

Üzleti folyamatok Webszolgáltatás alapokon

BPEL

Motiváció

- Pl.: Bank:
 - Ahány beszállító, annyi technológia, módszertan, protokoll...
 - Régi eszközöket soha nem selejteznek le
- Meglévő workflow eszközök
 - Gyártófüggők
(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: ...?

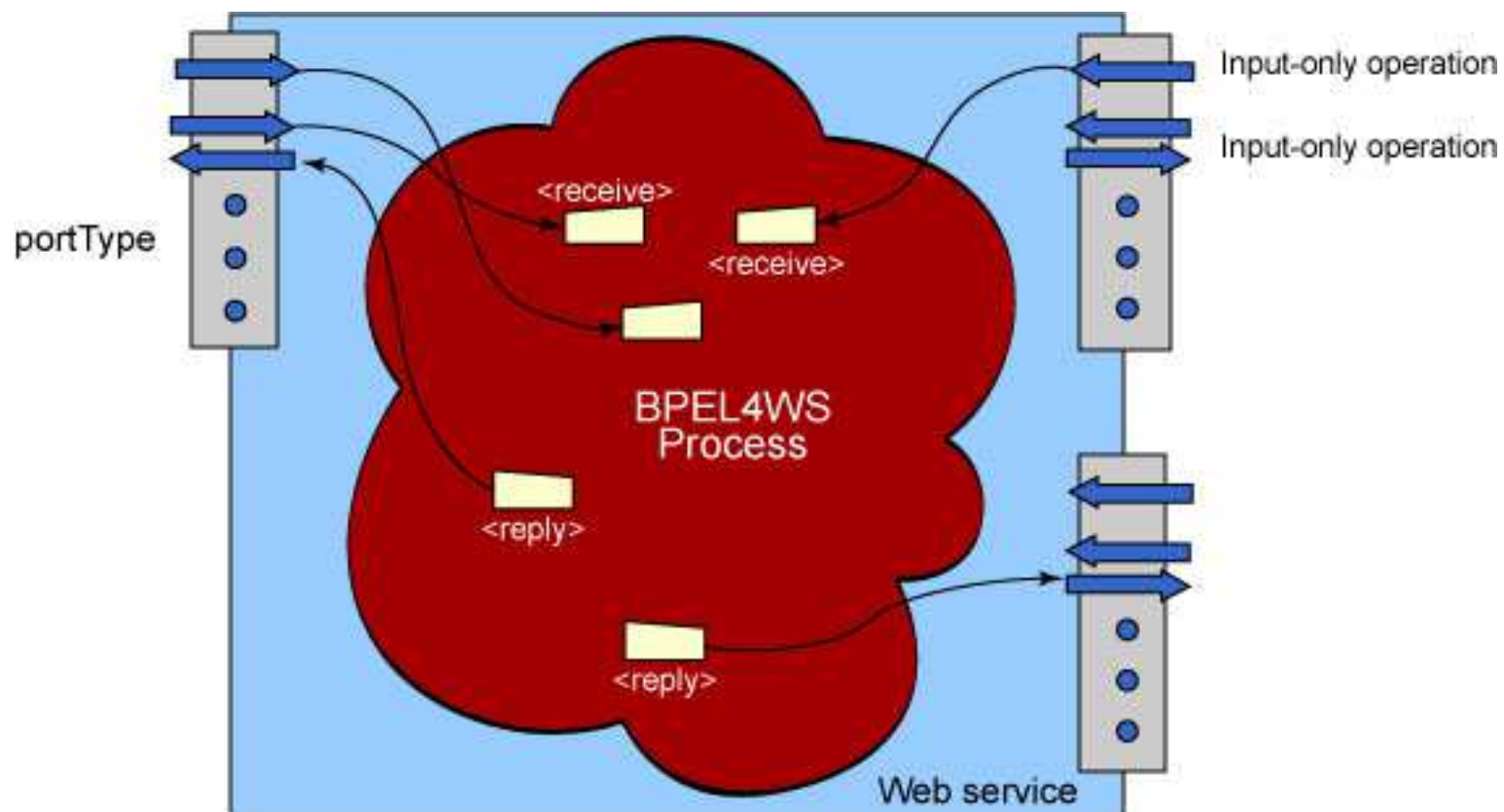
BPEL (BPEL4WS)

- 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



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)

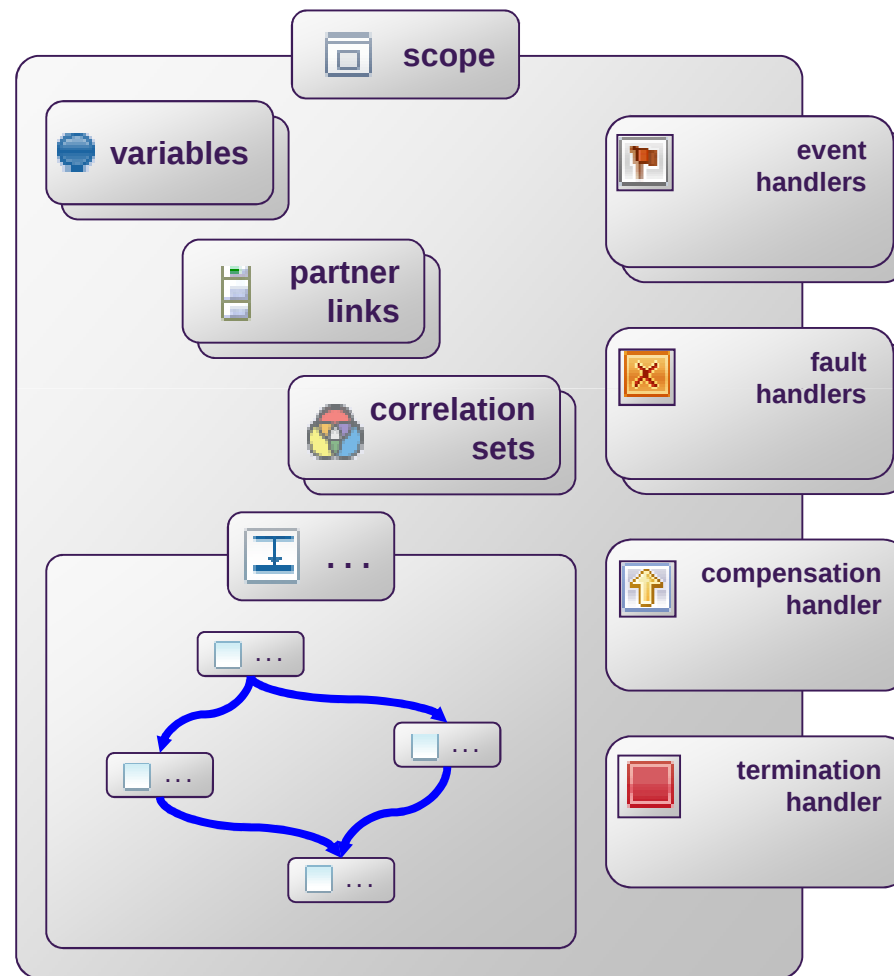
```
<sequence name="EchoSequence">
```

```
  <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>
```

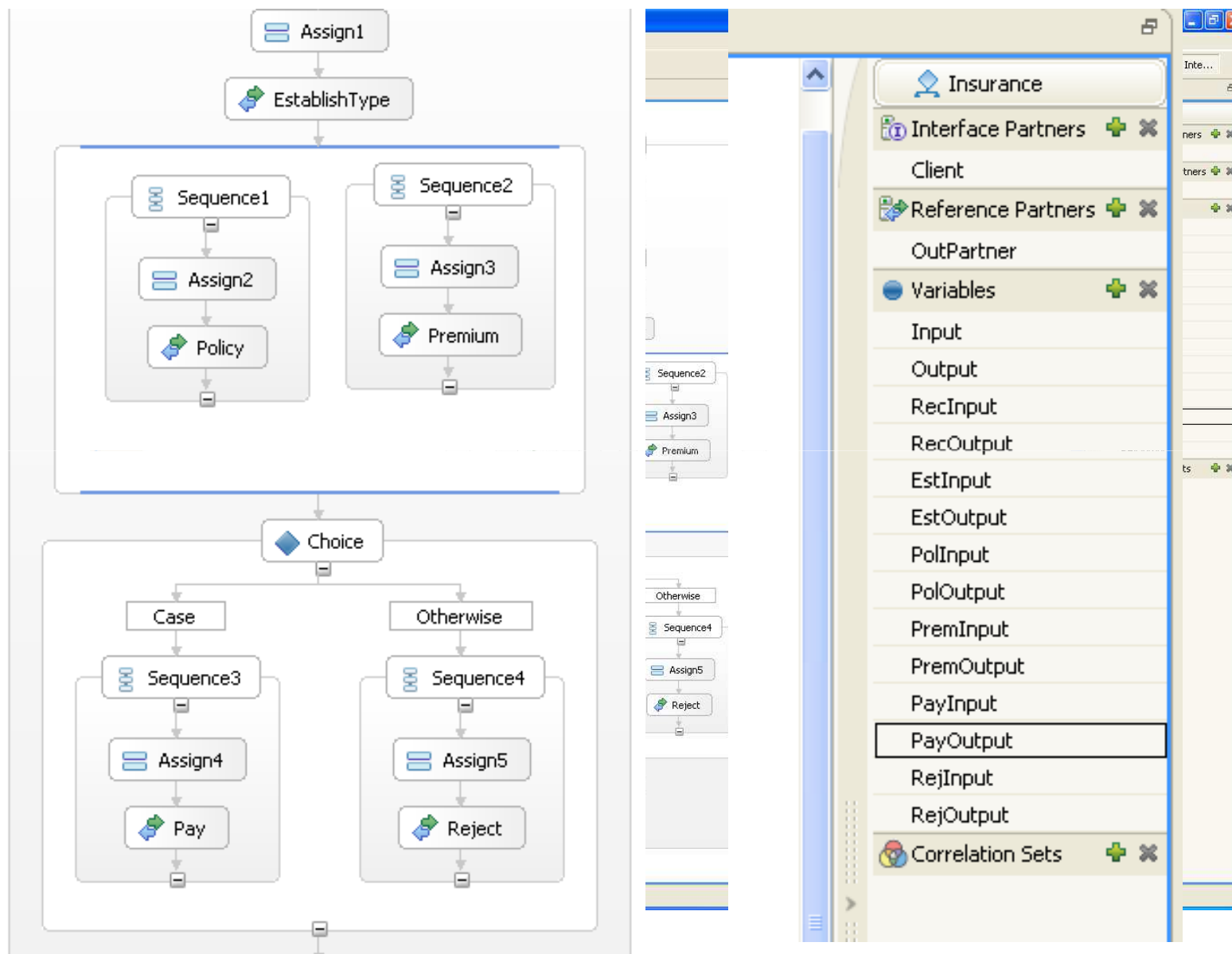
Logikai felépítés



BPEL szerkesztés

- IBM
 - BPEL4WS Eclipse Plug-in
- Oracle BPEL Designer (régen Collaxa)
 - Eclipse
 - JDeveloper
- Más fejlesztőeszközök kimenete
 - WebSphere Business Integrator 5.1, 6.0
 - 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 alkalmazásszerverre

BPEL futtatókörnyezetek

- Business Processes for Web Services Java Runtime (BPWS4J)
 - IBM Java alkalmazás (WebSphere, Apache Tomcat)
- Oracle BPM
- BizTalk Server (2004-es verziótól)
- Open source futtató környezetek
 - Twister
 - ActiveBPEL engine
 - Bexee BPEL execution
 - Eclipse SOA tools Platform – B2J
 - Natív Java készítése

Hibakezelés BPEL-ben

- Scope szintjén
 - Egyedi elem vagy folyamat részlet
- Fault handling
 - Hiba típusokra definiálható
 - Akkor hívódik meg, ha meghíúsul az adott rész
 - Mint Java-ban az exception handling
 - Alulról felfele hívódik meg
- Compensation handling
 - Mi történjen, ha meghíúsul egy tranzakció, de bizonyos részei lefutottak?
 - Erőforrásokat nem lehet zárolni, mert elosztottak, egy folyamat sokáig futhat (long-running transactions)
 - Adatbázis lock vs emberi feladat távoli szerveren
 - Compensation handling definiálható scope szinten
 - Akkor hívódik meg, ha valamiért vissza akarjuk vonni az adott lépések hatását
 - Pl. cancellation email
 - Rekurzívián lefele hívódik meg a hierarchiában
 - A végrehajtó motor felel azért, hogy azok a változó értékek rendelkezésre álljanak, amik a folyamat meghívásakor érvényben voltak
 - Pl. „rendeltem X könyvet” (X=10) → „lemondok 10 könyv megrendelését

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)

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

Modellezési nyelvek kapcsolata

- BPMN → BPEL
 - Szabvány definiál leképezést
- UML-AD → BPEL
 - Létezik hozzá UML profile
 - Szemantika különbözik....
- UML ↔ BPMN
 - Egy BPMN folyamat UML tervezés alapját képezheti
 - Részletesebb nézetek, pl struktúra definíció
 - Számtalan eszköz kezeli mindkét modellt