



Modellezési feladatok

Adatmodellezés

Intelligens rendszerfelügyelet (VIMIA370)

Készítették: Kocsis Imre, Micskei Zoltán, Szatmári Zoltán, Tóth Dániel

Utolsó módosítás: 2011. június 24.

Verzió: 1.1

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

1 Bevezető

Az *Intelligens rendszerfelügyelet* tantárgy keretében különböző rendszerek modellezési lehetőségeibe is belekóstolunk. Akár egy egyszerű modell elkészítése is nagyon hasznos lehet:

- segít összegyűjteni és megérteni az adott szakterület fogalmait,
- szisztematikus módszert ad, hogy összegyűjtsük az előkerülő fogalmakhoz kapcsolódó szabályokat és kényszereket („egy rendeléshez hány kapcsolattartót lehet megadni?”, „kötelező-e kitölteni az értesítési telefonszámot, ha az e-mail meg van adva?”, stb.),
- szabványos modellezési nyelvek használatával egyértelműbbé tehetjük, hogy mit értünk az egyes elemeken és kapcsolatokon,
- lehetőség nyílik, hogy automatikus ellenőrzéseket valósítsunk később meg (megadtunk-e minden szükséges adatot, kiszámoljuk a rendszer bizonyos jellemzőjét).

A tantárgy keretében két különböző feladatot néztünk meg részletesebben a félév során:

- *Adatmodellek készítése*: egy adott terület fogalmait és azok kapcsolatát gyűjtjük össze. Tipikusan ez egy magas szintű, kezdeti modell, ami még nem az implementáció közeli részletekre koncentrál.
- *Szolgáltatásbiztonság vizsgálata*: összetett rendszerek rendelkezésre állását, hibatűrését, adott hibajelenségek diagnosztikáját segítjük különböző hibamodellek összeállításával.

A tantárgy vizsgájának gyakorlati részében ilyen feladatok megoldását várjuk el, ez a segédlet a vizsgára való felkészülést segíti. Javasoljuk, hogy a kidolgozott mintapéldákat is először mindenki próbálja *önállóan* megoldani, és csak utána nézze meg a megoldást. A modellezés is egy olyan készség, amit csak gyakorlással lehet fejleszteni, ezért érdemes utána a gyakorló feladatokat is önállóan megoldani (pusztán a megoldás átolvasása még nem elég ahhoz, hogy később alkalmazni is tudjuk az ott látott ismereteket).

1.1 Adatmodellek készítése

Adatmodellek készítése során egy adott terület fogalmait és azok kapcsolatát gyűjtjük össze. Többféle leírási formát használhatunk erre, mi most a tárgy keretében UML osztálydiagramokat alkalmazunk.

A felkészüléshez első lépésként nézzük át a modellezés előadás anyagát (legyünk tisztában a modell – metamodell fogalmával, az UML alapvető elemkészletével és azok jelentésével).

A modellezés nem egy egzakt folyamat, egy adott környezethez sokféle modellt lehet készíteni (pl., egy adott tulajdonságot attribútumként jelenítünk meg vagy külön osztályban ábrázoljuk, vagy milyen mértékben használunk öröklést, stb.). Ezért a megadott megoldásokhoz képest természetesen más megoldások is elképzelhetőek. Ezeknek a „jóságát” nehéz definiálni, az lehet szempont, hogy tartalmaz-e minden megadott adatot, mennyire könnyű később bővíteni, mennyire kényelmes használni, mennyire egyértelmű, stb.

Tipikus hibák és általános tanácsok:

- Metamodellek készítése esetén nem kell a kapcsolatnak megfelelő attribútumokat felvenni (tehát pl. arraylistek a kapcsolódó osztályoknak megfelelően), ez annál magasabb szintű modell.
- Egy adatmodellbe túl sok értelme nincsen interfészeket berakni, főleg olyat, aminek nincs egy módusa sem (attribútumot meg eleve nem illik interfészbe rakni). Használjunk helyette inkább absztrakt osztályokat.
- UML példány modell készítése esetén már nem szokás a kompozíciót berajzolni, csak sima vonallal jelöljük az objektumok közötti linkeket (bár sok UML modellező eszköz kompozíciót használ példány szinten is).

- A modellezés feladatnál érdemes elolvasni a teljes feladat szövegét, mert ha meg van adva egy konkrét környezet, amit utána példány modellként el kell készíteni, az sokat segíthet.
- Érdemes valami egységes elnevezési koncepciót használni, osztálynévbe PascalCase, példány névben tipikusan camelCase formát használunk. Lehetőleg ne használjunk ékezetet vagy szóközt modell elemek nevében, az csak feleslegesen megnehezíti később a feldolgozását.
- Mindig legyen a példányoknak egyértelmű neve, azzal lehet azonosítani őket.

2 Kidolgozott mintapéldák

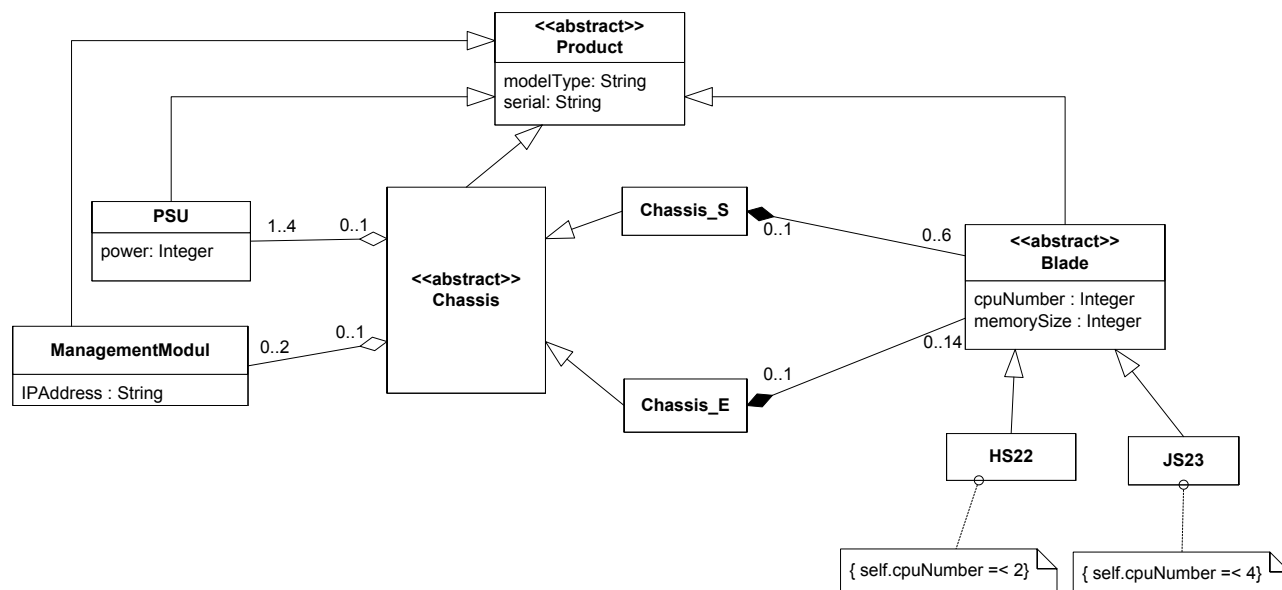
2.1 IBM BladeCenter

2.1.1 A feladat szövege

1. IBM BladeCenter rendszerek modellezéséhez készítsen egy egyszerű metamodell, melynek segítségével a következő adatokat tudjuk majd tárolni. Egy BladeCenter rendszer egy keretből (chassis) áll, amibe penge szervereket (blade) lehet berakni. Jelenleg E és S típusú keretekkel foglalkozunk, az E-be 14 darab, az S-be 6 darab penge fér. A kereteket és pengéket az IBM a modell számukkal azonosítja, az egyes konkrét termékeknek pedig egyedi sorozatszáma van. A keretekbe a pengéken kívül kell még tápegység (maximum négy fér egy keretbe, különböző teljesítményű modellek kaphatóak) és legfeljebb kettő úgynevezett menedzsment modul. A menedzsment modulon keresztül lehet távolról felügyelni a keretet, a modult ilyenkor IP címével érjük el. A pengékről tárolni akarjuk a bennük lévő fizikai CPU-k számát és a memória méretét. Két féle pengét akarunk jelenleg nyilvántartani, a 4 CPU foglalattal rendelkező JS23-ast és a két CPU foglalatos HS22-est. (6 pont)
2. A fenti metamodellhez készítsen el egy példánymodellt. Egy 8677-3TG modellű E-s keretet vettünk az eBay-en. A keret két 74P4452 típusú 2000 wattos tápegységgel és egy menedzsment modullal érkezett, a modult még nem állítottuk be. A modul sorozatszáma 11373P92. A keret egy darab pengével érkezett, egy 7996-60 típusú JS23-assal, amiben 2 processzor és 64 GB memória van. A modellben jelölje a hiányzó adatokat is, amiket még ki kéne tölteni a metamodell alapján. (4 pont)

2.1.2 Egy lehetséges megoldás

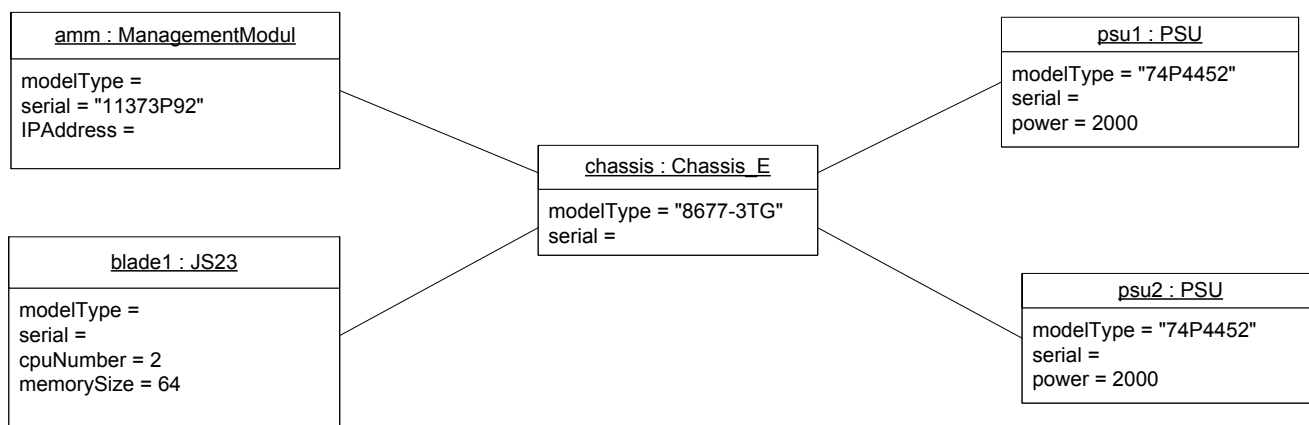
1.



Megjegyzések és tapasztalatok:

- Bevezettünk egy absztrakt őosztályt, hogy a mindenkinél szereplő modell- és szériaszámot egyszerű legyen kezelni.
- Figyeljük meg, hogy ez egy analízis szintű modell, tehát pl. láthatóságot nem feltétlen kell bele rakni.
- A konkrét pengék maximális CPU száma itt OCL kényszerek (constraint) segítségével van jelezve. Ez lehetne akár egy sima megjegyzés is jelen esetben, vagy például az attribútum típusát felül lehetne definiálni a leszármazott osztályban egy megfelelő értéktartományú típusal.
- A tápegységek (PSU) számossága itt most 1..4, de akár lehetne 0..4 is (mindkettő mellett lehet érvelni).
- A chassis-blade kapcsolat reprezentálása volt még érdekes a modellben. Sima asszociáció esetén ugye az a gond, hogy akkor lehetne, hogy egy Blade példány egyszerre tartozzon egy S és ez E kerethez is. Ezért itt kompozíciót (composite aggregation) használtunk, ami az UML-ben egy erősebb fajtája az aggregációnak, egyszerre csak egy kompozícióként jelölt kapcsolatban szerepelhet egy példány.
- Sokadszorra előkerült, de újfent érdemes kihangsúlyozni, hogy interfészt ne nagyon rakjunk adatmodellbe. Egy interfész attól interfész, hogy metódusai vannak (különösen problémás dolog attribútumokat rakni egy interfészbe). Adatmodellben absztrakt osztályokat használunk.
- Fontos, hogy az attribútumnak legyen típusa. Ehhez használjuk az UML általános típusait (String, Integer, Boolean...), és ne valamelyik programnyelv speciális implementáció közeli típusát (pl. int32).
- Az asszociációkat és az asszociáció végeket el lehet nevezni. Mivel most itt a legtöbb szerep egyértelmű volt, ezért ettől eltekintettünk a modell megalkotása során.

2. Itt megint az a lényeg, hogy a saját magunk megadott metamodellnek típushelyes példánya legyen a modell, és ki lehessen fejezni a példában szereplő elemeket.



- Arra figyeljünk, hogy az UML példány szinten már nem szokta jelölni az aggregációt vagy kompozíciót, ott már csak sima kapcsolatok (link) vannak.
- Ugyanúgy nem lehet már multiplicitást sem megadni a kapcsolatokon. Ha valamiből két példányom van, akkor azt két külön, különböző nevű objektummal kell ábrázolni.
- A példány neve elhagyható (bár nem javasolt), azonban a kettőspontot és mögötte a típusnevet kötelező megadni, ettől tudom, hogy az minnek a példánya.
- Ha egy attribútumnak nincsen értéke, akkor hagyjuk üresen az érték részt, és ne „???”-et írjunk oda.

2.2 SharePoint alkalmazások modellezése

2.2.1 A feladat szövege

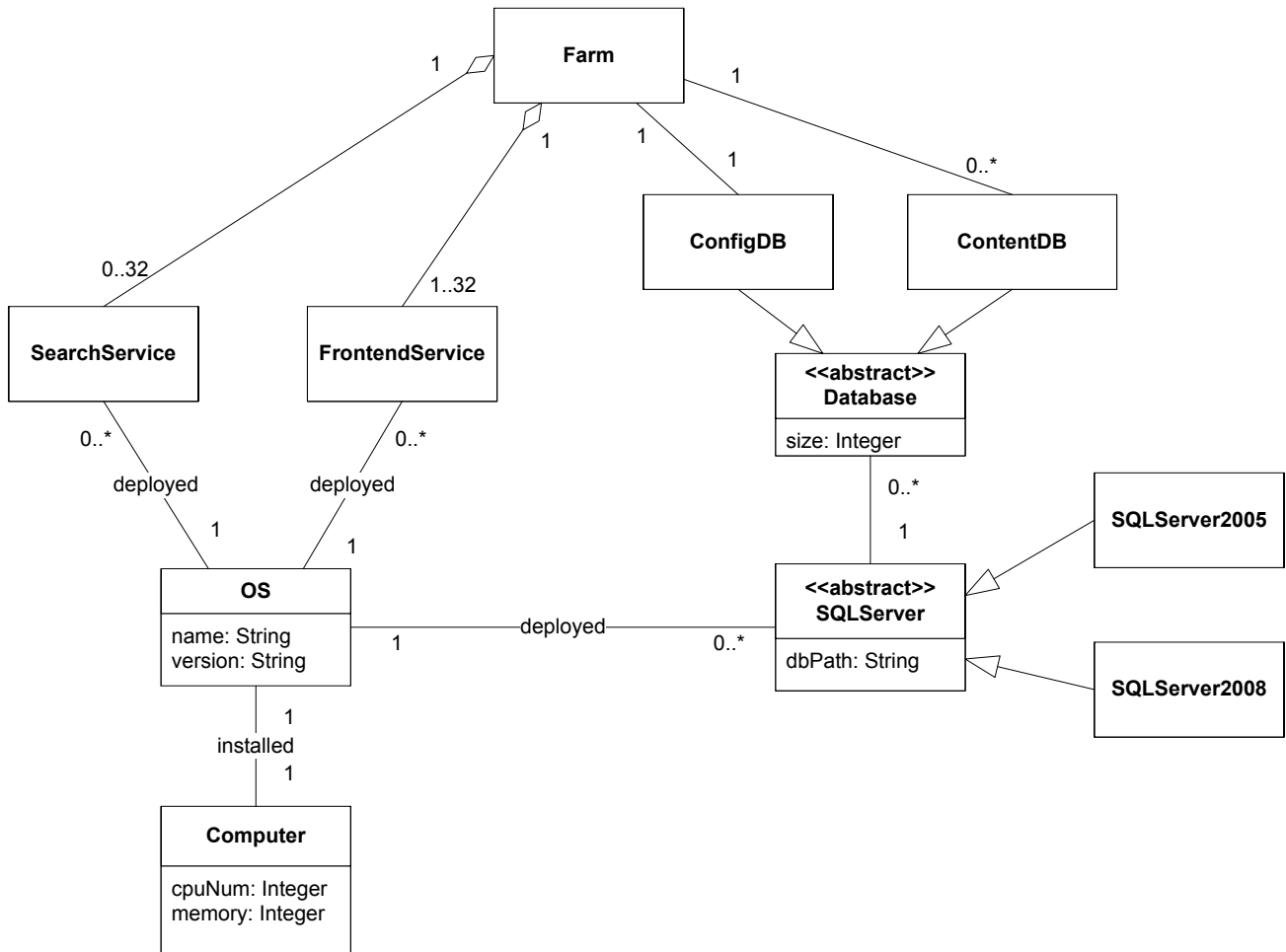
a) Microsoft SharePoint platformra fejlesztünk alkalmazásokat, és a fejlesztői és teszt rendszerekhez használt infrastruktúrák modellezéséhez kell egy metamodell készítenünk. A SharePoint flexibilis telepítési opciókat ajánl. A telepítés alapeleme a farm. Egy farm működéséhez legalább egy web frontend szolgáltatás kell, és opcionálisan lehet kereső szolgáltatást is telepíteni. A web frontend és keresés telepíthető ugyanarra a számítógépre, ezekből a szerepekből külön-külön legfeljebb 32 lehet a farmban. A modellben tárolni szeretnénk, hogy melyik szolgáltatás melyik számítógépre van telepítve, azon milyen operációs rendszer van (annak mi a verziója), valamint, hogy a számítógépben hány processzor és mennyi memória van. A farm működéséhez ezen kívül szükség van az adatokat tároló adatbázisokra. Pontosan egy darab konfigurációs adatbázis kell, és tetszőleges sok tartalom adatbázist adhatunk meg. Az adatbázisokról tudni akarjuk a méretüket. Az adatbázisokat SQL Server 2005 és 2008-on tárolhatjuk, az adatbázis szerverről az alapértelmezett adatbázis elérési útvonalat jegyezzük fel. A metamodellben figyeljünk a multiplicitások jelölésére! (6 pont)

b) Készítsünk egy példány modellt a fenti metamodellhez. Egy közepes méretű tesztrendszerünk van. A farm két frontend szerverből áll, az egyikre telepítve van a kereső szolgáltatás is. Ezen kívül van egy SQL 2008 adatbázis szerverünk, melyen a 100 MB-os konfigurációs adatbázison kívül egy 500 MB-os és egy 3 GB-os tartalom adatbázis van. Az adatbázis szerver egy négyprocesszoros, 32 GB-os, a két frontend pedig egy-egy kétprocesszoros, 8 GB memóriával rendelkező gép. (3 pont)

c) Módosítsuk az a) feladatban elkészített metamodell úgy, hogy jelölni tudjuk, hogy ha a szolgáltatásokat virtuális gépekre telepítjük. Rajzolja le külön a metamodell megváltozott részét! (1 pont)

2.2.2 Egy lehetséges megoldás

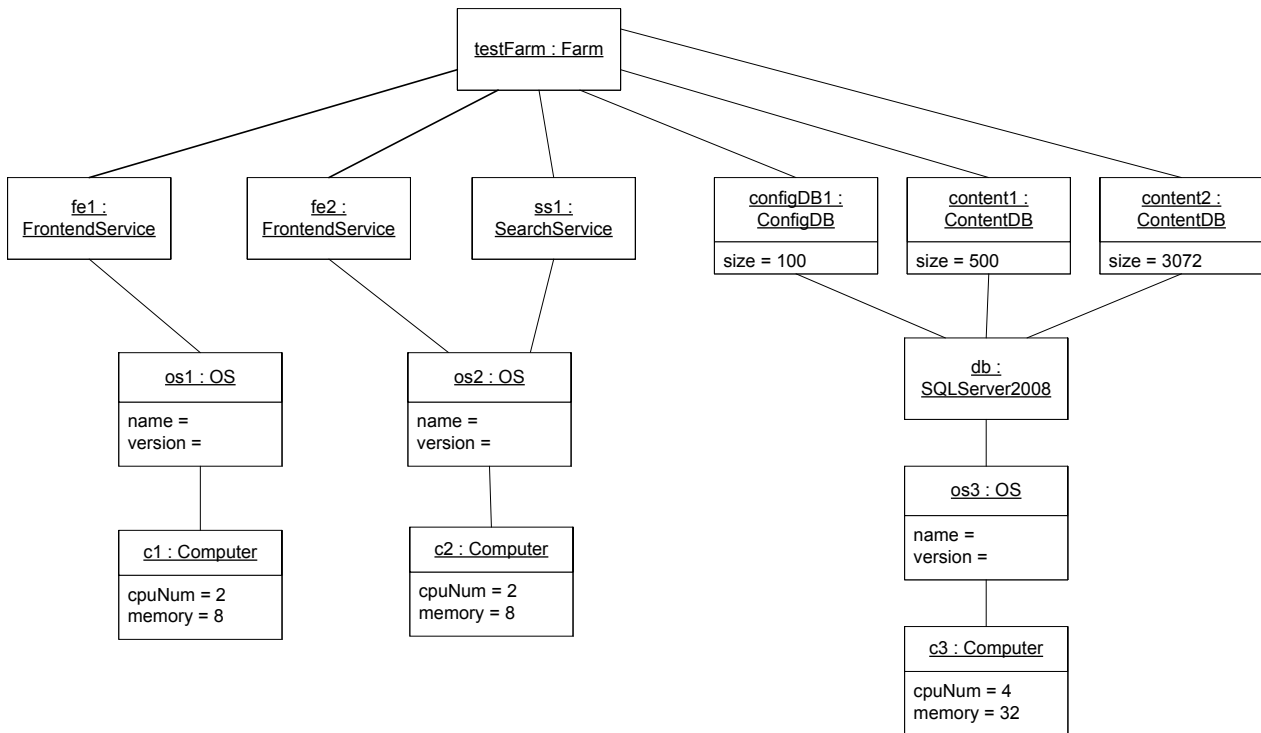
a)



Persze sok mindent lehet kicsit másképp modellezni:

- Az operációs rendszer lehet a számítógép attribútuma.
- A különböző adatbázis típusok nem külön osztályok lesznek, hanem a farmból két különböző elnevezett asszociáció indul.
- A különböző szolgáltatásoknak is lehet egy absztrakt ősosztályt készíteni.
- Az SQL szerver típusát lehet egy megfelelő enumeráció típusú attribútummal megadni.
- A farm és a szolgáltatások között mehet sima asszociáció is, bár a kompozíció talán szerencsésebb.
- ...

b) Itt a lényeg, hogy az a)-ban megadott metamodell típushelyes példánya legyen a megadott modell.



c) Itt is sokféle lehetőség van:

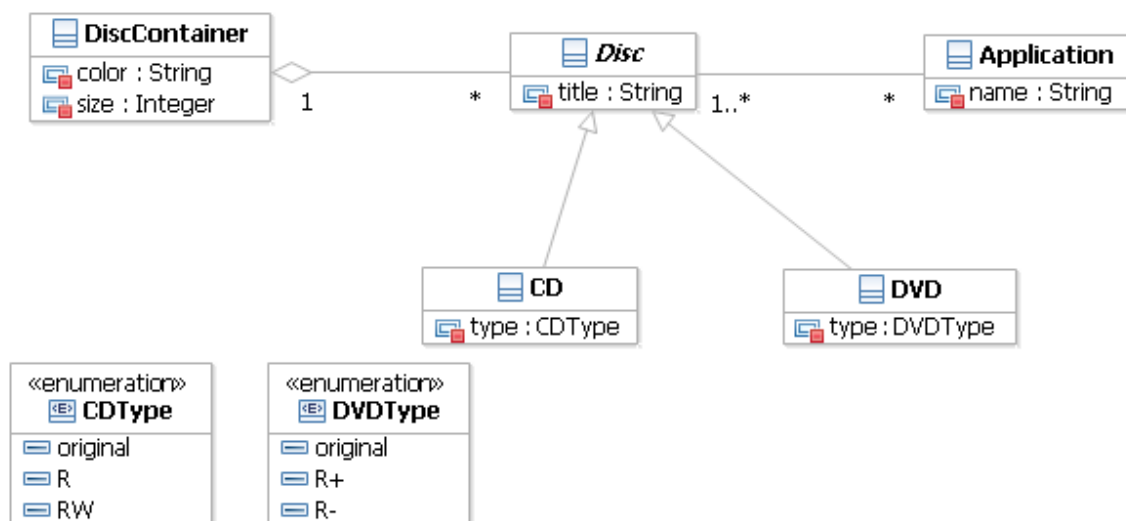
- isVirtual attribútum felvétele a Computer osztályba
- VirtualMachine osztály bevezetése, ami bekerül az OS és a Computer közé
- ...

3 Gyakorló feladatok

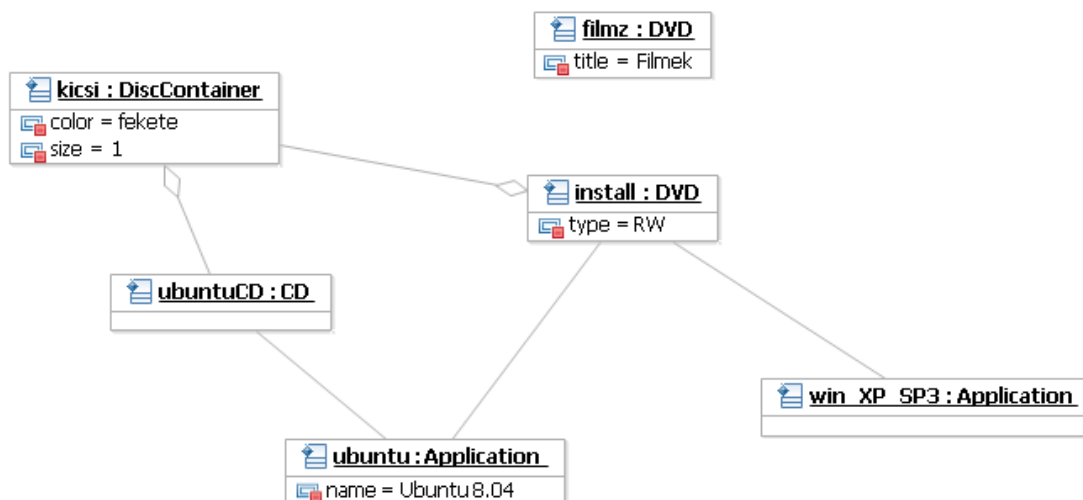
A következő fejezet rövidebb modellezési példákat tartalmaz, melyek segítenek az alapok elsajátításában.

3.1 CD tárolás

Hogy átlássuk a nagyüzemi CD és DVD írás beindulása óta kialakult káoszt, a lemezekről és tárolókról a következő információkat tartjuk nyilván.



Típushelyes példánya-e a fenti UML osztálydiagramon megadott metamodellnek a következő objektumdiagramon lévő modell?

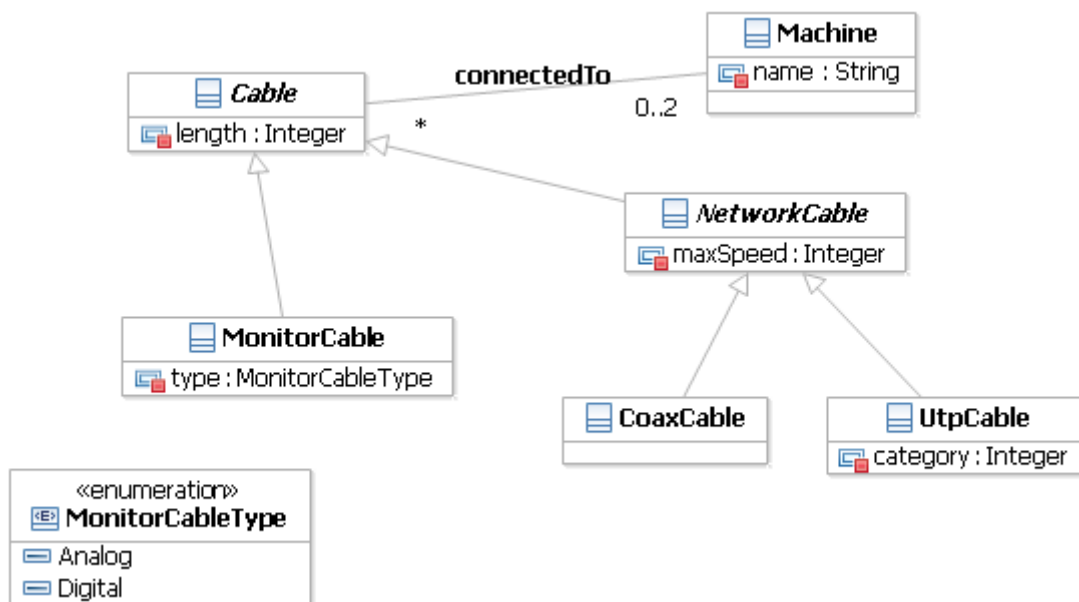


3.2 Fájrendszer jogosultságok

Készítsen egy olyan UML osztálydiagrammal megadott metamodellt, mellyel a fájlokra vagy könyvtárakra beállított fájlrendszer jogosultságokat lehet leírni! Minden elemhez egy jogosultsági listát lehet rendelni. A lista egy eleme egy entitásból (felhasználó vagy csoport) áll, akire a jogosultság vonatkozik, és egy jogosultságból áll. A lehetséges jogosultságok a következők: nincs hozzáférés, olvasás, írás és teljes hozzáférés.

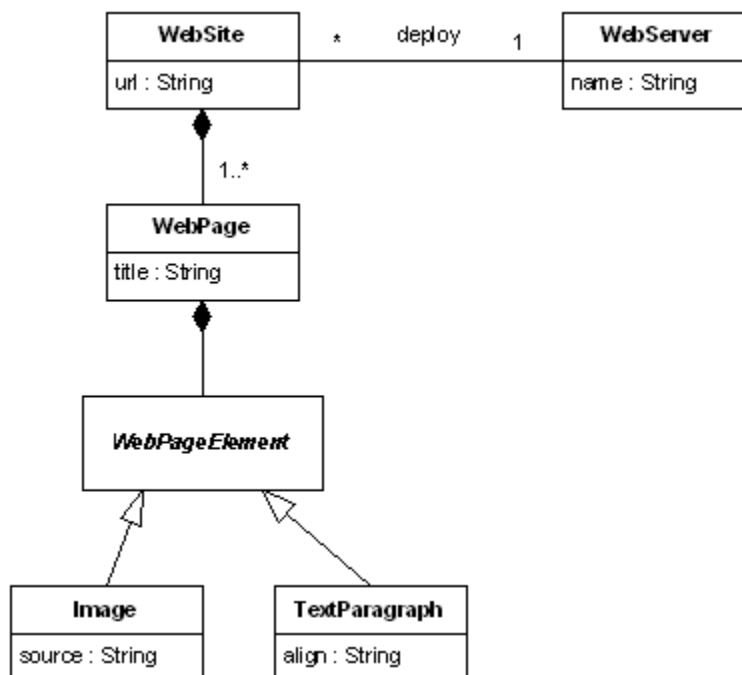
3.3 Hálózati kábelek

Rendet kéne rakni a fiókokban elfekvő és a számítógépekbe bekötött rengeteg számítógépes kábel között. Van két gépünk, mindegyik bekötve a hálózatba és monitorral ellátva, és a rendszergazda úgy tippeli, hogy a raktárban még van legalább egy pót monitorkábel és két hosszabb UTP kábel, sőt még mintha egy koax kábel is maradt volna a hőskorból. Készítsen mindezek dokumentálására egy UML objektumdiagramot az alábbi UML osztály diagramnak megfelelően!



3.4 Webhelyek ábrázolása

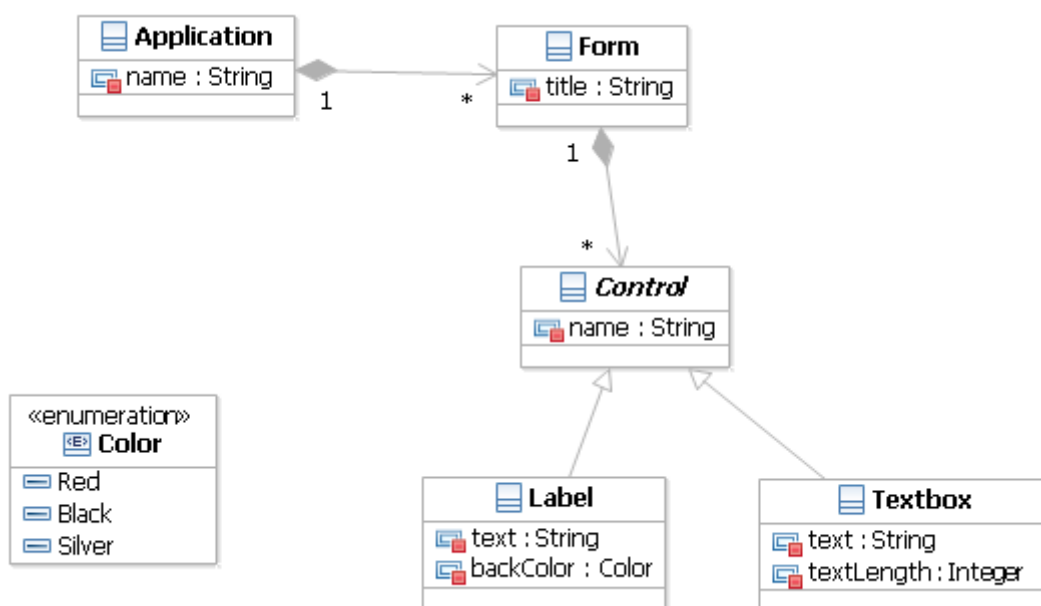
Adott a következő, webhelyeket leíró UML osztálydiagram:



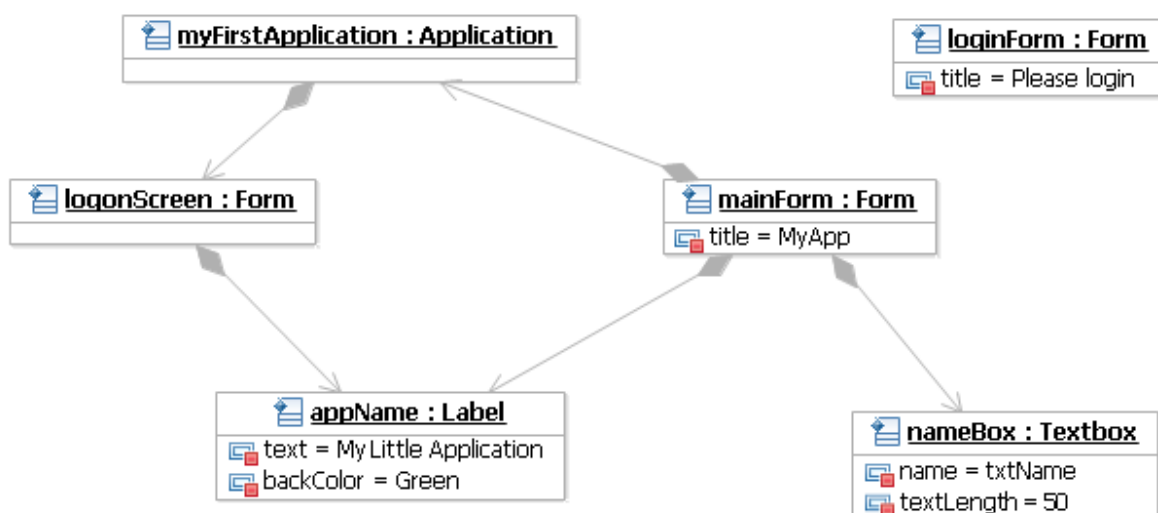
Készítse el egy olyan UML objektumdiagram példányát ennek, ami egy két lapból álló webhelyet ábrázol, amiben minden oldalon legalább két elem van.

3.5 UI modellezés

A programunk felületének leírásához a következő metamodelt használjuk.



Sikerült-e a fenti UML osztálydiagramon megadott modellnek egy típushelyes példányát megalkotnunk a következő objektumdiagramon?

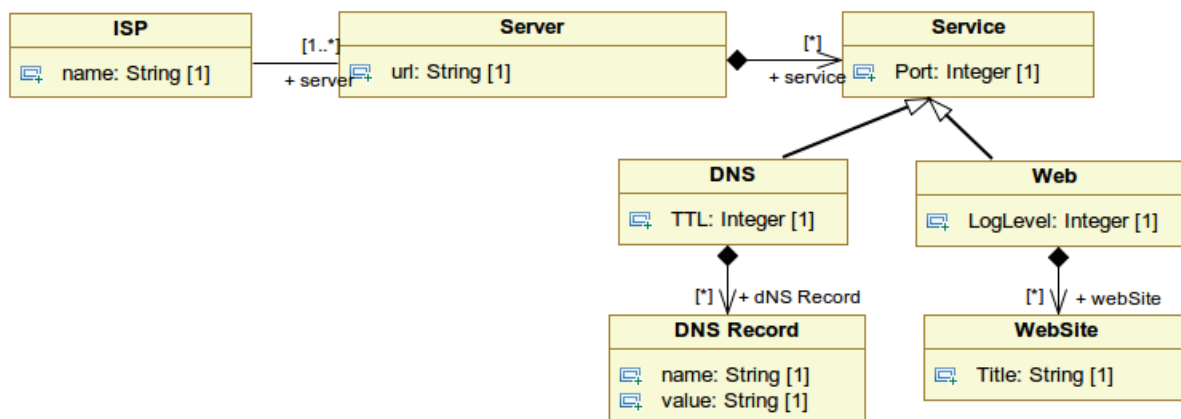


3.6 Alkalmazások nyilvántartása

Készítsen egy olyan metamodelt, és ábrázolja egy UML osztálydiagramon, ami számítógépre telepített alkalmazásokat tart nyilván. Az alkalmazásokhoz megadható a nevük és a verziójuk, valamint, hogy a számítógép melyik meghajtójára telepítettük (a meghajtókat a betűjelükkel azonosítjuk). Tároljuk továbbá, hogy melyik alkalmazásnak ki a gyártója, és mi a gyártó weboldala. A kapcsolatoknál ábrázolja azok számosságát is!

3.7 Internetszolgáltatók

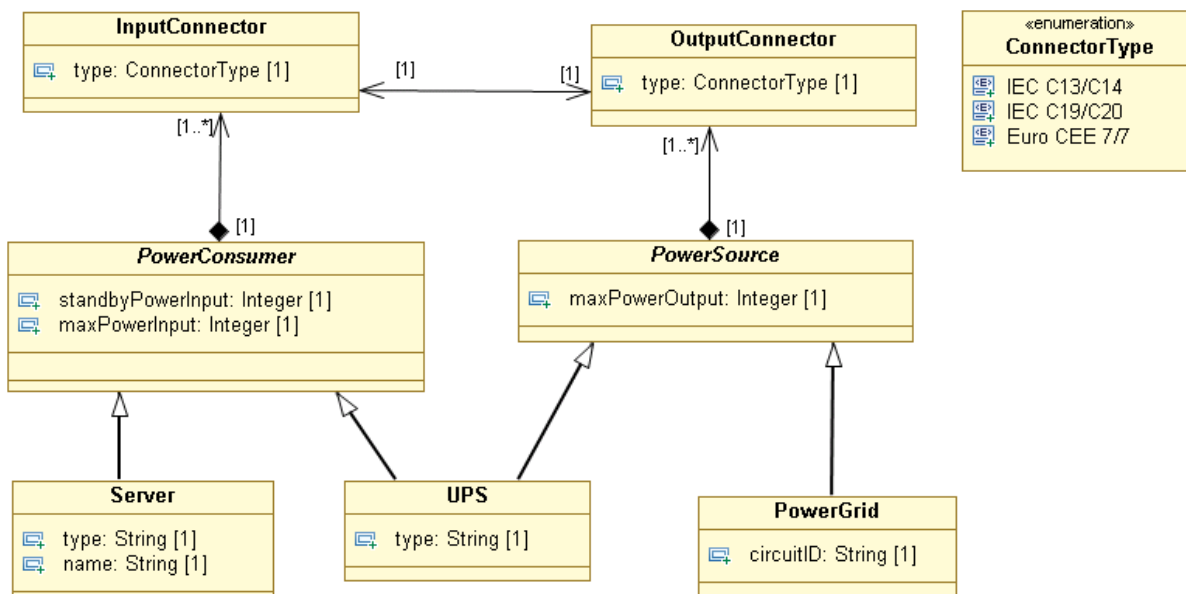
Adott a következő, internet szolgáltatót és szolgáltatásait leíró UML osztálydiagram:



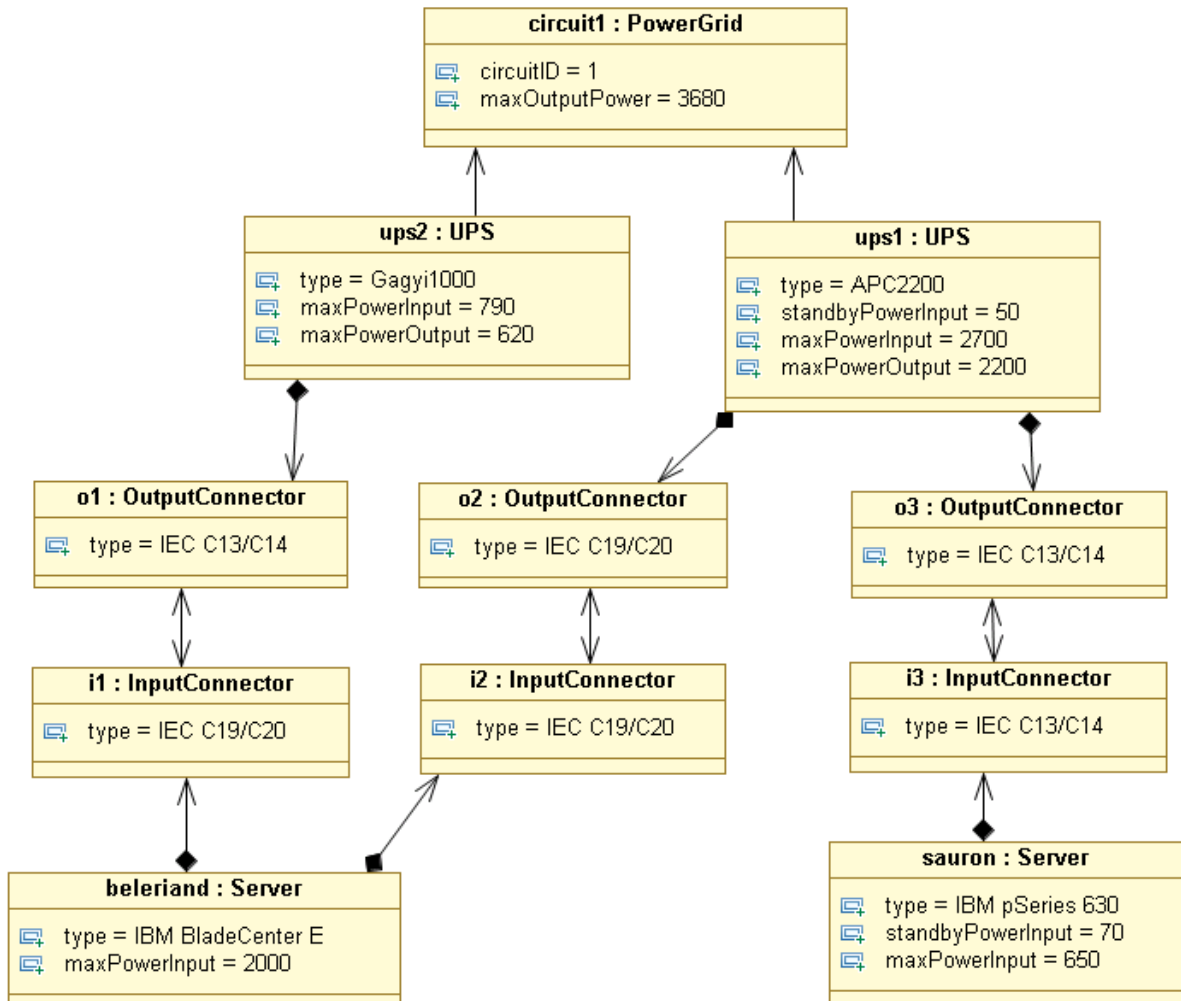
Készítse el egy olyan UML objektumdiagram példányát ennek, ami modellez egy internetszolgáltatót, melynek legalább két kiszolgáló szervere van, melyek közül az egyik DNS, a másik DNS és Web szolgáltatást is nyújt. A DNS szerverek minimum 1 bejegyzést, a web szerverek minimum 2 weboldalt szolgáltatnak ki.

3.8 Tápellátás

A szerverszobát behálózó vezetékek közül a gépek tápellátását biztosító erősáramú kábelek bekötésének dokumentálására valamint vizsgálatára a következő metamodelt dolgoztuk ki:



Típushelyes példány-e a következő modell a fenti metamodelleknek? Ha nem, adja meg, hogy milyen hibák találhatóak a modelltípusban!



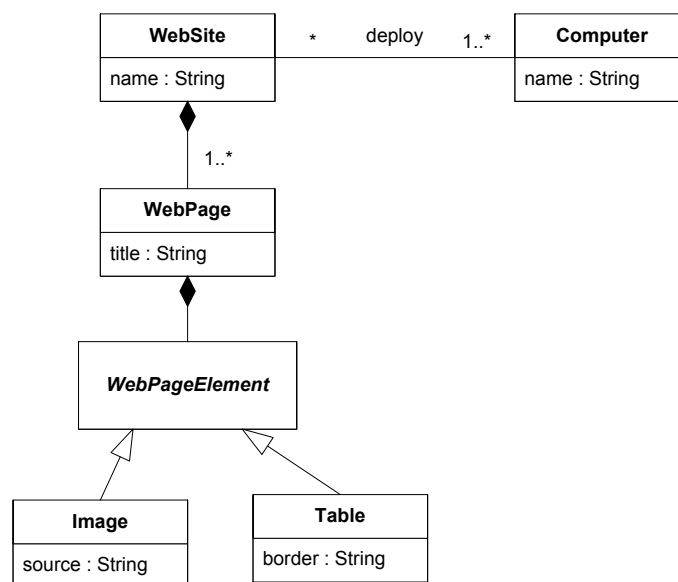
Azon túl, hogy a modell típushelyes példánya legyen a metamodellnek, szeretnénk a bekötés helyességét is vizsgálni. Határozzon meg legalább két – metamodellben nem feltétlenül szerepeltethető – feltételt, amit mindenképpen érdemes lenne ellenőrizni egy erősáramú hálózat üzembiztos működése érdekében!

4 Korábbi vizsgafeladatok

A következő fejezetben korábbi vizsgafeladatokat gyűjtöttünk összes, hasonlóakra lehet számítani.

4.1 Fürtök bevezetése

Adott a következő metamodel, mellyel IT infrastruktúrák egy részletét lehet leírni:



1. Módosítsa úgy a metamodelt, hogy bevezeti a fürt fogalmát! A fürt legalább egy számítógépből áll, és külön neve és IP címe van. (5 pont)
2. Készítsen egy olyan példány modellt, melyben egy két lapból álló webhelyet egy két csomópontból álló fürtre telepítünk! Minden weblap legalább egy elemet tartalmazzon! (15 pont)

4.2 Háttértárak modellezése

1. Készítsen egy olyan metamodelt mellyel leírhatóak a háttértárak következő alapfogalmai és a közöttük lehetséges kapcsolatok: merevlemez, partíció, fájlrendszer! Néhány jellemző attribútumot is vegyen fel! A partíciók esetén nem kell figyelembe venni a PC partíciós tábla sajátosságait (4 elsődleges, kiterjesztett, C/H/S címezés), de egyéb általános alaptulajdonságokat (sorszám, bootolhatóság, kezdőcím, méret, típus) jelölje! A metamodel alkalmas legyen többféle fájlrendszer megkülönböztetésére is! (5 pont)
2. Egészítse ki a metamodelt, hogy logikai kötetek leírására is alkalmas legyen! A logikai kötetkezelő képes pillanatkép készítésére is, így az ehhez szükséges alapfogalmak is jelenjenek meg a metamodelben! (7 pont)
3. Készítse el a következő rendszer modelljét az imént kidolgozott metamodel példányaként. Egy gépben egy 36GB-os SCSI és egy 500GB-os SATA merevlemez található. A SCSI merevlemez elején egy 10GB-os rendszerpartíció található, melyen egy RedHat Enterprise Linux foglal helyet. A SCSI lemezen található maradék hely és a teljes 500GB-os merevlemez pedig egy kötetcsoporthoz tartozik, melynek neve „StoreVG”. A StoreVG-n definiált 2 darab egyenként 150 GB-os logikai kötet mindegyikét egy-egy Windows Server használja (SAN-on keresztül, aminek nem kell megjelennie a modellben). A logikai kötetek közül az egyikről készült egy pillanatkép 30 GB-os méretben. Feltételezheti, hogy a GB-ot mindenütt egyforma egységekben számolják. (8 pont)

4.3 Hálózati eszközök modellezése

- a) Készítsen metamodell t hálózati eszközök nyilvántartására. A számítógépeinkben különböző sebességű és gyártójú hálózati kártyák lehetnek. A számítógépek hálózati kábelekkel vannak összekötve, melyekről tárolni szeretnénk, hogy milyen típusúak. Végül az egyes gépek vagy cross-kábel vagy pedig switchek segítségével vannak összekötve. Vannak menedzselhető és nem menedzselhető switcheink is. Végezetül szeretnénk tárolni, hogy mik a számítógépek IP beállításai (IP cím, alhálózati maszk, DHCP használata, stb.). (10 pont)
- b) A fenti metamodellnek készítse el egy példány modelljét, ami a következő eszközöket modellezi. A server1 és server2 gépek egy nem menedzselhető switch segítségével vannak összekötve. CAT-5-ös UTP kábeleket és Gigabites Intel hálózati kártyákat használunk. A switch egy Gigabites 24 portos eszköz. C osztályú címeket használunk, a server1 IP címe 10.40.1.1, a server2 pedig a DHCP-től a 10.40.1.10 címet kapta. A server2-ben van ezen kívül van egy 100 Mbites Intel hálózati kártya, amit jelenleg nem használunk. (4 pont)
- c) A fenti metamodell nem tartalmaz néhány, a szerverekben használatos megoldást. Módosítsa a metamodell megfelelő részét (rajzolja le újra külön!) úgy, hogy tartalmazza a több portos hálózati kártyák és a NIC teaming fogalmát. Egy több portos kártyán több különböző port van, amik ugyanolyan sebességűek, de külön-külön MAC címmel rendelkeznek. Egy team két hálózati kártya összefogását jelenti, melyek kívülről egy kapcsolatként látszanak (tehát pl. ugyanaz az IP címük). (5 pont)
- d) A kiegészített metamodellben nem lehet minden jólformáltsági kényszeret pusztán osztálydefiníciókkal és multiplicitásokkal megadni. Mondjon egy ilyen kényszert! (1 pont)

4.4 Számítógép felépítésének modellezése

1. Számítógépek belső felépítésének modellezéséhez készítse el egy metamodell t! A számítógépben van egy alaplap, melyről szeretnénk tárolni a chipsetének típusát. A számítógép processzoráról a frekvenciáját akarjuk nyilvántartani. A memóriák kapcsán a memóriák méretét és típusát akarjuk feljegyezni, valamint azt, hogy melyik modul melyik memóriafoglalatban van. A gépben SATA vagy SCSI merevlemez-vezérlők lehetnek, az ezekre kötött merevlemezekről azok méretét és fordulatszámát tároljuk. Minden elem kapcsán tároljuk még a gyártót és a termék azonosítóját. (10 pont)
2. Készítsen el egy példánymodell t, ami a következő számítógépet reprezentálja: Intel Core 2 Duo 2.0 GHz-es CPU, 2x2GB Kingston DDR3-800-as RAM (az egyes és hármás memóriafoglalatban), Intel DP45SG alaplap Intel P45 Express chipsettel, és két 500 GB-os Samsung SATA merevlemezrel. (5 pont)
3. Egészítse ki a metamodell t, hogy a következő szerverekben használt funkciókat is meg lehessen benne adni: többprocesszoros gépek, ECC-t (error checking code) támogató memóriamodulok használata, memory mirroring (azaz megadható, hogy az egyik memóriamodul tartalmát egy másikba tükrözze az alaplap). (5 pont)

4.5 Címtárak modellezése

Egy tanácsadó cég különböző vállalatok számára központi címtárak kiépítésével foglalkozik. A kiépítés folyamatának automatizálásához egy megfelelő metamodell tervezését tűzték ki célul, melynek segítségével a megrendelőik igényeit precízen is le tudják majd írni.

A céges infrastruktúrák lényeges elemei az azt alkotó számítógépek és a közöttük futó hálózati kapcsolatok. Ezek a kapcsolatok lehetnek közvetlen, vagy valamilyen hálózati eszközön, switchen vagy hubon keresztül történő összeköttetések. A switch esetében tárolni szükséges a menedzsment IP címét. Minden számítógépen különböző szolgáltatások futhatnak, melyeket autentikáció szempontjából a cég szerver és kliens osztályokba csoportosít. A szerver jellegű szolgáltatások közös jellemzője, hogy melyik

hálózati interfészen, milyen porton várják a kliensek csatlakozását. Ebbe a csoportba tartozik az Active Directory szolgáltatás, amit a tartomány teljes neve jellemez és az OpenLDAP szolgáltatás, amit a RootDN attribútuma ír le. A kliens csoportba tartozó szolgáltatások mindegyike kapcsolódik legalább egy központi címtár szolgáltatáshoz.

- a) Készítse el a metamodellt hogy a fent megfogalmazott modellezési céloknak eleget tegyen! (6 pont)
- b) A cég egyik ügyfele vegyesen Windows és Linux kliensekkel is dolgozik, melyeken különböző szolgáltatások futnak. A megrendelés egy Active Directory és egy LDAP címtár szolgáltatást is tartalmaz. Az előbbin alapulva működik a Windows fájlmegosztás autentikációja, a webservert pedig az LDAP szolgáltatást használja fel. Az Active Directory és a windowsos fájlserver külön gépen fut, a webservert és az LDAP szolgáltatást pedig egy harmadik, linuxos gépre lett telepítve. Készítsük el a konkrét megrendeléshez kapcsolódó példánymodellt a korábban megtervezett metamodell alapján! Jelenítse meg azokat az attribútumokat is, amiknek az értékét nem adtuk meg. (4 pont)

4.6 Open Compute szerverek

A Facebook néhány hónappal ez előtt az Open Compute Project nyílt forrású projekt keretében publikálta az oregoni Prineville-ben kialakított adatközpontjuk specifikációit. A fejlesztésnél a legfontosabb célként az energiahatékonyságot tartották szem előtt, és ennek megfelelően alakítottak minden részegységet az optimum eléréséhez. A sok egyedi fejlesztés mellett természetesen itt is nagyon hasonló a felépítés más adatközpontokhoz, azonban a meglévő modellekkel nem lehetne kényelmesen leírni a konfigurációt. Ezen probléma leküzdésére a jelen feladat a konfiguráció leírását lehetővé tevő metamodell kialakítása.

A rendszer alapegysége a szerver¹. A szervereket modellek alapján állítják össze. A szerver modell leírja, hogy az milyen lemezeket, milyen villamos tápegységeket, ventilátorokat és alaplapot tartalmaz. Az alaplapon találhatóak a CPU-k (a típussal azonosítjuk), I/O portok (lehet SATAII, Ethernet és USB) és memória (a típussal azonosítjuk). A szervereket három oszlopos szekrényekben (triplet) tárolják, minden oszlopban 30-at. Az esetleges áramingadozások és áramszünetek átvészelésére szünetmentes tápegységeket használnak, amelyek egyenként két triplet áramellátására képesek. A teljes adatközpont a párba állított triplettekből és az áramellátásukat biztosító szünetmentes tápegységekből épül fel.

- a) Készítsen el egy metamodellt, amely a fent leírt konfiguráció leírását lehetővé teszi! (4 pont)
- b) A Facebooknál kétféle szervermodellt használnak: az AMD és az Intel alapút. A szerver modellek adott paraméterekkel rendelkeznek (CPU, memória...), és ezek alapján állítják össze a szervereket. Az egyes szerverek alkatrészei legfeljebb a gyári számokban különböznek egymástól. Készítsen egy példány modellt a fenti metamodellhez amely leírja a következő képzeletbeli AMD szervermodellt.
- c) *FBAMD-001*: Open Compute Project AMF alaplap (két AMD Opteron 6100 sorozatú processzor helytel, 24 memória foglalattal, 6 SATAII porttal, 3 USB porttal), két Opteron 6132 HE processzor, 2 * 16GB RDIMM memória (DDR3 PC12800 ECC), 2 darab HD204UI típusú merevlemez. A merevlemezek a 0-s és 1-es SATA portra csatlakoznak. Az áramellátást egy Open Compute Project 450W tápegységgel biztosítják. (4 pont)
- d) A fenti példányhoz tartozó információk egy részét nem biztos, hogy meg lehet adni az a) feladatban megadott metamodellben. Mivel kéne még azt kiegészíteni, hogy minden fontos információ bekerülhessen? (2 pont)

4.7 Virtualizációs menedzsment alkalmazás

- a) Szeretnénk egy saját alkalmazással betörni a virtualizációs piacra, amivel hypervisorokat és virtuális gépeket lehet platformfüggetlenül menedzselni. Ehhez viszont először át kéne látni, hogy

¹ A projekt dokumentációjában a szervereket nem említik, csak a bezáró egységként szolgáló keretre (chassis) hivatkoznak. Az egyértelműség kedvéért a feladat szövegében eltérünk ettől a konvenciótól.

milyen fogalmakkal kell dolgozni. Készítsünk tehát egy UML modellt, ami a szakterület legfontosabb elemeit áttekinti. Vannak hypervisor megoldásaink, amikről a verziójukat és a nevüket akarjuk tárolni. Jelenleg két implementációt támogatunk (VMware ESXi és Xen), ESXi esetén azt kell még tudni, hogy mi a menedzment interfész neve, Xen esetén pedig a dom0-ban futó operációs rendszer típusát. A rendszerben ezen kívül vannak virtuális gépeink, amik valamilyen operációs rendszert vagy rendszereket futtatnak (az operációs rendszert a gyártó és a verzió azonosítja), továbbá valamelyik hypervisor példányon futnak. A hypervisorok valamilyen fizikai gépre vannak feltelepítve. A fizikai és virtuális gépekről egyaránt a processzorok számát és a memória méretét akarjuk nyilvántartani. A virtuális gépekről tárolni kell továbbá, hogy hány és mekkora virtuális lemez tartozik hozzájuk. (6p)

- b) Készítsünk egy példány modellt a fenti metamodellhez, amiben legalább két hypervisor és három darab virtuális gép van. (2p)
- c) Az alkalmazás új verziójában már a haladó funkciókat is támogatni kell, egészítsük ki a modellt ennek megfelelően. Virtuális gépeket lehet sablonból létrehozni. A hypervisorokat lehet fürtökbe szervezni, ilyenkor opcionálisan be lehet kapcsolni a hibatűrés vagy erőforrás-kiegyenlítési funkciókat a fürtön. (2p)

4.8 Biztonsági mentések modellezése

- a) Biztonsági mentést tervezünk az infrastruktúránkhoz, melyhez az automatizálást elősegítendő egy metamodell készítése a feladatunk. Első körben „pull” típusú mentést szeretnénk végrehajtani, melyhez a megfelelő eszközt már ki is választottuk. Az eszköz alapeleme a biztonsági mentés kötet, melyhez tartoznak a különböző kliensek. A kötethez tároljuk a fájlrendszerbeli elérési útját, míg a különböző kliensekhez azok nevét, IP címét és azt az időinformációt, hogy mentést mikor kell készíteni róla (ez jellemzően egy napon belüli időpont). Minden klienshez tárolunk pillanatképeket, melyeknek rögzítjük a készítés dátumát és méretét. A pillanatképek lehetnek teljes mentésből származóak vagy inkrementális jellegűek. Ez utóbbi esetén tároljuk azt, hogy melyik korábbi pillanatképhez képest készültek. A mentéseket nem végezzük el minden esetben a kliens teljes fájlrendszerén, gyakran alkalmazunk tiltó listákat, melyek azon könyvtárak elérési útjait tartalmazzák, amik kimaradnak a mentésből. Tiltó listát lehet globálisan megadni a mentési kötethez, vagy pedig lehet kliensenként definiálni. (4 pont)
- b) A fenti metamodellhez készítsen el egy példánymodellt. A `/backupvolume` útvonalon elérhető mentési kötethez 3 kliens, a *yoda* szerver (192.168.5.23), a *luke* szerver (192.168.5.24) és a *vader* szerver (192.168.5.13) tartozik. Globálisan tiltjuk a `/proc`, `/sys` és `/dev` könyvtárak mentését, míg a *luke* szerveren a `/media/images` lokálisan is tiltva van. A kliensek mentése sorban 1:00, 2:00 és 3:00 időpontokban fut le minden nap hajnalban. A mentést ezen a héten indítottuk el, így eddig egy teljes mentés készült minden kliensről 2011. 06. 13-án és egy erre épülő inkrementális a tegnapi napon. (4 pont)
- c) A biztonsági mentésért felelős rendszerünket a továbbiakban ki akarjuk egészíteni windowsos hosztokhoz is használható „push” típusú mentés lehetőségével. Gondoljuk át, hogyan kéne kiegészíteni a metamodellt, hogy az ilyen jellegű mentéseket is támogassa! A teljes metamodellt nem kötelező újra lerajzolni, de egyértelműen jelöljük a kiegészítéseket. (2 pont)