

CIM Core Model

Paller Gábor

2004.10.07

CIM

- Common Information Model
- Általános formátum rendszerek leírására, mindenekelőtt (de nem kizárólag) menedzsment célból.
- A specifikáció magja a rendszerleírási koncepció, de az idők folyamán a CIM-hez egyéb specifikációk (pl. menedzsment protokollok) is kapcsolódtak
- A CIM-nek részei a Core Model és rendszerleíró modellek gyűjteményei (pl. CIM Application – alkalmazásmenedzsment, CIM Device – számítógépalkatrészek és perifériák, CIM Network – hálózati elemek). A kész modellek specializálásával könnyen le lehet írni bármilyen konkrét rendszert
- A szabvány gazdája a Distributed Management Task Force (DMTF)

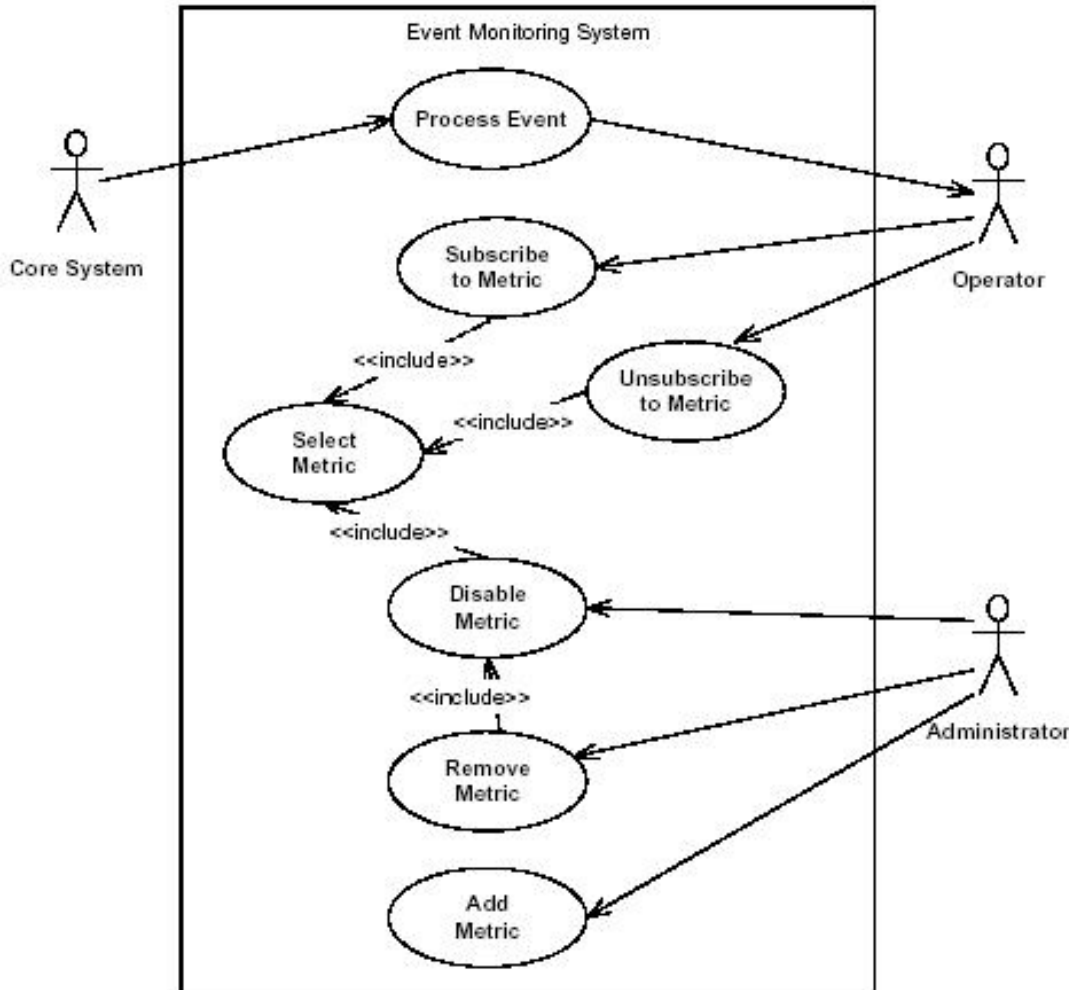
CIM koncepciók

- Objektum-orientált
 - Instancia-osztály
 - Öröklődés
 - Asszociációk (objektumok kapcsolata)
 - Örökölhető metódusok
- Core Model: minden CIM séma alapja
- A modellt UML-ben vagy a CIM saját objektumleíró nyelvén (MOF) írják le.

UML

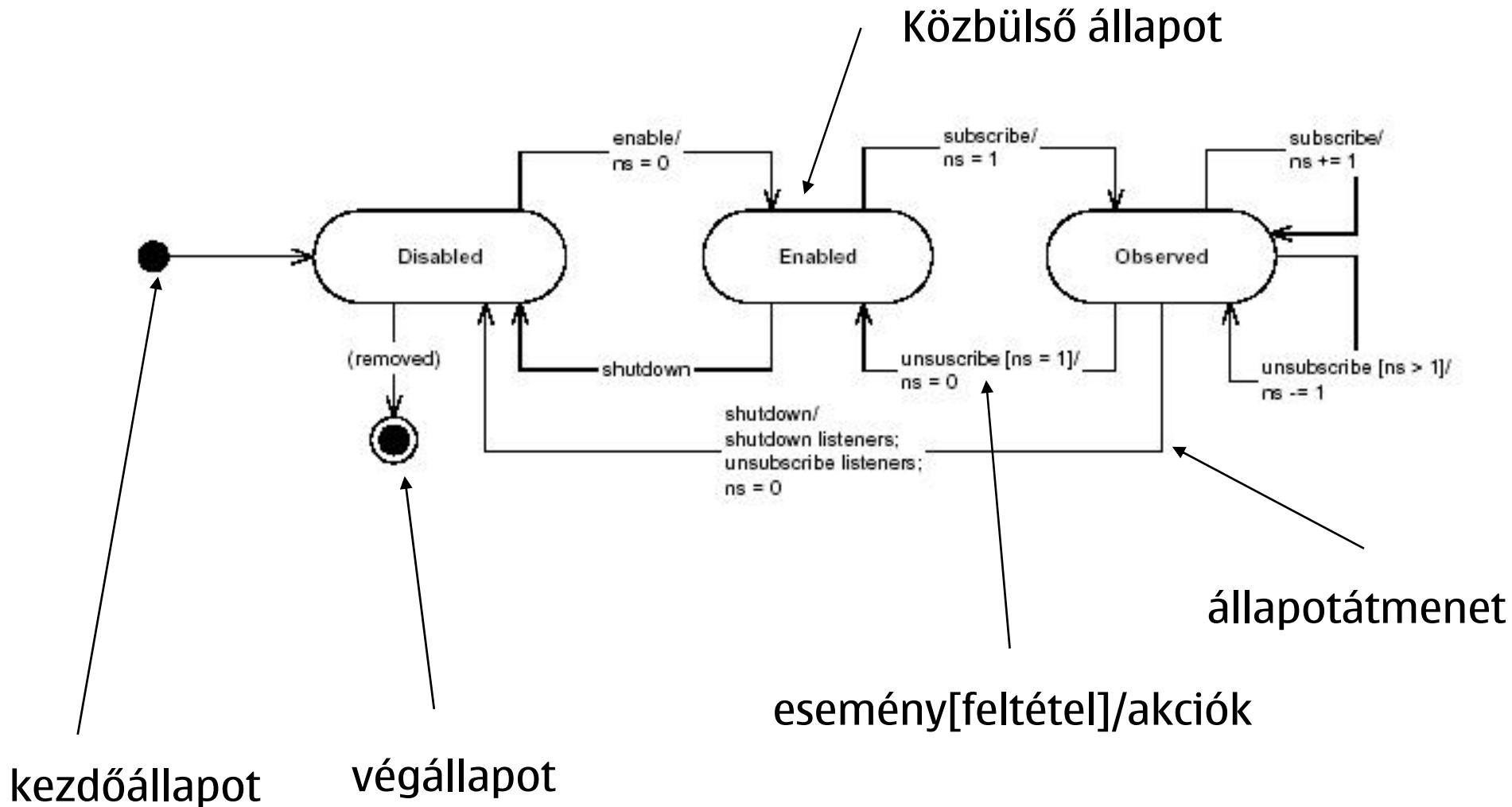
- Unified Modeling Language
- Cél: a vevő és a szoftvergyártó által is megérthető módon leírni a rendszer aspektusait
- Grafikus alapú, szabványos diagramokból építkezik. Gyakorlatilag egy képi nyelv.
- Részei:
 - use case diagram
 - osztálydiagram
 - viselkedési diagramok
 - állapotgép
 - aktivitásdiagram
 - szekvenciadiagram
 - kollaborációs diagram
 - implementációs diagramok
 - komponensdiagram
 - telepítési diagram

Use case diagram

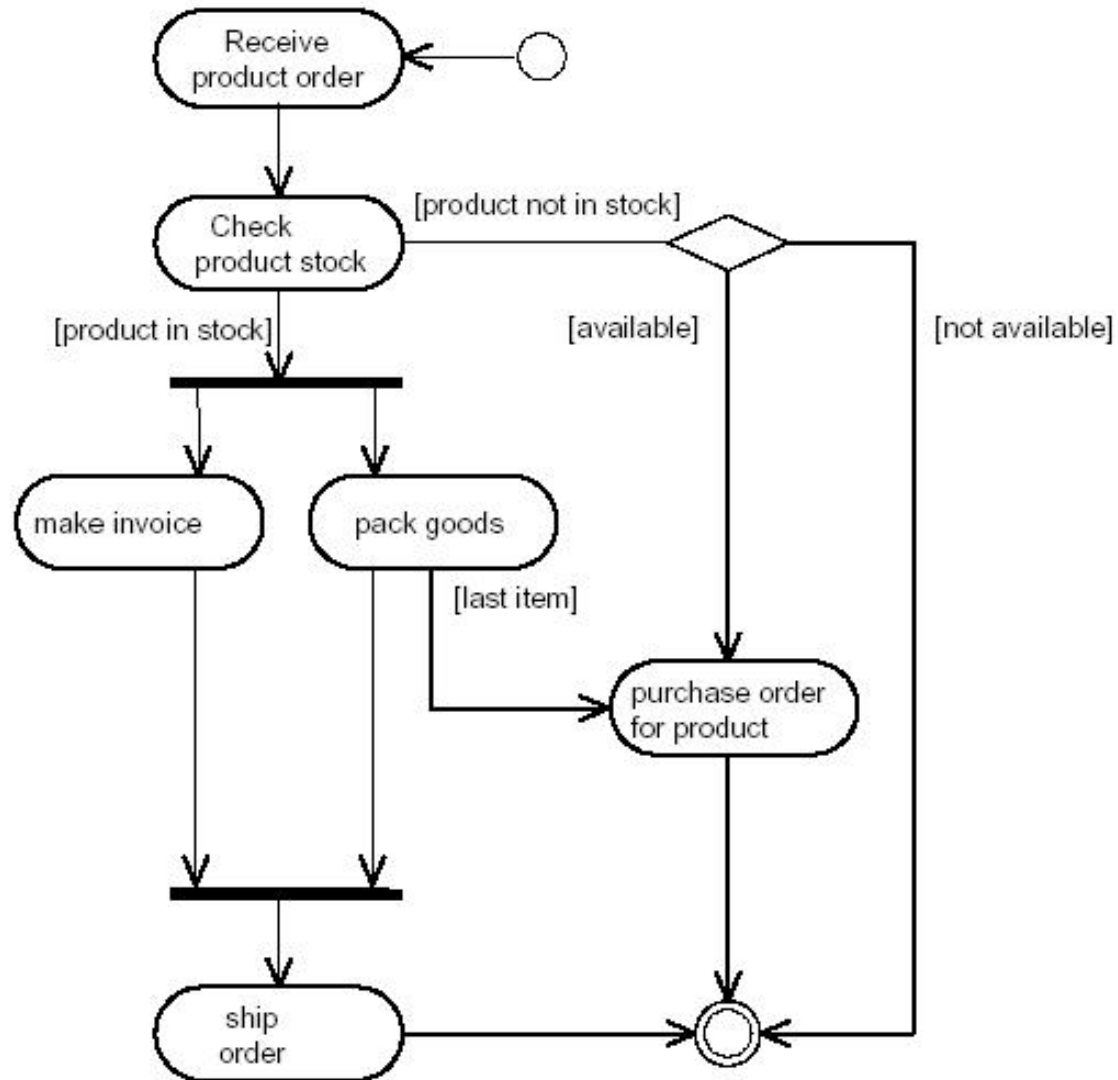


- Pálcikaember: szereplő (actor), ember vagy gép
- Ellipszis: felhasználói eset neve, amivel a szereplő kapcsolatban áll.
- Felhasználói eset: a rendszer használatának egy scénáriója. Részei: előfeltételek, szereplők, sztori, utófeltételek

Állapotgép diagram

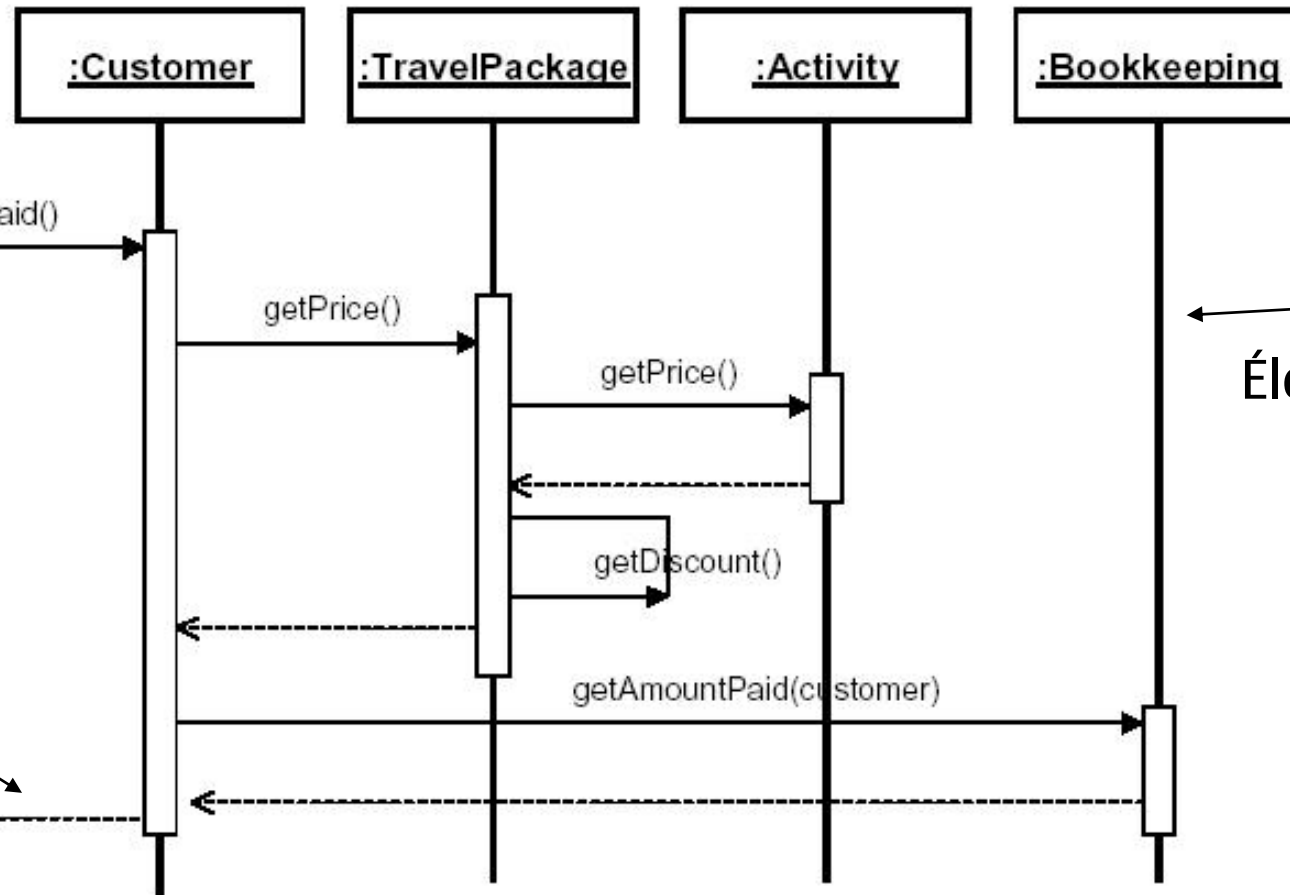


Aktivitás diagram



Szekvenciadiagram

objektum

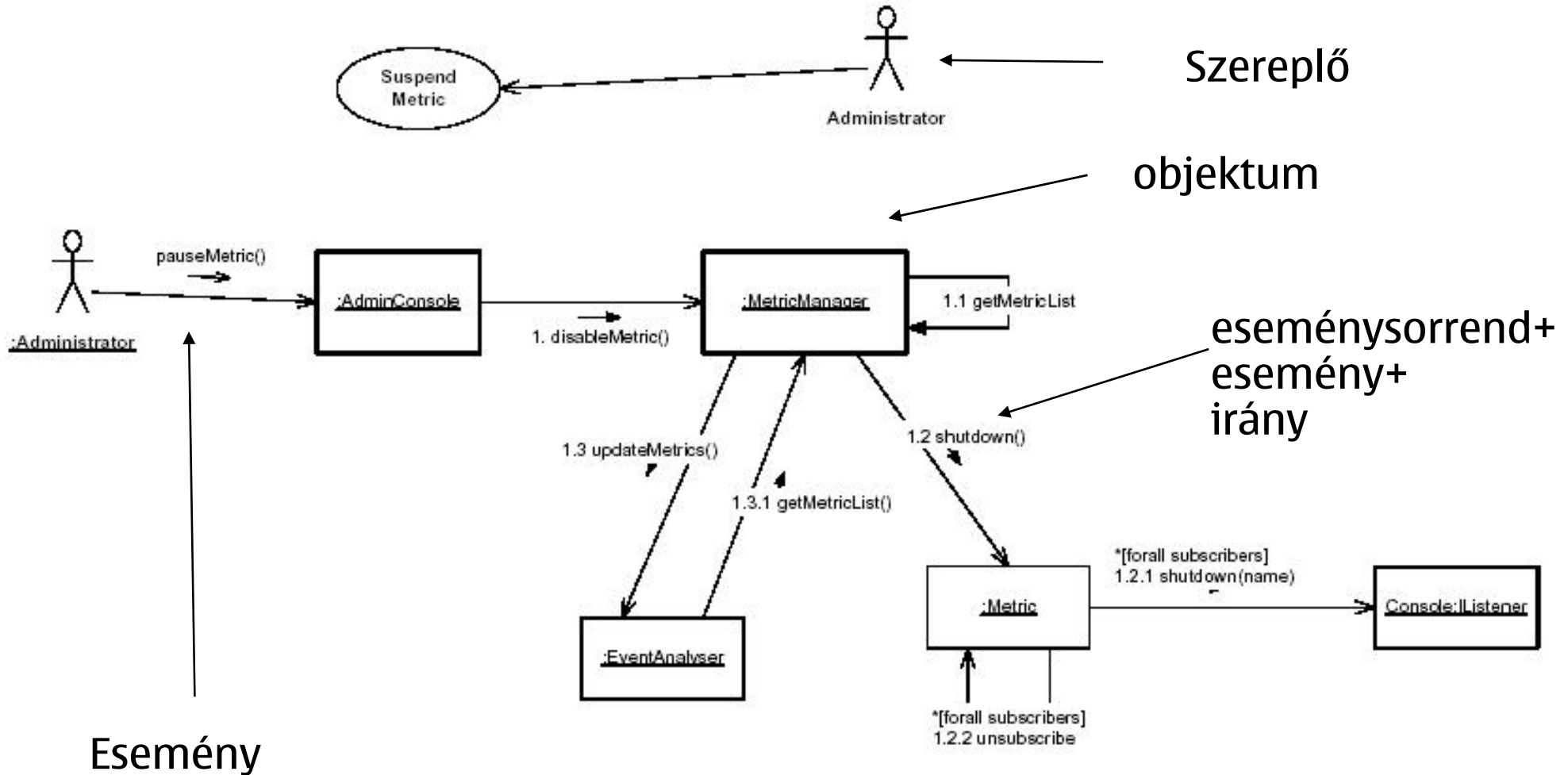


Életvonal

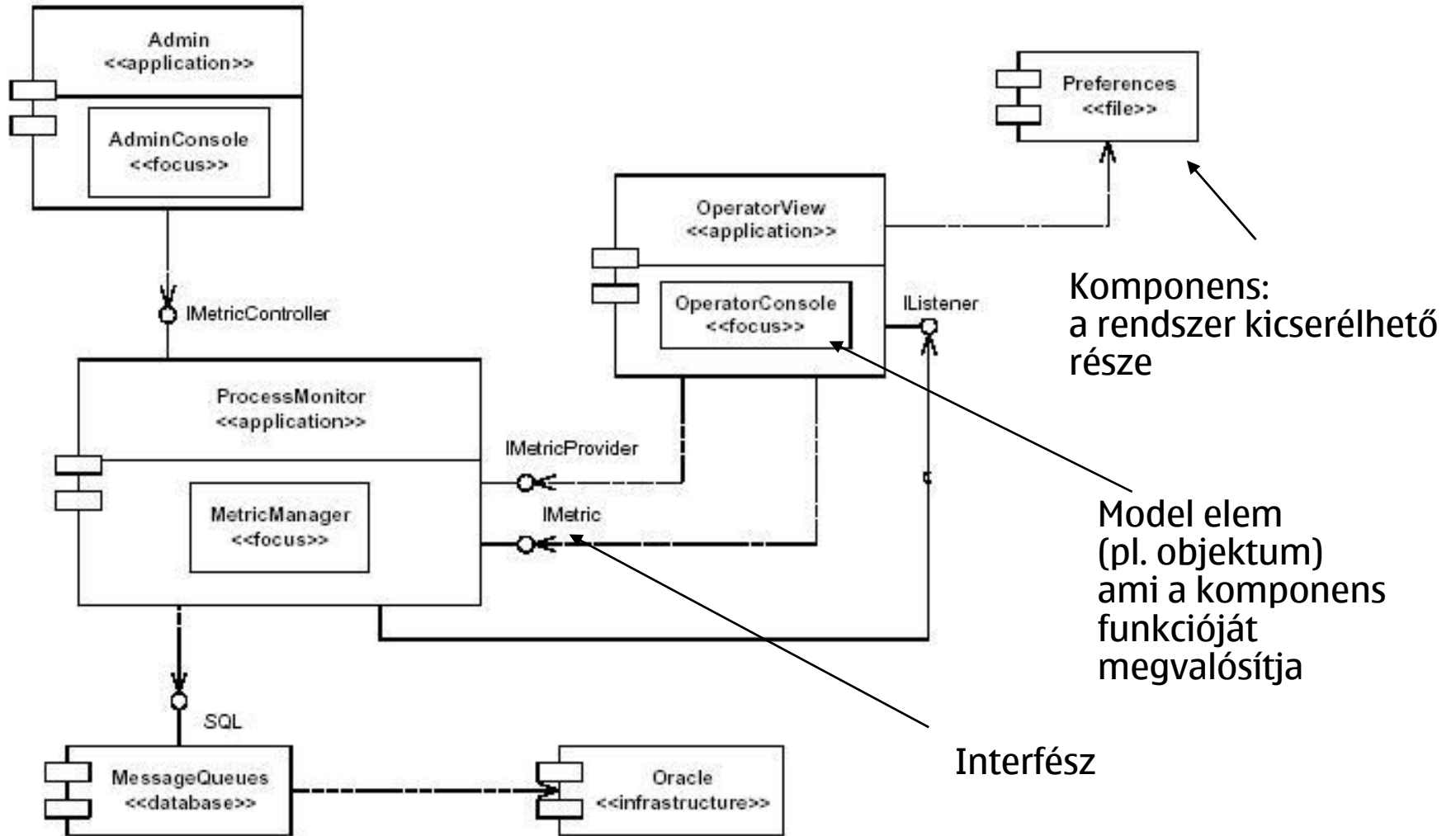
Esemény, kérés

Esemény, válasz

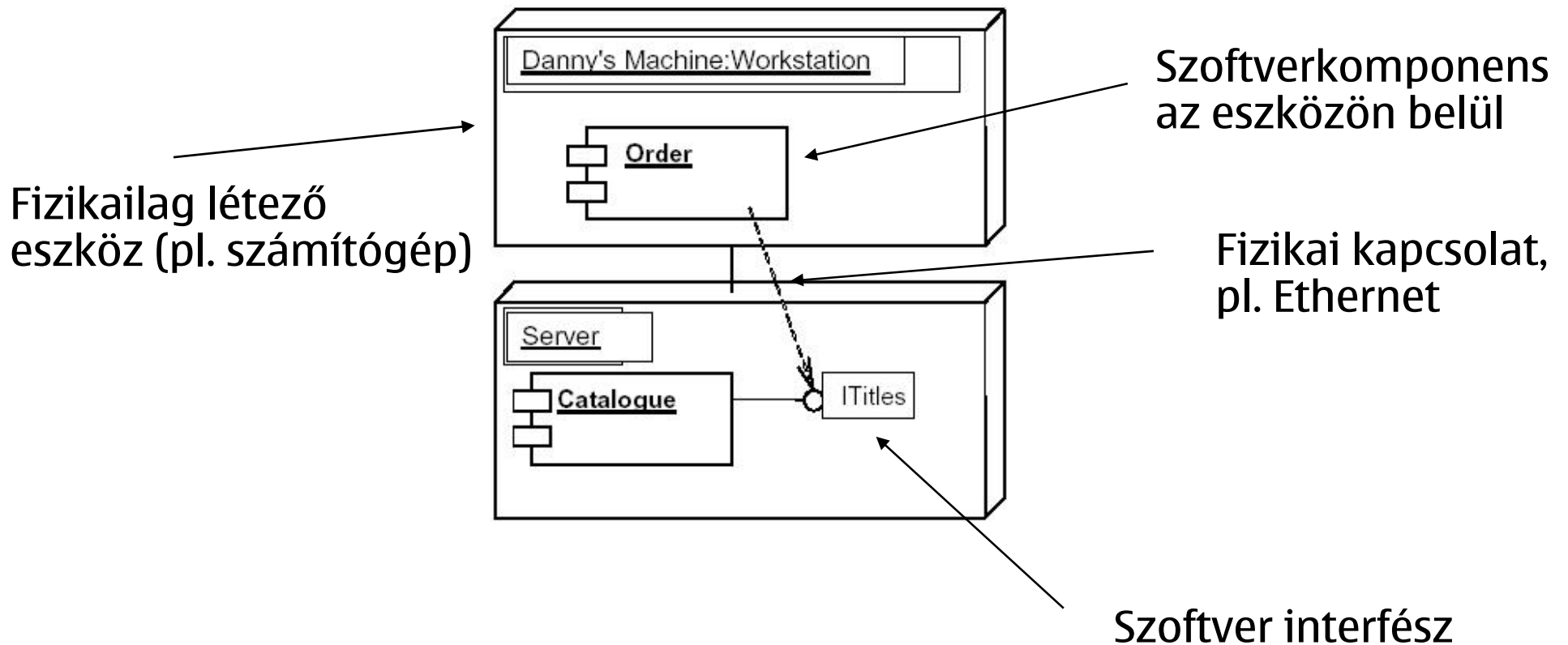
Kollaborációs diagram



Komponensdiagram



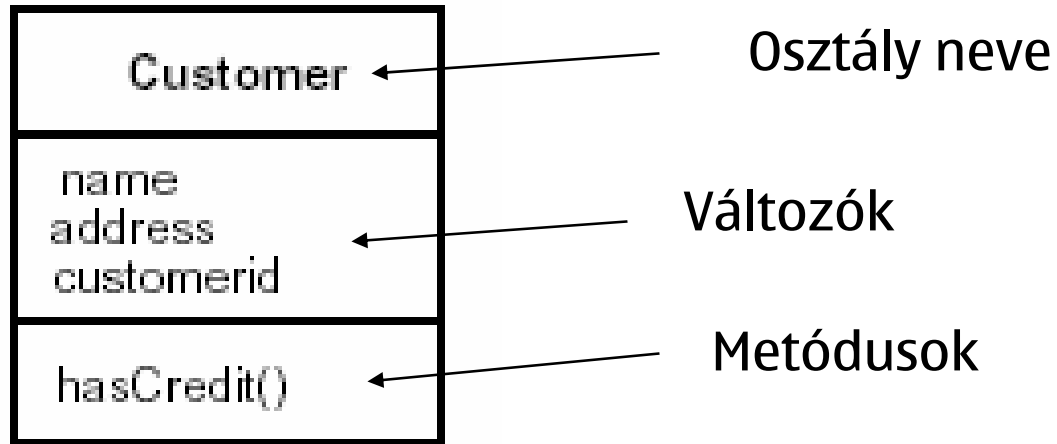
Telepítési diagram



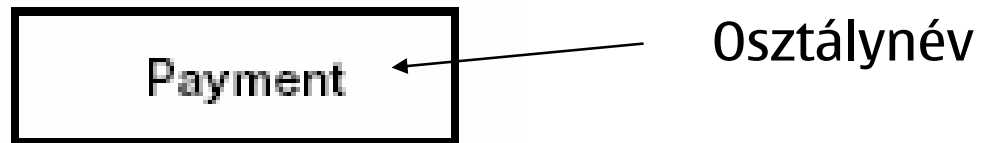
Osztálydiagram

- A legbonyolultabb és leggyakrabban használt UML diagram
- Osztályok és egyedeik leírására, valamint azok kapcsolatára való
- Igazában csak ez kell a CIM megértéséhez

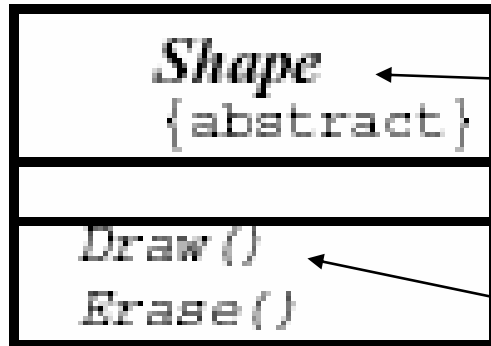
Osztályok



Vagy egyszerűen:

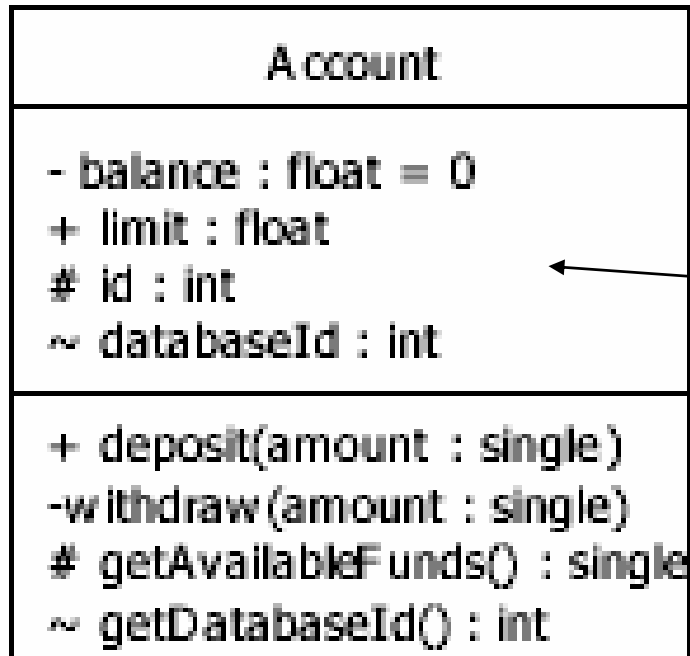


Variációk osztályokra



Absztrakt osztály (nem hozhatók belőle létre egyedek, de le lehet belőle származtatni).
Dőlt betű jelzi

Absztrakt metódusok (az osztály nem implementálja, de a leszármazottaknak implementálniuk kell, különben ők is absztrakt osztályok lesznek)



Láthatóság:

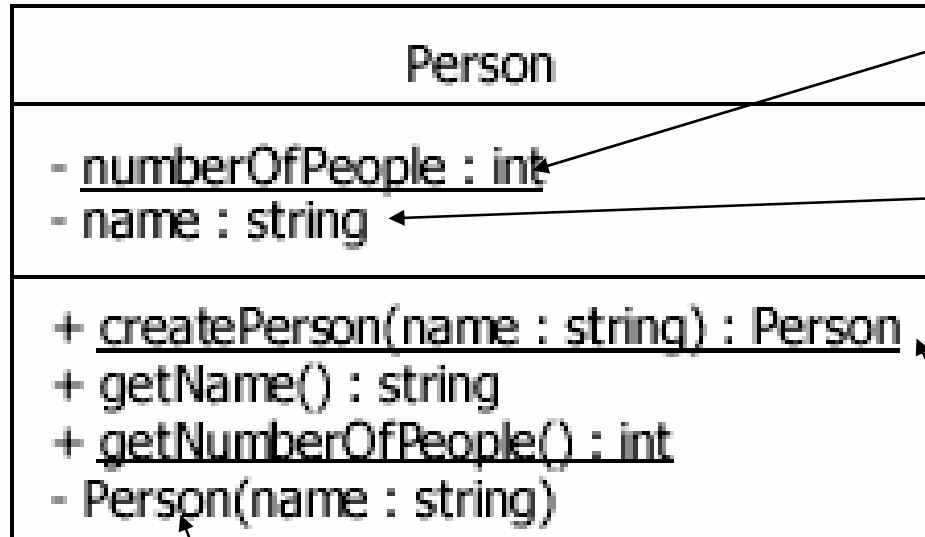
+ public (mindenki elérheti)

- private (csak az osztályból látható)

- protected (csak az osztályból és leszármazottaiból látható)

~ - package (csak azokból az osztályokból látható, amik ugyanabban az UML csomagban vannak)

Osztály és egyedváltozók



Osztályváltozó:
egy osztályhoz egy
van belőle.

Egyedváltozó: az osztály
minden egyedéhez van
egy példány

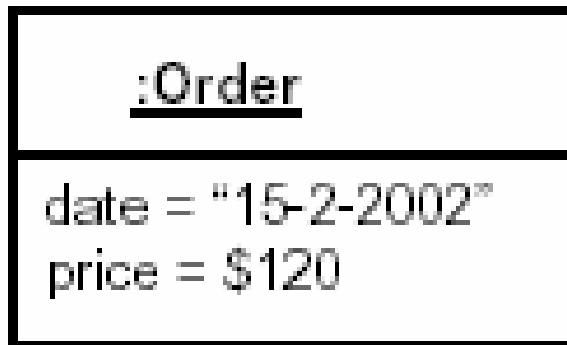
Egyedmetódus:
osztály és egyedváltozókat
is el tud érni.

Osztálymetódus:
csak osztályváltozókat
tud elérni

Osztály és egyedei

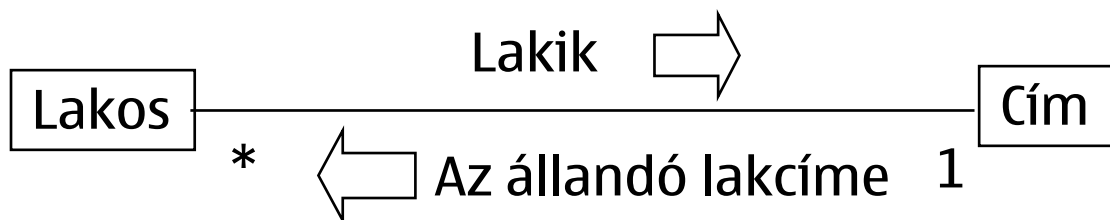


← osztály



← "Order" egy egyede

Asszociációk



Megfejtés:

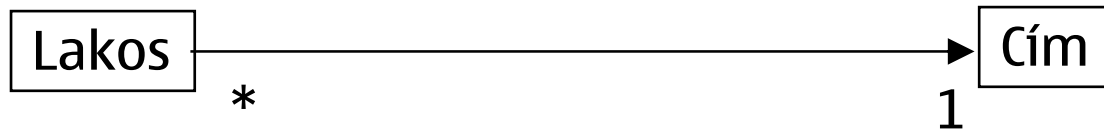
Lakos lakik a címen

Cím az állandó lakcíme a lakosnak

Egy címhez több lakos tartozhat

Egy lakosnak csak egy lakcíme lehet

Asszociációk (2)



Az asszociációnak lehet iránya.

Ez lehetséges navigációs irányt jelent

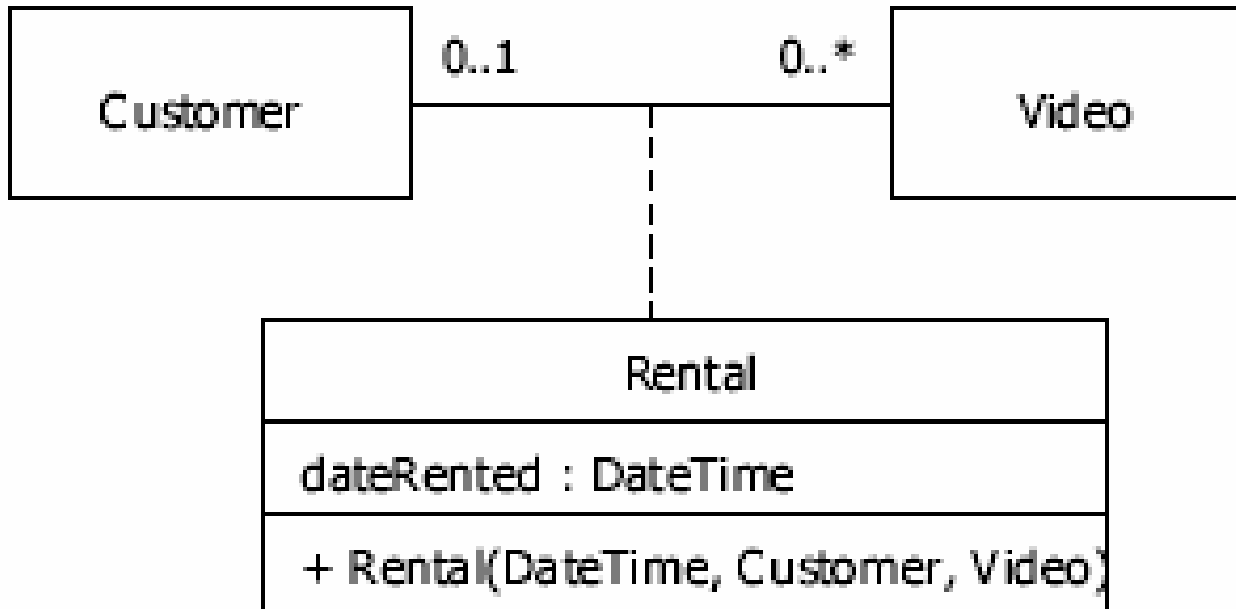
Megfejtés:

Lakos lakik a címen

Egy címhez több lakos tartozhat

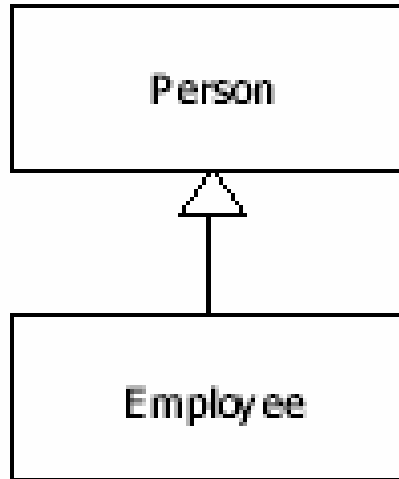
A cím objektum azonban nem tartalmaz információt a lakosokról, visszafelé navigálni nem lehet.

Asszociációs osztályok

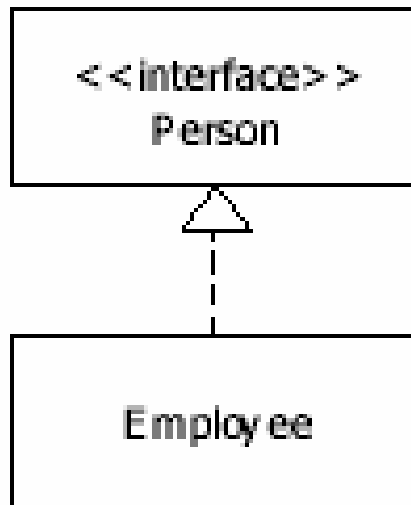


A **Customer** és a **Video** között a **Rental** osztály egyedei teremtenek kapcsolatot. A **Customer** **Rental** egyedeket tárol és a **Rental** egyedek hivatkoznak a **Video** egyedekre.

Leszármazás és megvalósítás

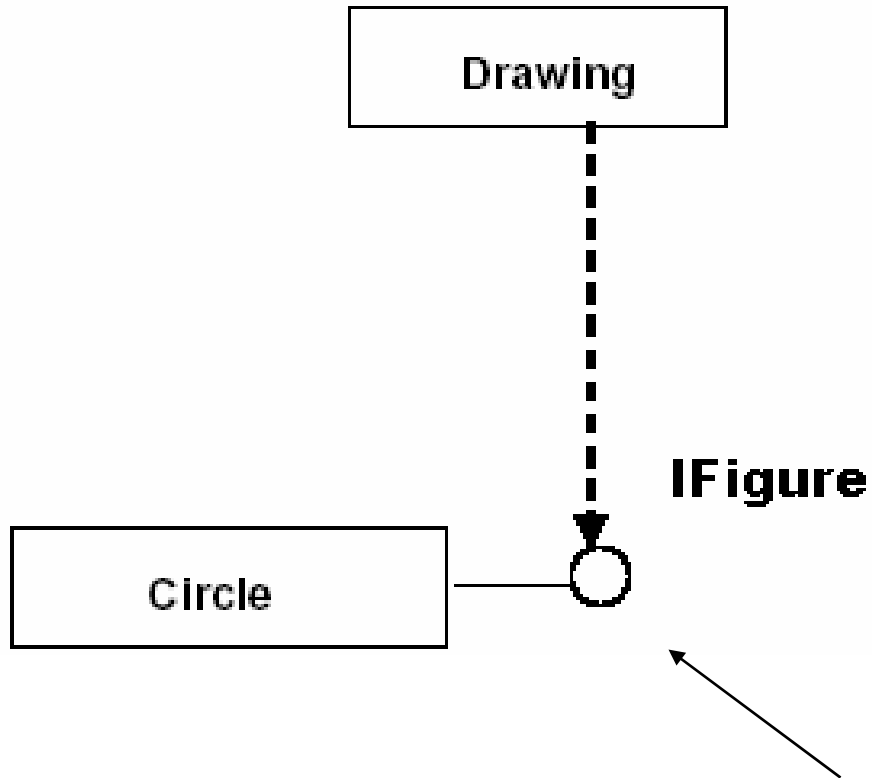


Employee osztály a Person osztály leszármazottja



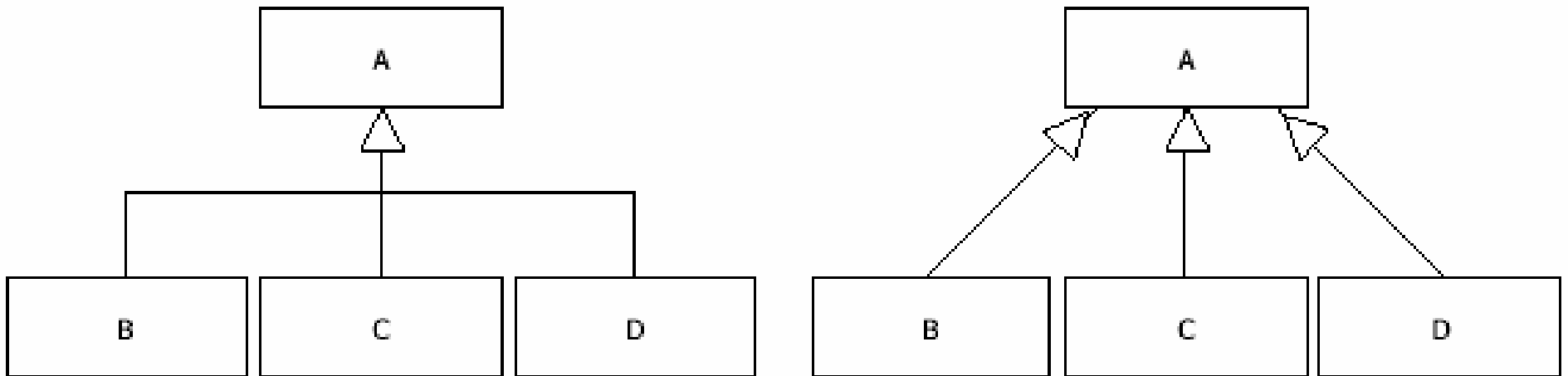
Employee osztály megvalósítja a Person interfészt. Az interfész előírja, milyen attribútumoknak és metódusoknak kell lennie a megvalósító osztályban, de implementációt nem ad, azt a megvalósító osztálynak kell nyújtania.

Megvalósítás (2)



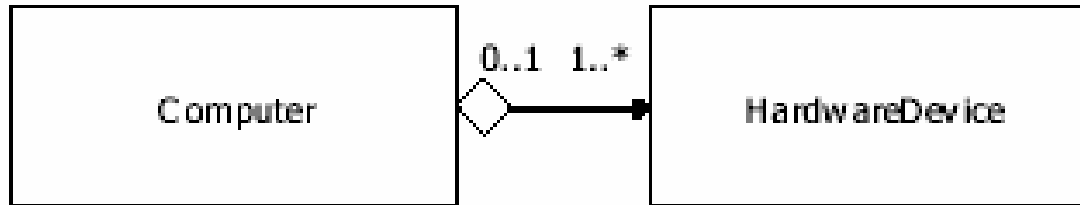
A Drawing objektum a Circle objektumot az IFigure interfészen keresztül éri el.

Lehetséges variációk az ábrákra

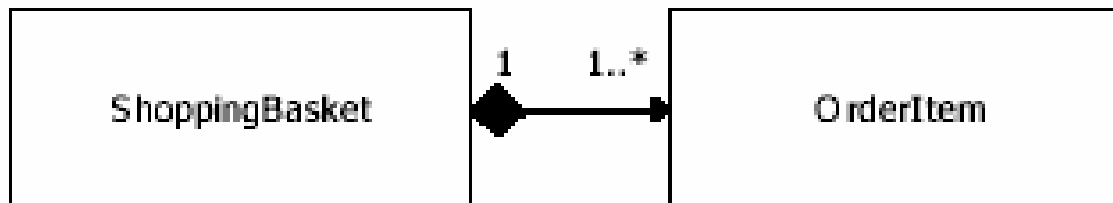


Ugyanazt jelenti

Aggregáció és kompozíció

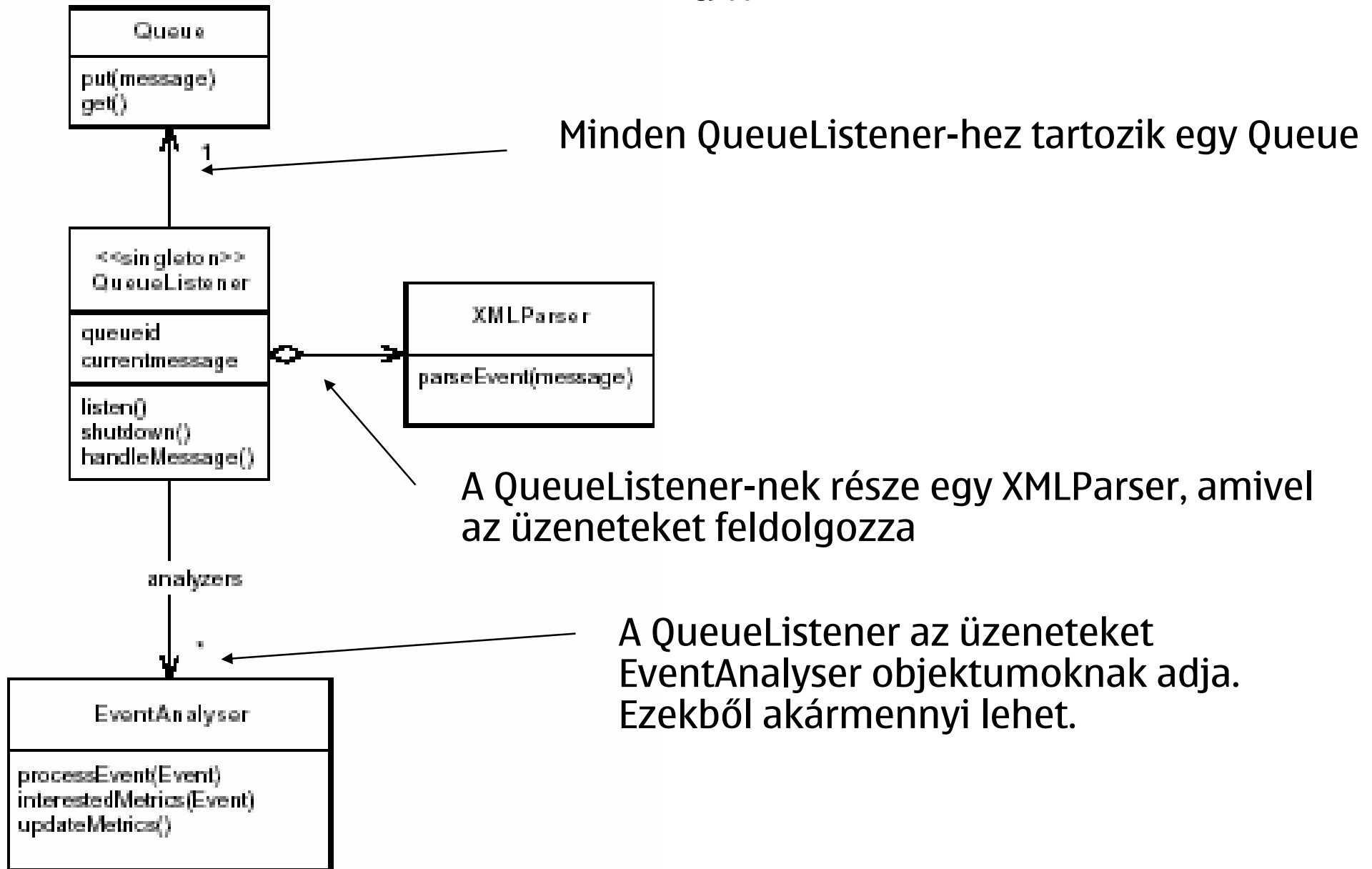


Aggregáció: a Computer egy vagy több HardwareDevice-ből áll. A Computer-t szétszerelhetem (megszüntethetem), de a HardwareDevice-ok megmaradnak, tehát a HardwareDevice-ok élettartama nem függ a Computer élettartamától.



Kompozíció: A ShoppingBasket OrderItem-ekből áll. Az OrderItem-ek csak egy ShoppingBasket részeként élnek, önállóan nincs értelmük, így élelciklusuk a ShoppingBasket élelciklusához kötődik.

1. nélda

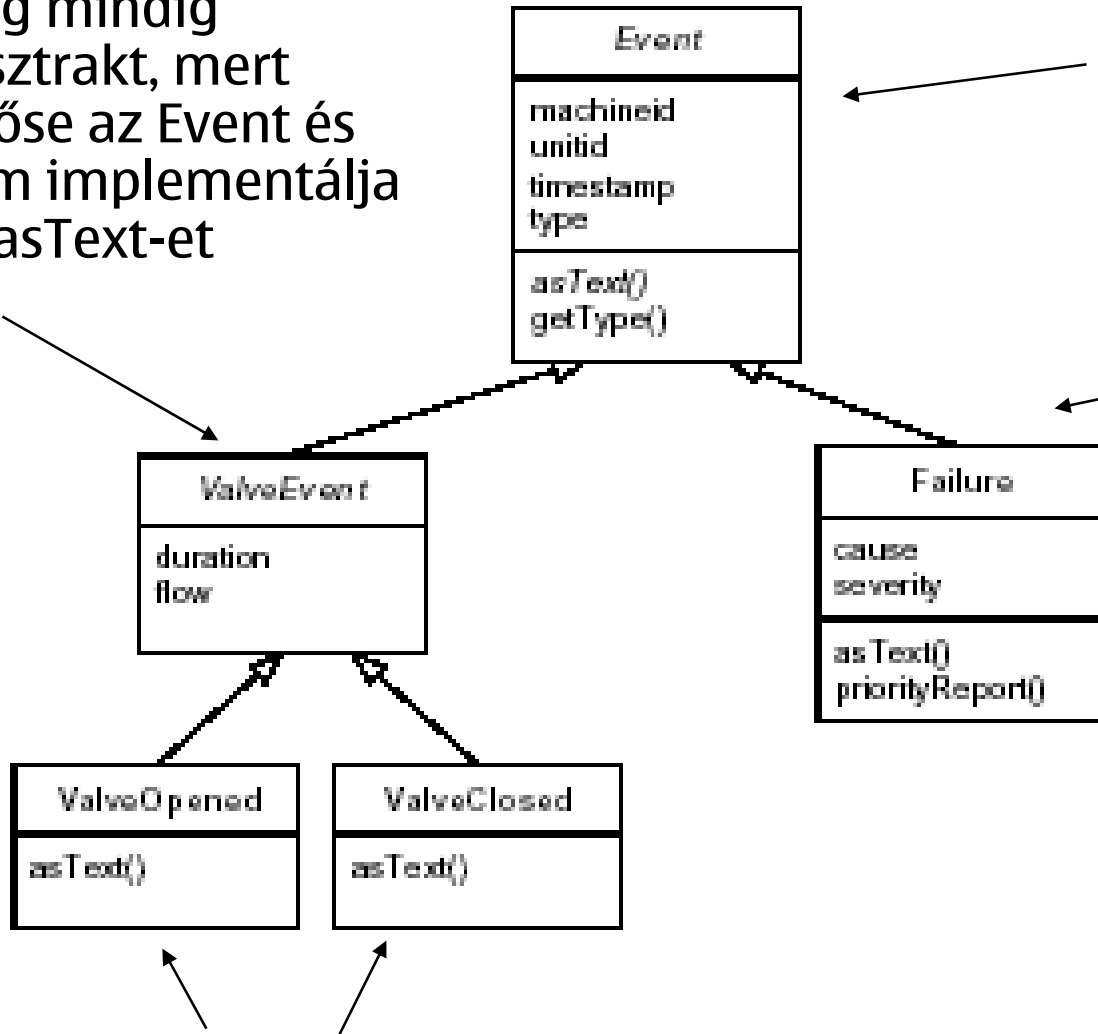


2. példa

A ValveEvent még mindig absztrakt, mert az őse az Event és nem implementálja az asText-et

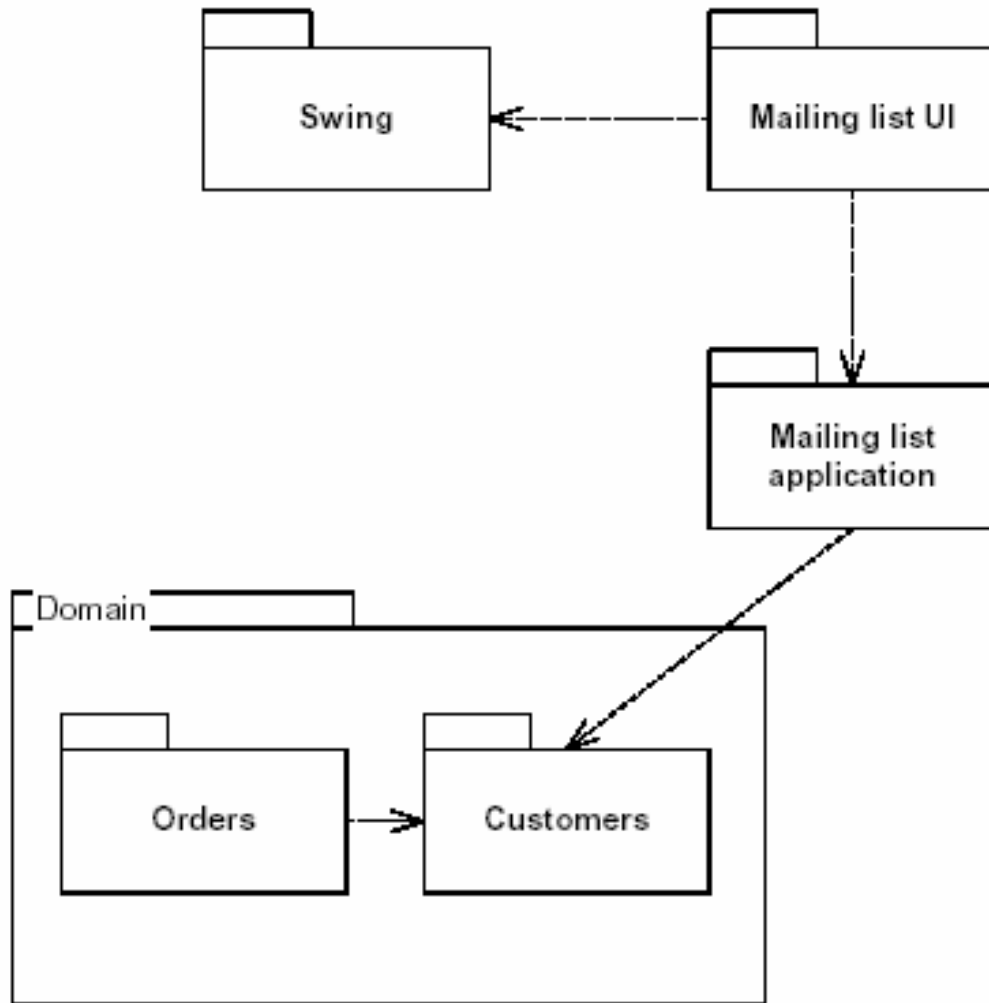
Az Event absztrakt osztály mert van egy asText() absztrakt metódusa

A Failure egy Event leszármazott, már nem absztrakt, mert implementálja az asText metódust



A ValveOpened és ValveClosed osztályok nem absztraktak, van asText metódusuk.

Csomagok

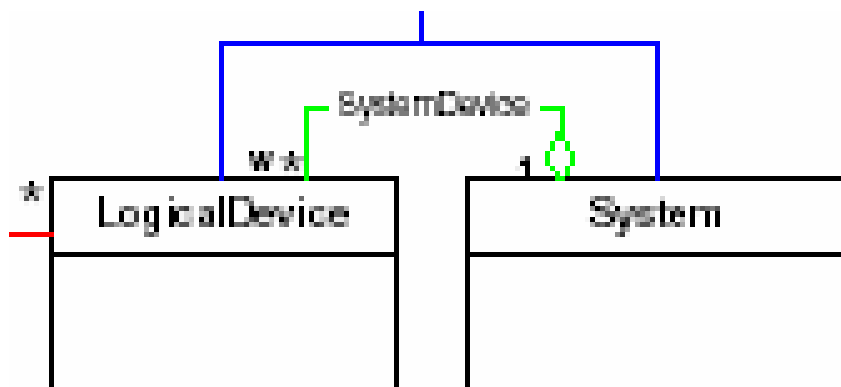


- A csomagok a model elemeinek egy csoportját gyűjtik egy szimbólum alá.
- A csomagok újabb csomagokból gyűjthetők.
- Relációk: <<access>>, <<import>>

Vissza a CIM-hez

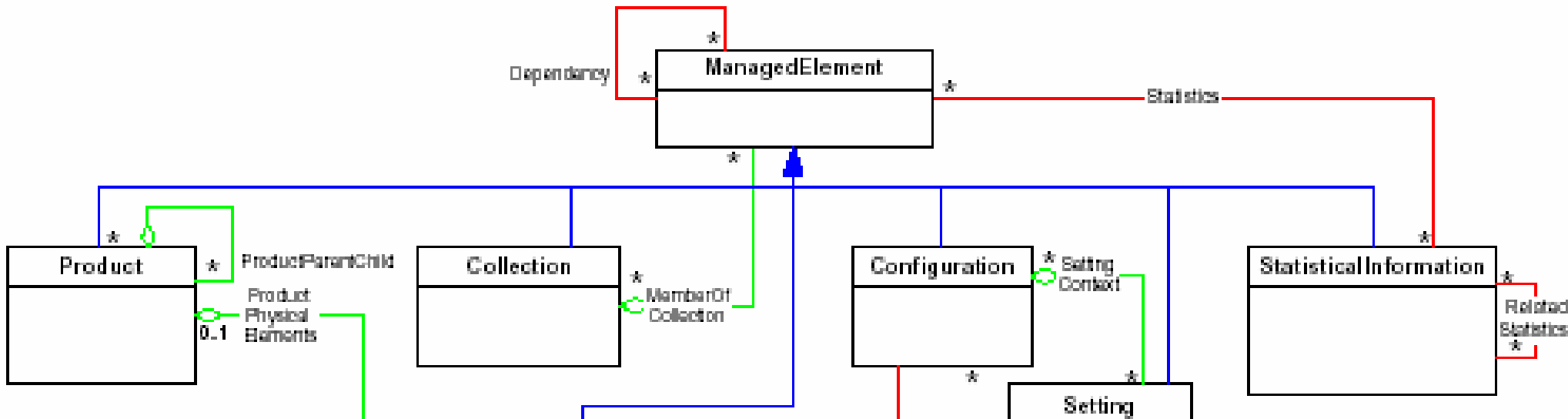
Gyenge asszociációk

- A CIM az UML-t a "gyenge asszociáció" fogalmával bővíti
- A gyenge asszociáció azt jelenti, hogy a csatolt objektum annak az objektumnak a névterében létezik, amihez csatolták.
- Példa: a számítógépben van egy c: meghajtó. Sok gépben van c: meghajtó, ezért ez a név csak akkor értelmes, ha megmondjuk, hogy a ws2347-es gép c: meghajtójáról van szó. A meghajtó objektum tehát gyenge asszociációval kötődik a számítógép objektumhoz.
- A gyenge asszociációval csatolt objektum kulcsai a csatoló objektum névterében léteznek (OSI terminológiával: ezek a kulcsok egy RDN részei)



Egy System sok LogicalDevice-t tartalmazhat a SystemDevice asszociáción keresztül. A LogicalDevice kulcsai a System kulcsain keresztül értelmeződnek, ezért az asszociáció "gyenge".

CIM "top" objektumok



- ManagedElement az objektumfa gyökere
- Egy ManagedElement függőségeken keresztül kötődik más ManagedElement-ekhez és statisztikai adatok tartoznak hozzá
- A gyökérobjektum változatlanságának garantálása miatt más tulajdonságot itt nem definiáltak, ezeket a leszármazott osztályokban definiáljuk.

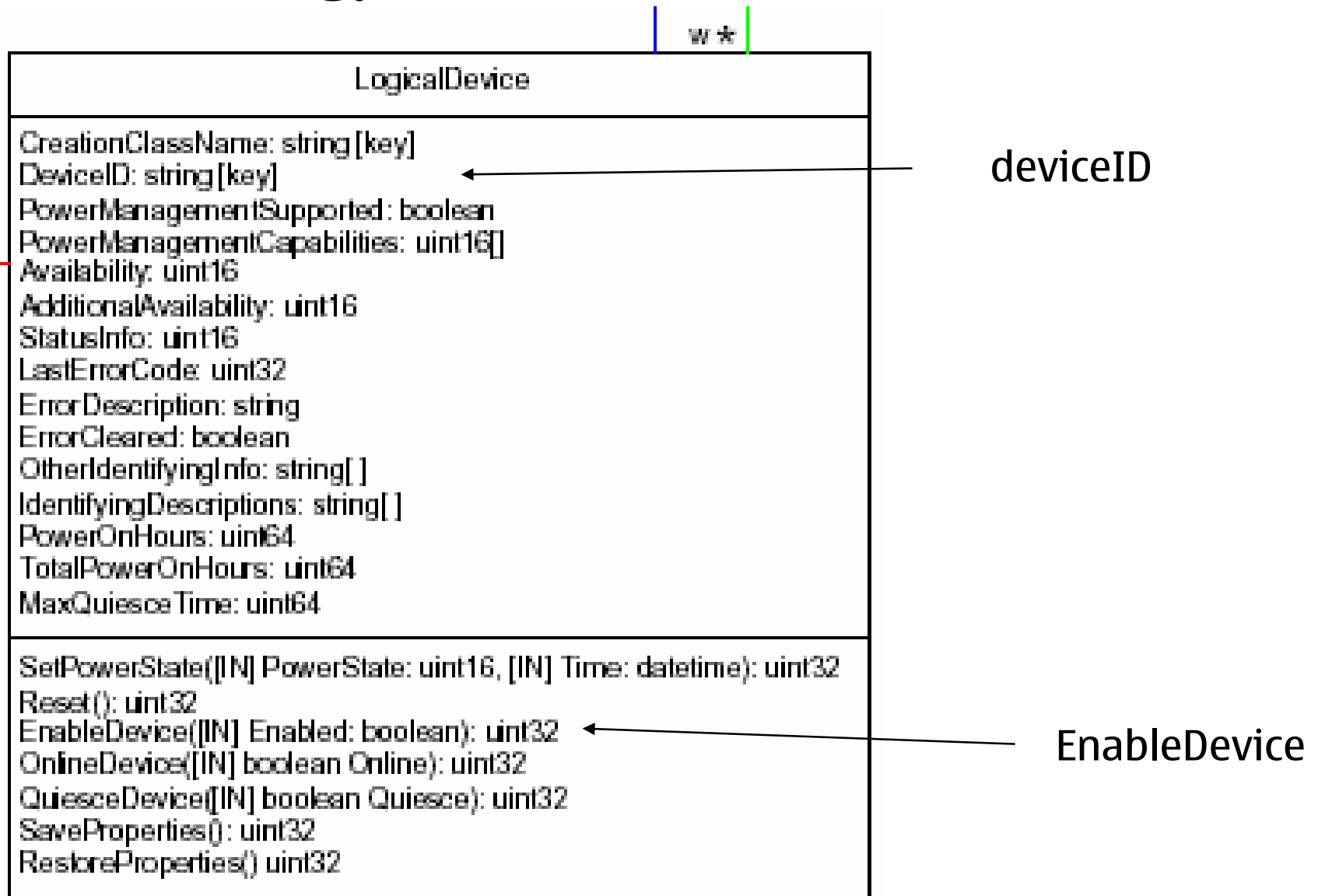
Szöveges formátum: MOF

- Managed Object Format
- [Abstract, Description ("An abstraction or emulation of a hardware entity, that may " "or may not be Realized in physical hardware. ... ")]
class CIM_LogicalDevice : CIM_LogicalElement {
 [Key, MaxLen (64)]
 string DeviceID;
 ...
 [Description ("Boolean indicating that the Device can be power managed. ...")]
 boolean PowerManagementSupported;
 [Description ("Requests that the LogicalDevice be enabled (\\"Enabled\\" "
 "input parameter = TRUE) or disabled (= FALSE). ...)"]
 uint32 EnableDevice([IN] boolean Enabled);
 ...
};

MOF (2)

- Csak részlet!
- [] közötti szöveg: metainformáció (az osztály, attribútumok vagy metódusok leírása)
- Metainformáció: ez az osztály absztrakt, az ember által érthető leírása a Description() elemben van.
- class: A CIM_LogicalDevice osztály a CIM_LogicalElement leszarmazottja
- A deviceID nevű attribútuma része a kulcsstruktúrának (ld. később)
- Van egy egy PowerManagementSupported nevű boolean attribútuma
- Van egy EnableDevice nevű metódusa, ami egy boolean-t vár paraméterül és egy 32 bites egészt ad vissza. [OUT] paraméter metainformáció lehetséges, a metódusnak lehet több visszatérési értéke.

Ugyanez UML-ben



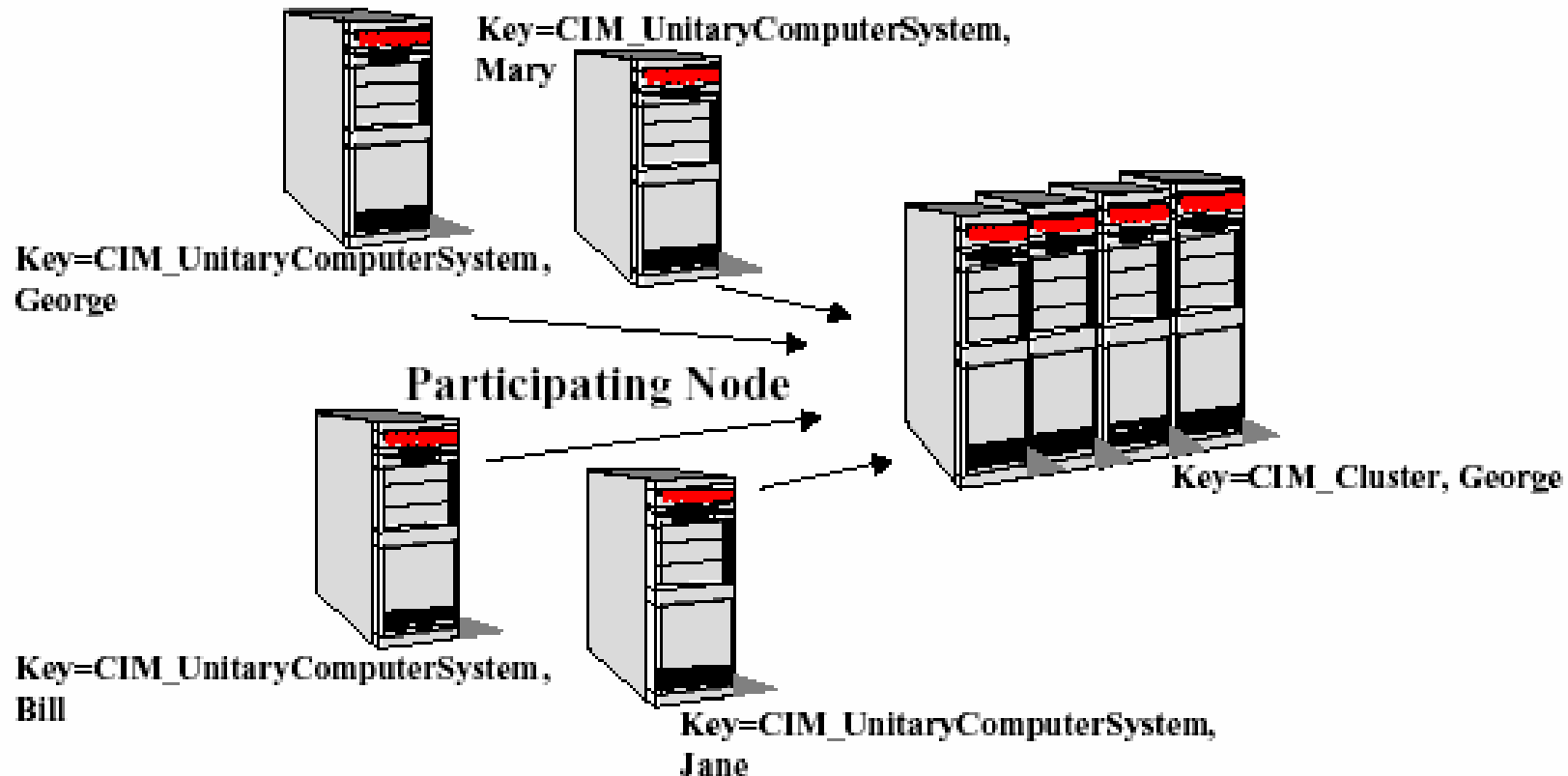
Típusok megszorítása MOF-ban

- [... ValueMap {"1", "2", "3", "4", "5"},
Values {"Other", "Unknown", "Enabled",
"Disabled", "Not Applicable"}, ..]
uint16 StatusInfo;
- A ValueMap a megengedhető értékeket írja le (akkor is string, ha az alaptípus pl. egész, mert a CIM-ben minden stringre konvertálódik)
- A Values az érték szöveges reprezentációját adja (pl. menedzsmint konzolon való kijelzésre)
- ValueMap elmaradhat, ekkor 0-tól egyesével számozza a felsorolást
- BitValues és BitMaps: használata hasonló a Value-ValueMap-hoz, de a BitMaps bitpozíciókat ír le és az alaptípust (pl. egészt) bitmap értékek tárolására használjuk.

Elnevezés CIM-ben

- Minden osztálynak van u.n. kulcsstruktúrája
- Ez attribútumok halmaza, amelyek kulcsként vannak megjelölve
- Az osztály konkrét egyede ezeken az attribútumokon keresztül található meg
- Ha az osztály gyenge asszociációval van csatolva egy másik osztályhoz, a másik osztály kulcsstruktúrája képezi az elnevezés felsőbb szintjét, aminek kontextusában az osztály kulcsstruktúrája értelmeződik.

Elnevezés CIM-ben: példa



- Kulcsok a CIM_Cluster-ben és a CIM_UnitaryComputerSystem-ben:
CreationClassName, Name
- "George" egyedi számítógép neve: CIM_UnitaryComputerSystem, George
- A cluster neve: CIM_Cluster, George

CIM események

- Nem a Core Model része
- Nincs teljesen készen
- Objektorientált eseménymodell a CIM részére
- Fogalmak
 - Esemény (event): valami, ami érdekelheti az operátort, pl. egy memóriamodul tönkrement
 - Jelzés (indication): konkrét üzenet, amit az ágens küld és a menedzselte rendszerben történt dolgokra vonatkozik, pl. paritáshiba az x. memóriamodulban.
 - A jelzések eseményekre utalnak. A jelzést küldi az ágens, ebből az operátor, vagy a menedzselő rendszer eseményekre következtet.
- A jelzések számára a CIM objektumhierarchiát definiál
- A CIM jelzésobjektumok sokféle módon lehetnek kódolva, a CIM a CIM-XML kódolást definiálja (ld. később)

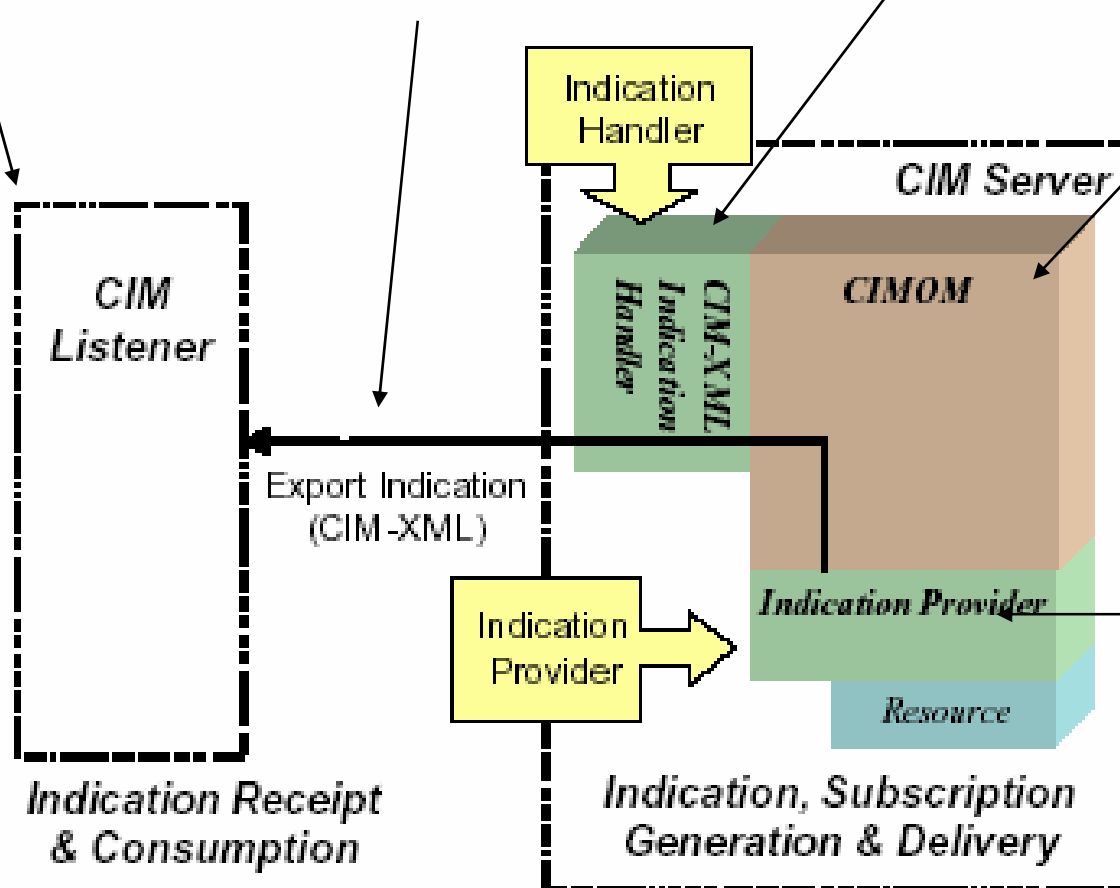
CIM eseményarchitektúra

Fogadó
(pl. menedzsment
rendszer)

A jelzés fizikai
eljuttatása a fogadóhoz

Komponens, ami ténylegesen elküldi
a jelzést a fogadónak

Ágens, ami nyilvántartja,
milyen jelzésekre kíváncsi
a fogadó



Export Indication
(CIM-XML)

Indication
Provider

Indication
Handler

CIM Server

CIM-XML
Indication
Handler

CIMOM

Indication Provider

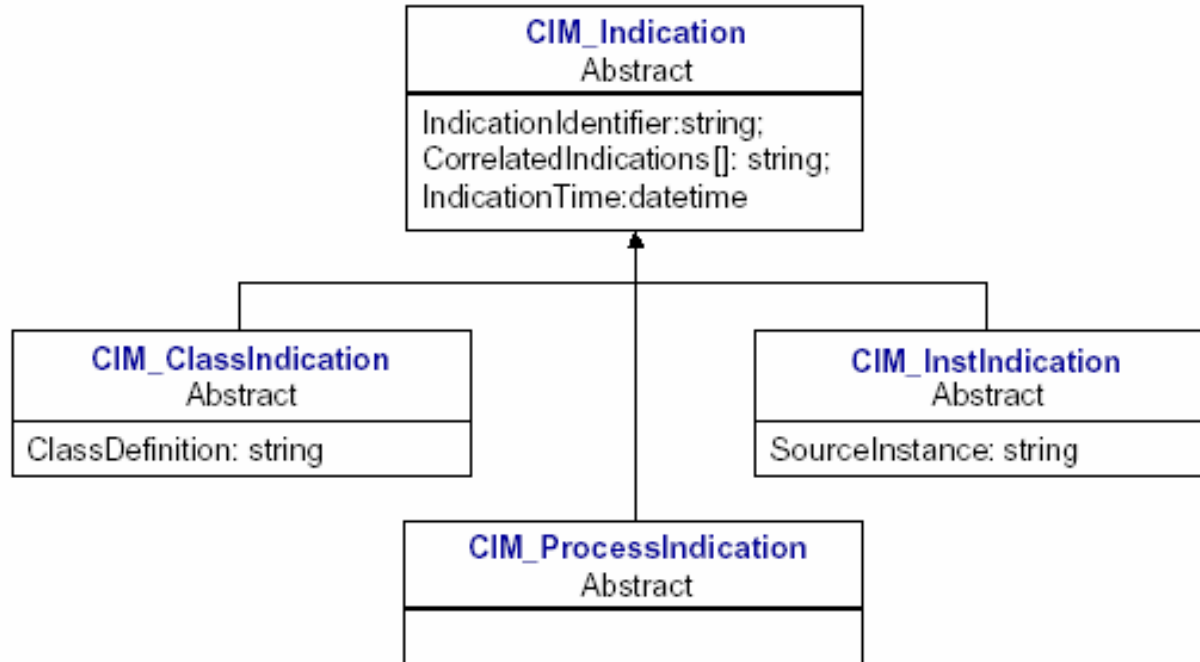
Resource

Indication Receipt
& Consumption

Indication, Subscription
Generation & Delivery

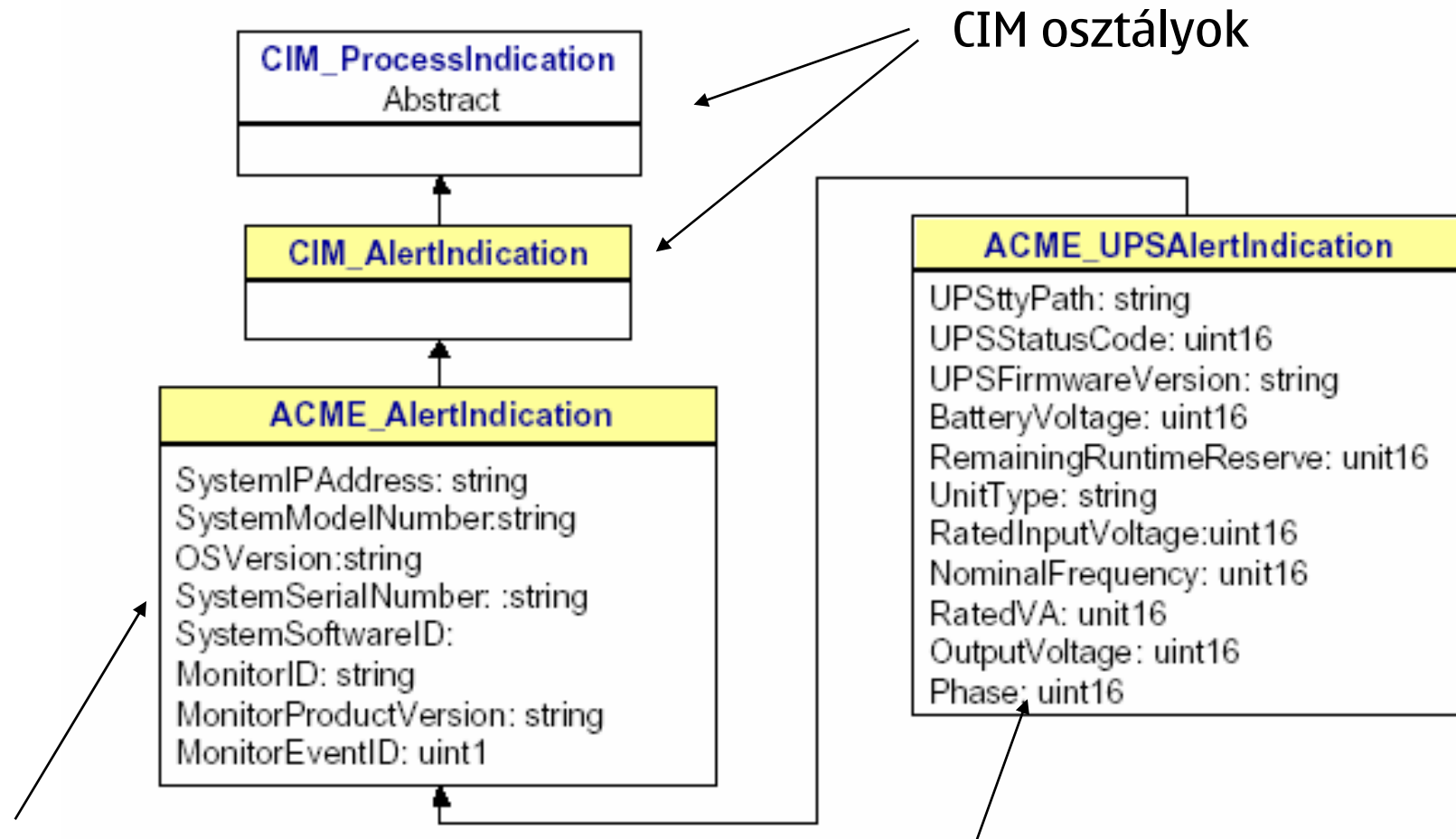
A jelzés forrása
(pl. hibafigyelő rendszer)

Jelzések



- CIM_ClassIndication: CIM osztály jött létre, módosult vagy pusztult el
- CIM_InstIndication: CIM egyed jött létre vagy pusztult el, metódusát hívták, attribútumát írták vagy olvasták
- CIM_ProcessIndication: minden, menedzselte rendszerrel kapcsolatos jelzés őse.

Példa: hogyan hozzunk létre saját jelzéseket

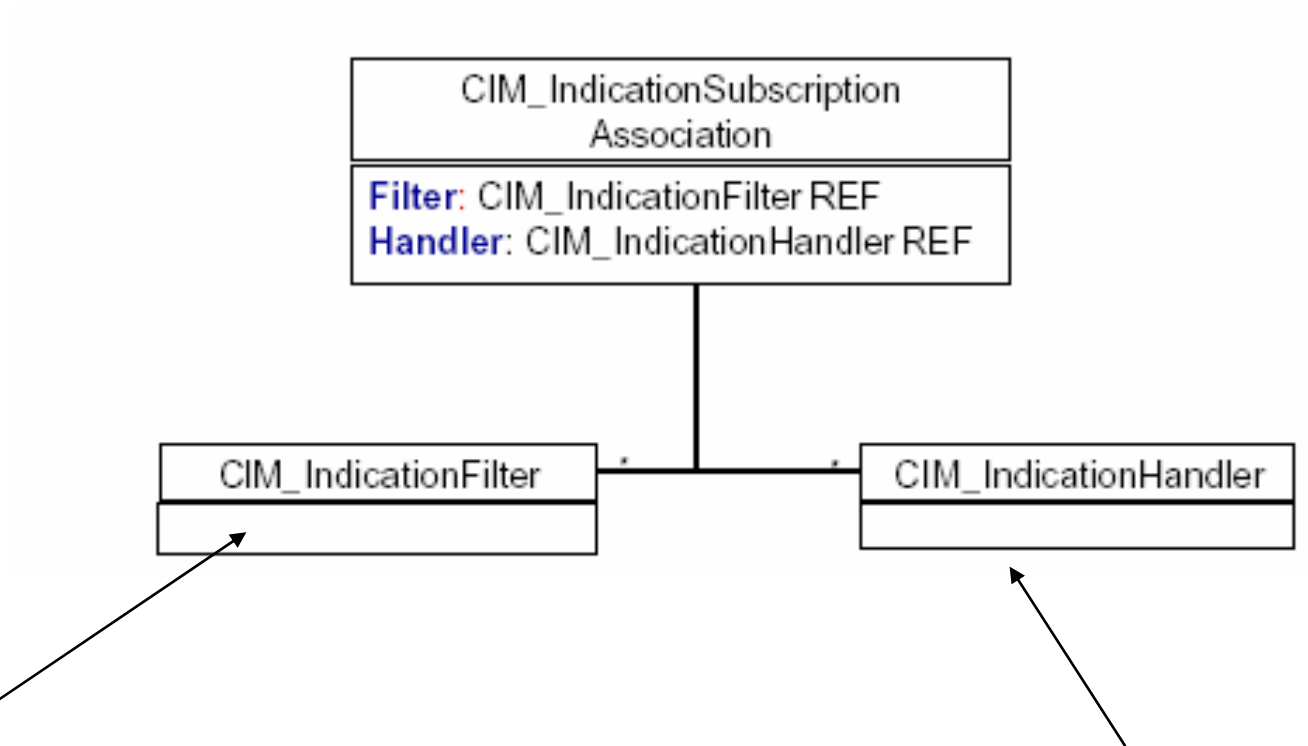


Az ACME gyártó osztályai: általános hiba és annak UPS hibára specializált változata.

Előfizetések: a jelzések szűrése

- Nagyon sok jelzés keletkezhet
- Egyes menedzsment rendszerek csak bizonyos jelzésekre lehetnek kíváncsiak
- A CIM eseménymodel az előfizetés (subscription) segítségével teremt kapcsolatot a generált jelzések és a fogadó között.
- Az előfizetések szintén menedzselhetők és számos módon keletkezhetnek (lehetnek előre létrehozottak, létrehozhatja őket az ágens, a menedzselt rendszer, stb.)

Előfizetések



Szűrő: milyen jelzéseket küldjük?

Hova és milyen formátumban küldjük? Szabványos IndicationHandler: CIM_IndicationHandlerCIMXML

Szűrő példa

Információ emberi operátoroknak

CIM_IndicationFilter		
	Property Name	Value
	Caption	Generate critical UPS alerts for /dev/tty0p1
	Description	Filter used by ACME to generate critical UPS alerts for device 'tty0p1'. This indication will return all properties associated with the class ACME_UPSAlertIndication.
Key	SystemCreationClassName	CIM_UnitaryComputerSystem
Key	SystemName	server001.acme.com
Key	CreationClassName	CIM_IndicationFilter
Key	Name	ACMESubscription12345
	SourceNamespace	root/cimv2
Required	Query	SELECT * FROM ACME_UPSAlertIndication WHERE PerceivedSeverity = Critical AND UPSttyPath = "/dev/tty0p1"
Required	QueryLanguage	WQL

Kulcsok: ez alapján menedzselhető az előfizetés

Szűrő WQL nyelven. ACME_UPSAlertIndication-okat enged át, ahol a PerceivedSeverity mező értéke Critical és a jelzés a /dev/tty0p1 portra csatolt UPS-től jött.