ZÁZNAM PROCESU TVORBY INFORMAČNÍHO SYSTÉMU

CAPTURING OF AN INFORMATION SYSTEM DEVELOPMENT

Marek Pícka

Anotace:
Tento článek pojednává o novém způsobu záznamu procesu tvorby informačního systému, který je založen na myšlence postupného vývoje a transformací mezi prvky vyvíjeného systému. Tento proces je řízem modelem transformací prvků ve zvolené metodologii. Tento model transformací prvků popisuje metodologii a umožňuje lépe pochopit vztahy a transformace mezi prvky zvolené metodologie. Instancí tohoto modelu je vlastní záznam průběhu vývojového procesu.

Klíčová slova:
Metodologie, BORM, softwarové inženýrství, transformace

Abstract:
This article is discussed a new way of capturing of an information system development, which is based on the idea of gradual development and transformation among elements of developed system. This process is driven by model of elements transformation in a selected methodology. This model of elements transformation describes methodology. It allows understanding relations and transformations among elements of the methodology.

Keywords:
Methodology, BORM, software engineering, transformation

ÚVOD

Charles Darwin ve své evoluční teorii tvrdí, že všechny druhy živých organizmů vznikly postupným vývojem z jiných druhů. Každý druh má svého předka z kterého se vyvinul – žádný druh nebyl stvořen. Vývoj druhů je zachycován takzvaným stromem života – ze společného kmene postupně raší větve nových druhů.

Při vývoji informačního systému je výhodné postupovat podobným způsobem. Každý nový prvek použitý při vývoji informačního systému musí mít své předky, z kterých se vyvinul (transformoval). Při dodržení tohoto pravidla se při vývoji informačního systému neobjeví prvky vytvořené libovůlí návrháře informačního systému. Znázorněním postupu vývoje informačního systému nebude strom (jako u živých organizmů), protože nový prvek může vzniknout transformací z více rodičovských prvků.

CÍL A METODIKA

Cílem této práce je vytvořit způsob záznamu procesu návrhu informačních systémů, který vychází z myšlenky, že každý nový prvek, který vznikne v průběhu tvorby informačního systému musí mít důvod své existence, musí mít své předky z kterých vznikne. Tento přístup by měl splňovat tyto požadavky:

- omezení chyb při tvorbě informačního systému,

- dobrou dokumentovatelnost procesu návrhu informačního systému,
- začlenění do stávajících metodik návrhu informačního systému,
- zvýšení rychlosti tvorby a údržby informačního systému.

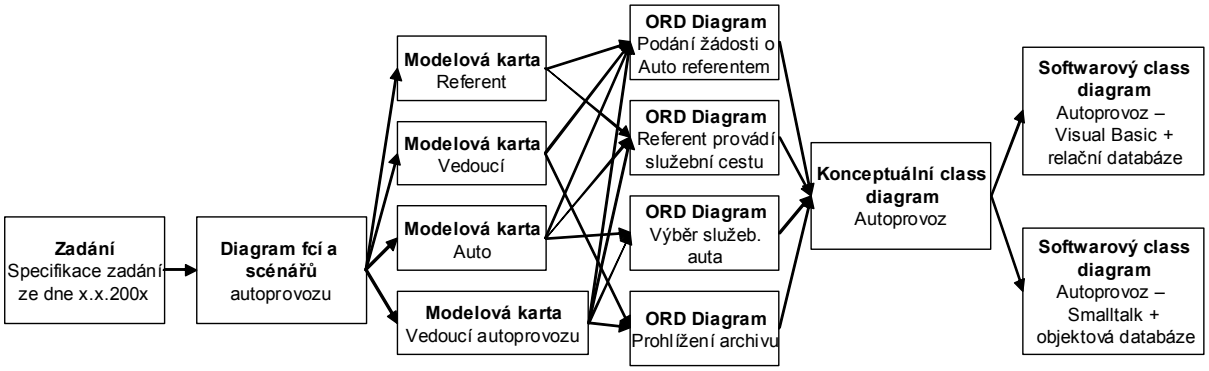
Metodou použitou v této práci je rozšířit o možnost záznamu návrhu informačního systému metodologii BORM a následně získané poznatky zobecnit pro možnost použití tohoto způsobu i v jiných metodologiích.

ZÁZNAM PROCESU TVORBY INFORMAČNÍHO SYSTÉMU

Vývojář informačního systému pracuje tak, že už existující prvky, vznikající při tvorbě informačního systému (požadavky, dokumenty, diagramy, třídy, metody atd.), transformuje pomocí pravidel a svých zkušeností v jiné prvky. Každý nový prvek vznikající při tvorbě informačního systému musí mít důvod své existence, a měl by mít svého předka (nebo více předků) z kterého vznikl. Pokud se objeví prvek bez předka, tak to bude patrně nějaký vstupní prvek (například požadavky za zadání) nebo chyba. Existence takového prvku by měla být náležitě zdůvodněna.

Pro záznam činnosti vývojáře a tím zdokumentování procesu tvorby informačního systému lze použít diagram, kde budeme zaznamenávat vytvořené prvky informačního systému a postupné transformace mezi nimi. V následujících příkladech bude použit projekt autoprovozu společnosti, která půjčuje referentům na jednotlivé pracovní cesty automobily ze svého autoparku. Pracovní cesta podléhá schválení vedoucím a konkrétní automobily přiděluje na tyto cesty vedoucí autoparku. Je použito metodologie BORM (Business Object Relation Modeling). Podrobněji o tomto projektu a BORMu v [5].

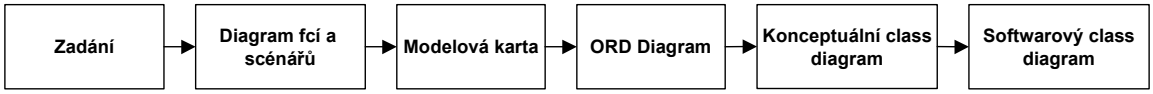
Záznam postupu tvorby informačního systému autoprovozu v metodice BORM, zachycený na úrovni diagramů a dokumentů (tj. ne moc podrobně), je na obrázku 1. Zde se postupovalo od zadávacího dokumentu, přes diagram funkcí a scénářů, z něhož jsou odvozeni modelové karty participantů (*Referent, Vedoucí, Auto, Vedoucí autoprovozu*). Participanti se účastní procesů znázorněných pomocí Object-Relation diagramů (*Podání žádosti o auto, Výběr služebního auta* atd.). Z těchto diagramů byl potom odvozen konceptuální třídní diagram autoprovozu, zachycující konceptuální objekty a vztahy mezi nimi. Z tohoto diagramu byly odvozeny softwarové třídní diagramy (pro implementaci pomocí relační a objektové databáze) a ty nakonec implementovány.



Obrázek 1 - záznam procesu tvorby autoprovozu

Pokud se na obrázek 1 podíváme z metaúrovně – tj. se nebudeme věnovat konkrétním dokumentům a diagramům vzniklým v průběhu tvorby informačního systému, ale budeme se věnovat typům dokumentů a diagramů – dostaneme pohled na metodologii (viz obrázek 2).

Tento pohled definuje transformace používané v dané metodologii a přehledně ukazuje průběh práce s touto metodologií.



Obrázek 2 - záznam metodologie BORM

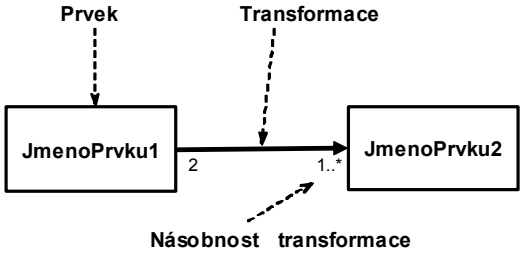
Pro podrobný záznam průběhu procesu potřebujeme pracovat s mnohem jemnějšími částmi systému než jsou dokumenty a diagramy. Potřebujeme pracovat na úrovni objektů, tříd, asociací atd.

Způsob záznamu návrhu informačního systému

Pro zachycení procesu návrhu informačního potřebujeme jednak diagram zachycující vlastní průběh procesu, tak i diagram zachycující model tohoto procesu. Každá metodologie tvorby informačních systémů má vlastní model zachycující transformaci jednotlivých prvků, s kterými daná metodologie pracuje. A každá tvorba nového informačního systému, podle dané metodologie, je novým průchodem řízeným tímto modelem, Tento průchod je zaznamenáván pomocí diagramu transformací instancí prvků.

Diagram zachycující průběh metodologie (tj. diagram transformací prvků) se skládá z těchto základních částí (viz obrázek 3):

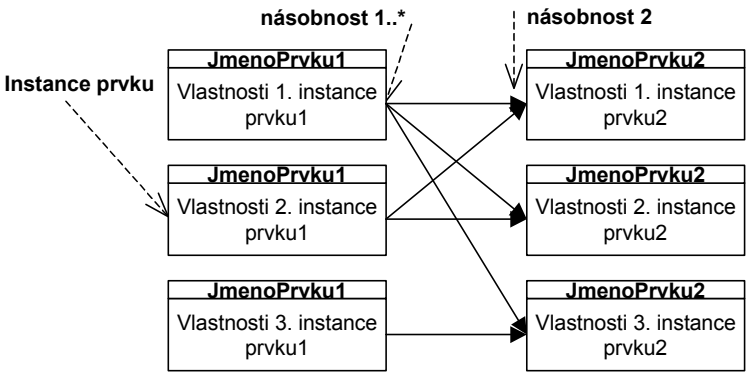
- **Prvek** – reprezentuje část systému s kterou metodologie pracuje. Například v metodologiích založených na jazyku UML (např. [1]) to jsou třídy, asociace mezi třídami, případy užitých činností (use casy), aktoři, stavy, přechody atd. Prvek je reprezentován obdélníkem se svým názvem.
- **Transformace** mezi prvky – reprezentuje přerod jednoho nebo více prvků do jiného (nebo více) prvků. Transformace je reprezentována šipkou mezi prvky. Pokud nový prvek vzniká pomocí více transformací (tj. do prvku směřuje více šipek – transformací) musí před vznikem nového prvku existovat všechny prvky rodičovské.
- **Násobnost** transformace – vyjadřuje, násobnost mezi instancemi prvků.



Obrázek 3 - model transformací prvků

Diagram zachycující průběh vlastní tvorby informačního systému (tj. diagram transformací instancí prvků) se skládá z těchto základních částí (viz obrázek 4):

- **Instance prvku –** je instancí prvku z diagramu transformací prvků. Je znázorněna obdélníkem s poli se jménem prvku a s polem zachycujícím vlastnosti instance.
- **Instance transformace** – je instancí transformace, dodržuje násobnosti zadané v diagramu transformací prvků.



Obrázek 4 - příklad možného průběhu transformace instancí prvků

Model transformací prvků na obrázku 3 vede například na diagram transformací instancí prvků zachycený na obrázku 4. Diagram transformací instancí je instancí diagramu transformací prvků (obr. 4 je instancí obr. 3). Na obrázku 4 je vidět, že odpovídá svému modelu – např. *jménoPrvku2* má přesně dva předchůdce *jmenoPrvku1* a to má 1 až n následníků.

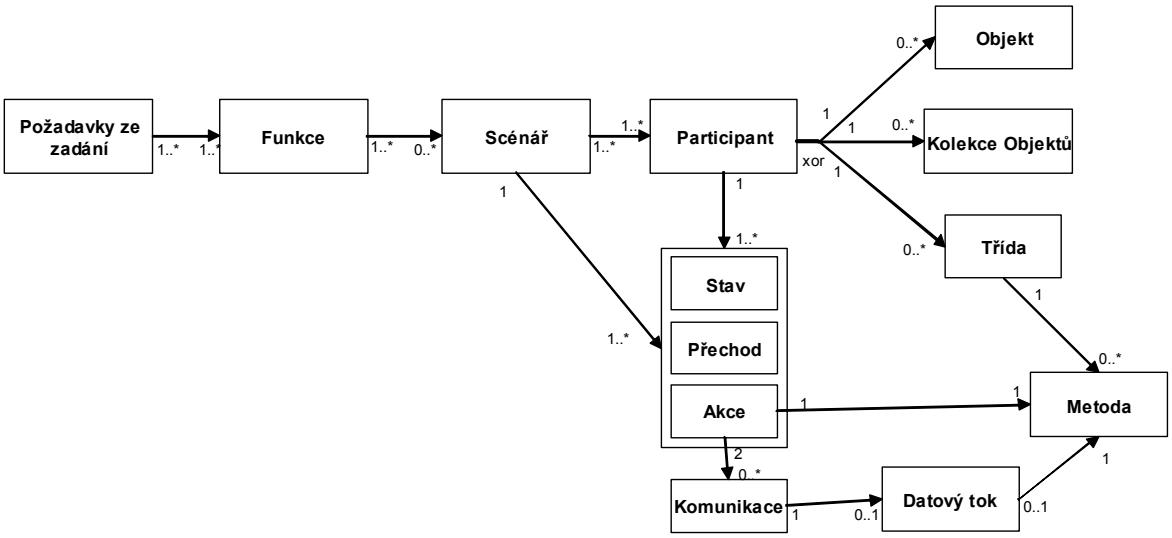
Použití záznamu tvorby informačního systému

Pokud jsou použity principy popsané v předcházejících odstavcích získáme možnost lépe řídit a kontrolovat vývojový proces. Získáme zejména tyto výhody:

- Řídit vývojový proces modelem průběhu metodologie – v každém okamžiku průběhu vývojového procesu jsme schopni určit, v co se bude transformovat prvek s kterým pracujeme. Například CASE nástroj nám může nabízet možné transformace.
- Kontrolovat průběh vývojového procesu – tím, že máme dán model metodologie, jsme schopni kontrolovat její dodržování.
- Dokumentovat průběh vývojového procesu – záznamem průběhu vývojového procesu získáme postup, jak byl informační systém vytvářen, kde budou transformace jednotlivých prvků graficky znázorněny, případně i okomentovány. Zejména nestandardní transformace by měli být vždy okomentovány.
- Snadno provádět změny – pokud provedeme změnu v již hotové části projektu, jsme schopni vysledovat všechny instance prvků, které tato změna ovlivní – budou to všechny instance prvků, jejichž předkem byla změněná instance prvku.
- Vysvětlovat metodologie – modelem transformací prvků získáme přehledný model metodologie, získáme model, z kterého je jasná geneze prvků metodologie.
- Vylepšovat metodologie – pokud se často vyskytuje nestandardní transformace (tj. třeba transformace neodpovídající zcela modelu metodologie) která je v daném případe užitečná, je na zvážení, zda nezařadit tuto transformaci do modelu metodologie.
- automatizované, nebo poloautomatizované provádění transformací – některé transformace mezi prvky lze provádět automaticky, případně poloautomaticky s malým zásahem vývojáře.

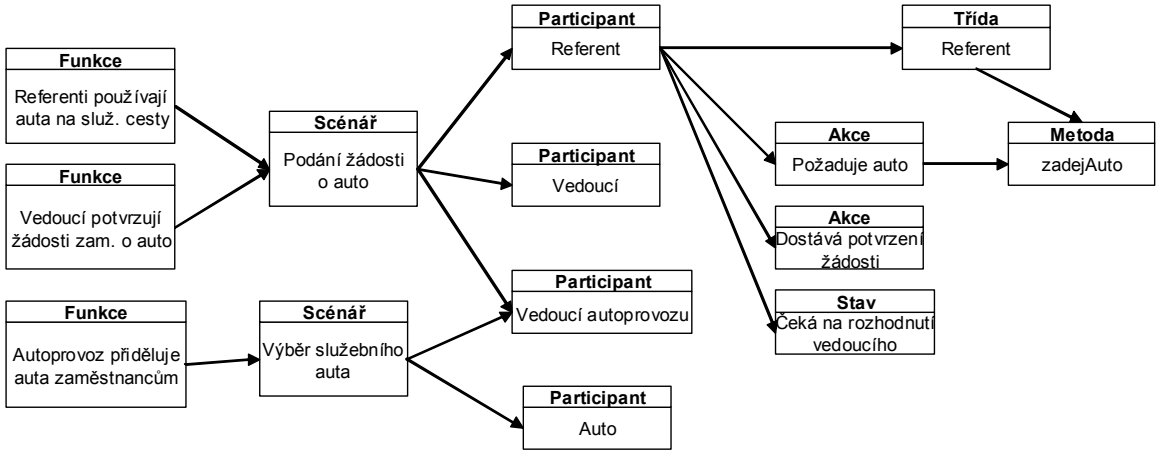
Příklad pro metodologii BORM

Pro aplikaci výše popsaných postupů je vhodná zejména objektová metodologie návrhu informačních systémů BORM, protože byla již od počátku vyvíjena s myšlenkou na postupné malé transformace jednotlivých prvků metodologie.



Obrázek 5 - ukázka modelu transformací prvků metodologie BORM

Na obrázku 5 je znázorněna část modelu transformací prvků metodologie BORM. Na obrázku je znázorněno, že prvky *Stav, Přechod* a *Akce* vznikají ve stejném okamžiku (nelze určit, který prvek vzniká dříve) ze *Scénáře* a *Participantu*. Další zajímavé transformace jsou mezi *Participantem* a *Objektem, Kolekcí Objektů* a *Třídou* – mezi těmito transformacemi je vztah xor – tj. provede se právě jedna z nich.



Obrázek 6 - ukázka průběhu projektu informačního systému autoprovozu

Na obrázku 6 je znázorněna malá část průběhu vývojového procesu informačního systému autoprovozu. Malá část je znázorněna z důvodu rozsáhlosti diagramu, pokud by měl zachycovat celý průběh – při vývoji informačního systému automobilu bylo použito 4 funkce, 5 scénářů, 5 participantů, 22 akcí, 25 stavů atd. Z těchto počtů je vidět, že diagram transformací instancí prvků je typicky velmi rozsáhlý a pokud se má používat, je nutná podpora jeho automatického vytváření pomocí softwarových nástrojů (např. CASE).

Začlenění přístupu do existujících metodologií

Tento způsob řízení a kontroly procesu vývoje informačního systému lze začlenit do existujících metodologií pomocí rozšíření jejich metamodelu o část popisující transformace použitelné v dané metodologii. Tato část by měla být samostatná a nezávislá část (a patrně i nepovinná) metamodelu, aby šlo jejím přidáním k existujícímu metamodelu získat možnost řízení a kontroly procesu vývoje informačního systému.

DISKUZE

Mezi nevýhody tohoto přístupu k zachycení průběhu tvorby informačního systému patří nutnost podpory ze strany softwarových CASE nástrojů (diagram transformací instancí prvků je typicky rozsáhlý a potřebuje podporu pro automatickou tvorbu) a omezení tvořivosti návrháře informačního systému (těsné svázání s metodikou). Omezení tvořivosti návrháře informačního systému lze také ovšem přivítat – omezí se možnost vzniku chyb.

Budoucí práce

Nejdříve je nutno prozkoumat teoretické aspekty tohoto přístupu – například vlastnosti a druhy transformací, možnosti záznamu grafů atd. Následně rozšířit metamodel metodologií BORM a UP (Unified Process) o možnosti záznamu průběhu procesu tvorby informačního systému a pokusit se o rozšíření nějakého CASE nástroje o tuto možnost.

ZÁVĚR

Tento způsob zachycení transformací prvků v metodologii umožňuje řízení a kontrolu procesu návrhu informačního systému. Zároveň podává informaci o samotné metodologii a jejím průběhu a tím přispívá k jejímu pochopení. Zachycení průběhu tvorby informačního systému (pomocí diagramů transformace instancí prvků) umožňuje dobře dokumentovat průběh tvorby systému a umožňuje snadno vyhledávat prvky systému, které jsou ovlivněny změnou v systému.

Literatura:

1. Arlow, J.; Neustadt, I.: *UML a unifikovaný proces vývoje aplikací*, Computer Press, Brno 2003, ISBN 80-7226-947-X
2. Fowler, M: *Refaktoring – zlepšení existujícího kódu*, Grada Publishing, Praha 2003, ISBN 80-247-0299-1
3. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J: *Návrh programů pomocí vzorů*, Grada Publishing, Praha 2003, ISBN 80-247-0302-5
4. Object Management Group (OMG). *OMG Unified Modeling Language Specification - Version 1.5*. 2003. http://www.omg.org.
5. Merunka, V.; Polák, J.; Carda, A.: *Umění systémového návrhu*, Grada Publishing, Praha 2003, ISBN 80-847-0424-2
6. Pícka M. Metamodel BORMu jako rozšíření metamodelu UML. *Objekty 2003. sborník konference OBJEKTY 2003*. Ostrava 2003. ISBN 80-248-0274-0.

Kontakt:

Ing. Marek Pícka, Česká zemědělská univerzita, Fakulta provozně-ekonomická, katedra informačního inženýrství, Kamýcká 129, 165 21 Praha 6-Suchdol, email: picka@pef.czu.cz