

M Ű E G Y E T E M 1 7 8 2
BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
MŰSZAKI TUDOMÁNYÁG – INFORMATIKAI TUDOMÁNYSZAK

**ESEMÉNYVEZÉRELT MODELLTRANSZFORMÁCIÓK
SZAKTERÜLET-SPECIFIKUS NYELVEKBEN**

PHD ÉRTEKEZÉS TÉZISEI

RÁTH ISTVÁN
OKL. MÉRNÖK–INFORMATIKUS

TÉMAVEZETŐ:
DR. VARRÓ DÁNIEL
EGYETEMI DOCENS

BUDAPEST, 2011. MÁRCIUS

Köszönetnyilvánítás. Elsőként szeretnék köszönetet mondani konzulensemnek, Dr. Varró Dánielnek, aki a kezdetektől töretlen lelkesedéssel segítette munkámat számtalan ötletével, tudományos és szakmai tapasztalatával és értékes tanácsaival, amelyek nélkül nem járhattam volna sikerrel.

Ezúton köszönöm minden kollégámnak is a segítséget: a Budapesti Műszaki Egyetem Mérés-technika és Információs Rendszerek Tanszékén a Hibatűrő Rendszerek Kutatócsoport vezetőjének, Dr. Pataricza András professzornak és Dr. Majzik Istvánnak munkám folyamatos támogatásáért vagyok hálás. Nagyra értékelem éveken átívelő együttműködésünket a VIATRA2 csoporttal: köszönöm Dr. Balogh András, Bergmann Gábor, Hegedüs Ábel, Horváth Ákos, Ujhelyi Zoltán, és Dr. Varró Gergely hozzájárulását a disszertációban bemutatott eredményekhez. Köszönettel tartozom továbbá Schmidt András és Vágó Dávid barátaimnak kezdeti segítségükért, Kovács Máté, Dr. Polgár Balázs és Tóth Dániel kollégáimnak, és korábbi MSc hallgatóimnak: Ökrös Andrásnak és Horváth Ádámnak az ötleteikért, javaslataikért, és a VIATRA2 projektbe fektetett sok munkáért.

Szeretném kifejezni hálámat Prof. Dr. Reiko Heckelnek és Dr. Paolo Torrini tudományos munkatársnak (University of Leicester) az eredményes kutatási együttműködésünkért. Köszönöm a SENSORIA projektbeli eredményes együttes munkát Philip Mayernek (Ludwig-Maximilians-Universität München). Köszönettel tartozom Dr. Krzysztof Czarneckinek és csoportjának (University of Waterloo), hogy támogatták kutatómunkámat és kiküldetésemet.

Munkámat az alábbi európai uniós kutatási projektek támogatásával végeztem: SENSORIA (IST-3-016004, FP6), DIANA (AERO1-030985, FP6), SecureChange (ICT-FET-231101, FP7) és MO-GENTES (ICT-216679, FP7). A disszertáció elkészítése során a BME Villamosmérnöki és Informatikai Kar Doktorjelölti Ösztöndíja támogatott.

Végül, nagyon szépen köszönöm családomnak és barátaimnak a sok áldozatot és türelmet.

1. Bevezető

1.1. Modellvezérelt szoftvertervezés

A szoftvertervezés mindennapi gyakorlata során számos régóta ismert problémával kerülünk szembe új és új formában, a különböző szakterületeken működő rendszerek növekvő bonyolultsága miatt. A szoftverfejlesztés során az egyik legnagyobb kihívás, hogy a – félreérthető, pontatlanul megfogalmazott – követelmények teljesülésének vizsgálatával folyamatos kockázatcsökkentésre kell törekedni, mivel a későn felfedezett hibák javítási költsége exponenciálisan nő az idővel. A félreértésekből származó tervezési hibák elkerülésének kulcsa a tervezési technikák absztrakciós szintjének a célterület fogalmaihoz történő közelítése, mivel ez lehetővé teszi a szakterületi szakértők közvetlen bevonását a fejlesztési folyamatba. Ennek érdekében növelni kell az újrahasznosíthatóságot és a követelmények nyomomonkövethetőségét a tervezői eszközökben és teljes szoftver-életcikluson keresztül.

Ezen kihívásokra kínál választ a *modellvezérelt szoftverfejlesztés* paradigmája, amelynek alapötlete, hogy a fejlesztő a tervezés és implementáció során többnyire magas absztrakciós szintű mérnöki modellekkel dolgozik. A hagyományos programozás feladatát automatikus kódgenerátorok veszik át, a kézi munkára jellemző magas hibaráta csökkentése céljából. A forráskódgenerálás során kiemelt cél, hogy lehetőleg minimális mértékű legyen a kézi kódolás aránya, így a modellezés hatáskörén kívül eső logika könnyebben kézben tartható. A generatív technikákon túlmenően fontos kiemelni, hogy lehetőség nyílik a magasszintű rendszermodellek formális módszereken (pl. modellellenőrzés, statikus analízis, tesztgenerálás) alapuló verifikációjára és validációjára is.

Az Object Management Group (OMG) szabványosító testület, az UML dokumentációs alkalmazásain túlmutató ajánlásként kiadta a Modellvezérelt Architektúra (Model Driven Architecture, MDA [MDA01]) szabványt. Az MDA lényegében egy – az OMG olyan kulcstechnológiáit, mint az UML modellező nyelv, a Meta Object Facility (MOF) metamodellező keretrendszer, az XML Metadata Interchange (XMI) adatszere-szabvány, és a Common Warehouse Metamodel (CWM) metaadat-leíró nyelv ötvöző – fejlesztési életciklus-ajánlásokat és tervezési mintákat összegző ajánlás.

Az MDA értelmében világosan el kell választani egymástól az ún. platformfüggetlen modelleket (Platform Independent Models, PIM) a platformspecifikus modellektől (Platform Specific Models, PSM), így a szoftvertervezést egy három lépcsőből álló folyamatként képzelhetjük el. A tervezés során elsőként a platformfüggetlen modell készül el, a probléma absztrakt, technológiai korlátoktól elvonatkoztatott elemzése alapján. Ezt követi a platformspecifikus modell (fél)automatikus származtatása, mely lépés során a modellt technológiaspecifikus kiegészítő információkat hozzáadásával finomítjuk. Ezek a reprezentációk a futtatandó alkalmazás egy-egy érvényes konfigurációját kódolják, mely lehetővé teszi – utolsó lépésként – a tényleges forráskód automatikus generálását. A különböző modellezési szintek és ábrázolásmódok közötti automatikus átalakításokat *modelltranszformációk* segítik.

A modelltranszformáció szerepe az MDA-ban. A modelltranszformációk kulcsfontosságú szerepet játszanak az MDA folyamataiban. Segítségükkel modellezési nyelvek közötti és nyelveken belül értelmezett leképezéseket automatizálhatunk. A *horizontális transzformációk* során azonos absztrakciós szinten elhelyezkedő modellek közötti átalakítás történik (pl. tervezési aspektusok szinkronizációja PIM-ekben), míg *vertikális transzformációk* esetén absztrakciós szinteket átlépve a rendszermodellek különböző reprezentációit származtatjuk egymásból (pl. külső paraméterezéssel a PIM-PSM leképezésnél). A transzformációs leképezéseket széleskörűen használhatjuk, pl. kódgenerálásra vagy a rendszermodellek matematikai leképezésére a formális verifikációhoz,

de bevethetők olyan hagyományosabb feladatok támogatására is, mint például a modellek verziókezelése, a fejlesztőeszközök különböző változatai közötti adatmigráció, modellalapú tesztgenerálás, jólformáltsági kényszerek ellenőrzése vagy akár a kódvisszafejtés.

Szakterület-specifikus modellezés. Bár az ipari alkalmazások során nem mindig követik az MDA-hoz hasonló specifikus módszertanok ajánlásait, a gyakorlatban az általános értelemben vett modellközpontú megközelítés már bizonyította, hogy komoly előnyökhöz juttathatja az alaptechnikáit magasszinten alkalmazó mérnököket [FC04]. Manapság az egyik legnépszerűbb iránymű a *szakterület-specifikus modellezési nyelvek* (domain-specific modeling language, DSML) köré épül, melyeket a szakterület-specifikus programozási nyelvek (mint pl. az SQL) modellezési analógiájaként abból a célból fejlesztettek ki, hogy az általános szoftvertechnológiai elemek helyett (vagy azokat kiegészítve), a tervezés során közvetlenül egy-egy problématerületre jellemző fogalmakkal és kapcsolatokkal dolgozhassunk. A szakterület-specifikus nyelvek őseinek az 1960-as és 70-es évek speciális programozási nyelveit, és az azokat általánosító aspektusorientált programozási technikákat tekinthetjük. Ezek célközönsége azonban még a programozók voltak, ezzel szemben a modellezési nyelvek célja már – a programozók a valódi problémára történő koncentrációjának segítése mellett – a szakterületi szakértők közvetlen bevonása a szoftvertervezésbe.

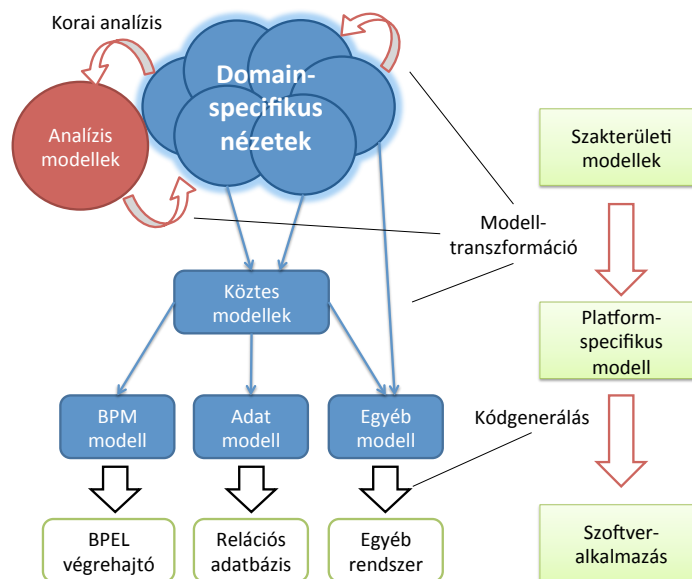
A DSML-ek UML-lel szembeni fő előnye tehát, hogy lehetővé teszik a szakterület-specifikus tudás precízebb leképezését a szoftvertechnológia fogalmaira. Az ilyen modellek szemantikája is pontosabban specifikált, így könnyebben használhatók automatizált környezetben, pl. kódgenerálásra. A szakterület-specifikus modellezésre épülő fejlesztési modell legtöbbször ún. „top-down” megközelítésként először a problémát leíró fogalmi tér modellezési nyelvbe történő szintézisét írja elő (ezt mérnöki nyelvtervezésnek – language engineering – nevezzük). Az így előállított tervezési környezet a szoftveres kötöttségektől nagymértékben független, a valódi problémára fókuszált tervezői munkát tesz lehetővé. Fontos előnye, hogy az ilyen absztrakt, a terület szempontjából kiemelten fontos aspektusokat és információkat tömörítő modellek segítségével jóval hatékonyabb korai analízis végezhető.

1.1.1. Szakterület-specifikus modellezés a modellvezérelt fejlesztésben

Az informatikai rendszertervezés mai gyakorlatában az MDA típusú ajánlásokat gyakran ötvözik egyéb modellalapú módszerekkel is, így – az eredeti OMG szabványt kiterjesztve – sokszor nem tisztán UML modellekkel, hanem szakterületi modellekkel is találkozhatunk MDA elveken megvalósított fejlesztőeszközökben (1. ábra).

Ebben a megközelítésben a mérnöki tervezőmunka lényegi része olyan egyedi grafikus (és szöveges) modellező eszközök segítségével történik, melyeket kifejezetten az adott szakterület igényei alapján alakítottak ki. Ezen eszközök lehetőséget biztosítanak arra, hogy a komplex rendszert több fontos tervezési aspektus együttes felhasználásával tervezhessük meg. A tervezési aspektusok használhatnak szabványos modellezési nyelveket (pl. UML-t vagy UML profilokat), és a fejlesztő cég vagy akár a konkrét feladat igényei szerint testreszabott szakterület-specifikus modellezési nyelveket is.

Az ilyen módon felépített komplex PIM a fejlesztőeszközök láncolatán (development toolchain) végighaladva, átmeneti reprezentációk során keresztül fokozatosan egy szintén több aspektust hordozó PSM-mé alakul át. A nyomomonkövethetőségi modellekkel (traceability models) összekapcsolt köztes modellek különböző célokat szolgálhatnak (mint például a korábban már említett formális analízis, telepítési konfiguráció generálás, alacsony szintű forráskódmodellek). A folyamat végén előáll a teljes alkalmazás fordítható és futtatható forráskódja.



1. ábra. A modelltranszformációk és szakterületi modellek együttműködése

1.1.2. Nyelvtervezés a fejlesztési folyamatokban

A fenti architektúrából következik, hogy a szakterületi modellező nyelveket felhasználó eszközök alkalmazása sok esetben a támogató modellezőkörnyezet kifejlesztését is igényli. Így, bár a DSML-ek alkalmazása komoly előnyökkel járhat, számolni kell a megnövekedett fejlesztési költségekkel és az ilyen meta-programozási feladatok összetettsége sem elhanyagolható. Emiatt ennek a technológiának a gyakorlati alkalmazhatóságát elsősorban a *nyelvtervezési eszközök* (language engineering tools) hatékonysága határozza meg. Ez annak ellenére is igaz, hogy ma már számos DSML elérhető dobozos termékként és kapcsolódó nemzetközi szabványként (pl. a SysML, AADL, AUTOSAR, UML-MARTE a beágyazott rendszerek szoftvereinek fejlesztéséhez, vagy a WS-BPEL, BPMN és a SoaML a szolgáltatásorientált nagyvállalati szoftverrendszerek fejlesztéséhez használatos). Ennek oka, hogy a legjobb szabványosító szándék ellenére sem mindig alkalmazható egy-egy ilyen nyelv közvetlenül, szinte mindig felmerülnek olyan egyedi testreszabási igények, melyeket nyelvtervező eszközök segítségével elégíthetünk ki.

Az egyik legnépszerűbb ipari nyelvtervezési keretrendszer az Eclipse Modeling Framework (EMF [EMF]). Természetesen léteznek egyéb ilyen ipari és akadémiai eszközök is (pl. [GME, LLMC04, Mic]), melyek közös tulajdonsága, hogy kiemelten támogatják a szakterületi nyelvek köré épülő modellezési eszközök gyors prototipizálását. Segítségükkel a prototípusok később teljesértékű eszközökké fejleszthetők tovább, általában olyan integrált programozási környezetbe ágyazva, mint amilyen az EMF, és a grafikus megjelenítést lehetővé tevő Graphical Modeling Framework (GMF, [GMF]) esetén maga az Eclipse is. A nyelvek fejlesztését általában a következő aspektusok támogatják:

- *Konkrét szintakszis*, mely meghatározza egy (modellezési) nyelv felhasználói „felületét”, azaz azt a grafikus, vagy szöveges ábrázolásmódot, mellyel a felhasználó dolgozik.
- *Absztrakt szintakszis*, mely a nyelv fogalmait, belső szimbólumait, és azok gépi ábrázolását (a metamodellt) specifikálja.
- *Az ábrázolásmódok közötti leképezések*, melyek segítségével az absztrakt és konkrét szintakszis közötti leképezést támogatják.

takszis modelljeit szinkronizálhatjuk egymáshoz. A leképezés lehetőségei alapvetően meghatározzák a modellezési nyelv absztrakciós képességeit, azaz hogy a modellező környezet milyen mértékben képes a sokszor bonyolult gépi ábrázolás sajátosságait elrejtetni a felhasználó elől, a lényeges információ kiemelése érdekében.

- *Jólformáltsági szabályok*, melyek az absztrakt szintakszis kiegészítésével olyan kényszereket határoznak meg, melyek elengedhetetlenek a precíz használat, és tervezési hibák elkerülése érdekében.
- *Végrehajtási szemantika*, mely a DSML dinamikus viselkedését, a modellezett rendszer és környezetének interakcióit határozza meg. A fejlett tervezőeszközökben lehetőség van arra, hogy a végrehajtási szemantika segítségével tervezési időben szimuláljuk a rendszer viselkedését, jóval az implementáció (kódgenerálás) előtt.

1.2. Célkitűzések

Annak ellenére, hogy rendkívül nagy mennyiségű fejlesztési munkát fektettek az ipari nyelvtervezési eszközök fejlesztésébe (példaként említhető, hogy az EMF környezet 2003 óta létezik, és több főverzió-váltáson esett át a kezdetek óta), a szakterületi nyelvek fejlesztése továbbra is drága, időigényes és sokszor igen összetett feladat, amely különleges felkészültséget igényel [DAR06].

Érdekes – és egyben egybecseng az MDA esetében megfigyeltekkel – hogy az egyik kulcsprobléma az *ipari* DSM eszközöknél is a modelltranszformációk nehézkes (lényegében nagyon kezdetleges) integrációs lehetősége. Ennek sajnálatos következménye, hogy az ilyen keretrendszerekkel készített eszközök általában *statikusak és zártak*, az alábbi értelemben:

- Bár a legtöbb eszközben van lehetőség kódgenerátorok írására, a modellintegráció (azaz a modellezési nyelvek összefüggő nyelvcsaládokba történő rendezése) nehézkes – sok esetben ad-hoc technikákat használnak (pl. XML szintű adattranszformációk), melyek újrahajthatósága korlátolt.
- Szinte minden eszközre jellemző az absztrakt és konkrét szintakszisok kötöttsége, azaz a korlátozott absztrakciós képesség a vizuális megjelenítés, és a belső ábrázolás között, mely negatívan hat a nyelvek használhatóságára.
- A nyelvtervező mérnököknek általában igen bonyolult specifikációs nyelveket (pl. OCL) kell megtanulniuk még egyszerű jólformáltsági szabályok leírására is. Ez sokszor oda vezet, hogy a nyelveket csak ritkán és felületesen egészítik ki összetett jólformáltsági szabályokkal, ez pedig gyengíti a nyelv precizitását és robosztusságát.
- A legtöbb DSML a rendszertervezés statikus aspektusaira korlátozódik, mivel a legtöbb eszközben magasszintű támogatás híján nagyon nehézkes testreszabott dinamikus szemantika definiálása és végrehajtása. Léteznek ugyan kiváló dedikált modellszimulációs eszközök (mint például a MATLAB), ezek általában igen drágák és összetettek, így nem igazán alkalmasak egyszerűbb nyelvek gyors prototipizálásához.

A fentiekén túlmenően elmondható, hogy a mai DSML eszközökben az automatizált modellfeldolgozást támogató formalizmusok és technológiák igen változatosak és kevésbé integrálhatók. Ezért az egyedi kiegészítések és testreszabott szolgáltatások mindig az egyes eszközök programozási felületeihez csatlakozó, alacsony szintű megoldások, így igen drága a kifejlesztésük. Ma nincs olyan általános megközelítés, mely a fent említett problémák mindegyikét egy

egységes fogalmi és technológiai keretben lefedné, és ez a DSM technológia további széleskörű elterjedését gátló, megoldandó probléma.

Disszertációmban amellet érvelek, hogy egy fejlett modelltranszformációs technológia betöltheti ezt az űrt, és a szakterületi nyelvtervezés specifikus kihívásaihoz igazítható megfelelő kiterjesztések segítségével. Ezekre az alapokra építve számos fejlett nyelvtervezési aspektus támogatható hatékonyan, így lényegesen könnyítve a nyelvtervező mérnökök munkáját, és javítva a szakterületi modellezőeszközök képességeit, szolgáltatásait.

- **1. kihívás: Nincs általános megközelítés az egységes, több nézőpontú szakterületi modellezéshez.** A nyelvtervező eszközök általában önálló modellezési nyelvek, vagy lazán kapcsolódó nyelvcsaládok tervezését támogatják. Az azonos vagy hasonló fogalmak közötti leképezések (melyek kulcsszerepet játszanak a koheners, többnézőpontú rendszermodellezésben) megfogalmazása és automatizált végrehajtása korlátokba ütközik. Mivel a legtöbb modellezési keretrendszer statikus típusrendszerekkel dolgozik, ezért azon modellelemeket, amelyek több tervezési aspektus szempontjából is hordoznak információt, replikálni kell, és ez konzisztencia- és redundanciaproblémákat vet fel.
- **2. kihívás: Az absztrakt és konkrét szintakszis ábrázolásmódok különválasztása korlátozott a nyelvtervezési eszközökben.** A jelenlegi grafikus modellező nyelvek tervezését támogató keretrendszerek közös vonása, hogy az absztrakt és konkrét szintakszisban ábrázolt modellek között szoros strukturális kapcsolatot feltételeznek. Egyszerűbb eszközök (mint pl. a MetaEdit+ [MEP]) a kétféle reprezentációt logikailag egy-egy kapcsolattal kötik össze. Fejlettebb rendszerekben (mint pl. a GMF), különválasztott modellezési rétegeket alkalmaznak, amelyek között a kapcsolatot egy összetett szinkronizációs mechanizmus biztosítja, amelyet a nyelvtervező csak korlátozott mértékben tud testreszabni: problémát jelent (i) egy létező absztrakt metamodellhez lényegi egyszerűsítéseket alkalmazó grafikus szintakszis definíciója, valamint (ii) egy adott grafikus nyelvhez egy speciális, saját igényeknek megfelelő absztrakt szintakszis hozzárendelése. A gyakorlatban ez a korlátozás élesen behatárolja a DSML-ek használhatóságát, különösen az összetett és szerteágazó tervezési aspektusokat lefedő nyelvek esetén.
- **3. kihívás: Nincs általános és integrált támogatás a DSML-ek viselkedési szemantikájának magasszintű leírására és tervezési idejű, interaktív végrehajtására.** A hagyományos DSM eszközökben nincs integrált támogatás a viselkedési/végrehajtási szemantika definíciójára és szimulációs végrehajtására. Bár ez a probléma áthidalható dedikált szimulációs eszközök integrált alkalmazásával, az ilyen megoldások alkalmazása esetén a tervező mindig kompromisszumokra kényszerül, mivel a szimulációs környezeteket nem speciálisan a testreszabható DSML-ekre jellemző igények és felhasználási módok figyelembe vételével alakították ki. Ennek folyományaként a dinamikus DSML-ek gyors prototipizálása csak költségesen leküzdhető akadályokba ütközik.
- **4. kihívás: Nincs általános technika a modellváltozásokra adandó reakciók leírására inkrementális modelltranszformációk esetén.** Mivel a szakterületi modellezőeszközökkel végzett munkára a gyakori modellváltozások jellemzőek, a különféle tervezési szolgáltatások modelltranszformációs támogatásának akadálya, hogy a jelenlegi transzformációs eszközök csak a kötegelte végrehajtási modell támogatják, és nincs mód a modellváltozások egységes és magasszintű feldolgozására. Bár az inkrementális modelltranszformációk (mint pl. a modellszinkronizáció) megvalósíthatók a mai eszközökkel is, ezek csak az elemi modellváltozások feldolgozását támogatják (ill. nagymértékben támaszkodnak a kiegészítő nyomonkövethetőségi modellekre a változások észrevétele céljából). Továbbá, a változások

detektálására ad-hoc kiegészítéseket és kódmódosításokat alkalmazó módszerek is léteznek, összességében tehát elmondható, hogy nincs lehetőség olyan transzformációk végrehajtására, melyek a modelleket és azok változásait is egyenrangú módon kezelik.

- **5. kihívás: Nincs beépített támogatás inkrementális modellszinkronizációhoz és inkrementális kódgeneráláshoz a nyelvtervező eszközökben.** A modern modelltranszformációs eszközökkel külső technológiaként kiegészített nyelvtervező keretrendszerek nem nyújtanak integrált és egyszerűen használható szolgáltatásokat olyan magasszintű nyelvintegrációs feladatokhoz, mint például az azonnali és inkrementális modellszinkronizáció és az inkrementális kódgenerálás – ezeket mindig speciális kiegészítésként, akár az adott keretrendszer teljeskörű (és így igen drága) átalakításával kell megoldani.

A fenti kihívásokat gyakorlati eszközintegrációs projekteknél személyesen is tapasztalt problémákból vezettem le. Az eszközintegráció, mint probléma az összetett fejlesztési projektek nagyon heterogén eszközparkjának velejárója, és különösen a modellvezérelt fejlesztés során jelentős kihívás, mivel itt nagy mennyiségű, szemantikailag gazdag információt kell sokszor bonyolult, számos önmagában is összetett eszközből álló láncon koordináltan végigvezetni.

2. Kutatási módszer és új eredmények

Kutatási tevékenységem fő gyakorlati motivációja az eszközintegráció, és a szakterület-specifikus nyelvek tervezési költségeinek a modelltranszformáció integratív alkalmazásán keresztül csökkentése volt. A VIATRA2 transzformációk eseményvezérelt végrehajtással és a modellváltozások magasszintű feldolgozásával való kiterjesztése (1. tézis) új lehetőségeket nyitott a DSML tervező eszközök fejlett szolgáltatásainak kidolgozásához (2. tézis). Az így létrehozott transzformációkat és modellezési nyelveket gyakorlati alkalmazásként különféle eszközintegrációs problémák megoldására használtam fel (3. tézis).

2.1. Eseményvezérelt modelltranszformációk

Ahogy a gyakorlati tapasztalatom megmutatta, a modellezési nyelvek tervezésének magasszintű, hatékony és automatizált támogatása egy újfajta, a kötegelt végrehajtástól eltérő modelltranszformációs végrehajtási modell kidolgozását igényelte. A szakirodalomban publikált, kapcsolódó megoldások [GdL07, MFF⁺06] áttekintése után világossá vált ezek két alapvető korlátozó előfeltétele: (i) nagymértékben támaszkodtak nyomomonkövethetőségi modellekre (traceability models) használatára az eseményfogalom reprezentációjához, valamint (ii) működésükhöz elengedhetetlenül szükséges volt a tervező és futtatókörnyezet nagymértékű, szoros technológiai integrációja és kiegészítése (pl. a felhasználói felülethez adott visszahívó függvények segítségével voltak csak képesek a változási események regisztrációjára).

Ezen korlátozások feloldása érdekében fejlesztettem ki az eseményvezérelt gráftranszformációk általános technikáját. A megoldás alapját a kutatócsoportunkban 2006-2007-ben elkezdett, inkrementális gráfmintaillesztést megcélzó kutatómunka [22] adta, valamint adaptáltam a szakirodalomban ismert élő transzformációk (live transformations [HLR06]) ötletét is. Az általam javasolt módszer központi eleme, hogy a változási esemény fogalmát mint a modellre illesztett gráfminta illeszkedési halmazában bekövetkezett kvantitatív változást határozzuk meg, amely teljes mértékben független a transzformációs nyelvtől és a konkrét tervezési/modellezési környezet technikai részleteitől is (ill. arra csak jóval szűkebb megkötéseket tesz). Kidolgoztam egy eseményvezérelt transzformációs nyelvi dialektust is, a VIATRA2 keretrendszer transzformációs nyelvének kiegészítéseként, valamint javasoltam hozzá precíz végrehajtási szemantikát is [20]. A

megvalósított kiegészítő modul bekerült a hivatalos VIATRA2 terjesztésbe, és ma mint a rendszer eseményvezérelt végrehajtó modulja elérhető.

1. tézis. Javaslatot tettem az eseményvezérelt modelltranszformációk új specifikációs és végrehajtási módszerére. Az eseményvezérelt transzformációk a kötegelt végrehajtással ellentétben folyamatosan a háttérben futnak, és a modellek változására azonnal, felhasználói beavatkozás nélkül reagálnak.

- 1.1 Kidolgoztam az eseményvezérelt transzformációk inkrementális gráfmintaillesztésen alapuló *alapkoncepcióját* [20], ahol a modell változásait a gráfminták illeszkedési halmazainak változásaként definiáljuk. A modelltraszformációs műveletek a feldolgozás során a változás (delta) jellegű információra közvetlenül támaszkodhatnak.
- 1.2 Kidolgoztam az eseményvezérelt szabályok várakozási sorral kiegészített véges automata alapú *végrehajtási szemantikáját*, amely az esemény-feltétel-akció szabályokra építve formálisan megadja a szabályok jelentését és kiértékelésének menetét [7].
- 1.3 A VIATRA2 keretrendszer modelltranszformációs nyelvére építve *specifikációs nyelvet* dolgoztam ki eseményvezérelt transzformációkhoz ([20,7,4]), mely alapján a szabályok (gráf-triggererek) esemény-feltétel-akció hármassok segítségével írhatók le formálisan.
- 1.4 A transzformációs nyelv és szemantika alapján *hatékony megvalósítási technikákat* javoltam az eseményvezérelt transzformációk végrehajtására, és *mérésekkel igazoltam annak hatékonyságát* ([21,6,17,7]).

A későbbi, kollaboratív kutatómunka során több ízben is felhasználtuk az e tézis eredményeit ([4,2]). Az eseményvezérelt transzformációs rendszer első változatát Ökrös András MSc hallgatómmal közös munkában fejlesztettük ki. Az inkrementális gráfmintaillesztő Bergmann Gábor doktorandusz PhD munkája. Az 1. téziscsoport eredményeit a disszertáció 5. fejezete tárgyalja.

2.2. Automatikus modelltranszformációk szakterület-specifikus modellezési nyelvekben

A munkám elején végzett irodalomkutatás [35] alapján, valamint a 2004-2005 közötti időszakban végzett ipari fejlesztési tapasztalatok elemzése után Vágó Dáviddal lefektettük egy újszerű, szakterület-specifikus modellezési nyelvek tervezésére és használatára szolgáló keretrendszer alapjait – az eszköz a *ViatraDSM* nevet kapta. A keretrendszer tervezési alapelveit a korabeli eszközök gyakorlati hiányosságainak kiküszöbölése céljából határoztuk meg. Két fő problémát kívántunk megoldani: (i) a DSML-ek absztrakt és konkrét szintakszisainak teljeskörű, és magas szinten paraméterezhető szétválasztását, a korábban megjelent technikák [MFF⁺06] általánosításával, valamint (ii) magasszintű, integrált modelltranszformáció-alapú megoldás kidolgozását az olyan fejlett nyelvtervezési aspektusok támogatásához, mint a jólformáltsági kényszerek azonnali, tervezési időben történő kiértékelése és a tervezési idejű szimuláció diszkrét eseményrendszerek [SV08] alapján.

A ViatraDSM rendszer a VIATRA2 transzformációs keretrendszer modellezési és transzformációs szolgáltatásaira épülő, rugalmas és bővíthető felépítésű felületet biztosít. A rendszerbe illeszkedő szakterületi nyelveket külön definiált absztrakt és konkrét szintakszis leírások, valamint opcionális, kézi testreszabást lehetővé tevő segédmodulok (plug-in-ek) határozzák meg. A beillesztett szakterületi nyelvek modelljein a VIATRA2 segítségével, tetszőleges funkció esetén felhasználhatunk gráfminta alapú lekérdezéseket és gráftranszformáció alapú módosításokat. Az eredményeink publikálása után megjelent ipari rendszerek, mint pl. az Eclipse GMF is,

a ViatraDSM-éhez hasonló architektúrára épülnek, de továbbra is hiányzik belőlük az integrált transzformációk támogatása.

2. tézis. Kifejlesztettem egy domain-specifikus modellezési keretrendszert, melynek alapszolgáltatásai integrált eseményvezérelt modelltranszformációkra épülnek. A keretrendszer magas szintű támogatást nyújt különböző nyelvtervezési aspektusok (kiemelten az absztrakt-konkrét szintakszis ábrázolások közötti leképezések, jólformáltsági kényszerek, szimuláció, nyelvközi leképezések) tervezési idejű, hatékony végrehajtásához.

- 2.1 Kidolgoztam a *különböző szakterületekhez tartozó nyelvi modellek metamodell-bázisú integrációját*, melynek segítségével az egységes modelltérben tárolt nyelvi modellekkel egyszerre több aspektusból dolgozhatunk [19] (1. kihívás). Kapcsolódó eredményként kidolgoztam egy általános, *tetszőleges nyelv absztrakt és konkrét szintakszisa közötti nyomonkövethetőségi kapcsolatokat leírni képes metamodellező keretrendszert* is [29,7] (2. kihívás).
- 2.2 Kidolgoztam egy módszert az *absztrakt- és konkrét szintakszisának teljeskörű szétválasztására eseményvezérelt modelltranszformációk alapján* [7,23,24]. A módszer segítségével olyan vizuális nyelvek fejleszthetők, ahol a grafikus modellábrázolás tetszőleges mértékben egyszerűsítheti (tömörítheti) az absztrakt modelleket (2. kihívás).
- 2.3 Újszerű megközelítést javasoltam a *szakterületi modellezőnyelvek dinamikus szemantikájának tervezési idejű, interaktív végrehajtásához eseményvezérelt transzformációkkal megadott szabályok alapján* [23,28,27,19] (3. kihívás).
- 2.4 A javasolt módszerek hatékonyságát szisztematikus mérésekkel [21,6,17,11] igazoltam. A mérésekhez *prototipikus implementációt* készítettem, amelyek a ViatraDSM eszköz szabadon hozzáférhető változatában is elérhetőek [26].

A 2-es téziscsoport eredményeit a disszertáció 3. (2.1), 6 (2.2) és 7. (2.3-2.4) fejezetei tárgyalják. A saját munkámat bemutató eredményekhez Ökrös András MSc hallgatóm, és Vágó Dávid járult hozzá (a DSM szimulátor első előzetes, hagyományos transzformációkat és imperatív programkóddal megírt lekérdezéseket alkalmazó változatának elkészítésével).

2.3. Eszközintegráció változásvezérelt modelltranszformációkkal

Változásvezérelt transzformációk. Az eseményvezérelt transzformációk gráftrigger nyelve új-szerű, és rugalmasan használható technikát nyújt a folyamatosan futó, forrás- és célmodellek közötti szinkronizáció megvalósítására. A gyakorlati alkalmazások során azonban felmerült az igény arra, hogy a hagyományos, szinkron transzformációk kiegészítéseként (ahol a propagáció a műveletekkel szinkronban, azonnal megtörténik) lehetőséget biztosítsunk az aszinkron végrehajtáshoz is (mely esetben a szinkronizáció időben, és térben is – pl. külső táron elérhető modellek – eltolható, lásd 5. kihívás). Ezért, az eseményvezérelt transzformációk kibővített általánosításaként kidolgoztam a *változásvezérelt transzformációkat* [15]. Ebben a megközelítésben a modellek változásait változásleíró modellekké sorosítjuk, és így azok feldolgozása – bizonyos feltételek mellett – az eredeti modellektől függetlenül megvalósítható. Ezt a technikát inkrementális kódgenerálásra használtam fel a fejlesztőeszközök integrációjának céljából (lásd a MO-GENTES EU projekttel kapcsolatos információkat alább), valamint ez képezi az alapját az újabb SecureChange EU projektben kifejlesztett alkalmazásoknak is.

Eszközintegráció. A kutatási eredmények gyakorlati ipari alkalmazásaként mind az eseményvezérelt transzformációkat, mind pedig az erre épülő nyelvtervező keretrendszert felhasználtam a kutatócsoport számára, európai uniós projekteken végzett munkámban. A projektmunka közvetlen motivációt is biztosított új kutatási irányokhoz. A SENSORIA FP6 projekt keretében részt vettem a SENSORIA Development Environment (SDE [3]) nevű, általános célú eszközintegrációs keretrendszer kifejlesztésében. Az SDE elsődlegesen modell-alapú fejlesztőeszközök automatizált, együttes használatát teszi lehetővé Eclipse környezetben [KLN04]. Az első kapcsolódó eredmény a VIATRA2 transzformációs keretrendszer integrációs moduljainak megtervezése és kifejlesztése, mely lehetővé tette a gráftranszformációk általános szolgáltatásként történő meghívását egy szolgáltatásorientált architektúrát alkalmazó rendszerben. Az SDE alapszolgáltatásainak bővítéseként kifejlesztettük a távoli eljárás-hívás hatékony támogatását is, mely kiemelt fontosságúnak bizonyult a későbbi, MOGENTES EU projektbeli felhasználás során.

A MOGENTES projektben (immár önálló munkaként) továbbfejlesztettük az eszközintegrációs keretrendszert. Javaslatot tettem egy eszközintegrációs folyamatok precíz megtervezését támogató, ViatraDSM-es szakterületi modelleket alkalmazó keretrendszerre, amelyet a kutatócsoport több tagjával közösen meg is valósítottunk. A változásvezérelt transzformációk alkalmazásával kifejlesztettem a munkafolyamatok gyors prototipizálását lehetővé tevő rendszert, amelyben az inkrementális kódgenerálás segítségével a modellezőeszközben végrehajtott változtatások közvetlenül, a már telepített leírófájlból propagálódnak.

Összefoglalásként elmondható, hogy az eszközintegráció területén sikeresen alkalmaztam valamennyi kutatási eredményemet (szakterület-specifikus nyelvek, esemény- és változásvezérelt transzformációk).

3. tézis. Javaslatot tettem a *változásvezérelt transzformációs technikára*, amelyek segítségével a forrásmodellek változásait (leíró modelleket, vagy eseményobjektumokat) dolgozhatjuk fel automatikusan, és aszinkron módon továbbíthatjuk azokat akár külső tárolón található célmodellekhez. A technológiát inkrementális kódgenerálásra alkalmaztam, a modellbázisú eszközintegrációs keretrendszer folyamattervező moduljában. Az eszközintegrációs rendszerben a változásvezérelt transzformációk általános célra, az egyes szolgáltatások közötti inkrementális adat- és modellátadásra is felhasználhatóak.

- 3.1 Kidolgoztam a változásvezérelt transzformációk általános technikáját [15], amelyek a példánymodellek változásait dolgozzák fel, és segítségükkel akár külső modellek között is megvalósítható az inkrementális modelltranszformáció.
- 3.2 Új szoftver-keretrendszert javasoltam az eseményvezérelt, *inkrementális kódgenerálás* megvalósításához (5. kihívás), melynek során a generálás alapját adó forrásmodell változásait közvetlenül, inkrementális módon alkalmazzuk a telepített célnyelvi leírófájlokban [4]. Kidolgoztam egy *metamodellt a változási modellek leírására*, melynek segítségével memóriabeli és külső modelleken értelmezett atomi és összetett műveletek is modellezhetők.
- 3.3 *Hatékony megvalósítási módszereket* javasoltam a *távoli szolgáltatás-elérés* megvalósítására a SENSORIA Development Environment (SDE) eszközintegrációs keretrendszerben, valamint ezen módszereket alkalmaztam a VIATRA2 alapú modelltranszformációk távoli szolgáltatásként való végrehajtására [3].
- 3.4 Az SDE továbbfejlesztésével javaslatot tettem egy *kiterjesztett eszközintegrációs architektúrára*, melyben a modelltranszformációkat transzparens és kooperáló szolgáltatásként meghívva összetett transzformációs láncokat hozhatunk létre. Az információáramlás megvalósítására változásvezérelt transzformáció alapú technikákat javasoltam [16,1].

3.5 Az inkrementális kódgenerálási technikát egy saját fejlesztésű, munkafolyamatokat leíró modellező nyelvre alkalmaztam. Iparban szabványosított nyelvek (SPEM, EPF és jPDL) általánosításával kifejlesztettem egy *modellalapú keretrendszert a szoftverfejlesztési folyamatok precíz tervezésére* [16]. Kidolgoztam a magasszintű specifikációs nyelv, és az alacsony-szintű, végrehajtható folyamatleírónyelv közötti inkrementális kódgenerálási eljárásokat [15,4].

A hármas téziscsoport 3.1, 3.2 és 3.5 altéziseinek eredményeit a disszertáció 8-as, míg a 3.3-as és 3.4-es eredményeit a 9-es fejezet mutatja be. A C függelék részletesen bemutatja az SDE keretrendszert.

A SENSORIA Development Environment fejlesztését Philip Mayer (Ludwig-Maximilians-Universität München) koordinálta. A távoli eljáráshívás kifejlesztésében Horváth Ádám MSc hallgatóm működött közre. A Hibatűrő Rendszerek Kutatócsoport több tagja (kiemelten Dr. Polgár Balázs) járult hozzá a MOGENTES eszközüntegrációs platform kifejlesztéséhez.

3. Az eredmények gyakorlati alkalmazásai

3.1. A VIATRA2 keretrendszer eseményvezérelt transzformációvégrehajtó modulja

A VIATRA2 keretrendszer eseményvezérelt végrehajtó moduljának első változatát Ökrös András és Bergmann Gábor MSc hallgatóim segítségével készítettük el. Az ő munkájuk a BME Villamosmérnöki és Informatikai Karának Tudományos Diákköri Konferenciáján, 2007-ben kari első díjat, később, 2009-ben pedig az országos konferencián az Informatika Szekcióban is első díjat nyert (a dolgozat további konzulensei Varró Gergely és Varró Dániel voltak).

Munkánk eredménye szerves részét képezi a VIATRA2 keretrendszer Eclipse.org weboldalról is elérhető változatának, a Release 3 kódjelű verzió megjelenése (2008 szeptember) óta [VIA]. Ilyen minőségében fontos szerepet játszott kutatócsoportunk sokféle tevékenységében, részét képezte kutatási prototípusoknak pl. az eszközüntegrációs SENSORIA FP6 és MOGENTES FP7 EU projektekben [16], és újabban követelménymodelleken értelmezett adatbiztonsági tulajdonságok inkrementális vizsgálatára is használtuk a SecureChange EU FP7 projektben [4]. A keretrendszer fontos szerepet játszik a ViatraDSM eszközben, valamint a benne megtalálható kód és tervezési elvek szolgáltatták az alapot a VIATRA2 csoport egyéb eredményeihez is [HVS10].

3.2. ViatraDSM

A disszertációm domain-specifikus nyelvekre vonatkozó eredményeinek gyakorlati megvalósítását a ViatraDSM eszköz tartalmazza, mely a VIATRA2 szofvercsomag nyilvánosan is elérhető, nyílt forrású komponense. A dolgozatban részletesen ismertetett kapcsolódó eredményeket a kutatási projektekhez kapcsolódó prototipikus nyelvek fejlesztésében, valamint az oktatásban (önálló laborok és szakmai választható tárgyak) is hasznosítottuk.

3.3. Eszközüntegráció a SENSORIA és MOGENTES EU kutatási projektekben

A kiemelt részvételünkkel kifejlesztett SDE, és MOGENTES keretrendszereket elsősorban a projektpartnereink használták, szerte Európában. A SENSORIA projektben elsősorban szolgáltatás-orientált architektúrájú rendszerek tervezésének, korai analízisének és automatikus kódgenerálásának támogatására használták, a MOGENTES projektben pedig vasúti szoftverfejlesztési folyamatokban automatikus tesztgenerálásra és végrehajtásra vetettük be.

Az általunk elvégzett munkában, elsősorban a távoli szolgáltatás-elérés adaptációjában Horváth Ádám MSc hallgatóm működött közre. Az ebből készült TDK dolgozat a BME-VIK Kari TDK konferencián 3. díjat nyert 2008-ban.

3.4. Sztochasztikus modellszimulációs keretrendszer

A szakterületi nyelvek tervezési idejű modellszimulációs keretrendszerének továbbfejlesztésével, a University of Leicester kutatócsapatával szoros együttműködésben kifejlesztettünk egy általános célú, sztochasztikus rendszerszimulációs eszközt (VIATRA-GraTS [14]). A kutatás-fejlesztést a SENSORIA projekt keretében, Prof. Dr. Reiko Heckel irányításával és Dr. Paolo Torrini közreműködésével végeztük, 2009-2010 folyamán. Ezt az eszközt elsősorban peer-to-peer VoIP protokollanalízisre használták [12], de fontos szerepet játszik a leicesteri egyetem MSc és PhD szoftvermérnöki programjában is, az oktatás és labormunka részeként.

3.5. EMF-IncQuery

A VIATRA2 csoport legújabb eredménye az EMF-IncQuery [11] keretrendszer, mely a VIATRA2 technológiával kapcsolatos kutatási eredmények ([22,21]) széleskörű, az Eclipse Modeling Framework platformon történő alkalmazását teszi lehetővé. Ezáltal az ebben a disszertációban leírt eredmények – és általában a VIATRA2 technológia – számos nyílt és kereskedelmi termékhez egyszerűen integrálható lesz.

4. Kapcsolódó publikációk

Publikációk száma: 29

Lektorált publikációk száma: 24

Független idézetek közelítő száma: 50

4.1. Könyvfejezetek (3)

- [1] András Balogh, Gábor Bergmann, György Csertán, László Gönczy, Ákos Horváth, István Majzik, András Patariza, Balázs Polgár, István Ráth, Dániel Varró, Gergely Varró. Workflow-driven tool integration using model transformations. In Gregor Engels, Claus Lewerentz, Wilhelm Schaefer, Andy Schuerr, and Bernhard Westfechtel, editors, *Graph Transformations and Model-Driven Engineering*, volume 5765 of *Lecture Notes in Computer Science*, pages 224–248. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-17322-6_11.
- [2] Gábor Bergmann, Artur Boronat, Reiko Heckel, Paolo Torrini, István Ráth, and Dániel Varró. Advances in model transformation by graph transformations: Specification, Analysis and Execution. In *Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing*. Springer, 2010.
- [3] István Ráth and Philip Mayer. The SENSORIA Development Environment. In *Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing*. Springer, 2010.

4.2. Külföldön megjelent, idegen nyelvű folyóiratcikkek (5)

- [4] Bergmann Gábor, István Ráth, Gergely Varró, and Dániel Varró. Change-driven model transformations: Change (in) the rule to rule the change. *Journal of Software and Systems Modeling*, 2010. Elfogadva.
- [5] Paolo Torrini, Reiko Heckel, István Ráth, and Gábor Bergmann. Stochastic graph transformation with regions. *Electronic Communications of the EASST, Proceedings of the Ninth International Workshop on Graph Transformation and Visual Modeling Techniques*, 2010.
- [6] Ákos Horváth, Gábor Bergmann, István Ráth, and Dániel Varró. Experimental assessment of combining pattern matching strategies with VIATRA2. *International Journal on Software Tools for Technology Transfer (STTT)*, 12:211–230, 2010. 10.1007/s10009-010-0149-7.
- [7] István Ráth, András Ökrös, and Dániel Varró. Synchronization of abstract and concrete syntax in domain-specific modeling languages. *Software and Systems Modeling*, 9:453–471, 2010. 10.1007/s10270-009-0122-7.
- [8] Gábor Bergmann, István Ráth, and Dániel Varró. Parallelization of graph transformation based on incremental pattern matching. *Electronic Communications of the EASST, Proceedings of the Eighth International Workshop on Graph Transformation and Visual Modeling Techniques*, 18, 2009.

4.3. Konferenciakiadványban megjelent előadások (21)

4.3.1. Nemzetközi konferenciákon megjelent idegennyelvű előadások (17)

- [9] Gábor Bergmann, Zoltán Ujhelyi, István Ráth, and Dániel Varró. A Graph Query Language for EMF models. *Theory and Practice of Model Transformations, Proceedings of the 4th International Conference on Model Transformations (ICMT)*, 2011. Elfogadva, elfogadási arány: 25%.
- [10] Ábel Hegedüs, Zoltán Ujhelyi, István Ráth, and Ákos Horváth. Visualization of traceability models with domain-specific layouting. In *Proceedings of the Fourth International Workshop on Graph-Based Tools*, 2010.
- [11] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. Incremental evaluation of model queries over EMF models. In Dorina Petriu, Nicolas Rouquette, and Øystein Haugen, editors, *Model Driven Engineering Languages and Systems*, volume 6394 of *Lecture Notes in Computer Science*, pages 76–90. Springer Berlin / Heidelberg, 2010. Elfogadási arány: 21%; DOI: 10.1007/978-3-642-16145-2_6.
- [12] Ajab Khan, Reiko Heckel, Paolo Torrini, and István Ráth. Model-based stochastic simulation of P2P VoIP using graph transformation. In *Proceedings of the 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications*, 2010.
- [13] Ábel Hegedüs, Gábor Bergmann, István Ráth, and Dániel Varró. Back-annotation of simulation traces with change-driven model transformations. In *Proceedings of the Eighth International Conference on Software Engineering and Formal Methods*, pages 145–155, Pisa, 09/2010 2010. IEEE Computer Society, IEEE Computer Society. Elfogadási arány: 22%.
- [14] Paolo Torrini, Reiko Heckel, and István Ráth. Stochastic simulation of graph transformation systems. In David Rosenblum and Gabriele Taentzer, editors, *Fundamental Approaches*

- to *Software Engineering*, volume 6013 of *Lecture Notes in Computer Science*, pages 154–157. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-12029-9_11. Elfogadási arány: 24%.
- [15] István Ráth, Gergely Varró, and Dániel Varró. Change-driven model transformations. In Andy Schürr and Bran Selic, editors, *Model Driven Engineering Languages and Systems, 12th International Conference, MODELS 2009, Denver, CO, USA, October 4-9, 2009. Proceedings*, volume 5795 of *Lecture Notes in Computer Science*, pages 342–356. Springer, Springer, 2009. Springer Best Paper Award and ACM Distinguished Paper Award; Elfogadási arány: 18%.
- [16] Balázs Polgár, István Ráth, Zoltán Szatmári, and István Majzik. Model-based Integration, Execution and Certification of Development Tool-chains. In *2nd ECMDA Workshop on Model-Driven Tool and Process Integration*, 2009.
- [17] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. Efficient model transformations by combining pattern matching strategies. In Richard Paige, editor, *Theory and Practice of Model Transformations*, volume 5563 of *Lecture Notes in Computer Science*, pages 20–34. Springer Berlin / Heidelberg, 2009. Elfogadási arány: 23%; DOI: 10.1007/978-3-642-02408-5_3.
- [18] András Balogh, András Pataricza, and István Ráth. Automated verification and validation of domain specific languages and their applications. In *Proceedings of the 4th World Congress for Software Quality*, pages 1–6, Bethesda, USA, 2009.
- [19] István Ráth, Dávid Vágó, and Dániel Varró. Design-time simulation of domain-specific models by incremental pattern matching. In *IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008, Herrsching am Ammersee, Germany, 15-19 September 2008, Proceedings*, pages 219–222. IEEE, 2008. Elfogadási arány: 29%.
- [20] István Ráth, Gábor Bergmann, András Ökrös, and Dániel Varró. Live model transformations driven by incremental pattern matching. In Antonio Vallecillo, Jeff Gray, and Alfonso Pierantonio, editors, *Proc. First International Conference on the Theory and Practice of Model Transformations (ICMT 2008)*, volume 5063/2008 of *Lecture Notes in Computer Science*, page 107–121. Springer Berlin / Heidelberg, Springer Berlin / Heidelberg, 2008. Elfogadási arány: 31%.
- [21] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. A benchmark evaluation of incremental pattern matching in graph transformation. In Hartmut Ehrig, Reiko Heckel, Grzegorz Rozenberg, and Gabriele Taentzer, editors, *Graph Transformations*, volume 5214 of *Lecture Notes in Computer Science*, pages 396–410. Springer Berlin / Heidelberg, 2008. Elfogadási arány: 40%; DOI: 10.1007/978-3-540-87405-8_27.
- [22] Gábor Bergmann, András Ökrös, István Ráth, Dániel Varró, and Gergely Varró. Incremental pattern matching in the VIATRA transformation system. In *GRaMoT’08, 3rd International Workshop on Graph and Model Transformation*. 30th International Conference on Software Engineering, 2008.
- [23] István Ráth and Dániel Varró. Challenges for advanced domain-specific modeling frameworks. In *International Workshop on Domain Specific Program Development (DSPD 2006)*, Nantes, France, July 2006.
- [24] István Ráth. Declarative mapping between abstract and concrete syntax of domain-specific visual languages. In *The Proceedings of the Fifth Conference of PhD Students in Computer Science*, 2006.

- [25] András Balogh, Attila Németh, András Schmidt, István Ráth, Dávid Vágó, Dániel Varró, and András Pataricza. The VIATRA2 model transformation framework. *Tool demo at the First European Conference on Model Driven Architecture - Foundations and Applications*, 2005.

4.3.2. Magyar nyelvű konferencia-előadások (4)

- [26] Ráth István. Modelltranszformációk integrált alkalmazása domain-specifikus nyelvekben. In *Tavaszi Szél Konferenciakiadvány*, 2009.
- [27] István Ráth. Enhancing design-time model execution in domain-specific languages by incremental pattern matching. In *Proceedings of the 16th PhD Minisymposium*, pages 16–20. Budapest University of Technology and Economics, Department of Measurement and Information Systems, 2009.
- [28] István Ráth. Design-time simulation of domain-specific modeling languages by interactive model transformation. In *Proceedings of the 15th PhD Minisymposium*, pages 58–62. Budapest University of Technology and Economics, Department of Measurement and Information Systems, 2008.
- [29] István Ráth. Challenges for advanced domain-specific modeling frameworks. In *Proceedings of the 14th PhD Minisymposium*, pages 118–120. Budapest University of Technology and Economics, Department of Measurement and Information Systems, 2007.

4.4. Egyéb (6)

4.4.1. Technical report (4)

- [30] Budapest University of Technology and Economics. With contributions by István Ráth. Report on the Framework Implementation. *MOGENTES Project Deliverables (projektbeszámoló kutatási jelentések) D2.2.a, b, c*. 2009 június – 2011 március.
- [31] Ábel Hegedüs, István Ráth, Dániel Varró. From BPEL to SAL and Back: a Tool Demo on Back-Annotation with VIATRA2. *SEFM Posters and Tool Demo Session Track*, 09/2010. pp. 35–42. ISBN: 978-88-7958-006-9.
- [32] Ákos Horváth, Dénes Monostori, András Balogh, Imre Kocsis, Gergely Pintér, Antal Fazakas, István Ráth, Dániel Varró, Ivo Viglietti, Massimo Cifaldi and Tobias Schoofs. Report on the definition of the AIDA development means. *DIANA Project Deliverable (projektbeszámoló kutatási jelentés) DC3.1* 2010. április 20.
- [33] Philip Mayer, István Ráth, Ádám Horváth. Report on the SENSORIA Development Environment. *SENSORIA Project Deliverables (projektbeszámoló kutatási jelentések) D7.4.c–d*. 2008. augusztus 31. – 2010. január 13.

4.4.2. TDK dolgozat

- [34] Schmidt András, Ráth István, Vágó Dávid. Automated model transformations in domain specific visual languages. *Tudományos Diákköri Dolgozat*, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2005. Első díj a BME Villamosmérnöki és Informatikai Kar TDK konferenciáján (2005), első díj az Országos TDK Konferencián (2007).

4.4.3. Diplomaterv

- [35] Ráth István. Declarative specification of domain specific visual languages. Diplomaterv, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2006.

Hivatkozások

- [DAR06] Hanna Farah Daniel Amyot and Jean-François Roy. Evaluation of Development Tools for Domain-Specific Modeling Languages. *System Analysis and Modeling: Language Profiles, Springer LNCS*, 4320/2006:183–197, December 2006. DOI 10.1007/11951148_12.
- [EMF] *Eclipse Modeling Framework*. <http://www.eclipse.org/emf>.
- [FC04] David S. Frankel and Steve Cook. Domain-specific modeling and model driven architecture. *MDA Journal*, 2004. <http://www.bptrends.com/publicationfiles/01-04COLDomSpecModelingFrankel-Cook.pdf>.
- [GdL07] Esther Guerra and Juan de Lara. Event-driven grammars: Relating abstract and concrete levels of visual languages. *Software and Systems Modeling*, 6(3):317–347, 2007.
- [GME] GME. The Generic Modeling Environment. <http://www.isis.vanderbilt.edu/Projects/gme>.
- [GMF] Eclipse Graphical Modeling Framework. <http://www.eclipse.org/gmf>.
- [HLR06] David Hearnden, Michael Lawley, and Kerry Raymond. Incremental Model Transformation for the Evolution of Model-Driven Systems. In *Proc. of 9th International Conference on Model Driven Engineering Languages and Systems (MODELS 2006)*, volume 4199 of LNCS, pages 321–335, Heidelberg, Germany, 2006. Springer Berlin.
- [HV09] Ákos Horváth and Dániel Varró. CSP(M): Constraint satisfaction problem over models. In Andy Schürr and Bran Selic, editors, *Model Driven Engineering Languages and Systems, 12th International Conference, MODELS 2009, Denver, CO, USA, October 4-9, 2009. Proceedings*, volume 5795 of *Lecture Notes in Computer Science*, pages 107–121. Springer, Springer, 2009. Acceptance rate: 18%.
- [HVS10] Ákos Horváth, Dániel Varró, and Tobias Schoofs. Model-driven development of ARINC 653 configuration tables. In *29th IEEE & AIAA Digital Avionics System Conference (DASC)*, Salt Lake City, US, 10/2010 2010. IEEE, IEEE.
- [KLN04] Gabor Karsai, Andras Lang, and Sandeep Neema. Design patterns for open tool integration. *Software and Systems Modeling*, 4(2):157–170, 2004.
- [LLMC04] Tihamér Levendovszky, László Lengyel, Gergely Mezei, and Hassan Charaf. A systematic approach to metamodeling environments and model transformation systems in VMTS. In *Proc. GraBaTs 2004: International Workshop on Graph Based Tools*. Elsevier, 2004.
- [MDA01] *The Object Management Group: Model Driven Architecture — A Technical Perspective*, September 2001. <http://www.omg.org>.
- [MEP] MetaCase MetaEdit+. <http://www.metacase.com/mep/>.

- [MFF⁺06] Pierre-Alain Muller, Franck Fleurey, Frédéric Fondement, Michel Hassenforder, Rémi Schneckenburger, Sébastien Gérard, and Jean-Marc Jézéquel. Model-Driven Analysis and Synthesis of Concrete Syntax. *Model Driven Engineering Languages and Systems, Springer LNCS*, 4199/2006:98–110, November 2006. DOI 10.1007/11880240_8.
- [Mic] Microsoft. DSL Tools. <http://lab.msdn.microsoft.com/teamsystem/workshop/dsltools/default.aspx>.
- [SV08] Eugene Syriani and Hans Vangheluwe. Programmed graph rewriting with DEVS. In Andy Schürr, Manfred Nagl, and Albert Zündorf, editors, *Applications of Graph Transformations with Industrial Relevance*, volume 5088 of *Lecture Notes in Computer Science*, pages 136–151. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-89020-1_11.
- [VIA] VIATRA2 Framework. An Eclipse GMT Subproject (<http://www.eclipse.org/gmt/>).