

NÁSTROJE PRO DATOVÉ MODELOVÁNÍ

Vojtěch Merunka

Katedra informačního inženýrství, PEF, ČZU Praha
merunka@pef.czu.cz

ABSTRAKT:

Datové modelování je nový obor softwarového inženýrství, který se zabývá nejen výukou objektového programování a modelování, ale i nástroji a technikami sloužícími pro ověření správnosti analýzy softwarových aplikací. V tomto článku je diskutován vlastní obor datového modelování a použitelné počítačové nástroje. Je také podána informace o novém vyučovaném předmětu na ČZU a knize zabývající se tímto tématem.

KLÍČOVÁ SLOVA:

CASE Daskalos, BlueJ, objektově orientovaná analýza, objektová databáze, výuka informatiky, datové modelování, UML, analýza systémů.

1 Úvod

V minulosti byl každý, kdo dovedl pracovat s počítačem, považován automaticky za programátora. S rozvojem hotových programů nejrůznějšího použití již nebyla znalost principů činnosti počítačů tolik potřeba, a tak se objevili takzvaní „uživatelé“. Podle tohoto historického dělení na „programátory“ a „uživatele“ lze dnes prohlásit, že pravděpodobně více než 99% lidí pracujících s počítačem nejsou „programátoři“, ale jen „uživatelé“. Dokonce i většina z těch, kteří se programování učili ve škole, jej v praxi nikdy nepoužije.

Proto nepovažuji toto dělení za rozumné. Uživatelem počítače může být například malé dítě, které ještě neumí ani číst a psát, ale dobře ovládá svoji oblíbenou počítačovou hru, dokáže počítač správně zapnout i vypnout a umí spustit jakýkoliv nainstalovaný program pomocí ikonky a menu grafického uživatelského rozhraní. Uživatelem ale může být také fyzik, statistik nebo třeba architekt, který pracuje se složitým softwarem vyžadujícím hluboké znalosti v daném oboru.

Uživatelem je totiž dnes každý, kdo pracuje s počítačem. Proto tato kategorie ztrácí smysl. Znalosti konstrukce počítačů, principů činnosti počítače a programování počítačů nejsou pro využívání počítačů téměř potřeba a je to tak dobře. (Podobně jako znalosti teorie elektromagnetického pole nejsou nezbytné pro poslech rozhlasu nebo znalost principu pracovního cyklu tepelného stroje pro řízení automobilu.) Například ekonom, chce-li pracovat s počítačem, může a dokonce musí zůstat ekonomem. Nemůžeme po něm chtít, aby se stal „programátorem pro ekonomy“.

2 Co to je datové modelování

Vysvětlovat, jak se počítačové programy a to včetně například grafického uživatelského rozhraní a spolupráce s operačním systémem programují, je dnes pro většinu uživatelů počítačů zbytečné, protože pracují s hotovým softwarem.

Být „uživatelem s řidičským průkazem“ na hotové programy však v mnoha případech odborné praxe nestačí. Není totiž pravda, že na všechno je k dostání hotový software, který předem uspokojuje požadavky uživatelů. Není to možné. Proto je značná část počítačových programů nastavitelných, lze je přizpůsobovat a měnit. A tam, kde ani toto nepomáhá, se musí nové programy

vyrobit na míru. To je především vlastnost podnikových informačních systémů a vůbec všech programů na zpracování dat.

Proto by měl kvalifikovaný uživatel vědět, jak svůj hotový program přizpůsobit pro práci s údaji z nových úloh, tato nová data si vytvořit a pracovat s nimi. A když to nestačí, tak by měl umět tvůrcům softwaru dobře definovat svoje požadavky na novou funkčnost tak, že popíše i prakticky otestuje, jak a s jakou informací potřebuje na počítači pracovat. Právě tento soubor znalostí patří do oboru datového modelování.

Scott Ambler ve své knize „The Object Primer“ (Ambler 2004) definuje datové modelování jako

„Data modeling is the act of exploring data-oriented structures. Like other modeling artifacts data models can be used for a variety of purposes, from high-level conceptual models to physical data models.“

Jinou, mnohem stručnější, definici přináší Wikipedia:

In computer science, data modeling is the process of structuring and organizing data.

Datové modelování je specifická část softwarového inženýrství. Datové modelování nemá za cíl tvorbu programů ani obsluhu databázových systémů. Naivním uživatelům se sice nástroje datového modelování, jako jsou například diagramy nebo formální zápis zdají být totožné s programováním, ale ve skutečnosti tomu tak není. Datové modelování ale na druhou stranu není ani pouhé kreslení diagramů a psaní manažerské dokumentace, jak si mnoho lidí od IT myslí. Proto si pro větší názornost datové modelování porovnáme s programováním následující tabulkou.

	PROGRAMOVÁNÍ	DATOVÉ MODELOVÁNÍ
Používání programovacích jazyků.	Všechny prostředky jazyka včetně např. překladače a prostředků pro ladění a testování.	Jen vybraná část programovacích jazyků jako nástroj pro zápis dat a pro manipulace s daty.
Používání knihoven softwarových komponent.	Pro realizaci funkčních požadavků vytvářeného softwaru.	Nepoužívá se.
Používání návrhových vzorů.	Všechny vzory především pro realizaci funkčních požadavků vytvářeného softwaru.	Jen některé strukturální vzory pro popis dat.
Používání formálního aparátu.	Nepoužívá se.	Je nástrojem pro popis dat a pro manipulace s daty. (Výroková logika, operace s množinami, ...)
Používání diagramů při analýze.	Všechny druhy diagramů včetně diagramů popisujících chování a změny systému v čase.	Jen diagramy popisující vlastnosti dat a vztahy mezi nimi.

Tab. 1. Srovnání datového modelování a programování.

3 Jak poznat správný návrh?

Řada analytiků se domnívá, že pro každý objekt či přesněji třídu objektů existuje jedno jediné univerzálně použitelné nejlepší řešení bez ohledu na řešený problém. Tedy že například fakturu modelujeme správně takto ... osoby takto ... auta takto ... apod. A že datové modelování konkrétních úloh je jen o vybírání a propojování předem jednoznačně daných popisů tříd.

Domnívám se, že to není pravda. V různých úlohách totiž potřebujeme pro zdánlivě stejné objekty jiné atributy jiné chování. Nikdy neděláme „simulační model celého světa“, ale vždy jen nějakou horizontálně i vertikálně vymezenou a zjednodušenou část z celého světa. Nejde jen o to, že spoustu atributů v daných problémech nepotřebujeme a tak že je zbytečné je na objektech držet nepoužité. Jsou totiž i případy, kdy některé atributy u stejných mít nesmíme a jindy musíme. Například v lékařské ordinaci je věk pacienta potřebný, ale v obchodním informačním systému si lze představit situace, kdy věk osob nejen nepotřebujeme, ale ze zákona ho dokonce používat nesmíme. Tento malý příklad s osobami tak podává důkaz, že konkrétní struktura objektů i vazby mezi objekty navzájem je velmi závislá na konkrétním požadavku na práci s těmito objekty podle aktuálního zadání.

Datové modelování je proto složitým procesem, který si lze představit jako přeměnu zadání do konkrétních struktur a dat. Naštěstí zde nejsme odkázáni jen na intuici a zkušenost analytiků, ale máme k dispozici několik pokročilých metod, které mohou tento proces usnadnit a upřesnit. Jde o následující techniky:

1. Objektová normalizace.
2. Využívání návrhových vzorů.
3. Refaktoring.
4. Přístup MDA (Model Driven Development).

4 Ověření správnosti analýzy pomocí počítačových programů

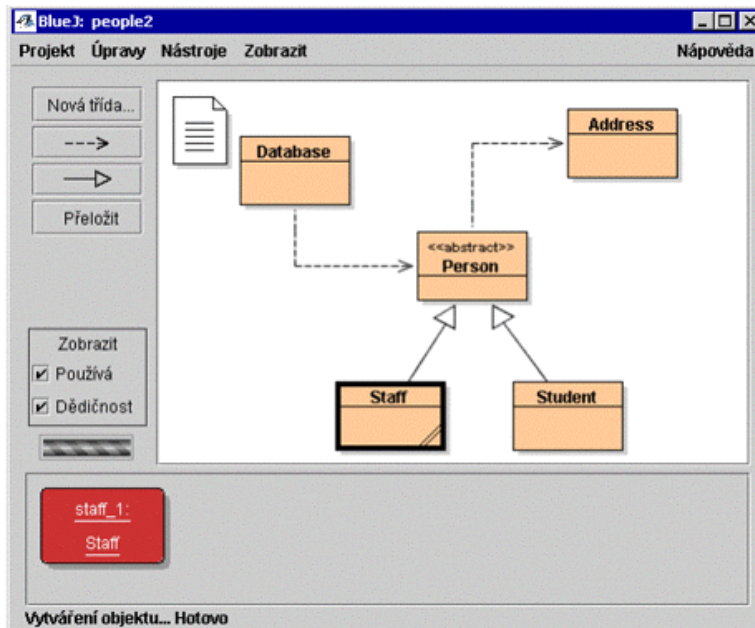
Z dosavadního výkladu vyplývá, že sestavení datového modelu v podobě diagramů tříd objektů nestačí, protože běžné modelovací nástroje neobsahují žádné prostředky k ověření správnosti navrženého modelu. Proto se v poslední době objevují počítačové programy, ve kterých se model nejen „nakreslí“, ale je možné s ním prakticky pracovat podobně jako s daty v databázových systémech.

4.1 BlueJ

Prostředí BlueJ bylo vyvinuto v druhé polovině devadesátých let Michaelem Köllingem a Johnem Rosenbergem jako reakce na tehdejší neuspokojivý stav vývojových prostředí použitelných při výuce objektového programování v jazyce Java.

Jak říká přední český propagátor jazyka Java, Rudolf Pecinovský, BlueJ je integrované vývojové prostředí navržené speciálně pro výuku základů objektově orientovaného programování v jazyce Java. Jeho autoři se totiž při výuce programování potýkali s různými nepříjemnými vlastnostmi stávajících prostředí a rozhodli se vytvořit prostředí, které by maximálně vyhovovalo potřebám úvodních kurzů programování na vysokých školách a v různých školicích střediscích. Jimi vytvořené prostředí má následující základní vlastnosti:

1. Je objektově orientované. Začínající studenti se v něm setkávají s třídami a jejich instancemi daleko dříve než se zdrojovým kódem. BlueJ jim umožňuje přímo vytvářet objekty a volat jejich metody, aniž by potřebovali cokoli vědět o jazyku, v němž je vše naprogramováno.
2. Vizualizuje strukturu tříd a objektů. BlueJ uživatelům umožňuje (stejně jako některé profesionální vývojové nástroje) definovat třídy, rozhraní a jejich vzájemné závislosti přímo v UML diagramu tříd, přičemž prostředí samo vygeneruje příslušné kostry definovaných tříd a rozhraní. Oproti jiným prostředím však navíc umožňuje přímo, interaktivně vytvářet jednotlivé objekty a volat jejich metody.

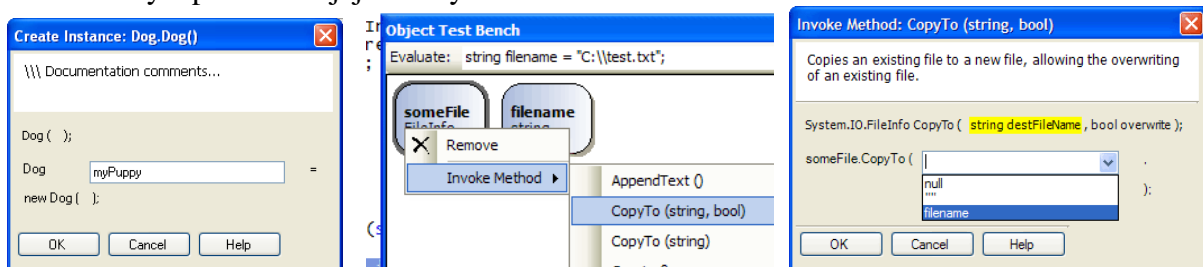


Obr. 1. Ukázka BlueJ (převzato z <http://www.ceskaskola.cz>)

I když BlueJ vzniklo jako nástroj pro výuku základů objektového programování, tak je velmi dobře využitelné také při analytické práci právě k ověřování správnosti návrhu analytického datového modelu. BlueJ bylo prvním počítačovým programem na světě, ve kterém bylo možné sestavit model tříd způsobem známým z CASE nástrojů, ale na rozdíl od nich, kde model je pouhým obrázkem, bylo možné k jednotlivým třídám tvořit skutečné datové objekty a pracovat s nimi.

4.2 .NET

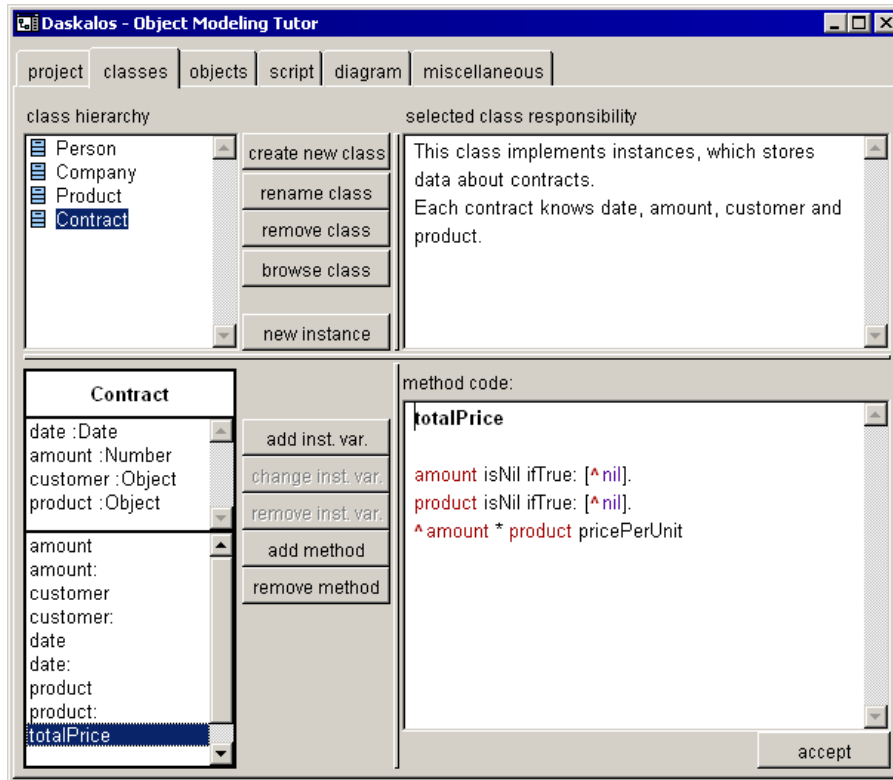
Na základě úspěchu BlueJ vznikl podobný nástroj i v prostředí .NET firmy Microsoft. Je to Object Test Bench, který je součástí balíku Visual Studio. Tento nástroj podporuje přístup „Object-level testing“. Jde o možnost jednotlivě vytvářet konkrétní datové objekty uvnitř modelu tříd, vykonávat na nich metody a pracovat s jejich daty.



Obr. 2. Příklad práce s konkrétními objekty

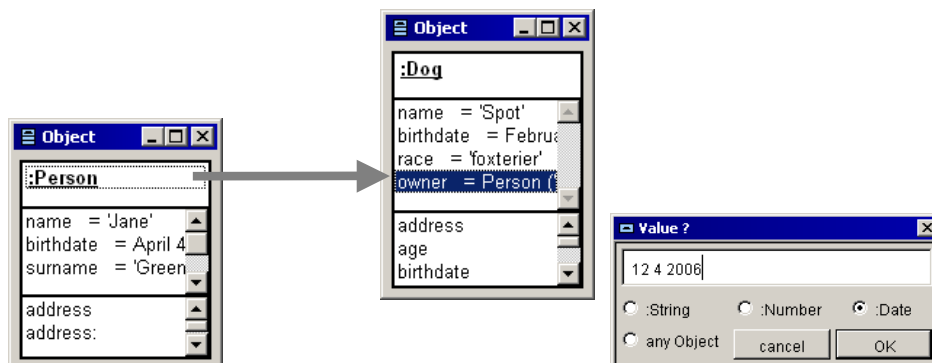
4.3 Daskalos

Daskalos (z řeckého slova učitel) je počítačový program pro objektivě orientované datové modelování. V Daskalu je možné vytvořit třídy a množiny objektů a programovat metody. Protože vzniklý kód je součástí standardního vývojového prostředí Smalltalku, tak lze Daskalos použít jako vizuální nástroj pro tvorbu datových objektů určených pro běžně vyvíjené aplikace.



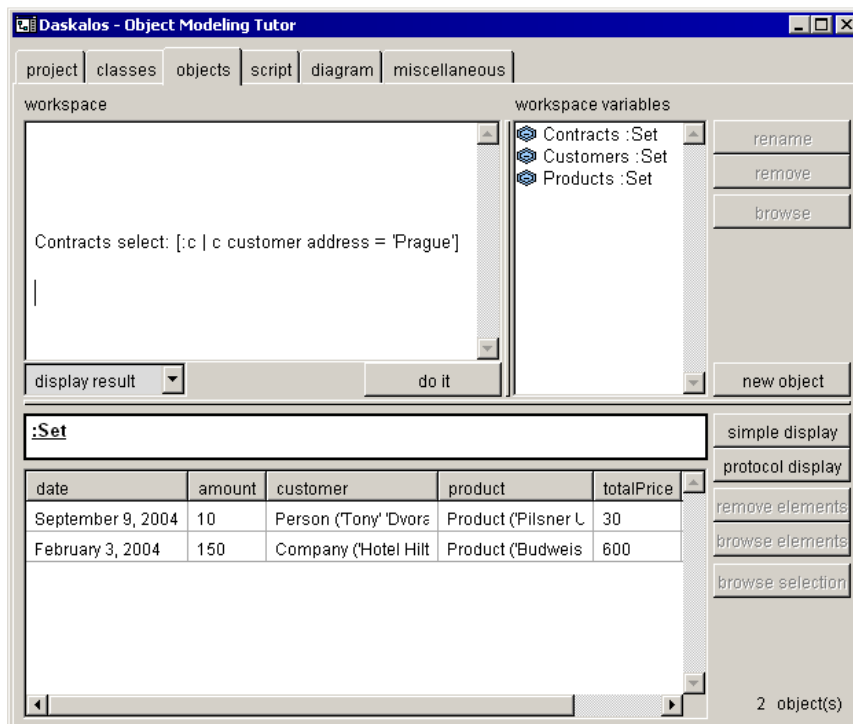
Obr. 3. Tvorba tříd a metod.

Daskalos dovoluje objekty testovat. Objekty a třídy objektů jsou zobrazovány podle standardu UML přičemž s obsahem takto zobrazených symbolů lze přímo pracovat. Objekty lze vyčleňovat do samostatných oken a takto zobrazeným objektům lze také posílat zprávy přímo kliknutím na zobrazený objekt nebo přímo manipulovat s atributy objektů způsobem táhni-pusť:



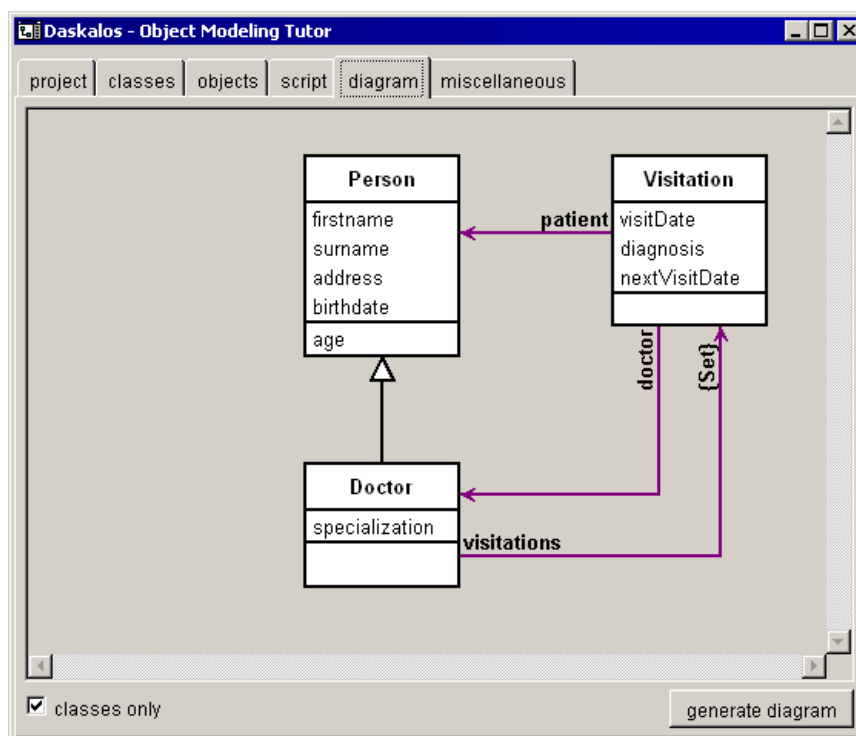
Obr. 4. Manipulace s objekty.

Pro komplikovanější operace s objekty – například kladení dotazů nad množinami či jinými kolekcemi objektů a pro transformace modelu – je možné využít pracovní panel, ve kterém lze příslušné výrazy vyhodnocovat a pracovat s jejich výsledky:



Obr. 5. Pracovní panel.

Objekty, které jsou potřeba k testování, je možné vytvářet nejen vizuálními prostředky, ale i obvyklým způsobem ze zdrojového kódu. Třídy a množiny objektů, se kterými se pracuje, jsou v Daskalu zobrazovány také v podobě diagramu tříd. Symboly tohoto diagramu, jejich obsah a vazby mezi nimi jsou synchronizovány se skutečným obsahem objektů z pracovního panelu. To znamená, že podoba diagramu se mění podle toho, jak se s objekty v pracovním panelu pracuje. Pokud například v paměti není žádný konkrétní objekt, který skládá jiný objekt, tak se v diagramu vazba skládání mezi symboly neobjeví.



Obr. 6. Panel s diagramem.

Projekt se v Daskalu ukládá nejen do datového souboru ve formátu XML, ale je také generována dokumentace obsahující zdrojové kódy, data i diagramy ve formátu HTML.

The screenshot shows a web browser displaying the Daskal-generated HTML documentation for a project titled "Creatures - Persons and Dogs". The page is divided into several sections:

- Workspace:** Contains code for workspace objects, including a set of friends and workspace objects.
- Script:** Contains code for creating and initializing objects, such as a person and a dog.
- Diagram:** A UML class diagram showing the relationships between classes. It includes an abstract class 'Creature' with subclasses 'Person' and 'Dog'. 'Person' and 'Dog' are associated with each other via an 'owner' relationship.
- Classes:** Detailed documentation for the 'Creature' and 'Person' classes, including their instance variables, methods, and non-accessing methods.

Obr. 7. Dokumentace ve formátu HTML zobrazená webovým prohlížečem.

5 Předmět datové a znalostní modelování na ČZU

Cílem předmětu je podat základy moderních metod modelování dat. Základním paradigmatem je objektově orientovaný přístup na konceptuální úrovni, přičemž konkrétní implementační datové modely jako je síťový, relační, objektově relační a objektový třídě-instanční budou vykládány jako jednotlivé transformace základního konceptuálního modelu. Budou vykládány a prakticky procvičovány možnosti návrhu datových struktur pro softwarové aplikace s využitím základních pojmů jako jsou pojmy entita, množina, objekt, metoda, zpráva, třída, instance třídy a kolekce (množina) objektů. Praktická cvičení budou využívat vizuální modelovací nástroj, zjednodušený programovací jazyk a část notace UML.

Předmět je koncipován jako souběžný s předmětem Algoritmizace a měl by spolu s ním vytvořit základ pro následnou výuku programování, databází a projektování.

	<i>přednášky</i>	<i>cvičení</i>
1	Úvod do teoretických základů moderního programování, lambda-kalkul.	Seznámení se s modelovacím nástrojem. Praktická cvičení.
2	Základy konceptuálního objektově orientovaného přístupu, pojmy třída objektů, kolekce objektů. Polymorfismus objektů. Dědění objektů a skládání objektů.	
3	Objektový model výpočtu. Metody a zprávy. Lambda výraz jako metoda i samostatný objekt. Využití při modelování znalostí.	Seznámení se s programovacím jazykem. Praktická cvičení.
4	Úvod do modelovacího jazyka UML.	
5	Pojem protokol objektu. Realizace atributů reprezentujících znalosti pomocí skládání i pomocí metod.	Práce s daty předem připravených úloh.
6	Možnosti dotazování nad množinami.	
7	Příklad reálného projektu s datovými objekty.	Zadání semestrálních projektů.
8	Konceptuální normalizace.	
9	Strukturální návrhové vzory.	Modifikace dat v předem připravených úlohách.
10	Behaviorální návrhové vzory.	
11	Refaktoring.	Konzultace semestrálních projektů.
12	Problematika změny objektového schématu. Migrace objektů mezi verzemi. Transformace dat reprezentujících znalost. Otázky návrhu „správné“ struktury.	
13	Různá konkrétní implementační paradigmatu modelování dat a znalostí.	Prezentace semestrálních projektů.
14	Implementace v různých aplikačních programovacích jazycích. Polymorfismus, dědění a skládání v Javě, C#, PHP a dalších.	

Obr. 8. Tab. 1. obsah předmětu.

6 Kniha Datové modelování

O objektově orientovaném přístupu byla již napsána řada prací. Tyto publikace se ale zaměřují na to, aby se čtenář naučil objektově programovat. Ve většině případů jde proto o učebnice smíšených programovacích jazyků (například C++, C# nebo Java), ve kterých je objektový přístup obsažen jen částečně a v kombinaci se starším procedurálním přístupem. Tato kniha je jiná. Má za cíl popsat a objasnit možnosti objektově orientovaného přístupu v datovém modelování. Obsah knihy je rozdělen a do dvou částí.

- V první části jsou diskutovány techniky a nástroje vlastního datového modelování. Výklad se zde opírá o formální aparát pro popis chování objektů využívající lambda-kalkul, který zde plní roli jednotícího prvku. Pro praktickou demonstraci je použit původní software a zjednodušený čistý objektově orientovaný programovací jazyk.
- Druhá část se zabývá pohledem na datové modelování v širších souvislostech projektování a tvorby informačních systémů. V této části jsou rozebírány také různé problémy spojené s tvorbou softwaru a naznačena některá možná řešení v perspektivě informačního managementu.

Vyložené principy jsou využitelné především v aplikovaném výzkumu, při tvorbě softwaru ve většině dnes používaných programovacích jazyků a databázových systémů, v projektech

softwarového inženýrství pro formulaci a verifikaci požadavků na informační systémy a nebo při práci s nejrůznějším aplikačním softwarem pro zpracování dat.

Tato kniha je také doporučeným studijním textem pro posluchače Provozně ekonomické fakulty České zemědělské univerzity v Praze pro předmět Datové a znalostní modelování a Elektrotechnické fakulty ČVUT v Praze pro předmět Objektové modelování.

Součástí knihy je také webová stránka (<http://kii.pef.czu.cz/~merunka/books/DatoveModelovani/>) s nekomerční licencí použitého vývojového prostředí, ve kterém si lze prakticky vyzkoušet předloženou látku.

7 Závěr

Objektově orientovaný přístup se dnes stal hlavním způsobem tvorby softwaru i analýzy systémů. I když ale dnes máme k dispozici CASE nástroje, výkonná vývojová prostředí, sofistikované knihovny a komponenty, tak není pravda, že vytvářené systémy jsou bez problémů. Nejen studenti, ale i vývojáři ve firmách občas vytvářejí velmi bizarní výtvořky. Setkáváme se s nesprávným používáním vazeb a hierarchií mezi objekty, krkolomnými triky v kódu atp. Problém takových aplikací není ale vždy v tom, že nefungují. Naneštěstí jsou mnohdy v chodu díky moderním komponentám, vývojovým prostředím a výkonnému hardwaru opravdu podivné konstrukce. O to je potom horší debata s tvůrcem, že by měl ve vlastním zájmu systém přepracovat, protože argumentům o obtížné údržbě, rozšiřitelnosti, spolehlivosti, nebezpečí nekonzistence a redundance dat, ... není ochoten naslouchat.

Datové modelování založené na práci s reálnými objekty je souborem nástrojů a technik, které původně vzniklo z potřeb výuky, ale dnes je prakticky použitelným prostředkem pro přesnější a správnější analýzu softwarových aplikací.

Tento článek obsahuje látku podporovanou grantem MSM6046070904 na výzkum v oblasti znalostních databázových systémů.

8 Literatura

Ambler S.: *The Object Primer 3rd Edition, Agile Model Driven Development with UML 2*, Cambridge University Press, 2004 ISBN#: 0-521-54018-6

Merunka V.: *Datové modelování*, Alfa Publishing, Praha 2007, ISBN 80-86851-54-0.

Pecinovský R.: *Myslíme objektově v jazyku Java 5.0*. Grada 2004. ISBN 8024709414.