

UML usecase modellezés, állapotterképek

1. Mik azok a használati esetek (use case, UC)? Hogyan épül fel egy használati eset leírása?
2. Milyen kapcsolatban lehetnek az egyes aktorok, UC-ek?
3. Hogyan fejthető ki részletesebben egy UC?
4. Hogyan követhetők nyomon az egyes UC-ek (avagy mi a hatásuk) a tervezés későbbi fázisaiban?
5. Mik azok a protokoll statechartok és miben különböznek a hagyományos állapotterképektől? Milyen módon valósítható meg az elosztott komponensek közötti kommunikáció?

Domain-specifikus modellezési nyelvek

1. Mi szükséges egy modellezési nyelv definiálásához? Mit értünk absztrakt szintaxis alatt? Mi a konkrét szintaxis? Miért van rá szükség?
2. Mit értünk jólformáltsági kényszer alatt? Mik a legtipikusabb (sok nyelvben előforduló) kényszerek? Hogyan lehet egy nem tipikus (nyelv specifikus) kényszert kifejezni (példával)?
3. Értelmezhető-e példányosítási ill. finomítási viszony él típusok (asszociációk, relációk) között? Mi következik ebből a végpontokra nézve?
4. Mik az absztrakt szintaxis és konkrét jelölésrendszer közötti viszony multiplicitása? Miért?
5. Mik az absztrakt modell és egy rögzített jelölésrendszer szerinti konkrét megjelenése közötti viszony multiplicitása? Miért?
6. Mutassa be (szabadon választva) a MOF vagy az EMF modellezés metaszintjeit! Illusztrálja egy egyszerű Jedi-adatbázis (név, mester-padawan viszony) példán keresztül!
7. Milyen kétféle módon adható meg egy viselkedésmodellezési nyelv dinamikus szemantikája? Melyik hogy működik?

Eclipse Modeling Framework

1. Ismertesse a legalapvetőbb Ecore struktúrát, azaz az Ecore metamodelljét! Írjon egy egyszerű metamodell egy személygépjárműhöz.
2. Ismertesse az EMF fejlesztési workflowt, kitérve, hogy egyes pontokban milyen elemek jönnek létre (file-ok, modellek, forráskód, stb.)
3. Ismertesse a genmodelt, mire használják és milyen jellegű attribútumokat lehet vele beállítani?
4. Ismertesse a generált EMF.model, EMF.edit és EMF.editor projekteket. Térjen ki a generált Java kódban alkalmazott modelltárolási megoldásokra. Egy egyszerű példán keresztül ismertesse a generált Java osztályok és Interfészek kapcsolatát.

Kódgenerálás

1. Ismertesse a 3 legelterjedtebb kódgenerálási megközelítést. Mik az erősségeik és gyengeségeik?
2. Mi az AST? Miért fontos a kódgenerálás körében és milyen előnyöket/hátrányokat hoz be a használata.
3. Mik a generátor modell használatának előnyei és hátrányai?
4. Ismertessen legalább 3, a kódgenerálás körében ismert problémát és adjon rájuk megoldási ötleteket.

5. Milyen szabályok alapján származtathatók egy domain-specifikus modellből egy alkalmazás implementációjának interfészei és osztályai? Milyen funkcionalitás támogatásáig lehet eljutni kódgenerálásban (pl. EMF és Xtend segítségével)?
6. Definiálja a refactoring és a reverse engineering fogalmát! A HF fejlesztése során milyen refactoring lépésekkel találkozhatott?

Modelltranszformációk, gráftranszformációk

1. Ismertesse azon területeket, ahol a modelltranszformáció jól hasznosítható! Adjon példát egy konkrét transzformációs nyelvre!
2. Rajzolja fel és ismertesse egy általános modelltranszformációs keretrendszer felépítését!
3. Ismertesse a gráftranszformáció alapjait! Írjon fel egy egyszerű szabályt és mutassa be, hogyan alkalmazható egy modellen!
4. Ismertesse a gráftranszformációs formalizmus kiterjesztésének lehetőségeit! Ezek közül adjon példát legalább egyre.

Lekérdezések, validáció

1. Ismertesse az EMF-IncQuery rendszer lekérdezőnyelvét!
2. Hogyan használható az EMF-IncQuery modellalapú fejlesztőeszközökben? Milyen funkciókat támogathat? Mondjon pár példát hasonló technológiákra! Miben hasonlítanak és különböznek?
3. Mi az OCL célja? Jellegetesen hol (milyen UML diagramokban) használunk OCL kényszereket? Hogyan kapcsolható az OCL EMF modellekhez?
4. OCL kifejezésekben hogyan kell megfogalmazni a navigációt és a halmazelméleti műveletek végrehajtását?
5. Hogyan használható az EMF-IncQuery, illetve Eclipse OCL technológia domain-specifikus modellek validációjára? Melyek az egyes technológiákban nehezen megfogalmazható és drágán kiértékelhető szabályok?

Konkrét szintaxis

1. Hasonlítsa össze a szöveges és grafikus szerkesztőkben a szerkesztés lépéseit!
2. Ismertesse a modell-alapú szöveges és grafikus nyelvek használatának előnyeit és hátrányait!
3. Soroljon fel és hasonlítsa össze olyan technológiákat, amelyekkel grafikus szintaxis rendelhető egy DSM-hez!
4. Milyen kihívások vannak grafikus szerkesztők automatikus elrendezéssel (layouting) történő kiegészítésében?
5. Mutassa be egy szöveges nyelv feldolgozásának folyamatát!
6. Soroljon fel és hasonlítsa össze szöveges nyelvek fejlesztését támogató eszközöket!

SysML és SysMod

1. Mi az SysML célja? Ismertesse röviden a nyelvcsaládot és az UML-hez képest új nyelveket!
2. Ismertesse a V fejlesztési modell különböző megközelítéseit? Miért fontos a nyomkövethetőség ?
3. Röviden mutassa be a SysMod módszertanát és legfontosabb lépéseit.

Architektúra modellező nyelvek, AADL, AUTOSAR

1. Miért van szükség architektúra leíró nyelvekre? Soroljon fel néhány példát! Milyen módszerrel tervezhetjük meg a rendszer architektúráját az egyes komponensek megvalósítása nélkül?
2. Mi az AADL? Milyen szakterületeken használható? Ismertesse az AADL alapvető tulajdonságait! Milyen módon csoportosíthatók az AADL nyelvi elemei? Soroljon fel néhány nyelvi elemet? Hogyan kapcsolódik az AADL az UML-hez?
3. Mi az AUTOSAR? Milyen szakterületeken használható? Ismertesse az AUTOSAR alapvető tulajdonságait! Ismertesse röviden az AUTOSAR által javasolt tervezési folyamatot! Mit jelent egy komponens az AUTOSAR-ban és hogyan kommunikálnak a komponensek?
4. Mi az EAST-ADL? Miben különbözik más architektúra leíró nyelvektől (AADL, AUTOSAR, UML, SysML)? Milyen szinteket definiál az EAST-ADL?

Modell menedzsment és fejlesztési folyamatok

1. Ismertesse a RUP fejlesztési módszertant és azon belül térjen ki, hogy milyen módon illeszkedik hozzá a modellvezérelt fejlesztési paradigma
2. Ismertesse a modell menedzsment szempontjából fontos területeket külön kiemelve azokat, amelyek nem jelennek meg hagyományos szoftver fejlesztési folyamatok esetén
3. Ismertesse a Test és a Domain-driven fejlesztési módszertanokat.