

MODELOVANIE PRIESTOROVÝCH DÁT V MODEL DRIVEN DEVELOPMENT

Branislav, DEVEČKA¹, Ivan, MUDROŇ¹, Josef, STROMSKÝ², Martin, KRČMARIK¹

¹ Institut geoinformatiky, Hornicko-geologická fakulta, VŠB-TU Ostrava,
17.listopadu 15/2172, 708 33, Ostrava Poruba, Česká republika

Branislav.devecka.st@vsb.cz

² Elvac Solutions s.r.o., Hasičská 53, 700 30, Ostrava, Czech Republic
josef.stromsky@elvac.eu

Abstrakt

Model Driven Development (MDD) predstavuje relatívne nový smer pre vývoj softvérových systémov. Tento spôsob vývoja softvéru sa vyznačuje tým, že obracia modely do podstaty a zameriava sa na vývoj softvérového systému nezávisle na možných platformách, ktoré budú použité. Táto zmena zamerania, od kódovo orientovaného po modelovo orientovaný vývoj, predstavuje významný posun v rozvoji, ktorý využíva rast abstrakcie. Pre podporu tohto rozvoja softvéru prijala skupina Object Management Group (OMG) prístup Model Driven Architecture (MDA). Podľa Arlowa (2005) architektúra MDA definuje predstavu vývoja softvéru na základe modelu. MDA je podporovaná tromi kľúčovými modelovacími technológiami OMG, ktoré sú založené na UML: MOF (Meta-Object Facility) – slúžia ako repozitáre modelu, CWM (Common Warehouse Metamodel) – štandard, ktorý popisuje, ako reprezentovať databázové modely (schémata), OLAP a datamining modely a XMI (XML Metadata Interchange) – mapovanie, pomocou ktorého môžeme vyjadriť UML modely v XML.

V mojom príspevku sa venujem hlavne možnostiam využívania MDD pre modelovanie priestorových dát a GIS. V príspevku sa snažím poukázať na problémy ktoré, súvisia s používaním priestorových dát v MDD metodológií.

Abstract

Model Driven Development (MDD) is a relatively new direction for the development of software systems. This method of software development is characterized by the fact that models turn to nature and focus on on the development of software systems regardless of possible platforms that will be used. This change of focus from code-oriented to model-oriented development, represents a significant shift in development that uses abstraction growth. To support the software development team adopted the Object Management Group (OMG) Model Driven Architecture (MDA) approach. According Arlowa (2005) MDA architecture defines the idea of software development based on the model. MDA is supported by three keys OMG modeling technologies, which are based on UML: MOF (Meta-Object Facility) - serve as a model repositories, CWM (Common Warehouse Metamodel) - a standard that describes how to represent database models (diagrams), and OLAP data mining models and XMI (XML Metadata Interchange) - mapping, which can express UML models in XML.

In my contribution I deal mainly with the options of using MDD for modeling the spatial data and GIS. The contribution is trying to highlight the problems that are related to the use of data in MDD methodologies..

Klíčová slova: UML, Vývoj Riadený Modelom, Model Driven Architecture

Keywords: UML, Model Driven Development, Model Driven Architecture

UVOD

Činnosť vývoja softvéru prešla za posledné roky rýchlym vývojom. Stále viac vyžaduje používanie štandardizovaných metód a techník, ktoré sú všeobecne známe. Dnešný svet začal mať potrebu rýchleho vývoja softvéru. V súčasnej dobe je hlavným záujmom softvérových architektov dobre porozumieť problémovým oblastiam a tak vytvárať riešenia, ktoré vyhovujú skutočným potrebám jednotlivých užívateľov.

Pre porozumenie tohto problému a zníženie zložitosti navrhovaného systému sa používa technika modelovania. Model je zjednodušenie reality (Erikson 2004). Preto technologická spoločnosť postupne prechádza na automatizovaný spôsob vývoja softvéru. Začali vznikať prvé nástroje a vývojové prostredia, ktoré umožňujú generovať zložité a plnohodnotné aplikácie na základe dobre definovaných modelov.

Metodika, ktorá priniesla v tejto oblasti revolučný prístup sa nazýva Model Driven Architecture (OMG, 2003). Tento prístup umožňuje vývoj systémov využívajúcich modely v rôznych úrovniach abstrakcie. Pre vytváranie modelov využíva Unified Modeling Language (UML). Základné princípy tejto metodiky sú vysvetlené v ďalších kapitolách.

Súčasný vývoj GIS smeruje od používania proprietárnych dátových štruktúr pre ukladanie priestorových dát k plnému využitiu databázového prostredia, t.j. ukladaniu a zatiaľ aspoň čiastočnému spracovaniu v relačných či objektovo-relačných alebo objektových DBMS. S tým súvisí otázka dátového modelovania objektov s geometrickou zložkou, aké modely používať aké koncepty využívať ako ich vyvíjať a využívať.

Motiváciou pre riešenie tejto otázky je fakt, že metóda MDD/MDA sa v súčasnosti v hojnej miere využíva pre tvorbu jak webových tak desktopových aplikácií slúžiacich pre správu rôznych oblastí, ako napríklad dispečerské systémy, systémy pre správu podnikov a obchodných činností, systémy verejnej správy a pod. Všetky tieto systémy viac či menej pracujú z geografickými informáciami. K tomu obvykle využívajú externé GIS softvéry, ktoré vyžadujú ďalšie náklady a často sú zbytočne zložité pre daný účel. Preto je mojou snahou podporiť ekonomickejšie riešenia, ktoré by podporovali tvorbu geografických aplikácií pomocou metodiky MDA/MDD.

KONCEPTUÁLNE MODELOVANIE PRIESTOROVÝCH DÁT

Tvorba databázy obvykle začína procesom analýzy. Účelom analýzy je pochopiť a popísať potreby užívateľov v rámci priestorovej databázy (Bédard, 1999). Konceptuálne schémy sú považované za vhodný prostriedok pre vyjadrenie výsledkov procesu analýzy. Poskytujú vyjadrenie reálnych javov z hľadiska užívateľského vnímania. Usporiadávajú ich do tried, vlastností, vzťahov a operácií (Brodeur, 2004).

Modelovanie geografických databáz má niekoľko zvláštností, ktoré spôsobujú vývoj špecifických riešení pre túto oblasť. Zoznam požiadaviek je prezentuje Lisboa Filho a lochpe (1999). Tato štúdia uvádza osem skupín požiadaviek.

Friis-Christensen et al. (2001) opisuje prehľad podmienok pre modelovanie geografických dát, ktoré rozdeľuje do piatich skupín: časopriestorové vlastnosti, role, združenia, obmedzenia a kvalita dát.

Konceptuálne modelovanie databáz nám pomáha porozumieť a popísať obsah budúcej priestorovej databázy. Tak tiež pomáha zvládnuť zložitost' problému, aby sa zjednodušila výmena a overovanie nápadov, zlepšil programovací proces a celkovo tak prispel k jednoduchšej údržbe výsledného systému. Inými slovami, databázové modely môžeme chápať ako nástroj myslenia, nástroj komunikácie, nástroj vývoja a nástroj dokumentácie (Bédard, 1999).

MDA/MDD

MDD predstavuje relatívne nový smer pre vývoj softvérových systémov. Tento spôsob vývoja softvéru sa vyznačuje tým, že obracia modely do podstaty a zameriava sa na vývoj softvérového systému nezávisle na možných platformách, ktoré budú použité. Táto zmena zamerania, od kódovo orientovaného po modelovo orientovaný vývoj, predstavuje významný posun v rozvoji, ktorý využíva rast abstrakcie. Prístup MDD umožňuje abstrakciu detailov implementovanej technológie a využitie konceptov bližších problémovej oblasti. V dôsledku toho sú modely používané pri vývoji systému ľahšie špecifikovateľné, zrozumiteľné a udržiavateľné. Sú robustné k zmenám technológii na implementáciu systému (Selic 2003).

Pre podporu tohto rozvoja softvéru prijala skupina Object Management Group (OMG) prístup MDA. Architektúra MDA definuje predstavu vývoja softvéru na základe modelu. (Arlow 2005). Prístup MDA je založený na tom, že model nám riadi výrobu architektúry spustiteľného softvéru. Model systému je popis

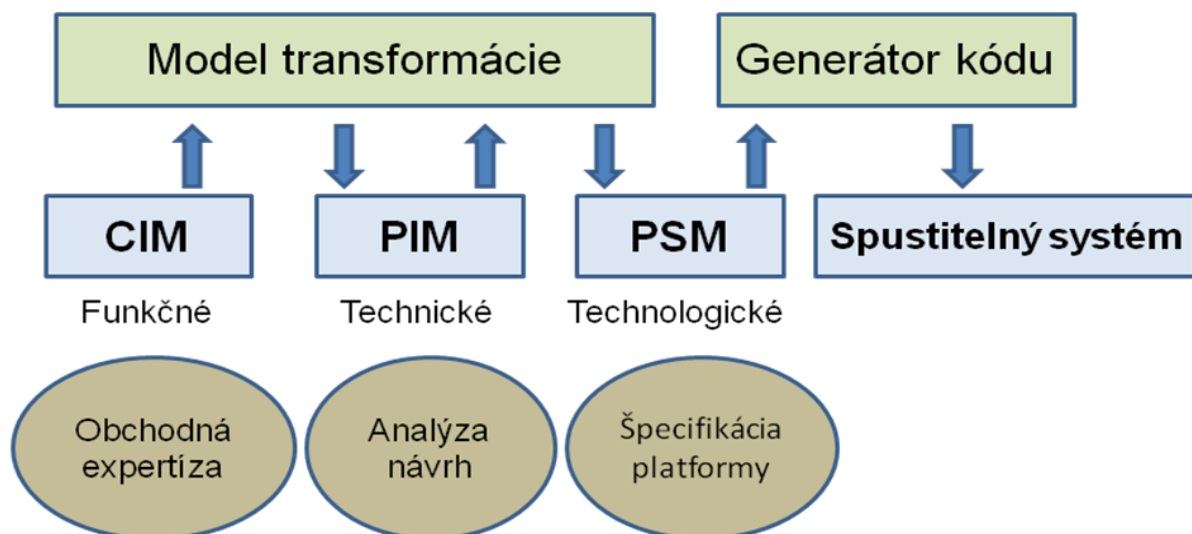
pomocou špecifickej notácie. Podľa Eriksona et. al. (2004) sú artefakty vyrobené v MDA formálne modely, to znamená, ide o modely, ktoré môžu byť porozumené počítačmi.

MDA je podporovaná tromi kľúčovými modelovacími technológiami OMG, ktoré sú založené na UML: MOF (Meta-Object Facility) – slúžia ako repozitáre modelu, CWM (Common Warehouse Metamodel) – štandard, ktorý popisuje, ako reprezentovať databázové modely (schémata), OLAP a datamining modely a XMI (XML Metadata Interchange) – mapovanie, pomocou ktorého môžeme vyjadriť UML modely v XML (Klimeš, 2004).

Podľa MDA je softvér vytváraný ako sled transformácií modelov vytváraných v modelovacom nástroji kompatibilnom s architektúrou MDA. V MDA sú systémové požiadavky modelované pomocou abstraktného modelu nezávislom na počítači „*Computation Independent Model*“ (CIM). Tento model sa nazýva doménový model alebo tiež aj obchodný model. Pre popis modelu používa známy slovník z domény expertov. CIM neukazuje detailnú štruktúru systému, ale prostredie, v ktorom bude systém fungovať. Je vhodný pre porozumenie problému (OMG 2003).

V druhej úrovni abstrakcií hľadáme model nezávislý na platforme „*Platform Independent Model*“ (PIM). Pod týmto modelom rozumieme model s pomerne vysokou úrovňou abstrakcie, ale nezávislý na konkrétnej implementačnej technológii (OMG 2003).

Neskôr by mal byť PIM transformovaný do modelu závislom na konkrétnej platforme „*Platform Specific Model*“ (PSM). PSM je postavený tak, aby nám stanovil podmienky systému z hľadiska implementačných konštruktorov, ktoré sú k dispozícii v konkrétnej implementačnej technológii. Relačná databáza PSM môže obsahovať napríklad výrazy ako „tabuľka“, „stĺpec“, „ cudzí kľúč “. PIM môže byť transformovaný do jedného alebo viacerých PSM modelov. Pre každú špecifickú technologickú platformu je generovaný zvlášť. Ďalším krokom je transformovanie jednotlivých PSM na zdrojový kód. Táto transformácia je pomerne priama, pretože každý PSM je prispôbený pre určitú technológiu (Nalon 2010).



Obr. 1. Schéma transformácie modelov MDA podľa OGC (upravené z goobiz, 2007)

Z výsledného modelu PSM je vygenerovaný celý zdrojový kód aplikácie a pomocné artefakty, ako sú dokumentácia, simulačný program, súbory zostavenia a deskriptory nasadenia. Pre dosiahnutie tohto cieľa musí model UML byť výpočtovo úplný, to znamená, že sémantika všetkých operácií musí byť vyjadrená v akčnom jazyku (ASL, ATL a pod.) (Arlow 2004). V takomto jazyku sa nadefinujú pravidlá, podľa ktorých prebieha automatická transformácia medzi jednotlivými modelmi.

Kľúčovým prvkom MDA je to, že transformácia by mala byť automaticky vykonaná. OMG poskytuje niekoľko spôsobov, ako transformovať modely v MDA. Jeden z nich je transformácia pomocou UML profilov. CIM a PIM môžu byť pripravené pomocou UML profilov v nezávislej platforme. Tento model môžeme potom transformovať do druhého profilu v špecifickej platforme (OMG 2003).

SÚČASNÝ STAV PROBLEMATIKY

V dnešnej dobe existuje mnoho formalizmov pre tvorbu modelov priestorových aplikácií. Všetky tieto modely boli navrhnuté s ohľadom na vyššiu úroveň abstrakcie, ktorá pomáha návrhárom v prvej fáze tvorby geo-aplikácií. Táto fáza, kde sa problémy oblasti berú do úvahy bez ohľadu na implementačné detaily, sa nazýva konceptuálna úroveň. V MDA táto fáza zodpovedá abstraktnému CIM modelu. Všetky formalizmy používané pre návrh geografických aplikácií pôsobia ako CIM. Formalizmy GeoProfil a Perceptory, približené v nasledujúcich podkapitolách, stanovujú základne pravidla pre transformácie modelov do nižšej úrovne.

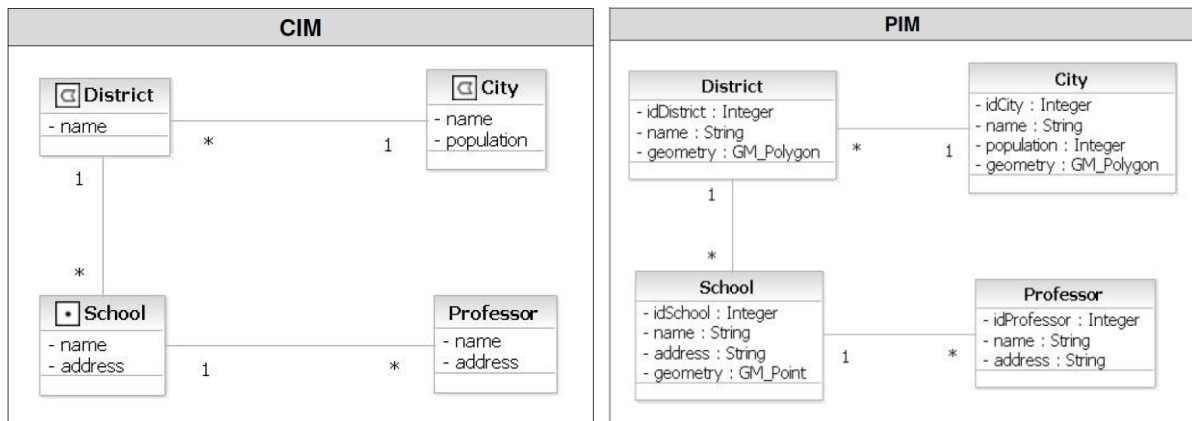
GeoProfil

V tejto oblasti sa podarilo navrhnuť formalizmus GeoProfil (Lisboa-Filho, 2010) formalizmus založený na UML, ktorý umožňuje vytvárať CIM na vysokej úrovni. V ďalšej práci od Nalon (2010) sú stanovené pravidlá pre transformáciu modelu CIM do PIM s využitím formalizmu GeoProfilu. Táto transformácia sa v klasickom prístupe nazýva Logicko-konceptuálne mapovanie. Nižšie, v uvedenej tabuľke, sú znázornené prvky GeoProfilu tak, ako sú závislé na triedach medzinárodných noriem rádu ISO 19100. Výhodou Geoprofilu je, že ho môžeme integrovať do rôznych CASE nástrojov (CASE RSM, StarUML...)

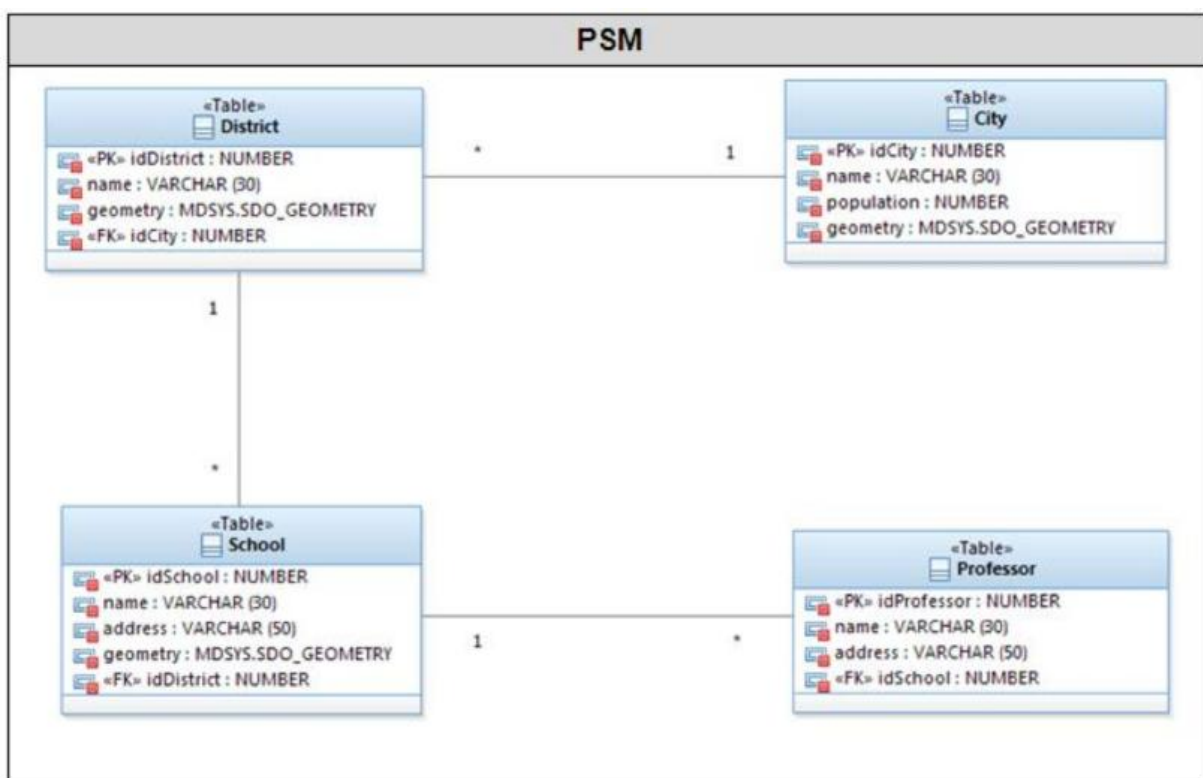
Tab 1. závislosť medzi prvkami GeoProfile a normami radu 19100 (Nalon 2010)

Requirements of BDGeo modeling	GeoProfile	Classes in the international standards	Standard
Geographical objects in the object view	<i>Point</i>	<i>GM_Point</i>	ISO 19107
	<i>Line</i>	<i>GM_Curve</i>	ISO 19107
	<i>Polygon</i>	<i>GM_Surface</i>	ISO 19107
	<i>ComplexSpatialObj</i>	<i>GM_Complex</i>	ISO 19107
Geographical objects in the field view	<i>TIN</i>	<i>CV_TINCoverage</i>	ISO 19123
	<i>Isolines</i>	<i>CV_SegmentedCurveCoverage</i>	ISO 19123
	<i>AdjPolygons</i>	<i>CV_DiscreteSurfaceCoverage</i>	ISO 19123
	<i>GridOfPoints</i>	<i>CV_DiscreteGridPointCoverage</i>	ISO 19123
	<i>GridOfCells</i>	<i>CV_GridCell</i>	ISO 19123
	<i>IrregularPoint</i>	<i>CV_DiscretePointCoverage</i>	ISO 19123
Network elements	<i>Node</i>	<i>TP_Node</i>	ISO 19107
	<i>Arc</i>	<i>TP_Edge</i>	ISO 19107
	<i>UnidirectionalArc</i>	<i>TP_DirectedEdge</i>	ISO 19107
	<i>BidirectionalArc</i>	<i>TP_DirectedEdge</i>	ISO 19107
Temporal objects	<i>Temporal Object</i>	<i>TM_Object</i>	ISO 19108
	<i>Instant</i>	<i>TM_Instant</i>	ISO 19108
	<i>Interval</i>	<i>TM_Period</i>	ISO 19108

GeoProfil využíva UML spolu s jeho dostupnými prostriedkami. GeoProfil zhromažďuje v definícii hlavne požiadavky na priestorové databázy a využíva vlastnosti existujúcich koncepčných modelov (GeoOOA, MADS, Perceptory...). V práci Nalon (2011) môžeme vidieť transformáciu modelu do PSM (*Oracle spatial*). Pre transformáciu medzi modelmi využívajú definíciu transformačných pravidiel v jazyku ATL.



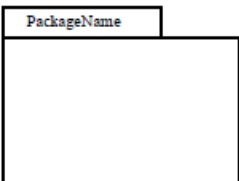
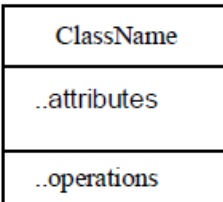
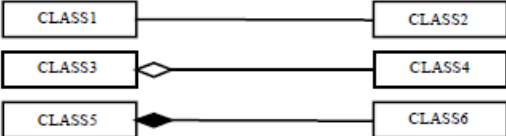
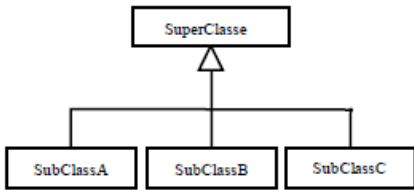
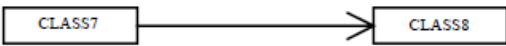
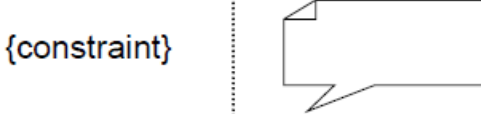
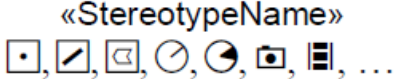
Obr. 2. Model CIM a PIM v GeoProfile (Nalon, 2011)



Obr. 3. Model PSM, pre Oracle Spatial (Nalon, 2011)

Perceptory

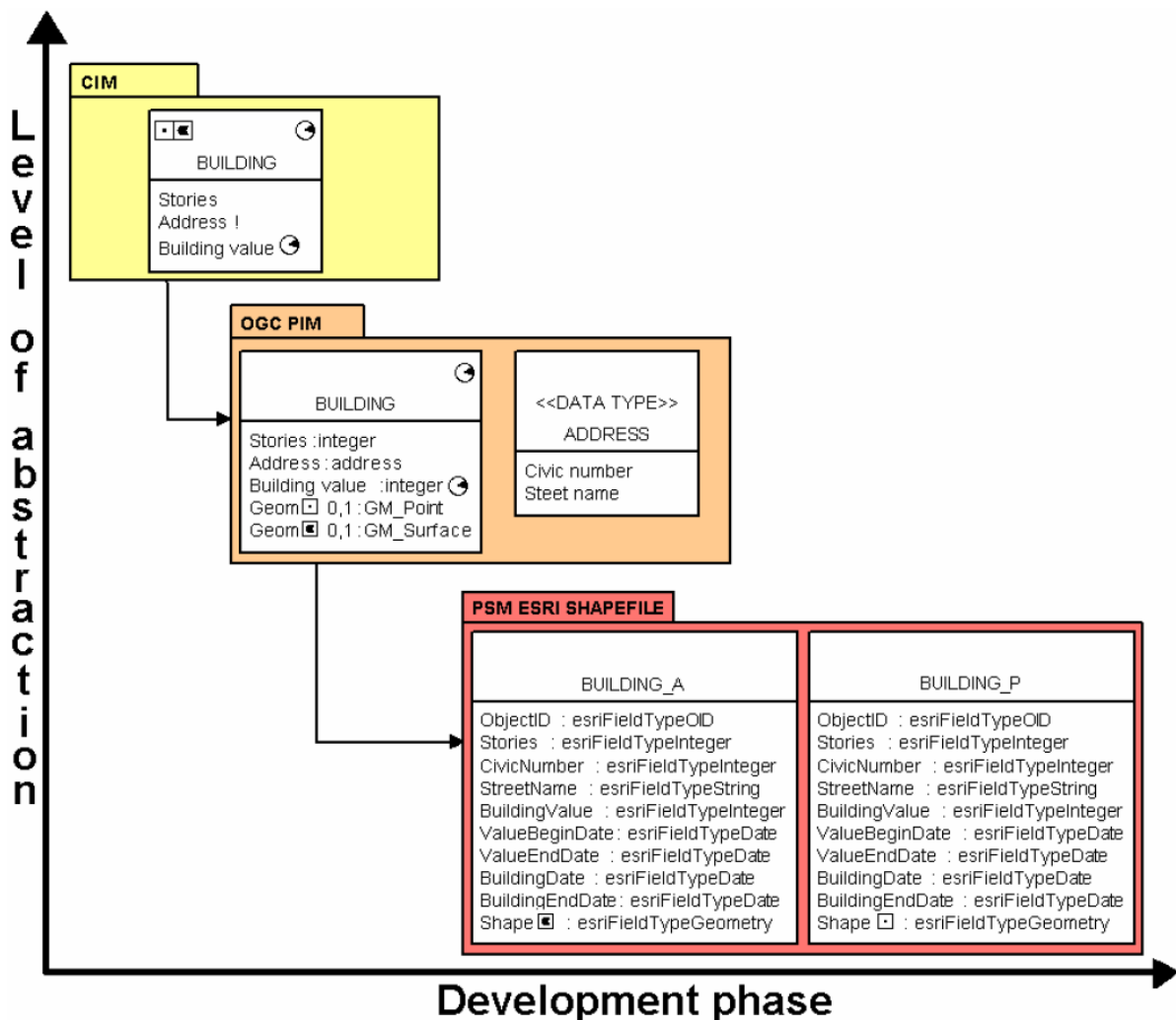
Ďalším formalizmom v tejto oblasti, ktorý je založený na UML, sú Perceptory (BÉDARD 1999). Perceptory používajú grafický jazyk postavený na UML diagrame tried (OMG 1997). Diagram tried objektov v UML rozšírené stereotypy založené na PVL (Plug-in for Visual Languages), vyjadrujú zodpovedajúce znalosti geografických a časových štruktúr databáz. Perceptory sú založené na norme ISO /TC211, to znamená, že prvky modelu perceptory sú tie, ktoré táto norma definuje (Bédard 1999). PVL a ISO/TC. Ako konceptuálne schémy sú dobrým doplnkom pre popis a modelovanie databáz. Zladenie týchto dvoch schém je demonštrované pomocou dedičnosti normy ISO 19107 a ISO 19108.

UML Feature		UML Notation
Package		
Class, attribute, and operation		
Relationship	Association, aggregation, and composition	
	Generalization/Specialization	
	Dependency	
Constraint and note		
Stereotype		

Obr. 4. prvky modelu Perceptory a ich grafický zápis

Perceptory používajú grafický jazyk postavený na UML diagrame tried (OMG, 1999). Používa UML stereotypy pre spracovanie priestorových a časových vlastnosti reálneho sveta do koncepcnej schémy. UML je de facto štandard v komunite informačných technológií, a ponúka mechanizmus pre definíciu nových prvkov ako priestorové a časopriestorové objekty. Tak tiež norma ISO / TC 211 a OGC vybralo UML pre modelovanie priestorových dát. Takže prvky modelu Perceptory sú tie, ktoré definuje táto norma: triedy, atribúty, operácie, vzťahy, balíčky, generalizácia, obmedzenia, poznámky, a stereotypy (Brodear, 2004).

Perceptory sú doposiaľ zamerané na statické diagrame UML. Piktogramy, ktoré používajú Perceptory, slúžia pre uľahčenie modelovania a sú používané predovšetkým na najvyššej úrovni modelovania. V MDA pôsobia skôr ako CIM než PIM alebo PSM (Bédart 2008). V práci Bédart (2007) môžeme vidieť transformáciu modelu Perceptory do modelu špecifickej platformy(PSM) (*ESRI Shapefile*).



Obr. 5. príklady CIM, PIM a PSM úrovni abstrakcie metódy MDA pre rovnakú aplikáciu pomocou piktogramov (Bedárd, 2007)

UML-GeoFrame

GeoFrame je koncepčný rámec, ktorý poskytuje základný diagram tried pre prvé kroky konceptuálneho dátového modelovania GIS aplikácií. Vzájomne využívanie UML tried a GeoFrame umožňuje riešenie väčšiny požiadaviek na modelovanie GIS aplikácií. Tento model je na úrovni CIM.

Ďalšie Konceptuálne modely

Ďalšie známe konceptuálne modely pre modelovanie priestorových databáz ako MADS, GeoOOA, OMT-G sú často využívané. Avšak ich výsledne modely sú na úrovni CIM a nie sú postavené na UML.

PROBLÉMY A NEVÝHODY MDD

Jednou z najväčších otázok problematiky vývoja softvéru metódou MDD je či ma zmysel využívať túto metodiku, alebo ostať pri kódovo zameranom prístupe. Táto metodika nie je relatívne mladá, ale stále prináša množstvo problémov. Väčšina nástrojov, ktoré podporujú túto metodiku bývajú nestabilne a často vygenerovaný kód obsahuje množstvo chýb a je nutne pristúpiť k ručnej oprave kódu. Na druhej strane nástroje, ktoré dokážu ako tak podporovať túto metodiku tak sú finančné nákladne a je otázna finančná návratnosť. Preto je nutne zvážiť niekoľko veľmi dôležitých aspektov pri rozhodnutí či použiť túto metodiku pri vývoji softvéru.

- Krivka učenia – nástroje na modelovanie sa vyvíjajú rýchlo a nie je jednoduché týmto nástrojom do hĺbky rozumieť

- Výber vhodných nástrojov – existuje mnoho projektov, ktoré sa zaoberajú vývojom nástrojov, ktoré by plne podporovali metodiku MDD. Tieto nástroje využívajú rôzne šablóny, rámce a generátory kódu, ktoré často prinášajú problémy a v niektorých prípadoch nedostatočne pokrývajú riešený problém.
- Štruktúrovanosť – často sú nástroje priveľmi štruktúrované pre popis riešeného problému
- Náklady na ladenie nástroja – aby sme mohli nástroj plnohodnotne využívať je potrebné ho vyladiť
- Náklady na zlyhanie

Slabé stránky MDD

Pri modelovaní metódou MDD sme narazili na niekoľko problémov.

- Slabá podpora zmien modelu (refactoring) –pri zmene pôvodného modelu často dochádza k chybám a výsledný kód vykazuje chyby.
- Transformácia sa ťažko testujú
- Podpora GIS – v modelovacích nástrojoch chýbajú šablóny, ktoré by umožňovali zobrazenie a editáciu priestorových dát. Pre podporu GIS je nutné využívať databázu, ktorá podporuje ukladanie a spracovanie priestorových dát, taktiež je nutné rozšíriť vývojové prostredie o komponenty, ktoré nám zabezpečia správne zobrazenie priestorových dát.

ZOSTAVENIE DÁTOVEHO MODELU

Základom každého informačného systému by mala byť správne navrhnutá databázová vrstva – jedna sa o schémy tabuliek a ich vzájomných vzťahov, ktoré by mali čo najlepšie odpovedať realite. Tento proces navrhovania často nazývame tiež modelovanie a jeho výsledok potom návrh modelu.

K tomu, aby bolo možné na najnižšej modelovej úrovni, ktorá je tiež často nazývaná konceptuálnou, navrhnuť model, ktorý bude využívať priestorové dáta a následne z tohto modelu vygenerovať zdrojový kód aplikácie, ktorá bude vedieť s týmito dátami pracovať je nutné špecifikovať základné typy priestorových dát. O to sa postaralo Konzorcium OGC (Open GIS Consortium), ktoré navrhlo tak zvaný geometrický objektovo orientovaný model.

V krátkosti sa tento pojem dá zhrnúť nasledujúco: Základom je dobre navrhnuť model budúcej aplikácie a z neho potom vygenerovať aplikáciu.

ZÁVER

Dnešné prístupy umožňujú konceptuálne modelovanie priestorových databáz na najvyššej úrovni abstrakcie. Pomocou definovaných pravidiel umožňujú transformáciu na modely nižších úrovní. Do akej miery sú tieto modely na nižšej úrovni použiteľné však nie je zatiaľ preskúmané. Pri tomto probléme vzniká viacero otázok: Sú dostačujúce stavajúce formalizmy? Je potreba ich rozšíriť? Je možné v súčasnosti modelovať geografické aplikácie pomocou MDA/MDD? Aké problémy to prináša?

Využitie tejto metodiky vývoja softvéru v GIS má veľkú perspektívu v dnešnej dobe je využívaná pri tvorbe softvéru v mnohých oblastiach (mestské informačné systémy, letiskové systémy, bankovníctvo a pod.) použitia sa tato metodika. Všetky tieto oblasti využívajú geografické informácie, respektíve používajú dáta, ktoré by bolo možné lokalizovať. V predchádzajúcej kapitole spomínané príklady využívajú túto metodiku pre tvorbu dátového modelu a vygenerovanie priestorovej databázy. A však pri tvorbe softvéru okrem spomínanej databázy je potrebné tieto dáta zobrazovať.

PODAKOVAVIE

Rád by som poďakoval spoločnosti ArteGIS s.r.o, ktorá finančne podporila výskum v oblasti Integrace GIS do metodologie Model Driven Development z vlastných prostriedkov a dotácie Moravskoslezkeho kraja v rámci dotačného programu Veda a výskum.

LITERATURA

- ARLOW, J.,(2004) Enterprise Patterns and MDA: *Building Better Software With Archetype Patterns and UML*, Addison-Wesley Professional, 2004 - Počet stran: 495
- BÉDARD, Yvan, (1999). *Visual Modelling of Spatial Database Towards Spatial PVL and UML*, Geomatica, Vol. 53, No. 2, pp. 169-185.
- BÉDARD, Y. and LARRIVÉE, S. (2008). *Modeling with Pictogrammic Languages*. In *Encyclopedia of GIS*, pages 716–725. Springer-Verlag.
- BRODEUR, Jean, BÉDARD, Yvan, PROULX, Marie-Josée, (2004): *Modelling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics*
- EA. (2010). Enterprise Architect. Disponível em: <http://www.sparxsystems.com/products/ea/>. Acesso em: 01 mar. 2010.
- ERIKSON, H., PENKER, M., LYONS, B and FADO, D. (2004), *UML 2 Toolkit*, OMG Press, 1th edition.
- FRIIS-CHRISTENSEN, A., TRYFONA, N. and JENSEN, C.S. (2001), “Requirements and research issues in geographic data modeling”, In: 9th ACM International Symposium on Advances in GIS, p. 2-8.
- GITZEL, R., KORTHAUS, A.,(2004): *The Role of Metamodeling in Model-Driven Development*,
- ISO/TC211. (2003). *ISO 19107: 2003 Geographic Information – Spatial Schema*. ISO, Geneva, Switzerland.
- ISO/TC211. (2002). *ISO 19108: 2002 Geographic Information – Temporal Schema*. ISO, Geneva, Switzerland.
- ISO/TC211. (2005). *ISO 19123: 2005 Geographic Information – Schema for Coverage Geometry and Functions*. ISO, Geneva, Switzerland.
- KLEPPE, A., WARMER, J. and BAST, W. (2003): *MDA Explained: The Model Driven Architecture: Practice and Promise*, Addison-Wesley Publishing Co., 2nd edition.
- KOSTER, G., PAGEL, B. and SIX, H. (1997): “GIS-Application development with GeoOOA”, In International Journal of Geographical Information Science, v. 11, n. 4, p. 307-335.
- KLIMES, C., PROCHAZKA, J.,(2004): *Využití MDA pro Integrovaný vývojový nástroj QI Builder*,
- LISBOA FILHO , J. and LOCHPE, C. (2008). *Modeling with a UML Profile*. In *Encyclopedia of GIS*, pages 691–700. Springer-Verlag.
- NALON, F.R., et al., (2010): *Using MDA and a UML Profile integrated with international standards to model geographic databases*, geoinfo2010-paper27.
- NALON, F.R., et al.: *Applying the Model Driven Architecture Approach for Geographic Database Design using a UML Profile and ISO Standards*, 2011
- OMG(a), (1997). UML Notation Guide, Version 1.1, 141 p., <http://www.rational.com/uml/>
- OMG, (2003). MDA Guide, v.1.0.1. OMG, Needham, MA, USA.
- OMG, (2007). Unified Modeling Language: Superstructure, v.2.1.2. OMG, Needham, MA, USA.
- PARENT, C., SPACCAPIETRA, S. and ZIMÁNYI, E. (2008). Modeling and multiple perceptions. In *Encyclopedia of GIS*, pages 682–690. Springer-Verlag.
- Pattern for Executable Service Specifications (PEXS) , dostupné z (citováno 23.4.2012): <http://www.goobiz.com/Pattern%20for%20Executable%20Specifications%20%28PEXS%29.htm>
- RSM. (2010). *Rational Software Modeler*. Disponível em: <http://www-01.ibm.com/software/awdtools/modeler/>. Acesso em: 01 mar. 2010.
- SELIC, B.: *The Pragmatics of Model-Driven Development*. *IEEE Software*, vol. 20, no. 5, September-October, 2003 , pp. 19 - 25.