

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií



Nástroje meta-CASE

(charakteristika, vývoj, přehled trhu, trendy)

Seminární práce z předmětu 4IT450

Přednášející: doc. Ing. Václav Řepa, CSc.

Jakub Laušman (xlauj05)

Lukáš Kulhavý (xkull08)

Jiří Tománek (xtomj41)

Letní semestr: 2008/2009

Obsah:

1. Úvod	2
2. Co je vlastně metamodel?	3
3. Skládání metamodelů	3
4. Druhy metamodelů	4
4.1. GOPRR.....	4
4.1.1. Podporované datové typy	4
4.1.2. Popis konceptu GOPRR	5
4.2. COMMA	6
4.2.1. Popis konceptu COMMA.....	6
4.3. MOF	7
5. Pro a proti Metamodelování	9
6. Domain specific modeling	11
6.1. UML Profily.....	11
7. Katalog meta-CASE nástrojů	11
7.1. Alfabet, IPSYS Toolbuilder	12
7.2. CA Erwin Data Modeler	12
7.3. MetaMill, ArgoUML.....	12
7.4. MetaView	12
7.5. ConceptBase.....	12
7.6. Coral	13
7.7. OpenSoul Metamodeler.....	13
7.8. MetaEdit+.....	13
7.9. Microsoft DSL Tools	13
7.10. GME.....	14
7.11. GEMS.....	14
8. Závěr.....	16
9. Zdroje	17

1. Úvod

Tato práce navazuje na práce vypracované v minulých semestrech. Cílem je nastínit současnou situaci na trhu s meta-CASE nástroji, a kterým směrem míří další vývoj. Práce je rozdělena na část teoretickou o metamodelování, a na část věnovanou jednotlivým metamodelovacím nástrojům, které jsou momentálně k dispozici. Druhá část představuje přehled nástrojů zmíněných v dřívějších pracích a nově nástroj GEMS projektu Eclipse.

2. Co je vlastně metamodel?

Tento pohled používají autoři UML a autoři CASE nástrojů - nedívají se na UML jako na diagramy, pro ně je základem UML metamodel (diagramy jsou pouze grafickou reprezentací metamodelu). Při tomto přístupu se často používá pojem model místo pojmu diagram, např. místo diagramu tříd se používá pojem model tříd. Metamodel se popisuje pomocí Meta-Object-Facility (MOF) - abstraktního jazyka pro specifikaci, vytváření a správu metamodelů (další standard OMG). Pro výměnu metamodelů se používá XMI - na XML založený standard (součást standardu UML).¹

Metamodel je složen z předpony meta a slova model. Předpona meta pochází z řečtiny a znamená v překladu „za“, „po“, „vně“, „mimo“ nebo „o jeden stupeň abstrakce výše“. Stejně jako u metadat, kde představují data o datech, je metamodelování o modelech modelů. Metamodelování zobrazuje abstraktně model o úroveň výš než model, který popisuje. Pokud aplikujeme definici platnou pro model, stává se metamodel zjednodušením modelu za určitým účelem. Metamodely lze dále vrstvit.

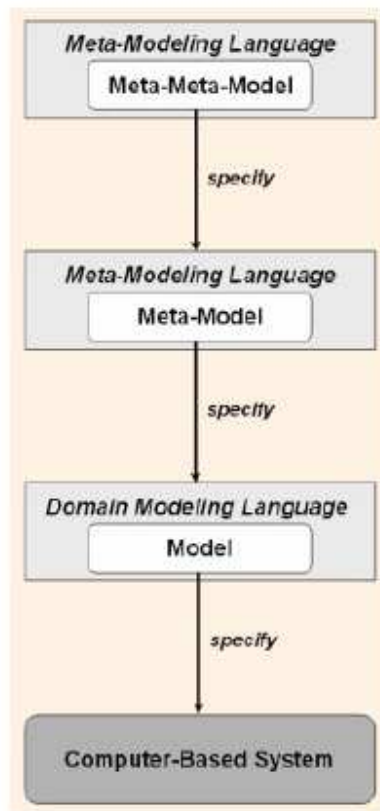
3. Skládání metamodelů

Pakliže děláme nějaký model, tak je možné k němu vytvořit i jeho metamodel. Tato architektura může ale pokračovat i dále a to sice do úrovně metamodelu metamodelu... neboli metametamodelu. Metametamodel většinou obsahuje seznam a popis objektů, které se mohou v metamodelu a dále v modelech vyskytovat. Metamodel pak specifikuje, jak jsou tyto objekty využity. Např. metamodel diagram tříd v UML je postaven na metamodelu MOF. MOF popisuje, že metamodely na něm postavené obsahují třídy, objekty, vztahy atd.²

Jak může vypadat vrstvení metamodelů, můžeme vidět na obrázku níže:

¹ Viz. zdroj č. 1

² Viz. zdroj č.2 str. 4



3

Tímto způsobem můžeme metamodely vrstvit prakticky do nekonečna, ale sami uznáte, že to by bylo velmi nepřehledné... V praxi se běžně užívá kolem 3 úrovní, ale samozřejmě záleží na každém, jak moc veliké vrstvení mu vyhovuje.

4. Druhy metamodelů

4.1. GOPRR

Hlavními modelovacími nástroji konceptu GOPRR jsou následující: Graph, Object, Property, Role and Relationship.

4.1.1. Podporované datové typy⁴

- String (např. jméno procesu)
- Number
- Boolean (např. primární klíč)
- Text (např. pole dokumentace)
- Vector graphic (např. mapa, nebo diagram)
- Collection of items (např. atributy objektu)
- Non-property (např. atribut v seznamu atributů objektu může být sám o sobě objektem).

³ Viz. zdroj č. 3 str. 5

⁴ Viz. zdroj č. 5

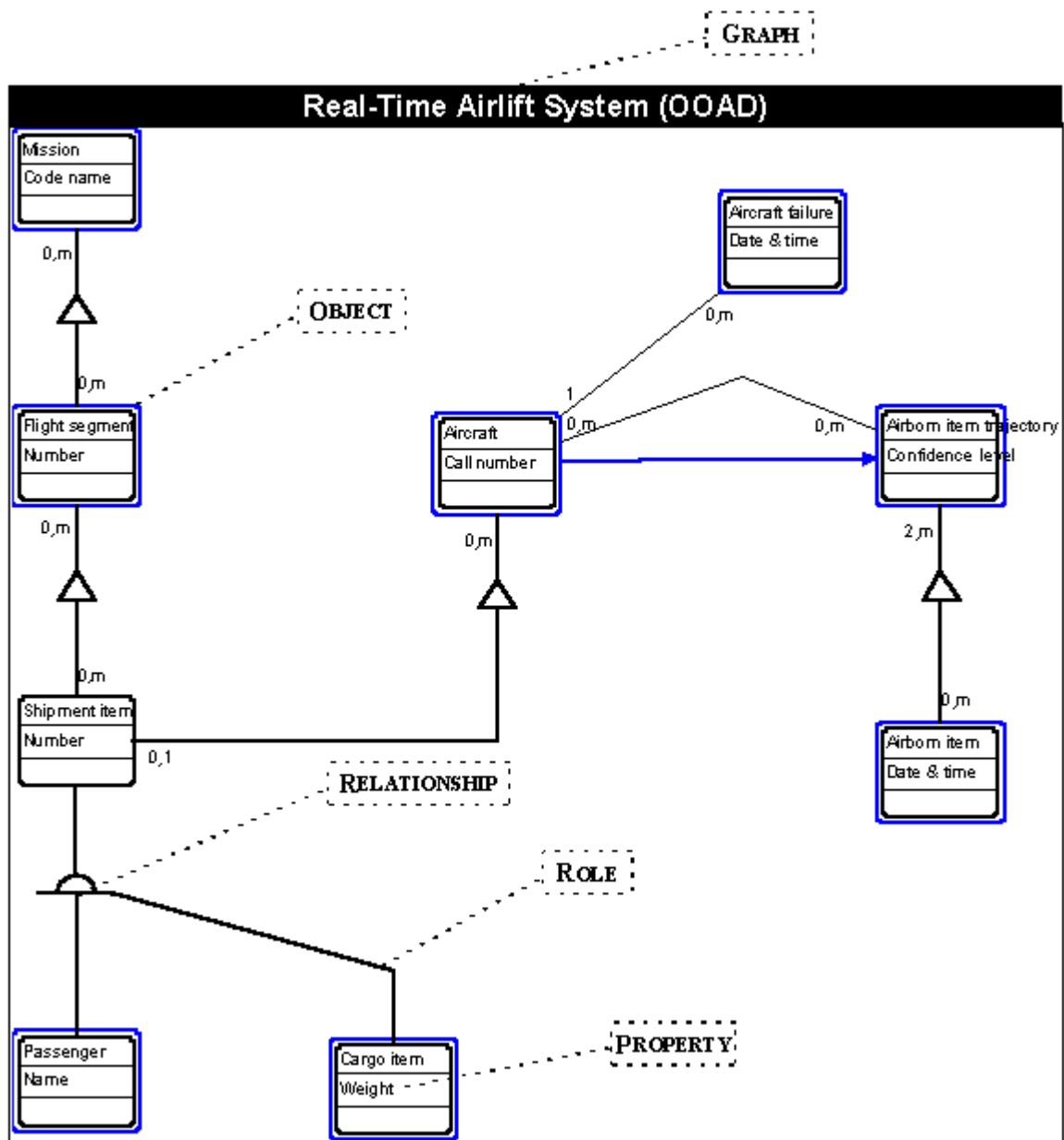
4.1.2. Popis konceptu GOPRR⁵

Níže naleznete krátký popis hlavních modelovacích nástrojů konceptu GOPRR.

- Diagram /graph/
 - o Kolekce objektů, vztahů, rolí, která definuje, co a jak lze spojovat dohromady
- Objekt /object/
 - o Definuje entitu, která může existovat sama o sobě
- Vlastnost /property/
 - o Charakterizuje graf, objekt, roli nebo vztah
- Role /role/
 - o Existuje mezi vztahem a objektem
- Vztah /relationship/
 - o Existuje mezi dvěma a více objekty

⁵ Viz. zdroj č. 2

Názorně předvedeme koncept GOPRR na následující ilustraci:



6

4.2. COMMA

4.2.1. Popis konceptu COMMA⁷

Projekt COMMA (common meta model architecture) popisuje metamodel na bázi pojmů ze světa objektově orientovaných metodologií. COMMA používá tyto základní pojmy:

- Pojem /koncept/
 - o Má jméno a atributy
- Dědění /inheritance/

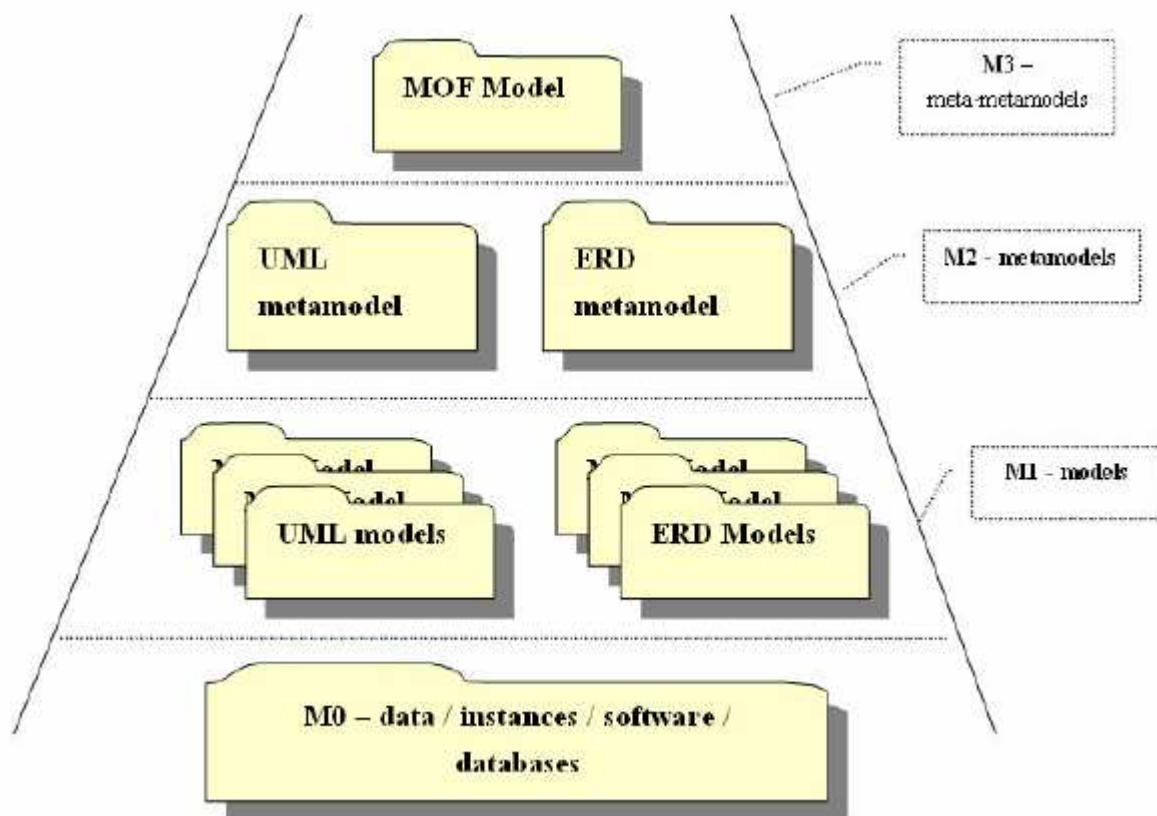
⁶ Viz. zdroj č. 5

⁷ Viz. zdroj č. 2

- Vyjadřuje relaci specializace
- Asociace /association/
 - Vyjadřuje vztah mezi pojmy
- Agregace /aggregation/
 - Vyjadřuje skládání, je to speciální případ asociace
- Role /role/
 - objevuje se, když objekt přijímá charakteristiky jiného objektu. Role je dočasná a objekt může mít i více rolí najednou.

4.3. MOF

MOF (Meta Object Facility) je standardem konsorcia OMG (Object Management Group) pro metamodel . MOF je založen na představě čtyř vrstev, z nichž vrstva vyšší je metavrstvou k vrstvě nižší. Znárodnění vrstvenosti je vidět na obrázku [4]. Jako vrstvu M2 je možné si představit například jazyk UML, jako vrstvu M1 konkrétní model a jako vrstvu M0 vygenerovaný kód z modelu. MOF je dobře popsán v [jaro2007], proto zde zmíníme spíše související standardy jako XMI a JMI.⁸



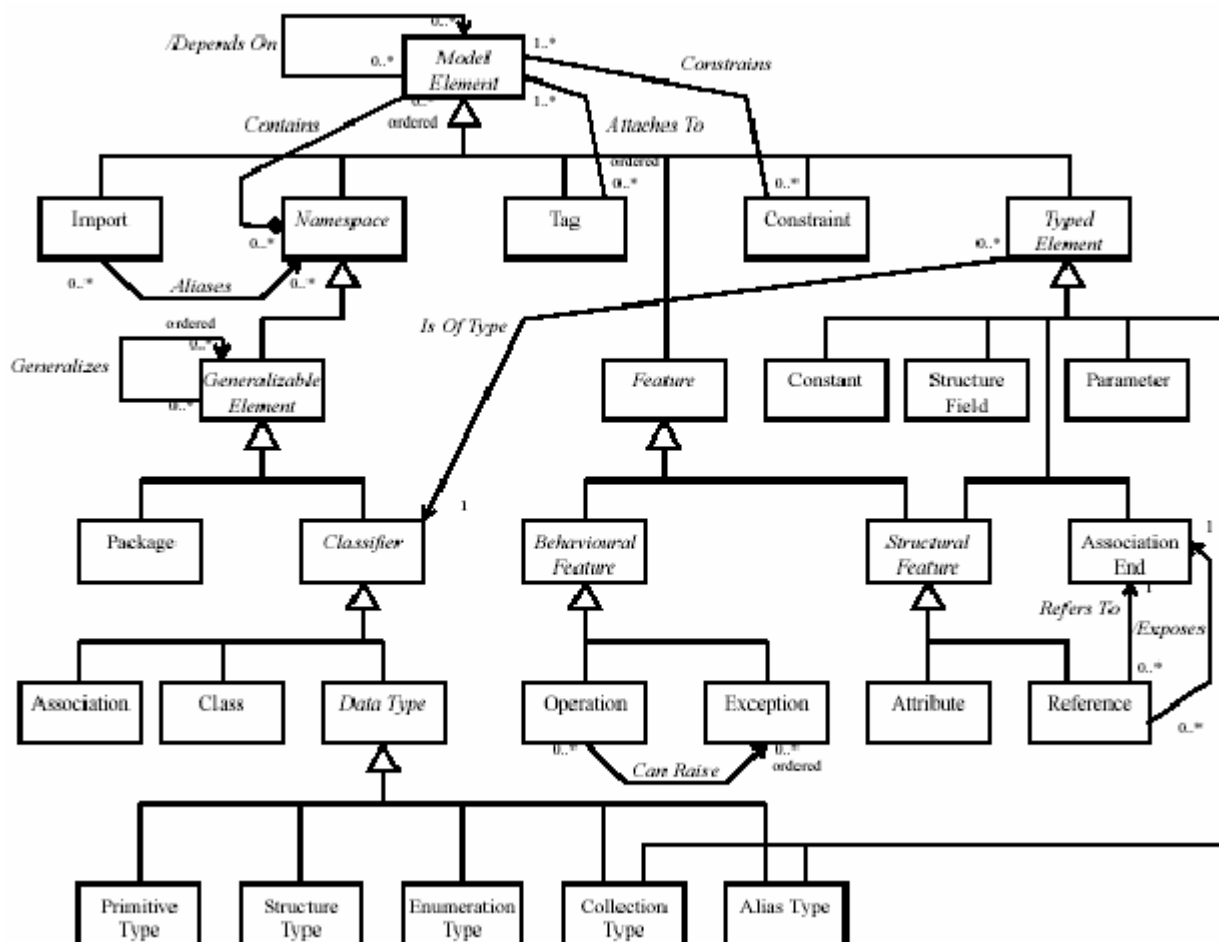
9

Detailní popis konceptu MOF pak najdete na <http://universalis.elibel.tm.fr/pub/metamodels/mof/mof.html#DataType>

⁸ Viz. zdroj č. 2 strana 6

⁹ Viz. zdroj č. 3 strana 10

Na obrázku níže pak naleznete popis MOF modelu v obrázkové formě. Vše, co vidíte na obrázku níže, můžete najít detailně popsané v zdroji č. 6:



10

Pro přenos metadat namodelovaných pomocí MOFu byl vyvinut konsorciem OMG standard XML metadata interchange (XMI). XMI je v podstatě jazyk na bázi XML, který byl vyvinut k popisu modelů na bázi MOFu. Pomocí XMI je tak možné přenést UML model mezi dvěma nástroji, pokud tento standard podporují.

Přestože je velice často zmiňovaný vztah MOFu a UML, tak hlavní náplní MOFu není metamodel pro UML. Největší využití nachází MOF v oblasti datových skladů a modelování metadat. Pokud chtějí mezi sebou komunikovat dvě aplikace využívající jiné datové úložiště, tak typicky využijí webových služeb, které je nutno explicitně vytvořit. Pokud by však obě datová úložiště využívaly pro popis svých dat MOF, byla by možná jejich komunikace čistě na bázi metadat. Takovou to komunikaci usnadňuje framework Java Metadata Interface (JMI). JMI umožňuje z popisu metadat v XMI vygenerovat konkrétní Java třídy s příslušnými metodami a s těmi dále pracovat pomocí jazyka Java.¹¹

¹⁰ Viz. zdroj č. 3 strana 11

¹¹ Viz. zdroj č. 2 strany 6, 7

5. Pro a proti Metamodelování¹²

- + Svoboda volby metody
- + Rychlejší a přehledný vývoj specifických řešení
- + MDA
 - *MDA definuje přístup ke specifikaci IT systémů, který odděluje specifikaci systémové funkcionality od specifikace implementace této funkcionality na specifické technologické platformě.*
 - *... ten samý model specifikující systémové funkcionality (může) být realizován na několika platformách ... kde dovoluje integraci různých aplikací na základě explicitního spojení jejich modelů, povolující integraci a interoperabilitu a podporující vývoj systému jako platformy technologie "come and go".¹³*
 - *Popis metamodelu MDA viz níže:*

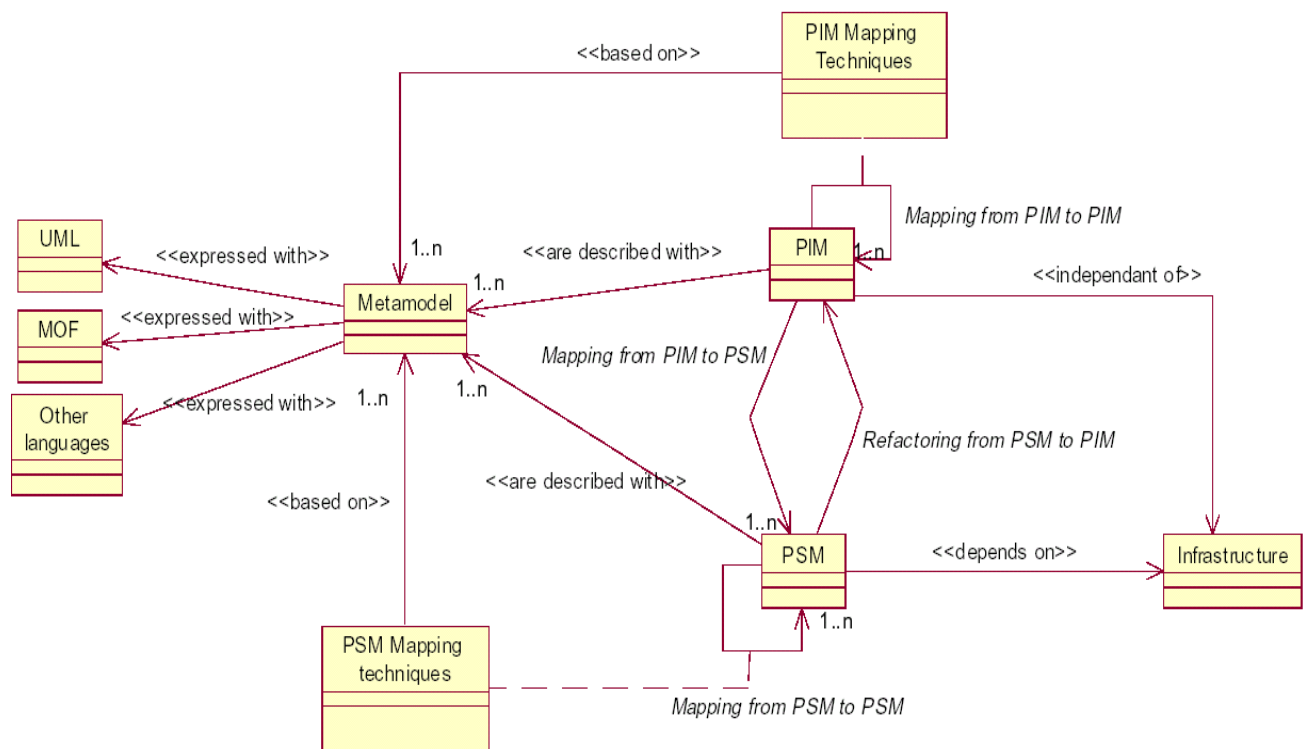


Figure 8. MDA Metamodel Description

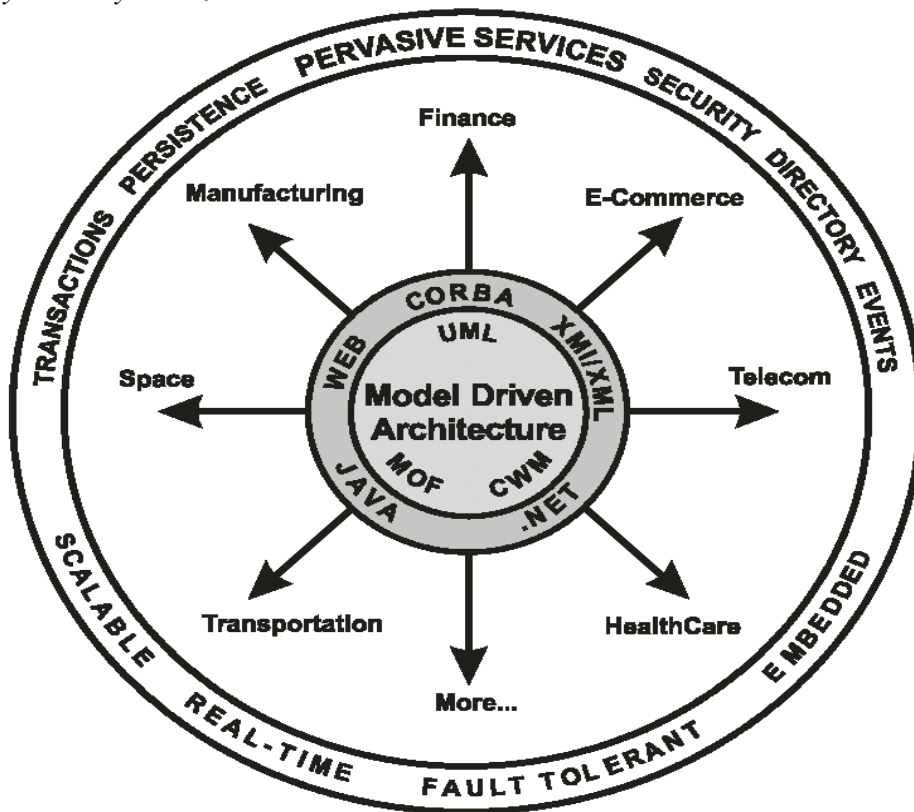
14

¹² Viz. zdroj č. 3 strana 7

¹³ Viz. zdroj č. 4 strana 4

¹⁴ Viz. zdroj č. 4 strana 12

Obrázek níže ukazuje specializované oblasti, ve kterých jsou metamodely MDA využívány a platformy na kterých běží ...



15

- + Formalizace best practises
- Pracnost vytvoření metamodelu
- Nevhodné pro řešení „běžných“ problémů

¹⁵ Viz. zdroj č. 4 strana 17 >>> specialized computing environments

6. Domain specific modeling¹⁶

Jedním z důvodů vzniku metamodelování byla potřeba vytvářet modely pro specifické oblasti, kde jeden metamodel, jako UML, nemusí být plně dostačující. Modelování pro specifické oblasti, jako např. finance, výrobní průmysl a další, se nazývá Domain specific modeling (DMS). Podstatná část DMS je tvořena generováním jazyka (Domain specific language - DSL) z výsledných modelů. Z modelu popisujícího např. výrobní proces je pak možné vygenerovat program zapsaný v jazyce, který se používá k řízení výrobní linky.

Představme si výrobní halu ovládanou pomocí specifického DSL jazyka. Pokud skupina technických expertů vytvoří metamodel a popis transformování modelů do daného jazyka, který algoritmus výroby řídí, bude možné všechny výrobní procesy jednoduše modelovat a automaticky spouštět bez jakéhokoliv ručního zásahu programátorů do kódu ovládacích programů. Tím se maximálně usnadní přechod od modelů přímo k výrobku. Zde je možnost, aby se prosadily metaCASE nástroje, jako například MetaEdit+, který transformace do DSL podporuje. Nicméně tvůrci dnes nejmasověji rozšířeného modelovacího jazyka UML se rozhodli, že umožní použití UML i v oblasti DSM, a zavedli standard UML Profily, kterým se věnuje následující podkapitola.

6.1. UML Profily¹⁷

Profily se definují pomocí značek (stereotype, tag), a omezujících pravidel (constraints), které se aplikují na konkrétní elementy modelu, jako jsou třída, atribut, aktivita atd. Pomocí profilů je možné přizpůsobit modely pro konkrétní oblast (finanční, zdravotní, výrobní) nebo pro konkrétní platformu (J2EE, .NET).

Nejnámějším UML Profilem je Systems Modeling Language SysML (<http://www.omg.sysml.org/>). Pomocí profilů je možné např. barevně označit elementy podle typu jejich stereotypů nebo přidat standardním elementům nové vlastnosti. UML profily podporuje většina kvalitnějších CASE nástrojů. V CASE nástrojích si pak můžete vytvořit vlastní profily nebo importovat existující.

Nicméně UML Profily neposkytují tolik možností rozšíření jazyka jako vytvořený metamodel v metaCASE nástroji. Profily nám neumožní například komplexní generování kódu z modelů nebo implementaci validací. Na druhou stranu, protože je jazyk UML velice známý, nekladou vysokou náročnost na schopnosti a vědomosti uživatele, jako je tomu při tvorbě metamodelů

7. Katalog meta-CASE nástrojů

Tato část má za úkol přiblížit trh s meta-CASE nástroji. Po sledování předchozích prací jsem se rozhodl pro určitý souhrn všech nástrojů v předchozích pracích zmíněných a nabídnout pohled filtrující pouze aktuální a „pravé“ meta-CASE nástroje.

Trendem se aktuálně zdá být vývoj nástrojů pro více platform, většinou tedy v jazyce JAVA. Zároveň je patrný úbytek komerčních nástrojů a stagnace (co do počtu produktů) na poli akademických nástrojů. Zároveň je třeba si povšimnout nulového vývoje za poslední rok celkově u všech produktů. Představím nově v sérii těchto prací nástroj GEMS projektu Eclipse.

¹⁶ Viz. zdroj č. 2 strana 7

¹⁷ Viz. zdroj č. 2 strana 7

7.1. Alfabet, IPSYS Toolbuilder

Tyto nástroje, zmíněné v první práci na téma meta-CASE [zima05], již neexistují a nejsou ani k dispozici jakékoli jejich produktové stránky či možnost jejich stažení. Firmy je původně vyvíjející se dnes evidentně věnují jiným oblastem softwarového vývoje. [zima05]

7.2. CA Erwin Data Modeler

Historickými kořeny produktů společnosti CA (Computer Associates Int.) jako např. CA Erwin Data Modeler a CA Erwin Model Manager jsou nástroje **Paradigm Plus** společnosti Platinum (odkoupenou CA v roce 1999) a **Allfusion Component Modeler**. Dle prezentačních materiálů firmy i dojmů z verze Data Modeler Community Edition se domnívám, že se nejedná o pravý meta-CASE nástroj, neboť jde spíše o modelování nad všeobecným metamodelem podniku a jeho činností. Meta-modelování je spíše doplňkem tam, kde standardní možnosti CA Erwin nestačí. CA Erwin Data Modeler je vhodný především jako součást komplexních řešení od firmy CA. [zima05][zima06][jaro07][jaro08]

URL	http://www.ca.com/us/data-modeling.aspx
Aktualizace	Verze 7.3, leden 2009
Cena	Komerční v řádu tisíců dolarů

7.3. MetaMill, ArgoUML

MetaMill a ArgoUML jsou modelovací nástroje, ovšem bez schopností pro metamodelování. Tudíž se nehodí pro potřeby této práce. [zima05][zima06]

7.4. MetaView

MetaView je ze zástupců metamodelovacích nástrojů vyvíjených na univerzitní půdě, konkrétně na University of Alberta, Kanada. Ovšem jeho poslední aktualizace byla již v roce 1997. Jeho grafické „možnosti“ tomu plně odpovídají a v konkurenci aktuálních programů nemá šanci obstát. [zima06]

URL	http://www.cs.ualberta.ca/~softeng/Metaview/project.html
Aktualizace	Verze 2.6, březen 1997
Cena	Zdarma pro nekomerční účely

7.5. ConceptBase

ConceptBase je více-uživatelský objektový manažer vyvíjený katedrou profesora Dr. Matthias Jarke na univerzitě RWTH Aachen v Německu. ConceptBase ukládá všechny třídy, meta třídy, instance atributy, role apod. jednotně jako objekty. Každý takový objekt může být následně kdykoli editován.

ConceptBase je založen na architektuře klient/server, podporující jazyky C++, Java, Prolog a další. Modelování je v ConceptBase založeno na abstrakci typu OMG, využívá datového modelu O-Telos (<http://www-i5.informatik.rwth-aachen.de/CBdoc/O-Telos.html>). Pro jeho využití hovoří nasazení na více než 500 místech po celém světě, jak pro výzkum tak pro výuku (Georgia State University – USA, North Carolina State University – USA, Universität Karlsruhe – Německo, Universität Hannover - Německo) .

ConceptBase je možné získat na 3měsíční zkušební dobu pro nekomerční a nevojenské využití. Ceny pro komerční využití jsou k získání na základě komunikace s týmem okolo prof. Jarke. [zima05][zima06][jaro07][jaro08]

URL	http://www-i5.informatik.rwth-aachen.de/CBdoc/
Aktualizace	Verze 7.1, květen 2008
Cena	Zdarma pro nekomerční účely

7.6. Coral

Produkt Coral je open-source nástroj vyvíjený v CREST (Centre for Reliable Software Technology, <https://research.it.abo.fi/research/crest>) při univerzitě Åbo Akademi v Turku, Finsko. Jakožto správný metamodelovací nástroj je nezávislý na jakémkoli metamodelu. Na stránkách je prezentován jako plný metamodelovací nástroj, neboli platí, že „modelovací jazyk, obsahující abstraktní a konkrétní syntax, je definován modelem“ a Coralu mimo jiné umožňuje vytvářet i nové modelovací nástroje.

Již na úvodní stránce se prezentuje implementací různých OMG standardů (MOF, UML) do svého rámce a využitím XMI pro výměnu diagramů. Zároveň je Coral prezentován tak, že nemá být náhradou za existující UML nástroje. [zima05][zima06]

URL	http://crestwiki.abo.fi/confluence/display/CRL/Home
Aktualizace	Verze 0.9.6, listopad 2007
Cena	Open-source, výzkumný nástroj zdarma

7.7. OpenSoul Metamodeler

OpenSoul Metamodeler je metamodelovací nástroj vystavěný nad standardem MOF. Byl vyvíjen na VŠE v Praze týmem studentů kolem profesora Řepy. Poslední aktualizace programu se však datuje do listopadu 2005 a další vývoj zřejmě v tuto chvíli neprobíhá. [jaro07][jaro08]

URL	http://metamodeler.sourceforge.net/
Aktualizace	Verze 0.07, listopad 2005
Cena	Open-source

7.8. MetaEdit+

Tomuto programu se již věnovalo mnoho prací dříve. Avšak i nadále se jeví jako komerčně nejúspěšnější provedení meta-CASE nástroje, avšak také prakticky jediné „čistokrevné“. Konkurencí mu snad mohou být pouze DSL tools od Microsoftu. Překvapivě se však zdá, jakoby vývoj poněkud „usnul na vavřínech“ a „aktuální“ verze 4.5 je na světě již přes rok. Pokud autoři minulé práce vyčetli nedostatky tomuto nástroji, domnívám se, že pro firmu MetaCase je mnohem důležitější věnovat se komerčnímu aspektu věci, tj. podpora uživatelů, schopnost spolupráce víceuživatelských týmů či např. API pro provázání se stávajícími nástroji firmy. [zima06][jaro07][jaro08][podzim08]

URL	http://www.metacase.com/
Aktualizace	Verze 4.5, březen 2008
Cena	Komerční v řádu tisíců dolarů (http://www.metacase.com/prices.html)

7.9. Microsoft DSL Tools

Ačkoli se zdá, a spousta dřívějších prací se vyjadřuje v tomto smyslu, že trh s meta-CASE nástroji je relativně malý a komerčně nepříliš zajímavý, pustil se do něj i gigant Microsoft. A rozhodně neočekávám, že by zde očekával vysoké podíly na trhu, zisky apod. Spíše se zdá, že Microsoft zaceluje mezeru ve svých vlastních produktech, kterou ve firmách musely suplovat produkty jiných firem spojené s MS různými překladači apod.

Produkt je dobře popsán v předcházející práci. S autory pak nesouhlasím v jejich názoru o pouhých výhodách použití Visual Studia jako prostředí pro doplněk DSL Tools. Jistě je tomu tak v případech, kdy je podnik založen celkově na produktech společnosti Microsoft. Ovšem je třeba brát v potaz dnešní vzrůstající oblibu open-source nástrojů, nasazování Linuxu místo Windows apod.

URL	http://www.microsoft.com/downloads/details.aspx?FamilyId=57A14CC6-C084-48DD-B401-1845013BF834&displaylang=en
-----	---

Aktualizace	Verze nov2005, prosinec 2005
Cena	Zdarma, nutno zakoupit Visual Studio Professional

7.10. GME

Projekt GME byl vyvíjen na univerzitě Vanderbilt, Nashville, USA. Poslední verzí je GME 7.6.29 z roku 2007. Jednalo se o propracovaný nástroj, v jehož pozadí lze nalézt profesora Schmidta, jakožto senior výzkumného vědce na institutu ISIS (Institute for Software Integrated Systems). Na první pohled se mi zdálo, že je škoda, že již tento projekt nepokračuje (přesněji řečeno, nejsou vidět nové verze). Avšak v zápětí jsem objevil prof. Schmidta mezi vývojáři projektu **GEMS** (viz dále) a to včetně dalších vývojářů z ISISu. Tudíž předpokládám, že lze GEMS považovat za volného nástupce GME, a že zkušenosti z vývoje GME se projeví právě u GEMS. [jaro07][jaro08]

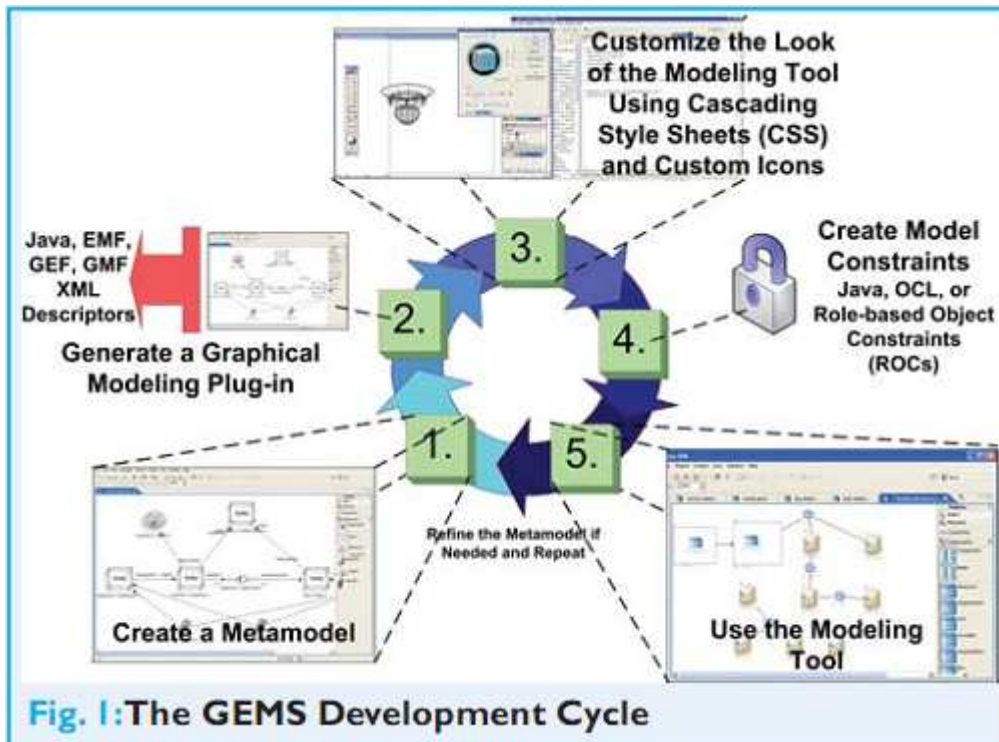
URL	http://www.isis.vanderbilt.edu/projects/gme/
Aktualizace	Verze 7.6.29, 2007
Cena	?opensource

7.11. GEMS

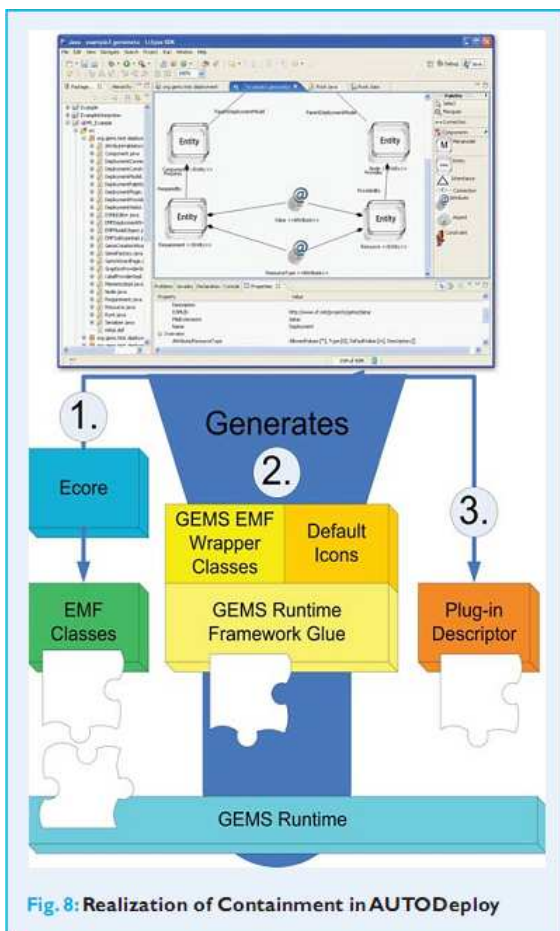
Práce z minulého semestru se zabývala nástrojem Eclipse GMF. Ovšem pokud se bavíme o nástrojích vhodných pro metamodelování, nabízí se od projektu Eclipse přímo nástroj GEMS (Generic Eclipse Modeling System). GEMS, tak jak již u výtvorů projektu Eclipse bývá zvykem, obsahuje mnoho nadstandardních funkcí a funkcionalit, které by jsme spíše očekávali u komerčních produktů (jako např. vzdálené aktualizace a dotazování se, tvorbu šablon, úprava vzhledů pomocí CSS či provázání modelů). Na projektu se účastní takové společnosti jako IBM, Siemens či PrismTech, a samozřejmě profesor Schmidt, autor GME.

GEMS se především snaží propojit možnosti stávajících nástrojů z portfolia Eclipse. Obrázek 1 zachycuje vývojový cyklus v GEMS. Myšlenka je taková, že v prostředí GEMS je nejprve vytvořen metamodel pro DSML (popisující entity, typy propojení, atributy...), který je použit pro vygenerování kódu EMF (Eclipse Modeling Framework), GMF (Graphical Modeling Tool), XML a jiných. Ty jsou využity pro tvorbu pluginu Eclipse, který umožní editaci DSML popsaného metamodelem. Třetí krok pak znázorňuje možnost změny vzhledu modelovacího nástroje. Ve čtvrtém kroku jsou vyjádřena omezení modelu, zajišťující, že GEMS bude produkovat pouze správné modely. Pátý krok je již praktickým využíváním modelovacího nástroje na specifické doméně. Takový nástroj může zahrnovat i generátory kódu.

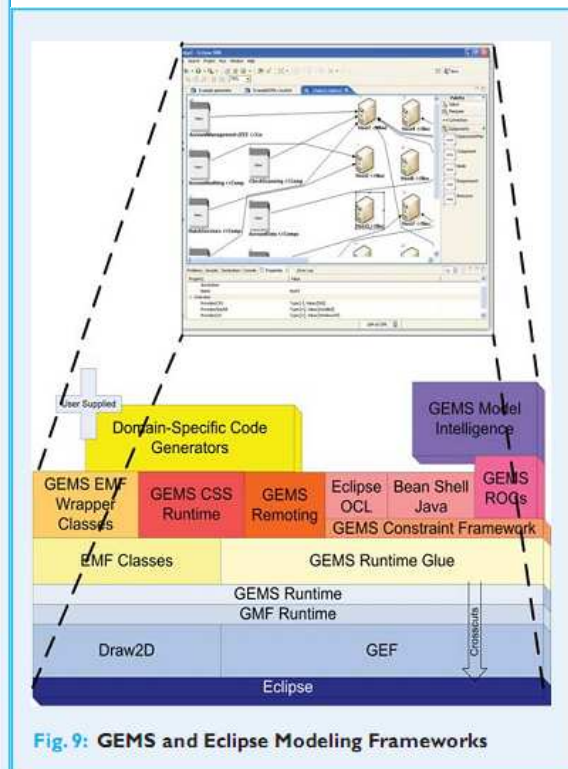
Obrázek 2 pak detailněji rozebírá druhý krok, tvorbu pluginu pro grafický modelovací nástroj nad vytvořeným metamodelem. A poslední obrázek vyjadřuje vztahy mezi vytvořenými částmi dle metamodelu, GEMS prostředím a EMF.



Obrázek 1 Vývojový cyklus GEMS



Obrázek 2 Druhý krok detailněji



Obrázek 3 Vztahy mezi metamodelem, prostředím GEMS a EMF

Ačkoli je GEMS (a poměrně často i jiné Eclipse projekty) méně intuitivní a občas evidentně zatížen tím, že je postaven právě na platformě Eclipsu, přijde mi nejzajímavější ze všech nástrojů. Bohužel i jeho poslední aktualizace je téměř rok stará.

URL	http://www.eclipse.org/gmt/gems/
Aktualizace	Verze 3.0RC4, červen 2008
Cena	Licence Eclipse

8. Závěr

Trh s meta-CASE nástroji je poměrně malý, což neumožňuje příliš investovat do jejich vývoje a tomu odpovídá i tendence klesajícího počtu komerčních produktů. Naopak je evidentní, že se jedná spíše o akademické projekty, či projekty open-source (či podobně licencované). V zásadě u naprosté většiny lze objevit akademické kořeny. Pro budoucí práce doporučuji sledovat projekt GEMS, sledovat zda Microsoft bude pokračovat ve svém snažení na tomto trhu a zda firma MetaCase a její produkt MetaEdit+ stále existuje a vyvíjí se. Všeobecná otázka zní, zda vůbec je v tuto chvíli nějaký projekt aktivní. Také předávám doporučení z předchozí práce, hledat ve vyhledávačích spíše pod pojmem „domain-specific-language“ a „domain-specific-modeling“ než pouze dotazem na „meta-CASE“.

*V rámci předmětu 4IT450 byla vytvořena, mimo jiných, práce na téma **CASE nástroje pro jazyk UML** (autoři bc. Vomáčka, bc. Klusák, bc. Skružný a bc. Šetek). V rámci této práce byl představen nástroj s podporou pro DSL, který v této práci chybí, **Borland Together**. (<http://www.borland.com/us/products/together/index.html>)*

Tato práce je ke stažení na stánkách s pracemi, které vznikly v rámci semináře CASE za období jaro 2009 <http://www.panrepa.org/CASE/>

9. Zdroje

1. Wikipedie: Otevřená encyklopedie: Unified Modeling Language [online]. c2009 [citováno 24. 03. 2009]. Dostupné z WWW: < http://cs.wikipedia.org/w/index.php?title=Unified_Modeling_Language&oldid=3605882 >
2. BLAHA Jan, HRABAL Jan: CASE (Computer Aided System Engineering): Zimní semestr 2008/2009
3. ZENTNER Albert: Metamodelování - OpenSoul Metamodeler UML profily v EA [online]. C2005 [citováno 24. 03. 2009]. Dostupné z WWW: < <http://nb.vse.cz/~repa/CASE/metamodeling.ppt> >
4. BORDELEAU Francis: MDA Overview: [online] c.2002 [citováno 24. 03. 2009] .Dostupné z WWW: < www.scs.carleton.ca/~francis/Courses/543/Slides/MDA-Overview.ppt >
5. KELLY Steven: GOPRR Description: [online] c. 1997 [citováno 24. 03. 2009]. Dostupné z WWW: < <http://metaphor.it.jyu.fi/a1goprr.html> >
6. Universalis Publisher: Metamodel MOF: [online] [citováno 24. 03. 2009]. Dostupné z WWW: < <http://universalis.elibel.tm.fr/pub/metamodels/mof/mof.html> >
7. CA Computer Associates Int.: Data Modeling from CA [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www.ca.com/us/data-modeling.aspx> >
8. The Metaview Project [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www.cs.ualberta.ca/~softeng/Metaview/project.html> >
9. ConceptBase – A deductive object manager for meta databases [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www-i5.informatik.rwth-aachen.de/CBdoc/> >
10. Home – Coral – MDE [online][přístup 28.4.2009]. Dostupné z WWW: < <http://crestwiki.abo.fi/confluence/display/CRL/Home> >
11. OpenSoul Metamodeler[online][přístup 28.4.2009]. Dostupné z WWW: < <http://metamodeler.sourceforge.net/> >
12. MetaCase - Domain-Specific Modeling with MetaEdit+ [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www.metacase.com> >
13. Download details: Microsoft Tools for Domain-Specific Languages CTP (November 2005 CTP Release for Visual Studio 2005 RTM) [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www.microsoft.com/downloads/details.aspx?FamilyId=57A14CC6-C084-48DD-B401-1845013BF834&displaylang=en> >
14. GME: Generic Modeling Environment | Institute for Software Integrated Systems [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www.isis.vanderbilt.edu/projects/gme/> >
15. GEMS Home page [online][přístup 28.4.2009]. Dostupné z WWW: < <http://www.eclipse.org/gmt/gems/> >
16. Eclipse Magazine vol. 6 [online][přístup 28.4.2009]. Dostupné z WWW: < http://www.cs.wustl.edu/~schmidt/PDF/Eclipse_Magazine_Volume06.pdf >
17. BALOGH István a kol.: Nástroje meta-CASE, zimní semestr 2005. Dostupné z WWW: < http://panrepa.org/CASE/meta_CASE.pdf >
18. APFELTHALER Jan a kol.: Nástroje meta-CASE, zimní semestr 2006. Dostupné z WWW: < http://panrepa.org/CASE/zima2006/meta_case_zima06.pdf >
19. BURIAN Petr, GOTTWALD Tomáš, PŘIKRYL Jan: Nástroje meta-CASE, letní semestr 2007. Dostupné z WWW < http://panrepa.org/CASE/jaro2007/meta_case_jaro2007.pdf >
20. KUBAL Pavel, VÁVRA Ondřej, FISCHER Roman: Nástroje meta-CASE, letní semestr 2008. Dostupné z WWW: < http://panrepa.org/CASE/jaro2008/meta_case_jaro2008.pdf >
21. BLAHA Jan, HRABAL Jan: Nástroje meta-CASE, zimní semestr 2008. Dostupné z WWW: < http://panrepa.org/CASE/podzim2008/meta_case_podzim2008.pdf >