

POUŽITÍ JAZYKA UML PŘI VÝVOJI SYSTÉMŮ PRACUJÍCÍCH V REÁLNÉM ČASE

Jan Šlechta

SOFTWARE DESIGN AGENCY, Antala Staška 1014/41, 140 00 Praha 4 - Krč, ČR
slechta.jan@seznam.cz

Abstrakt

Při vývoji software takových systémů:

- ◆ začínáme v zásadě buď na zelené louce a specifikujeme vlastnosti systému od počátku s danou mírou zkušenosti; hovoříme o empirickém vývoji software, systém můžeme modelovat jazykem UML,
- ◆ nebo používáme vlastnosti již hotových systémů a předěláváme software do jiné architektury; zde uplatníme pravidla reverzního inženýrství (reverse engineering - česky také "zpětné dokumentování"), jazykem UML můžeme tyto vlastnosti, potřebné pro "překlopení" architektury, zachytit,
- ◆ snem systémových inženýrů je samozřejmě prostředek, který by z UML notace generoval automaticky software pro danou architekturu, nebo dokonce pro různé architektury; toto umožněno zatím není, vyplývá to z vlastností jazyka UML jako takových,
- ◆ firma Software Design Agency přichází s iniciativou koncentrovaného UML, tj. CUMML (Concentrated Unified Modelling Language)

1. Použití jazyka UML

V zásadě se liší způsob práce při vývoji z nuly a při zpětné dokumentaci již existujícího systému. Problémem je zahrávat si s myšlenkou automatického generování kódu a platformy.

1.1 Empirický vývoj

Při empirickém vývoji je možno zamýšlené vlastnosti systému samozřejmě specifikovat jazykem UML. Problém se dostavuje ve chvílích, kdy co autor, to jiný zápis, tedy i jiný způsob vnímání vlastností systému. Pokud jsou zainteresovaní lidé ve vývojových týmech tolerantní je šance na úspěšné dokončení projektu (který se samozřejmě neobejde bez příslušného počtu round trips - spirálových opakování, což vyplývá z podstaty empirického vývoje).

Úspěšný výsledek empirického vývoje závisí jak na kvalitě vývojářů, tak na kvalitě znalců aplikačních oblastí, ale i na vzájemné komunikaci mezi nimi.

1.2 Zpětné dokumentování

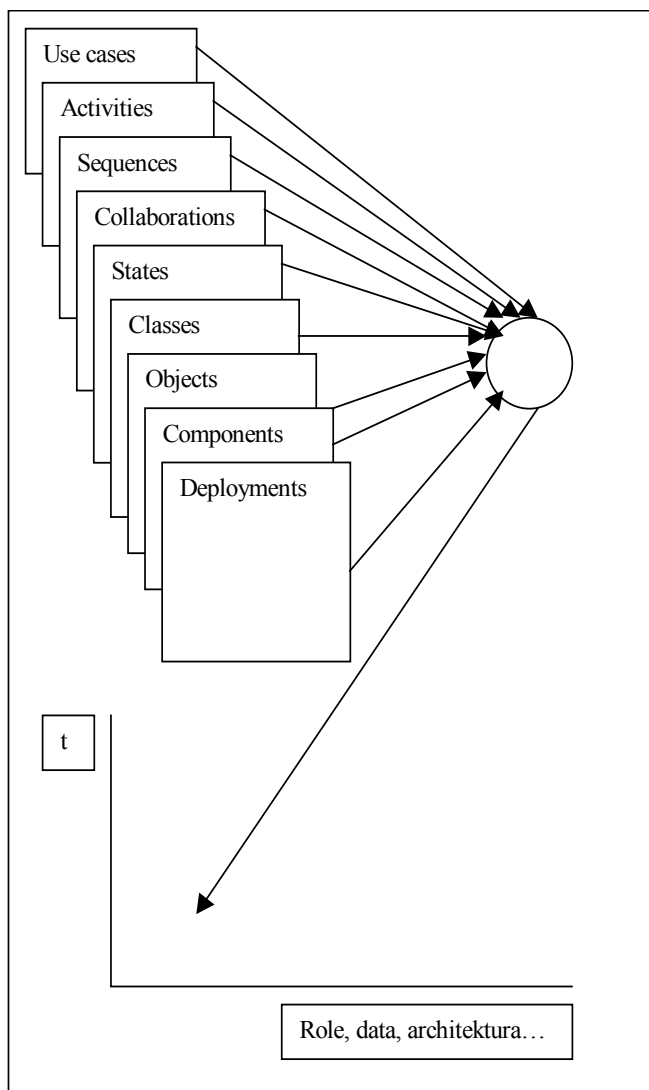
Jazyk UML může být vhodným nástrojem pro zpětné inženýrství, při konzultacích s vývojovými týmy vzniká "podniková metodika" užití jazyka UML. Ukazuje se, že vzniklé metodiky jsou adekvátní aplikacím, které týmy vyvíjejí. To je naopak zase adekvátní průmyslovému odvětví, ve kterém se vývoj děje. Různá průmyslová odvětví mají totiž své vlastní výhodné notace a architektury, které jsou standardizovány i na mezinárodní úrovni.

1.3 Automatické generování

Jazyk UML má za sebou 10 let standardizačního úsilí. Již ze 7 stanovených cílů na počátku vývoje jazyka [1] vyplývá, že nemůžeme počítat s jeho úplnou formálností a jednoznačností. Jednotlivé diagramy jsou různými pohledy na různé aspekty systému. Přitom aspekty a pohledy se překrývají, některé naopak chybí. UML je podle tvůrců otevřený systém a chybějící aspekty a pohledy lze dodefinovat [1], jak potom ale konfigurovat překládací automat ?

2. Jazyk CUML

Jazyk CUML je smělým pokusem koncentrovat pohledy a aspekty redundantně obsažené v jednotlivých diagramech jazyka UML do jediné abstraktní reprezentace; a tuto potom



přetrasformovat do formy, ve které jsou role, data, architektura aj. závislé na čase (obr. 1).

Následující aplikační příklady "Marketink cestovní kanceláře" (zkráceně MCK), "Narozeninový stroj" (NS) a "Bankomat" jsou převzaty každý z trochu jiného soudku systémů pracujících v reálném čase.

Zatímco MCK a NS jsou řízeny časem a uživatel u MCK navíc ještě úkoly; Bankomat je řízen událostmi a jeho uživatel úkoly, časem je pouze hlídán uživatel.

Doba odezvy je u prvních 2 systémů dimenzována na dny, zatímco váhající uživatel Bankomatu (stejně jako třeba telefonní síť) je penalizován návratem do původního stavu po několika vteřinách.

Zatímco u Bankomatu je architektura třívrstvá, u MCK a NS stačí pouze dvouvrstvá.

MCK na rozdíl od NS je charakterizováno pouze jednou misí rozdělenou sériově na 6 etap, zatímco NS je primárně složeno ze 2 paralelních misí zajišťujících blahopřání oslavencům na straně jedné a doplňování oslavenců na straně druhé; Bankomat je také pouze

1 mise rozdělená na etapy, ale jak již jsme řekli, řízená událostmi.

Nejsložitější, co se týká rolí je MCK s 5 rolemi (2 uživatelskými: externí Členové a interní Instruktor a 3 systémovými: Iniciátor, Plánovač a Archivátor), NS má 2 role Oslavenci a Administrátor; Bankomat zde má 3 role (uživatelské: externí Výběrčí a interní Výdejce peněz, systémové: Dispečer a Bankéři (administrují účty z banky, zde není specifikováno)).

2.1 Marketink cestovní kancelář

Psychologie systému: v 6 etapách dochází k vytříbení expedičního týmu; z počátečního rozsáhlého seznamu kandidátů se, vzájemnou interakcí CK a potenciálních zájemců v průběhu času, tvoří konečná sestava zákazníků.

Tabulka 1. Marketink cestovní kanceláře v jazyce CUMML

Etapa	Čas	Role	Transakce	Výjimky
Inicializace	T0	Iniciátor		
Plán 1	T1-7D	Plánovač	Změna T1:	
Dovednost	T1=T0+1M	Členi(0,a)	 Forname text, Surname text, Born datum, Email text; ASCENT ----- summit text, mountains text, altitude text, style text;	
Plán 2	T2-7D	Plánovač	Změna T2	
Uzávěrka a	"="	Instruktor	Vytvoření b:	
Kopie pasu	T2 =T1+3M	Členi(0,b)	 Passport jpg;	
Plán 3	T3-7D	Plánovač	Změna T3	
Uzávěrka b	"="	Instruktor	Vytvoření c	
Finance	T3 =T2+2M	Členi(0,c)	 HIRE ----- Item text; Limit varchar;	
Plán 4	T4-7D	Plánovač	Změna T4	
Uzávěrka c	"="	Instruktor	Vytvoření d	
Media	T4 =T3+3M	Členi(0,d)	 Medailon mpeg;	
Plán 5	T5-7D	Plánovač	Změna T5	
Uzávěrka d	"="	Instruktor	Vytvoření e	
Letenka	T5=T4+ 3M	Členi(0,e)	 Account varchar;	
Plán 6	T6-7D	Plánovač	Změna T6	
Uzávěrka e	"="	Instruktor	Vytvoření f	
Likvidace	T6 =T5+4M	Členi(0,f)	"="	
Plán 7	T5-7D	Plánovač	Změna T7	
Uzávěrka f	"="	Instruktor	Schválení f (stejný pattern jako vytvoření)	
Archivace	T7=T6+2M	Archivář	"="	

--	--	--	--	--

2.2 Narozeninový stroj

Psychologie systému: Postupně je tvořen seznam oslavenců tvořený e-mailovou adresou a datem narození. Jednou denně narozeninový stroj prohledá seznam oslavenců a má-li oslavenec dnes narozeniny, přijde mu e-mail s blahopřáním.

Tabulka 2.

Narozeninový stroj v jazyce CUMML - *mise* blahopřání oslavencům

Etapa	Čas	Role	Transakce	Výjimky
Inicializace				
Smyčka	Perioda T1=1D	Oslavenci(0,m)		

&

Tabulka 3.

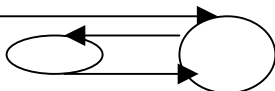
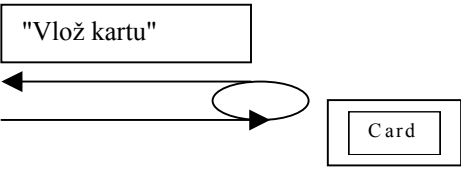
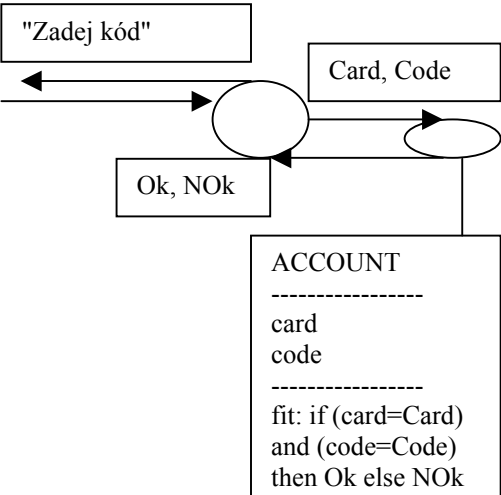
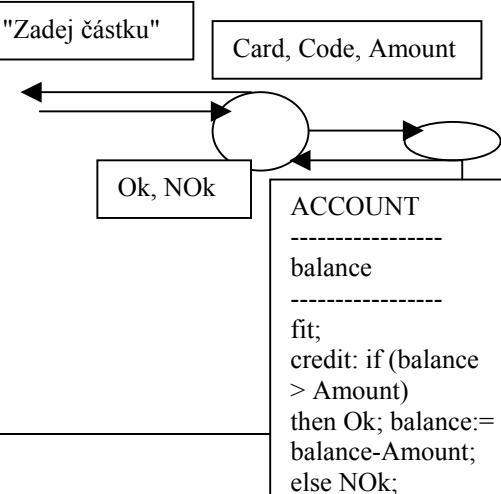
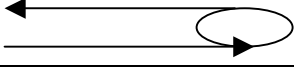
Narozeninový stroj v jazyce CUMML - *mise* doplňování oslavenců

Etapa	Čas	Role	Transakce	Výjimky
Inicializace		Administrátor		
Operace na seznamu	Hlídá T11	Administrátor		
Ukončení		Administrátor		

2.3 Bankomat

Psychologie systému: systém poskytuje uživateli možnost výběru peněz z libovolného bankomatu hierarchické distribuované sítě na základě souhlasného kódu karty a kódu PIN a dostatečném zůstatku prostředků na účtu, tj. vybíraná částka nesmí převýšit bilanci.

Tabulka 4.
Bankomat v jazyce CUML

Etapa	Čas	Role	Transakce	Výjimky
Inicializace		Dispečer		"Mimo službu": Inicializace
Karta	Hlídá T11	Výběřčí		"Otáliš T11": Karta "Špatná karta": Karta
Kód	Hlídá T12	Výběřčí		"Otáliš T12": Karta "Špatný kód": Kód "Nejsi v databázi": Karta
Částka	Hlídá T13	Výběřčí		"Otáliš T13": Karta "Chybná částka": Částka
Peníze	Hlídá T14	Výdejce		"Otáliš T14": Karta "Nejsou peníze": Karta
Návrat			Karta	

3. Komponenty pro systémy vyvíjené jako "MODEL DRIVEN"

Modely našich 3 příkladů jsou dále východiskem pro vývoj komponent platformy. Vhodnější implementací níželežících komponent pro systémy pracující v reálném čase, je použít na rozhraní komponenty dotazovací jazyky [3] než-li objektově-orientované metody [2]. To platí zejména u systémů s delší dobou odezvy (tj. MCK a NS).

I systémy s časově kritickou dobou odezvy, lze však pomocí komponent s rozhraním tvořeným dotazovacími jazyky, s úspěchem simulovat.

Literatura:

1. Page-Jones, Meilir. Základy objektově orientovaného návrhu v UML, Addison-Wesley, 2000, překlad Grada, 2001, s. 77-82
2. Greenfield-Franklin, Model Driven Development, Prentice-Hall, 2002
3. Pokorný, Jaroslav. Dotazovací jazyky, Karolinum, 2002