

FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2003. március 21-22.

A VIATRA MODELLTRANSZFORMÁCIÓS RENDSZER

Varró Dániel, Domokos Péter

Despite the Unified Modeling Language (UML) has recently become the de facto standard object-oriented modeling language in software engineering, the lack of a precise semantics for UML hinders its use especially in safety critical applications where a formal proof of correctness and consistency of the system model is indispensable. In the paper, we present the VIATRA model transformation system, which provides a means to automated and provenly correct transformations of UML-based system models into equivalent mathematical descriptions (such as Petri nets, finite automata, process algebra) by hiding the entire mathematical background from systems engineers during verification and validation of the design.

Noha az UML (Unified Modeling Language) modellezési nyelve napjainkban az objektum-orientált szoftvertervezés de facto szabványává vált, precíz szemantikájának hiánya azonban komoly nehézségeket okoz különösen szolgáltatásbiztos és biztonságkritikus alkalmazások tervezése során, amikor elengedhetetlen a rendszer helyességének formális analízise. A dolgozatban bemutatjuk a VIATRA modelltranszformációs rendszert, mellyel UML modellek automatikusan és bizonyítottan helyesen transzformálhatók különféle matematikai leírásokká (pl. Petri hálók, véges automaták, processz algebrák), elrejtve a mérnökök által nehezen kezelhető matematikai háttérrel az UML bázisú rendszermodellek verifikációja és validációja során.

Bevezetés

Napjainkra az UML (Unified Modeling Language) [8] modellezési nyelve, az objektum-orientált rendszertervezés de facto szabványává vált. Az UML sikere egyrészt szabványosságában, másrészt grafikus mivoltában rejlik. Szabványossága révén megszűnt az egyedi tervezési metodikákra korábban jellemző zűrzavar, hiszen a tervező és a programozó közös nyelvet használ a kommunikációra. Az UML nyelv közérthetőségét pedig grafikus jellege adja, a szoftverrendszereket különféle nézetekből *grafikus diagramok* segítségével lépésről lépésre finomítva tervezhetjük.

A mindennapi használat során azonban számos probléma merült fel az UML-lel kapcsolatban mind ipari, mind akadémiai felhasználásokban. Egyrészt az UML nyelv egysíkú, monolit volta miatt *nehezen hangolható speciális alkalmazási területekre* (például valós idejű alkalmazások, ütemezés), amely "házi szabványok" létrejöttéhez vezetett. A legfontosabb problémát azonban a *precíz szemantika* hiánya jelenti: noha a grafikus nyelv *elemkészlete precízen definiált* (formálisan), az egyes *elemek jelentését informális* (angol) nyelven definiálták. Ennek következtében a nyelv egyes elemei félreértelmezhetők: mást érthet rajta a tervező, és mást a programozó.

Formális módszerek az informatikában

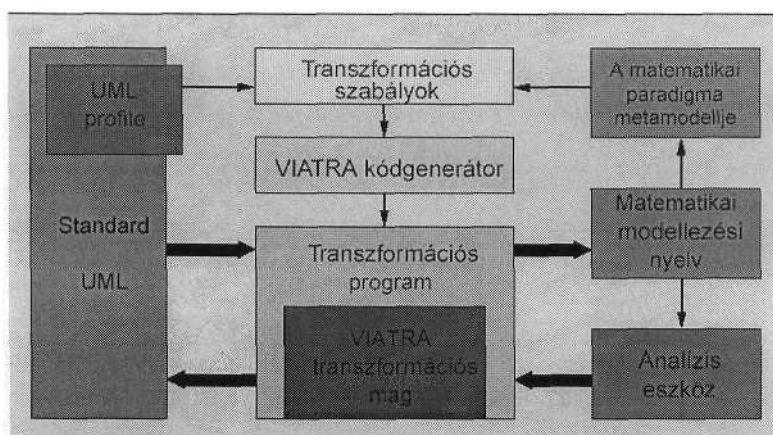
A precíz szemantika hiánya még hangsúlyosabban jelentkezik a nagymegbízhatóságú (pl. telekommunikációs és banki rendszerek) és biztonságkritikus alkalmazások (mint vasúti biztonságfelügyelő rendszerek, légitrafik-irányítás) esetén, ahol az előírt szolgáltatás elmulasztása anyagi ill. emberi veszteségekkel járhat. Ilyen rendszerekben a hagyományos tesztelés alapú hibakeresés eredményessége nem kielégítő, mivel minden hiba megfelelő tesztesettel való lefedése lehetetlen. E rendszerek **verifikációja** és **validációja** során matematikai módszerekkel kell bizonyítanunk, hogy a tervezett rendszer kielégítően fog viselkedni.

Szolgáltatásbiztos rendszerek matematikai verifikációját **formális módszerek** segítségével végezhetjük, de e módszerek használatának széleskörű elterjedését számos tényező gátolja. Használatukhoz egyrésztől nagyfokú matematikai jártasság szükséges, amely ritkán várható el egy rendszertervezőtől. Nem garantált továbbá a kapott *matematikai modellek valóságűsége*, valamint a tervezés alatt álló *rendszermodell és matematikai modelljének konzisztenciája* sem.

A VIATRA modelltranszformációs rendszer

A **VIATRA (Visual Automated model TRAnsformations)** modelltranszformációs rendszer [2,12] célja, hogy **automatikus, és bizonyítottan helyes modelltranszformációk** segítségével elérjük a matematikát a rendszermérnökök előtt [1], áthidalva ezáltal a grafikus rendszermodellek közérthetősége és a formális, matematikai modellek precizitása közti szakadékot.

A modelltranszformáció alapú rendszerverifikáció összefoglaló vázlatát az 1. ábrán látható.



1. ábra. Modelltranszformáció alapú rendszerverifikáció

Az alapgondolat a következő: az UML alapú rendszermodellt automatikus modelltranszformációk segítségével különféle matematikai modellekké (Petri hálók, Kripke automaták, adatfolyamhálók) alakítjuk, majd a matematikai modelleken elvégzett formális analízis eredményét szintén

automatikusan vetítjük vissza a rendszermodellbe, elfedve ezáltal a rendszertervező elől a teljes matematikai hátteret.

Metamodellezés. Ennek érdekében először is szükségünk van az egyes *modellezési nyelvek* (UML, mint forrásnyelv és a matematikai paradigmák, mint célnyelvek) és *modelljeik egységes és precíz leírására*. A modellek és modellezési nyelvek egységes tervezését *metamodellezésnek* hívják, melynek támogatására megalkottuk a VPM módszertanát [11], amely a matematikai definíciók mintájára kiterjeszti az öröklés fogalmát a modellezési nyelvek szintjére. Tapasztalataink azt mutatják, hogy a metamodel bázisú definíciók közérthetőbbek és kifejezőbbek a mérnöki alkalmazásterületeken, mint a tiszta matematikai leírasmód.

Szabályok: gráftranszformáció. A transzformációkat gráftranszformációs szabályok által specifikáljuk [12]. A gráftranszformációs szabályok [7] a számítástechnikai nyelvészet Chomsky által bevezetett nyelvtani szabályaihoz hasonlatosak, azonban a gráftranszformáció az utóbbival ellentétben szöveges minták helyett gráfminták manipulációját végzi. A gráftranszformációs szabályok egy UML dialektus (Profile) segítségével grafikusan adhatók meg egy tetszőleges UML CASE eszközben, jól illeszkedve ezáltal a metamodellezési környezethez. Az elkészült leírásokat az UML szabványos XMI formátumán keresztül táplálhatjuk be a VIATRÁ-ba, ezáltal egy nyílt, konkrét CASE eszközöktől független architektúrát kapunk.

Automatikus programgenerálás. E grafikus specifikációból a transzformáció algoritmus automatikusan generálható egy Prolog program formájában. A programgenerálás reflektív módon több modelltranszformációs lépésen keresztül valósul meg [9], mely által a modelltranszformációs szabályok *implementációját* szintén modelltranszformációs szabályok által *specifikáljuk*.

Formális helyességellenőrzés. A modelltranszformációk helyességét mind szintaktikai, mind szemantikai szinten vizsgálhatjuk. *Szintaktikai helyesség* esetén azt vizsgáljuk, hogy a generált modell vajon tényleg a célnyelv egy jólformált mondata-e. A transzformációk szintaktikus helyességét *planner algoritmusok* segítségével vizsgálhatjuk [12]. *Szemantikailag helyes* transzformáció esetén a forrásnyelvben előírt szemantikai tartalmat (pl. állapotok elérhetősége) kell megőriznünk a célnyelvben. A szemantikus helyességvizsgálathoz *modellellenőrzési módszereket* használhatunk, melynek során a gráftranszformációs szabályrendszert tranzíciós rendszerekké képezzük le [10].

Fontosabb alkalmazások. A VIATRA rendszer sikerrel került felhasználásra több hazai, ipari partnerekkel közös projektben. *UML állapottérképek helyességellenőrzése* során a modell operációs szemantikáját hierarchikus automaták segítségével specifikálhatjuk [4]. Az UML állapottérkép - hierarchikus automata transzformációja mintegy 40 szabályt használ. Biztonságkritikus környezetben

szükséges az *UML állapotterképek teljességellenőrzése*, melynek [5] egy modelltranszformáció alapú megközelítést adja a VIATRA rendszer felhasználásával. *UML modellek Petri háló alapú analizisét* célozza [6], melynek transzformációja szintén a VIATRA rendszerben készült. További transzformációk készültek el *UML alapú rendszertervek hibamodellezésére*, valamint az *absztrakt modellreprezentációk grafikus szintakszisának automatikus szintézisére* [3].

Hivatkozások

- [1] A. Bondavalli, M. Dal Cin, D. Latella, I. Majzik, A. Pataricza and G. Savoia: *Dependability Analysis in the Early Phases of UML Based System Design*. Int. Journal of Computer Systems - Science & Engineering, Vol. 16 (5), Sep 2001, pp. 265-275.
- [2] Gy. Csertán, G. Huszerl, I. Majzik, Zs. Pap, A. Pataricza, D. Varró: *VIATRA - Visual Automated Transformations for Formal Verification and Validation of UML Models*. In Proc. 17th Int. IEEE Conference on Automated Software Engineering (ASE 2002), Edinburgh, Scotland, 23-27 September 2002, IEEE CS Press, pp. 267-270, 2002.
- [3] P. Domokos, D. Varró: *An Open Visualization Framework for Metamodel-Based Modeling Languages*. In Proc. GraBaTs 2002, Int. Workshop on Graph-Based Tools, pp. 78-87, Vol. 72 (2), ENTCS, Elsevier, Barcelona, Spain, October 2002,
- [4] D. Latella, I. Majzik, M. Massink: *Automatic Verification of a Behavioural Subset of UML Statechart Diagrams Using the SPIN Model-checker*. Formal Aspects of Computing, Volume 11 Issue 6 (Springer) pp 637-664, 1999.
- [5] Zs. Pap, I. Majzik, A. Pataricza: *Checking General Safety Criteria on UML Statecharts*. Proc. 20th International Conference on Computer Safety, Reliability and Security 2001 (SafeComp 2001), LNCS 2187, Springer, pp. 46-55, September 25-27, 2001.
- [6] A. Pataricza: *Semi-decisions in the validation of dependable systems*. Supplementary of the Proc. IEEE DSN'01, The IEEE Int. Conference on Dependable Systems and Networks, pp. 114-115, Göteborg, Sweden, 30. June - 4. July 2001.
- [7] G. Rozenberg (ed.): *Handbook of Graph Grammars and Computing by Graph Transformation*. Volume 1: Foundations. World Scientific, 1999.
- [8] J. Rumbaugh, I. Jacobson, G. Booch: *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.
- [9] D. Varró: *Automated Program Generation for and by Graph Transformation Systems*. In Proc. AGT 2002, Workshop on Applied Graph Transformation, pp. 161-173., April 12-13., Grenoble, France, 2002
- [10] D. Varró: *Towards Symbolic Analysis of Visual Modeling Languages*. In Proc. GT-VMT 2002: Int. Workshop on Graph Transformation and Visual Modelling Techniques pp. 57-70, Vol. 72 (3), ENTCS, Elsevier, Barcelona, Spain, October 2002.
- [11] D. Varró, A. Pataricza: *Metamodeling Mathematics: A Precise and Visual Framework for Describing Semantic Domains of UML Models*. Proc. UML 2002, Int. Conference on UML, LNCS 2460, Springer, pp. 18-33, Dresden, Germany, October, 2002
- [12] D. Varró, G. Varró, A. Pataricza: *Designing the automatic transformation of visual languages*, Science of Computer Programming 44 (2) 2002, pp. 205-227.

Varró Dániel (doktorandusz)
e-mail: varro@mit.bme.hu

Domokos Péter (egyetemi hallgató)
e-mail: pdomokos@mit.bme.hu

Méréstechnika és Információs Rendszerek Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem
H-1521, Budapest, Magyar tudósok körútja 2.
Tel: +36 1 463 3579, Fax: +36 1 463 2667