

MODELLTRANSZFORMÁCIÓK INTEGRÁLT ALKALMAZÁSA DOMAIN-SPECIFIKUS NYELVEKBEN

Ráth István

BME Méréstechnika és Információs Rendszerek Tanszék

Absztrakt

A cikkben bemutatjuk egy új, a VIATRA2 **Error! Reference source not found.** modelltranszformációs keretrendszer szolgáltatásaira épülő szakterület-specifikus modellezési környezet alapjait. Megközelítésünk fő újdonsága, hogy a modelltranszformációs alapszolgáltatások szoros integrációjával magas szintű támogatást biztosítunk fontos nyelvtervezési aspektusokhoz (pl. absztrakt-konkrét szintakszis automatikus megfeleltetése, jólformáltsági kényszerek inkrementális kiértékelése, interaktív modellvégrehajítás). A hatékony végrehajtáshoz a VIATRA2 inkrementális gráfmintaillesztőjét használjuk fel.

Bevezetés

A szakterület-specifikus nyelvtervezés (domain-specific modeling) legfőbb célja, hogy a fontos nyelvtervezési aspektusokat integrálja egy koherens keretrendszerbe. A legfontosabb aspektusok a következők: *absztrakt szintakszis*, amely a nyelvi modellelemek gépi tárolási szabályrendszerét határozza meg (metamodellezés); *konkrét szintakszis*, amely a nyelvi elemek grafikus jelölését határozza meg; *jólformáltsági kényszerek*, amelyek az absztrakt szintakszishoz képest összetettebb helyességi szabályokat fogalmaznak meg, és kiemelt fontosságúak a modellek értelmezése szempontjából; *végrehajítási szemantika*, amely a modellek dinamikus viselkedését, interakcióit specifikálja; *transzformációs leképezések*, amelyekkel nyelvközi transzformációkat, illetve az egyes modellezési nézetek közötti leképezéseket definiálhatunk (ilyen például a kódgenerálás).

A jelenlegi eszközökben, mint például a MetaCase MetaEdit+, a Microsoft DSL Tools, openArchitectureWare vagy az Eclipse Graphical Modeling Framework, kifejtett támogatást találhatunk az első két aspektushoz: a nyelvi metamodellek könnyű létrehozására és a grafikus megjelenítés precíz, látványos megvalósítására törekednek. Vannak azonban olyan problémák, amelyek a gyakorlatban megnehezítik ezen nyelvek létrehozását és használatát. A legfontosabb ilyen az absztrakt és konkrét szintakszis ábrázolások közötti leképezés, melynek lényege, hogy a grafikai megjelenítés gyakran az absztrakt szintakszishoz kötött (azaz, strukturálisan hasonló, lényegében a legtöbb eszköz az absztrakt modellek közvetlen gráfleképezését támogatja). Ezáltal a komplex, sok elemből álló modellek megjelenítése túlságosan bonyolulttá válik, és elveszítjük a szakterület-specifikus modellezés legfontosabb előnyét, a magas absztrakciós szintből adódó

egyszerűséget és tömörséget. Néhány eszközben ezért külön modellezési réteget használnak a kétféle modellreprezentáció tárolására, azonban a közöttük létesíthető automatizált megfeleltetés kötött, nem szabadon definiálható és ez behatárolja a nyelvkialakítás rugalmasságát.

Egy további fontos probléma, hogy a jólfomáltsági kényszerek kezelése a ma elterjedt nyelvtervezési eszközökben általában a statikus (metamodellben megfogalmazott) szabályok szerkesztéskori, illetve pl. az Object Constraint Language (OCL) kényszerleíró nyelven megfogalmazott szabályok kötegelte kiértékelésére szorítkozik. A kiértékelés drága végrehajtása miatt az interaktív alkalmazások általában megelégednek a lokális kényszerek karbantartásával, melyek gyorsan, szerkesztés közben lefuttathatók. Nagy igény lenne azonban olyan rendszerekre, melyekkel az alkalmazási terület szempontjából fontosabb összetett kényszerek is szerkesztés közben kiértékelhetők, ezáltal biztosítva azt, hogy külön validáció nélkül is helyes modelleket rajzolhassunk.

Végezetül, harmadik kiemelten fontos területként az integrált modelltranszformációkkal és felhasználásukkal foglalkozunk. Ma még elvétve található fejlett modelltranszformációs szolgáltatás a modellezési eszközökben, vagyis a transzformációkat általában elkülönítve, speciális nyelveken írják, amelyek a modellek előzetes feldolgozását (import) és utólagos konverzióját (export) igénylik. A modelltranszformációk integrált alkalmazásával fejlett szolgáltatásokat lehetne a modellezési környezetből közvetlenül elérhető módon biztosítani, mint például a beágyazott rendszerek tervezésekor gyakran használt modellszimuláció (végrehajtás, animáció, model execution). A modellvégrehajtás segítségével a tervező nemcsak statikus modellábrázolásokat készíthet, hanem tervezés közben ki is próbálhatja a rendszer dinamikus viselkedését (a modellezési nyelvnek megfelelő absztrakciós szinten).

A cikkben bemutatjuk a ViatraDSM szakterület-specifikus modellezési keretrendszert, amely a VIATRA2 modelltranszformációs motor szolgáltatásaira alapozva hatékony, de ugyanakkor magas absztrakciós szintű és integrált szolgáltatásokkal valósítja meg a fentiekben említett célkitűzéseket. A ViatraDSM a teljes VIATRA2-höz hasonlóan, modularizált Eclipse plug-inek formájában érhető el (<http://eclipse.org/gmt/VIATRA2>).

Modellezési nyelvek tervezése transzformációkkal

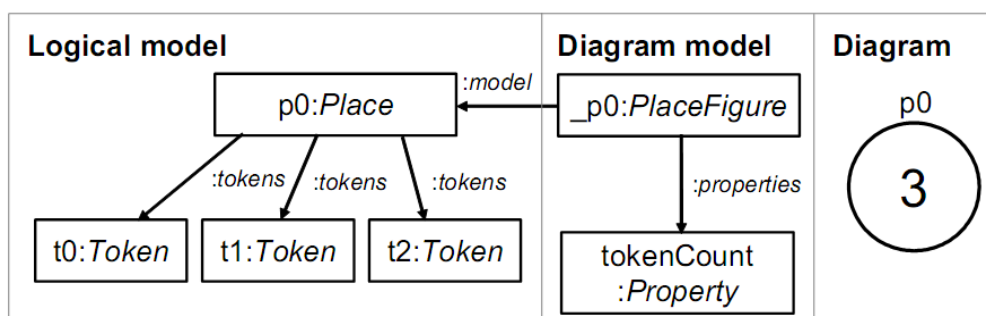
A jelen cikkben bemutatott legfontosabb új eredményünk lényege, hogy a integráljuk a szakterület-specifikus modellezést és a modelltranszformációt egy egységes keretrendszerben, ezáltal hatékony, de ugyanakkor magas absztrakciós szintű megoldást kínálunk a legfontosabb nyelvtervezési aspektusokhoz.

A technológia központjában a VIATRA2-höz készített eseményvezérelt transzformációs végrehajtó keretrendszer (*live transformation engine*, [7]). A

hagyományos, kötegelt végrehajtást alkalmazó, jelenleg elterjedt modelltranszformációs végrehajtással szemben az eseményvezérelt transzformációk folyamatosan, az egyes végrehajtási fázisok között is megőrzik a végrehajtási állapotot. Ezáltal lehetővé válik a transzformációk inkrementális, folytonos végrehajtása (pl. egy egyszerű egyirányú leképezés esetén a forrásnyelvi modellben végrehajtott változásokra a transzformációs rendszer azonnal reagálhat és létrehozhatja a megfelelő célmodellbeli változásokat). A végrehajtás gyorsaságát és hatékonyságát a VIATRA2 inkrementális mintaillesztője [3] biztosítja, amely a gráfminták illeszkedési halmazait hatékonyan gyorsítótárazza, és így a transzformációk végrehajtásának legköltségesebb, mintakeresési fázisa szinte azonnal végrehajtható (a gyorsítótárból történő olvasással). Az inkrementális mintaillesztő biztosítja a gyorsítótárak konzisztens állapotának megőrzését is: minden elemi modellmanipulációs művelet után frissül és így az egyes gráfminták illeszkedési halmazának változását nyomon követi. Az eseményvezérelt transzformációk specifikációs formalizmusának is ez az alapja: az esemény fogalmát egy-egy új illeszkedés megjelenésével vagy egy már létező illeszkedés eltűnésével definiáljuk.

Absztrakt-konkrét szintakszis szinkronizációja

A ViatraDSM keretrendszerben a korábbi rendszerekkel szemben egy magasabb absztrakciós szintű, modelltranszformációs megközelítést javasolunk az absztrakt-konkrét szintakszis reprezentációk közötti szinkronizációhoz, az eseményvezérelt formalizmus segítségével. A leképezéshez a nyelvtervező teljes szabadságot kap, mivel tetszőleges megfeleltetés megadható a VIATRA2 transzformációs nyelven [1] leírt eseményvezérelt szekvenciákkal – így például az elterjedt Eclipse-alapú GMF keretrendszerrel gazdagabb, rugalmasabban használható grafikus nyelveket hozhatunk létre.



1. ábra

Egy gyakori absztrakciós leképezést szemléltet az 1. ábra. Ebben a példában a Petri hálók nyelvéhez készített szerkesztőn mutatunk be egy gyakran használatos

absztrakciót: egy adott hely típusú elemhez kapcsolódó token példányok az absztrakt szintakszisban több különálló modellelemként, míg a konkrét szintakszisban számuk egy attribútum értéként jelenik meg (így „rejtjük el” a felhasználó elől a vizualizáció szempontjából nem fontos, sőt a rajzot feleslegesen bonyolító elemeket).

A felhasználó mind az absztrakt, mint a konkrét szintakszis modelleken végezhet módosításokat. Az eseményvezérelt transzformációk kétirányú szinkronizációt valósítanak meg, folyamatosan figyelve a modellek változásait konzisztensen tartják mindkét oldalt, azaz a konkrét szintakszisbeli attribútum érték aktuális állásához igazítják az absztrakt szintakszisbeli tokenpéldányok számát és fordítva. A transzformációs szabályokban megfeleltetési modelleket (correspondence model) használunk az egyes összekapcsolt elemek párosítására.

Jólformáltsági kényszerek inkrementális kiértékelése

A ViatraDSM rendszerben bevezetett integrált inkrementális mintaillesztő alkalmazásával lehetőség nyílik arra, hogy összetett (globális) kényszereket is inkrementálisan, a már kiszámolt eredmények teljes újraszámítása nélkül értékeljük ki. A kényszereket VIATRA2 gráfminták formájában fogalmazhatjuk meg. A mintanyelv a többszörösen egymásba ágyazható negatív alkalmazási feltételek segítségével univerzális és egzisztenciális kvantorokat használó elsőrendű logikai nyelvként alkalmas a teljes modellre kiterjedő, összetett jólformáltsági kényszerek leírására. Ahhoz, hogy a modell validációs állapotáról a felhasználói felületre kivezethető visszajelzést szolgáltatson, a ViatraDSM nyomkövetési modelleket (trace model) használ, melyek referenciákkal megjelölik az adott kényszert nem kielégítő (invalid) modellelemeket (a technika részletes ismertetése a [7] cikkben olvasható).

Interaktív modellvégrehajtás

A ViatraDSM rendszerben a modellvégrehajtást diszkrét modellek esetén értelmezzük, azaz olyan esetekben, ahol a rendszer teljes életciklusa kifejezhető véges számú diszkrét modellelem állapottereinek direkt szorzataként (az egyes elemek esetén támogatjuk az életciklus modellezését is, azaz a születés-halálozás típusú szimulációkat). A szakterület-specifikus nyelvek végrehajtási szemantikája gyakran ilyen formalizmusokra képezhető le (pl. állapotgépek, token játékok, adatfolyam hálózatok). A diszkrét modellvégrehajtás alapegységeit a VIATRA nyelven értelmezett gráftranszformációs szabályrendszerekkel írjuk le. Ezek nyelvspecifikus modellmanipulációs szabályok, amelyek a nyelv jólformáltsági kényszereit követve írják le az egyes végrehajtási lépéseket (pl. tüzelési szabályok, állapotátmenetek). Kiemelendő, hogy a végrehajtás integráltan, a tervezési

környezeten belül történik, azaz miközben a tervező felépíti a modelleket, rögtön ki is próbálhatja a dinamikus viselkedést – az eredmények közvetlenül a szakterület-specifikus modelleken láthatók (a technikához kapcsolódó alkalmazási példa részletes leírása a [12] cikkben olvasható).

Kapcsolódó munkák

A korábban említett szakterület-specifikus modellezési környezeteken kívül számos akadémiai és ipari eszköz létezik, közöttük olyanok is, amelyekben nyújtanak modelltranszformációs szolgáltatásokat. A TIGER projekt [5] (amely a 90-es évekbeli GenGEd [2] koncepcionális folytatása) alapvetően a szintakszisvezérelt szerkesztők automatikus generatív származtatására koncentrál (Eclipse plug-inek formájában). Legújabb formájában GMF-alapú szerkesztőket is képes generálni, amelyek *egyetlen* transzformációs szabállyal írják le szerkesztési szabályokat (vezérlési folyamat leírás lehetőségének hiányában nem lehet komplex, szimulációs jellegű szabályokat definiálni). A DiaMeta (a DiaGen [10] folytatása) projekt a korábbi hipergráf nyelvtanok helyett MOF specifikációs modelleket használ (a Moflon eszköz segítségével), látványos és hatékony vizualizációs technikákat használó grafikus szerkesztők generálására. Az absztrakt-konkrét szintakszis szinkronizációja specifikációs modellekből generált kódok alapján történik (rögzített szemantika alapján); online kényszer-kielégítési és szimulációs szolgáltatásokat nem tartalmaz. A Generic Modeling Environment [8], a GReAT modelltranszformációs keretrendszerrel [9] összekapcsolva szemantikus kapcsolás segítségével nyújt szimuláció-*specifikáció* támogatást szakterület-specifikus nyelvekhez. Fejlett metamodellezési szolgáltatásokat nyújt az ATOM3 [6] nevű modellezőeszköz. A modellek különböző konkrét szintakszisbeli nézetei és az absztrakt szintakszis közötti leképezést hármass gráfnyelvtanok (triple graph grammars) segítségével valósítja meg, valamint kezdetleges szimulációs támogatással is rendelkezik.

Összefoglalásként elmondhatjuk, hogy bár számos egyéb eszközben található támogatás a ViatraDSM eszközben megcélzott nyelvtervezési aspektusokhoz, a mi eszközünk ezeket integráltan, magas absztrakciós szinten formálisan specifikálható transzformációs programok formájában kezeli, és hatékony, inkrementális végrehajtási mechanizmussal támogatja a könnyű használhatóság és testreszabhatóság segítése érdekében.

Hivatkozások

- [1] D. Varró, A. Balogh: The Model Transformation Language of the VIATRA2 Framework. Science of Computer Programming 68(3) (October 2007) pp. 214–234.

- [2] R. Bardohl and H. Ehrig. Conceptual model of the graphical editor GENGED for the visual definition of visual languages. In Proc. Theory and Application to Graph Transformations (TAGT'98), Springer, 2000
- [3] G. Bergmann, A. Ökrös, I. Ráth, D. Varró, and G. Varró. Incremental pattern matching in the VIATRA transformation system. In: 3rd International Workshop on Graph and Model Transformation. 30th International Conference on Software Engineering, 2008.
- [4] J. de Lara and H. Vangheluwe. AToM3: A tool for multiformalism and meta-modelling. In Proc. 5th International Conference, Fundamental Approaches to Software Engineering, Grenoble, France, Springer, 2002.
- [5] K. Ehrig, C. Ermel, S. Hansgen, and G. Taentzer. Generation of visual editors as Eclipse plug-ins. In 20th IEEE/ACM International Conference on Automated Software Engineering, USA, ACM, 2005.
- [6] E. Guerra and J. de Lara. Event-driven grammars: Towards the integration of meta-modelling and graph transformation. In Proc. Graph Transformations, Second International Conference, Rome, Italy, Springer, 2004.
- [7] I. Ráth and G. Bergmann and A. Ökrös and D. Varró. Live model transformations driven by incremental pattern matching. 4th International Conference on Model Transformations, 2008.
- [8] G. Karsai, A. Agrawal, F. Shi, and J. Sprinkle. On the use of graph transformation in the formal specification of model interpreters. Journal of Universal Computer Science, 2003.
- [9] O. Köth and M. Minas. Generating diagram editors providing free-hand editing as well as syntax-directed editing. In GRATRA 2000 Joint APPLIGRAPH and GETGRATS Workshop on Graph Transformation Systems, Berlin, Germany, 2000.
- [10] M.Minas. Generating visual editors based on FUJABA/MOFLON and DiaMeta. Technical report, University Paderborn, 2006. Proc. 4th Fujaba Days, pp. 35-42, Technical Report tr-ri-06-275.
- [11] A. Schürr. Specification of graph translators with triple graph grammars. In Proc. WG94: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 1994.
- [12] I. Ráth, D. Vágó, D. Varró: Design-time simulation of domain-specific models by incremental pattern matching. In Proc. The 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, Herrsching am Ammersee, Germany, 2008.