METODIKA UWE (UML based Web Engineering)

Martin Molhanec

České vysoké učení technické – FEL, K-313
Technická 2, 166 27 PRAHA 6, Dejvice, Česká republika
tel.: (++420) 2 2435 2118
mailto: molhanec@fel.cvut.cz
http://martin.feld.cvut.cz/~molhanec

Abstrakt

Obsahem příspěvku je podrobný popis a stručné zhodnocení metodiky UWE (UML based Web Engineering) určené pro podporu tvorby webových sídel. Jedná se o velice aktuální problematiku, protože tvorba webových sídel je současným trendem v oblasti tvorby informačních systémů.

1 ÚVOD

Tento příspěvek navazuje na moje předchozí příspěvky na této konferenci v předešlých letech [7], [9], [10] a [12] a na konferenci Objekty [8] a [11]. Nechci opět opakovat to, co již bylo ve výše zmíněných příspěvcích napsáno. Proto jenom stručně. Webové metodiky jsou speciálním druhem metodik, které jsou primárně určené pro analýzu a návrh webových sídel a webových aplikací. Webové metodiky vznikly na základě o něco starších metodik pro návrh hypermediálních a hypertextových aplikací. Jejich společným základem jsou pochopitelně metodiky pro analýzu a návrh informačních systémů obecně. Znalost těchto webově orientovaných metodik je ČR velice minimální, ale podobně je tomu i v dalších zemích. Tato skutečnost je v rozporu se skutečností, že webové aplikace jsou v současné době na vrcholu zájmu všech softwarových vývojářů, uživatelů a obchodníků.

2 UML-BASED WEB ENGINEERING (UWE)

UML-based Web Engineering je moderní webová metodika, která vznikla na základě těch nejlepších zkušeností s předcházejícími metodikami. Autoři se jmenovitě zmiňují o návaznosti na metodiky RMM, OOHDM, WSDM a některé další již dříve navržené metody a postupy. Metodika UWE byla poprvé jako celek prezentována v doktorské práci Nory Koch [1] v roce 2000. Nicméně samotné doktorské práci předchází několik příspěvků na různých konferencích koncem 90 let, které jsou jednotlivými stupni k jejímu vytvoření, například práce [4].

UWE podporuje vývoj webové aplikace od počátečního sběru požadavků až po návrh implementace. Autoři se také ve svých posledních pracích pokoušejí o automatizaci implementace webové aplikace na základě předchozího návrhu aplikace metodikou UWE [2] a [3]. Speciální zřetel je autory metodiky brán zejména na systematičnost celého návrhu. Jedním z nejdůležitějších znaků metodiky UWE je fakt, že jako svoji notaci využívá notaci *Unified Modeling Language* (UML). Pro toto rozhodnutí vedla autory snaha využít pro metodiku UWE dnes již vlastně průmyslový standard modelovacího jazyka UML, který je v současné době dominantním objektově orientovaným modelovacím nástrojem a notací, těší se všeobecné znalosti a oblibě a v neposlední řadě existuje pro UML celá řada komerčních

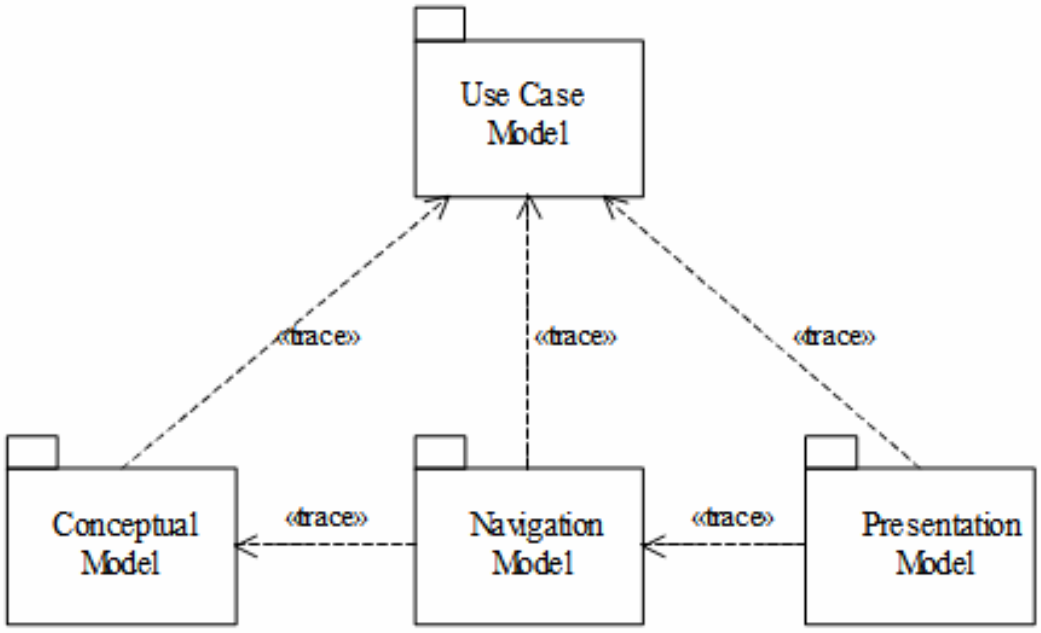
i nekomerčních modelovacích nástrojů. Podle autorů UWE metodiky jsou hlavní rysy jejich řešení tyto:

- použití standardní notace UML ve všech jejich modelech,
- přesná definice metodiky, tj. přesný a podrobný popis způsobu vytváření všech jednotlivých modelů a
- specifikace omezení (*constraints*), které zvyšují přesnost jednotlivých modelů.

Vlastní vývojový proces metodiky UWE se skládá ze 4 kroků:

- analýzy požadavků (*requirement analysis*), jejíž výsledkem je model užití (*use case model*),
- konceptuálního návrhu (*conceptual design*), jehož výsledkem je konceptuální model (*conceptual model*),
- návrhu navigace (*navigation design*), jehož výsledkem je model navigačního prostoru a struktury (*navigation space and structure model*) a
- návrhu prezentace (*presentation design*), jeho výsledkem je prezentační model (*presentational model*).

Vzájemné souvislosti mezi výše uvedenými modely jsou zachyceny na [Obr. 1]. Je patrné, že jednotlivé modely jsou na sobě závislé a na modelu UC (Use Case) jsou závislé všechny modely zbývající. Všemi výše uvedenými kroky metodiky UWE se budeme nyní podrobněji zabývat.



Obr. 1: Vzájemná souvislost mezi UWE modely

2.1 Analýza požadavků (Requirement Analysis)

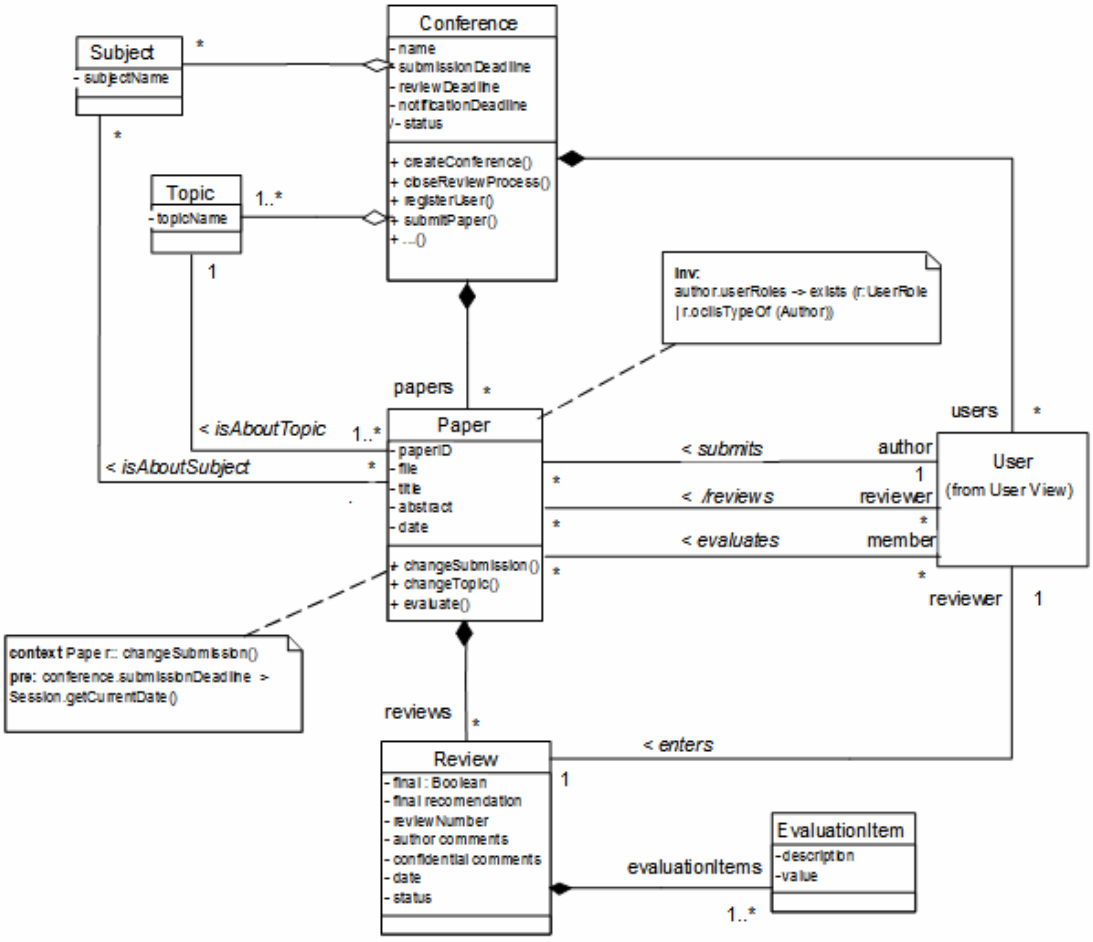
První krok metodiky UWE – analýza požadavků je v souladu s metodikou UP (Unified Process) s některými malými upřesněními. Jádrem metody je nalezení relevantních účastníků (Actors) a k nim se vztahujících případů užití (Use Cases). Analýza požadavků odpovídá na

otázku – kdo systém používá a jaké od něho vyžaduje funkce. Kroky vedoucí k vytvoření správného modelu UC jsou dle autorů UWE následující.

1. Nalezení účastníků (*actors*).
2. Nalezení činností (*activities*), které jednotliví účastníci vykonávají.
3. Seskupování jednotlivých činností do případů užití.
4. Stanovení vztahů mezi účastníky a případy užití.
5. Stanovení vztahů zahrnutí a rozšíření mezi jednotlivými případy užití.
6. Zjednodušení modelu využitím dědičnosti mezi účastníky a případy užití.

2.2 Konceptuální návrh (Conceptual Design)

Konceptuální analýza v metodice UWE vychází z osvědčených metod ER (Entity-Relationship) či OO (Object Oriented) datového modelování. Zájmem konceptuální analýzy jsou objekty potřebné pro vzájemnou komunikaci účastníků a systému. Tato komunikace je vyjádřena prostřednictvím případů užití, proto je konceptuální analýza závislá na analýze požadavků. Základními pojmy konceptuálního modelu v souladu s UML jsou: třída (class), vztah (association) a balíček (package). Konceptuální model používaný autory UWE využívá všech vymožeností UML, jako je například možnost definování odvozených (derived) atributů a vztahů nebo definování omezení v jazyce OCL. Příklad jednoho pohledu na konceptuální model je na [Obr. 2].



Obr. 2: Příklad částečného konceptuálního modelu

2.3 Navigační návrh (Navigation Design)

Zatímco první dva kroky metodiky UWE analýza požadavků a konceptuální návrh se nikterak významně neliší od jim odpovídajících kroků v nikoliv webových metodikách, jsou následující dva kroky – navigační a prezentační návrh typické právě a pouze pro webové a jim předcházející hypermediální metodiky!

Dle autorů metodiky UWE jsou obsahem navigačního návrhu dva postupně vytvářené a spolu souvisící modely: Model navigačního prostoru (Navigation Space Model) a Model navigační struktury (Navigation Structure Model). První model specifikuje které objekty mohou být při dané navigaci ve webové aplikaci navštíveny. Druhý model specifikuje jaký způsobem výše uvedené objekty mohou být při dané navigaci dosaženy.

2.3.1 Model navigačního prostoru (Navigation Space Model)

Model navigačního prostoru se vytváří na základě již získaného konceptuálního modelu. Základem tohoto modelu jsou dva modelovací konstrukty: navigační třída a přímá navigace. V terminologii webových sídel odpovídají pojmům webová stránka (web page) a odkaz (link).

- **Navigační třída (navigation class)**

  Navigační třída modeluje takové třídy, jejichž instance mohou být při pohybu po webovém sídle uživatelem navštíveny. Navigační třída se odvozuje od třídy konceptuální a má stejné jméno. Zjednodušeně řečeno: takové třídy konceptuálního modelu, které se při pohybu po webovém sídle stávají základní třídou nad kterou je navržena webová stránka, takové třídy jsou navigačními třídami. Pro reprezentaci navigačních tříd v UML je použit mechanismus stereotypů. Navigační třídě pak odpovídá stereotyp <<navigation class>>.

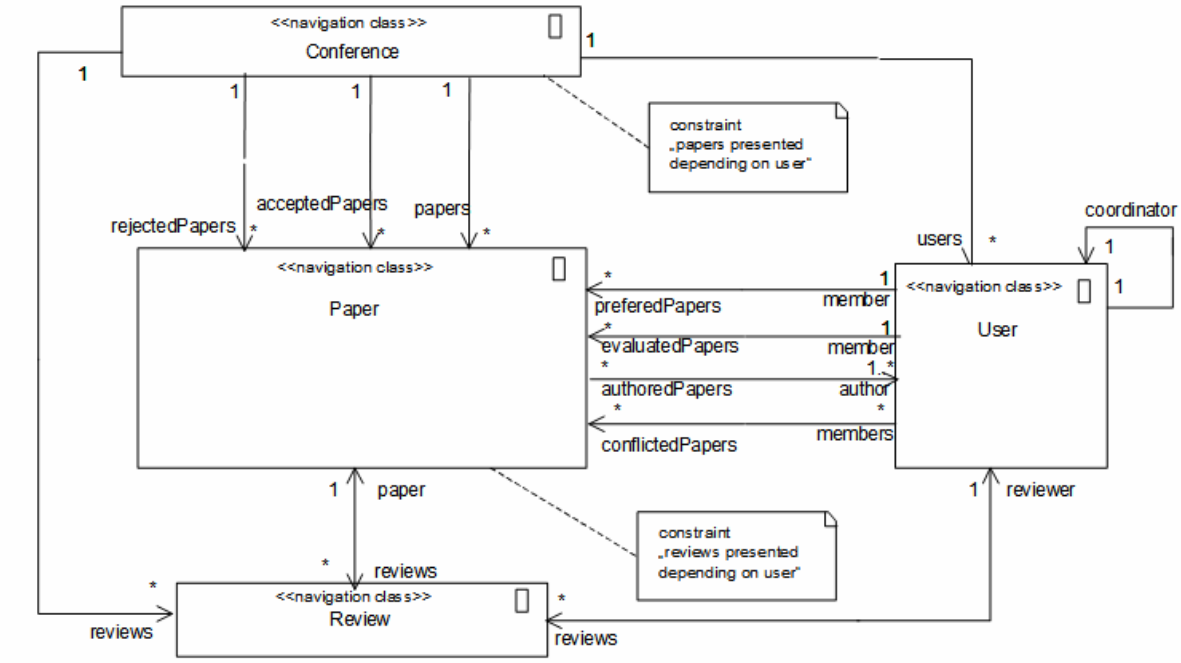
- **Přímá navigace (direct navigability)**

  Vztahy (associations) v modelu navigačního prostoru jsou chápány jako přímé navigace (přímé přechody) od počáteční (zdrojové) navigační třídy (source class) k cílové (koncové) navigační třídě (target class). Vztahy v modelu navigačního prostoru jsou orientované (jednosměrně nebo obousměrně). V UML diagramu je orientace vztahů přímé navigace znázorněna šipkou (arrow) na jednom či obou koncích vztahu. Navíc je každý konec vztahu (na straně šipky) doplněn jménem role (role name) a násobností.

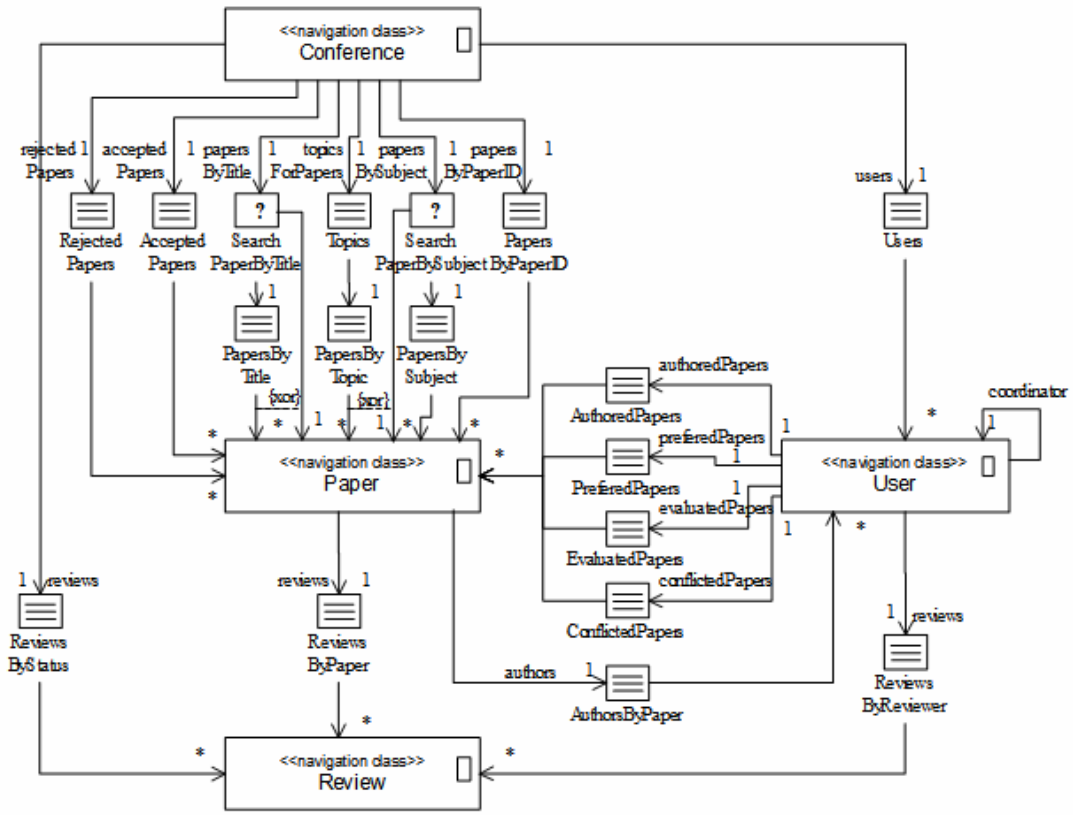
Podobně jako v případě konceptuálních modelů si můžeme pro lepší přehlednost kreslit model navigačního prostoru po menších částech. Ukázka takovéhoto částečného diagramu modelu navigačního prostoru pro uživatele účastník (member) z příkladu v tomto textu používaném od autorů metodiky UWE je na [Obr. 3].

2.3.2 Model navigační struktury (Navigation Structure Model)

Model navigační struktury vychází z modelu a zjemňuje model navigačního prostoru. Jestliže model navigačního prostoru specifikuje, které navigační třídy se účastní navigace po webovém sídle (webové aplikaci), model navigační struktury navíc definuje jakým způsobem se uskutečňuje navigace mezi jednotlivými navigačními třídami. Za tímto účelem autoři metodiky UWE zavádějí další konstrukt: přístupový element (access element). Autoři proto definují několik základních přístupových elementů: index, průvodce, dotaz a menu, jež budou podrobněji popsány v dalším textu. Pro jejich realizaci v UML je opět využito mechanizmu stereotypů [2]. Model navigační struktury se vytváří postupně ve dvou krocích transformací (doplněním, rozšířením) modelu navigačního prostoru. Příklad komplexního diagramu navigační struktury transformovaného z diagramu navigačního prostoru [Obr. 3] doplněním indexů, průvodců a dotazů je na [Obr. 4].



Obr. 3: Příklad modelu navigačního prostoru

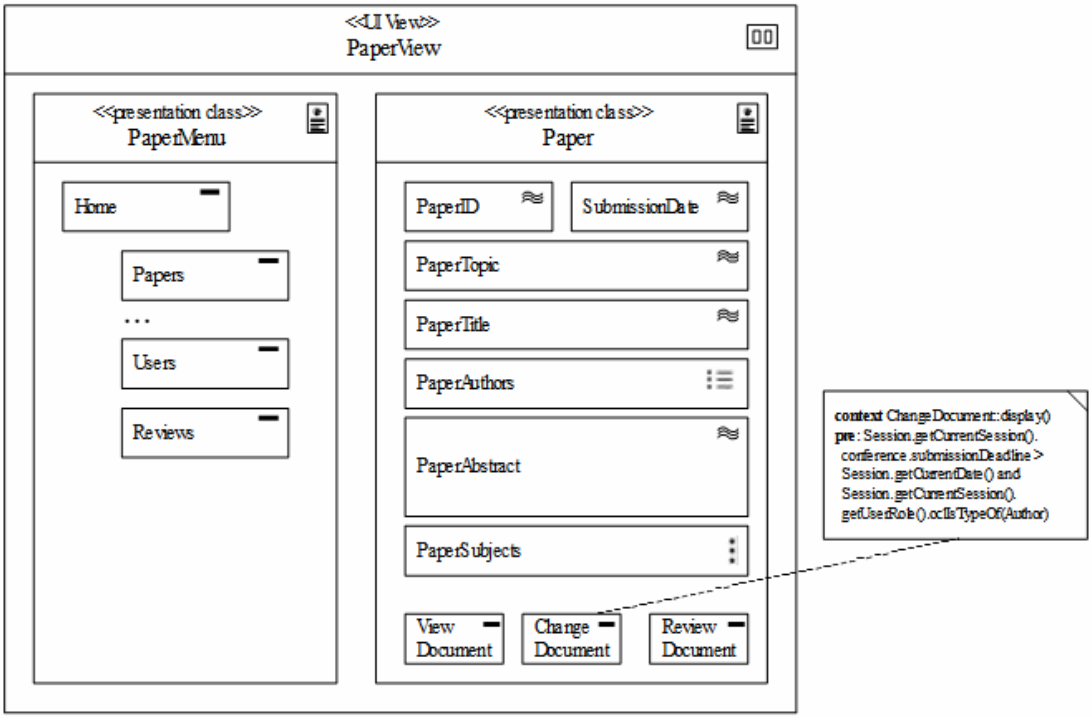


Obr. 4: Příklad modelu navigační struktury

2.4 Návrh prezentace (Presentation Design)

Posledním krokem metodiky UWE je návrh prezentace. I zde se autoři metodiky UWE snaží o co nejpřesnější a nejsystematičtější popis činnosti návrháře, tak aby bylo dosaženo vytčeného cíle. Prezentační model se skládá z množiny pohledů (views), které zobrazují obsah a strukturu jednoho navigačního uzlu (instance navigační třídy, webové stránky) a které popisují, jakým způsobem s nimi uživatel komunikuje. Autoři metodiky UWE nabízejí tři základní metody pro prezentační návrh:

- Náčrtky (*sketches*)
- Prezentační tabule (*storyboards*)
- Model prezentační toku (*presentation flow model*)



Obr. 5: Náčrtek

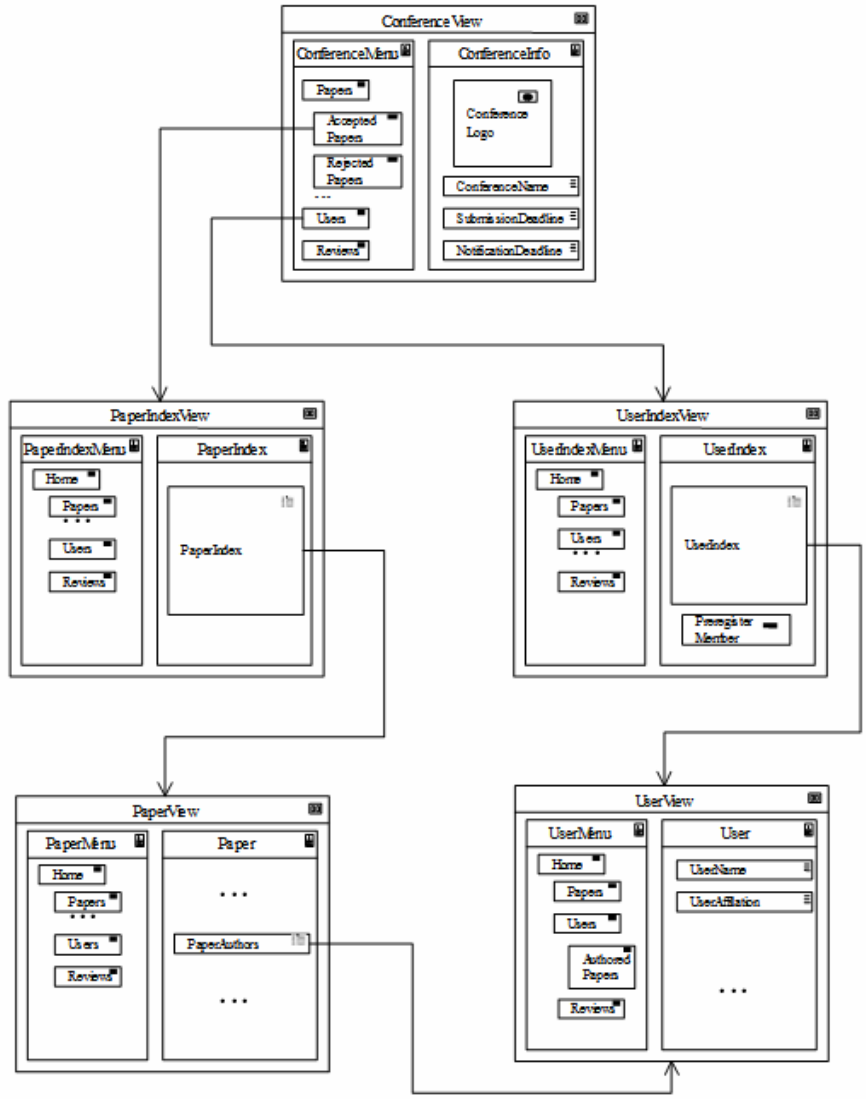
2.4.1 Náčrtky (sketches)

Náčrtkem autoři metodiky UWE rozumí grafické znázornění vzhledu uživatelského rozhraní určitého navigačního uzlu. Pro každý navigační uzel se vytváří jeden náčrtek. Pro každý atribut navigační třídy se vytváří jeden element uživatelského rozhraní. Pro názorné grafické znázornění náčrtků autoři metodiky UWE volí alternativní notaci UML [4]. Při této alternativně se místo obvyklého symbolu pro kompozici (composition) černě vyplněného malého kosočtverce (black diamond symbol) volí notace do sebe navzájem vnořených tříd, kdy celek je vnější třídou a část je vnitřní třídou. Toto alternativní grafické zobrazení lépe vystihuje vzájemné vnořování skutečných grafických elementů do sebe.

Kompletní náčrtek vzhledu uživatelského rozhraní pro uzel příspěvek (paper) ze zde uváděného příkladu je na [Obr. 5]. Všimněme si, že grafické zobrazení je i přes naše poměrně stručné seznámení se s metodou, dobře srozumitelné člověku obeznámenému s UML. Uživatelské rozhraní uzlu příspěvek je tvořeno dvěma prezentačními třídami, z nichž jedna (na obrázku vlevo) představuje implementaci menu a druhá (na obrázku vpravo) je částí webové stránky obsahující vlastní informace o příspěvku. Obě prezentační třídy obsahují pak řadu různých elementů uživatelského rozhraní.

2.4.2 Prezentační tabule (storyboards)

Model prezentační tabule vychází z navigačního modelu. Každý uzel navigačního modelu je však nahrazen jemu příslušejícím náčrtkem. Výsledkem je pak diagram, který kombinuje informace obsažené, jak v modelu navigačních tříd, tak v modelu náčrtků. Obrázek kompletního diagramu prezentační tabule odpovídající části diagramu [Obr. 6] je na [Obr. 6].



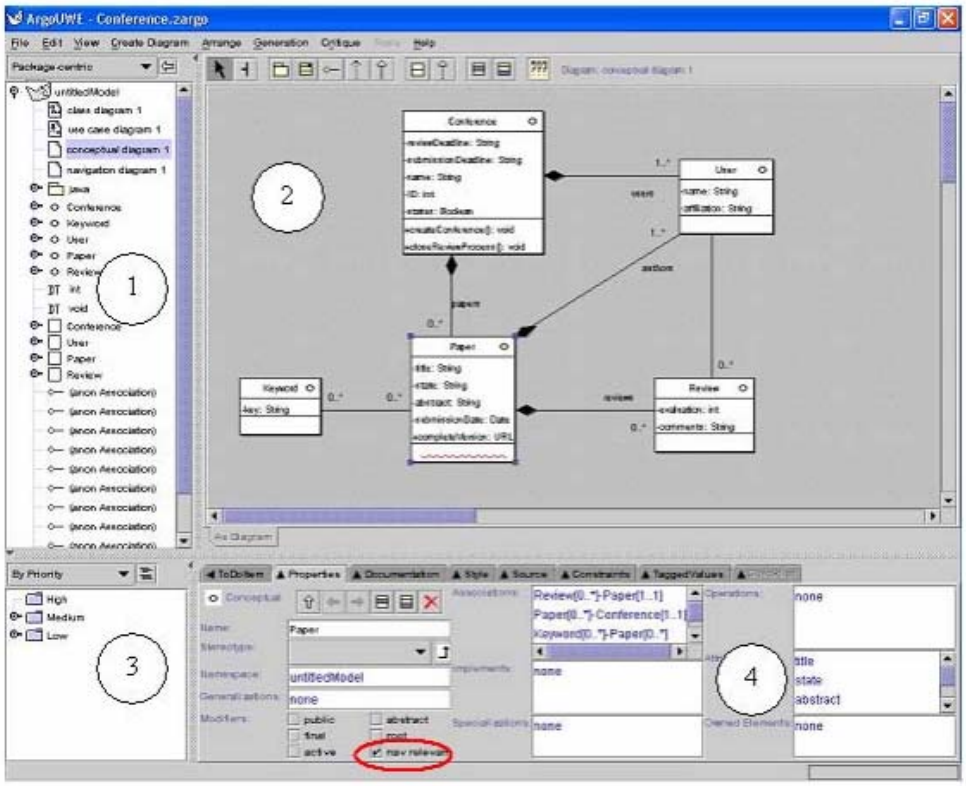
Obr. 6: Prezentační tabule

2.5 Podpora metodiky UWE v oblasti analýzy a návrhu

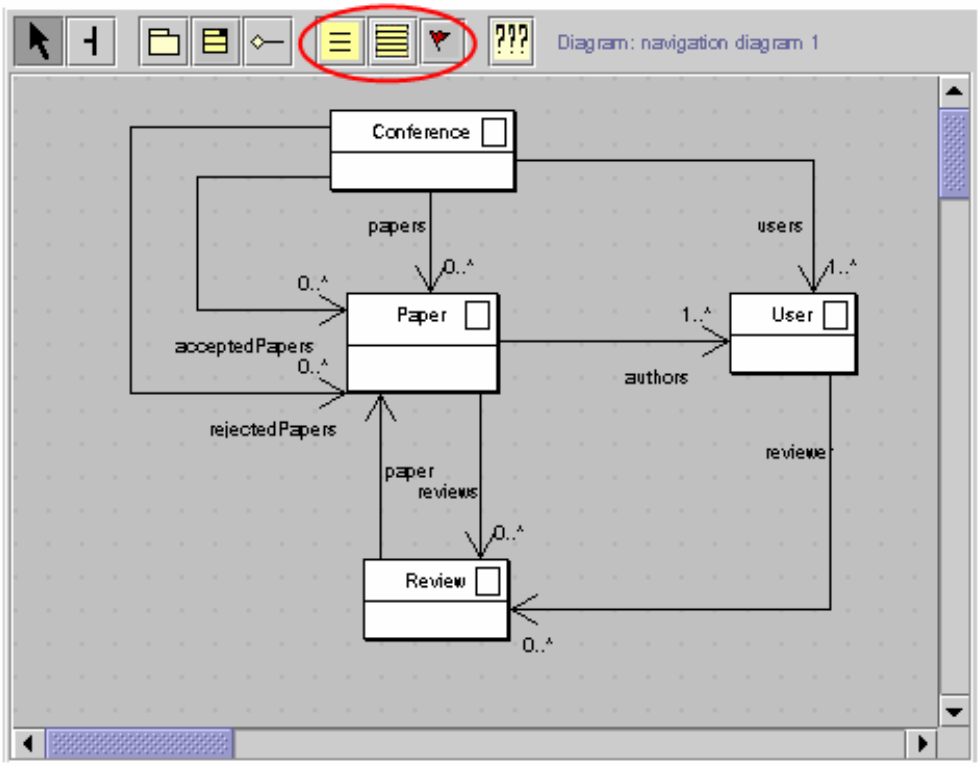
Vzhledem ke skutečnosti, že metodika UWE je založena na rozšiřování UML je nasnadě, že základním nástrojem na podporu metodiky jsou nástroje typu CASE určené pro UML. Metodika UWE využívá mechanismus rozšiřování UML pomocí tzv. profilů (lightweight UML profile). Obohacení UML o nové typy diagramů (navigační a prezentační) je dosaženo využíváním mechanismu stereotypů. Je tedy nutné, aby případný UML CASE nástroj uměl nejen pouhou notaci diagramů UML, ale byl schopen svého rozšiřování (extensions) v souladu s právě uvedenými mechanismy.

Nástroj, který pro podporu své metodiky využívají její autoři se jmenuje ArgoUWE [5] a je implementován jako zásuvný modul (plug-in) do otevřeného (open-source) modelovacího nástroje ArgoUML [6]. Jak zásuvný modul, tak samotný modelovací nástroj jsou napsány v jazyce Java. Rozšiřování produktu ArgoUML je zajištěno prostřednictvím knihovny UML

metamodelu Novosoft UML library (NSUML) a základního grafického editovacího prostředí (General Graph Editing Framework, GEF). Ukázky prostředí nástroje ArgoUML/ArgoUWE jsou na [Obr. 7] a [Obr. 8].



Obr. 7: ArgoUWE - konceptuální model



Obr. 8: ArgoUWE - model navigačního prostoru

2.6 Podpora metodiky UWE v oblasti implementace a realizace

Podpora metodiky UWE v oblasti implementace a realizace spočívá podobně, jako u jiných metodik, v podpoře generování zdrojového kódu aplikace. V případě webových aplikací jde především o generování kódu HTML webových stránek aplikace a generování kódu v jazyce vhodném pro programování dynamických webů (.php, .asp, .jsp, atp.). Automatické generování webové aplikace je v metodice UWE založeno na využití XML pro ukládání jednotlivých modelů metodiky UWE, na postupné transformaci těchto modelů a na výsledném generování webové aplikace pomocí na XML založeném publikačním frameworku. Tento nástroj je realizován nad populárním javovským frameworkem s názvem Cocoon, který je součástí otevřené množiny projektů projektu Apache.

3 ZÁVĚR

Metodika UWE je jedna z moderní metodik pro návrh webových aplikací/sídel. Je postavena na několika zásadách:

- Objektově orientovaný návrh
- Kompatibilita se standardem UML
- Speciální modely určené pro navigaci a prezentaci.
- Velice pečlivě a přesně popsané postupy.
- Snaha o dotažení metodiky do formy, která pokrývá celý životní cyklus projektu.
- Snaha o vytvoření nástrojů pro podporu analýzy, návrhu a implementace projektu.
- Navazuje na mnohé starší metodiky - především RMM a OOHDM.

Hlavní výhodou této metodiky vidím jednak ve využití notace kompatibilní s UML. Tato skutečnost může podpořit uplatnění metodiky i s jiným vývojovým nástrojem nežli s ArgoUWE/ArgoUML a tím podpořit její rozšíření i tam, kde vývojové a realizační týmy využívají již jiné UML CASE nástroje. Další předností metodiky je její přesný popis, který poskytuje návrháři opravdu dobré pokyny, jak postupovat při vytváření jednotlivých modelů metodiky.

Jaká bude další perspektiva této metodiky? V současné době se autorský tým věnuje především dotažení metodiky v oblasti realizace nástrojů pro její podporu (projekty ArgoUWE a UWEXML). Domnívám se, že metodika UWE je jedním z kandidátů, aby byla přijata některou velkou IT firmou (Microsoft, IBM,...) za základ jejího firemního nástroje orientovaného na vývoj webových aplikací/sídel a tím se jí dostalo masivního rozšíření.

4 LITERATURA

[1] Koch N. (2000). *Software Engineering for Adaptive Hypermedia Applications*. PhD. Thesis.

[2] Alexander Knapp, Nora Koch, Flavia Moser and Gefei Zhang. ArgoUWE: A CASE Tool for Web Applications. In First International Workshop on Engineering Methods to Support Information Systems Evolution (EMSISE03) held in conjunction with OOIS03, 14 pages, online publication, September 2003.

[3] Andreas Kraus and Nora Koch. Generation of Web Applications from UML Models using an XML Publishing Framework. In 6th World Conference on Integrated Design and Process Technology (IDPT), June 2002.

[4] Nora Koch, Andreas Kraus, and Rolf Hennicker. The Authoring Process of the UML-based Web Engineering Approach. In Daniel Schwabe, editor, First International Workshop on Web-oriented Software Technology (IWWOST01), online publication, 29 pages, June 2001.

[5] ArgoUWE- CASE Tool for Modeling Web Applications. On line: http://www.pst.informatik.uni-muenchen.de\projekte\uwe\argouwe.shtml

[6] ArgoUML – free UML modelling tool. On line: http://argouml.tigris.org

[7] Molhanec Martin: „Metodiky orientované na tvorbu webových sídel“, Tvorba software 2004, TANGER, Ostrava 2004

[8] Martin Molhanec: „WebML – Objektově orientovaná metodika pro tvorbu webových sídel“, Objekty 2003, VŠB-TU, Ostrava 2003. On line: http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2003/WebML-CO.pdf

[9] Martin Molhanec: „Metodologie orientované na tvorbu webových sídel“, Tvorba software 2003, TANGER, Ostrava 2003. On line: http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2003/WebMet-co.pdf

[10] Molhanec Martin: „Metodologie OOHDM, jazyk Lua a tvorba webových aplikací“, Tvorba software 2002, TANGER, Ostrava 2002. On line: http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2002/Lua.pdf

[11] Molhanec Martin: „The Object-Oriented Hypermedia Design Model (OOHDM) “, Objekty 2001, Česká zemědělská universita, Praha 2001. On line: http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2001/OOHDM.pdf

[12] Molhanec Martin: „Tvorba webových sídel jako inženýrský úkol“, Tvorba software 2001, TANGER, Ostrava 2001. On line: http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2001/metodika.pdf

V Praze, 25. března 2005
Martin Molhanec