



Metamodelování

a problematika jeho nasazení a použití

ve firemní praxi

Petr Tománek

Petra Hradecká

1. Abstrakt

V této práci bychom rádi objasnili pojem metamodelování a jeho odlišení od běžného náhledu na modelování. Dále bychom rádi odůvodnili přínos jeho nasazení ve firmě i s riziky, kterých je nutné si být vědomi. Tato práce by měla komplexně zodpovědět otázku vhodnosti použití metamodelování a nastítnit základní podporované vlastnosti, které je nutné mít na zřeteli při vytváření metamodelu pro firemní využití.

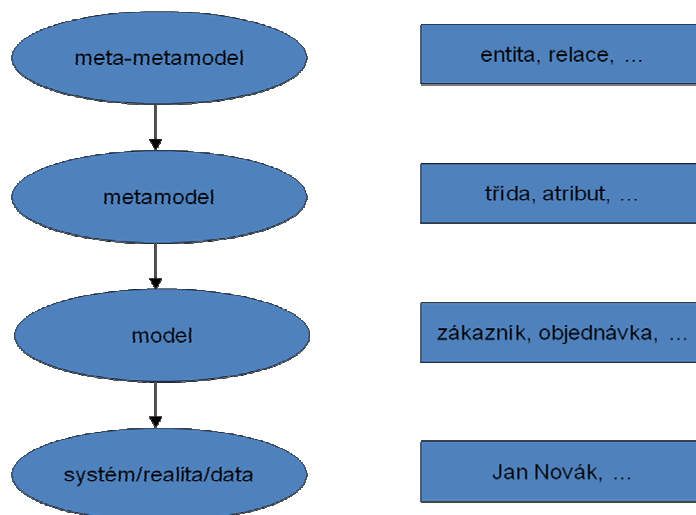
2. Co je metamodelování

Metamodelování je možné chápat jako nadstavbu modelování s výraznými prvky abstrakce, stejně jako modelování chápeme jako abstrakci a simplifikaci skutečnosti. Co je tedy ta forma abstrakce, která odlišuje metamodel od běžného modelu? Podívejme se nejprve, jak bychom měli chápat samotný význam slova metamodelování. Prefix meta vychází z řeckého výrazu, znamenající „nadto“, „poté“ a dnes jej používáme v přeneseném významu jako akronym pro opisné informace, vztahující se k zadanému tématu a popisující jej odpovídajícím výrazivem. Význam prefixu „meta“ můžeme chápat jako oproštění se od zkoumané problematiky a změnu úhlu pohledu směrem ke globální rovině pohledu a k upřednostnění přístupu, získaném pohledem shora na daný problém, vedoucím k jistému stupni generalizace údajů.

V informatice se prefix „meta“ používá jako výraz zobecnění, což lze vyjádřit na příkladu, kdy samotný metajazyk je jazykem, který je schopný popsat jazyky; metadata jsou data, popisující data a tedy metamodel je modelem, popisujícím běžné modely. Důvodem, který vedl ke vzniku metamodelování, je potřeba popsat odlišné datové struktury, které jsou používány jako komponenty běžně navrhovaných modelů.

Pokud jsme v minulosti byli schopni simplifikovat realitu do určujícího modelu a právě jsme definovali další úroveň abstrakce nad modelem, nazvanou metamodel, lze se oprávněně domnívat, že platí jakási forma indukční logiky a i na metamodel lze pohlížet jako na strukturu, jejíž komponenty lze zobecnit a tím vytvořit jakousi vyšší úroveň náhledu. Tato úroveň je definovaná jako meta-metamodel již Bézivinem¹ (viz. Obrázek 1). Vycházíme-li z logických zásad matematické indukce, lze kroky zobecnění neomezeně opakovat, v reálné problematice se však používá maximálně čtvrtá úroveň abstrakce, jelikož s každou další úrovní již ztrácíme adekvátnost činnosti a problém se již stává natolik abstraktním, že ztrácí vypovídací schopnost pro návrh systému a dostáváme se tímto již spíše do filosofické roviny.

¹Bézivin J. (1998): Who's Afraid of Ontologies?, Proceedings of OOPSLA'98 Workshop No.25 – CDIF, Vancouver, ziskatelné z <http://www.metamodel.com/oopsla98-cdif-workshop/bezivin1/>



Obrázek 1- Model, metamodel, meta-metamodel, Bézivin 1998

Problematikou použití prefixu „meta“, jako významu „o úroveň vyšší“, se zabývá např. Hřebejk, Řepa, 1999² Proto se v následujícím textu budeme zabývat pouze popisem metamodelování s několika málo odkazy na vrstvu meta-metamodelu.

Jak již bylo řečeno, metamodel vychází z popisu modelu jako systému, používaného pro vývoj aplikací, informačních systémů a simulací, např. podnikových procesů. Samotný model je tedy účelovou abstrakcí reality, při které se v maximální míře vynechávají údaje, nepotřebné pro vypovídající vlastnost modelu a současně je nutné dostatečným způsobem zachytit stávající stav skutečnosti.

Dochází k hledání společných vlastností u objektů reálného světa, jejich slučování dle potřeb, hledání kauzalit a vazeb mezi subjekty. Díky tomu se nad vrstvou reality vytváří vrstva modelů, ve které jsou prvky různé třídy a jejich vztahy, popsané metadaty (metadata pak určují model a strukturu reálných dat). Od vzniku modelování vykrytalizovalo několik rodin modelů obsahující různé entity dle potřeb modelů. Stejně tak vznikly poměrně robustní nástroje, obsahující pomůcky pro správu těchto různých modelů. Tato cesta vedla postupem času k tomu, že tyto nástroje se svojí robustností ztratily flexibilitu a schopnost přizpůsobení se změnám, tudíž uživatelé byli omezeni pevně definovanými vlastnostmi jednotlivých modelů, bez možnosti jejich rozšíření.

Tato potřeba vedla k dalšímu použití abstrakce a k přechodu k metamodelovému náhledu; ke vzniku meta-metadat, jimiž by se dosáhlo definování a customizace jednotlivých modelů. Metamodelování tedy vedlo ke tvorbě nových metodologií pro tvorby návrhů IS, pomocí generických modelů. Důsledek použití

²Hřebejk P., Řepa V., Meta-modelování. Příspěvek na konferenci DataSem 99, získatelné z <http://nb.vse.cz/~repa/veda/RaD.htm>

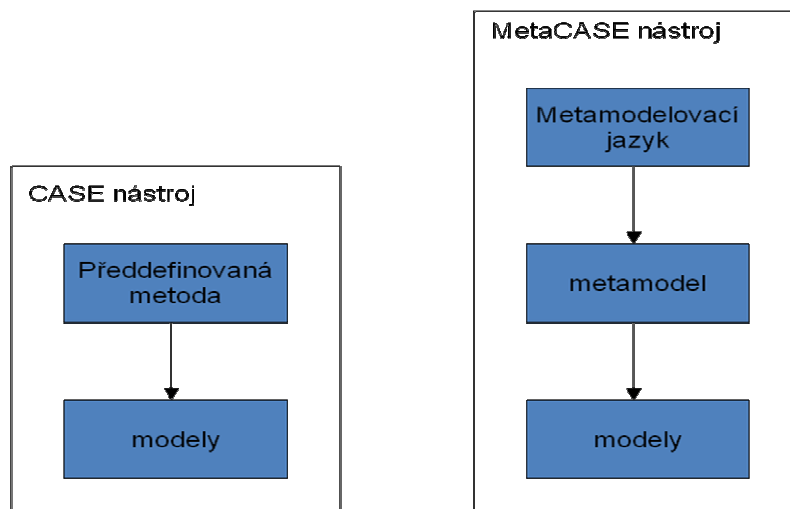
metamodelování je nezávislost na technologiích, tedy metamodely se často používají jako nadstavba a sjednotitelé různých modelačních technologií a modelačních jazyků – čímž dochází k překlenutí mezer mezi jednotlivými návrhovými systémy.

Stejně jako modely, jsou i metamodely značně rozdílné v závislosti na úhlu pohledů na problematiku a potřebách jeho developerů a implementátorů. Kritický dopad na kvalitu metamodelu má znalost a zkušenost osob s ním pracujících.

Metamodely se běžné používají jako:

- schéma pro sémantická data, která je potřeba uložit nebo předat mezi subjekty,
- jazyk podporující konkrétní schéma nebo proces,
- jazyk vyjadřující doplňující sémantiku přidanou k již existujícím informacím.

Jak jsme tedy uvedli, metamodel je používán jako nástroj pro definici modelu, čili jeho použití je primární při vývoji nástrojů zabývajících se modelováním – CASE nástrojů. Nástroje, využívající metamodelovací přístup, rozšiřují svou funkcionalitu o schopnost úpravy jednotlivých metod používaných při definici metadat, tvořících model. Oproti běžným CASE nástrojům, jejichž funkcionalita postihuje dvě nejnižší vrstvy modelu, zobrazeném na Obrázku 1, jsou tyto nástroje rozšířeny na vrstvy tři (čtyři pro pohled z úrovně meta-metamodelu), jak je patrné z Obrázku 2.



Obrázek 2- Srovnání CASE a MetaCASE

Tedy, podle doposud uváděné metodologie, je metaCASE odlišný pohled na metamodelování pomocí CASE nástrojů s určitou úrovní abstrakce. Podle ISO³ jedním z jazyků popisujících metaodely je standard Meta Object Facility (MOF), definovaný skupinou Object Management Group (OMG) a nejnověji uvedeným v

³http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32622

normě ISO 19502:2005. MOF jako jazyk specifikuje, vytváří a upravuje práci s technologií neurálních metamodelů, zajišťuje konzistenci mezi jednotlivými sadami modelů, implementuje design a použití repositoří aplikačních vývojových nástrojů atd. Repository se tedy stává úložištěm metadat (resp. jimi popsaných modelů), která za pomoci pravidel definovaných obdobnými metadaty hlídá svou integritu, resp. integritu modelů popsaných metadaty v ní obsažené. MOF je objektově orientovaný jazyk, který popisuje sám sebe, díky čemuž je vysoce modifikovatelný a rozšiřitelný. Současně definuje komunikační rozhraní, používaná pro správu a operace s modely, jejichž rozhraní dosud nebyla použita; k tomu používá Interface Definition Language (IDL) CORBA.

Typickými metamodely (metamodelovacími jazyky) navrženými OMG jsou UML, Sysel, SPEM nebo CWM.

3. Požití metamodelování v reálném prostředí firmy

Používání metamodelu ve firmě přináší sjednocení přístupu k modelování, čímž omezuje náklady na úpravy vznikající na základě chybných návrhů datových základů v modelech. Je nutné si ale uvědomit, že tato výhoda je vyvážená nutností být schopen operovat v mezích pokročilých znalostí použitých pro definici vývojového prostředí pro firemní zaměstnance, kteří díky němu mohou vyvíjet návrhy modelů. Splnění tohoto požadavku vynakládá extrémní nároky na personální zázemí u definice metamodelu, případně na finanční výdaje spojené s outsourcingem jeho vývoje a s tím se vyskytující možnost případného bezpečnostního rizika. Stále je ale nutné vést v patrnosti, že dokonalý metamodel nám nezaručuje vytváření správných modelů na jeho základě. Naopak ale, špatný metamodel vždy povede minimálně k omezení tvorby modelů, spíše pak povede ke špatně vygenerovaným modelům, jejichž použití zapříčiní výskyt chyb v modelech vzniklých během návrhu aplikací a špatnou funkcionalitu těchto aplikací.

Kvalitu vytvářeného metamodelu můžeme při jeho tvorbě a nasazování posuzovat z několika vzájemně propojených hledisek:

Dostatečný rozsah metamodelu – stanovit rozsah metamodelu je relativně obtížný úkol. Rozsah metamodelu musí být dostatečný na to, aby byl schopen pojmut veškerá data z modelů v něm definovaných, resp. aby tato data byly schopny pojmut jím definované modely. Současně je nutné, abychom s využitím zkoumaného metamodelu byli schopni definovat kompletní zamýšlené modely. Metrikou požadované kvality tohoto metamodelu pak budiž relativní/absolutní množství námi zamýšlených konceptů, které je schopen zkoumaný metamodel pojmut vůči počtu konceptů, které by pojmut měl. Předmětem zkoumání není pouze počet takto definovatelných modelů, ale i to, zda bylo nutné v modelech data nějak restrukturalizovat, zda jsou použitelná dle požadavků, je možné je uložit všechna, atp.

Jednou z možností, jak ověřit dostatečný rozsah metamodelu, je například aplikace postupu používaného ve statistice, kdy si vyměříme jistý, dostatečně velký, soubor dat a na něm provedeme testy naší hypotézy/metamodelu. Samozřejmě je nutné, aby naše data, kterými budeme metamodel testovat, obsahovala všechny nám zajímavé koncepty, jaké používáme, nebo budeme používat. Dle množství a kvality vybraných dat lze na určité hladině významnosti tvrdit, že metamodel je vytvořený správně.

Další, poněkud méně exaktní, zato však v praxi používanější, metoda, kterou lze použít v tomto i v následujících hlediscích, je po vypracování metamodelu jej předat v písemné formě někomu nezúčastněnému (ale současně na dostatečně odborné výši), např. kolegovi a požádat jej, aby nám z předaných materiálů vysvětlil, jak náš návrh chápe a co z něj on sám dedukuje, tedy zda došlo k přenosu sémantiky v zápisu definic metamodelu.

Technická kvalita – tak jako jiné uměle vytvořené artefakty, jsou i metamodely nejlépe analyzovatelné pod zátěží. Proto se doporučuje podobné testování, jako například testování počítačových programů, modelů, apod., kdy se do artefaktu zadávají extrémní nebo mezní hodnoty a zkoumá se jeho chování, zejména pak chybovost, s jakou je schopen tyto hodnoty zpracovat. V dalším pohledu je možné otestovat udržitelnost a soudržnost aplikací v metamodelu tím, že testování provedeme metodou aplikace imaginárních požadavků na úpravu modelu. Při té zohledňujeme kvalitu zpracování úpravy, schopnost zpracovat nestandardní vstupy a požadavky a testujeme další externality, které náš metamodel mohou potkat.

Rozšiřitelnost – toto kritérium bylo již rámcově zmíněno v předchozích kritériích, ale vzhledem k jeho důležitosti jsme se rozhodli jej zmínit v samostatném bloku. Nutnost podpory rozšiřitelnosti u modelu je neoddiskutovatelná, pokud je žádoucí, aby model pokrýval změny, které vyplynou z nových nároků v budoucnosti. Vzhledem k důležitosti postavení metamodelu nad modely, se samozřejmě adekvátně zvyšuje důležitost rozšiřitelnosti metamodelu, aby byl schopen jednoduše reflektovat budoucí zamýšlené změny – dopad případného nevhodného návrhu metamodelu bude ve formě přepracování tohoto metamodelu také důsledkem nutnosti přepracování všech modelů tímto metamodelem definovaných. Proto bychom si měli klást otázky, zda je zkoumaný metamodel schopen pojmut libovolný logický doplněk bez poškození nebo omezení své funkcionality a zpětně i modely sebou definované. Je vysoce pravděpodobné, že zkoumaný metamodel nebude plně rozšiřitelný ve všech směrech, které si lze představit, proto je omezit zkoumání možnosti jeho rozšíření na nejpravděpodobnější oblasti, ve kterých lze očekávat jeho budoucí úpravy a změny.

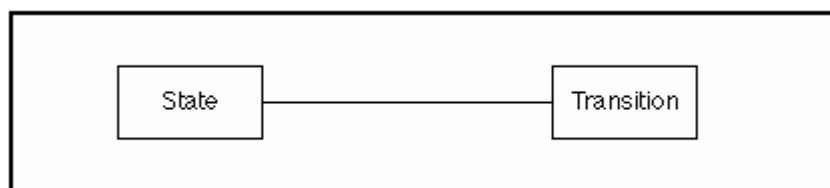
V tento okamžik nám vyvstává další problematická vlastnost, kterou námi zkoumaný metamodel musí obsahovat, a tou je kompatibilita komponent vytvořených ve stávajícím neupraveném metamodelu s nově rozšířeným metamodelem a vice versa. Vzhledem k tomu, že metamodel se vytváří spolu s modely, které má obsahovat, tedy v počáteční analytické fázi programování (modelování) je téměř jisté, že spolu s vývojem modelů bude nutné také metamodel obohatit o vlastnosti, které v původní verzi nebyly zamýšleny. Dalším důvodem vzniku potřeby úpravy metamodelu je fakt, že metamodel často bývá používán jako referenční struktura pro vyměňovaná data při styku mezi obchodními partnery. Pro rozšíření definice dat, se kterými firmy mezi sebou komunikují, je nutné, aby obě (všechny) strany komunikace přijaly upravený metamodel a ten zakomponovaly do svých datových struktur.

Se změnami a objevivším se verzováním modelu se také poprvé dostáváme k nutnosti vytvořit systém změnového managementu. S množstvím různých upravovatelů se zvyšuje pravděpodobnost kolize jednotlivých úprav a nutnost jejich kooperativní práce na jednom metamodelu bez ztráty změn vytvořených

různými zdroji během aktualizace metamodelu v jednom úložišti a současně jeho distribuci v současném stavu mezi všechny zdroje / upravovatele.

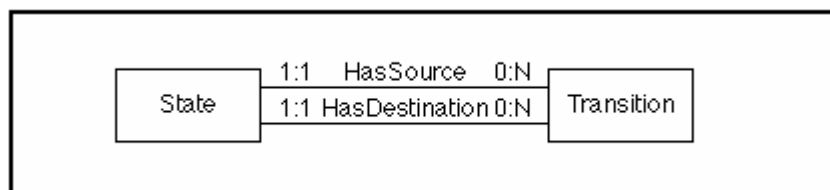
Kvalita definic v dokumentaci – u metamodelování je kvalita dokumentace primárním cílem a výsledkem vývoje metamodelu. Na kvalitní dokumentaci závisí použitelnost metamodelu v praxi, tj. správnost modelů, které jsme díky tomuto metamodelu vyvinuli a z globálního pohledu je dokumentace završením a zaevidováním veškeré práce, kterou jsme na metamodelu při jeho vývoji zanechali; jedná se o evidenci jednoznačně definovatelných vlastností objektů vyskytujících se v metamodelu - tedy ať už jde o entity nebo jejich vztahy, jejich vlastnosti, typy, prvky apod. Dá se říci, že bez dokumentace nedrží vývojář v ruce nic jiného, než graficky zobrazený shluk entit a vztahů mezi nimi, které lze vyložit různými způsoby. Dokumentace k metamodelu tedy je výsledným produktem metamodelování. Proto by měla být dokumentace vyhotovená do takové hloubky, aby znemožnila dvojí výklad libovolné části metamodelu a také dostatečně kvalitně popsala a definovala metamodel jako celek.

Pro příklad použijme volnou citaci textu uvedeného v jednom ze zdrojů⁴, kde je ilustrován příklad ze života s výsledky výstupu metamodelování:



Evidentně má tento fragment metamodelu za úkol vyjádřit, že existují stavy a jejich změny a že mezi nimi existuje relace. Bohužel toto grafické vyjádření je absolutně nedostačující pro dostatečnou dokumentaci metamodelu. Mezi jinými nám neodpovídá na otázky kardinality mezi stavy a změnami, jejich omezení, neříká nám nic o vlastnostech jednotlivých stavů a jejich změn, zda mají vůbec názvy apod. Kvalitní dokumentace (= kvalitní metamodel) dává odpověď na všechny tyto otázky a i na některé další. Zkusme se nyní podívat na stejný fragment metamodelu po rozpracování:

⁴ <http://www.metamodel.com/staticpages/index.php?page=20021010225607569>



Stav	příklad	
Název	String	povinný: ano
JeVýchozí	Boolean	povinný: ne
JeCílový	Boolean	povinný: ne

Změna stavu	příklad	
Trvání	Time	povinný: ne
SpouštěcíUdálost	String	povinný: ano
ProvedenáAkce	String	povinný: ne

Na první pohled je vidět dramatické zlepšení zdokumentování metamodelu, bohužel ale v momentě, kdy začneme tento metamodel používat, můžeme narazit na mnoho nejasností a nesrovnalostí. Příkladem může být problém v pojmenování struktur a prvků všeobecně. Vývojář, který výše uvedený metamodel vytvářel, pojmenoval struktury podle svého očekávání vlastností, které by měly mít. To ale nezaručuje, že jiný uživatel metamodelu bude mít stejné zkušenosti a očekávání, aby automaticky převzal všechny zamýšlené, ale neuvedené vlastnosti objektu v metamodelu, dokonce se může stát (a zhusta se tak stává), že na základě svých zkušeností může předpokládat vlastnosti právě opačné.

Příkladem budiž objekt Stav. Odkud víme, že to, co tvůrce pojmenoval Stav je to, co si my (potažmo každý jednotlivý uživatel) myslíme, že Stav je? Zde vždy vývojáři pomáhá abstrahovat název konceptu za obdobu abstraktního anglického „Foo“ (obecné pojmenování libovolného objektu) a zjistit, zda je schopen i po přejmenování objektu stále odvodit pouze na základě informací uvedených v metamodelu, co koncept prezentuje.

Takže věci, které lze vydedukovat stavu a jeho změnách, je to, že byly popsány v diagramu a tabulce uvedených výše. Ale co jsou? Jaký je jazyk, kterým popisujeme stavy a jejich změny? To je důvod, proč se metamodelování ve firmě používá.

Jak je z předchozího textu patrné, zpracování metamodelu je při jeho nasazování do běžného používání ve formě věc, od které odvisí celá úspěšnost implementace řešení na metamodelovací bázi. Samotná potřeba změny programového vybavení na to, které podporuje metamodelové prostředí, se jeví spíše podružná, jelikož v zásadě se pro běžné uživatele s prakticky minoritní změnou ovládacích prvků a techniky práce nezmění zprvu prakticky nic. S postupem času, kdy dojde ke změnám v metamodelech a tím i umožnění úpravy pracovních modelů se situace postupně změní, ale stále jde spíše o postupné rozšiřování možností, než o skokovou změnu odvislou od nasazení metamodelování.

Bavíme-li se ale v intencích zaměstnance zabývajícího se přípravou modelů na meta- úrovni, který dostane nyní do ruky nový nástroj pro provádění takovýchto úprav, dochází k markantnímu rozšíření jeho pracovních možností. Ale jak jsme již vzpomněli, tato výhoda je vykoupena nutností zavést ve firmě nové procesní postupy pro práci s metamodely, upravit pravomoce, upravit podmínky komunikace s obchodními partnery apod.

Nasazení metamodelovacích technik v reálné firmě není tedy pouze o změně náhledu na práci, ale nese s sebou značnou míru nejistoty díky nutnosti úprav mnoha aspektů firemní funkcionality, s čímž jsou často spojena významná rizika. Navíc změny nejsou omezeny pouze na firmu samotnou, ale dotýkají se i jejích partnerů, se kterými komunikuje. Projekt nasazení metamodellingu je pak nutné brát jako inicializaci projektů provádějících širší změny i mimo samotný objekt firmy.

4. Závěr

V předchozím textu jsme si vymezili pojem metamodelování a metaCASE a uvedli jsme základní podmínky pro úspěšné nasazení metaCASE ve firmě.

Jak je vidět, metamodelování je velmi obtížná práce, která rozšíří možnosti zpracování modelů ve firmě, na druhou stranu vyžaduje precizní zvládnutí problematiky. Pokud bychom měli provést shrnutí, pak primárním účelem metamodelování je potřeba automaticky generovat velké množství modelů nezbytných pro specifická odvětví. Využití metaCASE a jeho nasazení ve firmě je přínosem pro případy, kdy:

- použití metaCASE vede ke snížení času a nákladů na vývoj CASE
- metaCASE umožňuje implementaci vlastních metod, používaných během vývoje
- meta CASE umožňuje srovnání různých vývojových prostředí
- meta CASE může být použit jako modelovací nástroj v IS.

5. Zdroje

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32621

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32622

<http://www.metamodel.com/staticpages/index.php?page=20021010231056977>

<http://www.metamodel.com/staticpages/index.php?page=20021010225607569>

<http://en.wikipedia.org/wiki/Metamodeling>

<http://objekty.pef.czu.cz/Objekty98/documents/kozel.pdf>

http://panrepa.org/CASE/zima2006/meta_case_zima06.pdf

http://panrepa.org/CASE/meta_CASE.pdf

Wokoun, Graphical Representation in the Meta-Modeling Task, diplomová práce, VŠE 2006