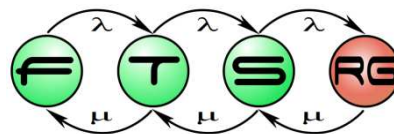


Folyamatmodellezés a gyakorlatban



Business Process Modeling Notation

Business Process Modeling Notation (BPMN)

- Business Process Management Initiative (BPMI)
 - 2004. május: BPMN 1.0 specifikáció
 - 2011: végleges BPMN 2.0
- Célok
 - Közérthetőség
 - Felhasználó
 - Üzleti elemző
 - kezdeti folyamatterv
 - Műszaki fejlesztő
 - Implementáció
 - Belső modell automatikus generálás céljára
 - BPEL4WS
 - Üzleti végfelhasználó (monitorozás, menedzsment)

Business Process Diagram (BPD)

- Folyamatábra-szerű grafikus diagram
- Elemtípusok
 - Adatfolyam
 - Összeköttetés
 - Tagolás
 - Artifact

Adatfolyam

Esemény

Állapotváltozás

Ok-hatás

Eseménytípusok:

Start, Intermediate, End



Tevékenység

Atomi/összetett

Taszk/alfolyamat



Átjáró

Szekvencia

konvergencia/divergencia

AND, OR, XOR, ...



Összeköttetés

Szekvencia Tevékenységek sorrendje a folyamatban
(nincs vezérlési folyamat a BPMN-ben)



Üzenet Két független folyamat
részvevő közötti
információcsere



Asszociáció Adat, szöveg stb.
hozzárendelés



Tagolás

Pool

Résztevő jelölése

Name	
------	--

Sáv

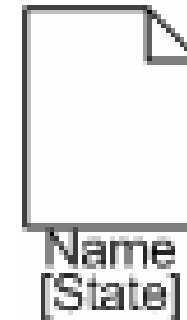
Tevékenységek
csoportosítása

Name	
Name	

Artifact

Adat
objektum

Szimbolikus
token



Csoport

Tevékenységek
csoportosítása

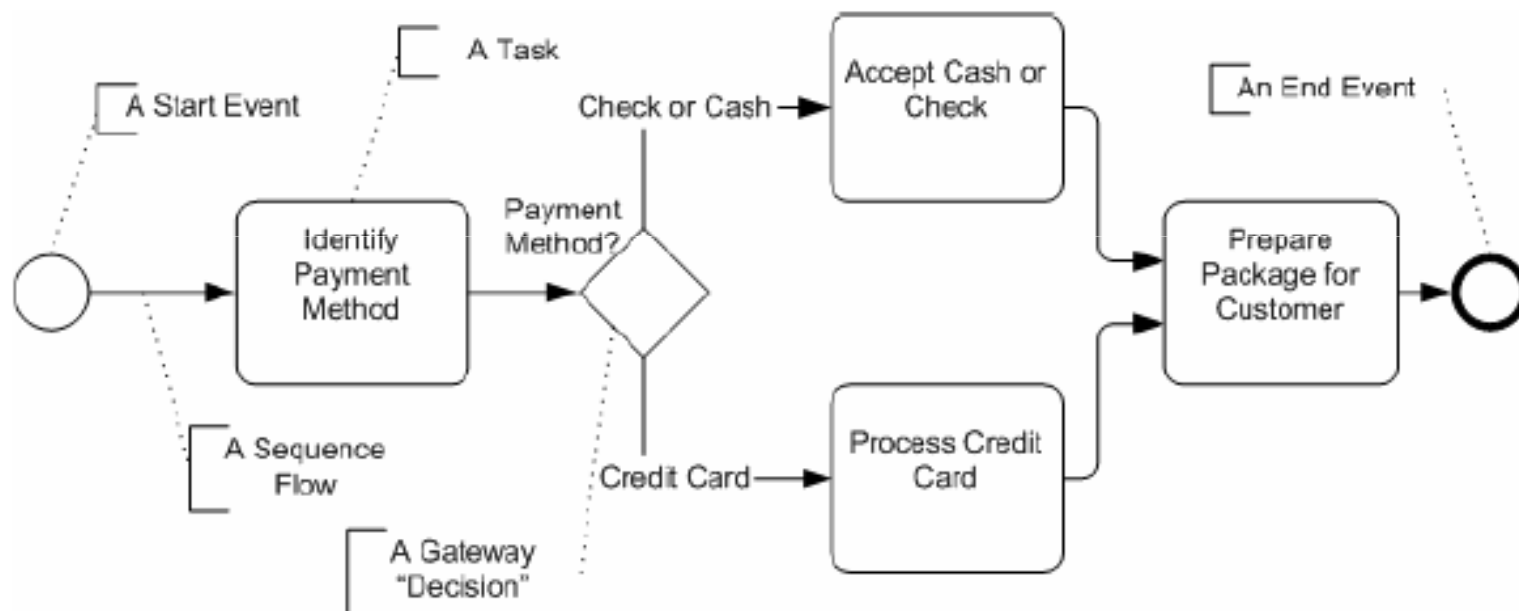


Annotáció

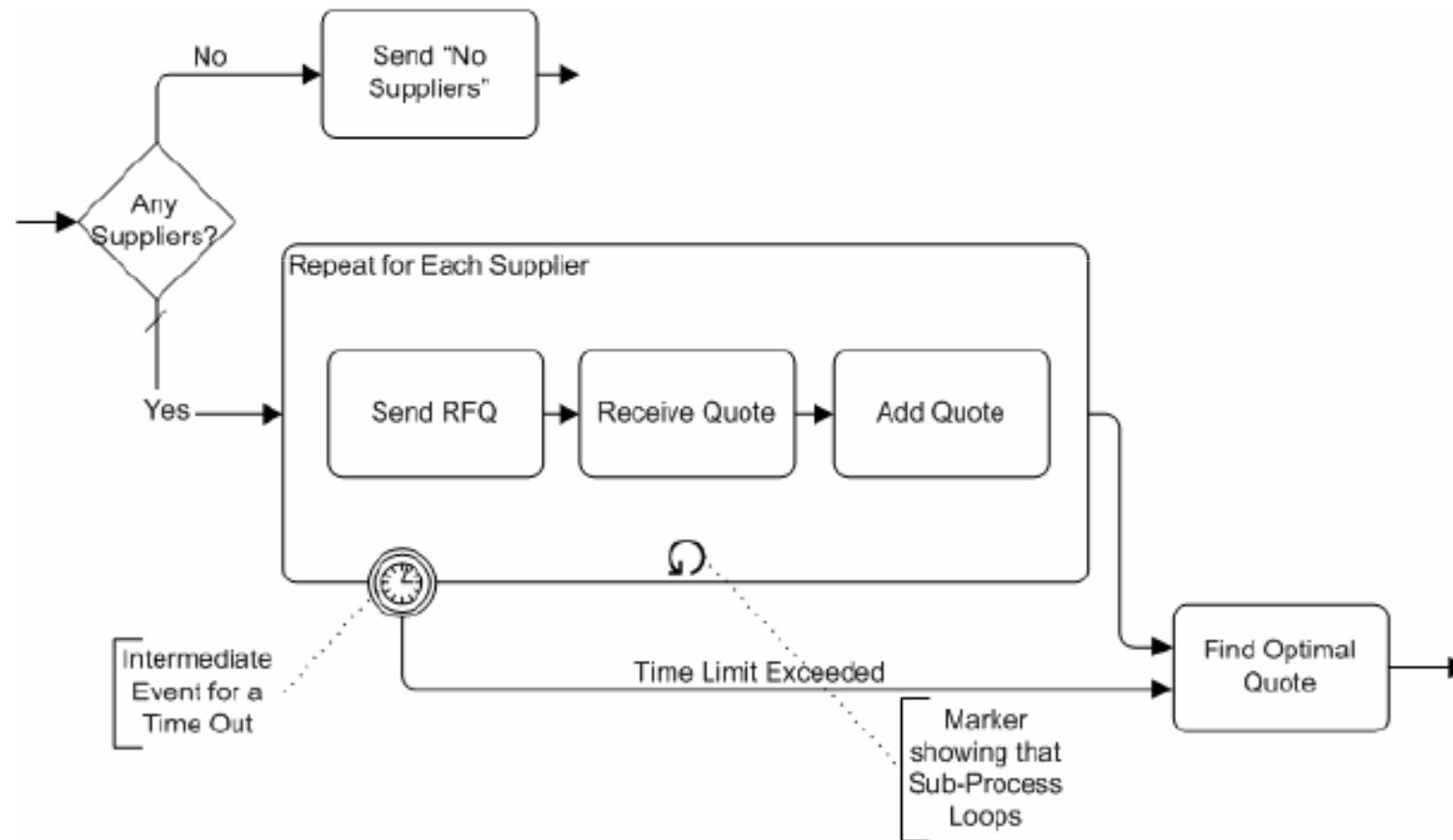
Kiegészítő szöveges
információ
(komment)



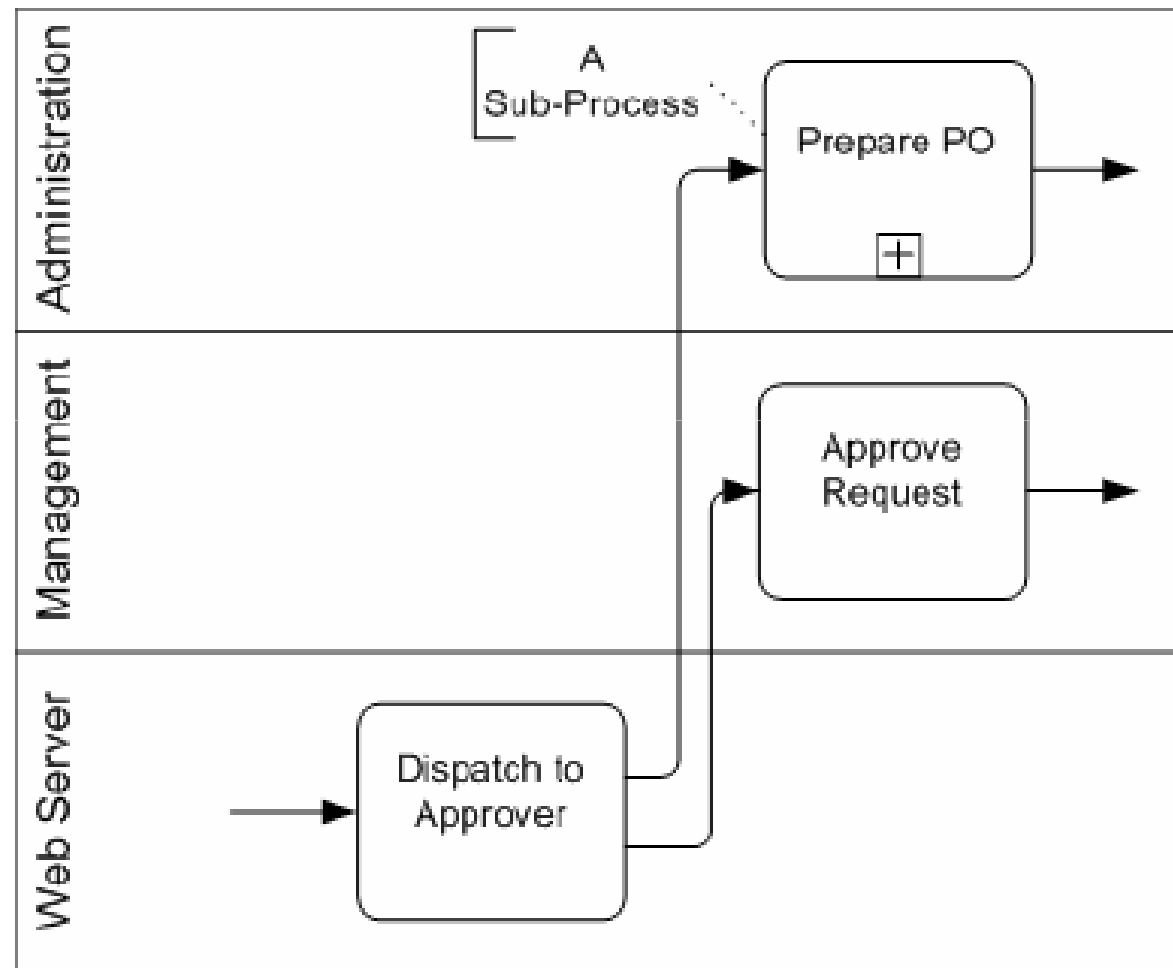
Példa: elemtípusok



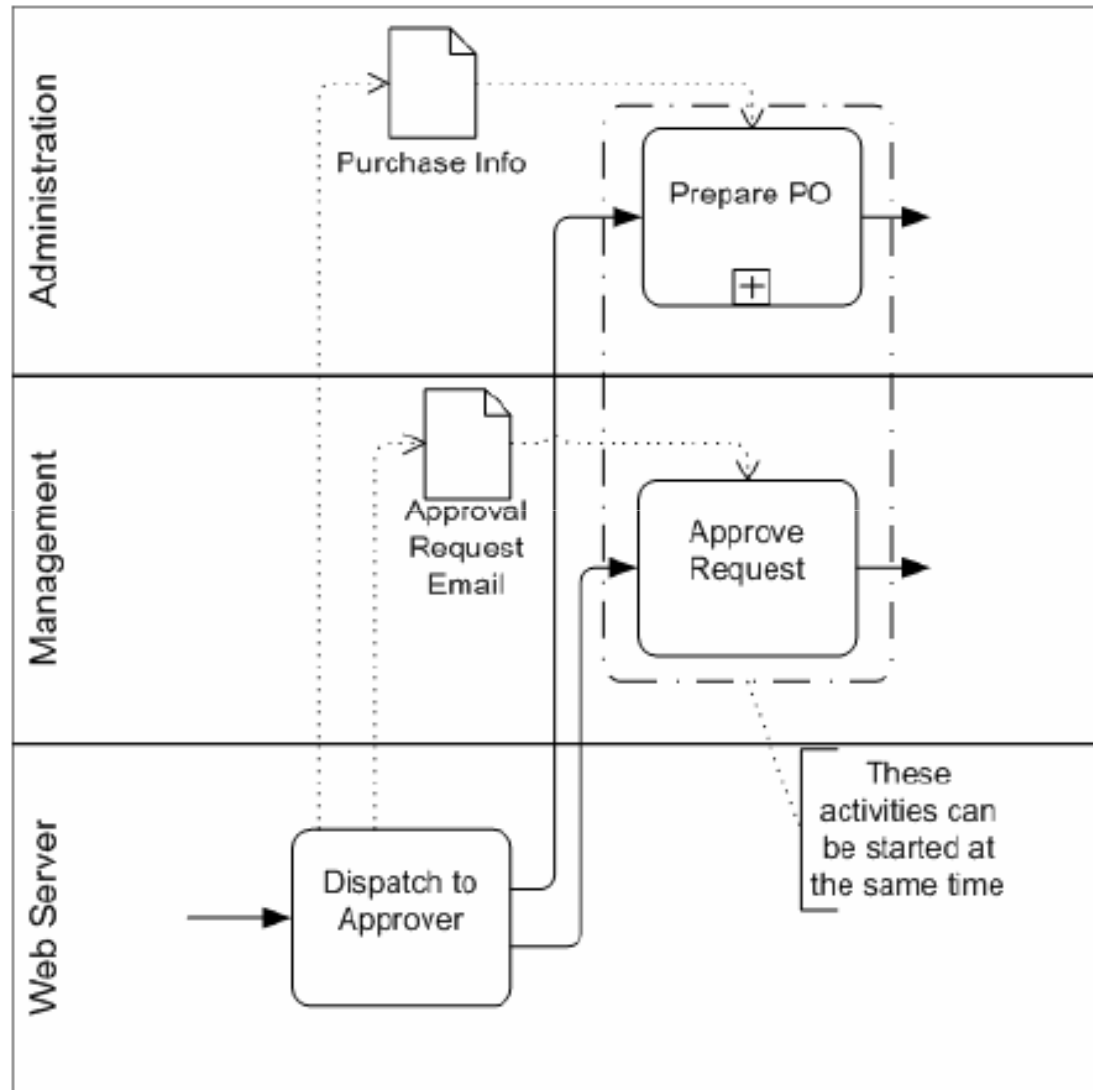
Hierarchikus modellezés



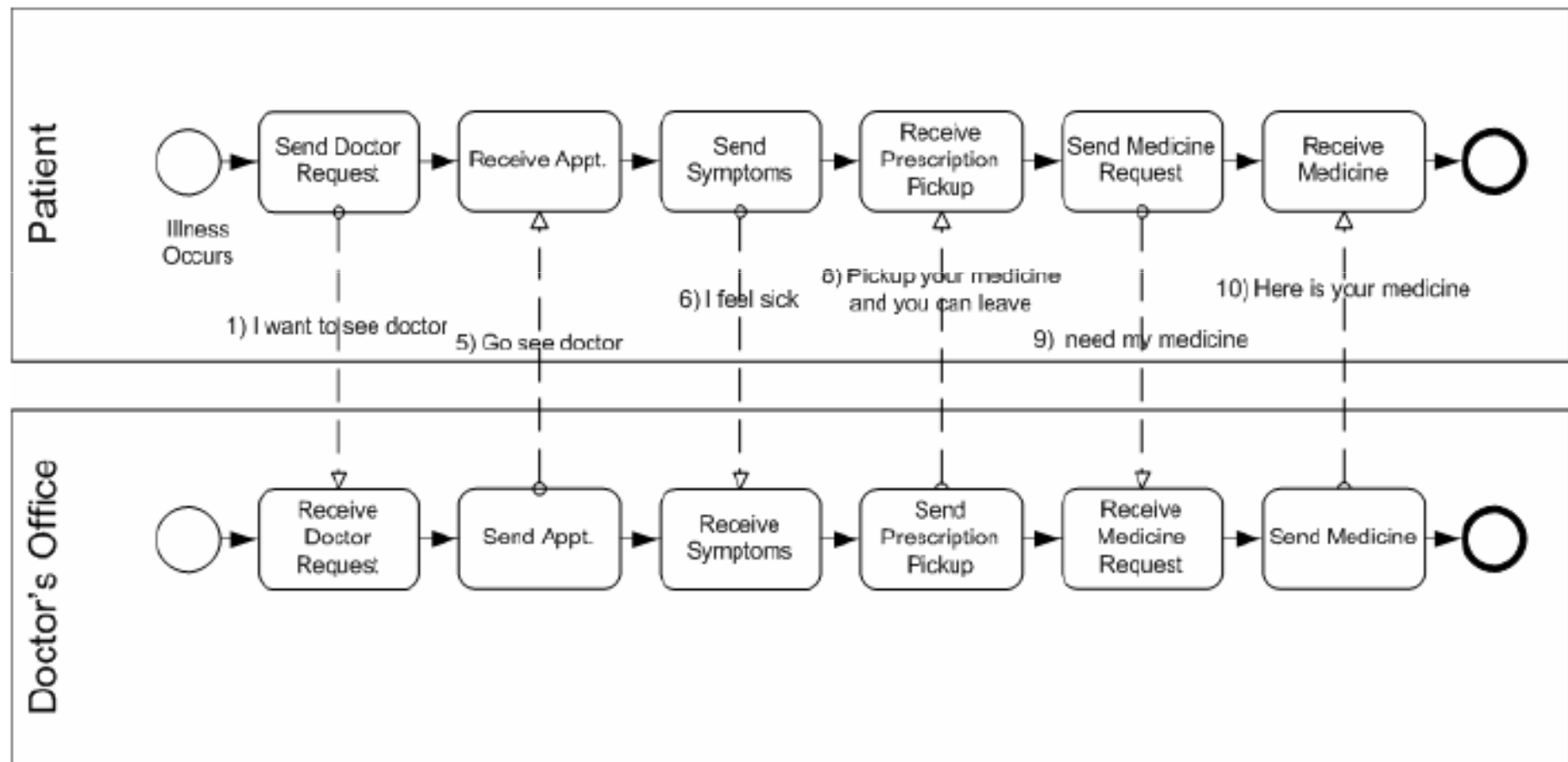
Tagolás



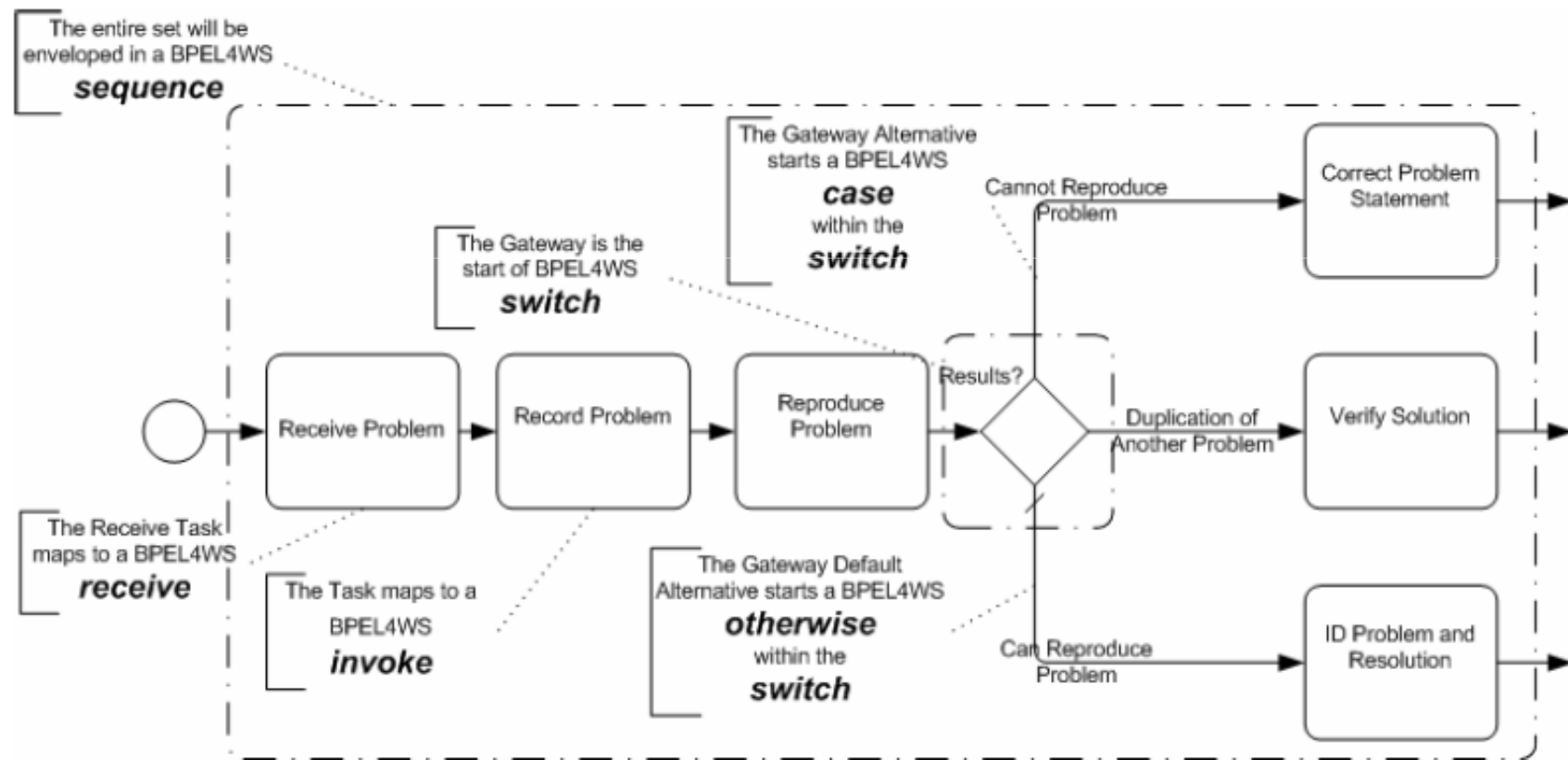
Adat objektumok, csoportok és annotáció



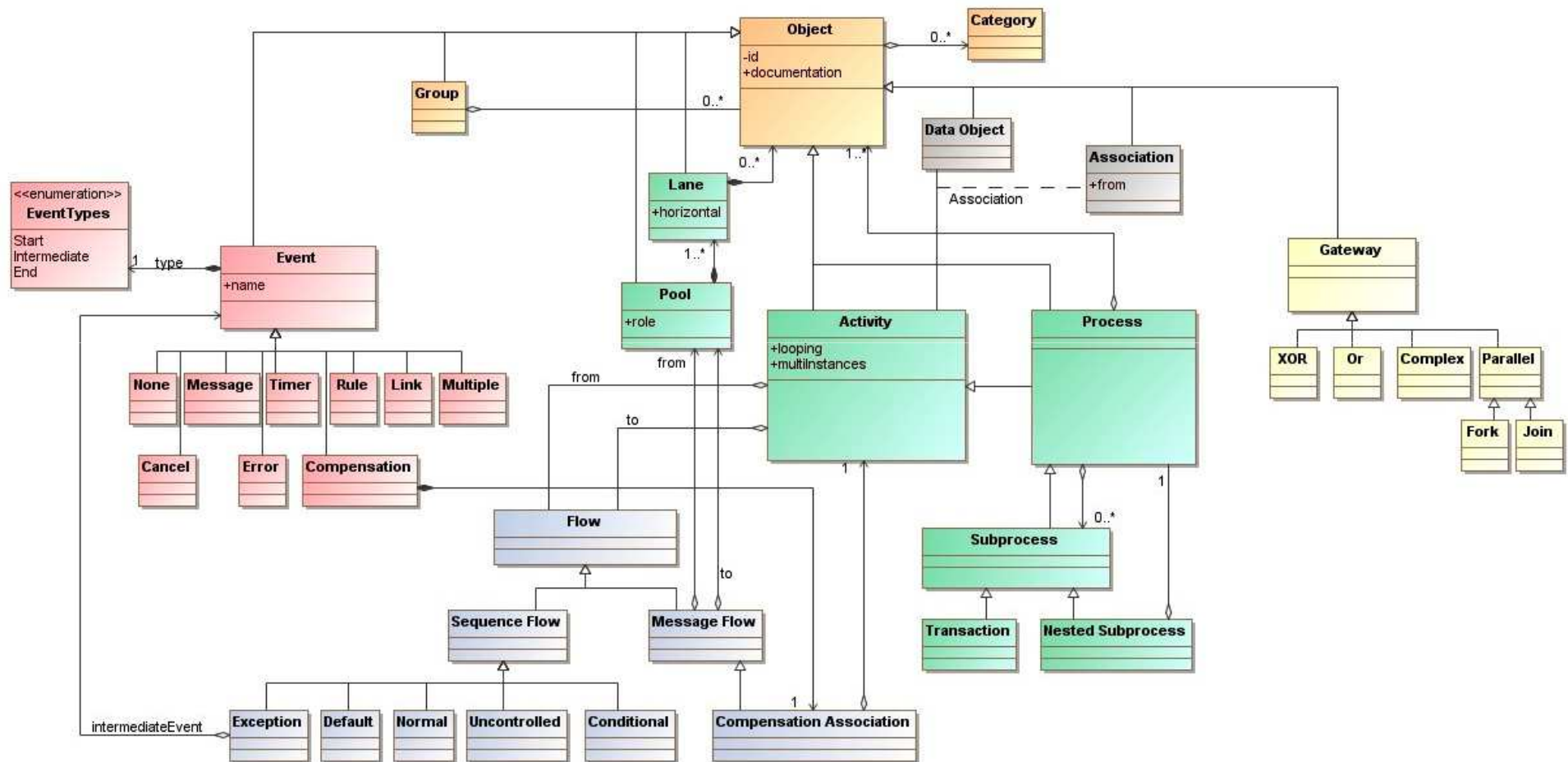
Együttműködés



Automatikus implementáció támogatása

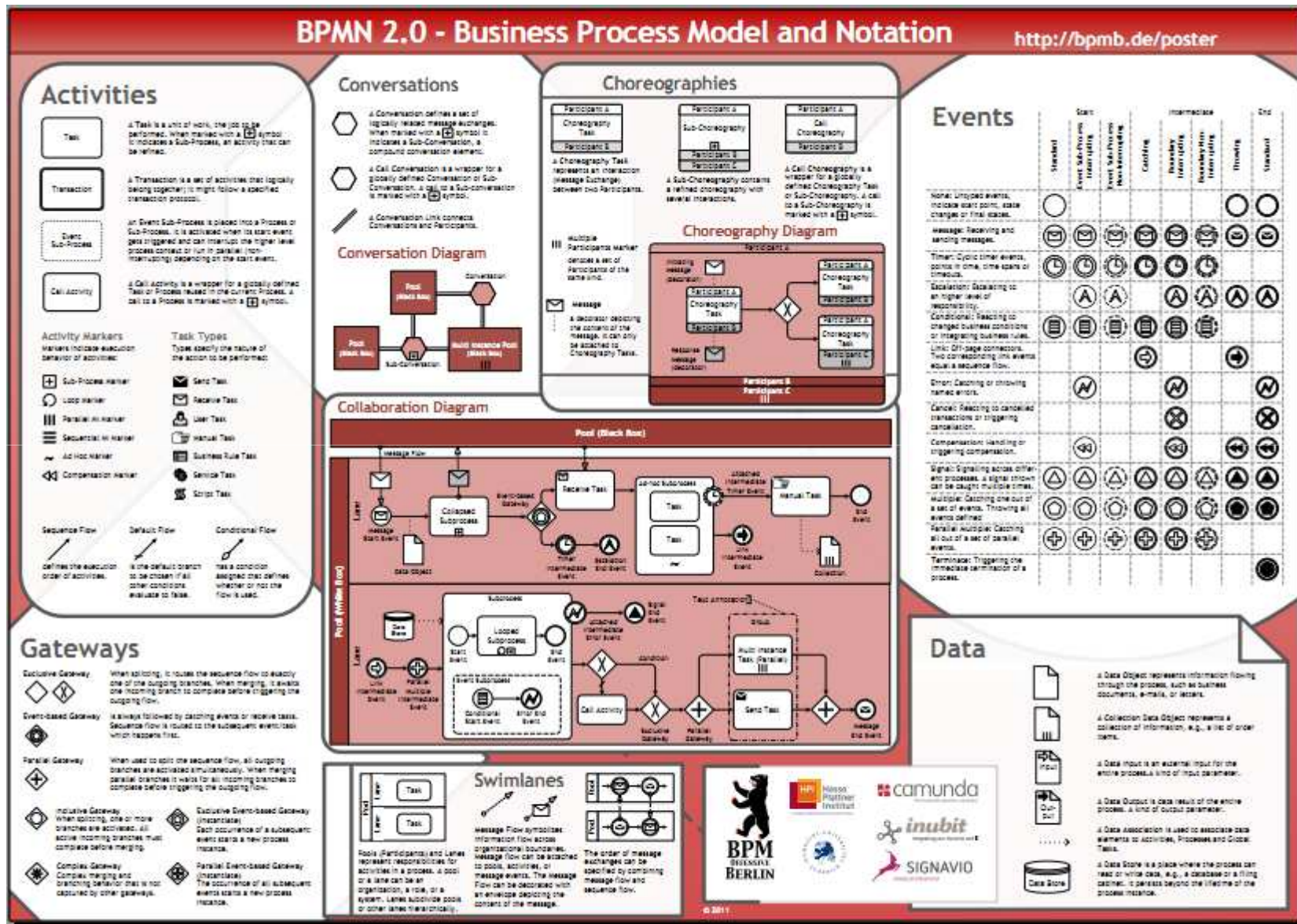


BPMN metamodel

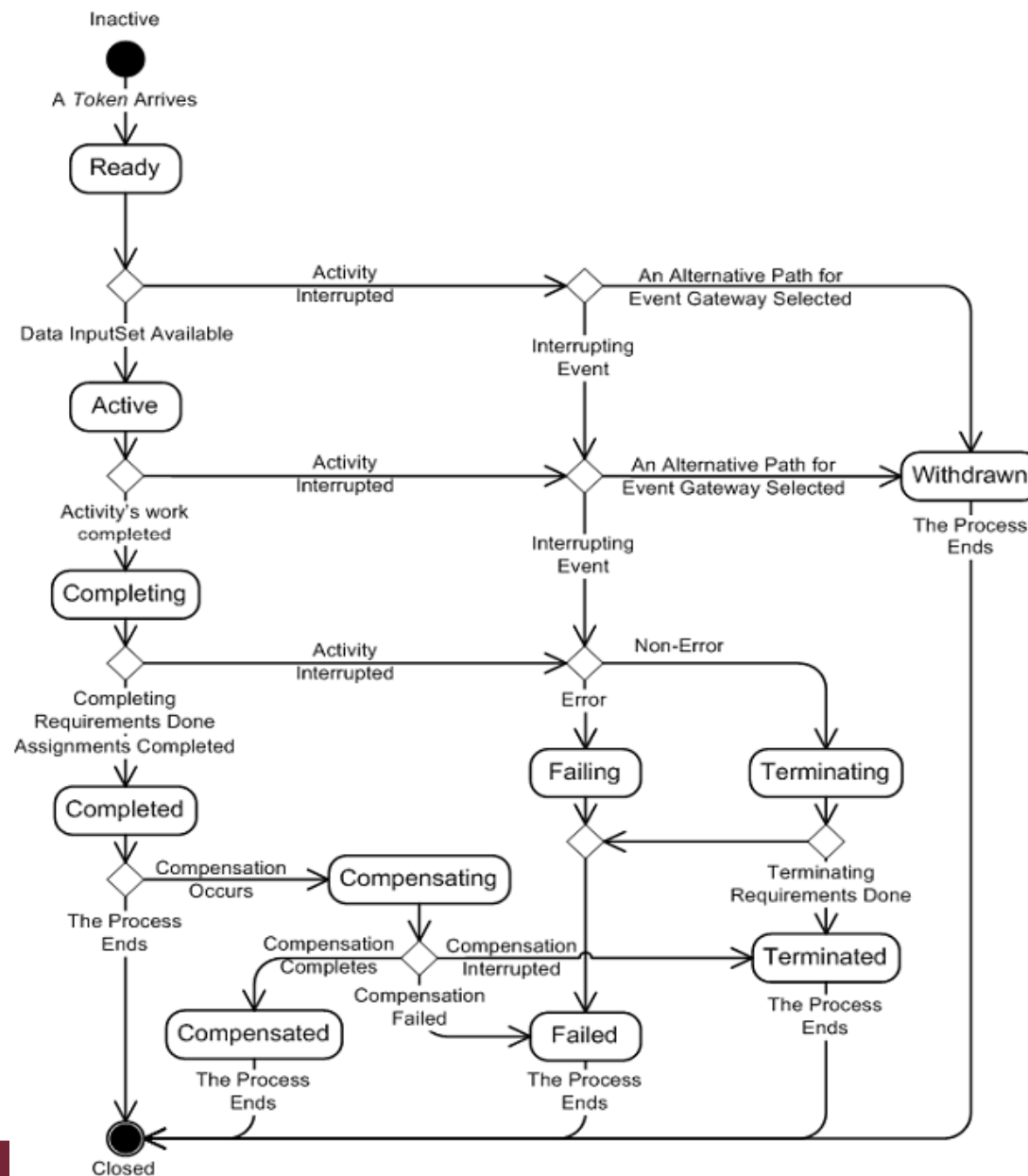


Forrás: <http://www.wsper.org/>

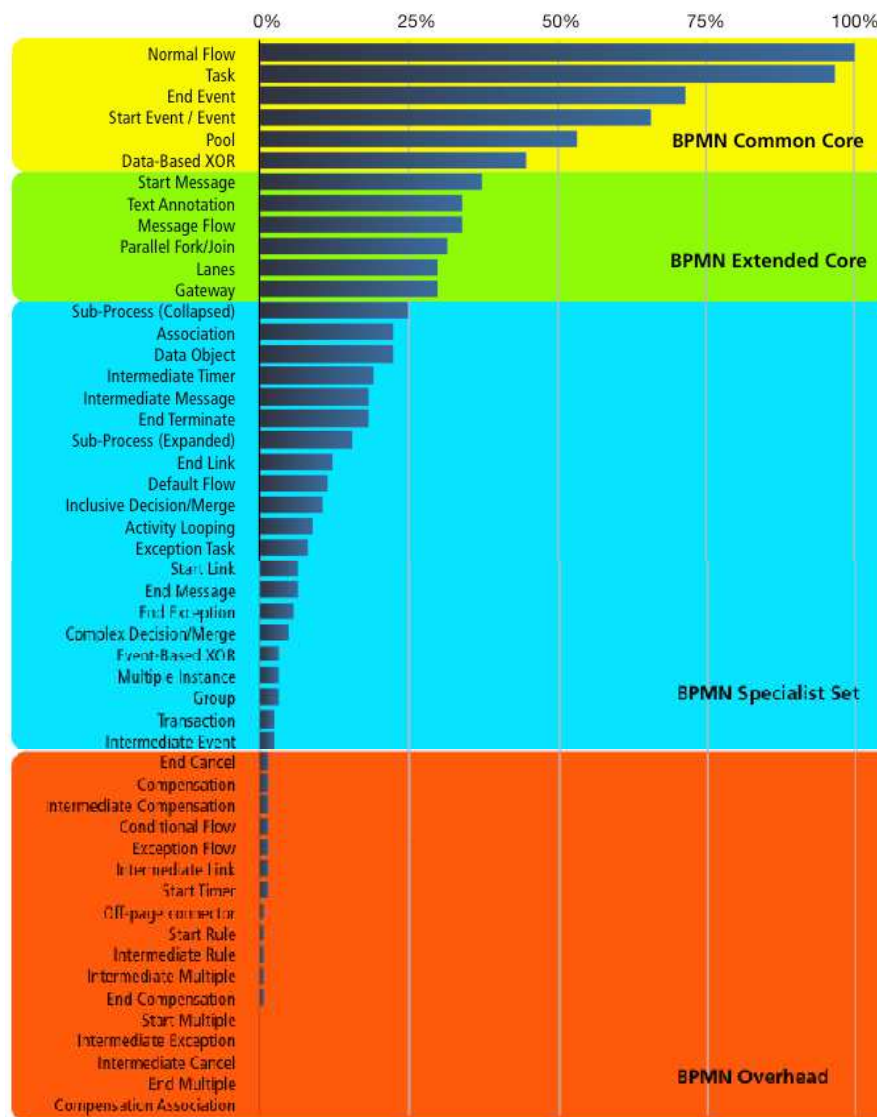
Nyelvi elemek (2.0)



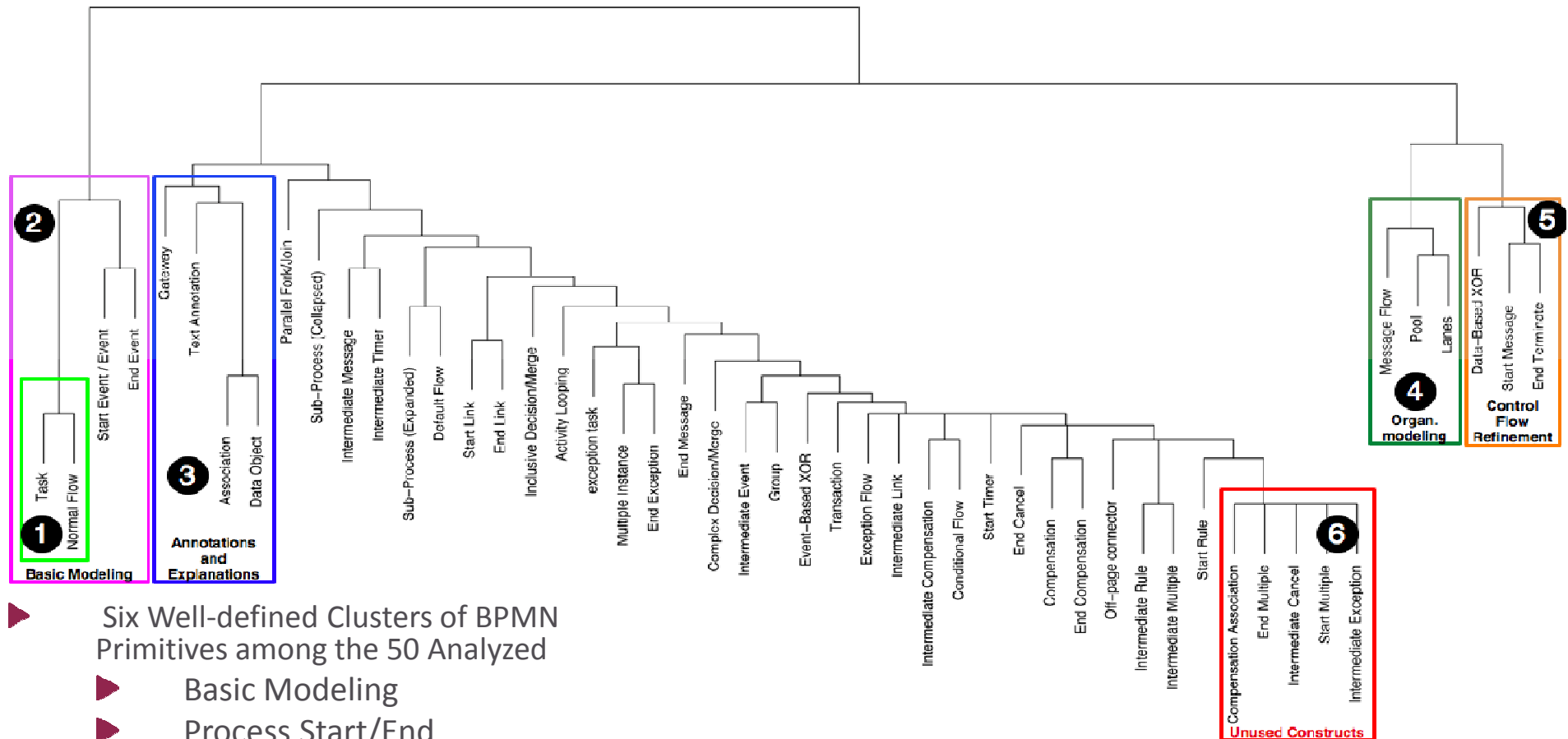
BPMN lépések állapotgépe



Miket használnak?



Csoportosítás



► Six Well-defined Clusters of BPMN Primitives among the 50 Analyzed

- Basic Modeling
- Process Start/End
- Control Flow Refinement
- Organization Modeling
- Annotations and Explanations

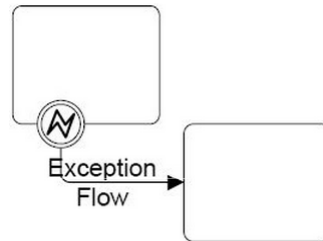
Source: Process Modelling. What Really Matters
Keynote of Michael Rosemann @ UNISCON2009 conference

Mi nem szerepel ezeken a modelleken?

- A végrehajtás és a környezet kapcsolata
- Szerepkörök
- Adatok
- Erőforrások
 - Típusok, példányok
- Végrehajtási idők, feltételek
- Kivételkezelés

BPMN hibakezelés

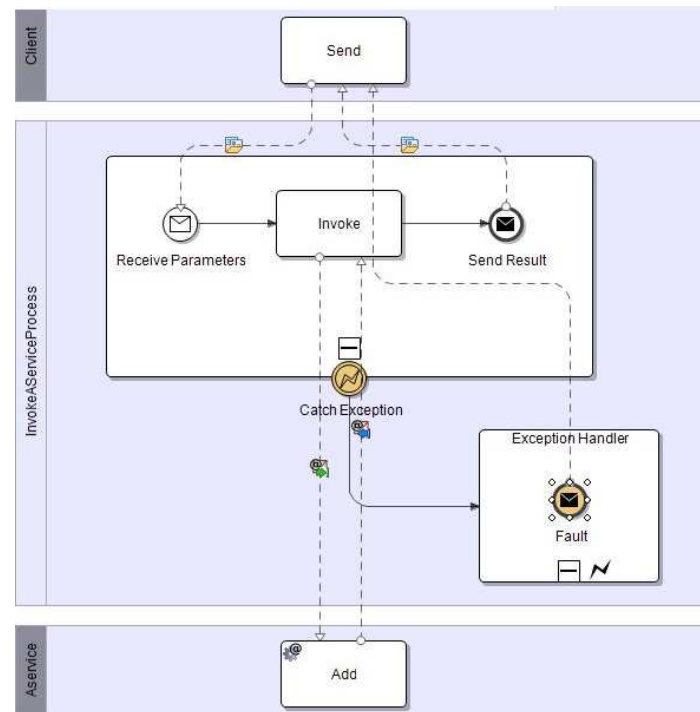
- Készüljünk fel az elemi lépések hibáira
- Belső hibakezelés: a folyamat valamely lépése indítja, bizonyos feltételek esetén



- Megj: a hiba korrigálására „hibatűrő minták” felhasználhatóak
 - Pl. recovery block, NVP alkalmazása

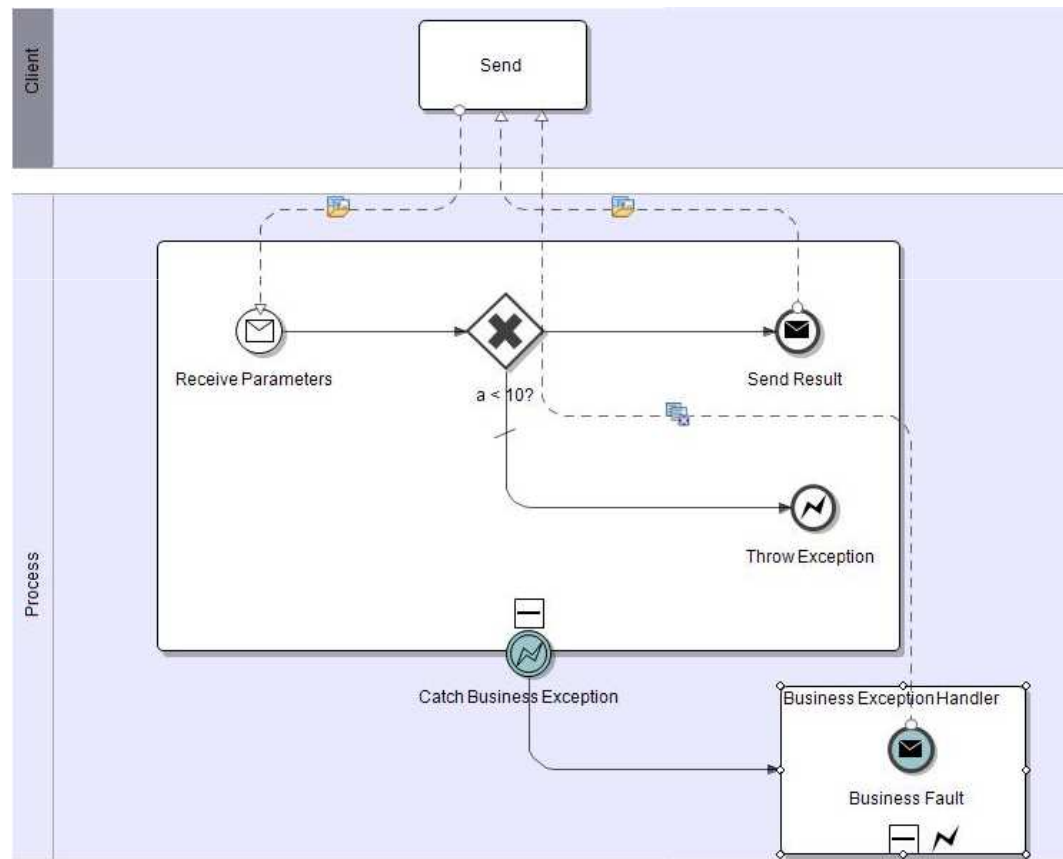
BPMN hibakezelés

- Technikai jellegű hibatípusok
 - Adat, időzítés, elérhetőség, belső hibák, ...
 - Megj: néhány nyelvben ezek elemi lépésekhez definiálhatóak (pl. webszolgáltatásoknál)

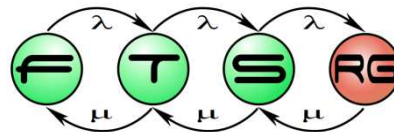


BPMN hibakezelés

- „Üzleti logikai hiba”
- Maga a folyamat ellenőríz bizonyos paramétereket



Végrehajtható munkafolyamatok



Végrehajtható munkafolyamatok

- BPMN
 - Business Process **Modeling Notation**
 - „Csak” modell és jelölésrendszer → nem végrehajtható
- Kérdés: lehet-e **végrehajtható** folyamatokat definiálni?
- Válasz: IGEN!
 - A munkafolyamat továbbra is egy modell
 - De szerkesztési időben forráskódra fordul (pl. Java)
- Elterjedt eszközök: BPEL, jBPM, MS WF

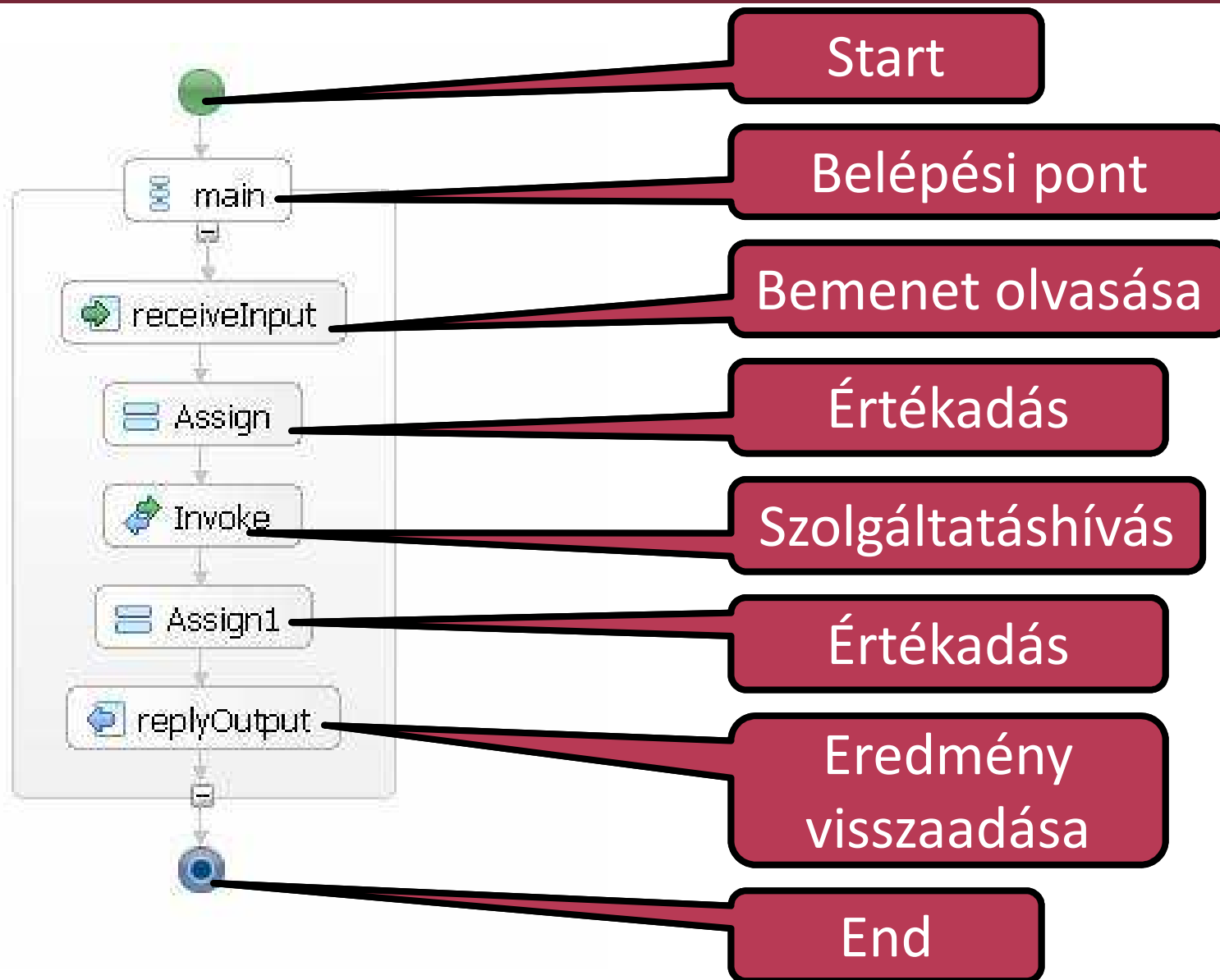
Pro's & con's

- Mi indokolja a folyamat alapú megközelítést?
 - Imperatív programozás esetén is egy munkafolyamat modell alapján fejlesztünk (specifikációban rögzített)
 - Könnyen átlátható (grafikus modellek)
- Mikor ne használjuk?
 - Ha túl nagy a folyamatmodell (nehezen kezelhető)
 - Ha túl egyszerű a funkcionalitás (nincs is valódi WF)
 - Teljesítménykritikus rendszereknél nem mindig jó
- További jellemző problémák
 - Nehezebb tesztelni és debugolni

BPEL

- Microsoft és IBM fejlesztés
- Távoli webszolgáltatásokat feltételez
- XML alapú
 - Itt „hangoljuk össze” a komponensek viselkedését (orchestration)
- Rendelkezik grafikus modellező felülettel
 - De végeredményben ezzel is az XML-t szerkesztjük
 - Intuitív elemkészlet: értékadás, ciklusok, WS hívás...
- A végeredmény mindig végrehajtható
 - Telepítés („deployment”) után hívhatja az alkalmazás

Egyszerű BPEL mintafolyamat

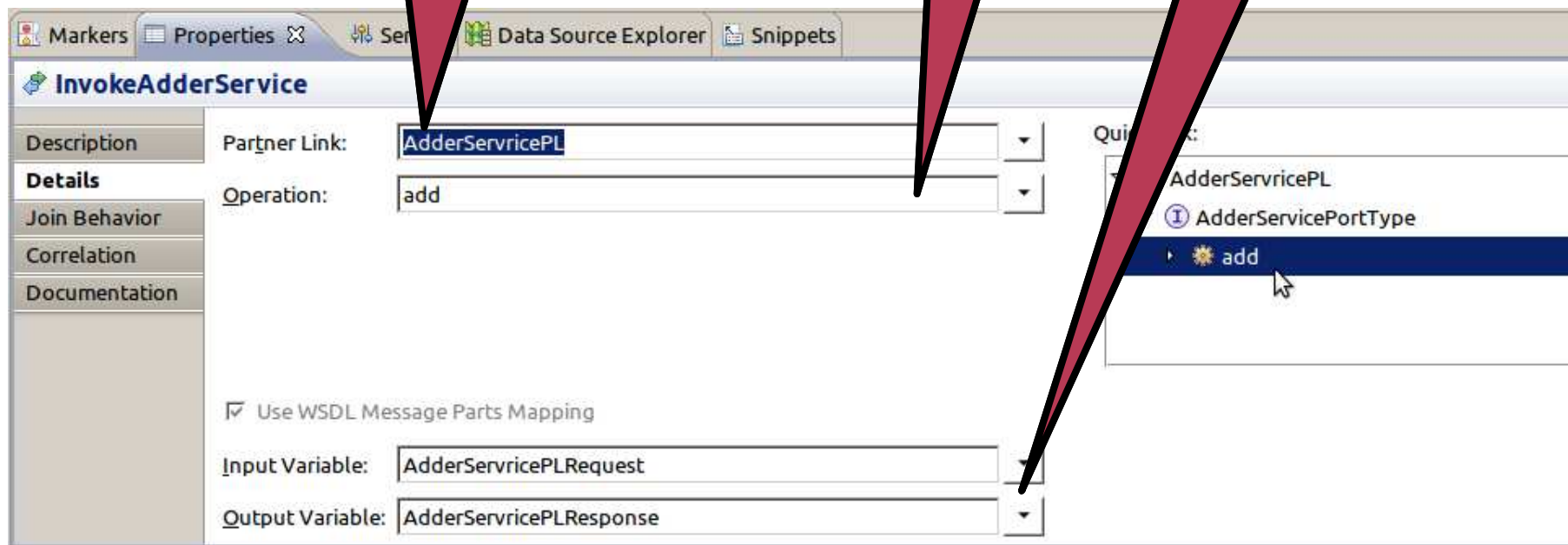


Példa: az „Invoke” elem paraméterezése

Partner link (a szolgáltatásra mutat)

WS metódus

Paraméterek



jBPM5

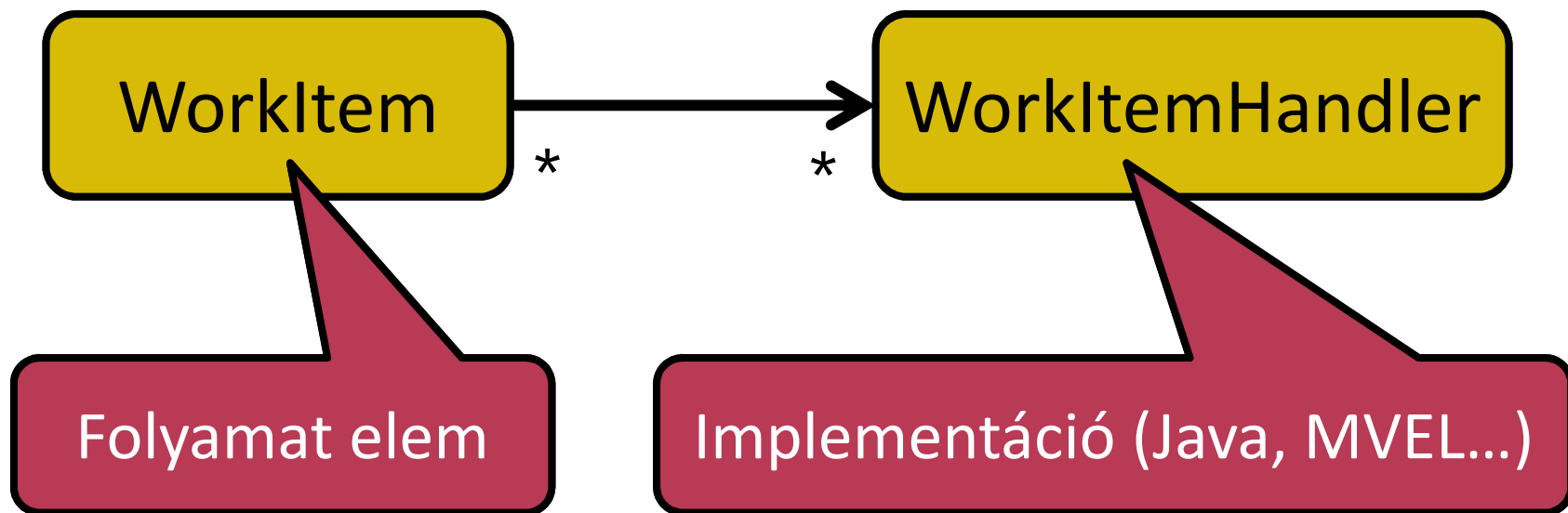
- RedHat fejlesztés
- A BPMN2.0 szabványra épül
- „Full lifecycle management”
 - A keletkezéstől a monitorozásig bezárólag
- XML alapú
- Rendelkezik grafikus modellező felülettel
 - De végeredményben ezzel is az XML-t szerkesztjük
- A végeredmény mindig végrehajtható
 - Telepítés („deployment”) után hívhatja az alkalmazás

jBPM5 filozófia

- Szakterület(Domain)-független megközelítés
 - Jellemzően általános elemek: elágazások, script node...
 - A fejlesztő a „problémára szabja” az elemkészletet
- WorkItem
 - A BPMN2.0 szabvány Service Task elemét valósítja meg
 - Ha a folyamatban egy ilyenre futunk, a hozzá kapcsolt WorkItemHandler végzi el a feladatot
 - A WorkItemHandlert regisztráljuk, az a toolbarra kerül
 - Ezután már egyszerűen használható a modellezésben
 - Előny: valóban „custom” funkcionalitás

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**

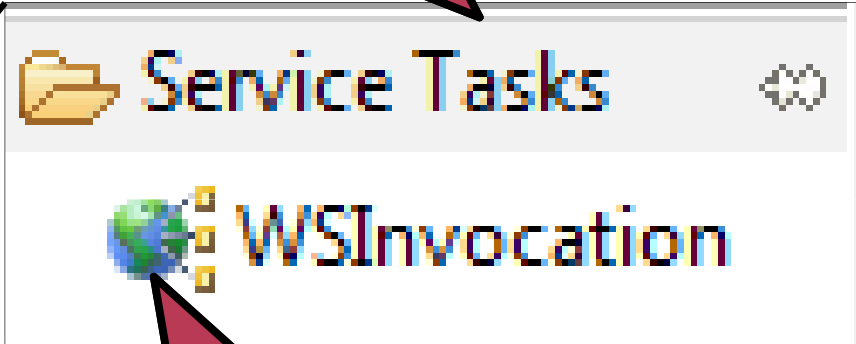
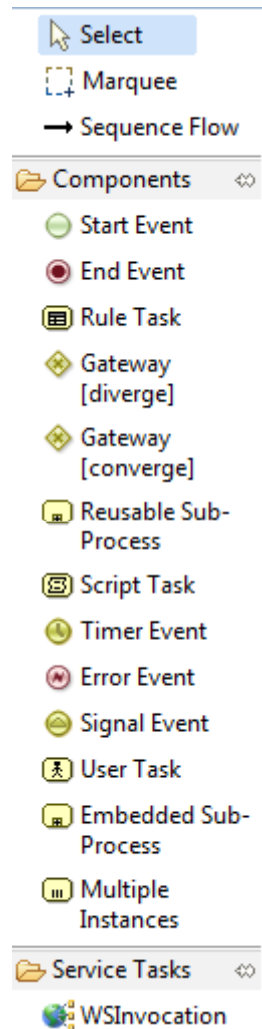
```
ksession.getWorkItemManager()  
    .registerWorkItemHandler(  
        "WSInvocation",  
        new WSInvocationWorkItemHandler()  
    );
```

- **WorkItem regisztrálása a toolbarra**

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

jBPM5 WorkItem példa

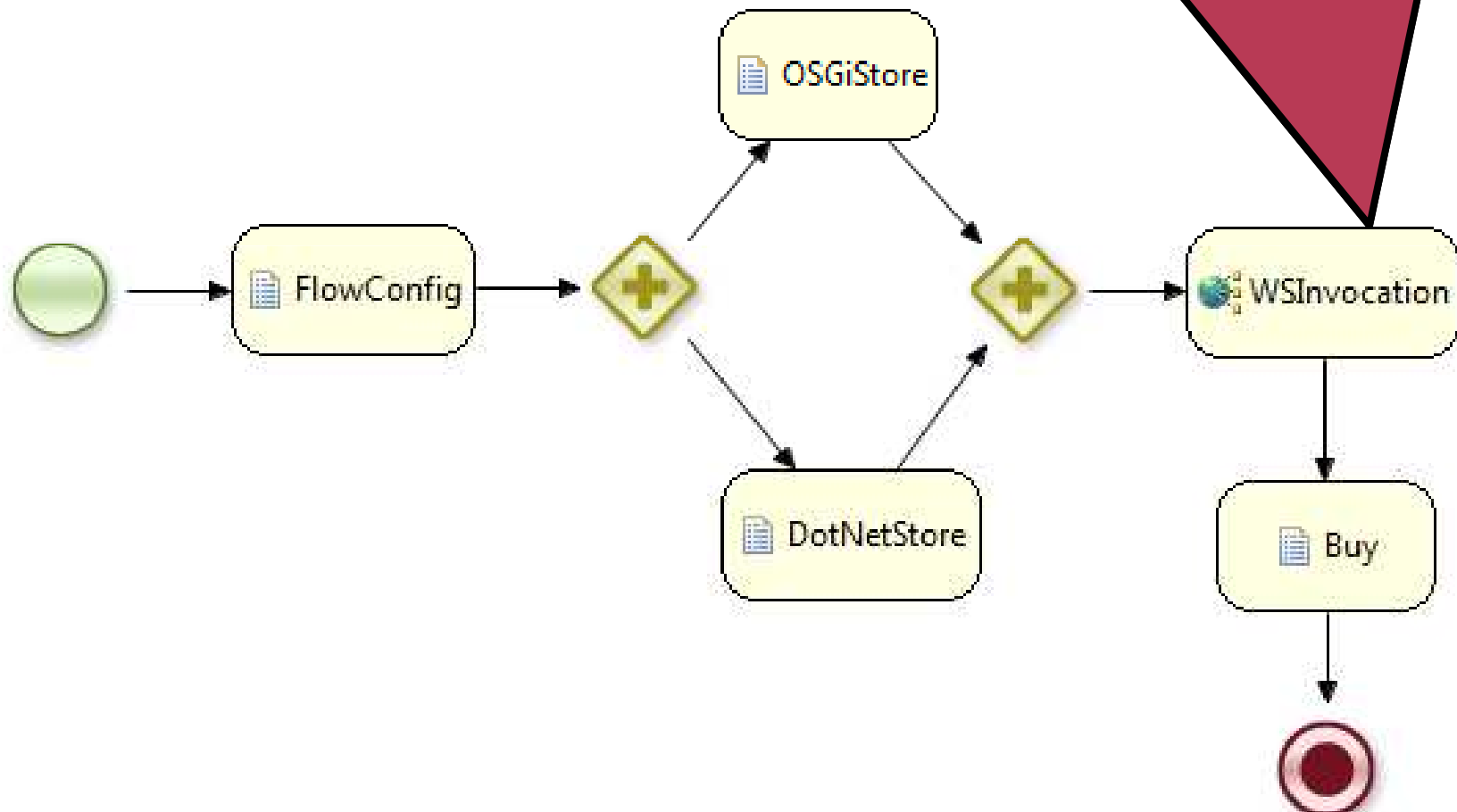
A palettán Service Task
elemként megjelent a
WSInvocation



Még az ikon is custom.

jBPM5 WorkItem példa

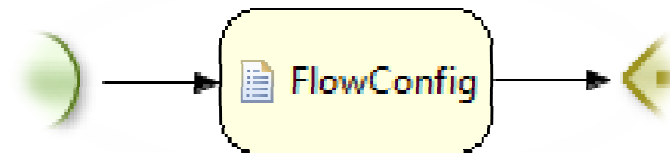
Az új elem beilleszthető a folyamatba. Ha ide érünk, meghívódik a megfelelő Handler.



„What's in the box?”

- Semmi meglepő: egy halom XML
- Példa: a folyamat első eleme (FlowConfig):

```
<scriptTask id="_9" name="FlowConfig" >  
  <script>  
    kcontext.setVariable("wsaddress",  
                          "http://localhost:1047/Services");  
    kcontext.setVariable("wsmessage", "getprice/1984");  
  </script>  
</scriptTask>
```



BPMN tervezőeszközök

- **jBPM Designer**
 - Eclipse BPMN
 - Tibco Business Studio
 - Intalio Designer
 - BPMN Composer
 - BPMN Designer
 - Bonita Open Solution
 - Adonis
 - Activiti
 - Obeo Designer
- + számtalan nem BPMN specifikus eszköz

Kihívások

- Szakterületspecifikus tudás reprezentációja
 - Minták, könyvtárak, szótárak
 - Hatékony modellezés
 - „Web2.0” kapcsolat
- Modell konzisztencia
 - Folyamatmodellek (verziók, együttműködés)
 - Folyamatok / szervezeti felépítés / IT erőforrások
- „Compliance”
 - Rugalmas tervezés mellett megfeleléség biztosítása

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

Egyéb források

- <http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/609cb540-3ca6-2a10-60a7-dc470a9b7adf>
- <http://community.intalio.com/tutorials/exception-handling.html>
- <http://www.conradbock.org/bock-bpmn-2-business-process-semantics-web.pdf>
- Stephen A. White (IBM): *Introduction to BPMN*