqPorts SET compNameCoordinateX='" +x+
                         "' WHERE compName='"+ data1b + "';";
74          stmt6.executeUpdate(sql);
75
76          sql="UPDATE ReqPorts SET compNameCoordinateY='" +y+"' WHERE compName='"+
                data1b + "';";
77          stmt6.executeUpdate(sql);
78
79          sql ="UPDATE Ports SET compNameCoordinateX='" +x+"'  WHERE compName='"+
                data1b + "';";
80          stmt6.executeUpdate(sql);
81
82          sql = "UPDATE Ports SET compNameCoordinateY='" +y+"' WHERE compName='"+
                 data1b + "';";
83          stmt6.executeUpdate(sql);
84          stmt6.close();
85        }
86        stmt5.close();
87
88       c.close();
89       } catch (Exception err) {
90           System.err.println(err.getClass().getName() + ": " + err.getMessage());
91      }
92  }
93
94  public void writingBinding1() {
95    Connection c;
96    Statement stmt;
```

E.16: ObjRep.java dosyası (devam).

```
97    int x=0;
98    int y=0;
99    try
100   {
101
102    Class.forName("org.sqlite.JDBC");
103    SQLiteConfig config = new SQLiteConfig();
104    config.enforceForeignKeys(true);
105    c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db", config.toProperties());
106   Statement stmt6 = c.createStatement();
107    Statement stmt5 = c.createStatement();
108    ResultSet rs5 = stmt5.executeQuery("SELECT * FROM ReqCompMth  ");
109    while (rs5.next())
110    {
111     String data1b = rs5.getString("methodName");
112     stmt = c.createStatement();
113
114     ResultSet rs1=stmt.executeQuery("SELECT * FROM ReqPorts WHERE
                                        methodName='"+data1b+"';");
115
116
117     while (rs1.next()) {
118
119      String data2a = rs1.getString("preCon1");
120      String data3a = rs1.getString("postCon1");
121
122      String data6a = rs1.getString("methodName");
123      String data7a = rs1.getString("compName");
124      String data8a = "";
125
126      int data1c = Integer.parseInt(rs1.getString("compNameCoordinateX"));
127      int data2c = Integer.parseInt(rs1.getString("compNameCoordinateY"));
128      x=data1c;
129      y=data2c;
130      int data3c=0;
131      int data4c=0;
132     Statement stmt2 = c.createStatement();
133
134     ResultSet rs2 = stmt2.executeQuery("SELECT compName, compNameCoordinateX,
                     compNameCoordinateY FROM Ports WHERE portType='Provided'
                     AND methodName='" + data6a + "';");
135
136     while (rs2.next()) {
137      data8a = rs2.getString("compName");
138      data3c = Integer.parseInt(rs2.getString("compNameCoordinateX"));
139      data4c = Integer.parseInt(rs2.getString("compNameCoordinateY"));
140
141     }
142     stmt2.close();
143
144     if((data2c==data4c) && ( data1c < data3c) )
145     {
146       x=data1c+150;
147       y=data2c;
148     }
149     else if((data2c==data4c) && ( data1c > data3c) )
150     {
151       x=data1c-180;
152       y=data2c;
```

E.16: ObjRep.java dosyası (devam).

```
153     }
154     else if((data1c==data3c) && ( data2c < data4c))
155     {
156       x=data1c;
157       y=data2c+120;
158     }
159     else if((data1c==data3c) && ( data2c > data4c))
160     {
161       x=data1c;
162       y=data2c-180;
163     }
164     else if( (data2c <data4c)&& ( data1c < data3c) )
165     {
166       y=data4c;
167       x=data1c+150;
168     }
169     else if( (data2c <data4c)&& ( data1c > data3c) )
170     {
171       y=data2c+80;
172       x=data1c-80;
173     }
174     else if( (data2c > data4c)&& ( data1c > data3c) )
175     {
176       y=data2c;
177       x=data1c-180;
178     }
179     else if ((data2c >data4c) && ( data1c < data3c))
180     {
181       y=data2c;
182       x=data1c+80;
183     }
184
185     try {
186       RandomAccessFile objects = null;
187       String tmpStr = "";
188       objects = new RandomAccessFile("template-object-rep/pro-req-object.xml", "r");
189       File file = new File("TEST5.xml");
190       FileWriter fr = new FileWriter(file, true);
191       BufferedWriter br = new BufferedWriter(fr);
192       PrintWriter pr = new PrintWriter(br);
193
194
195       while ((tmpStr = objects.readLine()) != null) {
196        tmpStr = tmpStr.replaceAll("OBJREPR",data7a + "-" + data6a +
                                          "REQUIRED-OBJECT"+"- OBJREPR");
197        tmpStr = tmpStr.replaceAll("OBJ-PROVIDED-REQUIRED", data7a + "-" + data6a +
                                        "REQUIRED-OBJECT");
198        tmpStr = tmpStr.replaceAll("place=\"240,110\"", "place=\""+x+","+y+"\"");
199           pr.println(tmpStr);
200
201         }
202         objects.close();
203         pr.close();
204         br.close();
205         fr.close();
206
207      } catch (IOException e) {
208           e.printStackTrace();
209      }
```

E.16: ObjRep.java dosyası (devam).

```
210
211       String sql = "UPDATE ReqPorts SET methNameCoordinateX='" +x+
                       "' WHERE methodName='"+ data1b + "';";
212       stmt6.executeUpdate(sql);
213
214       sql = "UPDATE ReqPorts SET methNameCoordinateY='" +y+"' WHERE methodName='"+
                 data1b + "';";
215       stmt6.executeUpdate(sql);
216       stmt6.close();
217
218      }
219
220      stmt.close();
221    }
222    stmt5.close();
223    c.close();
224  } catch (Exception err) {
225     System.err.println(err.getClass().getName() + ": " + err.getMessage());
226  }
227 }
228
229 public void writingBinding2()
230 {
231    Connection c;
232    Statement stmt;
233    int x=0;
234    int y=0;
235    try
236    {
237      Class.forName("org.sqlite.JDBC");
238      SQLiteConfig config = new SQLiteConfig();
239      config.enforceForeignKeys(true);
240      c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",config.toProperties());
241      stmt = c.createStatement();
242      Statement stmt6 = c.createStatement();
243      ResultSet rs1 = stmt.executeQuery("SELECT * FROM Ports WHERE portType='Provided'
                                          ");
244
245      while (rs1.next())
246      {
247        String data1a = rs1.getString("portType");
248        String data2a = rs1.getString("preCon1");
249        String data3a = rs1.getString("postCon1");
250        String data4a = rs1.getString("preCon2");
251        String data5a = rs1.getString("postCon2");
252        String data6a = rs1.getString("methodName");
253        String data7a = rs1.getString("compName");
254        String data8a="";
255
256        int data1c = Integer.parseInt(rs1.getString("compNameCoordinateX"));
257        int data2c = Integer.parseInt(rs1.getString("compNameCoordinateY"));
258        x=data1c;
259        y=data2c;
260        int data3c=0;
261        int data4c=0;
262        Statement stmt2 = c.createStatement();
263
264        ResultSet rs2 = stmt2.executeQuery("SELECT compName, compNameCoordinateX,
               compNameCoordinateY FROM ReqPorts WHERE  methodName='"+data6a+"';");
```

E.16: ObjRep.java dosyası (devam).

```
265         while (rs2.next())
266         {
267          data8a = rs2.getString("compName");
268          data3c = Integer.parseInt(rs2.getString("compNameCoordinateX"));
269          data4c = Integer.parseInt(rs2.getString("compNameCoordinateY"));
270
271         }
272         stmt2.close();
273
274        if((data2c==data4c) && ( data1c < data3c) )
275         {
276            x=data1c+100;
277            y=data2c;
278         }
279         else if((data2c==data4c) && ( data1c > data3c) )
280         {
281            x=data1c-200;
282            y=data2c;
283         }
284         else if((data1c==data3c) && ( data2c < data4c))
285         {
286            x=data1c;
287            y=data2c+80;
288         }
289         else if((data1c==data3c) && ( data2c > data4c))
290         {
291            x=data1c;
292            y=data2c-180;
293         }
294         else if( (data2c <data4c)&& ( data1c < data3c) )
295         {
296            y=data4c;
297            x=data1c+100;
298         }
299         else if( (data2c <data4c)&& ( data1c > data3c) )
300         {
301            y=data2c+80;
302            x=data1c-80;
303         }
304         else if( (data2c > data4c)&& ( data1c > data3c) )
305         {
306            y=data2c;
307            x=data1c-170;
308         }
309         else if ((data2c >data4c) && ( data1c < data3c))
310         {
311            y=data2c;
312            x=data1c+80;
313         }
314
315
316         try
317         {
318            RandomAccessFile objects= null;
319            String tmpStr="";
320            objects = new RandomAccessFile("template-object-rep/pro-req-object.xml",
                                              "r");
321            File file = new File("TEST5.xml");
322            FileWriter fr = new FileWriter(file, true);
```

E.16: ObjRep.java dosyası (devam).

```
323             BufferedWriter br = new BufferedWriter(fr);
324             PrintWriter pr = new PrintWriter(br);
325 
326             while ((tmpStr = objects.readLine()) != null)
327             {
328               tmpStr = tmpStr.replaceAll("OBJREPR",data7a + "-" + data6a +
                                                 "PROVIDER-OBJECT"+"-OBJREPR");
329               tmpStr = tmpStr.replaceAll("OBJ-PROVIDED-REQUIRED", data7a + "-" +
                                                 data6a + "PROVIDER-OBJECT");
330               tmpStr = tmpStr.replaceAll("place=\"240,110\"",
                                                 "place=\""+x+","+y+"\"");
331               pr.println(tmpStr);
332 
333             }
334             objects.close();
335             pr.close();
336               br.close()
337               fr.close();
338             }
339             catch (IOException e) { e.printStackTrace(); }
340 
341             String sql = "UPDATE Ports SET methNameCoordinateX='" +x+
                            "' WHERE methodName='"+ data6a + "';";
342             stmt6.executeUpdate(sql);
343 
344             sql = "UPDATE Ports SET methNameCoordinateY='" +y+"' WHERE methodName='"+
                      data6a + "';";
345             stmt6.executeUpdate(sql);
346              stmt6.close();
347           }
348 
349           stmt.close();
350           c.close();
351         }
352         catch (Exception err)
353         {
354             System.err.println(err.getClass().getName() + ": " + err.getMessage());
355         }
356     }
357 
358 }
359 
```

E.17: BindingRepr.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5  import java.nio.file.*;
6 
7  public class BindingRepr {
8 
9    public BindingRepr() {
10     writingBinding1();
```

E.17: BindingRepr.java dosyası (devam).

```
11      writingBinding2();
12    }
13    public static void main(String[] args) {
14      new BindingRepr();
15    }
16    public void writingBinding1() {
17      Connection c;
18      Statement stmt;
19      try
20      {
21        Class.forName("org.sqlite.JDBC");
22        SQLiteConfig config = new SQLiteConfig();
23        config.enforceForeignKeys(true);
24        c =DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db", config.toProperties());
25
26        Statement stmt5 = c.createStatement();
27        ResultSet rs5 = stmt5.executeQuery("SELECT * FROM ReqCompMth  ");
28        while (rs5.next())
29        {
30          String data1b = rs5.getString("methodName");
31          stmt = c.createStatement();
32
33          ResultSet rs1 = stmt.executeQuery("SELECT * FROM ReqPorts WHERE  methodName='" +
                                              data1b + "';");
34
35          while (rs1.next()) {
36              String data2a = rs1.getString("preCon1");
37              String data3a = rs1.getString("postCon1");
38              String data6a = rs1.getString("methodName");
39              String data7a = rs1.getString("compName");
40              String data8a = "";
41
42              int data1c = Integer.parseInt(rs1.getString("methNameCoordinateX"));
43              int data2c = Integer.parseInt(rs1.getString("methNameCoordinateY"));
44              int data3c=0;
45             int data4c=0;
46              Statement stmt2 = c.createStatement();
47              ResultSet rs2 = stmt2.executeQuery("SELECT compName,methNameCoordinateX,
                                                   methNameCoordinateY FROM Ports WHERE
                                portType='Provided' AND methodName='"  + data6a + "';");
48              while (rs2.next()) {
49               data8a = rs2.getString("compName");
50               data3c = Integer.parseInt(rs2.getString("methNameCoordinateX"));
51               data4c = Integer.parseInt(rs2.getString("methNameCoordinateY"));
52              }
53              stmt2.close();
54
55              try {
56                RandomAccessFile objects = null;
57                String tmpStr = "";
58                objects = new RandomAccessFile("template-binding-rep/binding-rep.xml",
                                                 "r");
59                File file = new File("TEST6.xml");
60                FileWriter fr = new FileWriter(file, true);
61                BufferedWriter br = new BufferedWriter(fr);
62                PrintWriter pr = new PrintWriter(br);
63
64                while ((tmpStr = objects.readLine()) != null) {
65                   tmpStr = tmpStr.replaceAll("BINDING-1",data7a + "-" + data8a + "-"
```

E.17: BindingRepr.java dosyası (devam).

```
                                                    + data6a + "BINDING-NAME");
66                  tmpStr = tmpStr.replaceAll("OBJREP-COMPONENT-NAME", data7a+
                                                    "-OBJREPR");
67                  tmpStr = tmpStr.replaceAll("OBJREP-PROVIDED-REQUIRED", data7a + "-"
                                                   + data6a + "REQUIRED-OBJECT-OBJREPR");
68                  tmpStr = tmpStr.replaceAll("place=\"508,110\"", "place=\"" +data1c
                                                    + ","+ data2c+"\"");
69                  pr.println(tmpStr);
70              }
71              objects.close();
72              pr.close();
73              br.close();
74              fr.close();
75            } catch (IOException e) {
76              e.printStackTrace();
77            }
78
79            try
80            {
81              RandomAccessFile objects= null;
82              String tmpStr="";
83              objects = new RandomAccessFile("template-sub-graph-id/sub-graph-id.xml",
                                                   "r");
84              File file = new File("TEST7.xml");
85              FileWriter fr = new FileWriter(file, true);
86              BufferedWriter br = new BufferedWriter(fr);
87              PrintWriter pr = new PrintWriter(br);
88
89              while ((tmpStr = objects.readLine()) != null)
90              {
91                  tmpStr = tmpStr.replaceAll("SUB-GRAPH-ID",data7a+"-"+ data8a+"-
                                                   "+data6a+"SUB-GRAPH");
92                  pr.println(tmpStr);
93              }
94              objects.close();
95              pr.close();
96              br.close();
97              fr.close();
98            }
99            catch (IOException e) { e.printStackTrace(); }
100         }
101
102         stmt.close();
103       }
104       stmt5.close();
105         c.close();
106     } catch (Exception err) {
107       System.err.println(err.getClass().getName() + ": " + err.getMessage());
108   }
109 }
110
111 public void writingBinding2()
112 {
113   Connection c;
114   Statement stmt;
115   try
116   {
117     Class.forName("org.sqlite.JDBC");
118     SQLiteConfig config = new SQLiteConfig();
```

E.17: BindingRepr.java dosyası (devam).

```
119       config.enforceForeignKeys(true);
120       c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                          config.toProperties());
121      stmt = c.createStatement();
122       ResultSet rs1 =stmt.executeQuery("SELECT * FROM Ports WHERE portType='Provided'
                                          ");
123
124       while (rs1.next())
125       {
126         String data1a = rs1.getString("portType");
127         String data2a = rs1.getString("preCon1");
128         String data3a = rs1.getString("postCon1");
129         String data4a = rs1.getString("preCon2");
130         String data5a = rs1.getString("postCon2");
131         String data6a = rs1.getString("methodName");
132         String data7a = rs1.getString("compName");
133         String data8a="";
134
135         int data1c = Integer.parseInt(rs1.getString("methNameCoordinateX"));
136         int data2c = Integer.parseInt(rs1.getString("methNameCoordinateY"));
137         int data3c=0;
138         int data4c=0;
139
140         Statement stmt2 = c.createStatement();
141         ResultSet rs2 = stmt2.executeQuery("SELECT compName,methNameCoordinateX,
                      methNameCoordinateY FROM ReqPorts WHERE methodName='"+data6a+"';");
142         while (rs2.next())
143         {
144            data8a = rs2.getString("compName");
145            data3c = Integer.parseInt(rs2.getString("methNameCoordinateX"));
146           data4c = Integer.parseInt(rs2.getString("methNameCoordinateY"));
147         }
148         stmt2.close();
149        try
150       {
151          RandomAccessFile objects= null;
152          String tmpStr="";
153          objects = new RandomAccessFile("template-binding-rep/binding-rep.xml", "r");
154          File file = new File("TEST6.xml");
155          FileWriter fr = new FileWriter(file, true);
156          BufferedWriter br = new BufferedWriter(fr);
157          PrintWriter pr = new PrintWriter(br);
158
159          while ((tmpStr = objects.readLine()) != null)
160          {
161            tmpStr = tmpStr.replaceAll("BINDING-1",data7a + "-" + data8a + "-" +
                                          data6a + "BINDING-NAME");
162            tmpStr = tmpStr.replaceAll("OBJREP-COMPONENT-NAME", data7a+"-OBJREPR");
163            tmpStr = tmpStr.replaceAll("OBJREP-PROVIDED-REQUIRED", data7a + "-" +
                                          data6a + "PROVIDER-OBJECT-OBJREPR");
164            tmpStr = tmpStr.replaceAll("place=\"508,110\"", "place=\""+data1c +","+
                                           data2c+"\"");
165            pr.println(tmpStr);
166           }
167           objects.close();
168           pr.close();
169           br.close();
170           fr.close();
171         }
```

E.17: BindingRepr.java dosyası (devam).

```
172          catch (IOException e) { e.printStackTrace(); }
173          try
174          {
175            RandomAccessFile objects= null;
176            String tmpStr="";
177            objects = new RandomAccessFile("template-sub-graph-id/sub-graph-id.xml",
                                                "r");
178            File file = new File("TEST7.xml");
179            FileWriter fr = new FileWriter(file, true);
180            BufferedWriter br = new BufferedWriter(fr);
181            PrintWriter pr = new PrintWriter(br);
182
183            while ((tmpStr = objects.readLine()) != null)
184            {
185              tmpStr = tmpStr.replaceAll("SUB-GRAPH-ID",data7a+"-"+ data8a+"-"+
                                              data6a+"SUB-GRAPH");
186              pr.println(tmpStr);
187            }
188            objects.close();
189            pr.close();
190            br.close();
191            fr.close();
192          }
193          catch (IOException e) { e.printStackTrace(); }
194        }
195
196        stmt.close();
197        c.close();
198      }
199      catch (Exception err)
200      {
201         System.err.println(err.getClass().getName() + ": " + err.getMessage());
202      }
203    }
204
205 }
206
```

E.18: ReJunction.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5  import java.nio.file.*;
6
7  public class ReJunction {
8
9    public ReJunction() {
10     writing1();
11     writing2();
12     writing3();
13     writing4();
14     writingDiyagramHeader();
15     writing5();
```

E.18: ReJunction.java dosyası (devam).

```
16      writing6();
17      writingDiyagramEnd();
18      writing7();
19      finalCheck();
20
21    }
22
23    public static void main(String[] args) {
24       new ReJunction();
25    }
26    public void writing1()
27    {
28      try {
29        RandomAccessFile objects = null;
30        String tmpStr = "";
31       objects = new RandomAccessFile("TEST1.xml", "r");
32       File file = new File("Final.mxm");
33       FileWriter fr = new FileWriter(file, true);
34       BufferedWriter br = new BufferedWriter(fr);
35       PrintWriter pr = new PrintWriter(br);
36
37       while ((tmpStr = objects.readLine()) != null) {
38             pr.println(tmpStr);
39       }
40       objects.close();
41       pr.close();
42       br.close();
43       fr.close();
44      } catch (IOException e) {
45         e.printStackTrace();
46      }
47    }
48    public void writing2()
49    {
50      try {
51        RandomAccessFile objects = null;
52        String tmpStr = "";
53        objects = new RandomAccessFile("TEST2.xml", "r");
54        File file = new File("Final.mxm");
55        FileWriter fr = new FileWriter(file, true);
56        BufferedWriter br = new BufferedWriter(fr);
57        PrintWriter pr = new PrintWriter(br);
58
59        while ((tmpStr = objects.readLine()) != null) {
60             pr.println(tmpStr);
61        }
62        objects.close();
63        pr.close();
64        br.close();
65        fr.close();
66      } catch (IOException e) {
67         e.printStackTrace();
68      }
69    }
70    public void writing3()
71    {
72      try {
73        RandomAccessFile objects = null;
74        String tmpStr = "";
```

E.18: ReJunction.java dosyası (devam).

```
75        objects = new RandomAccessFile("TEST3.xml", "r");
76        File file = new File("Final.mxm");
77        FileWriter fr = new FileWriter(file, true);
78        BufferedWriter br = new BufferedWriter(fr);
79        PrintWriter pr = new PrintWriter(br);
80
81        while ((tmpStr = objects.readLine()) != null) {
82             pr.println(tmpStr);
83        }
84        objects.close();
85        pr.close();
86        br.close();
87        fr.close();
88      } catch (IOException e) {
89         e.printStackTrace();
90      }
91    }
92    public void writing4()
93    {
94      try {
95        RandomAccessFile objects = null;
96        String tmpStr = "";
97        objects = new RandomAccessFile("TEST4.xml", "r");
98        File file = new File("Final.mxm");
99        FileWriter fr = new FileWriter(file, true);
100       BufferedWriter br = new BufferedWriter(fr);
101       PrintWriter pr = new PrintWriter(br);
102
103         while ((tmpStr = objects.readLine()) != null) {
104                   pr.println(tmpStr);
105         }
106         objects.close();
107         pr.close();
108         br.close();
109         fr.close();
110      } catch (IOException e) {
111         e.printStackTrace();
112      }
113   }
114   public void writingDiyagramHeader()
115   {
116     try {
117       RandomAccessFile objects = null;
118       String tmpStr = "";
119       objects = new RandomAccessFile("template-diyagram/diyagram-header.xml", "r");
120       File file = new File("Final.mxm");
121       FileWriter fr = new FileWriter(file, true);
122       BufferedWriter br = new BufferedWriter(fr);
123       PrintWriter pr = new PrintWriter(br);
124
125       while ((tmpStr = objects.readLine()) != null) {
126              pr.println(tmpStr);
127       }
128       objects.close();
129       pr.close();
130       br.close();
131       fr.close();
132     } catch (IOException e) {
133         e.printStackTrace();
```

E.18: ReJunction.java dosyası (devam).

```
134     }
135   }
136   public void writing5()
137   {
138     try {
139       RandomAccessFile objects = null;
140       String tmpStr = "";
141       objects = new RandomAccessFile("TEST5.xml", "r");
142       File file = new File("Final.mxm");
143       FileWriter fr = new FileWriter(file, true);
144       BufferedWriter br = new BufferedWriter(fr);
145       PrintWriter pr = new PrintWriter(br);
146 
147       while ((tmpStr = objects.readLine()) != null) {
148             pr.println(tmpStr);
149       }
150       objects.close();
151       pr.close();
152      br.close();
153       fr.close();
154      } catch (IOException e) {
155        e.printStackTrace();
156     }
157 }
158   public void writing6()
159   {
160     try {
161       RandomAccessFile objects = null;
162       String tmpStr = "";
163       objects = new RandomAccessFile("TEST6.xml", "r");
164       File file = new File("Final.mxm");
165       FileWriter fr = new FileWriter(file, true);
166       BufferedWriter br = new BufferedWriter(fr);
167       PrintWriter pr = new PrintWriter(br);
168 
169       while ((tmpStr = objects.readLine()) != null) {
170             pr.println(tmpStr);
171       }
172       objects.close();
173       pr.close();
174       br.close();
175       fr.close();
176     } catch (IOException e) {
177         e.printStackTrace();
178     }
179   }
180   public void writingDiyagramEnd()
181   {
182     try {
183       RandomAccessFile objects = null;
184       String tmpStr = "";
185       objects = new RandomAccessFile("template-diyagram/diyagram-end.xml", "r");
186       File file = new File("Final.mxm");
187       FileWriter fr = new FileWriter(file, true);
188       BufferedWriter br = new BufferedWriter(fr);
189       PrintWriter pr = new PrintWriter(br);
190 
191       while ((tmpStr = objects.readLine()) != null) {
192             pr.println(tmpStr);
```

E.18: ReJunction.java dosyası (devam).

```
193       }
194       objects.close();
195       pr.close();
196       br.close();
197       fr.close();
198     } catch (IOException e) {
199        e.printStackTrace();
200     }
201   }
202   public void writing7()
203   {
204     try {
205       RandomAccessFile objects = null;
206       String tmpStr = "";
207       objects = new RandomAccessFile("TEST7.xml", "r");
208       File file = new File("Final.mxm");
209       FileWriter fr = new FileWriter(file, true);
210       BufferedWriter br = new BufferedWriter(fr);
211       PrintWriter pr = new PrintWriter(br);
212
213       while ((tmpStr = objects.readLine()) != null) {
214              pr.println(tmpStr);
215       }
216       objects.close();
217       pr.println("</gxl>");
218       pr.close();
219       br.close();
220       fr.close();
221     } catch (IOException e) {
222        e.printStackTrace();
223     }
224
225   }
226   public void finalCheck ()
227   {
228     try {
229      RandomAccessFile objects = null;
230      String tmpStr = "";
231      objects = new RandomAccessFile("Final.mxm", "rw");
232      File file = new File("Final2.mxm");
233      FileWriter fr = new FileWriter(file, true);
234      BufferedWriter br = new BufferedWriter(fr);
235      PrintWriter pr = new PrintWriter(br);
236
237      while ((tmpStr = objects.readLine()) != null) {
238            tmpStr = tmpStr.replaceAll("&", "&amp;");
239            pr.println(tmpStr);
240      }
241      objects.close();
242      pr.close();
243      br.close();
244      fr.close();
245
246      File myObj = new File("Final.mxm");
247      if(myObj.exists() && !myObj.isDirectory())
248      {
249
250        if (myObj.delete())
251        {
```

E.18: ReJunction.java dosyası (devam).

```
252             System.out.println("Deleted the file: " + myObj.getName());
253         }
254         else {
255             System.out.println("Failed to delete the file.");
256         }
257       }
258
259     } catch (IOException e) {
260         e.printStackTrace();
261     }
262   }
263
264
265}
266
```

E.19: Main4.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5
6  public class Main4 {
7     Header  headerIns;
8     ProvidedActivity providedActivityIns;
9     RequiredActivity requiredActivityIns;
10    Binding bindingIns;
11    ObjRep objRepIns;
12    BindingReprActivity bindingReprActivityIns;
13    ReJunctionActivity rejunctionIns;
14    Connection c;
15    Statement stmt;
16    public Main4()
17    {
18       deleteFiles();
19       headerIns =new Header();
20       writingComponent();
21       providedActivityIns = new ProvidedActivity();
22       requiredActivityIns = new RequiredActivity ();
23       bindingIns = new Binding ();
24       objRepIns= new ObjRep();
25       bindingReprActivityIns = new BindingReprActivity();
26       rejunctionIns = new ReJunctionActivity();
27      }
28      public void deleteFiles()
29      {
30         File f1 =new File("FinalActivity.mxm");
31         f1.delete();
32         f1 =new File("FinalActivity2.mxm");
33         f1.delete();
34         f1 =new File("TEST1.xml");
35         f1.delete();
36         f1 =new File("TEST11.xml");
```

E.19: Main4.java dosyası (devam).

```
37          f1.delete();
38          f1 =new File("TEST10.xml");
39          f1.delete();
40          f1 =new File("TEST4.xml");
41          f1.delete();
42          f1 =new File("TEST5.xml");
43          f1.delete();
44          f1 =new File("TEST6.xml");
45          f1.delete();
46          f1 =new File("TEST7.xml");
47          f1.delete();
48      }
49      public static void main(String[] args)
50      {
51         new Main4();
52      }
53      public void writingComponent()
54      {
55        try {
56              Class.forName("org.sqlite.JDBC");
57              SQLiteConfig config = new SQLiteConfig();
58              config.enforceForeignKeys(true);
59              c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                                config.toProperties());
60             stmt = c.createStatement();
61
62              ResultSet rs1 = stmt.executeQuery("SELECT componentName FROM ComponentNames
                                                  ");
63              int countFinal=0;
64              while (rs1.next()) {
65                     String data1a = rs1.getString("componentName");
66                     Statement stmt2 = c.createStatement();
67                      try
68                      {
69                        RandomAccessFile objects= null;
70                        String tmpStr="";
71                        objects = new RandomAccessFile("template-component/Component.xml",
                                                         "r");
72                        File file = new File("TEST1.xml");
73                        FileWriter fr = new FileWriter(file, true);
74                        BufferedWriter br = new BufferedWriter(fr);
75                        PrintWriter pr = new PrintWriter(br);
76
77                        while ((tmpStr = objects.readLine()) != null)
78                        {
79                             tmpStr = tmpStr.replaceAll("COMPONENT-NAME-1",data1a);
80                             tmpStr = tmpStr.replaceAll("COMPONENT-NAME",data1a);
81
82                             pr.println(tmpStr);
83                        }
84                        stmt2.close();
85                        objects.close();
86                        pr.close();
87                        br.close();
88                        fr.close();
89                     }
90                     catch (IOException e)
91                     {
92                        e.printStackTrace();
```

E.19: Main4.java dosyası (devam).

```
93                    }
94
95                    Statement stmt3  = c.createStatement();
96                    ResultSet rs3 = stmt3.executeQuery("SELECT
                           parameterName,parameterType, parameterDefaultValue,componentName
                           FROM ParameterList WHERE componentName='"+data1a+"';" );
97                    while (rs3.next())
98                    {
99                      String data1b = rs3.getString("parameterName");
100                     String data2b = rs3.getString("parameterType");
101                     String data3b = rs3.getString("parameterDefaultValue");
102                     String data4b = rs3.getString("componentName");
103                     try
104                     {
105                       RandomAccessFile objects= null;
106                       String tmpStr="";
107                       objects = new RandomAccessFile("template-component/
                                                        ComponentParamList.xml", "r");
108                       File file = new File("TEST1.xml");
109                       FileWriter fr = new FileWriter(file, true);
110                       BufferedWriter br = new BufferedWriter(fr);
111                       PrintWriter pr = new PrintWriter(br);
112                          while ((tmpStr = objects.readLine()) != null)
113                          {
114                             tmpStr = tmpStr.replaceAll("PARAM-TYPE",data2b);
115                             tmpStr = tmpStr.replaceAll("PARAM-NAME",data1b);
116                             tmpStr = tmpStr.replaceAll("PARAM-VALUE",data3b);
117                             pr.println(tmpStr);
118                          }
119                          objects.close();
120                          pr.close();
121                          br.close();
122                          fr.close();
123
124                      }
125                      catch (IOException e)
126                      {
127                          e.printStackTrace();
128                      }
129                   }
130                   stmt3.close();
131                   try
132                   {
133                     RandomAccessFile objects= null;
134                     String tmpStr="";
135                     objects = new RandomAccessFile("template-component/Component2.xml",
                                                       "r");
136                     File file = new File("TEST1.xml");
137                     FileWriter fr = new FileWriter(file, true);
138                     BufferedWriter br = new BufferedWriter(fr);
139                     PrintWriter pr = new PrintWriter(br);
140
141                     while ((tmpStr = objects.readLine()) != null)
142                     {
143                        pr.println(tmpStr);
144                     }
145                     objects.close();
146                     pr.close();
147                     br.close();
```

E.19: Main4.java dosyası (devam).

```
148                     fr.close();
149                 }
150                 catch (IOException e)
151                 {
152                    e.printStackTrace();
153                 }
154
155                 ResultSet rs2 = stmt2.executeQuery("SELECT dataName,dataType,
                                    dataDefaultValue,componentName,arraySize FROM
                                    DataList WHERE componentName='"+data1a+"';" );
156                 while (rs2.next())
157                 {
158                     String data1b = rs2.getString("dataName");
159                     String data2b = rs2.getString("dataType");
160                    String data3b = rs2.getString("dataDefaultValue");
161                     String data4b = rs2.getString("componentName");
162                    String data5b = rs2.getString("arraySize");
163                     try
164                     {
165                         RandomAccessFile objects= null;
166                         String tmpStr="";
167                         objects = new RandomAccessFile("template-component/
                                                        ComponentDataList.xml", "r");
168                         File file = new File("TEST1.xml");
169                         FileWriter fr = new FileWriter(file, true);
170                         BufferedWriter br = new BufferedWriter(fr);
171                         PrintWriter pr = new PrintWriter(br);
172                         while ((tmpStr = objects.readLine()) != null)
173                         {
174                            tmpStr = tmpStr.replaceAll("DATA-TYPE",data2b);
175                            tmpStr = tmpStr.replaceAll("DATA-NAME",data1b);
176                            tmpStr = tmpStr.replaceAll("DEFAULT-VALUE",data3b);
177                             tmpStr = tmpStr.replaceAll("<string></string>",
                                                    "<string>"+data5b+"</string>");
178                            pr.println(tmpStr);
179                         }
180                         objects.close();
181                           pr.close();
182                           br.close();
183                           fr.close();
184                    }
185                    catch (IOException e)
186                    {
187                         e.printStackTrace();
188                    }
189                 }
190                 stmt2.close();
191                 try
192                 {
193                   RandomAccessFile objects= null;
194                   String tmpStr="";
195                   objects = new RandomAccessFile("template-
                                                  component/ComponentEnd.xml","r");
196                   File file = new File("TEST1.xml");
197                   FileWriter fr = new FileWriter(file, true);
198                   BufferedWriter br = new BufferedWriter(fr);
199                   PrintWriter pr = new PrintWriter(br);
200
201                   while ((tmpStr = objects.readLine()) != null)
```

E.19: Main4.java dosyası (devam).

```
202                     {
203                        pr.println(tmpStr);
204                     }
205                     objects.close();
206                     pr.close();
207                     br.close();
208                     fr.close();
209                   }
210                   catch (IOException e)
211                   {
212                      e.printStackTrace();
213                   }
214               }
215               stmt.close();
216               c.close();
217           } catch (Exception err)
218           {
219               System.err.println(err.getClass().getName() + ":"+err.getMessage());
220           }
221       }
222   }
```

E.20: ProvidedActivity.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5
6  public class ProvidedActivity {
7       Connection c;
8       Statement stmt;
9       public ProvidedActivity()
10      {
11          writingComponent();
12      }
13      public static void main(String[] args)
14      {
15         new ProvidedActivity();
16      }
17      public void writingComponent()
18      {
19         try {
20            Class.forName("org.sqlite.JDBC");
21            SQLiteConfig config = new SQLiteConfig();
22            config.enforceForeignKeys(true);
23            c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                              config.toProperties());
24           stmt = c.createStatement();
25           ResultSet rs1 = stmt.executeQuery("SELECT * FROM Ports WHERE
                                                portType='Provided' ");
26
27           while (rs1.next())
28           {
```

E.20: ProvidedActivity.java dosyası (devam).

```
29              String data1a = rs1.getString("portType");
30              String data2a = rs1.getString("preCon1");
31              String data3a = rs1.getString("postCon1");
32              String data4a = rs1.getString("preCon2");
33              String data5a = rs1.getString("postCon2");
34              String data6a = rs1.getString("methodName");
35              String data7a = rs1.getString("compName");
36              String data8a="";
37              Statement stmt2 = c.createStatement();
38
39              try
40              {
41                 RandomAccessFile objects= null;
42                 String tmpStr="";
43                 objects = new RandomAccessFile("template-provided-activity/
                                                   activity-provided-entrance.xml", "r");
44                 File file = new File("TEST11.xml");
45                 FileWriter fr = new FileWriter(file, true);
46                 BufferedWriter br = new BufferedWriter(fr);
47                 PrintWriter pr = new PrintWriter(br);
48
49                 while ((tmpStr = objects.readLine()) != null)
50                 {
51                     tmpStr = tmpStr.replaceAll("PROVIDER-OBJECT-1",data7a+"-"+
                                                      data6a+"PROVIDER-OBJECT");
52                     pr.println(tmpStr);
53                 }
54                 objects.close();
55                 pr.close();
56                 br.close();
57                 fr.close();
58              }
59              catch (IOException e) { e.printStackTrace(); }
60           }
61           stmt.close();
62           c.close();
63        }catch (Exception err)
64        {
65           System.err.println(err.getClass().getName() + ": " + err.getMessage());
66        }
67     }
68
69  }
70
```

E.21: RequiredActivity.java dosyası.

```
1   import java.util.*;
2   import java.io.*;
3   import java.sql.*;
4   import org.sqlite.SQLiteConfig;
5   import java.nio.file.*;
6
7   public class RequiredActivity {
8     Connection c;
```

E.21: RequiredActivity.java dosyası (devam).

```
9     Statement stmt;
10
11  public RequiredActivity() {
12      writingComponent();
13  }
14  public void writingComponent() {
15      try {
16        Class.forName("org.sqlite.JDBC");
17        SQLiteConfig config = new SQLiteConfig();
18        config.enforceForeignKeys(true);
19        c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                          config.toProperties());
20        Statement stmt5 = c.createStatement();
21        ResultSet rs5 = stmt5.executeQuery("SELECT * FROM ReqCompMth  ");
22        while (rs5.next())
23        {
24          String data1b = rs5.getString("methodName");
25          String data2b = rs5.getString("portName");
26          stmt = c.createStatement();
27          ResultSet rs1 = stmt.executeQuery("SELECT * FROM ReqPorts WHERE  portName='" +
                                            data2b + "';");
28          while (rs1.next())
29          {
30              String data2a = rs1.getString("preCon1");
31              String data3a = rs1.getString("postCon1");
32              String data6a = rs1.getString("methodName");
33              String data6aa = rs1.getString("portName");
34              String data7a = rs1.getString("compName");
35              String data8a = "";
36
37              String data13a = "";
38              String data14a = "";
39              String data15a = "";
40              String data16a = "";
41              String data17a = "";
42
43              Statement stmt2 = c.createStatement();
44              ResultSet rs2 = stmt2.executeQuery("SELECT * FROM Ports WHERE
                                    portType='Provided' AND portName='" + data6aa + "';");
45              while (rs2.next()) {
46                 data8a = rs2.getString("compName");
47                 data13a = rs2.getString("preCon1");
48                 data14a = rs2.getString("postCon1");
49                 data15a = rs2.getString("preCon2");
50                 data16a = rs2.getString("postCon2");
51                 data17a = rs2.getString("methodName");
52              }
53              stmt2.close();
54              try {
55                 RandomAccessFile objects = null;
56                 String tmpStr = "";
57                 objects = new RandomAccessFile("template-required-activity/
                                                  activity-required-entrance.xml", "r");
58                 File file = new File("TEST10.xml");
59                 FileWriter fr = new FileWriter(file, true);
60                 BufferedWriter br = new BufferedWriter(fr);
61                 PrintWriter pr = new PrintWriter(br);
62
63                 while ((tmpStr = objects.readLine()) != null) {
```

E.21: RequiredActivity.java dosyası (devam).

```
64                    tmpStr = tmpStr.replaceAll("REQUIRED-OBJECT-1",data7a + "-" +
                                                  data6a + "REQUIRED-OBJECT");
65                    tmpStr = tmpStr.replaceAll("SUB-GRAPH",data7a + "-" + data8a +
                                                   "-" + data6a + "SUB-GRAPH");
66                    pr.println(tmpStr);
67                }
68                objects.close();
69                pr.close();
70                br.close();
71                fr.close();
72             } catch (IOException e) {
73                 e.printStackTrace();
74             }
75             String data9a = "";
76             String data10a = "";
77             String data11a = "";
78             String data12a = "";
79             Statement stmt3 = c.createStatement();
80             ResultSet rs3 = stmt3.executeQuery("SELECT * FROM Methods WHERE
                                                 Methods.componentName='" + data7a+ "' AND
                                                 Methods.methodName='" + data6a + "'");
81             while (rs3.next())
82             {
83                data9a = rs3.getString("parameterName");
84                data10a = rs3.getString("methodName");
85                data9a = data9a.trim();
86                if (data9a != "") {
87                    Statement stmt4 = c.createStatement();
88                    ResultSet rs4 = stmt4.executeQuery("SELECT * FROM MethodParameter
                                                      WHERE componentName='"+ data7a +
                                                  "' AND parameterName='" + data9a + "'");
89                    while (rs4.next()) {
90                       data11a = rs4.getString("parameterType");
91                       data12a = rs4.getString("parameterName");
92                       try {
93                         RandomAccessFile objects = null;
94                         String tmpStr = "";
95                         objects = new RandomAccessFile("template-required-activity/
                                                        activity-parameters.xml", "r");
96                         File file = new File("deneme2/Test-Params.xml");
97                         FileWriter fr = new FileWriter(file, true);
98                         BufferedWriter br = new BufferedWriter(fr);
99                         PrintWriter pr = new PrintWriter(br);
100                        while ((tmpStr = objects.readLine()) != null) {
101                           tmpStr = tmpStr.replaceAll("PARAMETER-NAME", data12a);
102                           tmpStr = tmpStr.replaceAll("PARAMETER-TYPE", data11a);
103                           pr.println(tmpStr);
104                        }
105                        objects.close();
106                        pr.close();
107                        br.close();
108                        fr.close();
109                      } catch (IOException e) {
110                         e.printStackTrace();
111                      }
112                    }
113                    stmt4.close();
114                 }
115                 if (data9a.isEmpty()) {
```

E.21: RequiredActivity.java dosyası (devam).

```
116                     data11a = "";
117                     data12a = "";
118                     try {
119                       RandomAccessFile objects = null;
120                       String tmpStr = "";
121                       objects = new RandomAccessFile("template-required-activity/
                                                          activity-parameters.xml", "r");
122                       File file = new File("deneme2/Test-Params.xml");
123                       FileWriter fr = new FileWriter(file, true);
124                       BufferedWriter br = new BufferedWriter(fr);
125                       PrintWriter pr = new PrintWriter(br);
126
127
128                       while ((tmpStr = objects.readLine()) != null) {
129                          tmpStr = tmpStr.replaceAll("PARAMETER-NAME", data12a);
130                          tmpStr = tmpStr.replaceAll("PARAMETER-TYPE", data11a);
131                          pr.println(tmpStr);
132                       }
133                       objects.close();
134                       pr.close();
135                       br.close();
136                       fr.close();
137                     } catch (IOException e) {
138                        e.printStackTrace();
139                     }
140                   }
141
142               }
143               stmt3.close();
144
145               try
146               {
147                 RandomAccessFile objects = null;
148                 String tmpStr = "";
149                 objects = new RandomAccessFile("template-required-activity/
                                                    activity-request1-object-entrance.xml", "r");
150                 File file = new File("TEST10.xml");
151                 FileWriter fr = new FileWriter(file, true);
152                 BufferedWriter br = new BufferedWriter(fr);
153                 PrintWriter pr = new PrintWriter(br);
154                 while ((tmpStr = objects.readLine()) != null) {
155                    tmpStr = tmpStr.replaceAll("ACTIVITY-REQUEST-1", data7a + "-" +
                                                    data8a + "-" + data6a+"-ACTIVITY-REQUEST-1");
156                    tmpStr = tmpStr.replaceAll("PROMISE", data3a);
157                    tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
158                    pr.println(tmpStr);
159                 }
160                 objects.close();
161                 pr.close();
162                 br.close();
163                 fr.close();
164               } catch (IOException e) { e.printStackTrace();}
165               try
166               {
167                  RandomAccessFile objects = null;
168                  String tmpStr = "";
169                  objects = new RandomAccessFile("deneme2/Test-Params.xml", "rw");
170                  File file = new File("TEST10.xml");
171                  FileWriter fr = new FileWriter(file, true);
```

E.21: RequiredActivity.java dosyası (devam).

```
172                 BufferedWriter br = new BufferedWriter(fr);
173                 PrintWriter pr = new PrintWriter(br);
174                 while ((tmpStr = objects.readLine()) != null) {
175                       pr.println(tmpStr);
176                 }
177                 objects.close();
178                 pr.close();
179                 br.close();
180                 fr.close();
181               } catch (IOException e) { e.printStackTrace();}
182               try
183               {
184                 RandomAccessFile objects = null;
185                 String tmpStr = "";
186                 objects = new RandomAccessFile("template-required-activity/
                                                    activity-object-end.xml", "r");
187                 File file = new File("TEST10.xml");
188                 FileWriter fr = new FileWriter(file, true);
189                 BufferedWriter br = new BufferedWriter(fr);
190                 PrintWriter pr = new PrintWriter(br);
191                 while ((tmpStr = objects.readLine()) != null) {
192                       pr.println(tmpStr);
193                 }
194                 objects.close();
195                 pr.close();
196                 br.close();
197                 fr.close();
198               } catch (IOException e) { e.printStackTrace();}
199
200               try
201               {
202                 RandomAccessFile objects = null;
203                 String tmpStr = "";
204                 objects = new RandomAccessFile("template-required-activity/
                                                  activity-request2-object-entrance.xml", "r");
205                 File file = new File("TEST10.xml");
206                 FileWriter fr = new FileWriter(file, true);
207                 BufferedWriter br = new BufferedWriter(fr);
208                 PrintWriter pr = new PrintWriter(br);
209                 while ((tmpStr = objects.readLine()) != null) {
210                     tmpStr = tmpStr.replaceAll("ACTIVITY-REQUEST-2", data7a + "-" +
                                                 data8a + "-" + data6a+"-ACTIVITY-REQUEST-2");
211                     tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
212                     pr.println(tmpStr);
213                 }
214                 objects.close();
215                 pr.close();
216                 br.close();
217                 fr.close();
218               } catch (IOException e) { e.printStackTrace();}
219               try
220               {
221                 RandomAccessFile objects = null;
222                 String tmpStr = "";
223                 objects = new RandomAccessFile("deneme2/Test-Params.xml", "rw");
224                 File file = new File("TEST10.xml");
225                 FileWriter fr = new FileWriter(file, true);
226                 BufferedWriter br = new BufferedWriter(fr);
227                 PrintWriter pr = new PrintWriter(br);
```

E.21: RequiredActivity.java dosyası (devam).

```
228                 while ((tmpStr = objects.readLine()) != null) {
229                        pr.println(tmpStr);
230                 }
231                 objects.close();
232                 pr.close();
233                 br.close();
234                 fr.close();
235               } catch (IOException e) { e.printStackTrace();}
236               try
237               {
238                 RandomAccessFile objects = null;
239                 String tmpStr = "";
240                 objects = new RandomAccessFile("template-required-activity/
                                                    activity-object-end.xml", "r");
241                 File file = new File("TEST10.xml");
242                 FileWriter fr = new FileWriter(file, true);
243                 BufferedWriter br = new BufferedWriter(fr);
244                 PrintWriter pr = new PrintWriter(br);
245                 while ((tmpStr = objects.readLine()) != null) {
246                        pr.println(tmpStr);
247                 }
248                 objects.close();
249                 pr.close();
250                 br.close();
251                 fr.close();
252               } catch (IOException e) { e.printStackTrace();}
253
254               try
255               {
256                 RandomAccessFile objects = null;
257                 String tmpStr = "";
258                 objects = new RandomAccessFile("template-required-activity/
                                                   activity-send1-object-entrance.xml", "r");
259                 File file = new File("TEST10.xml");
260                 FileWriter fr = new FileWriter(file, true);
261                 BufferedWriter br = new BufferedWriter(fr);
262                 PrintWriter pr = new PrintWriter(br);
263                 while ((tmpStr = objects.readLine()) != null) {
264                        tmpStr = tmpStr.replaceAll("ACTIVITY-SEND-1", data7a + "-" +
                                                   data8a + "-" + data6a+"-ACTIVITY-SEND-1");
265                        tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
266                        pr.println(tmpStr);
267                 }
268                 objects.close();
269                 pr.close();
270                 br.close();
271                 fr.close();
272               } catch (IOException e) { e.printStackTrace();}
273               try
274               {
275                 RandomAccessFile objects = null;
276                 String tmpStr = "";
277                 objects = new RandomAccessFile("deneme2/Test-Params.xml", "rw");
278                 File file = new File("TEST10.xml");
279                 FileWriter fr = new FileWriter(file, true);
280                 BufferedWriter br = new BufferedWriter(fr);
281                 PrintWriter pr = new PrintWriter(br);
282                 while ((tmpStr = objects.readLine()) != null) {
283                        pr.println(tmpStr);
```

E.21: RequiredActivity.java dosyası (devam).

```
284                 }
285                 objects.close();
286                 pr.close();
287                 br.close();
288                 fr.close();
289               } catch (IOException e) { e.printStackTrace();}
290               try
291               {
292                 RandomAccessFile objects = null;
293                 String tmpStr = "";
294                 objects = new RandomAccessFile("template-required-activity/
                                                    activity-object-end.xml", "r");
295                 File file = new File("TEST10.xml");
296                 FileWriter fr = new FileWriter(file, true);
297                 BufferedWriter br = new BufferedWriter(fr);
298                 PrintWriter pr = new PrintWriter(br);
299                 while ((tmpStr = objects.readLine()) != null) {
300                        pr.println(tmpStr);
301                 }
302                   objects.close();
303                   pr.close();
304                   br.close();
305                   fr.close();
306               } catch (IOException e) { e.printStackTrace();}
307
308               try
309               {
310                 RandomAccessFile objects = null;
311                 String tmpStr = "";
312                 objects = new RandomAccessFile("template-required-activity/
                                                   activity-send2-object-entrance.xml", "rw");
313                 File file = new File("TEST10.xml");
314                 FileWriter fr = new FileWriter(file, true);
315                 BufferedWriter br = new BufferedWriter(fr);
316                 PrintWriter pr = new PrintWriter(br);
317                 while ((tmpStr = objects.readLine()) != null) {
318                      tmpStr = tmpStr.replaceAll("ACTIVITY-SEND-2", data7a + "-" +
                                                   data8a + "-" + data6a+"-ACTIVITY-SEND-2");
319                      tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
320                      pr.println(tmpStr);
321                 }
322                 objects.close();
323                 pr.close();
324                 br.close();
325                 fr.close();
326               } catch (IOException e) { e.printStackTrace();}
327               try
328               {
329                 RandomAccessFile objects = null;
330                 String tmpStr = "";
331                 objects = new RandomAccessFile("deneme2/Test-Params.xml", "r");
332                 File file = new File("TEST10.xml");
333                 FileWriter fr = new FileWriter(file, true);
334                 BufferedWriter br = new BufferedWriter(fr);
335                 PrintWriter pr = new PrintWriter(br);
336                 while ((tmpStr = objects.readLine()) != null) {
337                       pr.println(tmpStr);
338                 }
339                 objects.close();
```

E.21: RequiredActivity.java dosyası (devam).

```
340                 pr.close();
341                 br.close();
342                 fr.close();
343               } catch (IOException e) { e.printStackTrace();}
344               try
345               {
346                  RandomAccessFile objects = null;
347                  String tmpStr = "";
348                  objects = new RandomAccessFile("template-required-activity/
                                                     activity-object-end.xml", "r");
349                  File file = new File("TEST10.xml");
350                  FileWriter fr = new FileWriter(file, true);
351                  BufferedWriter br = new BufferedWriter(fr);
352                  PrintWriter pr = new PrintWriter(br);
353                  while ((tmpStr = objects.readLine()) != null) {
354                         pr.println(tmpStr);
355                  }
356                  objects.close();
357                  pr.close();
358                  br.close();
359                  fr.close();
360               } catch (IOException e) { e.printStackTrace();}
361
362               try
363               {
364                  RandomAccessFile objects = null;
365                  String tmpStr = "";
366                  objects = new RandomAccessFile("template-required-activity/
                                                     activity-diamond.xml", "r");
367                  File file = new File("TEST10.xml");
368                  FileWriter fr = new FileWriter(file, true);
369                   BufferedWriter br = new BufferedWriter(fr);
370                   PrintWriter pr = new PrintWriter(br);
371                   while ((tmpStr = objects.readLine()) != null) {
372                        tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST", data7a +
                                    "-" + data8a + "-" + data6a+"-DIAMOND-REQUEST-1-PRE");
373                        tmpStr = tmpStr.replaceAll("PRE-POST", "pre");
374                        tmpStr = tmpStr.replaceAll("CONDITION", data2a);
375                        pr.println(tmpStr);
376                   }
377                   objects.close();
378                   pr.close();
379                   br.close();
380                   fr.close();
381                } catch (IOException e) { e.printStackTrace();}
382
383                Statement stmt6 = c.createStatement();
384                ResultSet rs6 = stmt6.executeQuery("SELECT * FROM ReqPostCond  WHERE
                                                     methodName='" + data1b + "';");
385                int count=0;
386                String data4a="";
387                String data5a="";
388                while (rs6.next()) {
389                   count++;
                      data5a="";
390                   if ( (rs6.getString("preCon2")!= null) &&
                              !(rs6.getString("preCon2").isEmpty()) )
391                        data4a = rs6.getString("preCon2");
392                   if ( (rs6.getString("postCon2")!= null) &&
```

E.21: RequiredActivity.java dosyası (devam).

```
                        !(rs6.getString("postCon2").isEmpty()) )
393                   data5a = rs6.getString("postCon2");
394               if(count==1)
395               {
396                  try
397                  {
398                    RandomAccessFile objects = null;
399                    String tmpStr = "";
400                    objects = new RandomAccessFile("template-required-activity/
                                                      activity-diamond.xml", "r");
401                    File file = new File("TEST10.xml");
402                    FileWriter fr = new FileWriter(file, true);
403                    BufferedWriter br = new BufferedWriter(fr);
404                    PrintWriter pr = new PrintWriter(br);
405                    while ((tmpStr = objects.readLine()) != null) {
406                      tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST", data7a
                             + "-"+ data8a + "-" + data6a+"-DIAMOND-REQUEST-2-PRE");
407                      tmpStr = tmpStr.replaceAll("PRE-POST", "pre");
408                      tmpStr = tmpStr.replaceAll("CONDITION", data4a);
409                      pr.println(tmpStr);
410                    }
411                    objects.close();
412                    pr.close();
413                    br.close();
414                    fr.close();
415                  } catch (IOException e) { e.printStackTrace();}
416
417                  if (!data5a.isEmpty())
418                  {
419                    try
420                    {
421                      RandomAccessFile objects = null;
422                      String tmpStr = "";
423                      objects = new RandomAccessFile("template-required-
                                                    activity/activity-diamond.xml", "r");
424                      File file = new File("TEST10.xml");
425                      FileWriter fr = new FileWriter(file, true);
426                      BufferedWriter br = new BufferedWriter(fr);
427                      PrintWriter pr = new PrintWriter(br);
428                      while ((tmpStr = objects.readLine()) != null) {
429                             tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST",
                                                        data7a + "-" + data8a + "-"
                                                  + data6a+"-DIAMOND-REQUEST-2-POST");
430                            tmpStr = tmpStr.replaceAll("PRE-POST", "post");
431                            tmpStr = tmpStr.replaceAll("CONDITION", data5a);
432                            pr.println(tmpStr);
433                      }
434                      objects.close();
435                      pr.close();
436                      br.close();
437                      fr.close();
438                   } catch (IOException e) { e.printStackTrace();}
439                 }
440
441               }
442               if(count>1)
443               {
444                 try
445                 {
```

E.21: RequiredActivity.java dosyası (devam).

```
446                      RandomAccessFile objects = null;
447                      String tmpStr = "";
448                      objects = new RandomAccessFile("template-required-activity/
                                                        activity-diamond.xml", "r");
449                      File file = new File("TEST10.xml");
450                      FileWriter fr = new FileWriter(file, true);
451                      BufferedWriter br = new BufferedWriter(fr);
452                      PrintWriter pr = new PrintWriter(br);
453                      while ((tmpStr = objects.readLine()) != null) {
454                          tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST", data7a
                                                      + "-" + data8a + "-" + data6a+
                                                    "-DIAMOND-REQUEST-2-PRE-NEXT"+count);
455                          tmpStr = tmpStr.replaceAll("PRE-POST", "pre");
456                          tmpStr = tmpStr.replaceAll("CONDITION", data4a);
457                          pr.println(tmpStr);
458                      }
459                      objects.close();
460                      pr.close();
461                      br.close();
462                      fr.close();
463                     } catch (IOException e) { e.printStackTrace();}
464
465                     if (!data5a.isEmpty())
466                     {
467                       try
468                       {
469                         RandomAccessFile objects = null;
470                         String tmpStr = "";
471                         objects = new RandomAccessFile("template-required-activity/
                                                          activity-diamond.xml", "r");
472                         File file = new File("TEST10.xml");
473                         FileWriter fr = new FileWriter(file, true);
474                         BufferedWriter br = new BufferedWriter(fr);
475                         PrintWriter pr = new PrintWriter(br);
476                         while ((tmpStr = objects.readLine()) != null) {
477                                 tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST",
                                                            data7a + "-" + data8a + "-" +
                                           data6a+"-DIAMOND-REQUEST-2-POST-NEXT"+count);
478                                 tmpStr = tmpStr.replaceAll("PRE-POST", "post");
479                                 tmpStr = tmpStr.replaceAll("CONDITION", data5a);
480                                 pr.println(tmpStr);
481                         }
482                         objects.close();
483                         pr.close();
484                         br.close();
485                         fr.close();
486                        } catch (IOException e) { e.printStackTrace();}
487                     }
488
489                     try
490                     {
491                       RandomAccessFile objects = null;
492                       String tmpStr = "";
493                       objects = new RandomAccessFile("template-required-activity/
                                                        activity-request2-object-entrance.xml", "r");
494                       File file = new File("TEST10.xml");
495                       FileWriter fr = new FileWriter(file, true);
496                       BufferedWriter br = new BufferedWriter(fr);
497                       PrintWriter pr = new PrintWriter(br);
```

E.21: RequiredActivity.java dosyası (devam).

```
498                         while ((tmpStr = objects.readLine()) != null) {
499                             tmpStr = tmpStr.replaceAll("ACTIVITY-REQUEST-2", data7a
                                                        + "- " + data8a + "-" + data6a+
                                                    "-ACTIVITY-REQUEST-2-NEXT"+count);
500                             tmpStr = tmpStr.replaceAll("METHOD-NAME", data6a);
501                             pr.println(tmpStr);
502                         }
503                         objects.close();
504                         pr.close();
505                         br.close();
506                         fr.close();
507                     } catch (IOException e) { e.printStackTrace();}
508                     try
509                     {
510                       RandomAccessFile objects = null;
511                       String tmpStr = "";
512                       objects = new RandomAccessFile("deneme2/Test-Params.xml", "rw");
513                       File file = new File("TEST10.xml");
514                       FileWriter fr = new FileWriter(file, true);
515                       BufferedWriter br = new BufferedWriter(fr);
516                       PrintWriter pr = new PrintWriter(br);
517                       while ((tmpStr = objects.readLine()) != null) {
518                              pr.println(tmpStr);
519                       }
520                       objects.close();
521                       pr.close();
522                       br.close();
523                       fr.close();
524                     } catch (IOException e) { e.printStackTrace();}
525                     try
526                     {
527                       RandomAccessFile objects = null;
528                       String tmpStr = "";
529                       objects = new RandomAccessFile("template-required-activity/
                                                        activity-object-end.xml", "r");
530                       File file = new File("TEST10.xml");
531                       FileWriter fr = new FileWriter(file, true);
532                       BufferedWriter br = new BufferedWriter(fr);
533                       PrintWriter pr = new PrintWriter(br);
534                       while ((tmpStr = objects.readLine()) != null) {
535                              pr.println(tmpStr);
536                       }
537                       objects.close();
538                       pr.close();
539                       br.close();
540                       fr.close();
541                     } catch (IOException e) { e.printStackTrace();}
542                   }
543
544                 }
545                 stmt6.close();
546
547                 try
548                 {
549                     RandomAccessFile objects = null;
550                     String tmpStr = "";
551                     objects = new RandomAccessFile("template-required-activity/activity-
                                                        diamond.xml", "r");
552                     File file = new File("TEST10.xml");
```

E.21: RequiredActivity.java dosyası (devam).

```
553                   FileWriter fr = new FileWriter(file, true);
554                   BufferedWriter br = new BufferedWriter(fr);
555                   PrintWriter pr = new PrintWriter(br);
556                   while ((tmpStr = objects.readLine()) != null) {
557                          tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST", data7a
                                 + "-"  + data8a + "-" + data6a+"-DIAMOND-SEND-1-PRE");
558                          tmpStr = tmpStr.replaceAll("PRE-POST", "pre");
559                          tmpStr = tmpStr.replaceAll("CONDITION", data13a);
560                          pr.println(tmpStr);
561                   }
562                   objects.close();
563                   pr.close();
564                   br.close();
565                   fr.close();
566                } catch (IOException e) { e.printStackTrace();}
567
568                try
569                {
570                   RandomAccessFile objects = null;
571                   String tmpStr = "";
572                   objects = new RandomAccessFile("template-required-activity/activity-
                                                        diamond.xml", "r");
573                   File file = new File("TEST10.xml");
574                   FileWriter fr = new FileWriter(file, true);
575                   BufferedWriter br = new BufferedWriter(fr);
576                   PrintWriter pr = new PrintWriter(br);
577                   while ((tmpStr = objects.readLine()) != null) {
578                          tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST", data7a
                                       + "-"  + data8a + "-" + data6a+"-DIAMOND-SEND-1-POST");
579                          tmpStr = tmpStr.replaceAll("PRE-POST", "post");
580                          tmpStr = tmpStr.replaceAll("CONDITION", data14a);
581                          pr.println(tmpStr);
582                   }
583                   objects.close();
584                   pr.close();
585                   br.close();
586                   fr.close();
587                } catch (IOException e) { e.printStackTrace();}
588
589                if (!data16a.isEmpty())
590                {
591                   try
592                   {
593                     RandomAccessFile objects = null;
594                     String tmpStr = "";
595                     objects = new RandomAccessFile("template-required-activity/
                                                        activity-diamond.xml", "r");
596                     File file = new File("TEST10.xml");
597                     FileWriter fr = new FileWriter(file, true);
598                     BufferedWriter br = new BufferedWriter(fr);
599                     PrintWriter pr = new PrintWriter(br);
600                     while ((tmpStr = objects.readLine()) != null) {
601                          tmpStr = tmpStr.replaceAll("DIAMOND-SEND-REQUEST", data7a
                                 + "-" + data8a + "-" + data6a+"-DIAMOND-SEND-2-POST");
602                          tmpStr = tmpStr.replaceAll("PRE-POST", "post");
603                          tmpStr = tmpStr.replaceAll("CONDITION", data16a);
604                          pr.println(tmpStr);
605                     }
606                     objects.close();
```

E.21: RequiredActivity.java dosyası (devam).

```
607                     pr.close();
608                     br.close();
609                     fr.close();
610                    } catch (IOException e) { e.printStackTrace();}
611                   }
612
613                   try
614                   {
615                     RandomAccessFile objects = null;
616                     String tmpStr = "";
617                     objects = new RandomAccessFile("template-required-activity/
                                                        activity-lane.xml", "r");
618                     File file = new File("TEST10.xml");
619                     FileWriter fr = new FileWriter(file, true);
620                     BufferedWriter br = new BufferedWriter(fr);
621                     PrintWriter pr = new PrintWriter(br);
622                     while ((tmpStr = objects.readLine()) != null) {
623                          tmpStr = tmpStr.replaceAll("LANE-PARTITION", data7a + "-"
                                              + data8a + "-" + data6a+"-LANE-PARTITION");
624                          tmpStr = tmpStr.replaceAll("COMPONENT-NAME-REQUIRED",
                                                        data7a);
625                          tmpStr = tmpStr.replaceAll("COMPONENT-NAME-PROVIDED",
                                                        data8a);
626                          tmpStr = tmpStr.replaceAll("START-REQUEST", data7a + "-"
                                              + data8a + "-" + data6a+"-START-REQUEST");
627                          tmpStr = tmpStr.replaceAll("START-PROVIDED", data7a + "-"
                                             + data8a + "-" + data6a+"-START-PROVIDED");
628                          pr.println(tmpStr);
629                      }
630                      objects.close();
631                      pr.close();
632                      br.close();
633                      fr.close();
634                   } catch (IOException e) { e.printStackTrace();}
635                   try
636                   {
637                      RandomAccessFile objects = null;
638                      String tmpStr = "";
639                      objects = new RandomAccessFile("template-required-activity/
                                                        activity-binding.xml", "r");
640                      File file = new File("TEST10.xml");
641                      FileWriter fr = new FileWriter(file, true);
642                      BufferedWriter br = new BufferedWriter(fr);
643                      PrintWriter pr = new PrintWriter(br);
644                      while ((tmpStr = objects.readLine()) != null) {
645                          tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a +
                                                    "-" + data8a + "-" + data6a +
                                  "-DIAMOND-REQUEST-1-PRE-ACTIVITY-REQUEST-1-BINDING");
646                          tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                         data8a + "-" + data6a+"-DIAMOND-REQUEST-1-PRE");
647                          tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                           data8a + "-" + data6a+"-ACTIVITY-REQUEST-1");
648                          pr.println(tmpStr);
649                      }
650                      objects.close();
651                      pr.close();
652                      br.close();
653                      fr.close();
654                   } catch (IOException e) { e.printStackTrace();}
```

E.21: RequiredActivity.java dosyası (devam).

```
655
656                 try
657                 {
658                    RandomAccessFile objects = null;
659                    String tmpStr = "";
660                    objects = new RandomAccessFile("template-required-activity/
                                                      activity-binding.xml", "r");
661                    File file = new File("TEST10.xml");
662                    FileWriter fr = new FileWriter(file, true);
663                    BufferedWriter br = new BufferedWriter(fr);
664                    PrintWriter pr = new PrintWriter(br);
665                    while ((tmpStr = objects.readLine()) != null) {
666                         tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a +
                                                    "-" + data8a + "-" + data6a +
                                   "-ACTIVITY-REQUEST-1-DIAMOND-SEND-1-PRE-BINDING");
667                         tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                        data8a +  "-" + data6a+"-ACTIVITY-REQUEST-1");
668                         tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                         data8a + "-" + data6a+"-DIAMOND-SEND-1-PRE");
669                         pr.println(tmpStr);
670                    }
671                    objects.close();
672                    pr.close();
673                    br.close();
674                   fr.close();
675                 } catch (IOException e) { e.printStackTrace();}
676
677                 try
678                 {
679                    RandomAccessFile objects = null;
680                    String tmpStr = "";
681                    objects = new RandomAccessFile("template-required-activity/
                                                      activity-binding.xml", "r");
682                    File file = new File("TEST10.xml");
683                    FileWriter fr = new FileWriter(file, true);
684                    BufferedWriter br = new BufferedWriter(fr);
685                    PrintWriter pr = new PrintWriter(br);
686                    while ((tmpStr = objects.readLine()) != null) {
687                          tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a +
                                                      "-" +  data8a + "-" + data6a +
                                "-DIAMOND-SEND-1-PRE-ACTIVITY-SEND-1-PRE-BINDING");
688                          tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                       data8a + "-" + data6a+"-DIAMOND-SEND-1-PRE");
689                          tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                         data8a + "-" + data6a+"-ACTIVITY-SEND-1");
690                          pr.println(tmpStr);
691                    }
692                    objects.close();
693                    pr.close();
694                    br.close();
695                    fr.close();
696                 } catch (IOException e) { e.printStackTrace();}
697
698                 try
699                 {
700                    RandomAccessFile objects = null;
701                    String tmpStr = "";
702                    objects = new RandomAccessFile("template-required-activity/
                                                      activity-binding.xml", "r");
```

E.21: RequiredActivity.java dosyası (devam).

```
703                     File file = new File("TEST10.xml");
704                     FileWriter fr = new FileWriter(file, true);
705                     BufferedWriter br = new BufferedWriter(fr);
706                     PrintWriter pr = new PrintWriter(br);
707                     while ((tmpStr = objects.readLine()) != null) {
708                           tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a
                                                + "-" + data8a + "-" + data6a +
                                        "-ACTIVITY-SEND-1-DIAMOND-SEND-1-POST-BINDING");
709                           tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                              data8a + "-" + data6a+"-ACTIVITY-SEND-1");
710                           tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                          data8a + "-" + data6a+"-DIAMOND-SEND-1-POST");
711                           pr.println(tmpStr);
712                     }
713                     objects.close();
714                     pr.close();
715                     br.close();
716                     fr.close();
717                  } catch (IOException e) { e.printStackTrace();}
718
719                  try
720                  {
721                     RandomAccessFile objects = null;
722                     String tmpStr = "";
723                     objects = new RandomAccessFile("template-required-activity/
                                                      activity-binding.xml", "r");
724                     File file = new File("TEST10.xml");
725                     FileWriter fr = new FileWriter(file, true);
726                     BufferedWriter br = new BufferedWriter(fr);
727                     PrintWriter pr = new PrintWriter(br);
728                     while ((tmpStr = objects.readLine()) != null) {
729                           tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a
                                                + "-" + data8a + "-" + data6a +
                                      "-DIAMOND-SEND-1-POST-ACTIVITY-SEND-2-BINDING");
730                           tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                          data8a  + "-" + data6a+"-DIAMOND-SEND-1-POST");
731                           tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                            data8a + "-" + data6a+"-ACTIVITY-SEND-2");
732                            pr.println(tmpStr);
733                     }
734                     objects.close();
735                     pr.close();
736                     br.close();
737                     fr.close();
738                  } catch (IOException e) { e.printStackTrace();}
739
740                  if (!data16a.isEmpty())
741                  {
742                    try
743                   {
744                     RandomAccessFile objects = null;
745                     String tmpStr = "";
746                     objects = new RandomAccessFile("template-required-activity/
                                                      activity-binding.xml", "r");
747                     File file = new File("TEST10.xml");
748                     FileWriter fr = new FileWriter(file, true);
749                     BufferedWriter br = new BufferedWriter(fr);
750                     PrintWriter pr = new PrintWriter(br);
751                     while ((tmpStr = objects.readLine()) != null) {
```

E.21: RequiredActivity.java dosyası (devam).

```
752                         tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a +
                                                       "-" + data8a + "-" + data6a +
                                    "-ACTIVITY-SEND-2-DIAMOND-SEND-2-POST-BINDING");
753                         tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                              data8a + "-" + data6a+"-ACTIVITY-SEND-2");
754                         tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                        data8a + "-" + data6a+"-DIAMOND-SEND-2-POST");
755                         pr.println(tmpStr);
756                       }
757                       objects.close();
758                       pr.close();
759                       br.close();
760                       fr.close();
761                     } catch (IOException e) { e.printStackTrace();}
762                     stmt6 = c.createStatement();
763                     rs6 = stmt6.executeQuery("SELECT * FROM ReqPostCond  WHERE
                                              methodName='"+ data1b + "';");
764                     count=0;
765                     while (rs6.next())
766                     {
767                       count++;
                        data5a="";
768                       if ( (rs6.getString("preCon2")!= null) &&
                             !(rs6.getString("preCon2").isEmpty()) )
769                           data4a = rs6.getString("preCon2");
770                       if ( (rs6.getString("postCon2")!= null)  &&
                             !(rs6.getString("postCon2").isEmpty()) )
771                           data5a = rs6.getString("postCon2");
                        if (data5a.isEmpty())
                             data5a="";
772                       if (count==1)
773                       {
774                         try
775                         {
776                           RandomAccessFile objects = null;
777                           String tmpStr = "";
778                           objects = new RandomAccessFile("template-required-activity/
                                                          activity-binding.xml", "r");
779                           File file = new File("TEST10.xml");
780                           FileWriter fr = new FileWriter(file, true);
781                           BufferedWriter br = new BufferedWriter(fr);
782                           PrintWriter pr = new PrintWriter(br);
783                           while ((tmpStr = objects.readLine()) != null) {
784                                   tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2",
                                                        data7a +"-" + data8a + "-" + data6a +
                                   "-DIAMOND-SEND-2-POST-DIAMOND-REQUEST-2-PRE-BINDING");
785                                   tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-
                                        " + data8a + "-" + data6a+"-DIAMOND-SEND-2-POST");
786                                   tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                   "-" + data8a + "-" + data6a+"-DIAMOND-REQUEST-2-PRE");
787                                   pr.println(tmpStr);
788                           }
789                           objects.close();
790                           pr.close();
791                           br.close();
792                           fr.close();
793                         } catch (IOException e) { e.printStackTrace();}
794
795                         try
```

E.21: RequiredActivity.java dosyası (devam).

```
796                         {
797                           RandomAccessFile objects = null;
798                           String tmpStr = "";
799                           objects = new RandomAccessFile("template-required-activity/
                                                              activity-binding.xml", "r");
800                           File file = new File("TEST10.xml");
801                           FileWriter fr = new FileWriter(file, true);
802                           BufferedWriter br = new BufferedWriter(fr);
803                           PrintWriter pr = new PrintWriter(br);
804                           while ((tmpStr = objects.readLine()) != null) {
805                                  tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2",
                                                          data7a + "-" + data8a + "-" + data6a +
                                   "-DIAMOND-REQUEST-2-PRE-ACTIVITY-REQUEST-2-BINDING");
806                                  tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                  "-" + data8a + "-" + data6a+"-DIAMOND-REQUEST-2-PRE");
807                                  tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                      "-" +data8a + "-" + data6a+"-ACTIVITY-REQUEST-2");
808                                  pr.println(tmpStr);
809                           }
810                           objects.close();
811                           pr.close();
812                           br.close();
813                           fr.close();
814                         } catch (IOException e) { e.printStackTrace();}
815
816                         if (!data5a.isEmpty())
817                         {
818                           try
819                           {
820                             RandomAccessFile objects = null;
821                             String tmpStr = "";
822                             objects = new RandomAccessFile("template-required-activity/
                                                                activity-binding.xml", "r");
823                             File file = new File("TEST10.xml");
824                             FileWriter fr = new FileWriter(file, true);
825                             BufferedWriter br = new BufferedWriter(fr);
826                             PrintWriter pr = new PrintWriter(br);
827                             while ((tmpStr = objects.readLine()) != null) {
828                                    tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2",
                                                          data7a + "-" + data8a + "-" + data6a +
                                  "-ACTIVITY-REQUEST-2-DIAMOND-REQUEST-2-POST-BINDING");
829                                    tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                     "-" + data8a + "-" + data6a+"-ACTIVITY-REQUEST-2");
830                                    tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                 "-" + data8a + "-" + data6a+"-DIAMOND-REQUEST-2-POST");
831                                    pr.println(tmpStr);
832                             }
833                             objects.close();
834                             pr.close();
835                             br.close();
836                             fr.close();
837                           } catch (IOException e) { e.printStackTrace();}
838                         }

839                       }
840                     if(count>1)
841                     {
842                       try
843                       {
```

E.21: RequiredActivity.java dosyası (devam).

```
844                         RandomAccessFile objects = null;
845                         String tmpStr = "";
846                         objects = new RandomAccessFile("template-required-activity/
                                                    activity-binding.xml", "r");
847                         File file = new File("TEST10.xml");
848                         FileWriter fr = new FileWriter(file, true);
849                         BufferedWriter br = new BufferedWriter(fr);
850                         PrintWriter pr = new PrintWriter(br);
851                         while ((tmpStr = objects.readLine()) != null) {
852                               tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2",
                                                  data7a + "-" + data8a + "-" + data6a +
                      "-DIAMOND-SEND-2-POST-DIAMOND-REQUEST-2-PRE-BINDING-NEXT"+count);
853                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                        data8a + "-" + data6a+"-DIAMOND-SEND-2-POST");
854                               tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                    data8a + "-" + data6a+"-DIAMOND-REQUEST-2-PRE-NEXT"+count);
855                               pr.println(tmpStr);
856                           }
857                           objects.close();
858                           pr.close();
859                           br.close();
860                           fr.close();
861                         } catch (IOException e) { e.printStackTrace();}
862                         try
863                         {
864                           RandomAccessFile objects = null;
865                           String tmpStr = "";
866                           objects = new RandomAccessFile("template-required-activity/
                                                        activity-binding.xml", "r");
867                           File file = new File("TEST10.xml");
868                           FileWriter fr = new FileWriter(file, true);
869                           BufferedWriter br = new BufferedWriter(fr);
870                           PrintWriter pr = new PrintWriter(br);
871                           while ((tmpStr = objects.readLine()) != null) {
872                                   tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2",
                                                  data7a + "-" + data8a + "-" + data6a +
                      "-DIAMOND-REQUEST-2-PRE-ACTIVITY-REQUEST-2-BINDING-NEXT"+count);
873                                   tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                                        "-" + data8a + "-" + data6a+
                                          "-DIAMOND-REQUEST-2-PRE-NEXT"+count);
874                                   tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                                        "-" + data8a + "-" + data6a+
                                              "-ACTIVITY-REQUEST-2-NEXT"+count);
875                                   pr.println(tmpStr);
876                           }
877                           objects.close();
878                           pr.close();
879                           br.close();
880                           fr.close();
881                         } catch (IOException e) { e.printStackTrace();}
882
883                         if (!data5a.isEmpty())
884                         {
885                           try
886                           {
887                             RandomAccessFile objects = null;
888                             String tmpStr = "";
889                             objects = new RandomAccessFile("template-required-activity/
                                                          activity-binding.xml", "r");
890                             File file = new File("TEST10.xml");
```

E.21: RequiredActivity.java dosyası (devam).

```
                            FileWriter fr = new FileWriter(file, true);
                            BufferedWriter br = new BufferedWriter(fr);
                            PrintWriter pr = new PrintWriter(br);
                            while ((tmpStr = objects.readLine()) != null) {
                                    tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2",
                                                    data7a  + "-" + data8a + "-" + data6a +
                        "-ACTIVITY-REQUEST-2-DIAMOND-REQUEST-2-POST-BINDING-NEXT"+count);
                                    tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                           "-" + data8a + "-" + data6a+"-ACTIVITY-REQUEST-2-NEXT"+count);
                                    tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                        "-" + data8a + "-" + data6a+"-DIAMOND-REQUEST-2-POST-NEXT"+count);
                                    pr.println(tmpStr);
                             }
                            objects.close();
                            pr.close();
                            br.close();
                            fr.close();
                           } catch (IOException e) { e.printStackTrace();}
                          }
                        }
                       }
                       stmt6.close();
                    }
                    if (data16a.isEmpty())
                    {
                      try
                      {
                        RandomAccessFile objects = null;
                        String tmpStr = "";
                        objects = new RandomAccessFile("template-required-activity/
                                                        activity-binding.xml", "r");
                       File file = new File("TEST10.xml");
                       FileWriter fr = new FileWriter(file, true);
                       BufferedWriter br = new BufferedWriter(fr);
                       PrintWriter pr = new PrintWriter(br);
                       while ((tmpStr = objects.readLine()) != null) {
                              tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a +
                                                          "-" + data8a + "-" + data6a +
                                        "-ACTIVITY-SEND-2-DIAMOND-REQUEST-2-PRE-BINDING");
                              tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                                  data8a + "-" + data6a+"-ACTIVITY-SEND-2");
                              tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                              data8a +"-" + data6a+"-DIAMOND-REQUEST-2-PRE");
                              pr.println(tmpStr);
                        }
                        objects.close();
                        pr.close();
                        br.close();
                        fr.close();
                      } catch (IOException e) { e.printStackTrace();}

                      try
                      {
                        RandomAccessFile objects = null;
                        String tmpStr = "";
                        objects = new RandomAccessFile("template-required-activity/
                                                        activity-binding.xml", "r");
                        File file = new File("TEST10.xml");
                        FileWriter fr = new FileWriter(file, true);
```

E.21: RequiredActivity.java dosyası (devam).

```
940                     BufferedWriter br = new BufferedWriter(fr);
941                     PrintWriter pr = new PrintWriter(br);
942                     while ((tmpStr = objects.readLine()) != null) {
943                             tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a
                                                        + "-" + data8a + "-" + data6a +
                                    "-DIAMOND-REQUEST-2-PRE-ACTIVITY-REQUEST-2-BINDING");
944                             tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                          data8a + "-" + data6a+"-DIAMOND-REQUEST-2-PRE");
945                             tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                               data8a + "-" + data6a+"-ACTIVITY-REQUEST-2");
946                             pr.println(tmpStr);
947                     }
948                     objects.close();
949                     pr.close();
950                     br.close();
951                     fr.close();
952                   } catch (IOException e) { e.printStackTrace();}
953
954                   if (!data5a.isEmpty())
955                   {
956                     try
957                     {
958                       RandomAccessFile objects = null;
959                       String tmpStr = "";
960                       objects = new RandomAccessFile("template-required-activity/
                                                        activity-binding.xml", "r");
961                       File file = new File("TEST10.xml");
962                       FileWriter fr = new FileWriter(file, true);
963                       BufferedWriter br = new BufferedWriter(fr);
964                       PrintWriter pr = new PrintWriter(br);
965                       while ((tmpStr = objects.readLine()) != null) {
966                             tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a
                                                         + "-" + data8a + "-" + data6a +
                                  "-ACTIVITY-REQUEST-2-DIAMOND-REQUEST-2-POST-BINDING");
967                             tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                               data8a + "-" + data6a+"-ACTIVITY-REQUEST-2");
968                             tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                          data8a + "-" + data6a+"-DIAMOND-REQUEST-2-POST");
969                             pr.println(tmpStr);
970                       }
971                       objects.close();
972                       pr.close();
973                       br.close();
974                      fr.close();
975                     } catch (IOException e) { e.printStackTrace();}
976                   }
977                 }
978                 try
979                 {
980                    RandomAccessFile objects = null;
981                    String tmpStr = "";
982                    objects = new RandomAccessFile("template-required-activity/
                                                     activity-binding.xml", "r");
983                    File file = new File("TEST10.xml");
984                    FileWriter fr = new FileWriter(file, true);
985                    BufferedWriter br = new BufferedWriter(fr);
986                    PrintWriter pr = new PrintWriter(br);
987                    while ((tmpStr = objects.readLine()) != null) {
988                          tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a +"-"
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                                + data8a + "-" + data6a +
                                        "-START-REQUEST-DIAMOND-REQUEST-1-PRE-BINDING");
989                     tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                                data8a + "-" + data6a+"-START-REQUEST");
990                     tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                            data8a + "-" + data6a+"-DIAMOND-REQUEST-1-PRE");
991                     pr.println(tmpStr);
992                  }
993                  objects.close();
994                  pr.close();
995                  br.close();
996                  fr.close();
997               } catch (IOException e) { e.printStackTrace();}
998               try
999               {
1000                 RandomAccessFile objects = null;
1001                 String tmpStr = "";
1002                 objects = new RandomAccessFile("template-required-activity/
                                                    activity-binding.xml", "r");
1003                 File file = new File("TEST10.xml");
1004                 FileWriter fr = new FileWriter(file, true);
1005                 BufferedWriter br = new BufferedWriter(fr);
1006                 PrintWriter pr = new PrintWriter(br);
1007                 while ((tmpStr = objects.readLine()) != null) {
1008                        tmpStr = tmpStr.replaceAll("OBJECT-1-OBJECT-2", data7a
                                                      + "-" + data8a + "-" + data6a +
                                          "-START-PROVIDED-DIAMOND-SEND-1-PRE-BINDING");
1009                       tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-" +
                                                data8a +"-" + data6a+"-START-PROVIDED");
1010                       tmpStr = tmpStr.replaceAll("OBJECT-2", data7a + "-" +
                                             data8a + "-" + data6a+"-DIAMOND-SEND-1-PRE");
1011                       pr.println(tmpStr);
1012                 }
1013                 objects.close();
1014                 pr.close();
1015                 br.close();
1016                 fr.close();
1017              } catch (IOException e) { e.printStackTrace();}
1018
1019              try
1020              {
1021                 RandomAccessFile objects = null;
1022                 String tmpStr = "";
1023                 objects = new RandomAccessFile("template-required-activity/
                                                    activity-diyagram-entrance.xml",
"r");
1024                 File file = new File("TEST10.xml");
1025                 FileWriter fr = new FileWriter(file, true);
1026                 BufferedWriter br = new BufferedWriter(fr);
1027                 PrintWriter pr = new PrintWriter(br);
1028                 while ((tmpStr = objects.readLine()) != null) {
1029                        tmpStr = tmpStr.replaceAll("SUB-GRAPH", data7a + "-" +
                                                      data8a + "-" + data6a+"SUB-GRAPH");
1030                        pr.println(tmpStr);
1031                 }
1032                 objects.close();
1033                  pr.close();
1034                  br.close();
1035                  fr.close();
```

E.21: RequiredActivity.java dosyası (devam).

```
1036                    } catch (IOException e) { e.printStackTrace();}
1037
1038                    try
1039                    {
1040                       RandomAccessFile objects = null;
1041                       String tmpStr = "";
1042                       objects = new RandomAccessFile("template-required-activity/
                                                          activity-LaneRepr.xml", "r");
1043                       File file = new File("TEST10.xml");
1044                       FileWriter fr = new FileWriter(file, true);
1045                       BufferedWriter br = new BufferedWriter(fr);
1046                       PrintWriter pr = new PrintWriter(br);
1047                       while ((tmpStr = objects.readLine()) != null) {
1048                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                            "-" + data8a + "-" + data6a+
                                                             "-LANE-PARTITION-OBJECTREPR");
1049                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-"
                                          + data8a + "-" + data6a+"-LANE-PARTITION");
1050                               pr.println(tmpStr);
1051                       }
1052                       objects.close();
1053                       pr.close();
1054                       br.close();
1055                       fr.close();
1056                    } catch (IOException e) { e.printStackTrace();}
1057
1058                    int x=300;
1059                    int y=310;
1060                    try
1061                    {
1062                       RandomAccessFile objects = null;
1063                       String tmpStr = "";
1064                       objects = new RandomAccessFile("template-required-activity/
                                                          activity-objectRepr.xml", "r");
1065                       File file = new File("TEST10.xml");
1066                       FileWriter fr = new FileWriter(file, true);
1067                       BufferedWriter br = new BufferedWriter(fr);
1068                       PrintWriter pr = new PrintWriter(br);
1069                       while ((tmpStr = objects.readLine()) != null) {
1070                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                            "-" + data8a + "-" + data6a +
                                                        "-DIAMOND-REQUEST-1-PRE-OBJECTREPR");
1071                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-"
                                     + data8a + "-" + data6a+"-DIAMOND-REQUEST-1-PRE");
1072                               tmpStr = tmpStr.replaceAll("place=\"300,310\"",

"place=\""+x+","+y+"\"");
1073                               pr.println(tmpStr);
1074                       }
1075                       objects.close();
1076                       pr.close();
1077                       br.close();
1078                       fr.close();
1079                    } catch (IOException e) { e.printStackTrace();}
1080
1081                    try
1082                    {
1083                       x=x+200;
1084                       y=y;
```

E.21: RequiredActivity.java dosyası (devam).

```
1085                        RandomAccessFile objects = null;
1086                        String tmpStr = "";
1087                        objects = new RandomAccessFile("template-required-activity/
                                                    activity-objectRepr.xml", "r");
1088                        File file = new File("TEST10.xml");
1089                        FileWriter fr = new FileWriter(file, true);
1090                        BufferedWriter br = new BufferedWriter(fr);
1091                        PrintWriter pr = new PrintWriter(br);
1092                        while ((tmpStr = objects.readLine()) != null) {
1093                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                        "-" + data8a + "-" + data6a+
                                                    "-ACTIVITY-REQUEST-1-OBJECTREPR");
1094                              tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                    "-" + data8a + "-" + data6a+"-ACTIVITY-REQUEST-1");
1095                             tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                        "place=\""+x+","+y+"\"");
1096                             pr.println(tmpStr);
1097                        }
1098                        objects.close();
1099                        pr.close();
1100                        br.close();
1101                        fr.close();
1102                    } catch (IOException e) { e.printStackTrace();}
1103
1104                    try
1105                    {
1106                        x=x+400;
1107                        y=y;
1108                        RandomAccessFile objects = null;
1109                        String tmpStr = "";
1110                        objects = new RandomAccessFile("template-required-activity/
                                                    activity-objectRepr.xml","r");
1111                        File file = new File("TEST10.xml");
1112                        FileWriter fr = new FileWriter(file, true);
1113                        BufferedWriter br = new BufferedWriter(fr);
1114                        PrintWriter pr = new PrintWriter(br);
1115                        while ((tmpStr = objects.readLine()) != null) {
1116                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                        "-" + data8a + "-" + data6a+
                                                    "-DIAMOND-SEND-1-PRE-OBJECTREPR");
1117                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a + "-
                                     " + data8a + "-" + data6a+"-DIAMOND-SEND-1-PRE");
1118                               tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                        "place=\""+x+","+y+"\"");
1119                               pr.println(tmpStr);
1120                        }
1121                        objects.close();
1122                        pr.close();
1123                        br.close();
1124                        fr.close();
1125                     } catch (IOException e) { e.printStackTrace();}
1126
1127                     try
1128                     {
1129                        x=x+400;
1130                        y=y;
1131                        RandomAccessFile objects = null;
1132                        String tmpStr = "";
1133                        objects = new RandomAccessFile("template-required-activity/
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                    activity-objectRepr.xml","r");
1134                        File file = new File("TEST10.xml");
1135                        FileWriter fr = new FileWriter(file, true);
1136                        BufferedWriter br = new BufferedWriter(fr);
1137                        PrintWriter pr = new PrintWriter(br);
1138                        while ((tmpStr = objects.readLine()) != null) {
1139                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                    "-" + data8a + "-" + data6a+
                                                  "-ACTIVITY-SEND-1-OBJECTREPR");
1140                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                     "-" + data8a + "-" + data6a+"-ACTIVITY-SEND-1");
1141                               tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                       "place=\""+x+","+y+"\"");
1142                               pr.println(tmpStr);
1143                        }
1144                        objects.close();
1145                        pr.close();
1146                        br.close();
1147                        fr.close();
1148                       } catch (IOException e) { e.printStackTrace();}
1149
1150                       try
1151                       {
1152                          x=x+100;
1153                          y=y+200;
1154                          RandomAccessFile objects = null;
1155                          String tmpStr = "";
1156                        objects = new RandomAccessFile("template-required-activity/
                                                  activity-objectRepr.xml", "r");
1157                        File file = new File("TEST10.xml");
1158                        FileWriter fr = new FileWriter(file, true);
1159                        BufferedWriter br = new BufferedWriter(fr);
1160                        PrintWriter pr = new PrintWriter(br);
1161                        while ((tmpStr = objects.readLine()) != null) {
1162                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                    "-" + data8a + "-" + data6a+
                                                "-DIAMOND-SEND-1-POST-OBJECTREPR");
1163                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                       "-" + data8a + "-" + data6a+"-DIAMOND-SEND-1-POST");
1164                               tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                       "place=\""+x+","+y+"\"");
1165                                pr.println(tmpStr);
1166                        }
1167                        objects.close();
1168                        pr.close();
1169                        br.close();
1170                        fr.close();
1171                      } catch (IOException e) { e.printStackTrace();}
1172
1173                      try
1174                      {
1175                        x=x-300;
1176                        y=y;
1177                        RandomAccessFile objects = null;
1178                        String tmpStr = "";
1179                        objects = new RandomAccessFile("template-required-activity/
                                                  activity-objectRepr.xml", "r");
1180                        File file = new File("TEST10.xml");
1181                        FileWriter fr = new FileWriter(file, true);
```

E.21: RequiredActivity.java dosyası (devam).

```
1182                        BufferedWriter br = new BufferedWriter(fr);
1183                        PrintWriter pr = new PrintWriter(br);
1184                        while ((tmpStr = objects.readLine()) != null) {
1185                               tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                                  "-" + data8a + "-" + data6a+
                                                "-ACTIVITY-SEND-2-OBJECTREPR");
1186                               tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                     "-" + data8a + "-" + data6a+"-ACTIVITY-SEND-2");
1187                               tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                   "place=\""+x+","+y+"\"");
1188                               pr.println(tmpStr);
1189                        }
1190                        objects.close();
1191                        pr.close();
1192                        br.close();
1193                        fr.close();
1194                      } catch (IOException e) { e.printStackTrace();}
1195
1196                      if (!data16a.isEmpty())
1197                      {
1198                        try
1199                        {
1200                          x=x-200;
1201                          y=y;
1202                          RandomAccessFile objects = null;
1203                          String tmpStr = "";
1204                          objects =new RandomAccessFile("template-required-activity/
                                                  activity-objectRepr.xml","r");
1205                          File file = new File("TEST10.xml");
1206                          FileWriter fr = new FileWriter(file, true);
1207                          BufferedWriter br = new BufferedWriter(fr);
1208                          PrintWriter pr = new PrintWriter(br);
1209                          while ((tmpStr = objects.readLine()) != null) {
1210                                tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a
                                                  + "-" +data8a+"-"+data6a+
                                          "-DIAMOND-SEND-2-POST-OBJECTREPR");
1211                                tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                                  "-" + data8a + "-" + data6a+
                                                   "-DIAMOND-SEND-2-POST");
1212                                tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                   "place=\""+x+","+y+"\"");
1213                                pr.println(tmpStr);
1214                           }
1215                          objects.close();
1216                          pr.close();
1217                          br.close();
1218                          fr.close();
1219                        } catch (IOException e) { e.printStackTrace();}
1220                        stmt6 = c.createStatement();
1221                        rs6 = stmt6.executeQuery("SELECT * FROM ReqPostCond  WHERE
                                          methodName='" + data1b + "';");
1222                       count=0;
1223                       int nextX=x;
1224                       int nextY=y;
1225
1226                        while (rs6.next())
1227                        {
1228                          count++;
1229                          x=nextX;
```

E.21: RequiredActivity.java dosyası (devam).

```
1230                          y=nextY;
                               data5a="";
1231                        if ( (rs6.getString("preCon2")!= null) &&
                                 !(rs6.getString("preCon2").isEmpty()) )
1232                           data4a = rs6.getString("preCon2");
1233                        if ( (rs6.getString("postCon2")!= null)  &&
                                 !(rs6.getString("postCon2").isEmpty()) )
1234                           data5a = rs6.getString("postCon2");
                           if (data5a.isEmpty())
                               data5a="";
1235                        if (count==1)
1236                        {
1237                          try
1238                          {
1239                             x=x-200;
1240                             y=y;
1241                             RandomAccessFile objects = null;
1242                             String tmpStr = "";
1243                             objects =
                                      new RandomAccessFile("template-required-activity/
                                                            activity-objectRepr.xml", "r");
1244                             File file = new File("TEST10.xml");
1245                             FileWriter fr = new FileWriter(file, true);
1246                             BufferedWriter br = new BufferedWriter(fr);
1247                             PrintWriter pr = new PrintWriter(br);
1248                             while ((tmpStr = objects.readLine()) != null) {
1249                                   tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                 data7a + "-" + data8a + "-" + data6a
                                               + "-DIAMOND-REQUEST-2-PRE-OBJECTREPR");
1250                                   tmpStr = tmpStr.replaceAll("OBJECT-1", data7a
                                                        + "-" + data8a + "-" + data6a
                                                          +"-DIAMOND-REQUEST-2-PRE");
1251                                   tmpStr =
                                          tmpStr.replaceAll("place=\"300,310\"",
                                                            "place=\""+x+","+y+"\"");
1252                                   pr.println(tmpStr);
1253                              }
1254                              objects.close();
1255                              pr.close();
1256                              br.close();
1257                              fr.close();
1258                           } catch (IOException e) { e.printStackTrace();}
1259
1260                           try
1261                           {
1262                              x=x-200;
1263                              y=y;
1264                              RandomAccessFile objects = null;
1265                              String tmpStr = "";
1266                              objects =
                                       new RandomAccessFile("template-required-activity/
                                                            activity-objectRepr.xml", "r");
1267                              File file = new File("TEST10.xml");
1268                              FileWriter fr = new FileWriter(file, true);
1269                              BufferedWriter br = new BufferedWriter(fr);
1270                              PrintWriter pr = new PrintWriter(br);
1271                              while ((tmpStr = objects.readLine()) != null) {
1272                                    tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                  data7a + "-" + data8a + "-" + data6a +
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                    "-ACTIVITY-REQUEST-2-OBJECTREPR");
1273                                    tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                    data7a + "-" + data8a + "-" +
                                                   data6a+"-ACTIVITY-REQUEST-2");
1274                                    tmpStr =
                                              tmpStr.replaceAll("place=\"300,310\"",
                                                      "place=\""+x+","+y+"\"");
1275                                    pr.println(tmpStr);
1276                                 }
1277                                 objects.close();
1278                                 pr.close();
1279                                 br.close();
1280                                 fr.close();
1281                              } catch (IOException e) { e.printStackTrace();}
1282
1283                              if (!data5a.isEmpty())
1284                              {
1285                                 try
1286                                 {
1287                                   x=x-200;
1288                                   y=y;
1289                                   RandomAccessFile objects = null;
1290                                   String tmpStr = "";
1291                                   objects =
                                         new RandomAccessFile("template-required-activity/
                                                        activity-objectRepr.xml", "r");
1292                                   File file = new File("TEST10.xml");
1293                                   FileWriter fr = new FileWriter(file, true);
1294                                   BufferedWriter br = new BufferedWriter(fr);
1295                                   PrintWriter pr = new PrintWriter(br);
1296                                   while ((tmpStr = objects.readLine()) != null) {
1297                                        tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                       data7a +  "-" + data8a + "-" +
                                                  data6a + "-DIAMOND-REQUEST-2-POST-OBJECTREPR");
1298                                        tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                         data7a + "-" + data8a + "-"
                                                    + data6a+"-DIAMOND-REQUEST-2-POST");
1299                                        tmpStr =
                                              tmpStr.replaceAll("place=\"300,310\"",
                                                        "place=\""+x+","+y+"\"");
1300                                        pr.println(tmpStr);
1301                                    }
1302                                    objects.close();
1303                                    pr.close();
1304                                    br.close();
1305                                    fr.close();
1306                                  } catch (IOException e) { e.printStackTrace();}
1307                               }

1308                           }
1309                           if(count>1)
1310                           {
1311                             try
1312                             {
1313                               x=x-200;
1314                               y=y+150;
1315
1316
1317                               RandomAccessFile objects = null;
```

E.21: RequiredActivity.java dosyası (devam).

```
1318                                String tmpStr = "";
1319                                objects =
                                          new RandomAccessFile("template-required-activity/
                                                                activity-objectRepr.xml","r");
1320                                File file = new File("TEST10.xml");
1321                                FileWriter fr = new FileWriter(file, true);
1322                                BufferedWriter br = new BufferedWriter(fr);
1323                                PrintWriter pr = new PrintWriter(br);
1324                                while ((tmpStr = objects.readLine()) != null) {
1325                                      tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                  data7a + "-" + data8a + "-" + data6a +
                                       "-DIAMOND-REQUEST-2-PRE-OBJECTREPR-NEXT"+count);
1326                                      tmpStr = tmpStr.replaceAll("OBJECT-1", data7a
                                                          + "-"+ data8a + "-" + data6a +
                                                   "-DIAMOND-REQUEST-2-PRE-NEXT"+count);
1327                                      tmpStr =
                                                tmpStr.replaceAll("place=\"300,310\"",
                                                                  "place=\""+x+","+y+"\"");
1328                                      pr.println(tmpStr);
                                    }
                                    objects.close();
                                    pr.close();
                                    br.close();
                                    fr.close();
                                } catch (IOException e) { e.printStackTrace();}

                                try
                                {
                                    x=x-200;
                                    y=y;

                                    RandomAccessFile objects = null;
                                    String tmpStr = "";
                                    objects =
                                          new RandomAccessFile("template-required-activity/
                                                                activity-objectRepr.xml", "r");
                                    File file = new File("TEST10.xml");
                                    FileWriter fr = new FileWriter(file, true);
                                    BufferedWriter br = new BufferedWriter(fr);
                                    PrintWriter pr = new PrintWriter(br);
                                    while ((tmpStr = objects.readLine()) != null) {
                                           tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                    data7a + "-" + data8a + "-" + data6a +
                                             "-ACTIVITY-REQUEST-2-OBJECTREPR-NEXT"+count);
                                           tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                    data7a + "-" + data8a + "-" + data6a +
                                                        "-ACTIVITY-REQUEST-2-NEXT"+count);
                                           tmpStr =
                                                  tmpStr.replaceAll("place=\"300,310\"",
                                                                    "place=\""+x+","+y+"\"");
                                           pr.println(tmpStr);
                                    }
                                    objects.close();
                                    pr.close();
                                    br.close();
                                    fr.close();
                                } catch (IOException e) { e.printStackTrace();}

                                if (!data5a.isEmpty())
```

E.21: RequiredActivity.java dosyası (devam).

```
1361                                {
1362                                  try
1363                                  {
1364                                    x=x-200;
1365                                    y=y;
1366                                    RandomAccessFile objects = null;
1367                                    String tmpStr = "";
1368                                    objects =
                                          new RandomAccessFile("template-required-activity/
                                                                activity-objectRepr.xml", "r");
1369                                    File file = new File("TEST10.xml");
1370                                    FileWriter fr = new FileWriter(file, true);
1371                                    BufferedWriter br = new BufferedWriter(fr);
1372                                    PrintWriter pr = new PrintWriter(br);
1373                                    while ((tmpStr = objects.readLine()) != null) {
1374                                          tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                 data7a + "-" + data8a + "-" + data6a +
                                      "-DIAMOND-REQUEST-2-POST-OBJECTREPR-NEXT"+count);
1375                                          tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                 data7a + "-" + data8a + "-" + data6a +
                                                 "-DIAMOND-REQUEST-2-POST-NEXT"+count);
1376                                          tmpStr =
                                               tmpStr.replaceAll("place=\"300,310\"",
                                                              "place=\""+x+","+y+"\"");
1377                                          pr.println(tmpStr);
1378                                     }
1379                                     objects.close();
1380                                     pr.close();
1381                                     br.close();
1382                                     fr.close();
1383                                   } catch (IOException e) { e.printStackTrace();}
1384                                }
1385                             }
1386
1387                          }
1388                          stmt6.close();
1389                     }
1390                     if (data16a.isEmpty())
1391                     {
1392                       try
1393                       {
1394                          x=x-400;
1395                          y=y;
1396                          RandomAccessFile objects = null;
1397                          String tmpStr = "";
1398                          objects =
                                    new RandomAccessFile("template-required-activity/
                                                          activity-objectRepr.xml", "r");
1399                          File file = new File("TEST10.xml");
1400                          FileWriter fr = new FileWriter(file, true);
1401                          BufferedWriter br = new BufferedWriter(fr);
1402                          PrintWriter pr = new PrintWriter(br);
1403                          while ((tmpStr = objects.readLine()) != null) {
1404                                tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                                "-" + data8a+"-"+data6a+"-DIAMOND-REQUEST-2-PRE-OBJECTREPR");
1405                                tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                "-" + data8a + "-" + data6a+"-DIAMOND-REQUEST-2-PRE");
1406                                tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                          "place=\""+x+","+y+"\"");
```

E.21: RequiredActivity.java dosyası (devam).

```
1407                                pr.println(tmpStr);
1408                             }
1409                             objects.close();
1410                             pr.close();
1411                             br.close();
1412                             fr.close();
1413                            } catch (IOException e) { e.printStackTrace();}
1414
1415                            try
1416                            {
1417                               x=x-200;
1418                               y=y;
1419                               RandomAccessFile objects = null;
1420                               String tmpStr = "";
1421                               objects =
                                            new RandomAccessFile("template-required-activity/
                                                                  activity-objectRepr.xml", "r");
1422                               File file = new File("TEST10.xml");
1423                               FileWriter fr = new FileWriter(file, true);
1424                               BufferedWriter br = new BufferedWriter(fr);
1425                               PrintWriter pr = new PrintWriter(br);
1426                               while ((tmpStr = objects.readLine()) != null) {
1427                                     tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a
                                  + "-" +data8a+"-" +data6a+"-ACTIVITY-REQUEST-2-OBJECTREPR");
1428                                     tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                       "-" + data8a + "-" + data6a+"-ACTIVITY-REQUEST-2");
1429                                     tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                                  "place=\""+x+","+y+"\"");
1430                                     pr.println(tmpStr);
1431                                }
1432                               objects.close();
1433                               pr.close();
1434                               br.close();
1435                               fr.close();
1436                             } catch (IOException e) { e.printStackTrace();}
1437
1438                             if (!data5a.isEmpty())
1439                             {
1440                               try
1441                               {
1442                                   x=x-200;
1443                                   y=y;
1444                                   RandomAccessFile objects = null;
1445                                   String tmpStr = "";
1446                                   objects =
                                            new RandomAccessFile("template-required-activity/
                                                                  activity-objectRepr.xml", "r");
1447                                   File file = new File("TEST10.xml");
1448                                   FileWriter fr = new FileWriter(file, true);
1449                                   BufferedWriter br = new BufferedWriter(fr);
1450                                   PrintWriter pr = new PrintWriter(br);
1451                                   while ((tmpStr = objects.readLine()) != null) {
1452                                       tmpStr = tmpStr.replaceAll("OBJECTREPR",
                                                        data7a + "-" + data8a + "-" + data6a +
                                                        "-DIAMOND-REQUEST-2-POST-OBJECTREPR");
1453                                          tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                         data7a +"-" + data8a + "-" + data6a +
                                                                  "-DIAMOND-REQUEST-2-POST");
1454                                          tmpStr =
```

E.21: RequiredActivity.java dosyası (devam).

```
                                        tmpStr.replaceAll("place=\"300,310\"",
                                                "place=\""+x+","+y+"\"");
1455                                pr.println(tmpStr);
1456                            }
1457                            objects.close();
1458                            pr.close();
1459                            br.close();
1460                            fr.close();
1461                        } catch (IOException e) { e.printStackTrace();}
1462                    }
1463                }
1464
1465                try
1466                {
1467                    RandomAccessFile objects = null;
1468                    String tmpStr = "";
1469                    objects =
                                new RandomAccessFile("template-required-activity/
                                                activity-objectRepr.xml", "r");
1470                    File file = new File("TEST10.xml");
1471                    FileWriter fr = new FileWriter(file, true);
1472                    BufferedWriter br = new BufferedWriter(fr);
1473                    PrintWriter pr = new PrintWriter(br);
1474                    while ((tmpStr = objects.readLine()) != null) {
1475                        tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a +
                          "-" + data8a + "-" + data6a+"-START-REQUEST-OBJECTREPR");
1476                        tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                 "-" + data8a + "-" + data6a+"-START-REQUEST");
1477                        tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                "place=\""+400+","+200+"\"");
1478                        pr.println(tmpStr);
1479                    }
1480                    objects.close();
1481                    pr.close();
1482                    br.close();
1483                    fr.close();
1484                } catch (IOException e) { e.printStackTrace();}
1485                try
1486                {
1487                    RandomAccessFile objects = null;
1488                    String tmpStr = "";
1489                    objects =
                                new RandomAccessFile("template-required-activity/
                                                activity-objectRepr.xml", "r");
1490                    File file = new File("TEST10.xml");
1491                    FileWriter fr = new FileWriter(file, true);
1492                    BufferedWriter br = new BufferedWriter(fr);
1493                    PrintWriter pr = new PrintWriter(br);
1494                    while ((tmpStr = objects.readLine()) != null) {
1495                        tmpStr = tmpStr.replaceAll("OBJECTREPR", data7a
                                            + "-" + data8a + "-" + data6a +
                                              "-START-PROVIDED-OBJECTREPR");
1496                        tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                 "-" + data8a + "-" + data6a+"-START-PROVIDED");
1497                        tmpStr = tmpStr.replaceAll("place=\"300,310\"",
                                                "place=\""+1100+","+200+"\"");
1498                        pr.println(tmpStr);
1499                    }
1500                    objects.close();
```

E.21: RequiredActivity.java dosyası (devam).

```
1501                            pr.close();
1502                            br.close();
1503                            fr.close();
1504                           } catch (IOException e) { e.printStackTrace();}
1505
1506                           try
1507                           {
1508                             RandomAccessFile objects = null;
1509                             String tmpStr = "";
1510                             objects =
                                        new RandomAccessFile("template-required-activity/
                                                        activity-bindingRepr.xml", "r");
1511                              File file = new File("TEST10.xml");
1512                              FileWriter fr = new FileWriter(file, true);
1513                              BufferedWriter br = new BufferedWriter(fr);
1514                              PrintWriter pr = new PrintWriter(br);
1515                              while ((tmpStr = objects.readLine()) != null) {
1516                                   tmpStr = tmpStr.replaceAll("SUB-BINDING", data7a
                                                        + "-" + data8a + "-" + data6a +
                                      "-DIAMOND-REQUEST-1-PRE-ACTIVITY-REQUEST-1-BINDING");
1517                                   tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                                        "-" + data8a + "-" + data6a +
                                                  "-DIAMOND-REQUEST-1-PRE-OBJECTREPR");
1518                                   tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                                        "-" + data8a + "-" + data6a +
                                                "-ACTIVITY-REQUEST-1-OBJECTREPR");
1519                                   tmpStr = tmpStr.replaceAll("place=\"350,310\"",
                                                        "place=\""+350+","+310+"\"");
1520                                   pr.println(tmpStr);
1521                               }
1522                               objects.close();
1523                               pr.close();
1524                               br.close();
1525                               fr.close();
1526                            } catch (IOException e) { e.printStackTrace();}
1527
1528                            try
1529                            {
1530                               RandomAccessFile objects = null;
1531                               String tmpStr = "";
1532                               objects =
                                        new RandomAccessFile("template-required-activity/
                                                        activity-bindingRepr.xml", "r");
1533                               File file = new File("TEST10.xml");
1534                               FileWriter fr = new FileWriter(file, true);
1535                               BufferedWriter br = new BufferedWriter(fr);
1536                               PrintWriter pr = new PrintWriter(br);
1537                               while ((tmpStr = objects.readLine()) != null) {
1538                                      tmpStr = tmpStr.replaceAll("SUB-BINDING",
                                                        data7a + "-" + data8a + "-" + data6a +
                                             "-ACTIVITY-REQUEST-1-DIAMOND-SEND-1-PRE-BINDING");
1539                                      tmpStr = tmpStr.replaceAll("OBJECT-1", data7a
                                                        + "-" + data8a + "-" + data6a +
                                                "-ACTIVITY-REQUEST-1-OBJECTREPR");
1540                                      tmpStr = tmpStr.replaceAll("OBJECT-2", data7a
                                                        + "-" + data8a + "-" + data6a +
                                                "-DIAMOND-SEND-1-PRE-OBJECTREPR");
1541                                      tmpStr =
                                                 tmpStr.replaceAll("place=\"350,310\"",
                                                        "place=\""+610+","+310+"\"");
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                    "place=\""+1360+","+510+"\"");
1609                                      pr.println(tmpStr);
1610                                    }
1611                                    objects.close();
1612                                    pr.close();
1613                                      br.close();
1614                                      fr.close();
1615                                    } catch (IOException e) { e.printStackTrace();}
1616
1617                                    if (!data16a.isEmpty())
1618                                    {
1619                                      try
1620                                      {
1621                                         RandomAccessFile objects = null;
1622                                         String tmpStr = "";
1623                                         objects =
                                          new RandomAccessFile("template-required-activity/
                                                        activity-bindingRepr.xml", "r");
1624                                         File file = new File("TEST10.xml");
1625                                         FileWriter fr = new FileWriter(file, true);
1626                                         BufferedWriter br = new BufferedWriter(fr);
1627                                         PrintWriter pr = new PrintWriter(br);
1628                                         while ((tmpStr = objects.readLine()) != null) {
1629                                               tmpStr = tmpStr.replaceAll(
                                                                "SUB-BINDING", data7a + "-"
                                                                   + data8a + "-" + data6a +
                                                "-ACTIVITY-SEND-2-DIAMOND-SEND-2-POST-BINDING");
1630                                                tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                             data7a + "-" + data8a + "-" +
                                                   data6a + "-ACTIVITY-SEND-2-OBJECTREPR");
1631                                                tmpStr = tmpStr.replaceAll("OBJECT-2",
                                                    data7a + "-" + data8a + "-" + data6a +
                                                       "-DIAMOND-SEND-2-POST-OBJECTREPR");
1632                                                tmpStr =
                                                   tmpStr.replaceAll("place=\"350,310\"",
                                                             "place=\""+1000+","+510+"\"");
1633                                                pr.println(tmpStr);
1634                                           }
1635                                           objects.close();
1636                                           pr.close();
1637                                           br.close();
1638                                           fr.close();
1639                                      } catch (IOException e) { e.printStackTrace();}
1640                                      stmt6 = c.createStatement();
1641                                      rs6 = stmt6.executeQuery("SELECT * FROM ReqPostCond
                                                    WHERE  metotName='" + data1b + "';");
1642                                      count=0;
1643                                      while (rs6.next())
1644                                      {
1645                                        count++;
                                          data5a="";
1646                                        if ( (rs6.getString("preCon2")!= null) &&
                                                !(rs6.getString("preCon2").isEmpty()) )
1647                                          data4a = rs6.getString("preCon2");
1648                                        if ( (rs6.getString("postCon2")!= null)  &&
                                                !(rs6.getString("postCon2").isEmpty()) )
1649                                          data5a = rs6.getString("postCon2");
1650
1651                                        if (count==1)
```

E.21: RequiredActivity.java dosyası (devam).

```
1652                                    {
1653
1654                                      try
1655                                      {
1656                                        RandomAccessFile objects = null;
1657                                        String tmpStr = "";
1658                                        objects =
                                          new RandomAccessFile("template-required-activity
                                                              /activity-bindingRepr.xml","r");
1659                                        File file = new File("TEST10.xml");
1660                                        FileWriter fr = new FileWriter(file, true);
1661                                        BufferedWriter br = new BufferedWriter(fr);
1662                                        PrintWriter pr = new PrintWriter(br);
1663                                        while ((tmpStr = objects.readLine()) !=
                                                                                  null) {
1664                                             tmpStr = tmpStr.replaceAll(
                                                                  "SUB-BINDING", data7a +
                                                             "-" + data8a + "-" + data6a +
                                        "-DIAMOND-SEND-2-POST-DIAMOND-REQUEST-2-PRE-BINDING");
1665                                             tmpStr = tmpStr.replaceAll(
                                                               "OBJECT-1", data7a + "-" +
                                                                 data8a + "-" + data6a +
                                                     "-DIAMOND-SEND-2-POST-OBJECTREPR");
1666                                            tmpStr = tmpStr.replaceAll(
                                                               "OBJECT-2", data7a + "-" +
                                                                 data8a + "-" + data6a +
                                                "-DIAMOND-REQUEST-2-PRE-OBJECTREPR");
1667                                             tmpStr = tmpStr.replaceAll(
                                                                  " place= \"350,310\"",
                                                           "place=\""+870+","+510+"\"");
1668                                             pr.println(tmpStr);
1669                                        }
1670                                        objects.close();
1671                                        pr.close();
1672                                        br.close();
1673                                        fr.close();
1674                                      } catch (IOException e) { e.printStackTrace();}
1675
1676                                      try
1677                                      {
1678                                       RandomAccessFile objects = null;
1679                                       String tmpStr = "";
1680                                       objects = new RandomAccessFile(
                                                             "template-required-activity/
                                                             activity-bindingRepr.xml", "r");
1681                                       File file = new File("TEST10.xml");
1682                                       FileWriter fr = new FileWriter(file, true);
1683                                       BufferedWriter br = new BufferedWriter(fr);
1684                                       PrintWriter pr = new PrintWriter(br);
1685                                       while ((tmpStr = objects.readLine()) !=
                                                                                  null) {
1686                                           tmpStr = tmpStr.replaceAll(
                                                                  "SUB-BINDING", data7a + "-"
                                                                    + data8a + "-" + data6a +
                                        "-DIAMOND-REQUEST-2-PRE-ACTIVITY-REQUEST-2-BINDING");
1687                                             tmpStr = tmpStr.replaceAll("OBJECT-1",
                                                             data7a + "-" + data8a + "-" +
                                          data6a +"-DIAMOND-REQUEST-2-PRE-OBJECTREPR");
1688                                             tmpStr = tmpStr.replaceAll("OBJECT-2",
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                    data7a + "-" + data8a + "-" +
                                                data6a + "-ACTIVITY-REQUEST-2-OBJECTREPR");
1689                                             tmpStr =
                                                 tmpStr.replaceAll("place=\"350,310\"",
                                                            "place=\""+630+","+510+"\"");
1690                                             pr.println(tmpStr);
1691                                           }
1692                                           objects.close();
1693                                           pr.close();
1694                                           br.close();
1695                                           fr.close();
1696                                         } catch (IOException e) { e.printStackTrace();}
1697
1698                                         if (!data5a.isEmpty())
1699                                         {
1700                                           try
1701                                           {
1702                                            RandomAccessFile objects = null;
1703                                            String tmpStr = "";
1704                                            objects = new RandomAccessFile(
                                                            "template-required-activity/
                                                        activity-bindingRepr.xml", "r");
1705                                            File file = new File("TEST10.xml");
1706                                            FileWriter fr = new FileWriter(file, true);
1707                                            BufferedWriter br = new BufferedWriter(fr);
1708                                            PrintWriter pr = new PrintWriter(br);
1709                                            while((tmpStr = objects.readLine()) !=
                                                                                    null) {
1710                                                tmpStr = tmpStr.replaceAll(
                                                                "SUB-BINDING", data7a +
                                                            "-" + data8a + "-" + data6a +
                                                "-ACTIVITY-REQUEST-2-DIAMOND-REQUEST-2-POST-BINDING");
1711                                                tmpStr = tmpStr.replaceAll(
                                                                "OBJECT-1",data7a + "-" +
                                                                  data8a + "-" + data6a +
                                                        "-ACTIVITY-REQUEST-2-OBJECTREPR");
1712                                                tmpStr = tmpStr.replaceAll(
                                                                "OBJECT-2",data7a + "-" +
                                                                  data8a + "-" + data6a +
                                                          "-DIAMOND-REQUEST-2-POST-OBJECTREPR");
1713                                                tmpStr =
                                                       tmpStr.replaceAll("place=\"350,310\"",
                                                                "place=\""+390+","+510+"\"");
1714
                                                    pr.println(tmpStr);
1715                                             }
1716                                                objects.close();
1717                                                pr.close();
1718                                                br.close();
1719                                                fr.close();
1720                                            } catch (IOException e) { e.printStackTrace();}
1721                                          }

1722                                        }
1723                                        if(count>1)
1724                                        {
1725
1726                                         try
1727                                         {
```

E.21: RequiredActivity.java dosyası (devam).

```
1728                                    RandomAccessFile objects = null;
1729                                    String tmpStr = "";
1730                                    objects = new RandomAccessFile(
                                                    "template-required-activity/
                                                    activity-bindingRepr.xml", "r");
1731                                    File file = new File("TEST10.xml");
1732                                    FileWriter fr = new FileWriter(file, true);
1733                                    BufferedWriter br = new BufferedWriter(fr);
1734                                    PrintWriter pr = new PrintWriter(br);
1735                                    while ((tmpStr = objects.readLine()) != null)
                                        {
1736                                            tmpStr = tmpStr.replaceAll(
                                                                "SUB-BINDING", data7a +
                                                            "-" + data8a + "-" + data6a +
                        "-DIAMOND-SEND-2-POST-DIAMOND-REQUEST-2-PRE-BINDING-NEXT"+count);
1737                                        tmpStr = tmpStr.replaceAll(
                                                                "OBJECT-1", data7a +
                                                            "-" + data8a + "-" + data6a +
                                                        "-DIAMOND-SEND-2-POST-OBJECTREPR");
1738                                        tmpStr = tmpStr.replaceAll(
                                                                "OBJECT-2", data7a +
                                                            "-" + data8a + "-" + data6a +
                                    "-DIAMOND-REQUEST-2-PRE-OBJECTREPR-NEXT"+count);

1739                                            tmpStr =
                                                tmpStr.replaceAll("place=\"350,310\"",
                                                            "place=\""+870+","+660+"\"");
1740                                        pr.println(tmpStr);
1741                                     }
1742                                     objects.close();
1743                                     pr.close();
1744                                     br.close();
1745                                     fr.close();
1746                                } catch (IOException e) { e.printStackTrace();}
1747
1748                                try
1749                                {
1750                                    RandomAccessFile objects = null;
1751                                    String tmpStr = "";
1752                                    objects = new RandomAccessFile(
                                                        "template-required-activity/
                                                        activity-bindingRepr.xml", "r");
1753                                    File file = new File("TEST10.xml")
1754                                    FileWriter fr = new FileWriter(file, true);
1755                                    BufferedWriter br = new BufferedWriter(fr);
1756                                    PrintWriter pr = new PrintWriter(br);
1757                                    while ((tmpStr = objects.readLine()) != null)
                                        {
1758                                            tmpStr = tmpStr.replaceAll(
                                                                "SUB-BINDING", data7a +
                                                            "-" + data8a + "-" + data6a +
                        "-DIAMOND-REQUEST-2-PRE-ACTIVITY-REQUEST-2-BINDING-NEXT"+count);
1759                                        tmpStr = tmpStr.replaceAll(
                                                                "OBJECT-1", data7a +
                                                            "-" + data8a + "-" + data6a +
                                    "-DIAMOND-REQUEST-2-PRE-OBJECTREPR-NEXT"+count);

1760                                            tmpStr = tmpStr.replaceAll(
                                                                "OBJECT-2", data7a +
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                    "-" + data8a + "-" + data6a +
                                        "-ACTIVITY-REQUEST-2-OBJECTREPR-NEXT"+count);
1761                                          tmpStr =
                                         tmpStr.replaceAll("place=\"350,310\"",
                                                    "place=\""+630+","+660+"\"");
1762                                      pr.println(tmpStr);
1763                                   }
1764                                   objects.close();
1765                                   pr.close();
1766                                    br.close();
1767                                    fr.close();
1768                                } catch (IOException e) { e.printStackTrace();}
1769
1770                                if (!data5a.isEmpty())
1771                                {
1772                                  try
1773                                  {
1774                                      RandomAccessFile objects = null;
1775                                      String tmpStr = "";
1776                                      objects = new RandomAccessFile(
                                                     "template-required-activity
                                                    /activity-bindingRepr.xml", "r");
1777                                      File file = new File("TEST10.xml");
1778                                      FileWriter fr = new FileWriter(file, true);
1779                                      BufferedWriter br = new BufferedWriter(fr);
1780                                      PrintWriter pr = new PrintWriter(br);
1781                                      while ((tmpStr = objects.readLine()) != null)
                                         {
1782                                              tmpStr = tmpStr.replaceAll(
                                                                "SUB-BINDING", data7a +
                                                    "-" + data8a + "-" + data6a +
                                "-ACTIVITY-REQUEST-2-DIAMOND-REQUEST-2-POST-BINDING-NEXT"+count);
1783                                              tmpStr = tmpStr.replaceAll(
                                                                  "OBJECT-1", data7a +
                                                    "-" + data8a + "-" + data6a +
                                        "-ACTIVITY-REQUEST-2-OBJECTREPR-NEXT"+count);
1784                                              tmpStr = tmpStr.replaceAll(
                                                                  "OBJECT-2", data7a +
                                                    "-" + data8a + "-" + data6a +
                                     "-DIAMOND-REQUEST-2-POST-OBJECTREPR-NEXT"+count);

1785                                              tmpStr =
                                                 tmpStr.replaceAll("place=\"350,310\"",
                                                    "place=\""+390+","+660+"\"");
1786                                              pr.println(tmpStr);
1787                                          }
1788                                         objects.close();
1789                                         pr.close();
1790                                         br.close();
1791                                         fr.close();
1792                                       } catch (IOException e) { e.printStackTrace();}
1793                                   }
1794                                 }
1795
1796                               }
1797                               stmt6.close();
1798                            }
1799                            if (data16a.isEmpty())
1800                            {
```

E.21: RequiredActivity.java dosyası (devam).

```
1801                        try
1802                        {
1803                          RandomAccessFile objects = null;
1804                          String tmpStr = "";
1805                          objects = new RandomAccessFile(
                                                "template-required-activity/
                                                 activity-bindingRepr.xml", "r");
1806                          File file = new File("TEST10.xml");
1807                          FileWriter fr = new FileWriter(file, true);
1808                          BufferedWriter br = new BufferedWriter(fr);
1809                          PrintWriter pr = new PrintWriter(br);
1810                          while ((tmpStr = objects.readLine()) != null) {
1811                             tmpStr = tmpStr.replaceAll("SUB-BINDING", data7a +
                                                      "-" + data8a + "-" + data6a +
                                  "-ACTIVITY-SEND-2-DIAMOND-REQUEST-2-PRE-BINDING");
1812                              tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                               "-" + data8a + "-" +data6a+"-ACTIVITY-SEND-2-OBJECTREPR");
1813                              tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                                        "-" + data8a+"-" +data6a +
                                                "-DIAMOND-REQUEST-2-PRE-OBJECTREPR");
1814                              tmpStr = tmpStr.replaceAll("place=\"350,310\"",
                                                     "place=\""+870+","+510+"\"");
1815                              pr.println(tmpStr);

1816                           }
1817                           objects.close();
1818                           pr.close();
1819                           br.close();
1820                             fr.close();
1821                        } catch (IOException e) { e.printStackTrace();}
1822
1823                        try
1824                        {
1825                          RandomAccessFile objects = null;
1826                          String tmpStr = "";
1827                          objects = new RandomAccessFile(
                                                 "template-required-activity/
                                                activity-bindingRepr.xml", "r");
1828                          File file = new File("TEST10.xml");
1829                          FileWriter fr = new FileWriter(file, true);
1830                          BufferedWriter br = new BufferedWriter(fr);
1831                          PrintWriter pr = new PrintWriter(br);
1832                          while ((tmpStr = objects.readLine()) != null) {
1833                                tmpStr = tmpStr.replaceAll("SUB-BINDING",
                                              data7a + "-" + data8a + "-" + data6a +
                                "-DIAMOND-REQUEST-2-PRE-ACTIVITY-REQUEST-2-BINDING");
1834                                tmpStr = tmpStr.replaceAll("OBJECT-1", data7a
                                                  + "-"  + data8a + "-" + data6a +
                                                "-DIAMOND-REQUEST-2-PRE-OBJECTREPR");
1835                                tmpStr = tmpStr.replaceAll("OBJECT-2", data7a
                                                    + "-" + data8a + "-" + data6a +
                                                 "-ACTIVITY-REQUEST-2-OBJECTREPR");
1836                                tmpStr =tmpStr.replaceAll("place=\"350,310\"" ,
                                                    "place=\""+630+","+510+"\"");

1837                                pr.println(tmpStr);
1838                           }
1839                           objects.close();
1840                           pr.close();
```

E.21: RequiredActivity.java dosyası (devam).

```
1841                            br.close();
1842                            fr.close();
1843                          } catch (IOException e) { e.printStackTrace();}
1844
1845                          if (!data5a.isEmpty())
1846                          {
1847                            try
1848                            {
1849                              RandomAccessFile objects = null;
1850                              String tmpStr = "";
1851                              objects = new RandomAccessFile(
                                                    "template-required-activity/
                                                    activity-bindingRepr.xml", "r");
1852                              File file = new File("TEST10.xml");
1853                              FileWriter fr = new FileWriter(file, true);
1854                              BufferedWriter br = new BufferedWriter(fr);
1855                              PrintWriter pr = new PrintWriter(br);
1856                              while((tmpStr = objects.readLine()) != null) {
1857                                   tmpStr = tmpStr.replaceAll("SUB-BINDING",
                                              data7a + "-" + data8a + "-" + data6a +
                                    "-ACTIVITY-REQUEST-2-DIAMOND-REQUEST-2-POST-BINDING");
1858                                   tmpStr = tmpStr.replaceAll("OBJECT-1", data7a
                                                  + "-" + data8a + "-" + data6a +
                                                "-ACTIVITY-REQUEST-2-OBJECTREPR");
1859                                   tmpStr = tmpStr.replaceAll("OBJECT-2", data7a
                                                  + "-" + data8a + "-" + data6a +
                                                "-DIAMOND-REQUEST-2-POST-OBJECTREPR");
1860                                   tmpStr =
                                            tmpStr.replaceAll("place=\"350,310\"",
                                                  "place=\""+390+","+510+"\"");
1861                                   pr.println(tmpStr);
1862                               }
1863                              objects.close();
1864                              pr.close();
1865                              br.close();
1866                              fr.close();
1867                             } catch (IOException e) { e.printStackTrace();}
1868                          }
1869                        }
1870                        try
1871                        {
1872                           RandomAccessFile objects = null;
1873                           String tmpStr = "";

1874                           objects =new RandomAccessFile("template-required-activity/
                                                    activity-bindingRepr.xml", "r");
1875                           File file = new File("TEST10.xml");
1876                           FileWriter fr = new FileWriter(file, true);
1877                           BufferedWriter br = new BufferedWriter(fr);
1878                           PrintWriter pr = new PrintWriter(br);
1879                           while ((tmpStr = objects.readLine()) != null) {
1880                                 tmpStr = tmpStr.replaceAll(
                                                      "SUB-BINDING", data7a + "-"
                                                       + data8a + "-" + data6a +
                                      "-START-REQUEST-DIAMOND-REQUEST-1-PRE-BINDING");
1881                                 tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                                            "-" + data8a + "-" +
                                                  data6a + "-START-REQUEST-OBJECTREPR");
1882                                 tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
```

E.21: RequiredActivity.java dosyası (devam).

```
                                                    "-" + data8a + "-" + data6a +
                                                "-DIAMOND-REQUEST-1-PRE-OBJECTREPR");
1883                            tmpStr = tmpStr.replaceAll("place=\"350,310\"",
                                                    "place=\""+250+","+310+"\"");
1884                            tmpStr = tmpStr.replaceAll(
                                "id=\"1\"", "breakpoints=\"150,200 150,310\" id=\"1\"");
1885                            pr.println(tmpStr);
1886                         }
1887                         objects.close();
1888                         pr.close();
1889                         br.close();
1890                         fr.close();
1891                      } catch (IOException e) { e.printStackTrace();}
1892                      try
1893                      {
1894                         RandomAccessFile objects = null;
1895                         String tmpStr = "";
1896                         objects = new RandomAccessFile(
                                                        "template-required-activity/
                                                    activity-bindingRepr.xml", "r");
1897                         File file = new File("TEST10.xml");
1898                         FileWriter fr = new FileWriter(file, true);
1899                         BufferedWriter br = new BufferedWriter(fr);
1900                         PrintWriter pr = new PrintWriter(br);
1901                         while ((tmpStr = objects.readLine()) != null) {
1902                                tmpStr = tmpStr.replaceAll("SUB-BINDING",
                                                    data7a+"-" + data8a + "-" + data6a +
                                        "-START-PROVIDED-DIAMOND-SEND-1-PRE-BINDING");
1903                                tmpStr = tmpStr.replaceAll("OBJECT-1", data7a +
                                                    "-" + data8a + "-" + data6a +
                                                "-START-PROVIDED-OBJECTREPR");
1904                                tmpStr = tmpStr.replaceAll("OBJECT-2", data7a +
                                                    "-" + data8a + "-" + data6a +
                                                "-DIAMOND-SEND-1-PRE-OBJECTREPR");
1905                                tmpStr = tmpStr.replaceAll("place=\"350,310\"",
                                                    "place=\""+850+","+300+"\"");

1906                                tmpStr = tmpStr.replaceAll("id=\"1\"",
                                       "breakpoints=\"830,200 840,250\" id=\"1\"");
1907                                pr.println(tmpStr);
1908                         }
1909                         objects.close();
1910                         pr.close();
1911                         br.close();
1912                         fr.close();
1913                      } catch (IOException e) { e.printStackTrace();}
1914
1915                      try
1916                      {
1917                         RandomAccessFile objects = null;
1918                         String tmpStr = "";
1919                         objects = new RandomAccessFile(
                                                        "template-required-activity/
                                                    activity-diyagram-end.xml",
"r");
1920                         File file = new File("TEST10.xml");
1921                         FileWriter fr = new FileWriter(file, true);
1922                         BufferedWriter br = new BufferedWriter(fr);
1923                         PrintWriter pr = new PrintWriter(br);
```

E.21: RequiredActivity.java dosyası (devam).

```
                            while ((tmpStr = objects.readLine()) != null) {
                                    pr.println(tmpStr);
                            }
                            objects.close();
                            pr.close();
                            br.close();
                            fr.close();
                        } catch (IOException e) { e.printStackTrace();}

                        File myObj = new File("deneme2/Test-Params.xml");
                        if (myObj.exists() && !myObj.isDirectory()) {

                            if (myObj.delete()) {
                                System.out.println("Deleted the file: " +
                                                    myObj.getName());
                            } else {
                                System.out.println("Failed to delete the file.");
                            }
                        }
                    }
                    stmt.close();
                }
                stmt5.close();
                c.close();
            } catch (Exception err) {
                System.err.println(err.getClass().getName() + ": " + err.getMessage());
            }
        }
}
```

E.22: BindingRepActivity.java dosyası

```
import java.util.*;
import java.io.*;
import java.sql.*;
import org.sqlite.SQLiteConfig;
import java.nio.file.*;

public class BindingReprActivity {

  public BindingReprActivity() {
    writingBinding1();
    writingBinding2();
  }

  public static void main(String[] args) {
     new BindingReprActivity();
  }


  public void writingBinding1() {
    Connection c;
    Statement stmt;
    try
    {
      Class.forName("org.sqlite.JDBC");
      SQLiteConfig config = new SQLiteConfig();
```

E.22: BindingRepActivity.java dosyası (devam).

```
24          config.enforceForeignKeys(true);
25          c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                            config.toProperties());
26
27          Statement stmt5 = c.createStatement();
28          ResultSet rs5 = stmt5.executeQuery("SELECT * FROM ReqCompMth  ");
29          while (rs5.next())
30          {
31            String data1b = rs5.getString("methodName");
32            stmt = c.createStatement();
33
34            ResultSet rs1 = stmt.executeQuery("SELECT * FROM ReqPorts WHERE  methodName='" +
                                              data1b + "';");
35
36            while (rs1.next()) {
37                String data2a = rs1.getString("preCon1");
38                String data3a = rs1.getString("postCon1");
39                String data6a = rs1.getString("methodName");
40                String data7a = rs1.getString("compName");
41                String data8a = "";
42
43                int data1c = Integer.parseInt(rs1.getString("methNameCoordinateX"));
44                int data2c = Integer.parseInt(rs1.getString("methNameCoordinateY"));
45
46                int data3c=0;
47                int data4c=0;
48                Statement stmt2 = c.createStatement();
49                ResultSet rs2 = stmt2.executeQuery("SELECT compName,methNameCoordinateX,
                                                    methNameCoordinateY FROM Ports WHERE
                                        portType='Provided' AND methodName='" + data6a + "';");
50                while (rs2.next()) {
51                    data8a = rs2.getString("compName");
52                    data3c = Integer.parseInt(rs2.getString("methNameCoordinateX"));
53                    data4c = Integer.parseInt(rs2.getString("methNameCoordinateY"));
54                }
55                stmt2.close();
56
57                try {
58                    RandomAccessFile objects = null;
59                    String tmpStr = "";
60                    objects = new RandomAccessFile("template-binding-rep/binding-rep.xml",
                                                     "r");
61                    File file = new File("TEST6.xml");
62                    FileWriter fr = new FileWriter(file, true);
63                    BufferedWriter br = new BufferedWriter(fr);
64                    PrintWriter pr = new PrintWriter(br);
65
66                    while ((tmpStr = objects.readLine()) != null) {
67                           tmpStr = tmpStr.replaceAll("BINDING-1",data7a + "-" + data8a
                                                      + "-" + data6a + "BINDING-NAME");
68                           tmpStr = tmpStr.replaceAll("OBJREP-COMPONENT-NAME", data7a +
                                                       "-OBJREPR");
69                           tmpStr = tmpStr.replaceAll("OBJREP-PROVIDED-REQUIRED", data7a
                                                    + "-" + data6a + "REQUIRED-OBJECT-OBJREPR");
70                           tmpStr = tmpStr.replaceAll("place=\"508,110\"",
                                                     "place=\""+data1c+","+data2c+"\"");

71                           pr.println(tmpStr);
72                    }
```

E.22: BindingRepActivity.java dosyası (devam).

```
73                objects.close();
74                pr.close();
75                br.close();
76                fr.close();
77            } catch (IOException e) {
78                e.printStackTrace();
79            }

80            try
81            {
82               RandomAccessFile objects= null;
83               String tmpStr="";
84               objects = new RandomAccessFile("template-sub-graph-id/sub-graph-id.xml",
                                                "r");
85               File file = new File("TEST7.xml");
86               FileWriter fr = new FileWriter(file, true);
87               BufferedWriter br = new BufferedWriter(fr);
88               PrintWriter pr = new PrintWriter(br);
89
90               while ((tmpStr = objects.readLine()) != null)
91               {
92                 tmpStr = tmpStr.replaceAll("SUB-GRAPH-ID",data7a+"-"+ data8a +
                                                 "-"+data6a+"SUB-GRAPH");
93                 pr.println(tmpStr);
94               }
95               objects.close();
96               pr.close();
97               br.close();
98               fr.close();
99             }
100            catch (IOException e) { e.printStackTrace(); }
101         }
102         stmt.close();
103      }
104      stmt5.close();
105      c.close();
106    } catch (Exception err) {
107        System.err.println(err.getClass().getName() + ": " + err.getMessage());
108    }
109 }
110
111 public void writingBinding2()
112 {
113    Connection c;
114    Statement stmt;
115    try
116    {
117      Class.forName("org.sqlite.JDBC");
118      SQLiteConfig config = new SQLiteConfig();
119      config.enforceForeignKeys(true);
120      c = DriverManager.getConnection("jdbc:sqlite:ProMeLaDb.db",
                                         config.toProperties());
121      stmt = c.createStatement();
122      ResultSet rs1 = stmt.executeQuery("SELECT * FROM Ports WHERE portType='Provided'
                                          ");
123      while (rs1.next())
124      {
125         String data1a = rs1.getString("portType");
126         String data2a = rs1.getString("preCon1");
```

E.22: BindingRepActivity.java dosyası (devam).

```
127             String data3a = rs1.getString("postCon1");
128             String data4a = rs1.getString("preCon2");
129             String data5a = rs1.getString("postCon2");
130             String data6a = rs1.getString("methodName");
131             String data7a = rs1.getString("compName");
132             String data8a="";
133
134             int data1c = Integer.parseInt(rs1.getString("methNameCoordinateX"));
135             int data2c = Integer.parseInt(rs1.getString("methNameCoordinateY"));
136             int data3c=0;
137             int data4c=0;
138
139             Statement stmt2 = c.createStatement();
140             ResultSet rs2 = stmt2.executeQuery("SELECT compName,methNameCoordinateX,
                                                    methNameCoordinateY FROM ReqPorts WHERE
                                                    methodName='"+data6a+"';");
141             while (rs2.next())
142             {
143                data8a = rs2.getString("compName");
144                data3c = Integer.parseInt(rs2.getString("methNameCoordinateX"));
145                data4c = Integer.parseInt(rs2.getString("methNameCoordinateY"));
146             }
147             stmt2.close();

148             try
149             {
150               RandomAccessFile objects= null;
151               String tmpStr="";
152               objects = new RandomAccessFile("template-binding-rep/binding-rep.xml",
                                                 "r");
153               File file = new File("TEST6.xml");
154               FileWriter fr = new FileWriter(file, true);
155               BufferedWriter br = new BufferedWriter(fr);
156               PrintWriter pr = new PrintWriter(br);
157
158                while ((tmpStr = objects.readLine()) != null)
159                {
160                     tmpStr = tmpStr.replaceAll("BINDING-1",data7a + "-" + data8a +
                                                    "-" + data6a + "BINDING-NAME");
161                     tmpStr = tmpStr.replaceAll("OBJREP-COMPONENT-NAME", data7a+
                                                    "-OBJREPR");
162                     tmpStr = tmpStr.replaceAll("OBJREP-PROVIDED-REQUIRED", data7a +
                                                   "-" + data6a + "PROVIDER-OBJECT-OBJREPR");
163                     tmpStr = tmpStr.replaceAll("place=\"508,110\"",
                                                    "place=\""+data1c+","+data2c+"\"");
164                     pr.println(tmpStr);
165                }
166                objects.close();
167                pr.close();
168                br.close();
169                fr.close();
170             }
171             catch (IOException e) { e.printStackTrace(); }
172          }
173          stmt.close();
174          c.close();
175       }
176       catch (Exception err)
177       {
```

E.22: BindingRepActivity.java dosyası (devam).

```
178          System.err.println(err.getClass().getName() + ": " + err.getMessage());
179       }
180    }
181 }
182
```

E.23: ReJunctionActivity.java dosyası.

```
1  import java.util.*;
2  import java.io.*;
3  import java.sql.*;
4  import org.sqlite.SQLiteConfig;
5  import java.nio.file.*;
6
7  public class ReJunctionActivity {
8
9    public ReJunctionActivity() {
10     writing1();
11     writing2();
12     writing3();
13     writing4();
14     writingDiyagramHeader();
15     writing5();
16     writing6();
17     writingDiyagramEnd();
18     writing7();
19     finalCheck();
20   }
21   public static void main(String[] args) {
22     new ReJunctionActivity();
23   }
24   public void writing1()
25   {
26     try {
27        RandomAccessFile objects = null;
28        String tmpStr = "";
29       objects = new RandomAccessFile("TEST1.xml", "r");
30       File file = new File("FinalActivity.mxm");
31       FileWriter fr = new FileWriter(file, true);
32       BufferedWriter br = new BufferedWriter(fr);
33       PrintWriter pr = new PrintWriter(br);
34
35       while ((tmpStr = objects.readLine()) != null) {
36              pr.println(tmpStr);
37       }
38       objects.close();
39       pr.close();
40       br.close();
41       fr.close();
42     } catch (IOException e) {
43         e.printStackTrace();
44     }
45   }
46   public void writing2()
47   {
48     try {
```

E.23: ReJunctionActivity.java dosyası (devam).

```
49        RandomAccessFile objects = null;
50        String tmpStr = "";
51        objects = new RandomAccessFile("TEST11.xml", "r");
52        File file = new File("FinalActivity.mxm");
53        FileWriter fr = new FileWriter(file, true);
54        BufferedWriter br = new BufferedWriter(fr);
55        PrintWriter pr = new PrintWriter(br);
56
57        while ((tmpStr = objects.readLine()) != null) {
58               pr.println(tmpStr);
59        }
60        objects.close();
61        pr.close();
62        br.close();
63        fr.close();
64     } catch (IOException e) {
65         e.printStackTrace();
66     }
67   }
68   public void writing3()
69   {
70     try {
71       RandomAccessFile objects = null;
72       String tmpStr = "";
73       objects = new RandomAccessFile("TEST10.xml", "r");
74       File file = new File("FinalActivity.mxm");
75       FileWriter fr = new FileWriter(file, true);
76       BufferedWriter br = new BufferedWriter(fr);
77       PrintWriter pr = new PrintWriter(br);
78       while ((tmpStr = objects.readLine()) != null) {
79              pr.println(tmpStr);
80       }
81       objects.close();
82       pr.close();
83       br.close();
84       fr.close();
85     } catch (IOException e) {
86         e.printStackTrace();
87     }
88   }
89   public void writing4()
90   {
91     try {
92        RandomAccessFile objects = null;
93        String tmpStr = "";
94        objects = new RandomAccessFile("TEST4.xml", "r");
95        File file = new File("FinalActivity.mxm");
96        FileWriter fr = new FileWriter(file, true);
97        BufferedWriter br = new BufferedWriter(fr);
98        PrintWriter pr = new PrintWriter(br);
99
100        while ((tmpStr = objects.readLine()) != null) {
101               pr.println(tmpStr);
102        }
103        objects.close();
104        pr.close();
105        br.close();
106        fr.close();
107     } catch (IOException e) {
```

E.23: ReJunctionActivity.java dosyası (devam).

```
108         e.printStackTrace();
109       }
110   }
111   public void writingDiyagramHeader()
112   {
113     try {
114       RandomAccessFile objects = null;
115       String tmpStr = "";
116       objects = new RandomAccessFile("template-diyagram/diyagram-header.xml", "r");
117       File file = new File("FinalActivity.mxm");
118       FileWriter fr = new FileWriter(file, true);
119       BufferedWriter br = new BufferedWriter(fr);
120       PrintWriter pr = new PrintWriter(br);
121
122       while ((tmpStr = objects.readLine()) != null) {
123              pr.println(tmpStr);
124
125       }
126       objects.close();
127       pr.close();
128       br.close();
129       fr.close();
130     } catch (IOException e) {
131        e.printStackTrace();
132     }
133   }
134   public void writing5()
135   {
136     try {
137        RandomAccessFile objects = null;
138        String tmpStr = "";
139        objects = new RandomAccessFile("TEST5.xml", "r");
140        File file = new File("FinalActivity.mxm");
141        FileWriter fr = new FileWriter(file, true);
142        BufferedWriter br = new BufferedWriter(fr);
143        PrintWriter pr = new PrintWriter(br);
144
145        while ((tmpStr = objects.readLine()) != null) {
146               pr.println(tmpStr);
147        }
148        objects.close();
149        pr.close();
150        br.close();
151        fr.close();
152     } catch (IOException e) {
153          e.printStackTrace();
154     }
155   }
156   public void writing6()
157   {
158     try {
159       RandomAccessFile objects = null;
160       String tmpStr = "";
161       objects = new RandomAccessFile("TEST6.xml", "r");
162       File file = new File("FinalActivity.mxm");
163       FileWriter fr = new FileWriter(file, true);
164       BufferedWriter br = new BufferedWriter(fr);
165       PrintWriter pr = new PrintWriter(br);
166
```

E.23: ReJunctionActivity.java dosyası (devam).

```
167       while ((tmpStr = objects.readLine()) != null) {
168             pr.println(tmpStr);
169       }
170       objects.close();
171       pr.close();
172       br.close();
173       fr.close();
174     } catch (IOException e) {
175        e.printStackTrace();
176     }
177   }
178   public void writingDiyagramEnd()
179   {
180      try {
181        RandomAccessFile objects = null;
182        String tmpStr = "";
183        objects = new RandomAccessFile("template-diyagram/diyagram-end.xml", "r");
184        File file = new File("FinalActivity.mxm");
185        FileWriter fr = new FileWriter(file, true);
186        BufferedWriter br = new BufferedWriter(fr);
187        PrintWriter pr = new PrintWriter(br);
188
189        while ((tmpStr = objects.readLine()) != null) {
190               pr.println(tmpStr);
191        }
192        objects.close();
193        pr.close();
194        br.close();
195        fr.close();
196      } catch (IOException e) {
197          e.printStackTrace();
198      }
199   }
200   public void writing7()
201   {
202     try {
203       RandomAccessFile objects = null;
204       String tmpStr = "";
205       objects = new RandomAccessFile("TEST7.xml", "r");
206       File file = new File("FinalActivity.mxm");
207       FileWriter fr = new FileWriter(file, true);
208       BufferedWriter br = new BufferedWriter(fr);
209       PrintWriter pr = new PrintWriter(br);
210
211       while ((tmpStr = objects.readLine()) != null) {
212              pr.println(tmpStr);
213       }
214       objects.close();
215       pr.println("</gxl>");
216       pr.close();
217       br.close();
218       fr.close();
219     } catch (IOException e) {
220          e.printStackTrace();
221     }
222
223   }
224   public void finalCheck ()
225   {
```

E.23: ReJunctionActivity.java dosyası (devam).

```
226     try {
227       RandomAccessFile objects = null;
228       String tmpStr = "";
229       objects = new RandomAccessFile("FinalActivity.mxm", "rw");
230       File file = new File("FinalActivity2.mxm");
231       FileWriter fr = new FileWriter(file, true);
232       BufferedWriter br = new BufferedWriter(fr);
233       PrintWriter pr = new PrintWriter(br);
234 
235       while ((tmpStr = objects.readLine()) != null) {
236              tmpStr = tmpStr.replaceAll("&", "&amp;");
237              pr.println(tmpStr);
238       }
239       objects.close();
240       pr.close();
241       br.close();
242        fr.close();
243 
244        File myObj = new File("FinalActivity.mxm");
245        if(myObj.exists() && !myObj.isDirectory())
246        {
247           if (myObj.delete())
248           {
249              System.out.println("Deleted the file: " + myObj.getName());
250           }
251           else {
252              System.out.println("Failed to delete the file.");
253           }
254         }
255      } catch (IOException e) {
256           e.printStackTrace();
257       }
258   }
259 }
260 
```

Şablon dosyalarının olduğu klasörler aşağıda gösterilmiştir.

template-binding klasörünün içindeki dosyalar:

E.24: main-binding.xml dosyası.

```
1   <binding id="BINDING-NAME" nextConnectionId="3" oid="BINDING-NAME">
2     <relationship oid="_4_20362" type="Relationship_Interface_sysadmin_3336642846"
       typeName="Interface"></relationship>
3     <connection id="1">
4       <role oid="_4_20364" type="Role_Component_sysadmin_3336643073"
         typeName="Component"></role>
5       <object href="#COMPONENT-NAME"></object>
6    </connection>
7    <connecon id="2">
8      <role oid="_4_20366" type="Role_Provided_interface_sysadmin_3336642684"
         typeName="Provided interfae">
9            <slot id="ad4ecx" name="Name">
10              <property oid="_4_20370" type="Property_Name_user_3012480935"
                    typeName="Port Nme"></property>
11                <value>
12                  <string>PORT-NAME</string>
13                </value>
14            </slot>
15     </role>
16     <object href="#PROVIDER-REQUIRED-OBJECT"></object>
17   </connection>
18  </binding>
```

E.25: main-binding-req.xml dosyası.

```
1   <binding id="BINDING-NAME" nextConnectionId="3" oid="BINDING-NAME">
2      <relationship oid="_4_20374" type="Relationship_Interface_sysadmin_3336642846"
           typeName="Interface"></relationship>
3      <connection id="1">
4         <role oid="_4_20376" type="Role_Component_sysadmin_3336643073"
            typeName="Component"></role>
5         <object href="#COMPONENT-NAME"></object>
6      </connection>
7      <connection id="2">
8          <role oid="_4_20378" type="Role_Required_interface_sysadmin_3336642118"
             typeName="Required interface">
9             <slot id="ac4edq" name="Name">
10               <property oid="_4_20382" type="Property_Name_user_3012480935"
                   typeName="Port Name"></property>
11                <value>
12                       <string>PORT-NAME</string>
13                   </value>
14                </slot>
15                <slot id="ad80fm" name="buffer size">
16                    <property oid="_4_21612" type="Property_channel_sysadmin_3693662489"
                        typeName="channel"></property>
17                     <value>
18                       <int>1</int>
19                    </value>
20                </slot>
21            </role>
22            <object href="#REQUIRED-OBJECT"></object>
23     </connection>
24   </binding>
```

template-binding-rep klasörünün içindeki dosyalar:

E.26: binding-rep.xml dosyası.

```
1       <bindingRepr isStraight="true" oid="_4_20394" place="508,110" scale="1,1">
2               <binding href="#BINDING-1"
                        xmlns="http://www.metacase.com/gxlGOPRR"></binding>
3               <connectionRepr id="1">
4                       <objectRepr href="#OBJREP-COMPONENT-NAME"></objectRepr>
5               </connectionRepr>
6               <connectionRepr id="2">
7                       <objectRepr href="#OBJREP-PROVIDED-REQUIRED"></objectRepr>
8               </connectionRepr>
9       </bindingRepr>
```

template-component klasörünün içindeki dosyalar:

E.27: Component.xml dosyası.

```
1 <object id="COMPONENT-NAME-1" oid="COMPONENT-NAME-1"
        type="Object_Component_sysadmin_3084531214" typeName="Component">
2   <slot id="a87qh2" name="Type">
3      <property oid="_4_19711" type="Property_Type_sysadmin_3645786979"
          typeName="Type"></property>
4      <value>
5          <string>COMPONENT-NAME</string>
6      </value>
7   </slot>
8   <slot id="ab6e4g" name="parameters">
9      <property oid="_4_19713" type="Property_parameters_sysadmin_3649954688"
          typeName="parameters"></property>
10     <value>
11        <seq>
```

E.28: Component2.xml dosyası.

```
1          </seq>
2       </value>
3     </slot>
4     <slot id="a9e1ck" name="Data List">
5        <property oid="_4_19715" type="Property_Arguments_sysadmin_3647613610"
            typeName="Arguments-LTL"></property>
6        <value>
7          <seq>
```

E.29: ComponentParamList.xml dosyası.

```
<object oid="_4_492997" type="Object_Parameters_2_sysadmin_3649954478"
    typeName="Parameters">
    <slot id="ca6dpq" name="Parameter name">
        <property oid="_4_492999"
            type="Property_Parameter_value_sysadmin_3083940744"
            typeName="Parameter name"></property>
        <value>
            <string>PARAM-NAME</string>
        </value>
    </slot>
    <slot id="cd6ed3" name="Data type">
        <property oid="_4_493001" type="Property_Data_type_user_3012458459"
            typeName="Data type"></property>
        <value>
            <string>PARAM-TYPE</string>
        </value>
    </slot>
    <slot id="cc6dsk" name="Value">
        <property oid="_4_493003" type="Property_Value_sysadmin_3649954474"
            typeName="Value"></property>
        <value>
            <string>PARAM-VALUE</string>
        </value>
    </slot>
</object>
```

E.30: ComponentDataList.xml dosyası.

```
<object oid="_4_19689" type="Object_Argument_sysadmin_3088675622"
    typeName="Argument">
    <slot id="a36m3u" name="Data type">
        <property oid="_4_19691" type="Property_Data_type_user_3012458459"
            typeName="Data type"></property>
        <value>
            <string>DATA-TYPE</string>
        </value>
    </slot>
    <slot id="a46m4d" name="Name">
        <property oid="_4_19693" type="Property_Name_user_3012480935"
            typeName="Port Name"></property>
        <value>
            <string>DATA-NAME</string>
        </value>
    </slot>
    <slot id="a56m51" name="Default value">
        <property oid="_4_19695"
            type="Property_Default_value_sysadmin_3251445841"
            typeName="pre Condition"></property>
        <value>
            <string>DEFAULT-VALUE</string>
        </value>
    </slot>
```

E.30: ComponentDataList.xml dosyası (devam).

```
20              <slot id="a7gf8i" name="Array size">
21                      <property oid="_4_19697"
                            type="Property_Array_size_sysadmin_3650188689"
                            typeName="Array size"></property>
22                      <value>
23                         <string></string>
24                      </value>
25              </slot>
26      </object>
```

E.31: ComponentEnd.xml dosyası.

```
1           </seq>
2         </value>
3      </slot>
4   </object>
```

template-diyagram klasörünün içindeki dosyalar:

E.32: diyagram-header.xml dosyası.

```
1<diyagram displayGrid="false" grid="10,10" oid="_4_20310"
    timeStamp="2018-01-04T07:50:55.500Z"
    useGrid="true" xmlns="http://www.metacase.com/gxlGOPRRRepr">
2  <graph href="#_4_20311" xmlns="http://www.metacase.com/gxlGOPRR"></graph>
```

E.33: diyagram-end.xml dosyası.

```
1    </diyagram>
2 </graph>
```

template-header klasörünün içindeki dosyalar:

E.34: Header.xml dosyası.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <gxl dbName="GENEL-TEZ-test" dbPath="GENEL-TEZ-test" timeStamp="2021-05-03T11:36:40.867Z"
      version="5.5" xmlns="http://www.metacase.com/gxlGOPRR">
3       <graph id="_4_20311" oid="_4_20311" type="Graph_UML_Sequence_sysadmin_3658915051"
          typeName="Component_Diyagram">
4           <slot id="aculy2" name="Component_Diyagram">
5               <property oid="_4_20316" type="Property_UML_sequence_sysadmin_3658917709"
                  typeName="Component_Diyagram"></property>
6               <value>
```

E.34: Header.xml dosyası (devam).

```
                <string> Reverse SAwML</string>
            </value>
        </slot>
```

template-object-rep klasörünün içindeki dosyalar:

E.35: object-rep.xml dosyası.

```
    <objectRepr id="OBJREPR" oid="OBJREPR" place="240,110" scale="0.77,0.5975">
            <object href="#COMPONENT-NAME"
                 xmlns="http://www.metacase.com/gxlGOPRR"></object>
    </objectRepr>
```

E.36: pro-req-object.xml dosyası.

```
<objectRepr id="OBJREPR" oid="OBJREPR" place="240,110" scale="0.77,0.5975">
  <object href="#OBJ-PROVIDED-REQUIRED"
     xmlns="http://www.metacase.com/gxlGOPRR"></object>
</objectRepr>
```

template-provided klasörünün içindeki dosyalar:

E.37: provided-entrance.xml dosyası.

```
<object id="PROVIDER-OBJECT-1" oid="PROVIDER-OBJECT-1"
    type="Object_Provided_Connector_sysadmin_3658915832" typeName="Provided_Connector">
 <graph id="SUB-GRAPH-1" oid="SUB-GRAPH-1"
      type="Graph_Sequence_provided_sysadmin_3663484838"
      typeName="Sequence_provided">
  <slot id="a0ae11" name="Sequence">
    <property oid="_4_20006" type="Property_Sequence_sysadmin_3658917647"
       typeName="Sequence"></property>
    <value>
     <string></string>
    </value>
  </slot>
    <object id="LIFELINE-1" oid="LIFELINE-1" type="Object_LifeLine_sysadmin_3658927263"
      typeName="LifeLine">
     <slot id="a0v30b" name="Class name">
     <property oid="_4_20016" type="Property_Class_name_user_3012480910"
         typeName="Class name"></property>
     <value>
         <string>COMPONENT-NAME-PROVIDED</string>
     </value>
    </slot>
```

E.37: provided-entrance.xml dosyası (devam).

```
16    </object>
17    <object id="LIFELINE-2" oid="LIFELINE-2" type="Object_LifeLine_sysadmin_3658927263"
         typeName="LifeLine">
18     <slot id="a0v30b" name="Class name">
19      <property oid="_4_20011" type="Property_Class_name_user_3012480910"
           typeName="Class name"></property>
20      <value>
21         <string>COMPONENT-NAME-REQUIRED</string>
22      </value>
23     </slot>
24    </object>
```

E.38: provided-parameters.xml dosyası.

```
1  <object oid="_4_21526" type="Object_Parameter_sysadmin_3083934003" typeName="Parameter">
2    <slot id="a273tk" name="Parameter">
3       <property oid="_4_21528" type="Property_Parameter_value_sysadmin_3083940744"
             typeName="Parameter name"></property>
4       <value>
5         <string>PARAMETER-NAME</string>
6       </value>
7    </slot>
8    <slot id="a06te3" name="Data type">
9      <property oid="_4_21530" type="Property_Data_type_user_3012458459"
          typeName="Data type"></property>
10     <value>
11       <string>PARAMETER-TYPE</string>
12     </value>
13   </slot>
14 </object>
```

E.39: provided-binding-return-method-name.xml dosyası.

```
1  <binding id="SUB-BINDING-1" nextConnectionId="3" oid="SUB-BINDING-1">
2    <relationship oid="_4_20052" type="Relationship_Async_Call_sysadmin_3663487331"
        typeName="Result">
3       <slot id="af11fm" name="Service Name-2">
4          <property oid="_4_20054" type="Property_Service_Name_2_sysadmin_3669973713"
                      typeName="Service Name-2"></property>
5          <value>
6            <seq>
7              <object oid="_4_20046" type="Object_Operation_sysadmin_3019391465"
                         typeName="Methods/Events">
8                    <slot id="p1" name="Method/Event">
9                       <property oid="_4_20048"
                           type="Property_Method_name_user_3012459728"
                           typeName="Service Name"></property>
10                      <value>
11                         <string>METHOD-NAME</string>
12                      </value>
13                   </slot>
14                   <slot id="a0subs" name="Parameters">
15                      <property oid="_4_20050"
```

E.39: provided-binding-return-method-name.xml dosyası (devam).

```
                                type="Property_Parameters_sysadmin_3083934035"
                                typeName="Parameters"></property>
16                          <value>
17                            <seq>
```

E.40: provided-binding-return-end.xml dosyası.

```
1                         </seq>
2                       </value>
3                     </slot>
4                   </object>
5                 </seq>
6               </value>
7             </slot>
8             <slot id="aeahsn" name="post Condition">
9               <property oid="_4_20056" type="Property_Duration_user_3247744455" typeName="post
                    Condition"></property>
10              <value>
11                <string>RETURN-VALUE</string>
12              </value>
13            </slot>
14          </relationship>
15          <connection id="1">
16            <role oid="_4_20058" type="Role_Message_result_sysadmin_3663488592"
                    typeName="Message result"></role>
17            <object href="#LIFELINE-1"></object>
18          </connection>
19          <connection id="2">
20            <role oid="_4_20060" type="Role_Asyns_Message_sysadmin_3663486613"
                    typeName="result_Message"></role>
21            <object href="#LIFELINE-2"></object>
22          </connection>
23        </binding>
```

E.41: provided-binding-forward.xml dosyası.

```
1   <binding id="SUB-BINDING-2" nextConnectionId="3" oid="SUB-BINDING-2">
2     <relationship oid="_4_20031" type="Relationship_Call_sysadmin_3222914223"
          typeName="Sync_Call">
3       <slot id="a4joi" name="Service Name-2">
4         <property oid="_4_20033" type="Property_Service_Name_2_sysadmin_3669973713"
              typeName="Service Name-2"></property>
5         <value>
6           <seq>
7             <object oid="_4_20025" type="Object_Operation_sysadmin_3019391465"
                  typeName="Methods/Events">
8               <slot id="p1" name="Method/Event">
9                 <property oid="_4_20027"
                    type="Property_Method_name_user_3012459728"
                    typeName="Service Name"></property>
10                <value>
11                  <string>METHOD-NAME</string>
12                </value>
```

E.41: provided-binding-forward.xml dosyası (devam).

```
13                    </slot>
14                    <slot id="a0subs" name="Parameters">
15                      <property oid="_4_20029"
                           type="Property_Parameters_sysadmin_3083934035"
                           typeName="Parameters"></property>
16                      <value>
17                        <seq>
```

E.42: provided-binding-forward-end.xml dosyası.

```
1                         </seq>
2                       </value>
3                     </slot>
4                   </object>
5                 </seq>
6               </value>
7             </slot>
8             <slot id="a28wi8" name="pre Condition">
9               <property oid="_4_20035" type="Property_Default_value_sysadmin_3251445841"
                    typeName="pre Condition"></property>
10              <value>
11                <string>PRE-CONDITION</string>
12              </value>
13            </slot>
14            <slot id="a38wj8" name="post Condition">
15              <property oid="_4_20037" type="Property_Duration_user_3247744455"
                     typeName="post Condition"></property>
16              <value>
17                <string>POST-CONDITION</string>
18              </value>
19            </slot>
20          </relationship>
21          <connection id="1">
22            <role oid="_4_20039" type="Role_Message_from_sysadmin_3222914365"
                    typeName="Message from"></role>
23            <object href="#LIFELINE-2"></object>
24          </connection>
25          <connection id="2">
26            <role oid="_4_20041" type="Role_Test1_sysadmin_3663430484"
                     typeName="sync_Message"></role>
27            <object href="#LIFELINE-1"></object>
28          </connection>
29      </binding>
```

E.43: provided-diyagram-entrance.xml dosyası.

```
1  <diyagram displayGrid="false" grid="10,10" oid="_4_20008" timeStamp="2017-12-
       31T02:39:38.282Z" useGrid="true" xmlns="http://www.metacase.com/gxlGOPRRRepr">
2    <graph href="#SUB-GRAPH-1" xmlns="http://www.metacase.com/gxlGOPRR"></graph>
3    <objectRepr id="objcetRep-2" oid="objcetRep-2" place="150,131" scale="1.16667,2.14">
4          <object href="#LIFELINE-2" xmlns="http://www.metacase.com/gxlGOPRR"></object>
5      </objectRepr>
6      <objectRepr id="objcetRep-1" oid="objcetRep-1" place="530,130" scale="1.33333,2.15">
7        <object href="#LIFELINE-1" xmlns="http://www.metacase.com/gxlGOPRR"></object>
8      </objectRepr>
```

E.44: provided-bindingRepr-forward.xml dosyası.

```
1       <bindingRepr isStraight="true" oid="_4_20045" place="333,130" scale="2.38,2.06667">
2           <binding href="#SUB-BINDING-2"
                 xmlns="http://www.metacase.com/gxlGOPRR"></binding>
3           <connectionRepr id="1" objTargetOffset="0,-1">
4              <objectRepr href="#LIFELINE-2-OBJECTREPR"></objectRepr>
5           </connectionRepr>
6           <connectionRepr id="2" objTargetOffset="-20,0">
7              <objectRepr href="#LIFELINE-1-OBJECTREPR"></objectRepr>
8           </connectionRepr>
9       </bindingRepr>
```

E.45: provided-bindingRepr-return.xml dosyası.

```
1       <bindingRepr isStraight="false" oid="_4_20064" place="333,180"
           scale="0.986667,3.13333">
2              <binding href="#SUB-BINDING-1"
                   xmlns="http://www.metacase.com/gxlGOPRR"></binding>
3              <connectionRepr id="1" objTargetOffset="-13,50">
4                 <objectRepr href="#LIFELINE-1-OBJECTREPR"></objectRepr>
5              </connectionRepr>
6              <connectionRepr id="2" objTargetOffset="0,49">
7                     <objectRepr href="#LIFELINE-2-OBJECTREPR"></objectRepr>
8              </connectionRepr>
9       </bindingRepr>
```

E.46: provided-diyagram-end.xml dosyası.

```
1          </diyagram>
2       </graph>
3   </object>
```

E.47: provided-lifeline.xml dosyası.

```
1 <object id="LIFELINE-1" oid="LIFELINE-1" type="Object_LifeLine_sysadmin_3658927263"
          typeName="LifeLine">
2    <slot id="a0v30b" name="Class name">
3       <property oid="_4_20016" type="Property_Class_name_user_3012480910"
           typeName="Class name"></property>
4       <value>
5          <string>COMPONENT-NAME-PROVIDED</string>
6       </value>
7    </slot>
8 </object>
9 <object id="LIFELINE-2" oid="LIFELINE-2" type="Object_LifeLine_sysadmin_3658927263"
          typeName="LifeLine">
10    <slot id="a0v30b" name="Class name">
11      <property oid="_4_20011" type="Property_Class_name_user_3012480910"
           typeName="Class name"></property>
12      <value>
13         <string>COMPONENT-NAME-REQUIRED</string>
14      </value>
15    </slot>
16 </object>
```

template-provided-activity klasörünün içindeki dosyalar:

E.48: activity-provided-entrance.xml dosyası.

```
<object id="PROVIDER-OBJECT-1" oid="PROVIDER-OBJECT-1"
    type="Object_Provided_Connector_sysadmin_3658915832"
    typeName="Provided_Connector"></object>
```

template-required klasörünün içindeki dosyalar:

E.49: required-entrance.xml dosyası.

```
<object id="REQUIRED-OBJECT-1" oid="REQUIRED-OBJECT-1"
    type="Object_Required_Connector_sysadmin_3658915920" typeName="Required_Connector">
  <graph id="SUB-GRAPH-2" oid="SUB-GRAPH-2"
type="Graph_Sequence_required_sysadmin_3665137046"
      typeName="Sequence_required">
    <slot id="a097pp" name="Sequence">
      <property oid="_4_19855" type="Property_Sequence_sysadmin_3658917647"
        typeName="Sequence"></property>
      <value>
        <string></string>
      </value>
    </slot>
    <object id="LIFELINE-3" oid="LIFELINE-3" type="Object_LifeLine_sysadmin_3658927263"
      typeName="LifeLine">
      <slot id="a0v30b" name="Class name">
        <property oid="_4_19865" type="Property_Class_name_user_3012480910"
          typeName="Class name"></property>
        <value>
          <string>COMPONENT-NAME-PROVIDED</string>
        </value>
      </slot>
    </object>
    <object id="LIFELINE-4" oid="LIFELINE-4" type="Object_LifeLine_sysadmin_3658927263"
      typeName="LifeLine">
      <slot id="a0v30b" name="Class name">
        <property oid="_4_19860" type="Property_Class_name_user_3012480910"
          typeName="Class name"></property>
        <value>
          <string>COMPONENT-NAME-REQUIRED</string>
        </value>
      </slot>
    </object>
```

E.50: required-parameters.xml dosyası.

```
<object oid="_4_19895" type="Object_Parameter_sysadmin_3083934003" typeName="Parameter">
  <slot id="a273tk" name="Parameter">
    <property oid="_4_19897" type="Property_Parameter_value_sysadmin_3083940744"
      typeName="Parameter name"></property>
    <value>
      <string>PARAMETER-NAME</string>
    </value>
```

E.50: required-parameters.xml dosyası (devam).

```
7     </slot>
8     <slot id="a06te3" name="Data type">
9       <property oid="_4_19899" type="Property_Data_type_user_3012458459"
            typeName="Data type"></property>
10      <value>
11        <string>PARAMETER-TYPE</string>
12      </value>
13    </slot>
14  </object>
```

E.51: required-binding-return-method-name.xml dosyası.

```
1   <binding id="SUB-BINDING-3" nextConnectionId="3" oid="SUB-BINDING-3">
2     <relationship oid="_4_19907" type="Relationship_Result_required_sysadmin_3665138721"
          typeName="Result_required">
3       <slot id="a311wx" name="Service Name-2">
4         <property oid="_4_19909" type="Property_Service_Name_2_sysadmin_3669973713"
              typeName="Service Name-2"></property>
5         <value>
6           <seq>
7             <object oid="_4_19901" type="Object_Operation_sysadmin_3019391465"
                typeName="Methods/Events">
8               <slot id="p1" name="Method/Event">
9                 <property oid="_4_19903"
                    type="Property_Method_name_user_3012459728"
                    typeName="Service Name"></property>
10                <value>
11                  <string>METHOD-NAME</string>
12                </value>
13              </slot>
14              <slot id="a0subs" name="Parameters">
15                <property oid="_4_19905"
type="Property_Parameters_sysadmin_3083934035"
                    typeName="Parameters"></property>
16                <value>
17                  <seq>
```

E.52: required-binding-return-end.xml dosyası.

```
1                   </seq>
2                 </value>
3               </slot>
4             </object>
5           </seq>
6         </value>
7       </slot>
8       <slot id="a19abi" name="pre Condition">
9         <property oid="_4_19911" type="Property_Default_value_sysadmin_3251445841"
              typeName="pre Condition"></property>
10        <value>
11          <string>PRE-CONDITION</string>
12        </value>
13      </slot>
14      <slot id="a29ac8" name="post Condition">
15        <property oid="_4_19913" type="Property_Duration_user_3247744455"
            typeName="post Condition"></property>
```

E.52: required-binding-return-end.xml dosyası (devam).

```
16      <value>
17         <string>POST-CONDITION</string>
18      </value>
19    </slot>
20  </relationship>
21  <connection id="1">
22      <role oid="_4_19915" type="Role_Message_result_sysadmin_3663488592"
           typeName="Message result"></role>
23      <object href="#LIFELINE-3"></object>
24    </connection>
25    <connection id="2">
26       <role oid="_4_19917" type="Role_Asyns_Message_sysadmin_3663486613"
            typeName="result_Message"></role>
27       <object href="#LIFELINE-4"></object>
28    </connection>
29  </binding>
```

E.53: required-binding-forward.xml dosyası.

```
1   <binding id="SUB-BINDING-4" nextConnectionId="3" oid="SUB-BINDING-4">
2    <relationship oid="_4_19880" type="Relationship_Sync_Call_required_sysadmin_3665137222"
        typeName="Sync_Call_required">
3     <slot id="a311ta" name="Service Name-2">
4       <property oid="_4_19882" type="Property_Service_Name_2_sysadmin_3669973713"
           typeName="Service Name-2"></property>
5       <value>
6         <seq>
7           <object oid="_4_19874" type="Object_Operation_sysadmin_3019391465"
               typeName="Methods/Events">
8             <slot id="p1" name="Method/Event">
9               <property oid="_4_19876" type="Property_Method_name_user_3012459728"
                   typeName="Service Name"></property>
10              <value>
11                 <string>METHOD-NAME</string>
12              </value>
13            </slot>
14            <slot id="a0subs" name="Parameters">
15              <property oid="_4_19878" type="Property_Parameters_sysadmin_3083934035"
                   typeName="Parameters"></property>
16              <value>
17                <seq>
```

E.54: required-diyagram-entrance.xml dosyası.

```
1 <diyagram displayGrid="false" grid="10,10" oid="_4_19857"
      timeStamp="2017-12-31T02:15:33.366Z"
      useGrid="false" xmlns="http://www.metacase.com/gxlGOPRRRepr">
2  <graph href="#SUB-GRAPH-2" xmlns="http://www.metacase.com/gxlGOPRR"></graph>
3  <objectRepr id="objcetRep-4" oid="objcetRep-4" place="61,140" scale="0.966666,2.035">
4    <object href="#LIFELINE-4" xmlns="http://www.metacase.com/gxlGOPRR"></object>
5  </objectRepr>
6  <objectRepr id="objcetRep-3" oid="objcetRep-3" place="540,140" scale="1.01667,2.015">
7   <object href="#LIFELINE-3" xmlns="http://www.metacase.com/gxlGOPRR"></object>
8  </objectRepr>
```

E.55: required-bindingRepr-forward.xml dosyası.

```
<bindingRepr isStraight="false" oid="_4_19894" place="270,162" scale="2.84,1">
   <binding href="#SUB-BINDING-4" xmlns="http://www.metacase.com/gxlGOPRR"></binding>
   <connectionRepr id="1" objTargetOffset="0,20">
     <objectRepr href="#LIFELINE-4-OBJECTREPR"></objectRepr>
   </connectionRepr>
   <connectionRepr id="2" objTargetOffset="-10,20">
     <objectRepr href="#LIFELINE-3-OBJECTREPR"></objectRepr>
   </connectionRepr>
</bindingRepr>
```

E.56: required-binding-forward-end.xml dosyası.

```
            </seq>
          </value>
        </slot>
      </object>
     </seq>
   </value>
  </slot>
  <slot id="a1980j" name="pre Condition">
   <property oid="_4_19884" type="Property_Default_value_sysadmin_3251445841"
     typeName="pre Condition"></property>
   <value>
     <string>PRE-CONDITION</string>
   </value>
  </slot>
  <slot id="a2981y" name="promise">
   <property oid="_4_19886" type="Property_promise_sysadmin_3665137283"
      typeName="promise"></property>
    <value>
      <string>PROMISE</string>
    </value>
  </slot>
</relationship>
<connection id="1">
   <role oid="_4_19888" type="Role_Message_from_sysadmin_3222914365"
      typeName="Message from"></role>
   <object href="#LIFELINE-4"></object>
 </connection>
 <connection id="2">
   <role oid="_4_19890" type="Role_Test1_sysadmin_3663430484"
      typeName="sync_Message"></role>
   <object href="#LIFELINE-3"></object>
 </connection>
</binding>
```

E.57: required-bindingRepr-return.xml dosyası.

```
<bindingRepr isStraight="true" oid="_4_19921" place="295,220" scale="2.12,1.7">
   <binding href="#SUB-BINDING-3" xmlns="http://www.metacase.com/gxlGOPRR"></binding>
   <connectionRepr id="1" objTargetOffset="-10,80">
     <objectRepr href="#LIFELINE-3-OBJECTREPR"></objectRepr>
   </connectionRepr>
   <connectionRepr id="2" objTargetOffset="30,80">
      <objectRepr href="#LIFELINE-4-OBJECTREPR"></objectRepr>
   </connectionRepr>
</bindingRepr>
```

E.58: required-diyagram-end.xml dosyası.

```
        </diyagram>
      </graph>
  </object>
```

template-required-activity klasörünün içindeki dosyalar:

E.59: activity-required-entrance.xml dosyası.

```
<object id="REQUIRED-OBJECT-1" oid="REQUIRED-OBJECT-1"
        type="Object_Required_Connector_sysadmin_3658915920"
        typeName="Required_Connector">
    <graph id="SUB-GRAPH" oid="SUB-GRAPH"
           type="Graph_Activity_Diyagram_sysadmin_3769067230"
           typeName="Activity-Diyagram">
      <slot id="a0tir" name="ActivityDiyagram">
           <property oid="_4_250573"
              type="Property_Activity_Diyagram_sysadmin_3769077433"
              typeName="Activity_Diyagram"></property>
           <value>
             <string></string>
           </value>
      </slot>
```

E.60: activity-parameters.xml dosyası.

```
<object oid="_4_253567" type="Object_Parameter_sysadmin_3083934003" typeName="Parameter">
  <slot id="a273tk" name="Parameter">
    <property oid="_4_253569" type="Property_Parameter_value_sysadmin_3083940744"
      typeName="Parameter name"></property>
     <value>
        <string>PARAMETER-NAME</string>
    </value>
  </slot>
  <slot id="a06te3" name="Data type">
    <property oid="_4_253571" type="Property_Data_type_user_3012458459"
      typeName="Data type"></property>
    <value>
        <string>PARAMETER-TYPE</string>
    </value>
  </slot>
</object>
```

E.61: activity-request1-object-entrance.xml dosyası.

```
<object id="ACTIVITY-REQUEST-1" oid="ACTIVITY-REQUEST-1"
      type="Object_Activity_sysadmin_3769068109" typeName="ActivityRequest1">
  <slot id="a6bem8" name="Activity_streotype">
```

E.61: activity-request1-object-entrance.xml dosyası (devam).

```
3       <property oid="_4_256090"
type="Property_Name_Streotype_Single_1_sysadmin_3771843749"
          typeName="Name-Streotype-Single-1"></property>
4       <value>
5          <string>send-request</string>
6       </value>
7     </slot>
8     <slot id="a0481d" name="promise">
9       <property oid="_4_251395" type="Property_promise_sysadmin_3665137283"
          typeName="promise"></property>
10      <value>
11         <string>PROMISE</string>
12      </value>
13    </slot>
14    <slot id="a5fom6" name="Service Name">
15      <property oid="_4_253421" type="Property_Service_Name_2_sysadmin_3669973713"
           typeName="Service Name-2"></property>
16      <value>
17        <seq>
18         <object oid="_4_253434" type="Object_Operation_sysadmin_3019391465"
                typeName="Methods/Events">
19            <slot id="p1" name="Method/Event">
20              <property oid="_4_253436" type="Property_Method_name_user_3012459728"
                  typeName="Service Name"></property>
21              <value>
22                <string>METHOD-NAME</string>
23              </value>
24            </slot>
25            <slot id="a0subs" name="Parameters">
26             <property oid="_4_253438" type="Property_Parameters_sysadmin_3083934035"
                  typeName="Parameters"></property>
27              <value>
28                <seq>
```

E.62: activity-request2-object-entrance.xml dosyası.

```
1 <object id="ACTIVITY-REQUEST-2" oid="ACTIVITY-REQUEST-2"
       type="Object_ActivityRequest2_sysadmin_3771842944" typeName="ActivityRequest2">
2   <slot id="cqbfaj" name="Activity_streotype">
3       <property oid="_4_256332"
type="Property_Name_Streotype_Single_2_sysadmin_3771844245"
           typeName="Name-Streotype-Single-2"></property>
4       <value>
5          <string>receive-response</string>
6       </value>
7   </slot>
8   <slot id="cpbd9l" name="Service Name">
9       <property oid="_4_256334" type="Property_Service_Name_2_sysadmin_3669973713"
            typeName="Service Name-2"></property>
10      <value>
11        <seq>
12          <object oid="_4_256318" type="Object_Operation_sysadmin_3019391465"
              typeName="Methods/Events">
13           <slot id="p1" name="Method/Event">
14              <property oid="_4_256320" type="Property_Method_name_user_3012459728"
                    typeName="Service Name"></property>
15              <value>
16                 <string>METHOD-NAME</string>
```

E.62: activity-request2-object-entrance.xml dosyası (devam).

```
17              </value>
18          </slot>
19          <slot id="a0subs" name="Parameters">
20              <property oid="_4_256322" type="Property_Parameters_sysadmin_3083934035"
                    typeName="Parameters"></property>
21              <value>
22                <seq>
```

E.63: activity-send1-object-entrance.xml dosyası.

```
1   <object id="ACTIVITY-SEND-1" oid="ACTIVITY-SEND-1"
          type="Object_ActivitySend_sysadmin_3769158203" typeName="ActivitySend1">
2     <slot id="a6bg9h" name="Activity_streotype">
3       <property oid="_4_256167"
            type="Property_Name_Streotype_Single_3_sysadmin_3771844874"
            typeName="Name-Streotype-Single-3"></property>
4       <value>
5           <string>receive-request</string>
6        </value>
7       </slot>
8       <slot id="a4fp80" name="Service Name">
9          <property oid="_4_253448" type="Property_Service_Name_2_sysadmin_3669973713"
               typeName="Service Name-2"></property>
10         <value>
11          <seq>
12             <object oid="_4_253459" type="Object_Operation_sysadmin_3019391465"
                    typeName="Methods/Events">
13               <slot id="p1" name="Method/Event">
14                 <property oid="_4_253461" type="Property_Method_name_user_3012459728"
                       typeName="Service Name"></property>
15                 <value>
16                    <string>METHOD-NAME</string>
17                 </value>
18               </slot>
19               <slot id="a0subs" name="Parameters">
20                  <property oid="_4_253463"
                        type="Property_Parameters_sysadmin_3083934035"
                        typeName="Parameters"></property>
21                 <value>
22                   <seq>
```

E.64: activity-send2-object-entrance.xml dosyası.

```
1   <object id="ACTIVITY-SEND-2" oid="ACTIVITY-SEND-2"
          type="Object_ActivitySend2_sysadmin_3771844488" typeName="ActivitySend2">
2     <slot id="a1bfxx" name="Activity_streotype">
3        <property oid="_4_256295"
             type="Property_Name_Streotype_Single_4_sysadmin_3771844666"
             typeName="Name-Streotype-Single-4"></property>
4         <value>
5            <string>send-response</string>
6        </value>
7     </slot>
```

E.64: activity-send2-object-entrance.xml dosyası (devam).

```
8     <slot id="a0bfv2" name="Service Name">
9        <property oid="_4_256297" type="Property_Service_Name_2_sysadmin_3669973713"
              typeName="Service Name-2"></property>
10      <value>
11        <seq>
12         <object oid="_4_256287" type="Object_Operation_sysadmin_3019391465"
               typeName="Methods/Events">
13           <slot id="p1" name="Method/Event">
14             <property oid="_4_256289" type="Property_Method_name_user_3012459728"
                  typeName="Service Name"></property>
15             <value>
16               <string>METHOD-NAME</string>
17             </value>
18           </slot>
19           <slot id="a0subs" name="Parameters">
20             <property oid="_4_256291" type="Property_Parameters_sysadmin_3083934035"
                  typeName="Parameters"></property>
21             <value>
22               <seq>
```

E.65: activity-object-end.xml dosyası.

```
1               </seq>
2             </value>
3           </slot>
4         </object>
5       </seq>
6     </value>
7   </slot>
8 </object>
```

E.66: activity-diamond.xml dosyası.

```
1 <object id="DIAMOND-SEND-REQUEST" oid="DIAMOND-SEND-REQUEST"
         type="Object_Activity_Diamond_sysadmin_3769070158" typeName="Activity_Diamond">
2    <slot id="a0887q" name="pre_or_post">
3       <property oid="_4_251970" type="Property_pre_or_post_sysadmin_3769159763"
            typeName="pre_or_post"></property>
4       <value>
5          <string>PRE-POST</string>
6       </value>
7    </slot>
8    <slot id="a1890n" name="condition">
9       <property oid="_4_251986" type="Property_ActivityMessage_sysadmin_3769082355"
           typeName="ActivityMessage"></property>
10      <value>
11        <string>CONDITION</string>
12      </value>
13    </slot>
14 </object>
```

E.67: activity-lane.xml dosyası.

```
<object id="LANE-PARTITION" oid="LANE-PARTITION"
          type="Object_Lane_Partition_sysadmin_3769072366" typeName="Lane_Partition">
   <slot id="a04gcx" name="Required Component Name">
     <property oid="_4_251525" type="Property_Name_sysadmin_3645689316"
       typeName="Name"></property>
     <value>
       <string>COMPONENT-NAME-REQUIRED</string>
     </value>
   </slot>
   <slot id="cplnr" name="Provided Component Name">
     <property oid="_4_251527" type="Property_Name_sysadmin_3645689316"
       typeName="Name"></property>
     <value>
       <string>COMPONENT-NAME-PROVIDED</string>
     </value>
   </slot>
</object>
<object id="START-REQUEST" oid="START-REQUEST" type="Object_Start_user_3012721153"
      typeName="Start"></object>
<object id="START-PROVIDED" oid="START-PROVIDED" type="Object_Start_user_3012721153"
      typeName="Start"></object>
```

E.68: activity-binding.xml dosyası.

```
<binding id="OBJECT-1-OBJECT-2" nextConnectionId="3" oid="OBJECT-1-OBJECT-2">
  <relationship oid="_4_251538" type="Relationship_Activity_Message_sysadmin_3769082380"
       typeName="Activity-Message"></relationship>
  <connection id="1">
     <role oid="_4_251546" type="Role_Sends_user_3012479298" typeName="From"></role>
     <object href="#OBJECT-1"></object>
  </connection>
  <connection id="2">
     <role oid="_4_251548" type="Role_Receives_user_3012479314" typeName="To"></role>
     <object href="#OBJECT-2"></object>
  </connection>
</binding>
```

E.69: activity-diyagram-entrance.xml dosyası.

```
<diyagram displayGrid="false" grid="10,10" oid="_4_250575"
      timeStamp="2020-06-08T11:00:02.524Z"
      useGrid="true" xmlns="http://www.metacase.com/gxlGOPRRRepr">
<graph href="#SUB-GRAPH" xmlns="http://www.metacase.com/gxlGOPRR"></graph>
```

E.70: activity-LaneRepr.xml dosyası.

```
<objectRepr oid="OBJECTREPR" place="797.0,380" scale="3.535,2.27586">
  <object href="#OBJECT-1" xmlns="http://www.metacase.com/gxlGOPRR"></object>
</objectRepr>
```

E.71: activity-objectRepr.xml dosyası.

```
<objectRepr id="OBJECTREPR" oid="OBJECTREPR" place="300,310" scale="1.02333,0.625">
  <object href="#OBJECT-1" xmlns="http://www.metacase.com/gxlGOPRR"></object>
</objectRepr>
```

E.72: activity-bindingRepr.xml dosyası.

```
<bindingRepr isStraight="false" oid="_4_251552" place="350,310" scale="1,1">
 <binding href="#SUB-BINDING" xmlns="http://www.metacase.com/gxlGOPRR"></binding>
 <connectionRepr id="1">
     <objectRepr href="#OBJECT-1"></objectRepr>
 </connectionRepr>
  <connectionRepr id="2" objTargetOffset="-96,-1">
   <objectRepr href="#OBJECT-2"></objectRepr>
  </connectionRepr>
</bindingRepr>
```

E.73: activity-diyagram-end.xml dosyası.

```
        </diyagram>
      </graph>
 </object>
```

template-sub-graph-id klasörünün içindeki dosyalar:

E.74: sub-graph-id.xml dosyası.

```
<graph href="#SUB-GRAPH-ID"></graph>
```