

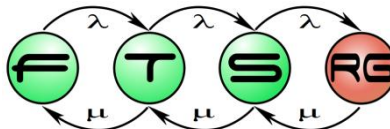
Folyamatmodellezés – implementáció

BPEL

jBPM

(XPDL)

Kitekintés: folyamatanalízis



Motiváció

- Pl.: Bank:
 - Ahány beszállító, annyi technológia, módszertan, protokoll...
 - Régi eszközöket soha nem selejteznek le
- Meglévő workflow eszközök
 - Gyártófüggőek
(adatbázis, Web szerver, stb. kötött)
 - Editor – Engine szorosan összekapcsolódik
- Összetett folyamatok építése
- Web service alapelemekből
- Kommunikációs protokoll ne legyen kötött

Miért Web service alapon?

- Laza kapcsolat
- Platformfüggetlenség
- Szabványos felület
- Folyamatok adatai nyílt formátumban (SOAP)
- További szolgáltatások (kivételkezelés, stb.)
- Web Service interface:
 - Szintaktika: WSDL
 - Szemantika: ...?

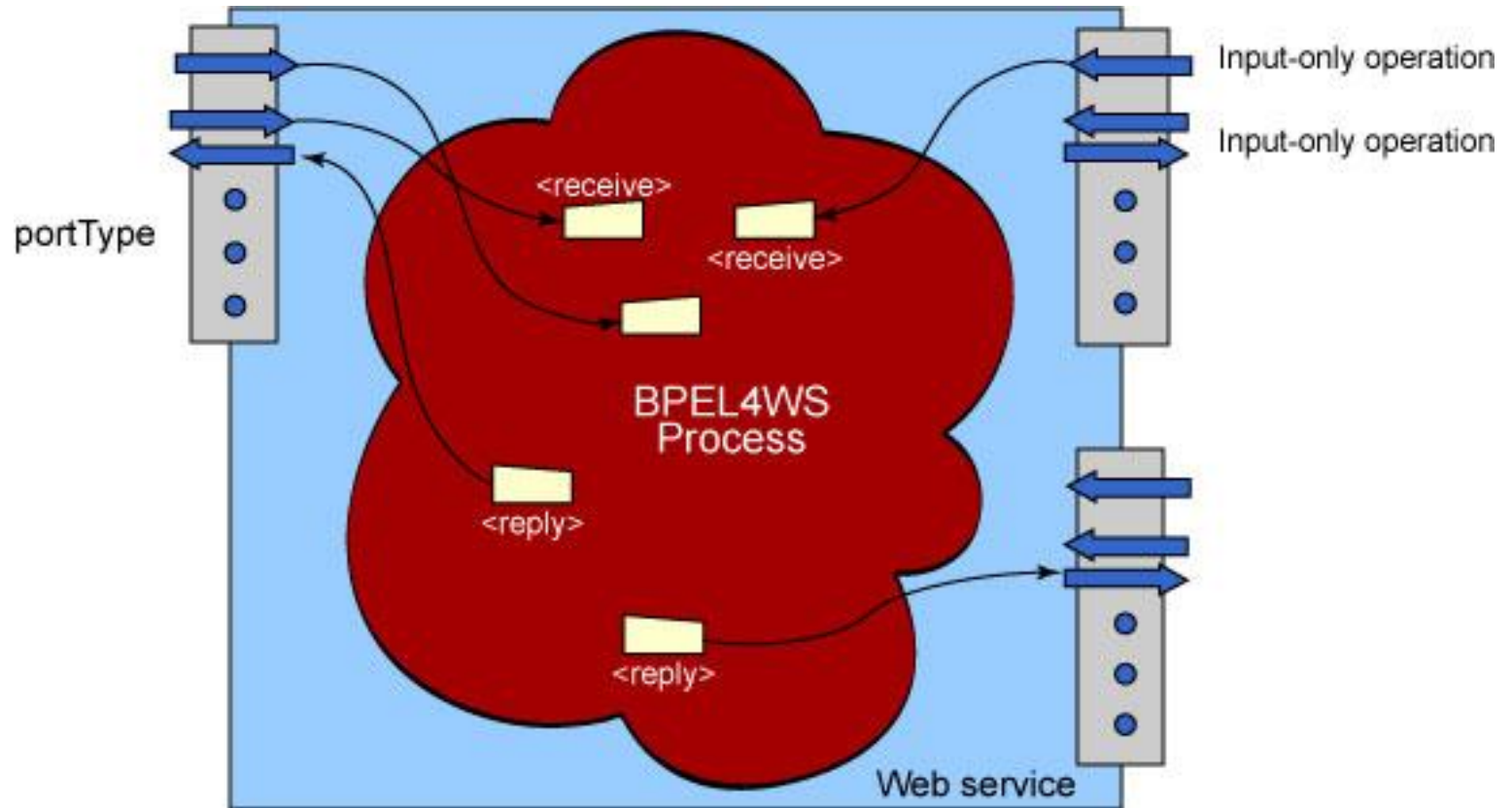
BPEL (BPEL4WS)

- Business Process Execution Language for Web Services
 - Microsoft, IBM, BEA, ...
- Üzleti folyamatok definiálása
 - Web service alapon
 - Web service-ként
- Grafikus tervezőeszköz(ök)
 - Integráció más tervezőeszközökkel
 - BPMN -> BPEL átalakítás (BPMN szabvány része)
- Java alapú futtatókörnyezet

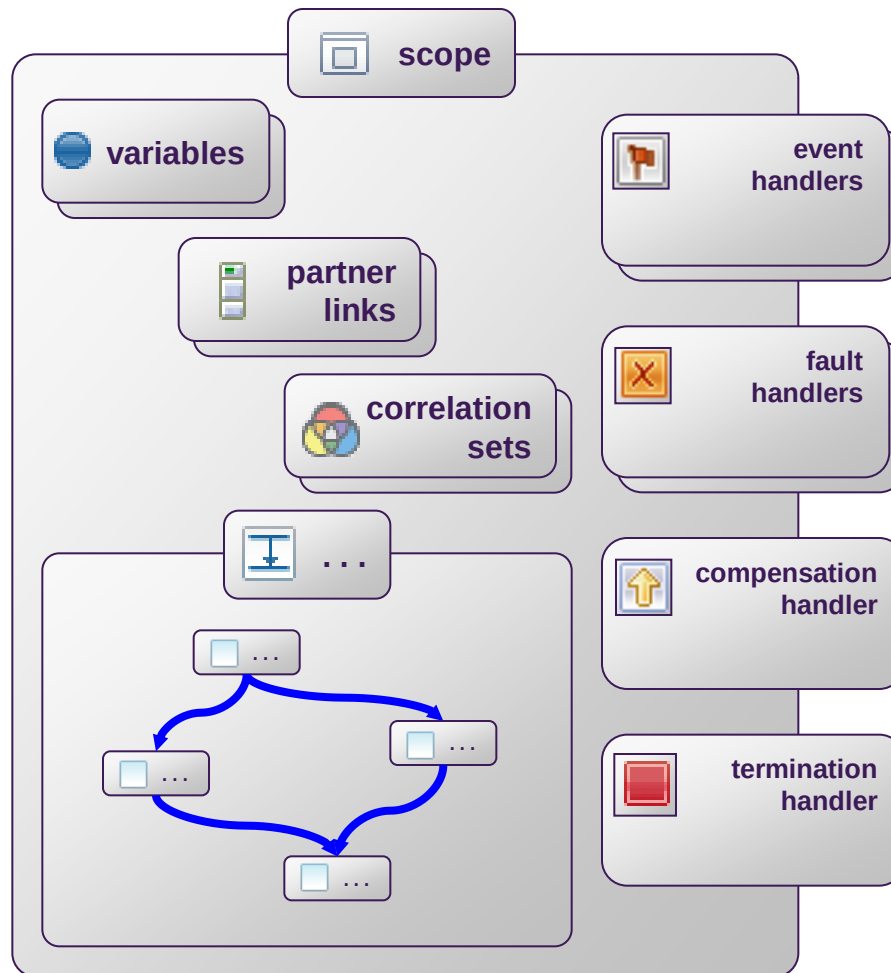
BPEL funkcionalitás

- Partnerek
- Adattípusok leírása
- Végrehajtási sorrend
- Kivételkezelés
- Két felhasználási mód
 - Végrehajtható folyamatok
 - Privát tartalom
 - Absztrakt processzek (Pl. szerződéshez)
 - Hozzáférhető

BPEL alapkonceptió



Logikai felépítés



A BPEL elemei

■ Process felépítése

`<partnerLink>` - Kapcsolat két szereplő közt

`<partner>` - `partnerLink`-ek halmaza

`<variable>` - folyamat globális változói, ideértve a kimenő/bejövő üzeneteket

`<Activity>` - folyamat lépései (tevékenységek)

A BPEL elemei 2.

■ Basic Activity (egyszerű tevékenység)

`<invoke>` - Web service meghívása

`<receive-reply>` - Web service funkcionalitás a partnerek felé

`<throw>` - belső hiba jelzése (kivétel)

`<wait>` - várakozás

`<assign>` - változók elérése, adatmanipuláció

`<compensate>` - compensationHandler hívás sikeresen lefutott lépés visszavonása

A BPEL elemei 3.

- Structured activity (strukturált tevékenység)
 - <sequence> - sorosan végrehajtandó lépések
 - <switch> - feltételes elágazás
 - <while> - ciklus
 - <pick> - üzenetek, alarmok kezelése
 - <flow> - párhuzamosan végrehajtható tevékenységek

BPEL elemei 4.

- `<scope>` : speciális structured activity:
 - Fő munkafolyamat : activity
 - Event handlers
 - Fault handlers
 - Compensation handlers
 - Lokális változók
 - Sorosítható változók (csak levél scope)

BPEL példa (minimalista)

```
<sequence name="EchoSequence">
```

```
  <receive name="EchoReceive"  
    partner="caller"  
    portType="tns:echoPT"  
    operation="echo"  
    container="request"  
    createInstance="yes"/>
```

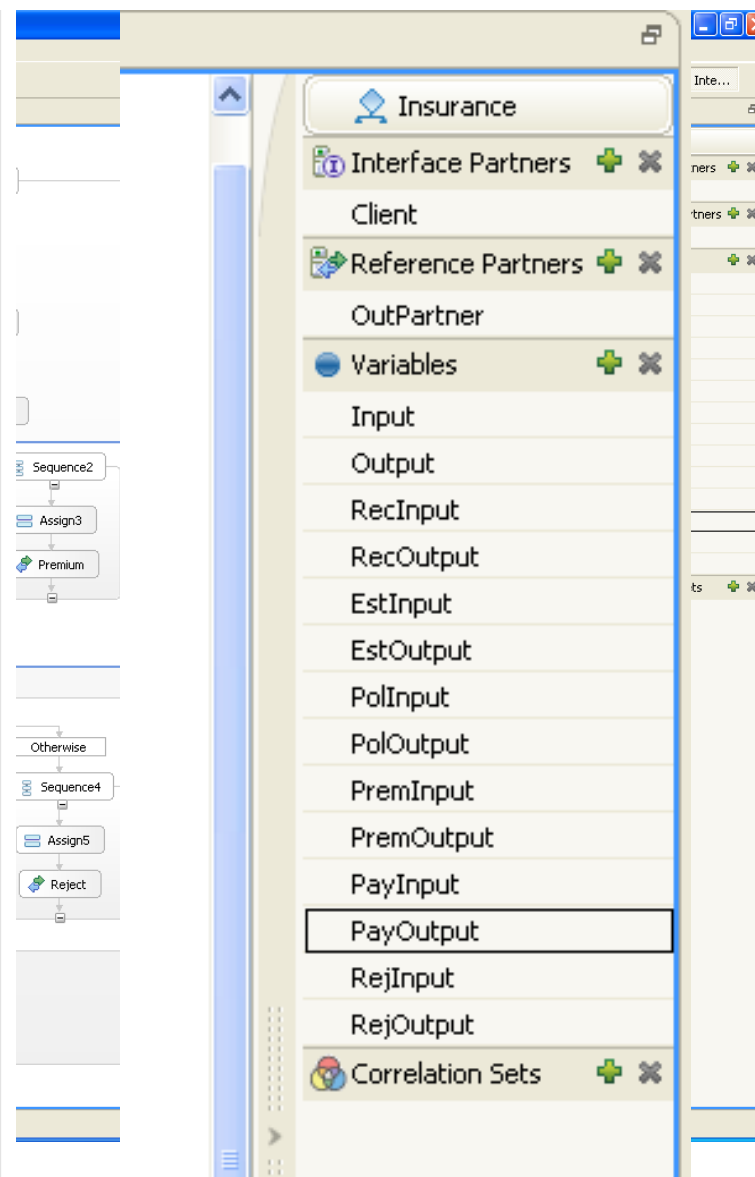
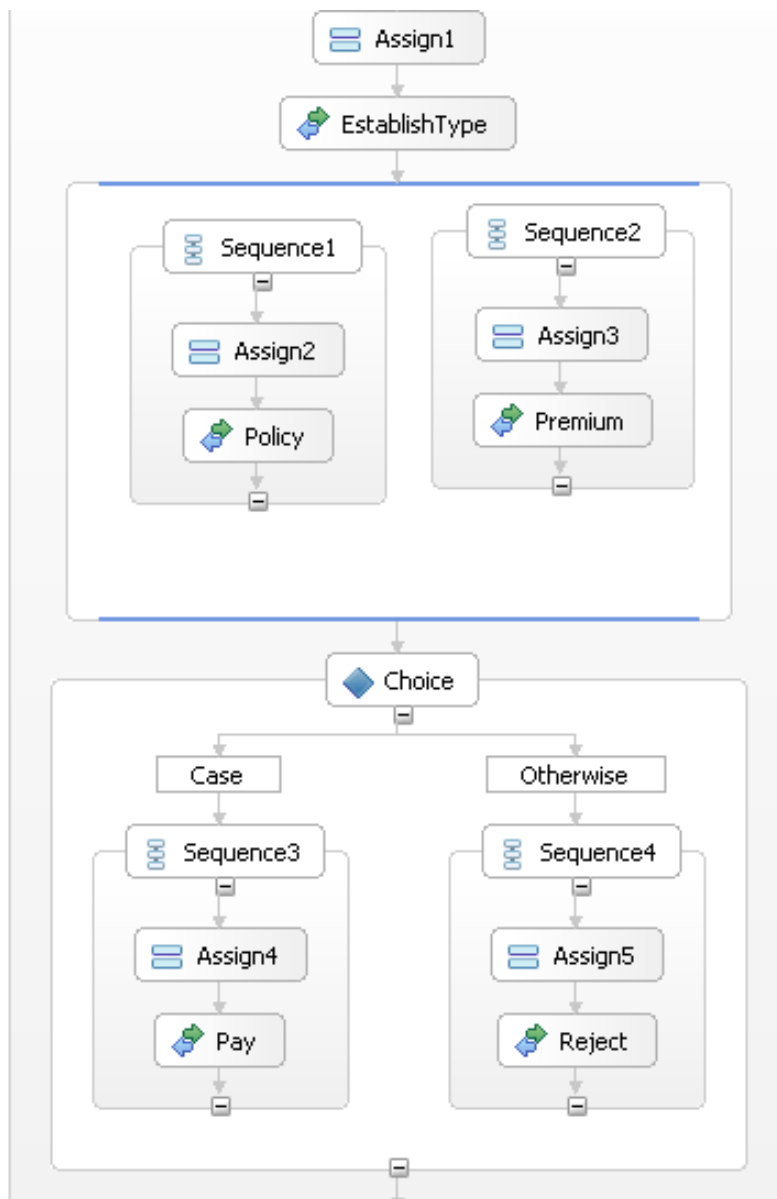
```
  <reply name="EchoReply"  
    partner="caller"  
    portType="tns:echoPT"  
    operation="echo"  
    container="request"/>
```

```
</sequence>
```

BPEL szerkesztés

- Eclipse BPEL
- Oracle BPEL Designer (régen Collaxa)
 - Eclipse
 - JDeveloper
- Más fejlesztőeszközök kimenete
 - IBM WebSphere Integraton Developer és utódai
 - Nem szabványos BPEL
 - Rational XDE
 - Sybase Power Designer
 - ...

Példa (IBM WebSphere Integration Developer)



BPEL futtatás

- „Workflow engine”
- Factory szolgáltatás
 - BPEL processz példányosítása
 - Paraméterezés
 - (Java API)
- Kivételek kezelése
- Adatok tárolása
 - Pl. minden „kompenzálható” lépésre a változók értéke a lépés *indításakor*
- Tipikusan plugin/alkalmazás Java alkalmazáserverre

Hibakezelés BPEL-ben

- Scope szintjén
 - Egyedi elem vagy folyamat részlet
- Fault handling
 - Hiba típusokra, változókra definiálható
 - Akkor hívódik meg, ha a fellépő hiba illeszkedik
 - Mint Java-ban az exception handling
 - Alulról felfele hívódik meg (scope-hierarchia)
- Compensation handling
 - Tranzakció megghiúsul?
 - Erőforrás zárolás?
 - Kompenzáció
 - Rekurzívan, fordított sorrendben
 - Változó értékek megtartása

BPEL integráció meglévő szoftverekkel

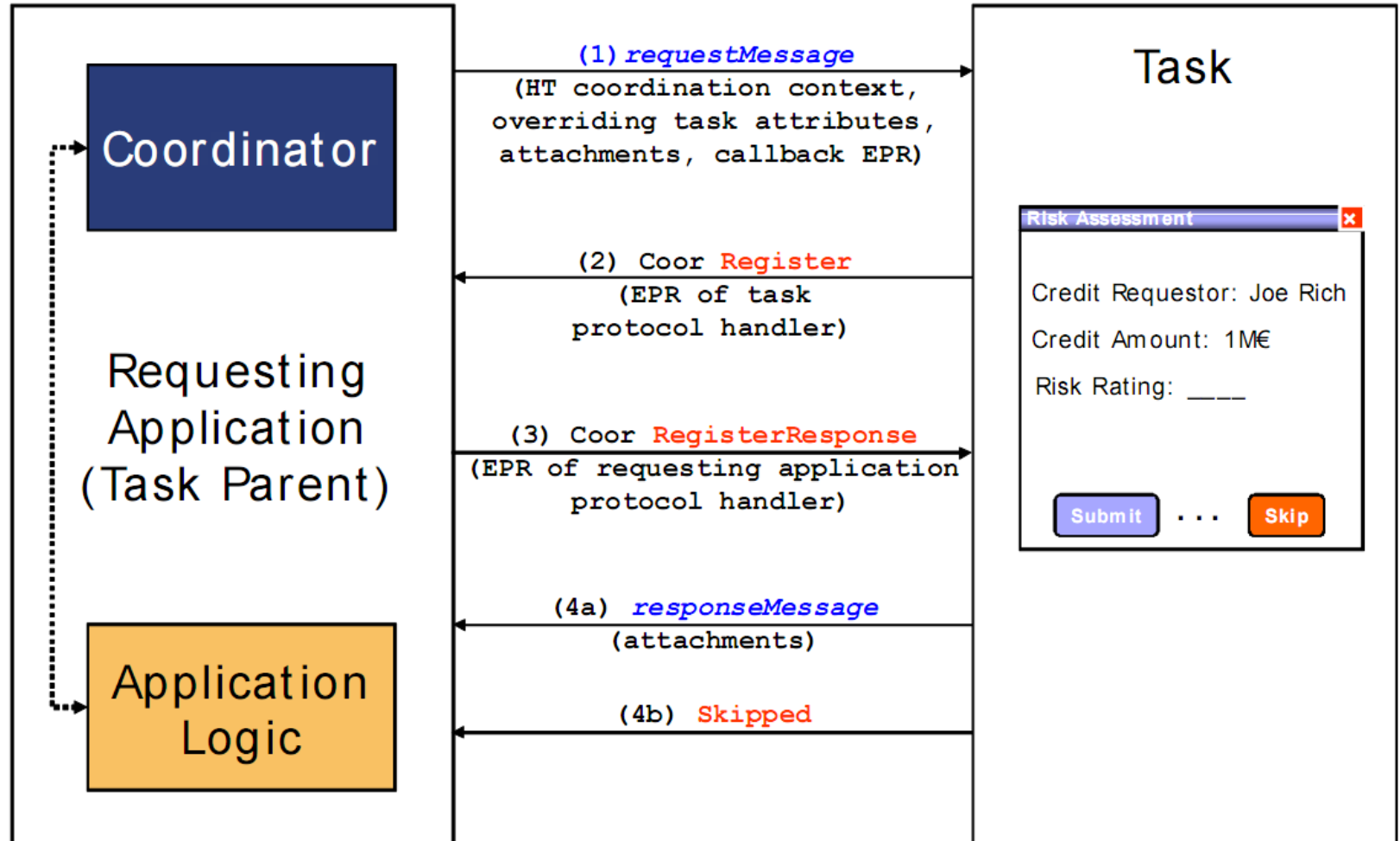
- Integrálás programnyelvekhez
 - Java package
 - Grafikus szerkesztés nehézkes
 - Kód karbantartás
- Kommunikáció
 - pl. JMS fölött
- Titkosítás
 - WS-Security
 - A BPEL futtató nem kell, hogy ismerje
- Adatbázis tárolt eljárások
 - DB2, MSSQL, Oracle ... Web service felülettel
- Adapterek (trigger, kimenethez)
 - Oracle: File rendszer, Adatbázis, SAP, ...
- Új javaslat: BPEL folyamatok mint komponensek integrálása (Service Component Architecture)
- Számptalan gyártóspeficikus kiegészítés
 - Pl. Oracle BPEL Extensions, IBM WID, ...

Ipari BPEL projektek (Oracle)

- Belgacom
 - 15000 kérés/nap, 2*2 CPU Solaris (RDBMS külön gépen)
 - Egyszerre pár száz aktív folyamatpéldány, pár száz lépés/folyamat
- Utazási iroda
 - 3 M kérés / óra, 20 CPU
 - Egyszerű folyamatok
- European Space Agency
 - Dokumentum feldolgozás

- Web szolgáltatások emberi végrehajtása. Ehhez:
 - Nyelv definiálása
 - Interface-k
 - Általános emberi szerepek
 - Protokoll
 - Állapottér
 - Típus: HumanTask és Notification

Protokoll



XPDL

- „Workflow adatcsere” szabvány
 - XML Process Definition Language
 - Workflow eszközök gyártói (WFMC)
 - Tárolási formátum (BPMN ezt nem rögzítette 2.0-ig)
 - „Szinkronban” a BPMN2 szabvánnyal
- „Szokásos” vezérlési szerkezetek
- Egyes lépések megvalósítása tetszőleges
 - „Activity”
- Kiterjeszthető

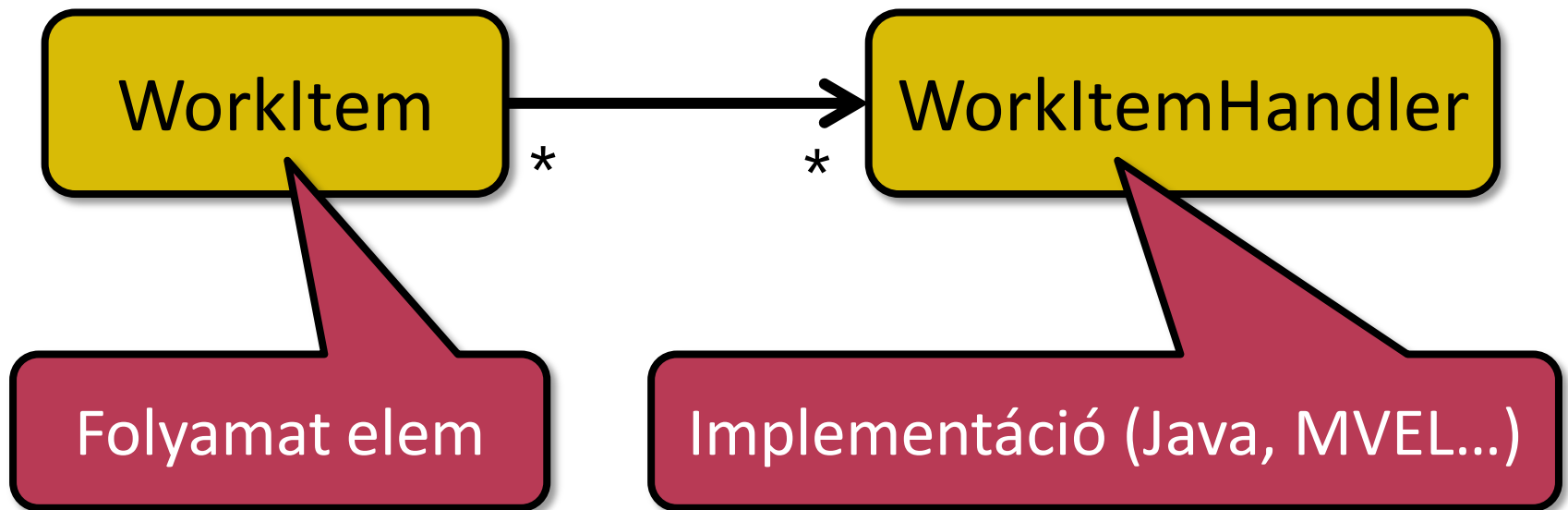
Modellezési nyelvek kapcsolata

- BPMN → BPEL
 - Szabvány definiál leképezést
- UML-AD → BPEL
 - Létezik hozzá UML profile
 - Szemantika különbözik....
 - (Action Semantics?)
- UML $\leftrightarrow$ BPMN
 - Egy BPMN folyamat UML tervezés alapját képezheti
 - Részletesebb nézetek, pl struktúra definíció
 - Számtalan eszköz kezeli mindkét modellt
- XPDL
 - Nincs egységes szemantika
 - Eszköz specifikus megvalósítás

JBoss jBPM alapú implementáció

jBPM5 WorkItem példa

- Webszolgáltatást szeretnénk hívni
- De ilyen elem nincs
 - Külső szolgáltatást igénybe vétele: Service Task
 - A szolgáltatás WS jellege már domain specifikus
 - Lehetne OSGi szolgáltatás, platform feature, ...



jBPM5 WorkItem példa

■ WorkItem elem definiálása

```
[  
  [  
    "name" : "WSInvocation",  
    "parameters" : [  
      "URL" : new StringDataType(),  
      "Message" : new StringDataType(),  
      "ReturnValue" : new IntegerDataType(),  
    ],  
    "displayName" : "WSInvocation",  
    "icon" : "icons/webservice.gif"  
  ]  
]
```

jBPM5 WorkItem példa

■ WorkItemHandler implementálása

```
package com.handlers;

+ import org.drools.process.instance.WorkItemHandler;

public class WSInvocationWorkItemHandler implements WorkItemHandler {

-    @Override
    public void abortWorkItem(WorkItem wi, WorkItemManager wim) {
        // TODO Auto-generated method stub
    }

-    @Override
    public void executeWorkItem(WorkItem wi, WorkItemManager wim) {
        String address = (String)wi.getParameter("address");
        String message = (String)wi.getParameter("message");
        System.out.println("Calling WS @" + address + ", with message " + message + "...");

        Client client = Client.create();
        WebResource webResource = client.resource(address + message);

        String s = webResource.get(String.class);

        System.out.println(s);

        wim.completeWorkItem(wi.getId(), null);
    }
}
```

jBPM5 WorkItem példa

- **WorkItem és WorkItemHandler összerendelése**

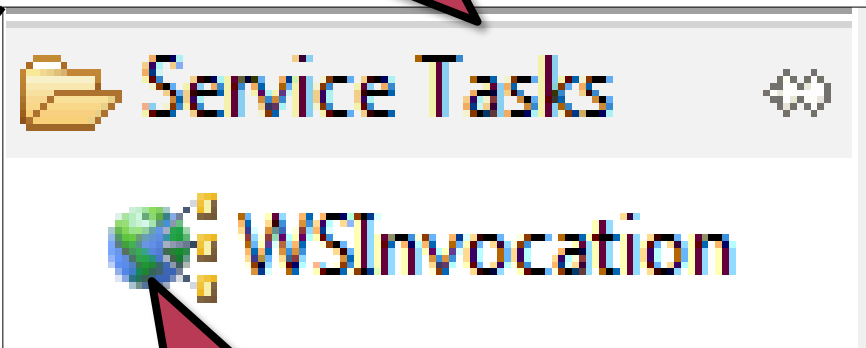
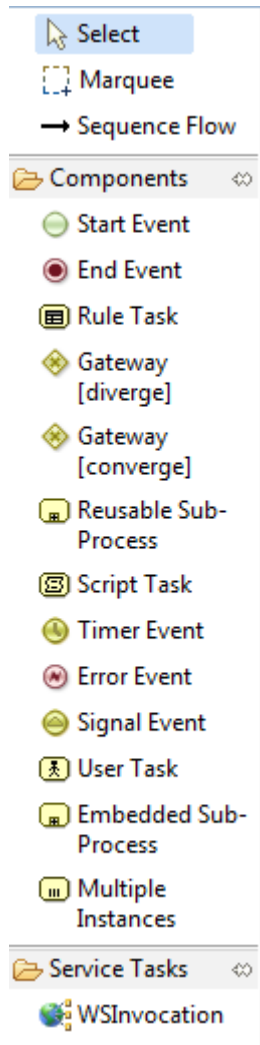
```
ksession.getWorkItemManager()  
    .registerWorkItemHandler(  
        "WSInvocation",  
        new WSInvocationWorkItemHandler()  
    );
```

- **WorkItem regisztrálása a toolbarra**

```
drools.workDefinitions = MyWorkDefinitions.wid
```

jBPM5 WorkItem példa

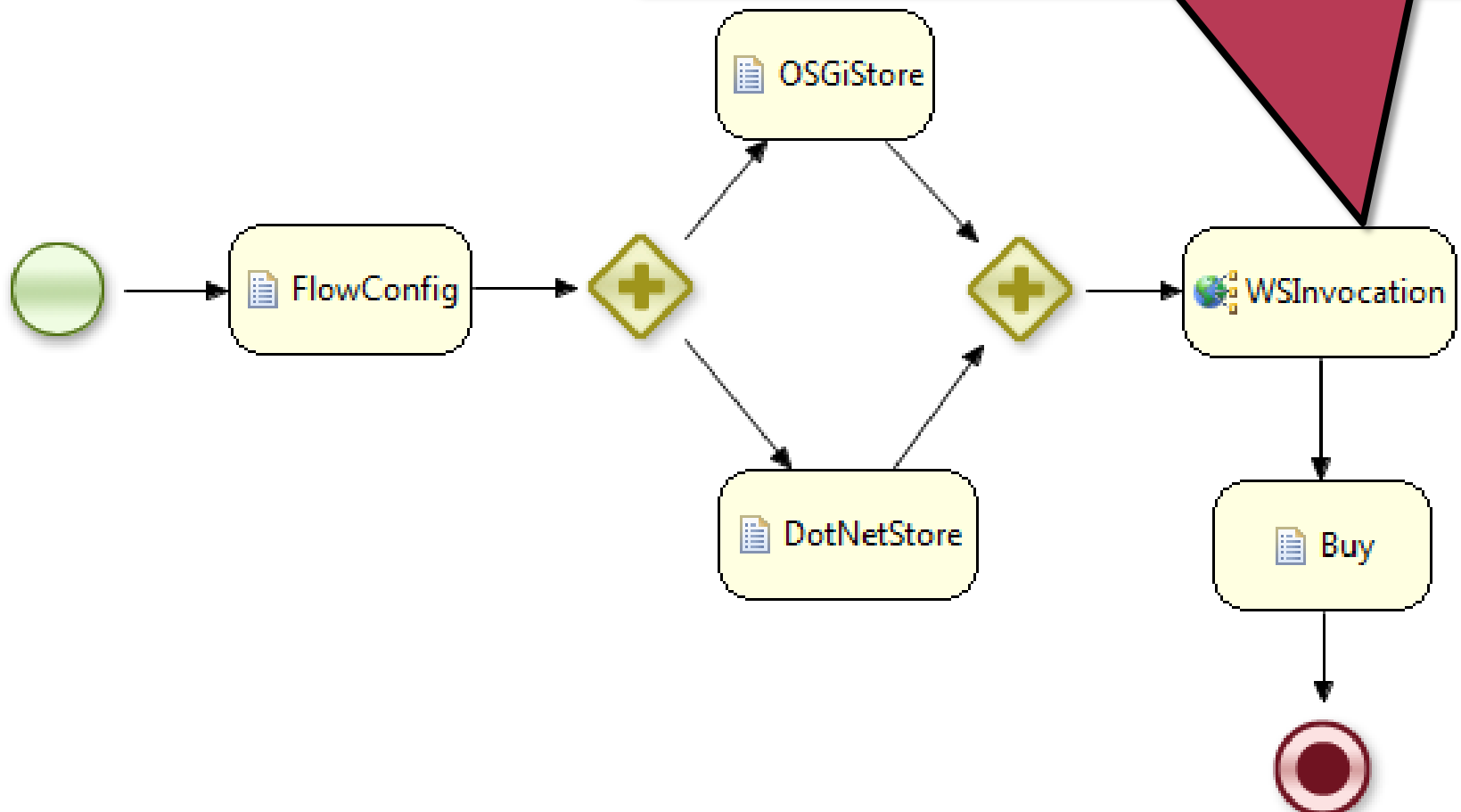
A palettán Service Task
elemként megjelent a
WSInvocation



Még az ikon is custom.

jBPM5 WorkItem példa

Az új elem beilleszthető a folyamatba. Ha ide érünk, meghívódik a megfelelő Handler.

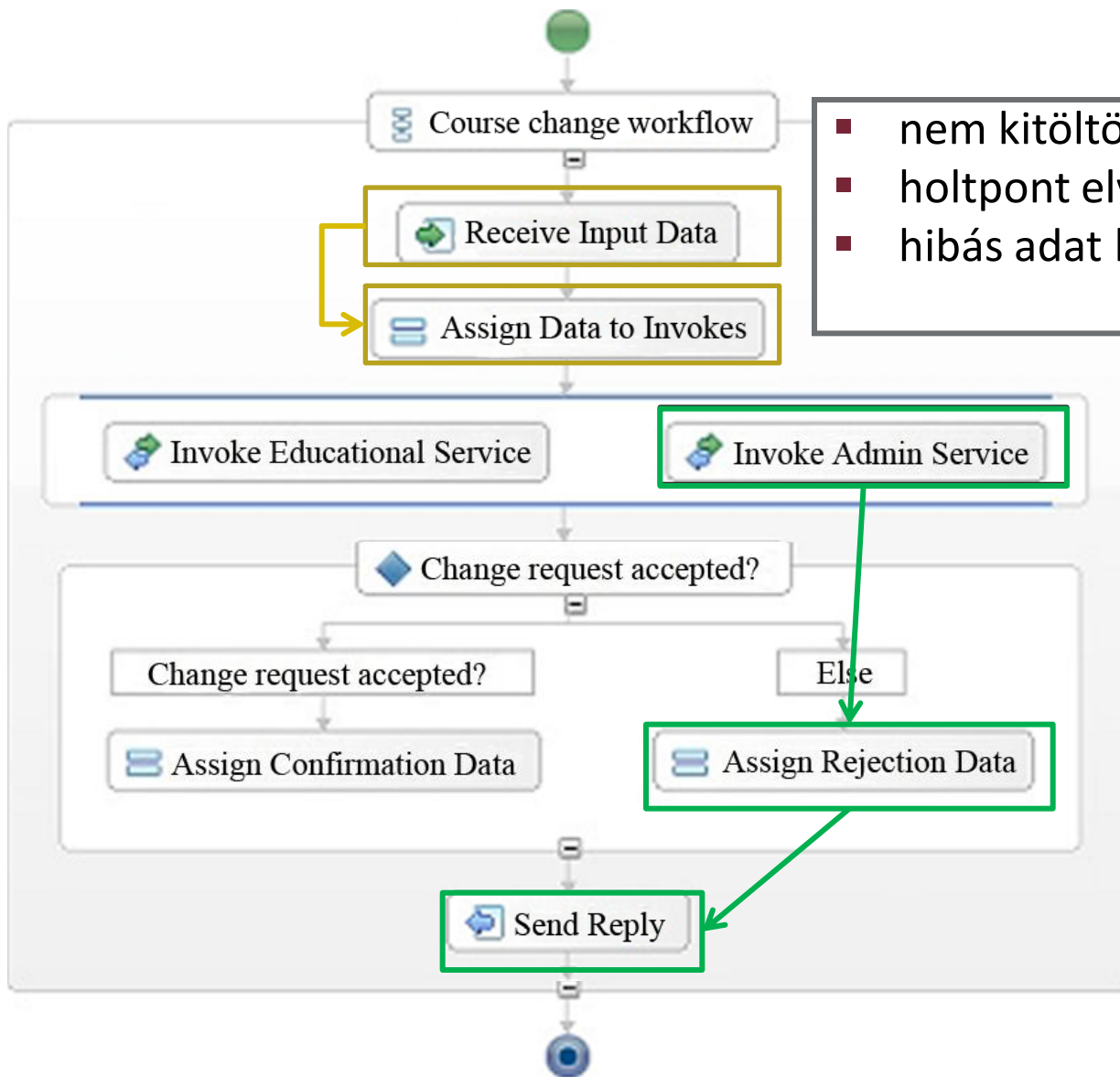


Kitekintés: üzleti folyamatok analízise

Üzleti folyamatok verifikációja

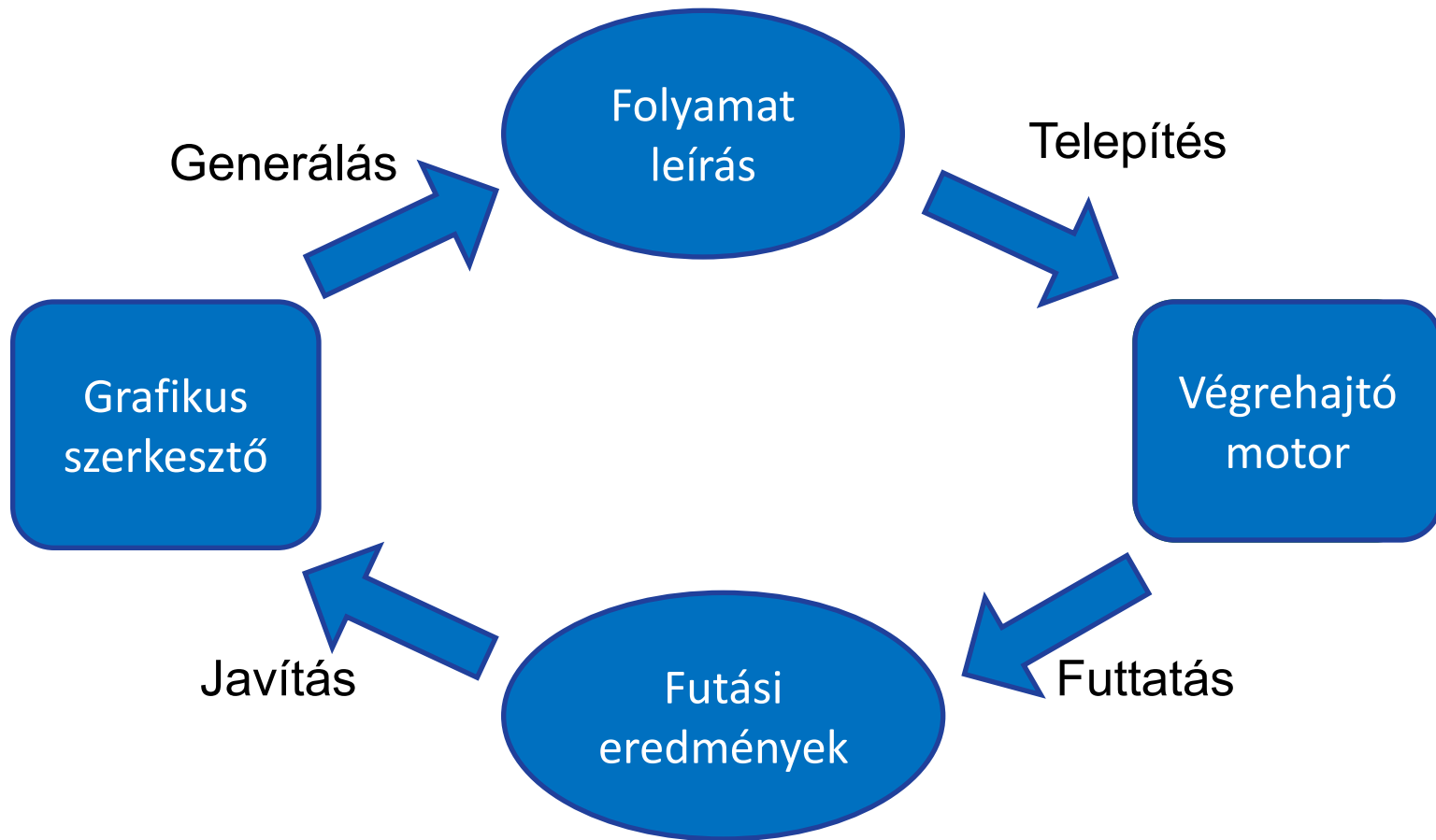
- Tervező eszközök csak szintaktikai ellenőrzést biztosítanak
- BPEL 2.0 szabványban statikus analízis
 - Megkötések a folyamatra (attribútumok, struktúra)
- Biztonságkritikus területeken való felhasználás
 - Pl. Bank, tőzsde, egészségügy
- Hogyan biztosítjuk a helyes működést?

BPEL munkafolyamatok hibái

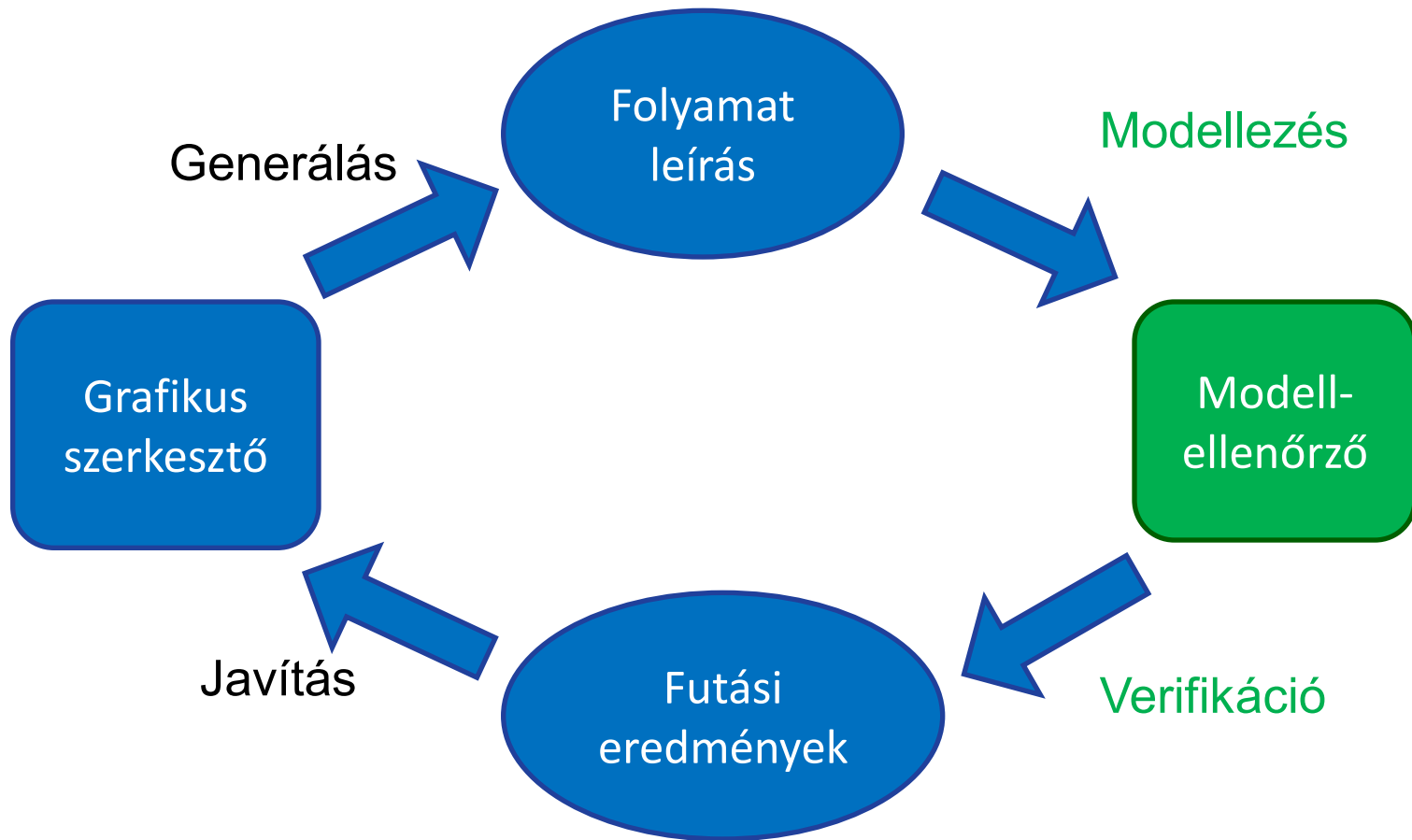


- nem kitöltött adat okozta hiba
- holtpont elveszett üzenet miatt
- hibás adat hatása a végrehajtásra

Munkafolyamatok vizsgálata



Munkafolyamatok vizsgálata



BPEL modellezése

- Formális matematikai modellre
- Nem-interpretált modellezés
 - Változó értékek, tevékenységek helyett állapotok
- Modelltranszformáció használatával
- Eredmény:
 - Ellenőrizhető modell

Ellenőrzés elvégzése

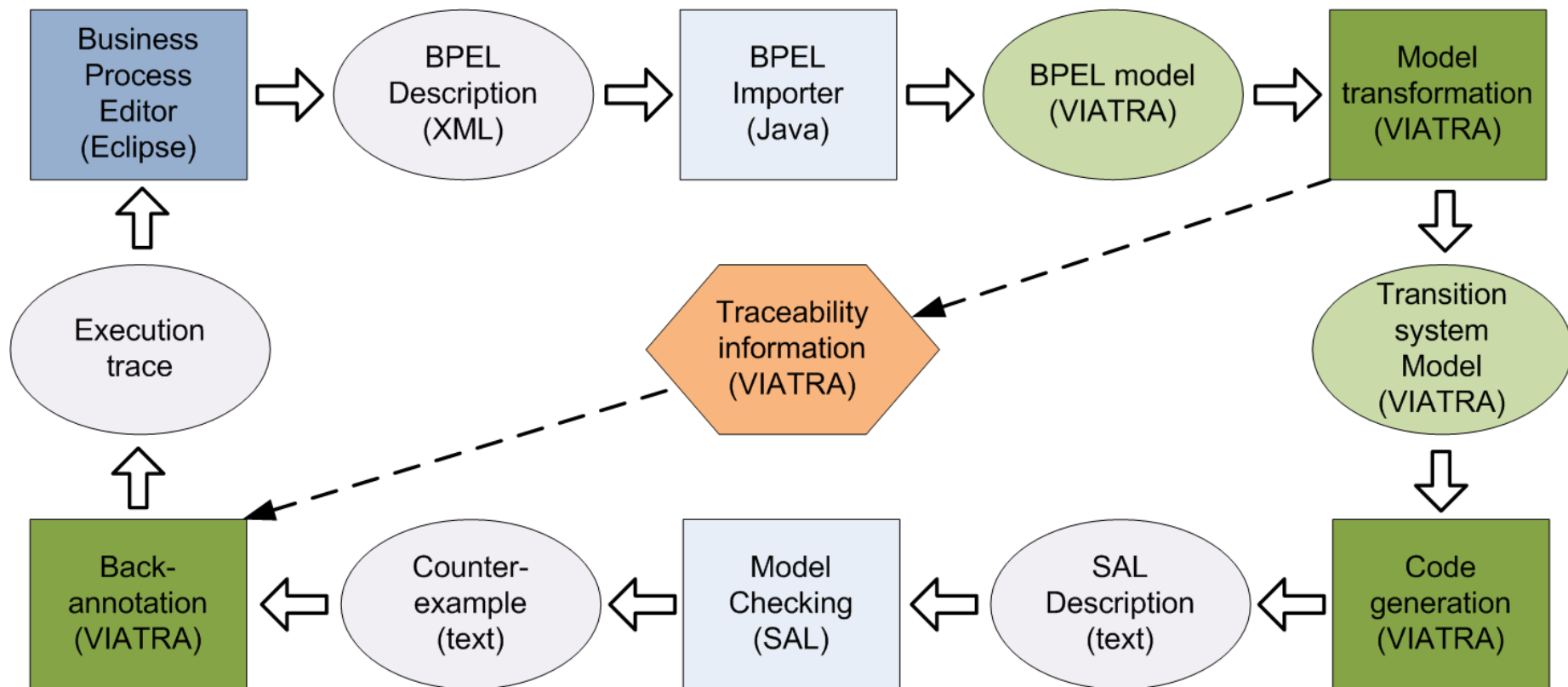
■ Kezdetben:

- Kézi követelmény leírás
- Parancssoros eszköz
- Eredmények kézi feldolgozása
- Nagyon lassú, körülményes

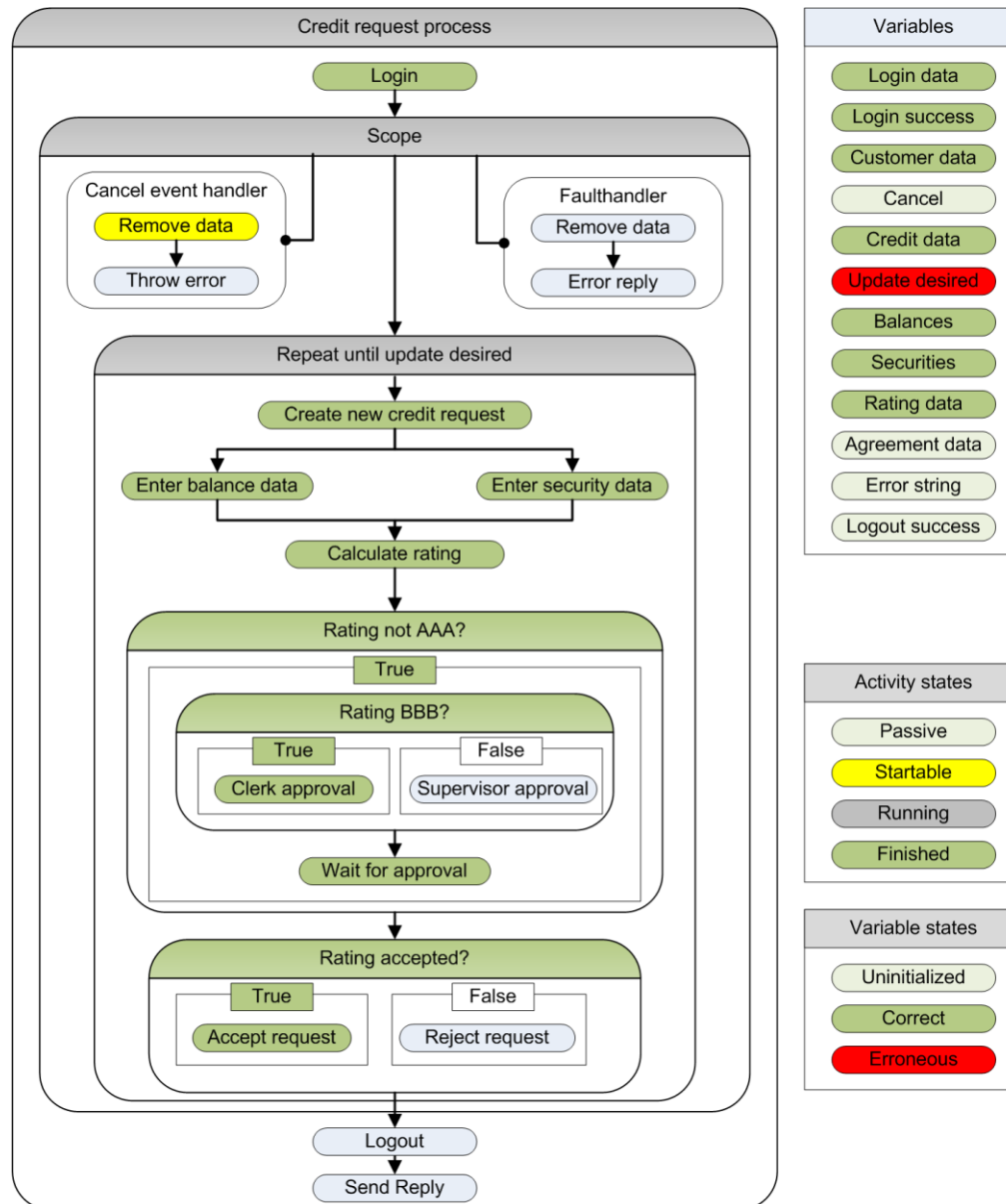
■ Helyette:

- Integráció a folyamat modellező rendszerbe
- Intuitív felhasználói felület
- Követelmény leírás támogatása
- Eredmények visszavetítése

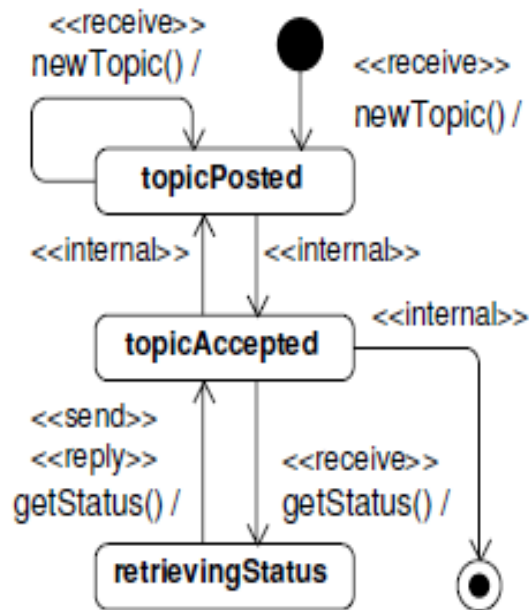
Transzformációs lánc



Az esettanulmány vizsgálata



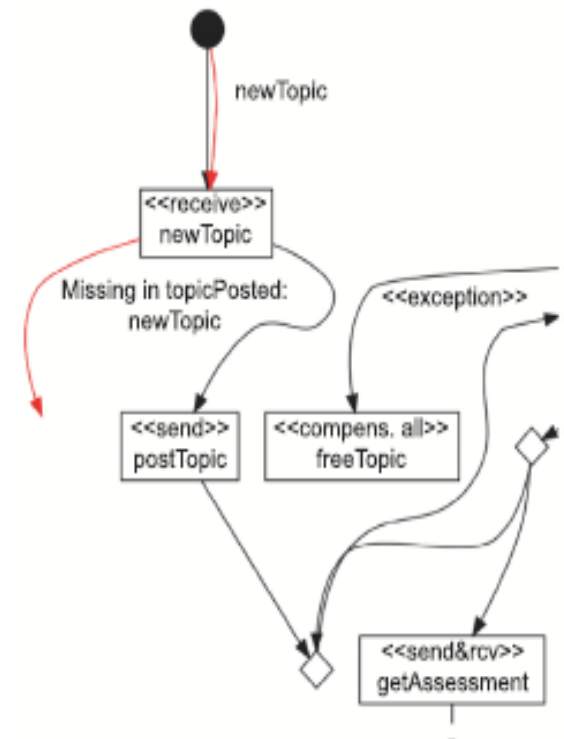
Példa: szolgáltatások és protokollok



(a) Wrong Tutor Protocol

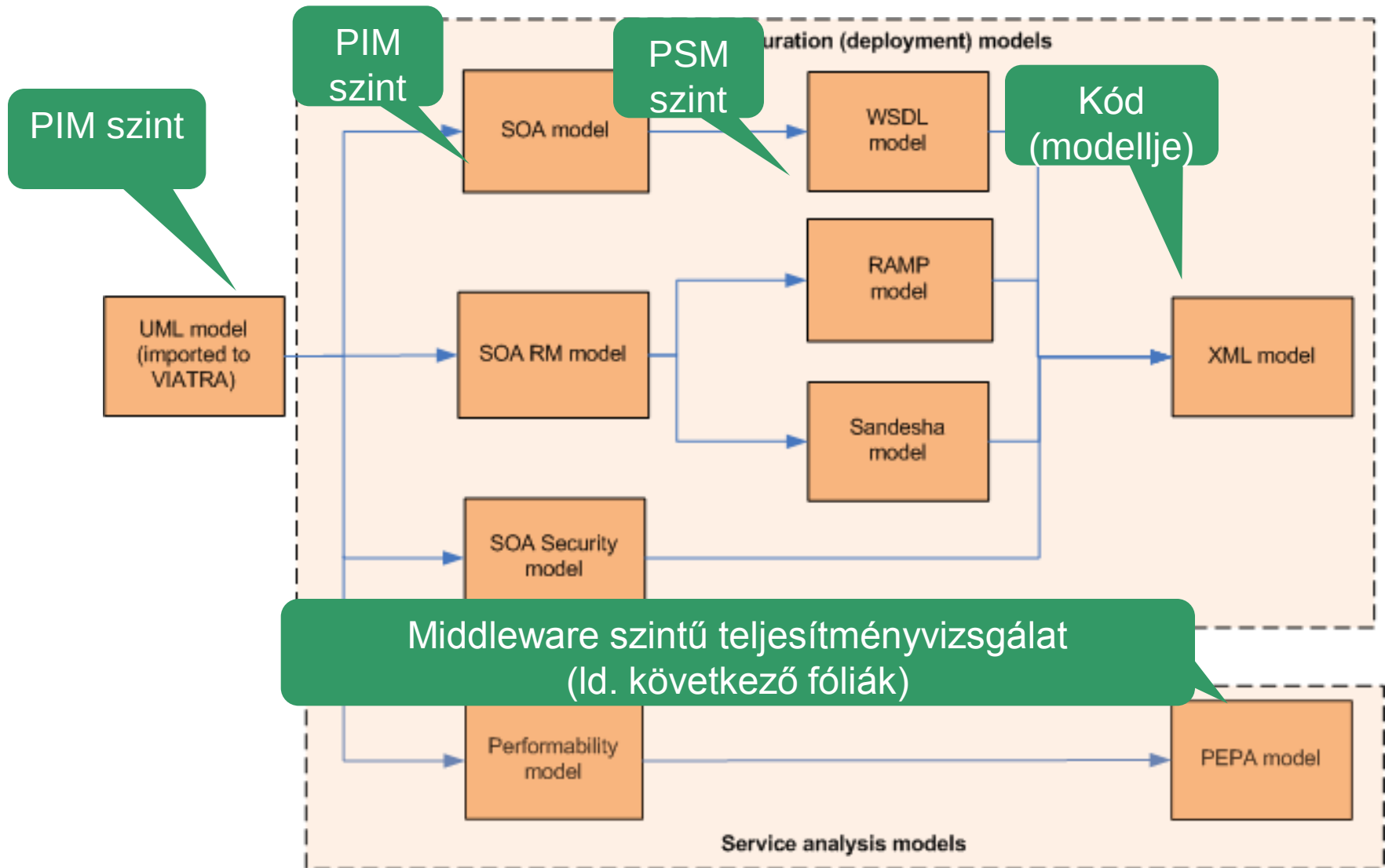


(b) Orchestration LTS



(c) Resulting Analysis Output

Telepítési (deployment) modellek



Transzformációk működése

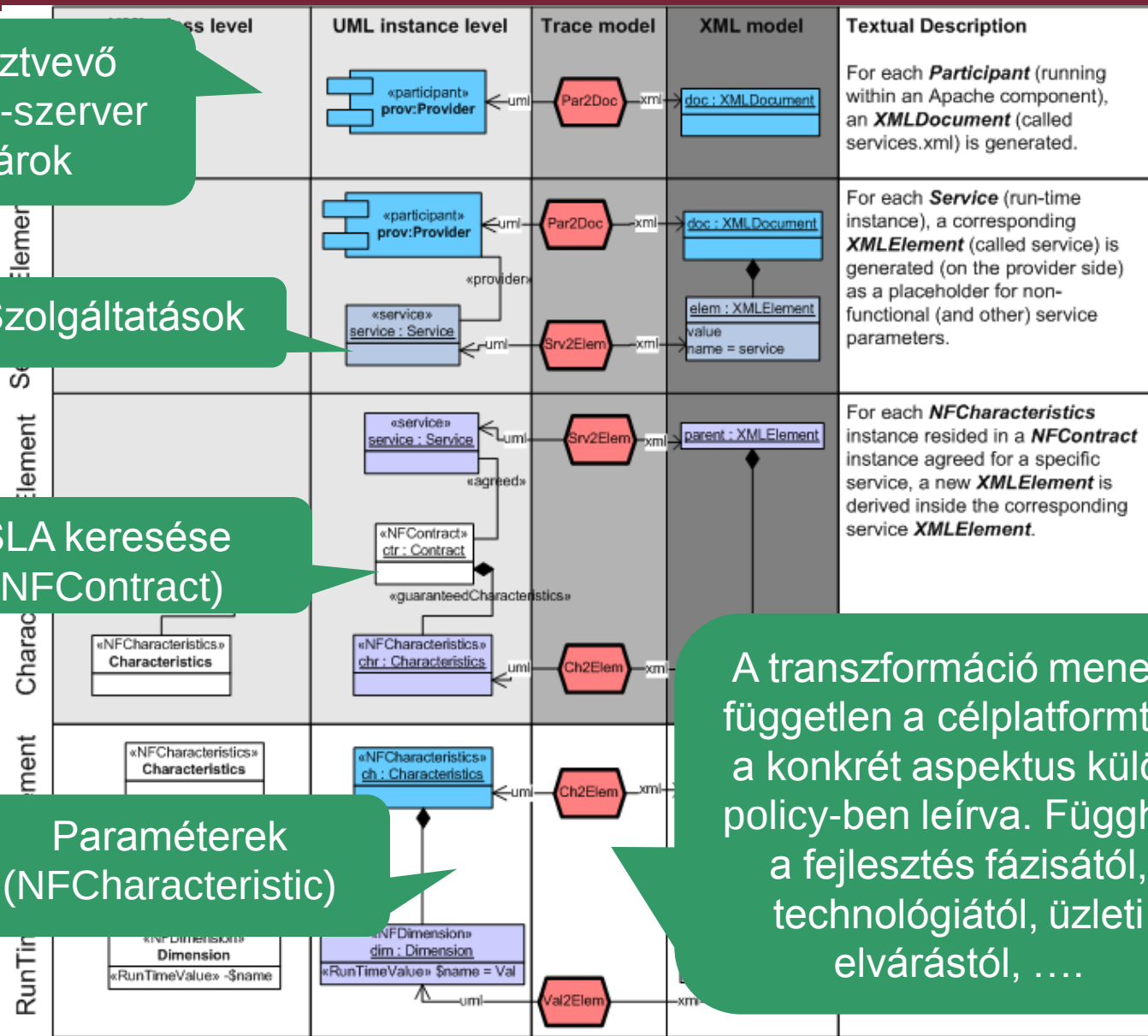
Résztevő
kliens-szerver
párok

Szolgáltatások

SLA keresése
(NFContract)

Paraméterek
(NFCharacteristic)

A transzformáció menete
független a célplatformtól,
a konkrét aspektus külön
policy-ben leírva. Függhet
a fejlesztés fázisától,
technológiától, üzleti
elvárástól,

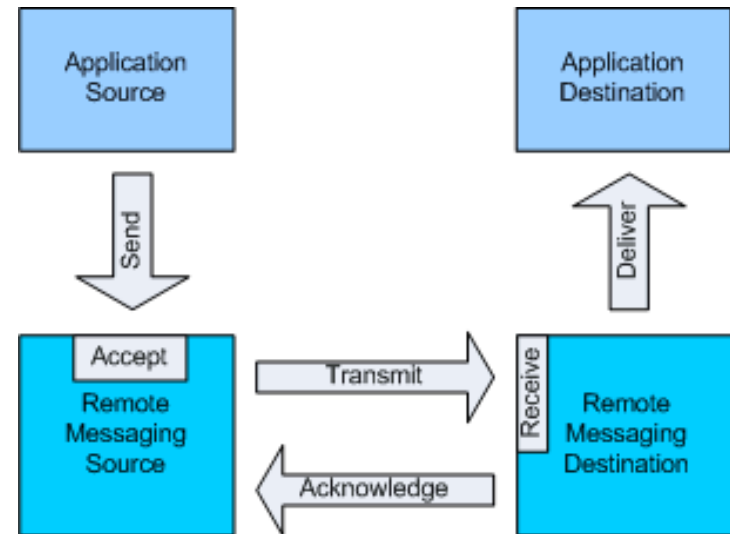


Szolgáltatás konfigurációk ellenőrzése

- Teljesítőkéesség vizsgálata
 - „Performability = Performance + Reliability”
- Mi történik hibák esetén?
 - Pl. a megbízható üzenetküldést garantáló middleware újraküldi az elveszett üzenetet → megnőhet a garantált válaszidő
 - Pl. túl kicsi a beállított timeout → téves újraküldések
- Mi a megbízhatóság ára? (teljesítményben)
- Hogyan állítsuk be az SLA paramétereket??

Felhasznált technológia

- Reliable messaging: WS-RM
 - „TCP protocol”
- Mit jelent?
 - Nyugtázás
 - üzenetsorrendezés
 - Duplikátumok szűrése
 - Garantált kézbesítés
- Üzenetküldési szemantika
 - At-least once
 - At-most-once
 - Exactly-once
- Több szabványból fejlődött ki (MS, IBM)
- implementációk
 - RAMP (IBM WebSphere Application Server)
 - Apache Sandesha (Axis2)
 - Microsoft Windows Communication Foundation
 - Bea WebLogic (→Oracle)

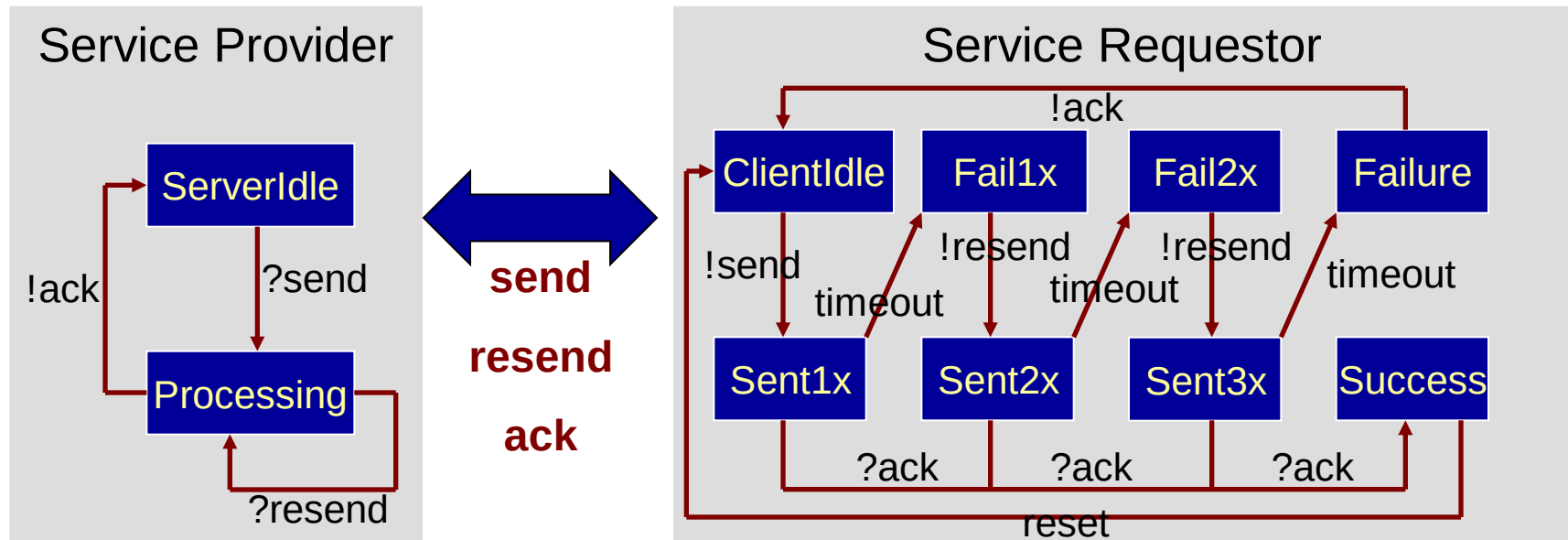


Mit modellezünk ebből?

- Absztrakt viselkedés
 - Szolgáltató
 - Kliens
- Üzenetkezelés paramétereit (származtatottak)
 - Üzenetkezelés módja
 - Újraküldések száma
 - **send, resend, ack** paraméterek
 - (exponenciális eloszlás)

Middleware modell

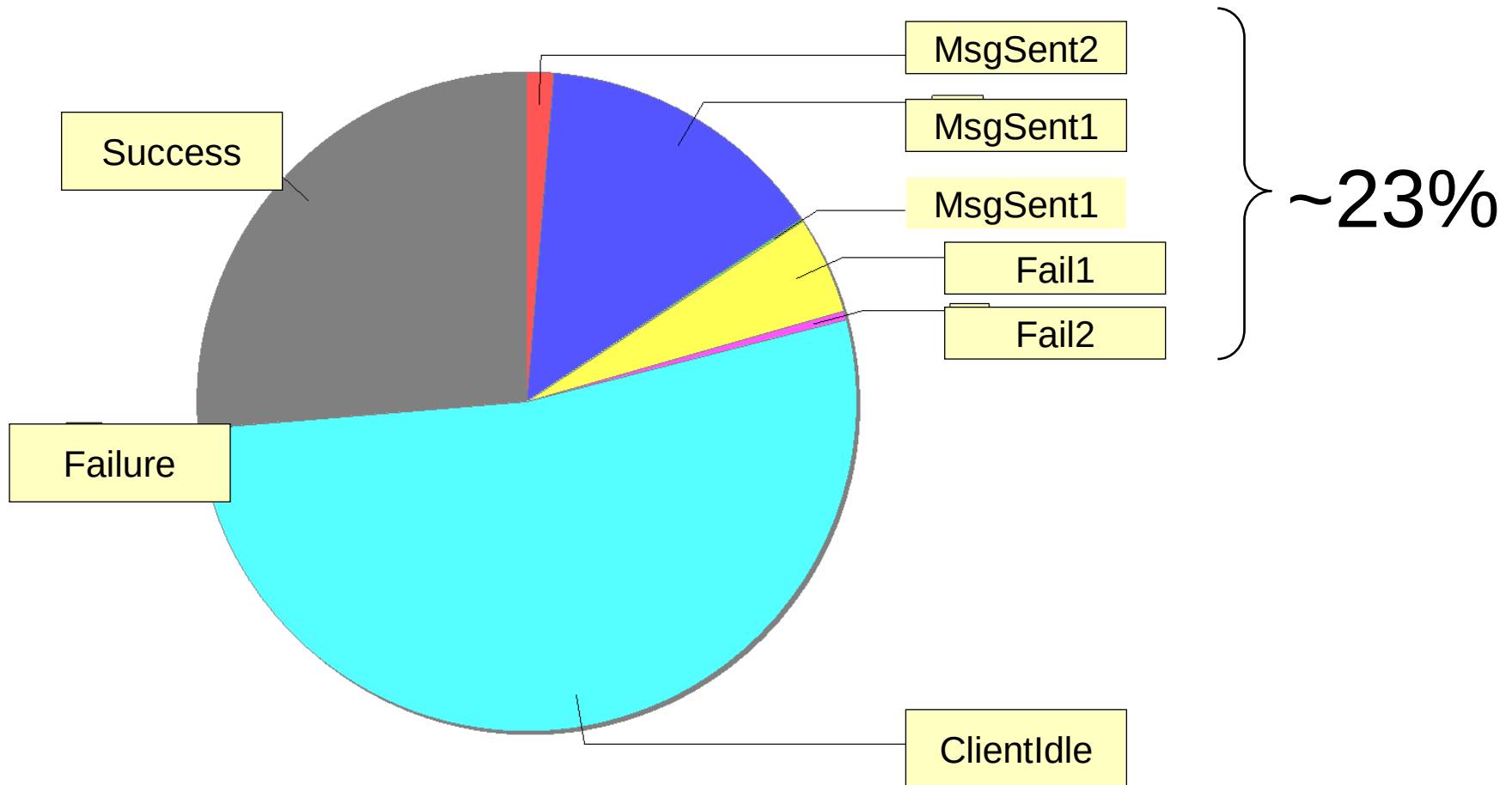
- Platform leíró
- A konkrét konfigurációs modell paraméterezi



Analízis eredmények: Kihasználtság

Állandósult állapot vizsgálata

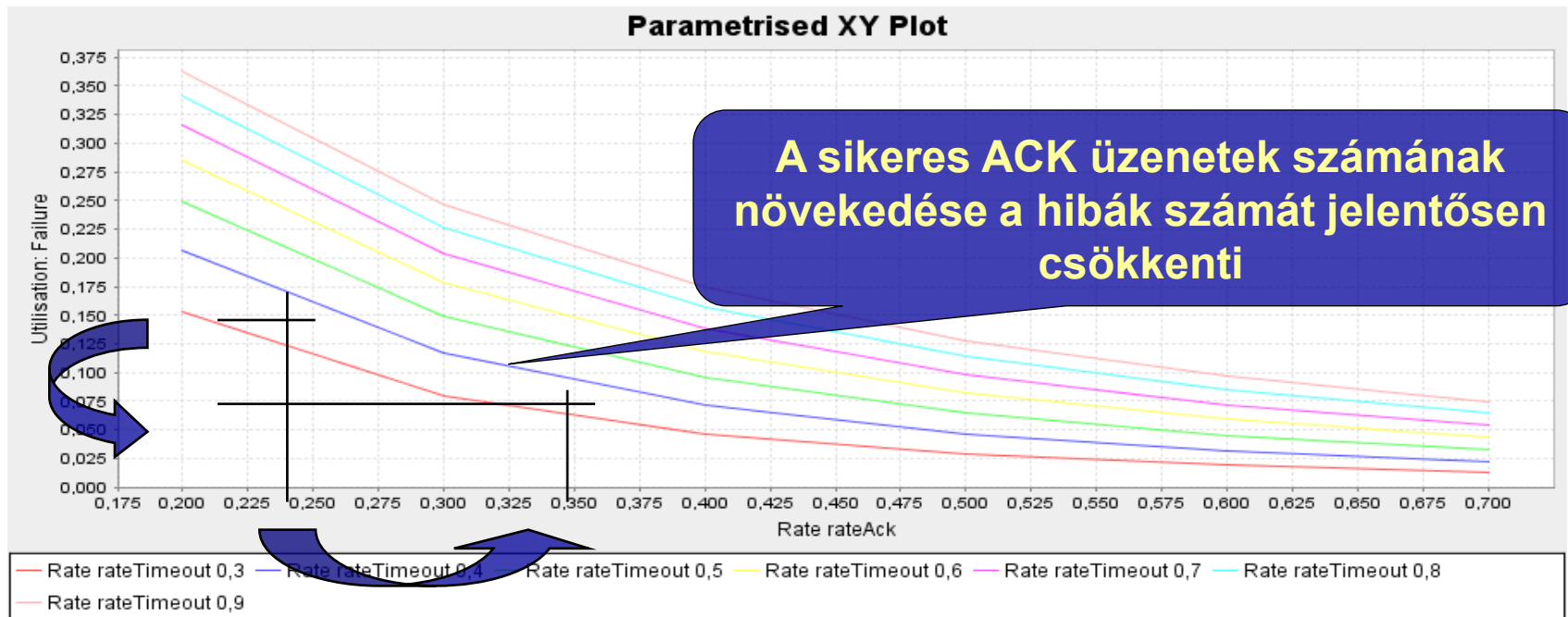
Az idő mekkora része telik a hibák kezelésével?



Analízis eredmények: érzékenységvizsgálat

Érzékenységvizsgálat: hol változtassunk?

Hogyan változik a rendszerszintű hiba valószínűsége a „resend” időzítés változásának hatására?



Források

- <http://www.workflowpatterns.com/documentation/documents/ce-xpdl.pdf>
- <http://www.slideshare.net/dgagne/bpm-standards-what-is-new-in-bpmn-20-and-xpdl-22-bbc-2011>
- http://www.omg.org/bpmn/Documents/XPDL_BPMN.pdf
- <http://www.workflowpatterns.com>
- https://www.inf.mit.bme.hu/sites/default/files/publications/GonczyHegedusVarro_MDDOverview-final.pdf
- <http://mycite.omikk.bme.hu/doc/46406.pdf>