

První zkušenosti s modelovacím nástrojem Craft.CASE

Vojtěch Merunka

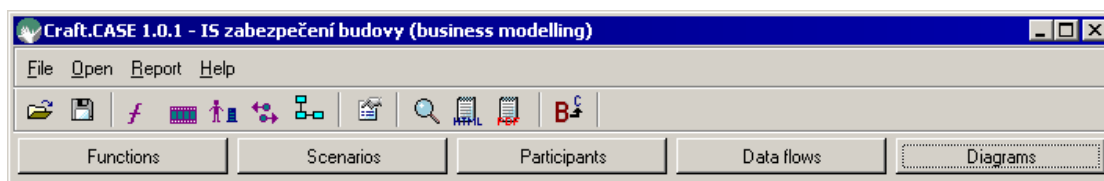
Katedra informačního inženýrství, PEF, ČZU Praha
merunka@pef.czu.cz

Abstrakt. Příspěvek seznamuje s první verzí modelovacího nástroje Craft.CASE, který je vyvíjený firmou e-Fractal s.r.o. pro mezinárodní poradenskou a konzultační firmu Deloitte.

Klíčová slova: CASE, Smalltalk, VisualWorks, Craft.CASE, BORM, objektivě orientovaná analýza, objektivá databáze

1 Úvod

Craft.CASE® je původní český modelovací a analytický CASE nástroj podporující metodu BORM®, která je založena na kombinaci objektivě orientovaného přístupu a procesního modelování. Nástroj vzniká ve firmě e-Fractal s.r.o. na zakázku pro mezinárodní poradenskou a konzultační firmu Deloitte & Touche (uživatele a spolutvůrce BORMu). Vedoucí vývoje ve firmě e-Fractal s.r.o. jsou ing. Ladislav Lenárt a ing. Petr Skála. Ve firmě Deloitte je hlavním zadavatelem Jiří Polák a analytikem je autor tohoto článku. Program je napsán v prostředí VisualWorks/Smalltalk a je určen pro použití ve Windows 2000 a XP.



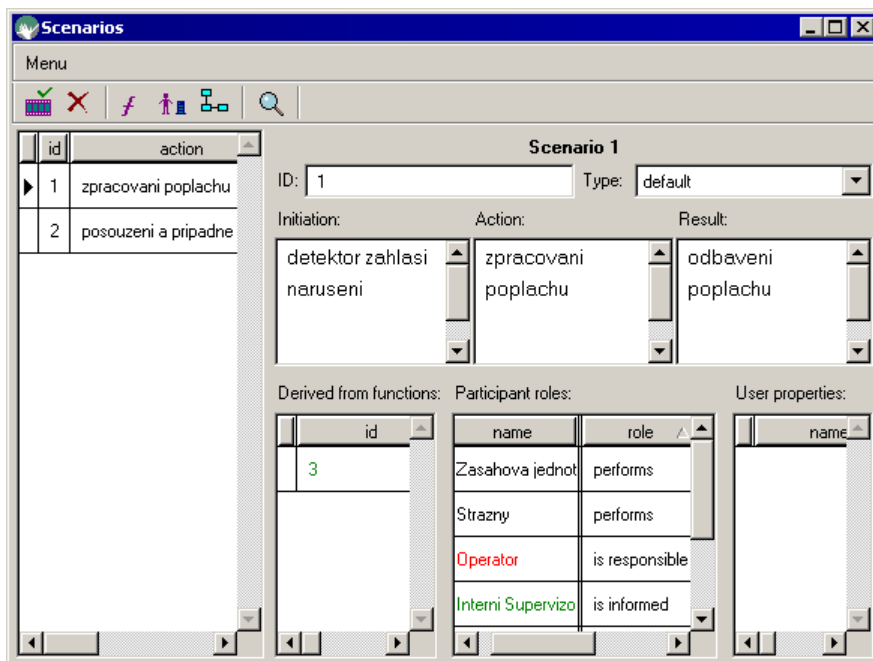
obr. 1 – hlavní (spouštěcí) okno

Craft.CASE verze 1.0 podporuje business analýzu, což je úvodní fáze metody BORM, a ve vývoji je také podpora konceptuální analýzy. V obou fázích analýzy se rozpoznává a modeluje zadání pro systém na základě objektivě modelování business procesů. Craft.CASE během modelování kontroluje úplnost a správnost modelu pomocí informací uložených v projektové databázi. V blízké budoucnosti se předpokládá rozšíření tohoto nástroje o katalogy návrhových vzorů, grafický simulátor procesů a víceuživatelský režim, který umožní pracovat na jednom projektu z více počítačů současně napojených na sdílený repozitář projektů implementovaný jako aplikace v objektivě databázovém systému Gemstone/S.

2 Business analýza v Craft.CASE – první krok metody BORM

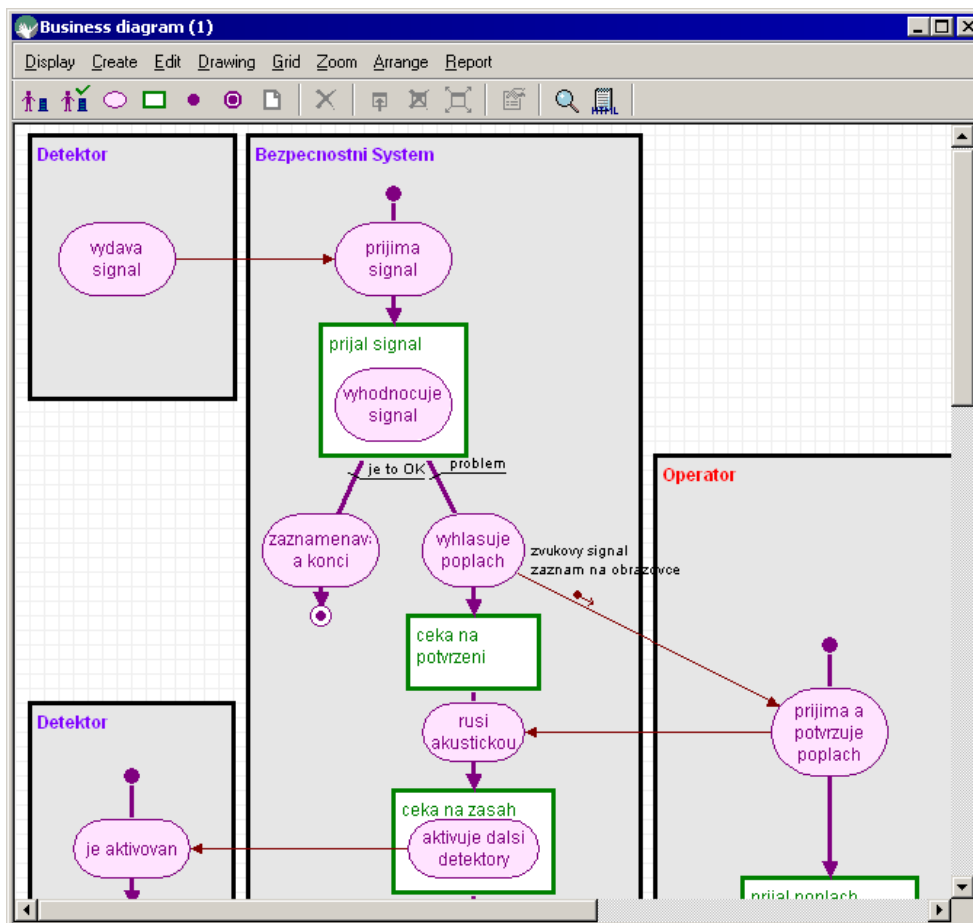
První fází je fáze business modelování. Tato fáze spočívá v rozpoznání a modelování problému. Zde se analyzuje celý kontext modelovaného systému – především objekty a procesy v organizaci, pro kterou se systém analyzuje. Ve složitějších případech je třeba sestavit dvě sady modelů. První z nich je tzv. AS-IS model, který zobrazuje stávající stav a po jeho dokončení následuje tzv. TO-BE stav, který zobrazuje novou strukturu objektů a procesů po implementaci systému.

V této fázi analýzy se nejdříve definují funkce systému podle požadavků zadavatele. Dalším krokem je sestavení scénářů z rozpoznávaných funkcí. Každý scénář obsahuje popis činností, které lze podrobně zobrazit procesním diagramem.



obr. 2 – editor scénářů

Součástí každého scénáře by měly být také participanty – objekty, které se procesu ve scénáři účastní. U participantů lze nastavovat jejich různé role v modelovaném procesu.



obr. 3 – příklad procesního diagramu

Tento původní diagram BORMu současně zobrazuje jak stavové diagramy každého z objektů - účastníků procesu ve scénáři, tak i jejich vzájemnou interakci v různých stavech včetně datových

toků, které si při této interakci mohou odehrávat. Stavový diagram participaanta zobrazuje jeho roli v procesu.

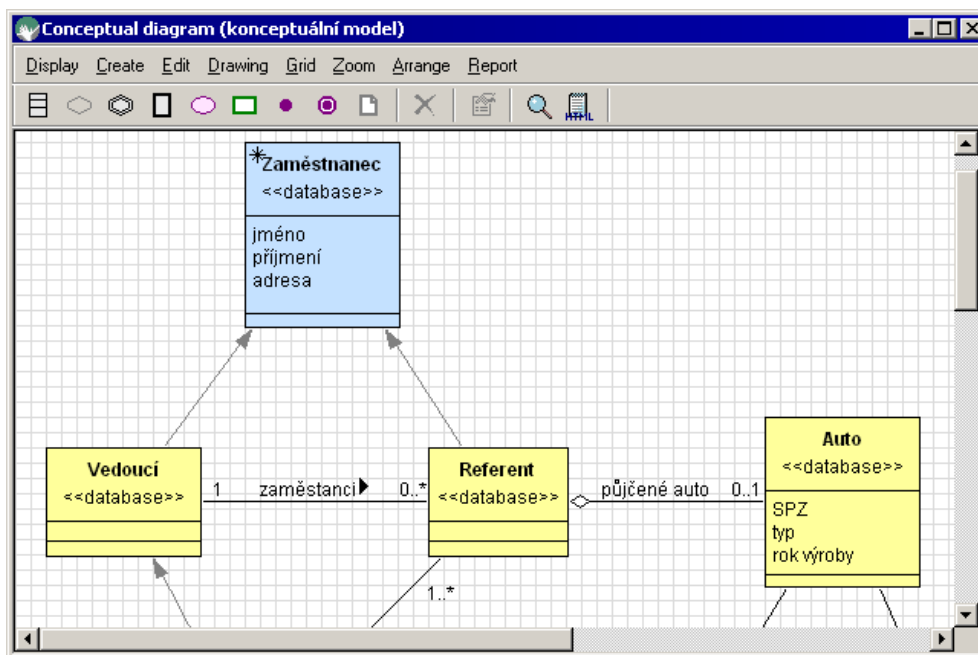
Druhý pohled je sled komunikací mezi aktivitami (činnostmi) různých objektů v různých stavech, což zobrazuje vlastní průběh procesu. Na proces je nahlíženo jako na spojení vzájemně komunikujících automatů.

Business analýza BORMu se používá k modelování požadavků na informační systém. Dalším možným použitím je modelování čistě procesních diagramů nikoliv za účelem pozdější implementace nějakého informačního systému (tedy následné konceptuální analýzy), ale přímo pro účely organizačního poradenství. Objektově orientované procesní modely BORMu slouží k nalezení slabin ve stávající organizaci a procesech a k návrhu změn, které by tyto nedostatky odstranily.

3 Konceptuální analýza v Craft.CASE – druhý krok metody BORM

Konceptuální analýza je postavena na upraveném standardu UML. Slouží jako spojovací článek mezi business modelem a softwarovým řešením. Podle BORMu jde o postupnou transformaci objektů a vazeb do podoby popisující softwarové řešení. Tato transformace podléhá určitým pravidlům, která Craft.CASE podporuje. Tato fáze modelování v Craft.CASE je velmi podobná klasickým nástrojům CASE používající jazyk UML. Odlišnosti spočívají ve dvou věcech:

- Pojmy v jazyce UML v této fázi modelování vycházejí z pojmů modelovaných v předchozí fázi. Craft.CASE tento vztah podporuje a kontroluje.
- Jazyk UML je zjednodušen, ale na druhou stranu také doplněn o některé nové prvky. To vše za účelem podpory objektově orientovaného konceptuálního modelování více nezávislého na implementačních prostředích smíšených programovacích jazyků (např. C++). Originální UML je totiž s těmito jazyky příliš těsně svázán. To nejen zbytečně komplikuje analýzu, ale také nedává dostatek výrazových prostředků pro implementaci v čistě objektově orientovaných prostředích a především objektových databázích.

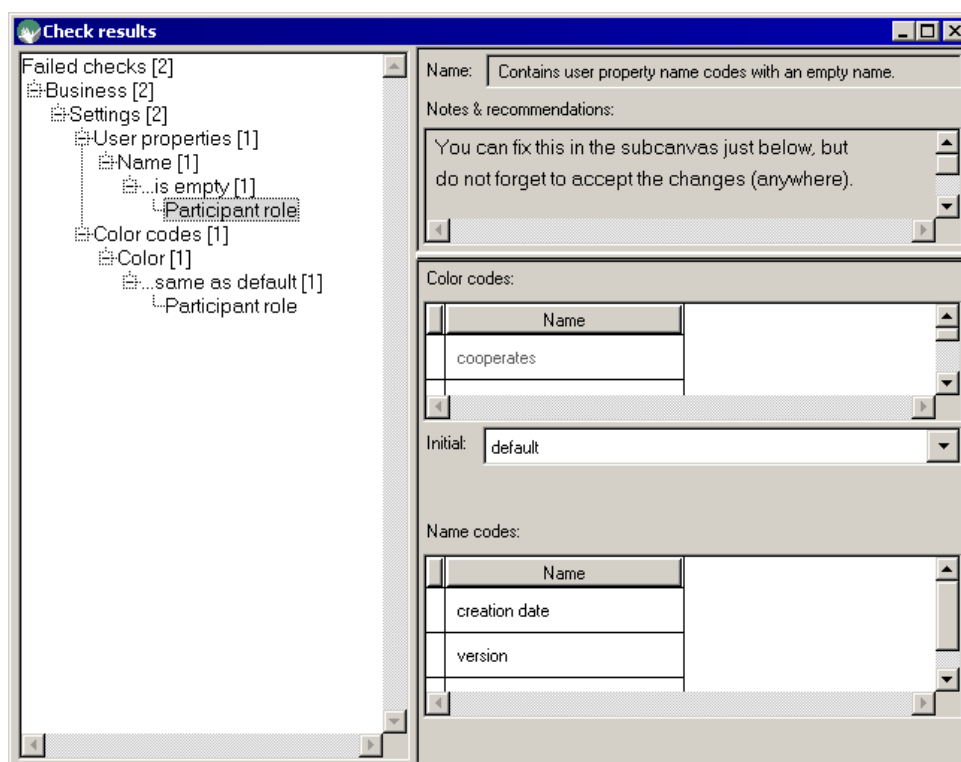


obr. 4 – příklad konceptuálního diagramu

4 Metamodel Craft.CASE

Všechny druhy diagramů, jejich elementů i dalších dat v Craft.CASE jsou navrženy podle jednoho metamodelu. Prvky všech modelů jsou nazývány „nodes“ (uzly) a vazby mezi nimi „connections“ (spojení). Každý uzel i spojení může mít různé proměnné. Uzly jsou například třídy objektů a spojení jsou například vazby skládání a dědění. Nebo podle jiného příkladu uzly jsou aktivity objektů a spojení jsou komunikace mezi aktivitami.

Craft.CASE při vytváření modelů dodržuje pravidla konkrétních uzlů a spojení a dovoluje modelovat jen takové údaje, které jsou pro příslušný typ uzlu a vazby přípustná. Kromě toho Craft.CASE obsahuje nástroj pro kontrolu konzistence a správnosti modelů, který hierarchickou formou zobrazuje nalezené nedostatky. Pro každou nalezenou chybu se zobrazí rada, jak ji lze napravit, dále editovatelné vlastnosti a pomocí menu pravého tlačítka myši lze lokalizovat nalezenou chybu přímo v diagramu:



obr. 5 – kontrola správnosti a úplnosti modelu

Diagramy lze kopírovat. Rovněž tak je možné kopírovat a vkládat objekty mezi diagramy.

Každý prvek (uzel i spojení) v databázi projektu má zajištěnou identitu nezávisle na hodnotách jeho atributů. Díky tomuto mechanismu, který je převzat z objektových databází, je možné mít například v jednom modelu dva různé objekty se stejným jménem nebo mít přehled o objektech, u kterých není ještě vyplněné jméno nebo ani ještě nejsou zakresleny v žádném diagramu.

Skupiny některých objektů (například aktivit v procesním diagramu) lze nahrazovat jediným objektem. Po takové transformaci diagramu jsou všechny vazby s okolím nahrazených původních objektů automaticky přepojeny na nový objekt.

5 XML výstup

Craft.CASE umožňuje uložit strukturu projektové databáze do XML souboru. V tomto souboru je úplná informace o všech objektech a vazbách. Tyto údaje jsou využitelné ke tvorbě generátorů

různých dalších výstupů, zdrojových kódů programovacích jazyků a nebo souborů pro přenos dat do jiných modelovacích nástrojů.

Craft.CASE využívá standard SIXX (Smalltalk Instance Exchange XML) Masashiho Umezawy. Standard SIXX patří mezi nástroje vývojového prostředí VisualWorks/Smalltalk firmy Cincom. I když je formát SIXX primárně určen pro jazyk Smalltalk, tak snadno zpracovatelný i v jiných programovacích jazycích.

Tento přístup byl zvolen záměrně. Cílem je otevřenost a rozšiřitelnost našeho nástroje. Formát výstupních dat je zveřejněn a jsou také k dispozici zdrojové kódy do XML ukládaných objektů.

6 Implementace Craft.CASE

Jak již bylo zmíněno, Craft.CASE je programován v prostředí VisualWorks v programovacím jazyce Smalltalk. V našich i evropských podmínkách tedy jde o velmi netradiční způsob implementace. S výjimkou podobného finského projektu Metaedit, který je také programován v prostředí VisualWorks/Smalltalk, jsou ostatní CASE nástroje programovány v jazyce C++. (A několik málo v jazyce Java). Obecně rozšířený názor na programování aplikací podobných naší je ten, že je potřeba použít jazyk pokud možno nižší úrovně a proto rozhodně ne Smalltalk. Tento názor vychází z přesvědčení, že abstraktnější jazyky nedokáží dostatečně efektivně zvládnout sestavení grafického editoru a obsluhu interní projektové databáze.

Naše zkušenosti však prokázaly něco jiného. První verze Craft.CASE se vyvíjela koncem roku 2004 přibližně čtyři měsíce včetně analýzy a testování. Průměrný počet vývojářů i analytiků po tuto dobu byl asi 1.5 člověka na den. To dohromady znamená pouze 6 člověkoměsíců. V případě použití jazyka nižší úrovně by to bylo jistě podstatně více.

Problémy nebyly ani s výkonností. Využili jsme knihovny pro dvojrozměrnou grafiku, práci s XML formátem a binární ukládání objektů na disk standardně dodávaných v prostředí VisualWorks. Přestože všechny tyto knihovny jsou napsané jen ve Smalltalku, tak jsme nenarazili na žádné problémy s výkonem.

Pro zajímavost ještě zmíním jednu pozoruhodnost; Craft.CASE je vytvářen v duchu zásad agilních metodik na platformě PC/Linux. Tam je postupně sestavován a testován. Základní uživatelskou platformou je ale Windows XP. Díky vlastnostem prostředí VisualWorks je tvorba instalace ostré verze pro Windows ze sestaveného programu na Linuxu otázkou několika desítek sekund.

7 Závěr

Projekt Craft.CASE je pro náš tým zajímavou a originální zkušeností. Samozřejmě, že námi zvolený přístup nemá jen samé výhody. Největším problémem, který nás tíží, je právě originalita řešení i použitého vývojového prostředí. Každý člen našeho malého týmu je v podstatě nezastupitelný, což představuje nezanedbatelnou míru rizika.

V době psaní tohoto článku se připravovala verze 1.1. Pokud vše půjde podle plánu, tak bude Craft.CASE doplněn o další diagramy v obou fázích modelování, možnost práce s šablonami, simulátor procesů a generování výstupů v různých formátech.

Jak bylo řečeno v úvodu, Craft.CASE je vyvíjen na zakázku pro mezinárodní poradenskou a konzultační firmu Deloitte. Zadání vychází ze dvou potřeb: 1) Jednoduše ovladatelný a na prostředky počítače nenáročný program. 2) Modelovací nástroj přesně šitý na míru metodě BORM, který ale také částečně konfigurovatelný, dokáže procesy simulovat a generuje výstupní

dokumentaci. V době psaní tohoto článku byly již ohlasy na jeho použití v projektech modelování procesů velké softwarové a stavební firmy. S pomocí Craft.CASE byl také zahájen projekt tvorby nové verze interní referenční příručky organizace a procesů telekomunikačních firem. Uživatelé kladně hodnotí nenáročnost programu, originalitu a jednoduchost obsluhy. Na druhou stranu je třeba přiznat, že někteří mají problémy s jeho grafickým rozhraním. Je tomu tak proto, že VisualWorks je jednotné vývojové prostředí, ve kterém lze programovat aplikace nejen pro Windows, ale i pro Unix a MacOS. Proto chování některých grafické komponent není identické s tím, jak vypadají a fungují produkty Microsoftu.

Nástroj je zdarma nabízen od listopadu 2004 českým a slovenským vysokým školám k výuce. Na naší fakultě jej studenti poprvé používali v letním semestru 2004-2005 ve výuce databází a projektování informačních systémů. Používá jej rovněž několik diplomantů, kteří se s ním naučili pracovat sami. Projekt Craft.CASE je též praktickým příkladem toho, že i v „neobvyklých“ programovacích jazycích lze udělat použitelný software. To jako učitel velmi oceňuji: Díky tomuto nástroji mají studenti možnost vidět, že principy čistého objektového programování a nerelačních objektových databází, které je učíme, jsou prakticky užitečné.

Literatura

1. Buchalceová A.: *Agilní metodiky*, sborník konference OBJEKTY 2002, ČZU Praha, ISBN 80-213-0947-4
2. Carda A., Merunka V., Polák J.: *Umění systémového návrhu - objektivě orientovaná tvorba informačních systémů pomocí původní metody BORM*. Grada, Praha 2003. ISBN 80-247-0424-2.
3. Lacko B.: *Vademekum objektivě orientované technologie*, sborník konference Programování, Ostrava 1993.
4. Merunka V.: *Objektový databázový systém Gemstone*, sborník konference OBJEKTY 2003. Ostrava 2003. ISBN 80-248-0274-0
5. Merunka V.: *Současná objektivě orientovaná vývojová prostředí založená na jazyce Smalltalk*, ve sborníku konference Tvorba softwaru 2000, Tanger Ostrava 2000, ISBN 80-85988-49-6
6. Molhanec M.: *Kritika některých chápání objektivě orientovaného paradigmatu*, Sborník 30. ročníku konference Tvorba softwaru, Ostrava 2004
7. Pícka Marek, Pergl Robert, Merunka Vojtěch: *Objektivě orientovaná tvorba softwaru*, skripta ČZU, Česká zemědělská univerzita, Praha 2004, ISBN 80-213-1159-2
8. Vírius M., Merunka V., *Unifikovaný modelovací jazyk UML I., II. a III.*, série tří článků, Chip Vogel Publishing Praha 2002, ISSN 1210-0684
9. webová stránka <http://www.cincom.com>, týkající se nástroje VisualWorks/Smalltalk.
10. webová stránka <http://www.craftcase.com>, týkající se nástroje Craft.CASE.
11. webová stránka <http://www.metacase.com>, týkající se nástroje MetaEdit.