

Logikai modellek készítése

Önálló laboratórium feladat összefoglalója

Izsó Benedek (G6TU3R)

Konzulens: Szatmári Zoltán

BME Méréstechnika és Információs Rendszerek Tanszék

Informatikai technológiák szakirány / Rendszertervezés ágazat, 2008/2009. II. félév

Kritikus beágyazott rendszereknél nem javasolt az eddig tanult módszerekkel (RAD) szoftvereket fejleszteni. Már tervezési fázisban érdemes felismerni és kiszűrni a hibákat. Ezt a feladat modellezésével érhetjük el. Az egyik leghatékonyabb módszer, ha modellünket ontológiával írjuk le, mert ezt egy logikai következtető fel tudja dolgozni, és segítségével számos ellenőrzést tudunk végrehajtani. Egy fejlesztő viszont tipikusan UML modellekkel írja le terveit, így ezt célszerű ontológia modellé konvertálni. Ehhez nyújt segítséget a VIATRA modelltranszformációs keretrendszer, mellyel meg lehet oldani, hogy például egy UML metamodellnek megfelelő modellt ontológia metamodellnek megfelelő modellé transzformáljunk, és végül az ontológia modellt egy logikai következtető számára feldolgozható fájlba exportáljuk.

A félév elején a Petri hálós példán keresztül megismerkedtem a modellezéssel, modelltranszformációval, ennek VIATRA rendszerben lévő reprezentációjával, illetve írtam PNML-be exportáló és onnan importáló VIATRA modult.

Ezután megismerkedtem a leíró logikákkal az *AL*, *ALC*, *SHIQ* nyelveken keresztül, majd az ontológiákkal. Egy ontológia a teljes tudásbázist írja le (ABox+TBox), ennek VIATRA-ban reprezentált metamodelljének megalkotása volt az önálló labor célja.

Felmerült a kérdés, hogy melyik nyelvnek a metamodelljét érdemes elkészíteni. Ennek kapcsán megismertem az RDF, RDFS elképzelést. Az OWL (Web Ontology Language) nyelv a legnagyobb kifejező erejű (a leíró logikákhoz képest is), így végül ennek a metamodelljét hoztam létre.

Sajnos az OWL nem eldönthető nyelv, így bizonyos esetekben érdemes ennek részhasználatát, az OWL DL –leíró logikai *SHOIN(D)*– vagy a teauruszok létrehozására is használható OWL Lite alnyelvet használni. Ezek bár kisebb kifejező erejűek, eldönthetőek és hatékonyabb következtető algoritmus írható rájuk.

Az OWL2 még nem szabvány, és még nem támogatja minden eszköz, így ennek felépítésével nem foglalkoztam.

Az OWL metamodell létrehozása történhetett volna saját erőből composite desing pattern szerint, viszont ez a nyelv részletes és pontos ismeretét igényelte volna, a klasszikus mintánál jóval bonyolultabb származási hierarchiával. A tervezés során felmerült a nyílt forráskódú EODM, illetve e projekt zárt forráskódú folytatásának (IODT) felhasználása. Ezek beolvasó modulja a szabványban leírt összes tesztesetet teljesíti, viszont az OWL szintaktikájú fájl Ecore metamodell példányaként képezi le.

Végül a Karlsruhe Egyetemen elkészített Ecore formátumban rendelkezésre álló MOF metamodellt sikerült alkalmaznom, amit a már létező VIATRA Ecore importterrel sikerült betölteni. Létrehoztam egy példa modellt. A meglévő metamodellt ki kellett egészíteni egy entitással, és egy abból kiinduló relációval. Az entitás az ontológia modell gyöker entitásának típusa kell legyen, a reláció pedig onnan mutat az egyes ontológia elemekre, így lehetővé téve az ontológia bejárását irányított gráfon keresztül.

A munka folytatásaként ehhez ontológia formátumba exportáló modult, és különböző típusú modellekről ontológia modellre transzformáló modulokat kell még írni.