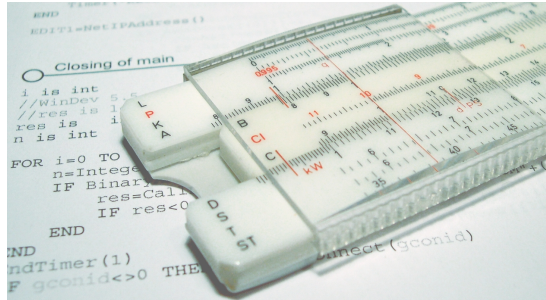


Szoftver architektúra, Architektúrális tervezés



Szoftver-technológia I.



Szoftver-technológia I.

Irodalom

- Ian Sommerville: Software Engineering, 7th e. chapter 11.
- Roger S. Pressman: Software Engineering, 5th e. chapter 14.
- Bass, Clements, Kazman: Software Architecture in Practice, Addison-Wesley, 2004

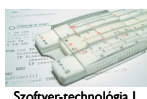


Szoftver-technológia I.

Mi a szoftver architektúra?

- A rendszert felépítő alrendszerek (szoftver komponensek) kerete
 - alrendszerek meghatározása
 - alrendszerek tulajdonságai
 - vezérlési, valamint kommunikációs kapcsolataik

3



Szoftver-technológia I.

Mi még a szoftver architektúra?

- Magasszintű terv
- A rendszer átfogó struktúrája
 - több nézet -> több struktúra
- Komponensek és összekötők összesége

4

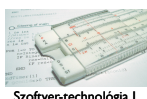


Szoftver-technológia I.

A szoftver architektúra szerepe

- Az érdekelt szereplők kommunikációjának lehetővé tétele
- A korai fejlesztési fázisok döntéseinek támogatása a követelmények tükrében
- Nagyléptékű újrafelhasználhatóság elősegítése

5

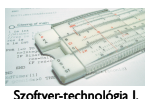


Szoftver-technológia I.

A szoftver architektúrák forrása

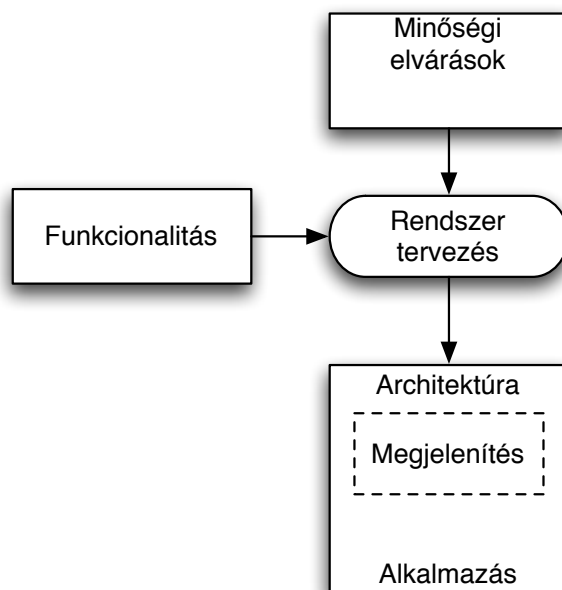
- Üzleti és technológiai döntések eredménye
- Meghatározó a környezet szerepe
- A fejlesztők céljai és stratégiája által befolyásolt követelmények vezetnek különféle szoftver architektúrákhoz

6

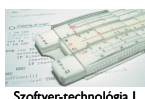


Szoftver-technológia I.

Az architektúra kialakítása



7

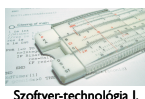


Szoftver-technológia I.

Az architektúra ciklus

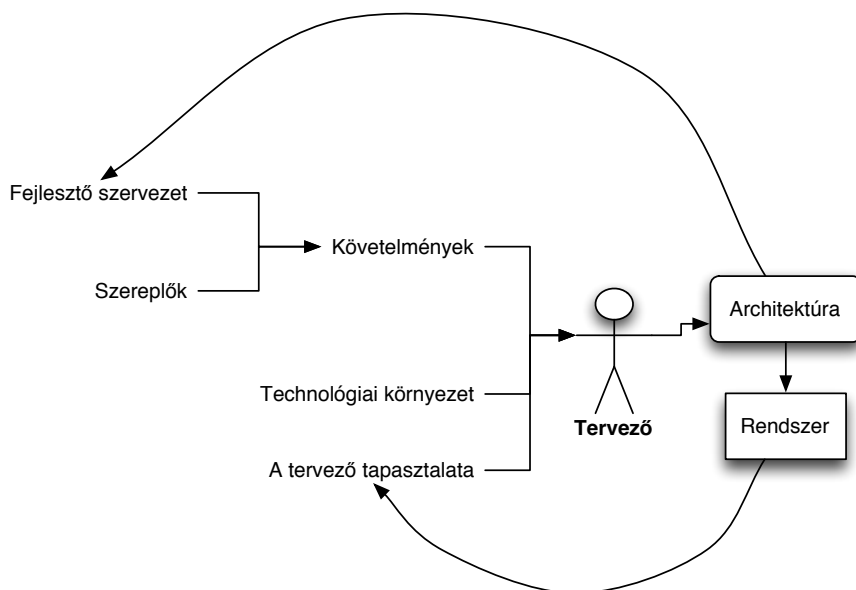
- Az architektúra meghatározó a fejlesztő szervezet szerkezetére
- Az architektúra befolyásolja a fejlesztő szervezet céljait
- Az architektúra hatással van a követelményekre
- A fejlesztés során a fejlesztők tapasztalatokat szereznek architektúráról
- Bizonyos rendszerek hatással vannak a fejlesztési kultúrára

8

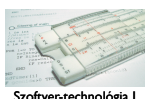


Szoftver-technológia I.

Az architektúra ciklus (folyt.)



9

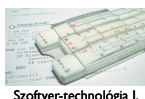


Szoftver-technológia I.

Mitől jó egy architektúra?

- Kis számú vezető tervező
- Jól definiált funkcionális követelmények és minőségi jellemzők
- Jól dokumentált az architektúra
- Az architektúra értékelésében az összes érintett szereplő részt vesz
- Kvantitatív minőségi mértékek
- Inkrementális architektúra implementáció
- Kevés erőforrás versenyhelyzet a fejlesztés során

10

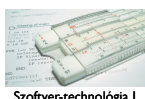


Szoftver-technológia I.

Architect



11



Szoftver-technológia I.

Struktúrális ököl szabályok

- Jól definiált modulok
 - információ elrejtés, szeparáció
- Jól definiált interfészek
- Kereskedelmi termékektől független architektúra
- Jól ismert architektúrális taktikák alkalmazása
- Egyszerű interakciós minták

12



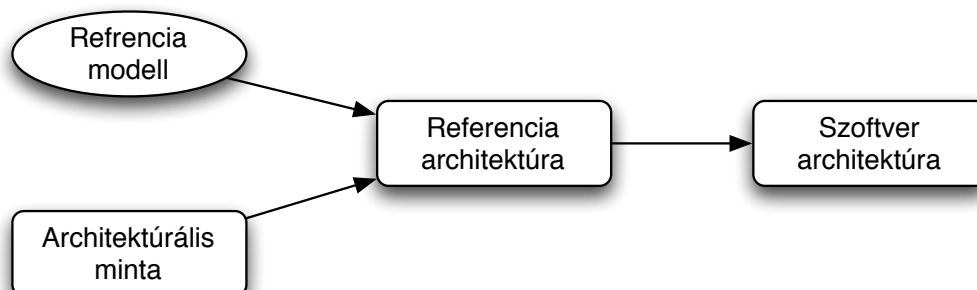
Architektúra elemek

- Architektúrális minta
 - típus elemek és kapcsolatok, kényszerek
 - pl. kliens–szerver minta
- Referencia modell
 - standard funkcionális felosztás és adatfolyam megoldások
 - pl. adatbázis kezelő rendszer
- Referencia architektúra
 - referencia modell leképezése szoftver elemekre
 - pl. ISO OSI architektúra

13



Architektúra elemek (folyt.)



14

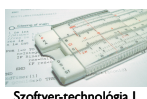


Szoftver-technológia I.

Architektúrális szerkezetek / nézetek

- Modul szerkezetek
 - funkcionális felbontás
- Komponens-és-összekötő szerkezetek
 - futási idejű komponensek és csatlakoztatások
- Allokációs szerkezetek
 - szoftver és külső elemek összerendelése

15



Szoftver-technológia I.

Architektúrális szerkezetek / nézetek (folyt.)

- Hogyan bontható a rendszer kódegységekre, modulokra?
- Hogyan struktúrázható a működő rendszer futási időben komponensekre, és lépnek egymással interakcióba ezek a komponensek?
- Milyen kapcsolatban van a rendszer a környezet nem szoftver elemeivel (pl. processzorok, fájlrendszerek, fejlesztő teamek)?

16



Szoftver-technológia I.

Szoftver struktúrák

- Modul szerkezetek
 - Dekompozíció
 - Használat
 - Réteg szerkezet
 - Osztályok, generalizálás
- Komponens és összekötő szerkezetek
 - Kliens-szerver
 - Folyamat
 - Konkurrencia
 - Osztott adatok
- Allokációs szerkezetek
 - Telepítési szerkezet
 - Munkakiosztás (fejlesztési)
 - Implementációs szerkezet (konfiguráció mgt.)

17



Szoftver-technológia I.

Architektúrális tervezés

- A rendszer-tervezés korai fázisa
- Specifikációs és tervezési fázis összekötése
- Specifikálással párhuzamosítható
- Fő rendszer-komponensek meghatározása

18



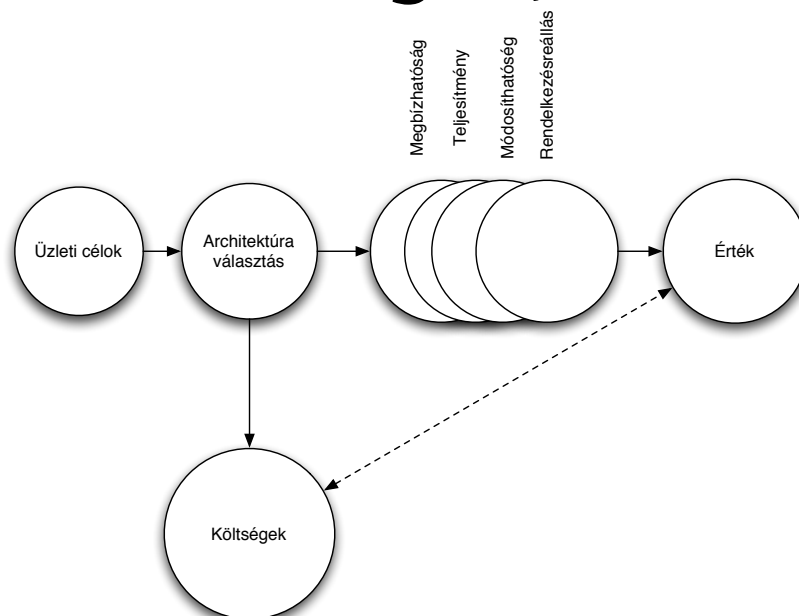
Architektúra és a rendszer jellemzői

- A rendszer architektúrája hatással van a rendszer minőségi jellemzőire
- Nem funkcionális követelmények szerepe
 - Teljesítmény
 - Biztonság
 - Megbízhatóság
 - Rendelkezésreállítás
 - Karbantarthatóság

19



Architektúra, költségek, érték



20



Szoftver-technológia I.

Minőségi jellemzők



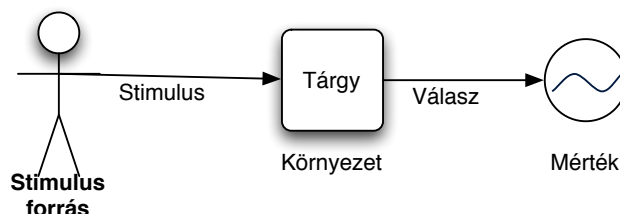
21



Szoftver-technológia I.

Minőségi jellemzők

- Minőségi jellemzők meghatározása
 - Stimulus forrás
 - Stimulus
 - Környezet
 - Tárgy
 - Válasz
 - Metrika



22



Stratégiák és taktikák

- Taktikák a minőségi célok elérésére
- A taktikák összessége a stratégia
- A taktikák mint tervezési lehetőségek
- A tervezési minták tartalmazznak taktikákat

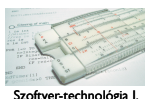
23



Taktikák (pl.)

- Magas rendelkezésreállási taktikák
 - Hibadetektálás
 - Ping/echo
 - Heartbeat
 - Kivételkezelés
 - Javítás
 - Szavazás
 - Aktív redundancia
 - Passzív redundancia
 - Visszaállítás
 - Árnyék működés
 - Állapot újraszinkronizálás
 - Megelőzés
 - Kikapcsolás, újraindítás
 - Tranzakció kezelés

24

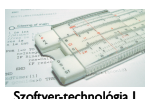


Szoftver-technológia I.

Architektúrális konfliktusok

- Nagyméretű komponensek jó teljesítményűek, de nehezen karbantarthatók
- A redundancia fokozza rendelkezésreállást, de adatbiztonsági problémákat okoz
- A biztonsági megoldások általában megnövelt adatmozgatással járnak

25



Szoftver-technológia I.

Architektúrális tervezési kérdések

- Vannak-e generikus alkalmazás architektúrák?
- Hogyan kell a rendszert elosztani?
- Hogyan kell a rendszert modulokra bontani?
- Milyen legyen az alkalmazás vezérlési szerkezete?
- Hogyan lehet az architektúrákat értékelni, összehasonlítani?
- Hogyan kell dokumentálni egy architektúrát?

26



Néhány elterjedt architektúrális modell

- Osztott adatokra épített architektúra
 - pl. CASE eszközök
- Kliens–szerver architektúra
 - pl. hálózati, kommunikációs szoftverek
- Réteg szerkezet architektúra
 - pl. hálózati referencia modellek
- Funkcionális csőrendszerek
- Eseményvezérelt rendszerek

27



Osztott adatok

- Az alrendszerek adatokat cserélnek
 - központi adatbázis
 - saját adatbázisok, explicit adatcsere
- Közös adatigényű alkalmazások

28

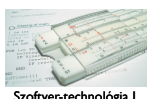


Szoftver-technológia I.

Osztott adatok (folyt.)

- Előnyök
 - nagy mennyiségű adat hatékony megosztása
 - központi adat kezelés lehetősége
- Hátrányok
 - minden részrendszerben azonos adatszerkezet
 - elosztottság nehézkes

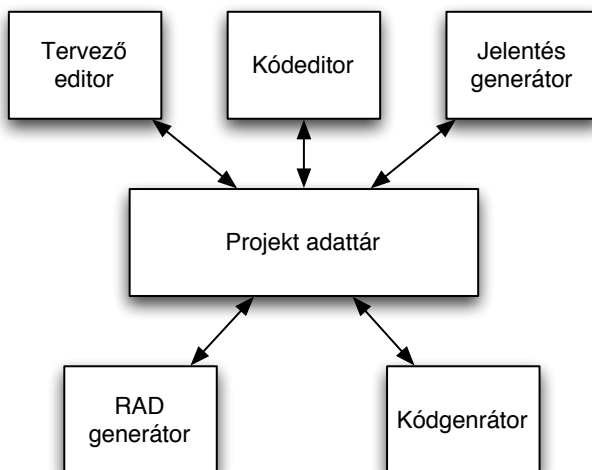
29



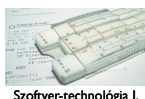
Szoftver-technológia I.

Osztott adatok (pl.)

- Integrált CASE eszköz



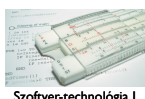
30



Kliens–szerver modell

- Elosztott rendszerek
 - adatok
 - feldolgozás
- Önnálló szolgáltatásokat nyújtó szerverek
- A szolgáltatásokat igénybevevő több, különböző kliens
- Kommunikációs mechanizmus
 - hálózat
 - üzenetkezelő

31



Kliens–szerver modell (folyt.)

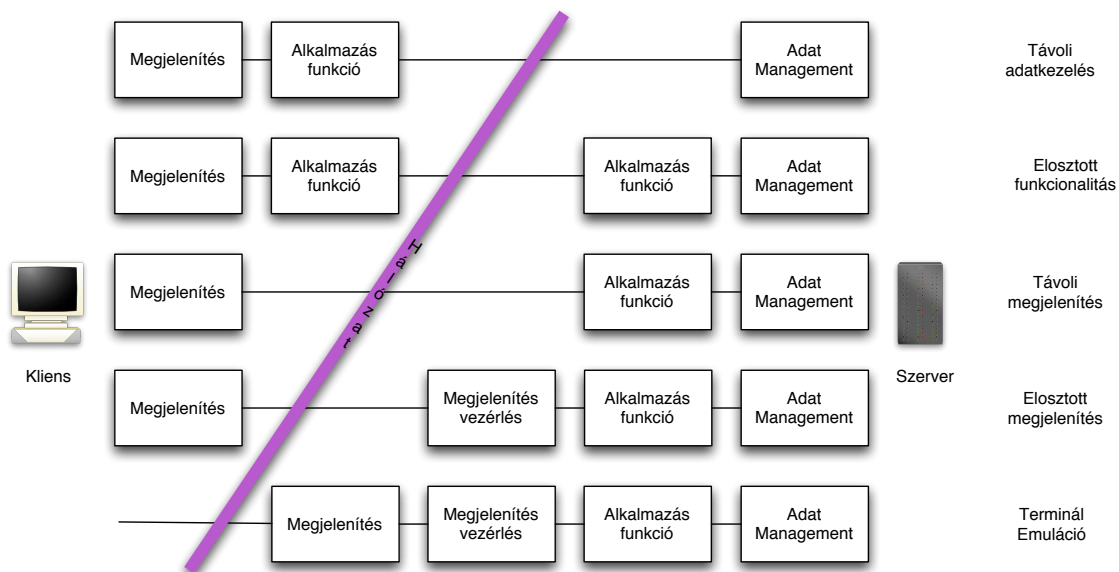
- Előnyök
 - Elosztott rendszer
 - Hálózat kihasználása
 - Olcsóbb hardvermegoldások
 - pl. NC
 - Skálázhatóság
- Hátrányok
 - Adat reprezentációs problémák
 - Redundáns management
 - Szolgáltatások leírása, megtalálása

32

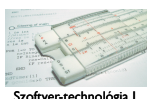


Szoftver-technológia I.

Üzleti alkalmazások klien-szerver megoldásai



33



Szoftver-technológia I.

Réteg szerkezet modell

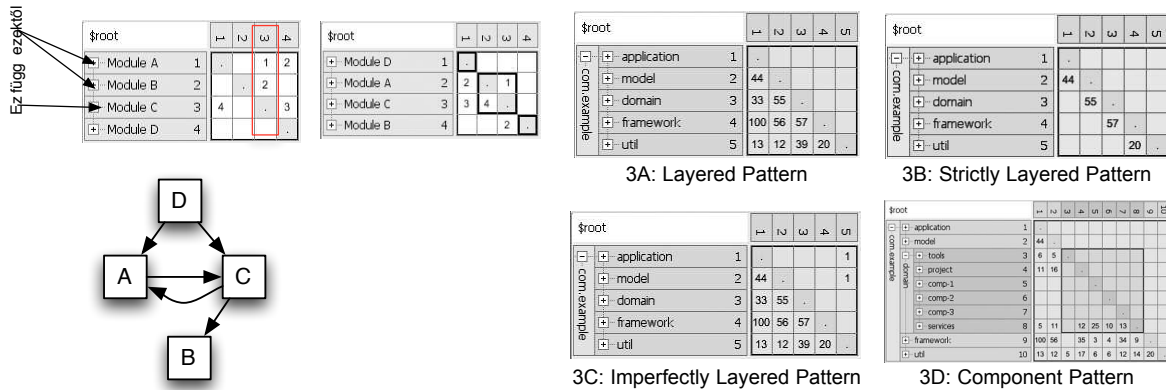
- Speciális alrendszer elrendezés
- Rétegekbe szervezett modulok, szolgáltatások
- Jól definiált interfészek
- Inkrementálisan, egymástól elkülönülten fejleszthető rétegek
- Mesterségesen kialakított rétegek

34



Réteg szerkezet

- Hierarchikus dekompozíció
 - alrendszer függőségei
 - szemléltetés: Dependency Structure Matrix

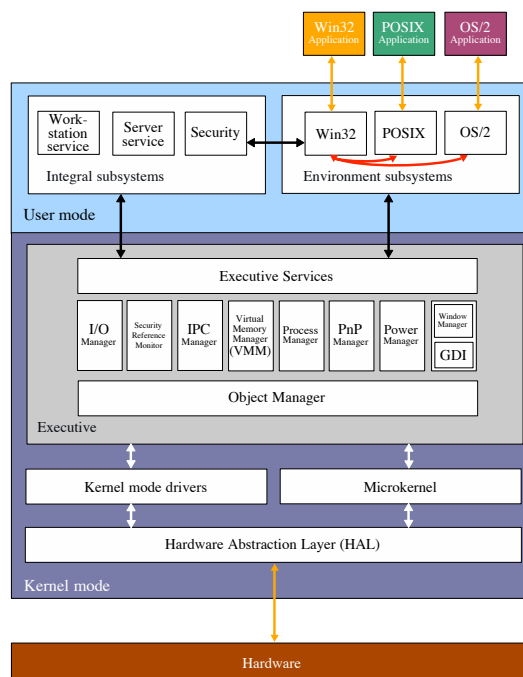


35



Réteg szerkezet (pl.)

- Windows NT



36



Funkcionális csőrendszerek

- Adatfolyam orientált megközelítés
- Cső, szűrő modell
- Szekvenciális, kötegelt feldolgozás
- Nem interaktív alkalmazások

37



Csőrendszer modell jellemzői

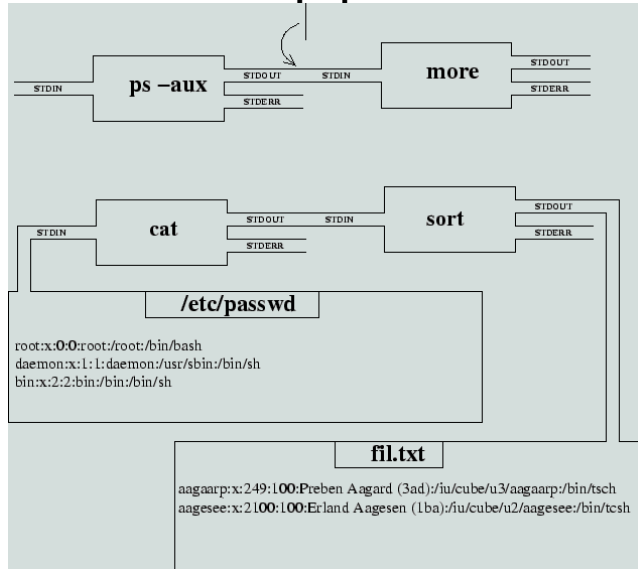
- Transzformációk újrafelhasználása
- Intuitíven kialakítható szerkezet
- Könnyen bővíthető új transzformációkkal
- Konkurens rendszerek is kialakíthatók
 - szinkronizáló csőrendszerek
- Kötött adatformátum a csövekben

38



Csőrendszer (pl.)

- UNIX pipe



ps -aux | more

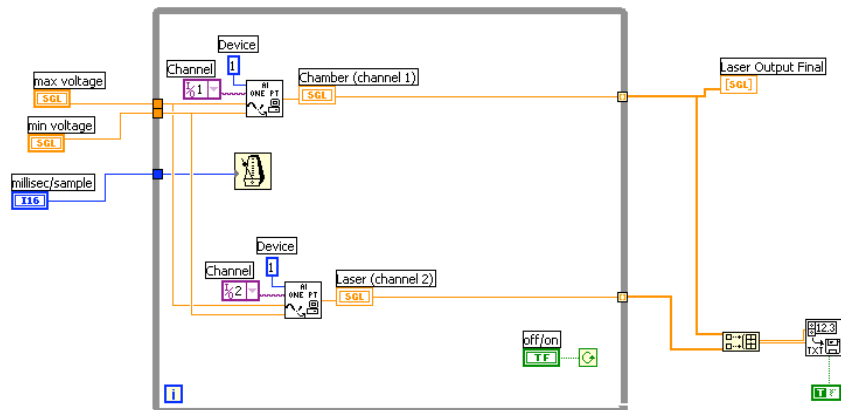
cat /etc/passwd | sort > fil.txt

39



Csőrendszer (pl.)

- Labview virtuális műszerek



40



Szoftver-technológia I.

Eseményvezérelt rendszerek

- A rendszer vezérlésén kívülről érkező események határozzák meg a viselkedést
- Eseményvezérelt modellek
 - broadcast modell
 - interrupt modell

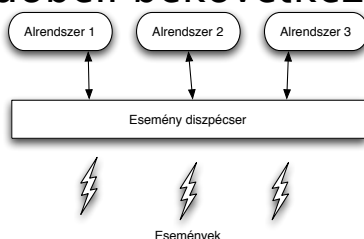
41



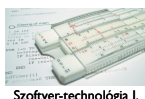
Szoftver-technológia I.

Broadcast modell

- Elosztott alrendszerek integrálása
- Az alrendszerek “előfizetnek” az őket érdeklő eseményekre az eseménykezelőnél
- Eseménytovábbító mechanizmus
 - eseménysor
- Az eseményekhez eseménykezelőket definiálnak
- Események időbeli bekövetkezése ismeretlen

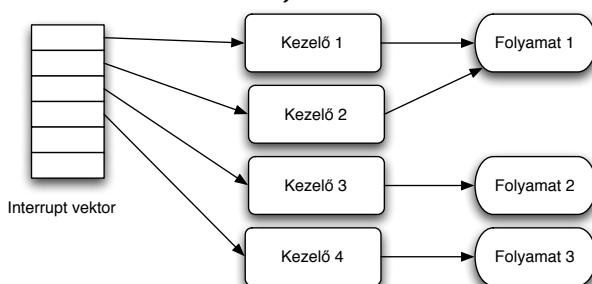


42

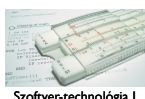


Interrupt modell

- Valós idejű rendszerek
- Definiált interrupt típusok és típusonkénti kezelő eljárások
- Interrupt vektor
- Gyors működés, nehéz tesztelés

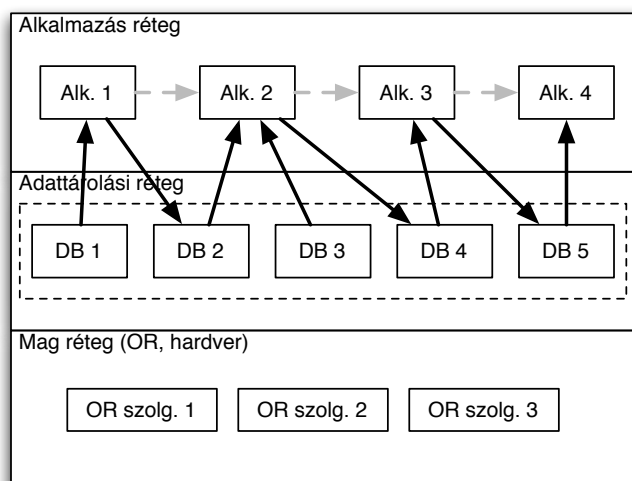


43

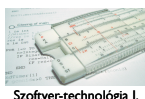


Architektúrális minták kombinációja

- Réteg szerkezet + csőrendszer + megosztott adatok



44



Összefoglalás

- A szoftver architektúra fogalma
- Architektúrális nézetek
 - modul szerkezetek
 - komponens-és-összekötő szerkezetek
 - allokáció szerkezetek
- Architektúrális taktikák szerepe
- Néhány klasszikus architektúra
 - osztott adatok, kliens-szerver, réteg szerkezet, eseményvezérelt