METAMODELOVÁNÍ V PRAXI

METAMODELLING AND CREATING IS

Marek Pícka[1]

Abstrakt
Tento článek pojednává o základních pojmech metamodelování, vysvětluje čtyřvrstvou architekturu pro metamodelování. Dále se věnuje metamodelovacím jazykům COMMA, GOPRR a MOF. V poslední části popisuje použití metamodelování zejména při tvorbě metodologií,v meta-CASE nástrojích, v repositářích a při modelování s využitím generických modelů.

Abstract:
This paper analysis the aplication of metamodelling in the practise. Fists of all there are describing of esential terms of metamodelling and four layer metamodel architecture. In the next part, there are mentions about the most widely used metamodels (GOPRR, metamodel UML a MOF). And in the last part there is a mention about the aplication of metamodelling in the practise ie metamodel as a grammar of methodologies, generical model etc.

Klíčová slova
Model, metamodel, meta-metamodel, metamodelování, metodologie, meta-CASE, MOF, generický model

Key words:
metamodel meta-metamodel methodology generic model

Úvod

V poslední době se objevilo nové „moderní“ slovo, nebo spíše předpona - *meta*. Provádíme operace nad metadaty. Metamodelujeme a tím vytváříme metamodel v meta-CASE nástroji, který je řízen metamodelem. Z metadat generujeme programy. Dokonce můžeme i metaprogramovat v metaprogramovacím jazyce.

Abychom pochopili význam slova je dobré se nejdříve podívat do encyklopedie (třeba [8]) a tam zjistíme, že meta- pochází z řečtiny a mimo jiné znamená za, po, vně, mimo. Další význam z hlediska informačních technologií najedeme zde [7] a říká nám, že meta- znamená o jeden stupeň abstrakce výše tj metadata jsou data popisující data, metamodel je model popisující model.

Architektura pro metamodelování

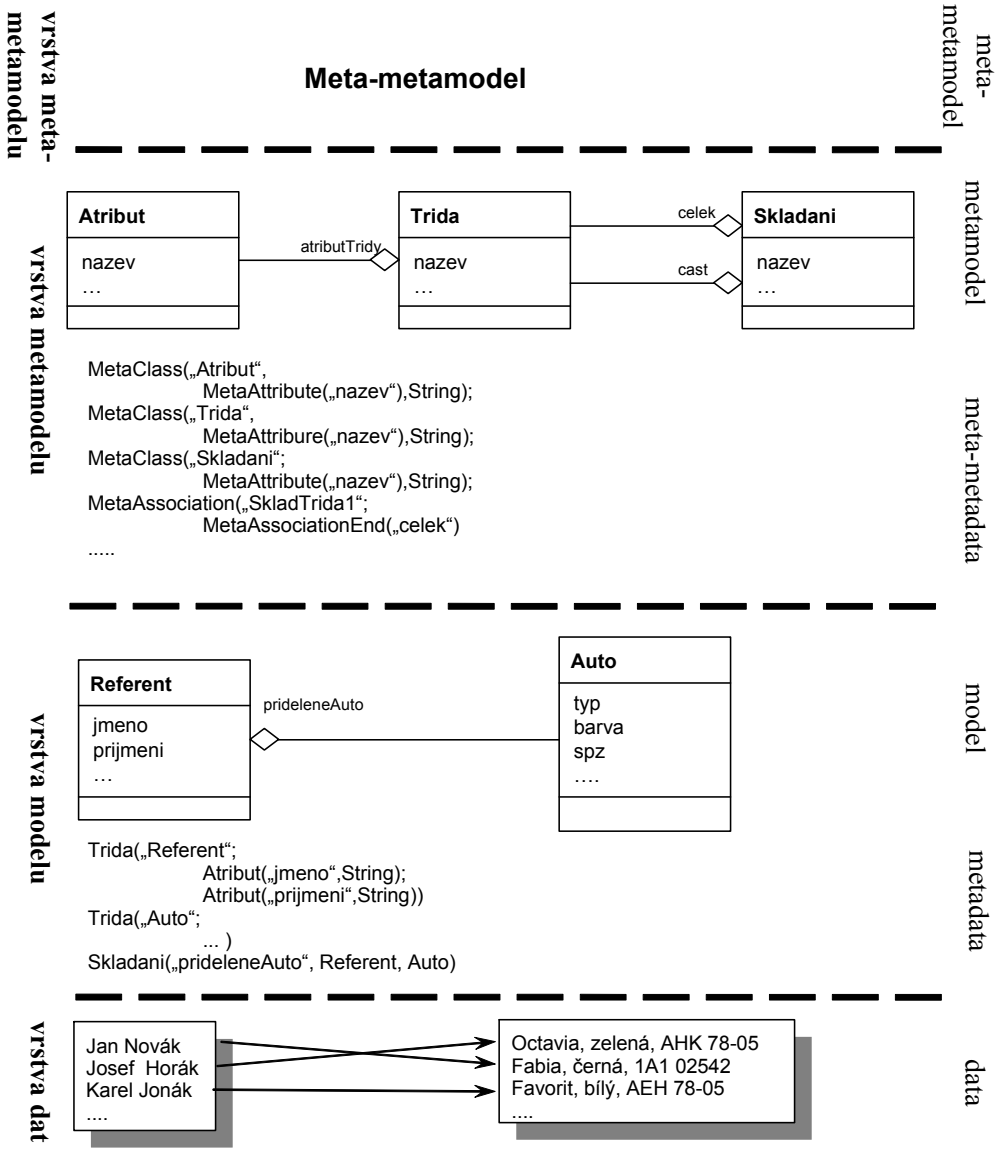
Klasický rámec pro metamodelování je podle [1] založen na čtyřvrstvé architektuře. Tyto vrstvy jsou obvykle popisovány takto:

- *Informační (datová) vrstva* zahrnuje popisovaná data.
- *Vrstva modelu* zahrnuje metadata popisující data v informační vrstvě. Metadata jsou neformálně shrnuta do modelů.

[1] Ing. Marek Pícka, Katedra informačního inženýrství, PEF ČZU Praha, picka@pef.czu.cz

- *Vrstva metamodelu* zahrnuje popis (tj. meta-metadata) definující strukturu a sémantiku metadat. Z meta-metadat jsou neformálně tvořeny metamodely. Metamodel je abstraktním jazykem pro popis různých druhů dat.
- *Vrstva meta-metamodelu* definuje strukturu a sémantiku meta-metadat (tj. popisuje metamodel).

Na obrázku č. 1 je ukázán klasický čtyřvrstvý rámec pro metamodelování. Příklad ukazuje metadata a metamodel pro jednoduchý příklad skládání referenta s automobilem. Další vrstva (metamodelu) obsahuje meta-metada a popisuje co je to *Trida, Skladani, Atribut* a jejich vztahy pomocí termínů z vrstvy meta-metamodelu (*MetaClass, MetaAssociation* aj.). Vrstva meta-metamodelu definuje metamodelovací konstrukce (*MetaClass* atd.).



Obrázek 1 – čtyřvrstvá architektura pro metamodelování

I když tento příklad ukazuje pouze jedem model a jeden metamodel, hlavní výhodou použití čtyř meta- vrstev je podpora mnoha modelů a metamodelů. Tak jako třída *Auto* popisuje mnoho instancí automobilů na datové vrstvě, tak *Trida, Atribut* a *Skladani* na vrstvě metamodelu popisují mnoho tříd, atributů a asociací na vrstvě modelu. To samé platí ve vztahu meta-metamodelu a metamodelu.

Hlavní výhody této architektury, pokud je správně navržena, jsou:

- Podporuje všechny modely a modelovací paradigmata,
- Dává do souvislosti různé typy metadat,
- Dovoluje inkrementální přidávání nových typů metadat,
- Podporuje výměnu metadat (modelů) a meta-metadat (metamodelů) mezi stranami, které používají stejný meta-metamodel.

Metamodelovací prostředky

Metamodelovací metody definují rámec pro metamodelování, který obvykle obsahuje definici meta-metamodelu a metamodelovacího jazyku, pomocí kterého je definován metamodel. Pro potřeby metamodelování v informačním a softwarovém inženýrství bylo vyvinuto mnoho přístupů pro tvorbu metamodelu - COMMA, GOPRR, MOF, OPRR, CoCoA, NIAM, COOM atd.. V dalších odstavcích jsou popsány tři metamodelovací rámce, které jsou vhodné pro definici metodik.

COMMA

Projekt COMMA (Common Object Methodology Architecture) se snažil identifikovat společné jádro všech objektově orientovaných metodologií, následně reprezentovat tyto základní pojmy pomocí metamodelu a vytvořit metamodely nejrozšířenějších objektově orientovaných metodologií (více viz [3]).
COMMA používá tyto základní pojmy:

- Pojem (Concept) – má jméno a atributy,
- Dědění (Inheritance) – vyjadřuje relaci specializace,
- Asociace (Association) – vyjadřuje vztah mezi pojmy,
- Agregace (Aggregation) – vyjadřuje sklání, 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.

Hlavním výsledkem projektu COMMA je vytvoření velni jednoduchého (ale mocného) objektově orientovaného metamodelovacího jazyka. Nevýhodou je, že tento projekt již skončil a neexistuje napojení na CASE nástroje.

GOPRR

Metamodelovací jazyk GOPRR (Graph-Object-Property-Role-Relationship) vznikl, jako součást disertační práce pana Kellyho (viz [5]), rozšířením jazyka OPRR. Hlavním úkolem bylo na základě GOPRRu vytvořit CAME (Computer Aided Method Engineering) nástroj MetaEdit+.
Základními prvky metamodelovacího jazyka GOPRR, už podle názvu, jsou:

- Diagram (Graph) – je kolekce objektů, vztahů a rolí, která definuje co a jak lze spojovat dohromady.
- Objekt (Object) – definuje co entitu která může existovat samo o sobě.
- Vlastnost (Property) – charakterizuje graf, objekt, roli nebo vztah.
- Vztah (Relationship) – existuje mezi dvěma a více objekty.
- Role – existuje mezi vztahem a objektem.

Více o MetaEditu+ na [6].

Rodina standardů OMG

Standardy OMG (Object Management Group) jsou založeny na čtyřvrstvé architektuře popsané v odstavci 0. Vrstvu meta-metamodelu popisuje standard MOF (Meta Object Facility – viz [1]) . OMG definuje několik metamodelů založených na MOF:

- metamodel UML (Unified Modeling Language) – standard pro objektový modelovací jazyk (více [2]),
- metamodel IDL (Interface Definition Language) – standard popisující objektová rozhraní tříd pro standard distribuovaných objektů CORBA, a jejich mapování do různých programovacích jazyků.
- metamodel CDW (Common Data Warehouse) – standard definující architekturu datových skladů.

Data mezi metamodely založenými na MOF mohou být vyměňována pomocí formátu XMI (XML Metadata Interchange).

MOF je samovysvětlující, tj. definuje sám sebe (a tedy neexistuje meta-meta-metamodel). Základními koncepty MOF jsou:

- třídy (Class)– modelují metaobjekty,
- asociace (Assotiation)– modelují binární relace mezi metaobjekty,
- datové typy (data Type)– modelují primitivní data,
- balíček (Package) – slouží k modularizaci modelu.

Metamodel UML úzce vychází z MOF a liší se jenom v drobnostech (např. umožňuje vícenásobné asociace). Například třída v UML je definována jako instance třídy „Class“ v UML metamodelu. Tato třída je definována jako instance třídy „Class“ z MOF modelu (meta-metamodelu). A nakonec třída „Class“ z MOF modelu je definována sama sebou.

Použití metamodelování v praxi

Metamodelování se využívá zejména pro popisu a tvorbě nových metodologií, při implementaci metodologií v CASE nástrojích, pro strukturování repositářů, při integraci systémů, při generování programů z modelů, generování reportů, a kontrole modelů. Znalost metamodelu lze také použít pro flexibilní návrh informačního systému pomocí generických modelů.

Využití metamodelu při popisu metodologií

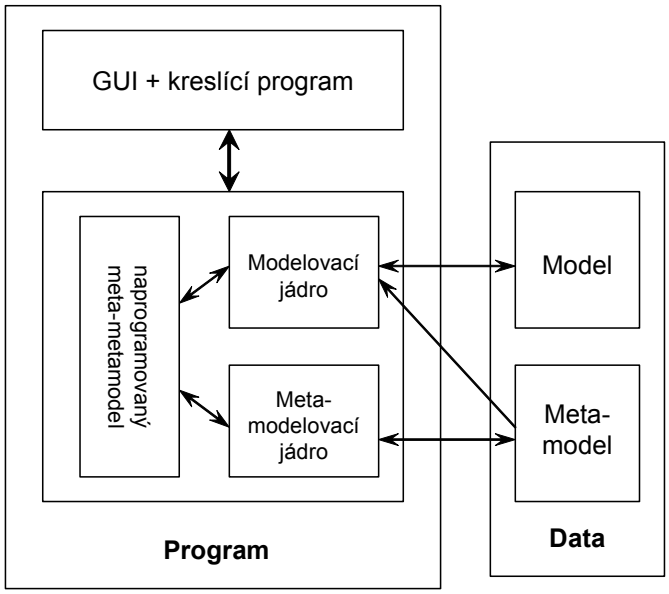
Pomocí metamodelu definujeme metodologie. Každá metodologie je definována pomocí svého metamodelu. Ten může být formálně nebo neformálně popsány. Neformální popisy metamodelů metodologií jsou např. uvedeny v učebnicích metodologií. Například se v každé učebnici UML dočteme, že třída se značí obdélníkem se třeni poli, v kterých jsou údaje jako jméno třídy, atributy třídy a její metody. Dále se dočteme, že asociace je vztah mezi třídami a jak se zakresluje plnou čarou. Pro hlubší pochopení a případnou počítačovou podporu modelování však potřebujeme formalizovaný popis metamodelu (třeba viz [2]).

Metamodelování je základem vývoje nových metodologií (ME – Method Engineering). Poskytuje také prostředky k nalezení společných vlastností různých metodologií, jejich standardizaci a případnou možnost jejich vzájemné kooperace.

CASE a Meta-CASE nástroje

Jedny z prvních aplikací metamodelu byly CASE nástroje. Metodologie jsou definovány pomocí metamodelu a CASE nástroje jej musí implementovat. Mezi další použití metamodelu u CASE nástrojů patří formáty na výměnu dat mezi CASE nástroji – starší CDIF a na XML založený XMI.

V pozdější době se objevily tzv. Meta-CASE (neboli také CAME- Computer Aided Method Engineering) nástroje (například MetaEdit+ viz [6]), které umožňují si definovat vlastní metamodel a tím definovat vlastní metdodologii.



Obrázek 2 –schéma Meta-CASE nástroje

Na obrázku č. 2 je schéma Meta-CASE nástroje. Hlavní části jsou:

- Kreslící jádro + GUI – stará se o vlastní kreslení a interakci s uživatelem.
- Naprogramovaný meta-metamodel – je to interpret jazyka meta-metamodelu.
- Modelovací jádro – je programováno pomocí metamodelu, vytváří vlastní model.
- Metamodelovací jádro – pomocí něho programujeme, v jazyce definovaným meta-metamodelem, metamodel. Toto jádro je hlavním rozdílem oproti klasickým CASE nástrojům, umožňuje nám programovat vlastní metodiky.
- Model – data modelu, typicky ukládaná do databáze.
- Metamodel – data meta-metamodelu, typicky ukládaná do databáze.

Meta-CASE nástroje někdy nazýváme metamodelem řízené programy, protože změnou metamodelu je můžeme přeprogramovat.

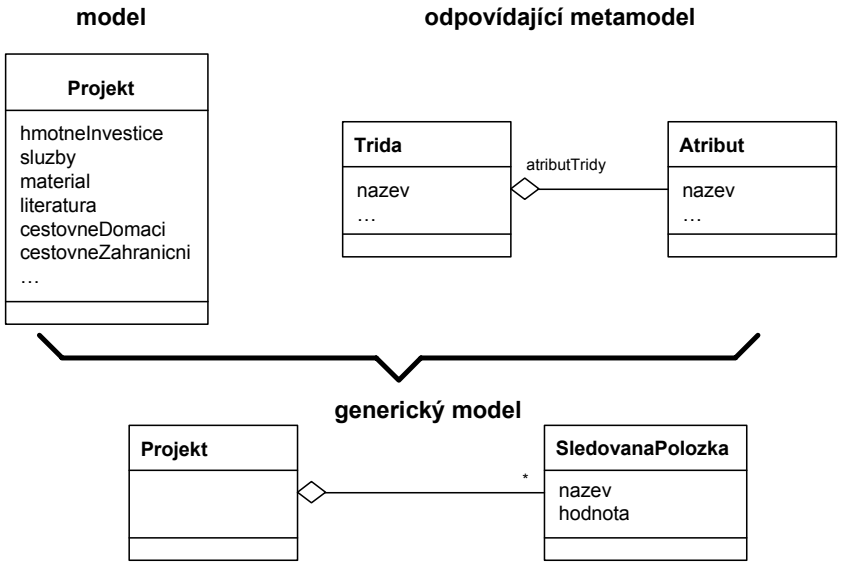
Metamodel při zpracování dat a metadat

Znalostí metamodelu dat umožňuje přesně pochopit strukturu dat a následně s nimi pracovat. Typický příkladem jsou různé repositáře metadat. Jedno z prvních použití metamodelu se týkalo repositářových standardů PCTE a ISO standard IRDS (Information Resource Dictionary Standard). Pokud máme metadata uložena v repozitáři (databázi) je možno tyto data snadno dále zpracovávat. Typickým příkladem takového zpravování je zjišťování konzistence, měření metrik, generování reportů a programů.

Metamodelování dovoluje popsat jednotným způsobem odlišné datové struktury více systémů, což poskytuje možnost je skládat dohromady ve vyšší systém, který dokáže pracovat s daty systémů, ze kterých je sestaven a tím pomoci k jejich integraci. A naopak metamodel poskytuje dobrou abstrakci systému a pomáhá nám k dobrému rozdělení systému na podsystémy a tím pomáhá zvládat komplexní projekty.

Generický model

Pro optimalizaci návrhu projektu je možno použít generický model (viz [4]). To je takový model, který v sobě kombinuje vlastnosti modelu a metamodelu. Více na obrázku č. 3.



Obrázek 3 - generický model

Na obrázku je třída `Projekt`, která je součástí „standardního“ návrhu informačního systému. Instance této třídy mají mnoho atributů (sluzby, material, literatura, cestovne atd.) týkajících se vyúčtování peněz v projektu. Tento návrh je jistě funkční, ale problém nastane pokud původně navržené atributy přestanou pokrývat naše potřeby (např. potřebujeme evidovat zvlášť domácí a zahraniční cestovné). Další nevýhodou tohoto návrhu je, že obvykle spousta atributů bude nulová (tj. atributy nám mohou chybět a zároveň přebývat). Elegantním řešením je vytvoření třídy `SledovanaPolozka` a její skládání (1:n) s třídou Projekt. Tím jsme dostali model, který má i vlastnosti metamodelu. Struktura metamodelu tedy slouží jako příklad jak postupovat při modelování.

Předností generických modelů je v tom, že vedou ke konstrukci snadno modifikovatelného software, a to dokonce i takovým způsobem, který nebyl očekáván při analýze.

Závěr

V současné době se metmodelování používá pro stále větší počet úloh. Mezi jeho nejdůležitější úkoly patří zajištění interoperability metod, podpora vytváření nových metodologií (ME – Method Engineering) pomocí metaCASE nástrojů, integrace systémů a dat a generování zdrojových textů z modelu. Pro tyto úkoly má mnoho metamodelovacích prostředků, z nichž zejména standardy organizace OMG (tj. metamodel UML, MOF) jsou velmi perspektivní.

Literatura

[1] *Meta Object Facility (MOF) Specification,* verze 1.4, v dostání na http://www.omg.org

[2] *OMG Unified Modeling Language Specification,* verze 1.4, v dostání na http://www.omg.org

[3] Henderson-Sellers, B.; Bulthuis, A.: *Object-Oriented Metamethods*, Springer-Verlag New York, 1998, ISBN 0-387-98257-4

[4] Polák, J.; Merunka, V.; Carda, A.: *Umění systémového návrhu*, Grada, Praha, 2003, ISBN 80-847-0424-2

[5] Kelly, S.: *GOPRR Metamodel*, apendix of doctor thesis „*Towards a Comprehensive MetaCASE and CAME Environment: Conceptual, Architectural, Functional and Usability Advances in Metaedit+*“, Ph.D. Thesis, Jyväskylä University, 1997

[6] Web firmy MetaCase: http://www.metacase.com

[7] http://www.metamodel.com

[8] *Malá československá encyklopedie*, 4. díl, Academia, Praha 1996