

Dr. Mileff Péter

## INTEGRÁLT SZOFTVER RENDSZEREK ÉS MINŐSÉGBIZTOSÍTÁSUK

SOA ÁTTEKINTÉS  
Miskolci Egyetem  
Általános Informatikai Tanszék

## SOA alapjai

- Service-Oriented Architecture, SOA
- Az elosztott rendszerek fejlesztésének módja.
- **Célja:** az IT eszközök komplexitásának a kezelésének egyszerűsítése
  - könnyebben újrafelhasználhatóság, egymással integrálhatóság
  - fejleszthetővé tesz, anélkül, hogy a ráépülő üzleti folyamatokat megzavarná.
- A rendszerek komponensei különálló szolgáltatások:
  - a szolgáltatások földrajzilag egymástól távol elhelyezkedő számítógépeken futhatnak.
- **Egy szolgáltatás lényege:**
  - a szolgáltatás biztosítása független a platformtól, az implementációs nyelvtől és a szolgáltatást igénybe vevő alkalmazástól.

2

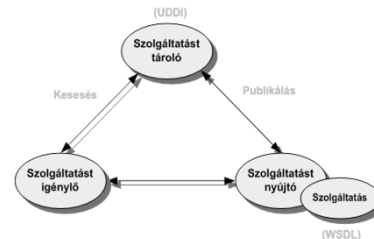
## SOA alapjai

- **Szolgáltatás:**
  - lazán csatolt, újrafelhasználható szoftverkomponens,
  - olyan diszkrét funkcionalitást zár be, amely elosztható és programozói eszközökkel elérhető.
- **Kialakulásának oka:** az internet rohamos terjedésével szükség volt elosztott modellekre.
  - Pl.: RPC, RMI, CORBA, ICE, Webszolgáltatások, stb
- **Webszolgáltatás:** olyan szolgáltatás, amely szabványos internetes és XML-alapú szabványokkal érhető el.
  - az információk elérhetővé válnak más programok számára is, egy webszolgáltatás-interfész definiálásával és publikálásával.
  - Ez az interfész definiálja az elérhető adatokat, és megadja, hogy hogyan lehet azokhoz hozzáférni.

3

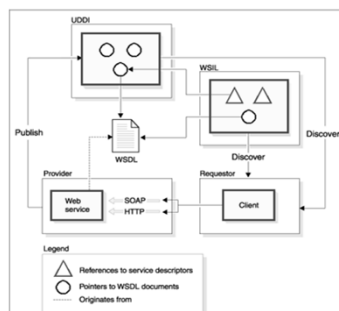
## SOA alapjai

- **SOA modell:**



4

## SOA alapjai



5

## SOA működése általánosan

### I. Szolgáltatások létrehozása:

1. A szolgáltatások gyártói megtervezik és implementálják a szolgáltatásokat.
2. WSDL nyelven specifikálják azokat
3. A szolgáltatásokról információkat publikálnak egy általánosan elérhető tárolóban
  - Erre az UDDI publikációs szabványt használják legtöbbször

6

## SOA működése általánosan

### II. Szolgáltatások igénybevétele:

- A szolgáltatások igénybe vevői (szolgáltatás kliensek), keresést hajtanak végre az UDDI-tárolóban,
  - Így megkapják a szolgáltatás nyújtóját is.
- Az alkalmazások összekapcsolódnak az adott szolgáltatással
  - Kommunikálnak azzal.
  - Általában SOAP nevű protokollt használnak.

A szolgáltatásorientált architektúra hatalmas fejlődést jelent, különösen az üzleti alkalmazásrendszerek szempontjából.

7

## SOA működése előnyei

- **Rugalmasságot biztosít:**
  - a szolgáltatások elkészíthetők helyileg, illetve igénybe vehetők külső szolgáltatóktól.
- **A szolgáltatások bármilyen nyelven implementálhatók:**
  - a vállalat különböző részein esetlegesen használt különböző platformok és implementációs technológiák együttműködésének.
- **Hatékony együttműködés:**
  - a szolgáltatások segítségével a cégek és más szervezetek képesek az együttműködésre
  - használhatják egymás üzleti funkcióit.
- **Nincs inkompatibilitási probléma:**
  - a fejlesztéseket a kezdetektől aktív szabványosítási folyamat követte.

8

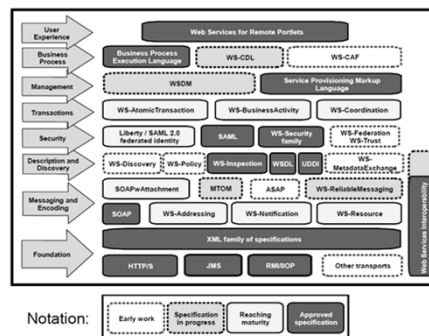
## SOA szabványok

- Alapvető szabványok a **webszolgáltatás** támogatására:



9

## Web szolgáltatás szabványok



10

## A webszolgáltatás-orientált architektúrák szabványai

- **SOAP:** üzenetcsere-szabvány, a szolgáltatások közötti kommunikációt támogatja.
  - Simple Object Access Protocol
  - Definiálja a szolgáltatások üzeneteinek szükséges és opcionális komponenseit.
  - Struktúrált és típusos XML dokumentumok
  - Alap esetben állapotmentes, egyirányú kommunikáció
- **WSDL:** webszolgáltatás definíciós nyelv (Web Service Definition Language – WSDL)
  - Definiálja, hogyan kell a szolgáltatások gyártóinak elkészíteniük szolgáltatásaik interfészét.
  - A web szolgáltatást mind absztrakt végpontot definiálja
  - A műveletek és az üzenetek is megfelelő absztrakcióval vannak leírva.

11

## A webszolgáltatás-orientált architektúrák szabványai

- **UDDI:** az Univerzális leírás, felderítés és integráció szabványa. (Universal Description, Discovery and Integration – UDDI)
  - Definiálja, hogy milyen komponenseket kell tartalmaznia a szolgáltatások specifikációinak.
  - Információkat tartalmaz a szolgáltatás gyártójáról, a szolgáltatásról, a szolgáltatás leírásának helyéről.
- **WS-BPEL:** szabványos munkafolyamat nyelv.
  - A különböző szolgáltatásokat tartalmazó folyamatprogramok definiálására használatos.

12

## A webszolgáltatás-orientált architektúrák szabványai

- **WSDL minták:**
  - <http://www.w3.org/2001/04/ws/ws-proceedings/uche/wsd.html>
  - <http://www.w3.org/2001/03/14-annotated-WSDL-examples.html>
  - <http://www.herongyang.com/WSDL/WSDL-20-Introduction-WSDL-Docment-Sample.html>
- **Egyéb információk:**
  - Apache Soap: <http://ws.apache.org/soap/>
  - Apache Axis2: <http://axis.apache.org/axis2/java/core/>

13

## SZOLGÁLTATÁSOK TERVEZÉSE...

14

## Szolgáltatások tervezése

- **Célja:** szolgáltatásorientált alkalmazásokban újrafelhasználható szolgáltatások kifejlesztése.
- **Biztosítani kell:**
  - a szolgáltatás egy újrafelhasználható absztrakciót reprezentáljon,
    - amely különböző rendszerek számára is hasznos lehet.
- Emberi intelligenciát igénylő folyamat.

15

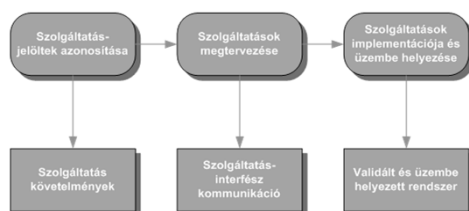
## Szolgáltatások tervezése

### Fázisai:

- **1. Szolgáltatásjelöltek azonosítása:** lehetséges szolgáltatások megkeresése, beazonosítása.
  - Későbbiekben implementálhatunk, és definiáljuk a szolgáltatások követelményeit.
- **2. Szolgáltatások megtervezése:** ebben a szakaszban tervezzük meg a logikai és a WSDL szolgáltatás-interfészeket.
  - A távolról hívható „metódusok” tervezése.
- **3. Szolgáltatások implementációja és üzembe helyezése:** Implementáljuk és teszteljük a szolgáltatásokat, majd a felhasználók számára hozzáférhetővé tesszük őket.

16

## Szolgáltatások tervezése



17

## SZOLGÁLTATÁSJELÖLTEK AZONOSÍTÁSA...

18

## Szolgáltatásjelöltek azonosítása

- A szolgáltatások az üzleti logikát kell támogassák!
- Azonosítási folyamat:
  - A szervezet üzleti folyamatainak megértése és elemzése.
  - Annak eldöntése, hogy milyen szolgáltatások szükségesek a folyamatok támogatásához.
- Cél:
  - olyan szolgáltatások kiválasztása, amelyek logikai egységet alkotnak.
  - függetlenek és újrafelhasználhatók
- Eredmény: az azonosított szolgáltatások halmaza, és a hozzájuk kapcsolódó követelmények.

19

## Szolgáltatás típusok

- Segédszolgáltatások: általános funkcionalitást implementálnak, melyet a különböző üzleti folyamatok használhatnak.
  - Pl.: egy pénznemátváltó szolgáltatás
- Üzleti szolgáltatások: ezek a szolgáltatások egy bizonyos üzleti folyamathoz kapcsolódnak.
  - Pl.: üzleti szolgáltatás lehetne például a hallgatók regisztrálása egy kurzusra.

20

## Szolgáltatás típusok

- Koordinációs vagy folyamatszolgáltatások: ezek sokkal általánosabb üzleti folyamatot támogatnak.
  - több különböző faktor és tevékenység is megjelenhet.
- Példa:
  - egy vállalat rendelési szolgáltatása, amelyben az ügyfelek leadhatják rendeléseiket és fizethetnek.

21

## SZOLGÁLTATÁS-INTERFÉSZEK TERVEZÉSE...

22

## Szolgáltatás-interfészek tervezése

- Célja: a szolgáltatáshoz kapcsolódó műveletek és paraméterek definiálása.
- 1. Az interfész logikai tervezése:  
A szolgáltatás követelményei alapján:
  - azonosítjuk a szolgáltatásokhoz tartozó műveleteket,
  - meghatározzuk a műveletek bemeneteit, kimeneteit, kivételeket.
- 2. Az üzenetek megtervezése:
  - El kell dönteni az üzenetek struktúráját,
  - és a szolgáltatás milyen üzeneteket küld és fogad.

23

## Szolgáltatás-interfészek tervezése

- A WSDL-fejlesztés:
  - a logikai és üzenetterveket átültetjük egy WSDL nyelven íródott absztrakt interfészleírásba.
- CASE eszközök támogatása:
  - Bizonyos eszközök képesek a logikai interfészleírást lefordítani a megfelelő WSDL-reprezentációba.
    - Vagy fordítva
  - Pl.: Eclipse, Netbeans, Visual Studio, stb.

24



## Szolgáltatások üzembe helyezése

**Fázisai:**

- **1. Implementáció:**
  - Általában szabványos programozási nyelvek segítségével történik
    - Olyanokkal, melyek tartalmaznak könyvtárakat a szolgáltatásfejlesztés széles körű támogatására
    - pl. C#, Java
- **2. Tesztelés:** a szolgáltatást tesztelni kell.
  - A bemenetek elemzése és osztályozása
  - Az összes lehetséges üzenet elkészítése, és ellenőrzése, hogy a megfelelő kimenő üzenetek érkeztek-e.
  - Célszerű kivételeket generálni minden üzenetnél
    - Hogy a szolgáltatás képes-e megbirkózni az érvénytelen bemenetekkel

26

## Szolgáltatások üzembe helyezése

- Ma már létezik több eszköz, amely képes a szolgáltatások vizsgálatára és tesztelésére
  - tesztesetek generálása WSDL-ből,
  - kapcsolatok tesztelése,
  - Leírók ellenőrzése, generálása
  - Automatizált tesztesél,
  - stb

27

## Tesztelést segítő alkalmazások

- **soapUI:** <http://www.soapui.org/>
- Verziók: Linux, MAC, Windows 32/64, portable/installer
  - Ingyenes és PRO fizetős verziók

28

## Tesztelést segítő alkalmazások

- **Web Service Call Composer:** <http://www.webinject.org/>
- Verziók: Windows
  - 15 napig ingyenes, pénzes verziók

29

## Tesztelést segítő alkalmazások

- **PushToTest:** <http://www.pushtotest.com/>
- Verziók: Linux, MAC, Windows
  - Rengeteg partner

30

## Szolgáltatások üzembe helyezése

- **3.Üzembe helyezés:**
  - a szolgáltatást használatra elérhetővé tesszük egy webszerveren.
  - A legtöbb szerverszoftver esetén ez egyszerű:
    - csak egy megadott könyvtárba kell telepíteni a végrehajtható szolgáltatást tartalmazó állományt.
    - Ha a szolgáltatást publikusnak szánjuk, akkor még egy UDDI-leírást is készítenünk kell.

31

**Köszönöm a figyelmet!**

32