

Abstrakt

Bakalárska práca je zameraná na objektovú analýzu a návrh webových aplikácií s využitím metodiky WebML. Jej cieľom je zhodnotiť, aké možnosti ponúkajú v súčasnosti metodiky zamerané na návrh hypermediálnych aplikácií a jednu z nich (prípadne mierne modifikovanú) odporučiť spoločnosti ComAp, s.r.o. Výstupom práce bude krátky dokument obsahujúci návrh zavedenia metodiky spolu s odporúčaním nástroja, v ktorom by bolo možné podľa danej metodiky modelovať.

V prvej časti práce popisujem metodiku WebML a jej jednotlivé modely. V ďalšej kapitole stručne popisujem iné metodiky na návrh webovo orientovaných aplikácií a porovnam ich podľa určitých kritérií. V tretej kapitole porovnávam nástroje WebRatio a Enterprise Architect - jeden z nich bude odporúčaný do spoločnosti ComAp ako nástroj vhodný na modelovanie. Stručne popíšem súčasný stav návrhu aplikácií v ComApe a zhrniem požiadavky na novú metodiku. Metodiku WebML predvediem na praktickom príklade z ComApu v piatej kapitole a demonštrujem určité modely z WebRatia ako aj z Enterprise Architectu.

Na záver predstavím metodiku návrhu aplikácií, o ktorej si myslím, že by bola pre ComAp prínosná a poukážem na pozitíva oproti metodike WebML. Dúfam, že vedenie spoločnosti ComAp prijme túto metodiku a začneme podľa nej navrhovať systémy.

Abstract

This work is focused on the object analysis and design of web applications using WebML. Its aim is to review the possibilities that various web-based methodologies offer. The output of this work is a short document for ComAp, Ltd., containing a recommendation of a web-based methodology and a tool supporting it.

First, I describe WebML and its models - conceptual, hypertextual and presentation models. Second, I briefly describe other methodology used for web application design and I compare them. In the third chapter, I compare WebRatio and Enterprise Architect which are tools used for modelling conceptual and other diagrams. One of them will be recommended for ComAp as a great tool used for modelling. I briefly describe present development process in ComAp and I compose requirements of a new methodology. I will present WebML design process on a real project from ComAp.

At the end, I introduce a methodology for web application design which should be lead into ComAp. I hope, that ComAp staff will like this methodology and we will use it for design web-based applications.

Obsah

Obsah	3
Úvod	5
1 Popis metodiky WebML	7
1.1 Konceptuálny model	9
1.2 Hypertextový model	10
1.2.1 Kompozičný model	11
1.2.2 Navigačný model	13
1.3 Prezentačný model	13
1.4 Implementačné nástroje	14
1.5 Modelovanie veľkých systémov	14
1.6 Nedostatky metodiky WebML	14
2 Metodiky podobne WebML	15
2.1 RMM	15
2.2 EORM	16
2.3 HDM	16
2.4 OOHDM	17
2.5 SOHDM	18
2.6 VHDM	19
2.7 eW3DT	19
2.8 WSDM	19
2.9 UWE	20
2.10 Porovnanie s WebML	20
3 Nástroje WebRatio a Enterprise Architect	22
3.1 WebRatio	22
3.1.1 Popis	22
3.1.2 Výhody a nevýhody	24

3.1.3	Hodnotenie	25
3.2	Enterprise Architect	25
3.2.1	Popis	25
3.2.2	Výhody a nevýhody	27
3.2.3	Hodnotenie	28
4	Vývoj v spoločnosti ComAp, s.r.o.	28
4.1	Súčasný stav	28
4.2	Vízia budúceho stavu	29
4.3	Ako to dosiahnuť?	30
5	Praktický príklad	31
6	Odporúčaná metodika pre ComAp, s.r.o.	33
6.1	Modelovací nástroj	33
6.2	Modely	33
6.3	Čo potrebujeme?	35
6.4	Prínos z WSDM	36
6.5	Zhrnutie	36
	Záver	37
	Reference	38
	Terminologický slovník	42
	Prílohy	44

Úvod

Webovo orientované informačné systémy vo väčšine prípadov slúžia na prezentáciu a aktualizáciu veľkých objemov dát. Webovými aplikáciami rozumieme napríklad:

- Komerčne orientované aplikácie, ako napríklad elektronické katalógy, aukcie, virtuálne tržiská.
- Obsahovo orientované aplikácie, napríklad on-line noviny, digitálne knižnice.
- Aplikácie orientované na služby, napríklad rezervačné systémy, turisticko - informačné systémy.
- Komunikačne orientované aplikácie - portály a chaty.

Vďaka dynamicky sa rozvíjajúcemu internetovému prostrediu dochádza k realizácii aplikácií pre podporu business procesov prostredníctvom webových aplikácií. Systémy kopírujú procesy a aktivity vo firme, napomáhajú dodržiavať presne stanovené postupy.

Rozhranie týchto aplikácií tvorí navigácia závislá na zobrazovanom obsahu. Dáta sa ukladajú do databázových systémov, vďaka ktorým je možné dynamicky modifikovať obsah stránky v závislosti na týchto údajoch.

Okrem klasických osobných počítačov sa čím ďalej tým viac využívajú alternatívne internetové prehliadače v telefónoch podporujúcich WAP, telefónoch tretej generácie, alebo v PDA prístrojoch. Tomu sa musí prispôbiť aj vývoj aplikácií, ktoré by mali byť nezávislé na platforme a prehliadačoch, teda celkovo na použitých technológiách. Pri takýchto aplikáciách je nevyhnutný aj rovnako nezávislý návrh.

Často sa stáva, že implementovaný systém nie je ohodnotený ako úspešný. Bežné praktiky návrhu webových aplikácií sa vôbec nezameriavajú na to, čo je pre webové informačné systémy charakteristické - hypertextová navigácia. Deje sa tak predovšetkým z nedostatku kvalitných metodík návrhu webových aplikácií. Dochádza k nasledujúcim problémom:

- datacentrické metódy nepokrývajú front-endovú časť hypertextových aplikácií,
- objektovo orientované metodiky (ako napríklad UML) nemodelujú hypertextovú navigáciu, ktorá je, ako bolo vyššie uvedené, charakteristická pre webovo orientované aplikácie,

- prezentácia nie je vôbec, alebo len minimálne modelovaná,
- aplikácia obsahuje množstvo ručne písaného programového kódu, ktorý by mohol byť použitím vhodného nástroja automaticky vygenerovaný.

Konceptuálne modelovanie aplikácií je dôležité, pretože umožňuje modelovať aplikáciu z platformne nezávislého pohľadu, nezávisle na používaných programovacích prostriedkoch. Rovnako dôležitý je pri návrhu aj hypertextový model (zložený z návrhu obsahu stránok a ich navigácie), ktorý by tiež mal byť modelovaný nezávisle na platforme. Vygenerovaním šablón a jednotlivých komponent (závislých na programovacom jazyku a platforme) sa vývojári zaoberajú až po počiatočnom návrhu systému.

Dôležitosť návrhu aplikácie v súčasnosti snáď nikto nespochybňuje. Pri zanedbaní návrhu dochádza predovšetkým k nasledujúcim problémom:

- Problémy s údržbou. Predovšetkým v databázach dochádza k redundanciám a nekonzistenciám.
- Problémy s použiteľnosťou. Ak nie sú určité informácie na stránkach konzistentné, užívateľ pravdepodobne stratí dôveru v cele webové sídlo.

Cieľom tejto bakalárskej práce je navrhnúť metodiku vývoja intranetových podnikových aplikácií pre spoločnosť ComAp, s.r.o., ktorá je svetovým lídrom na trhu elektrotechnických zariadení. Všetky intranetové aplikácie v ComApe sú webovými aplikáciami, vytvárané v jazyku PHP. Navrhovaná metodika bude preto s najväčšou pravdepodobnosťou vychádzať z metodiky WebML (Web Modelling Language), ktorá sa používa na návrh webových sídel.

Po podrobnom popise metodiky WebML ju porovnam s inými metodikami používanými pri analýze a návrhu webových sídel. Popíšem nástroj WebRatio, ktorý je jedným z mála programov podporujúcich metodiku WebML. Rozoberiem konkrétny projekt z ComApu, ktorý doposiaľ nebol realizovaný, jeho vývoj je však naplánovaný v priebehu najbližších mesiacov. Vytvorím základné modely tejto aplikácie vo WebRatiu. Ukážeme si aj program Enterprise Architect, ktorý síce nepodporuje priamo metodiku WebML, napriek tomu je možné vytvárať diagramy podobne tým z WebRatia. Enterprise Architect by mohol byť výbornou alternatívou pri modelovaní systémov. Na rovnakom projekte vyskúšam jeho návrh v Enterprise Architecte a porovnam návrh v oboch modelovacích nástrojoch.

Ďalej sa budem venovať súčasnému vývoju aplikácií v spoločnosti ComAp, s.r.o. a pokúsim sa modifikovať WebML metodiku priamo pre jej potreby. Na záver vypracujem krátke odporúčanie pre ComAp, s.r.o., obsahujúce návrh zavedenia metodiky WebML do súčasného procesu vývoja webových aplikácií. Výstupom tejto bakalárskej práce bude odporúčaná metodika návrhu webových systémov pre vývojárov ComApu vychádzajúca z metodológie WebML prispôbená špeciálne pre ich potreby.

1 Popis metodiky WebML

”Metodika WebML je jednou z najmladších metodik návrhu webových aplikácií. Jej korene siahajú na univerzitu v Miláne, kde ju vyvíja skupina nadšencov. Metodika WebML je určená na návrh hypermediálnych aplikácií a webových sídel. Je postavená na dvoch základných modeloch: konceptuálnom a navigačnom.” [Molh2] Je nutné dodať, že hypermediálnou aplikáciou rozumieme systém obsahujúci nielen text, ale aj multimediálne prvky, ktoré sú vzájomné poprepájané. Inormácie o metodike som čerpal predovšetkým z dokumentov [Kuru, Molh, Molh2, Frat] a článkov p. Zelenky (viď. zoznam literatúry).

Vývoj aplikácií je zložený z nasledujúcich základných krokov:

1. Definovanie a analýza požiadaviek. V prípravnej fáze získavame požiadavky kladené na web aplikáciu - jej hlavné ciele, požadovanú personalizáciu, rozdelenie užívateľov do základných rolí, zostavíme profil cieľového návštevníka a prípadne zostavíme nejaký príklad toho, čo by asi malo webové sídlo obsahovať na web stránkach.
2. Modelovanie konceptuálneho diagramu obsahujúceho entity, ich atribúty a vzťahy medzi jednotlivými entitami.
3. Návrh užívateľov a užívateľských skupín. Na základe personalizačných požiadaviek sa vytvoria užívateľské profily (role), pre ktoré sa budú v nasledujúcich krokoch vytvárať tzv. site-views - pohľady na webové sídlo z pohľadu istej skupiny užívateľov.
4. Hypertextový návrh
 - (a) “vo veľkom” predstavuje identifikovanie stránok a unitov a ich prepojenie,

- (b) “v malom” posudzuje individuálne každú stránku a každý unit na stránke.
5. Prezentačný design. Po tom, čo sú všetky stránky v hypertextovom modeli navrhnuté relatívne stabilne (finálne), je možné pre každú stránku navrhnuť jej prezentačné štýly.
 6. Realizácia, implementácia a testovanie.
 7. Údržba.

Niektoré z týchto krokov môžu byť vynechané - v závislosti od rozsiahlosti webovej aplikácie. Realizácia, testovanie, implementácia a údržba nás v tomto texte nebude zaujímať, pretože metodika WebML je zameraná na zefektívnenie a skvalitnenie analýzy a návrhu systémov, nie samotného vývoja aplikácií.

WebML modeluje synchronne, užívateľsky orientované aplikácie, pridržajúce sa nasledujúcich pravidiel:

- Žiadna akcia sa svojvoľne nespustí - vždy sa čaká na podnet užívateľa.
- Rozhranie sa zobrazuje užívateľom, ktorí v niektorých prípadoch trávajú s aplikáciou veľa času, musí byť teda užívateľsky prívetivé a jednoduché.
- Navigácia nie je (s malými výnimkami) nijak obmedzená.

Metodika WebML rozdeľuje špecifikáciu požiadaviek na dve fázy - ich zber a analýzu. Vo fáze zberu požiadaviek rozlišuje nasledujúce typy informácií a oblastí, ktorými by sme sa mali zaoberať:

- funkčné požiadavky, pri modelovaní ktorých je možné použiť Use Case diagramy,
- identifikácia užívateľov je tiež súčasťou Use Case diagramov,
- požiadavky na personalizáciu a dáta,
- ostatné požiadavky na výkonnosť, škálovateľnosť, dostupnosť a bezpečnosť.

Špecifikácia požiadaviek musí mať zrozumiteľné výstupy ako pre vývojárov, tak aj pre obchodníkov.

1.1 Konceptuálny model

Štruktúrálny, nazvaný tiež konceptuálny dátový model, vychádza zo štandardných Class diagramov zložených z entít, relácií (vzťahov medzi entitami) a atribútov jednotlivých entít. Model sa zaoberá dátami, s ktorými webová aplikácia pracuje. Pre návrh štruktúrného modelu je možné využiť ľubovoľný nástroj dátového modelovania - akýkoľvek CASE nástroj podporujúci notáciu UML.

Vzorový dátový model vytvorený v programe WebRatio znázorňuje obrázok 1 v prílohe. Štruktúrné diagramy sú zachytené aj na obrázkoch 11 a 12 - jeden vytvorený v Enterprise Architecte, druhý vo WebRatiu. Oba znázorňujú aplikáciu IntraTimer - vid'. kapitulu 5 - Praktický príklad.

Jednotlivé entity, ich atribúty a vzťahy medzi nimi umožňuje zachytávať štruktúrálny model v platformne nezávislej vizuálnej podobe. Atribúty je možné rozdeliť do nasledujúcich (univerzálnych) typov: String, Integer, Float, Blob a ďalšie. Zaujímavé je, že pri generovaní skriptov konkrétneho jazyka sa tieto univerzálne typy atribútov prekonvertujú do typov špecifických pre ten ktorý programovací jazyk.

Zo štruktúrného modelu je možné vytvoriť odvodený model. Podľa [Molh2] vzniká odvodený model zo štruktúrného modelu jeho rozšírením o tzv. odvodené atribúty, entity a vzťahy. Prakticky ide o to, že "každá webová stránka je spojená s odvodeným dátovým modelom, ktorý sa dá získať dotazom do pôvodného dátového modelu". [Molh2]. Je však nutné rozlišovať odvodené atribúty, ktorými sú napr. agregované, vypočítané atribúty, od odvodených entít a vzťahov, ktorými sa rozumie pohľad (tzv. view) do pôvodného dátového (štruktúrného) modelu. Podobné odvodené modely sa objavili aj v metodikách RMM a OOHDm.

[Molh2] ďalej uvádza, že získanie odvodeného dátového modelu je možné špeciálnym dotazovacím jazykom odvodeným z dotazovacieho jazyka OQL (Object Query Language) a bol pomenovaný WebML OQL. "Ide o veľmi bohatý jazyk zabudovaný do XML." [Molh2]

1.2 Hypertextový model

Tento model je najdôležitejším prvkom celej metodiky. Návrhár pomocou tohto modelu definuje chovanie celej webovej aplikácie. Rovnako ako v prípade konceptuálneho modelu, robí sa nezávislé na platforme. Hypertextový model sa zameriava na navigáciu medzi jednotlivými stránkami a na ich zloženie. Do modelu sa preto pridávajú hypertextové odkazy, čím sa dosiahne prepojenie informačného obsahu jednotlivých stránok odkazmi. Cieľom je navrhnúť spôsob navigácie na web stránkach.

Na obrázku 2 je veľmi jednoduchý príklad hypertextového modelu systému evidujúceho hudobných interpretov. Na jednej internetovej stránke je zoznam všetkých dostupných interpretov, na druhej stránke informácie o konkrétnom interpretovi a scroll unit (popis vid'. text nižšie). Medzi týmito dvoma stránkami je jednoduchý prechod - po kliknutí na interpreta v liste sa prenáša jeho ID stránku s detailmi.

Obrázok 13 je príkladom toho, ako je možné v Enterprise Architecte použiť alternatívne unity a linky k namodelovaniu hypertextového modelu obdobného tomu z WebRatia. Unity v tomto modeli bohužiaľ nie sú tak farebne odlišené, ako vo WebRatiu, napriek tomu je tento jednoduchý model prehľadný.

Na obrázkoch 15, 16 a 17 je zachytená časť hypertextového modelu aplikácie IntraTimer (vid. kapitolu 5). WebRatio bohužiaľ nedokáže exportovať model v podobe jedného obrázku, ale ľubovoľné ho rozseká.

Špecifikácia WebML udáva, že návrh aplikácie pozostáva (okrem iného) z jedného alebo viacerých hypertextových modelov - tzv. site views (pohľady na webové sídlo z pohľadu užívateľov rôznych rolí). Takéto pohľady poskytujú tri hlavne výhody:

- pohľady podporujú viac-užívateľský model, v ktorom majú skupiny užívateľov rôzne požiadavky a práva.
- náklady na vývoj môžu byť efektívne redukované práve vďaka tomu, že objekty sú rozdelené v skupinách do pohľadov.
- hypermediálne aplikácie sú jednoducho modifikovateľné a rozšíriteľné - dodatočné požiadavky na prezentáciu a navigáciu je možné relatívne jednoducho začleniť do existujúceho modelu.

Hypertextový model sa skladá z:

- kompozičného modelu, ktorý popisuje, z akých logických častí sa skladá webová stránka. Logickou časťou je napr. Data unit, Multidata unit, Scroller Index a Entry unit - ich význam rozoberieme v nasledujúcom texte. Každý prvok je vo WebML reprezentovaný špeciálnym grafickým značením a všetky (okrem prvku Entry) sa spájajú s inou entitou zo štruktúrného modelu.
- navigačného modelu, ktorý popisuje spôsob navigácie medzi rôznymi stránkami webového sídla.

Rovnako, ako v metodike OOHD, aj vo WebML je vhodné, resp. v mnohých prípadoch nevyhnutné, vytvárať pre jeden štruktúrný model viac hypertextových. Popisu site-views sa budeme venovať v ďalšom texte.

1.2.1 Kompozičný model

Kompozičný model, zachytávajúci logickú štruktúru užívateľského rozhrania, sa skladá z rôznych unitov. Unitom rozumieme atomický element slúžiaci na publikovanie istého obsahu na stránke, na vkladanie dát prostredníctvom formulára alebo na vyber niekoľkých objektov. Z toho vyplýva, že stránka (page) je kontajnerom obsahujúcim jeden alebo viac unitov. Tieto unity rozdeľujeme na nasledujúce typy (viď. obrázok 3):

- Data unit slúži na zobrazenie jedného objektu istej entity. Znárodňuje napríklad údaje o produkte (názov, popis, cena).
- Multidata unit slúži na publikovanie niekoľkých objektov v rámci jednej stránky. Príkladom môže byť prehľad sub-komponent, ktoré spolu tvoria jeden objekt. Spôsob zobrazenia je však vecou prezentačného modelu, nie štruktúrného.
- Index unit predstavuje zobrazenie niekoľkých objektov v podobe (indexovaného) zoznamu (napríklad produktov). Po kliknutí na položku zoznamu je užívateľ navigovaný na inú stránku - detailné informácie o konkrétnom produkte.
- Multichoice index unit je obdobou Index unitu, v ktorom je každý prvok zoznamu spojený s tzv. checkboxom (zaškrtávacie políčko). Pri odoslaní formulára je užívateľ navigovaný na inú webovú stránku, kam sa prenášajú informácie o zaškrtnutých

prvkoch.

- Hierarchical unit zobrazuje hierarchický zoznam, napríklad zoznam produktov rozdelených do jednotlivých kategórií
- Scroller unit je prvok umožňujúci prechádzať medzi jednotlivými objektmi určitej entity. Zvyčajne obsahuje možnosti: vpred, vzad, na začiatok, na koniec, prípadne odkaz späť na zoznam entít.
- Entry unit predstavuje vstupný formulár slúžiaci užívateľovi na zadávanie údajov. Po odoslaní nasleduje akcia navigácie na ďalšiu stránku, kam sa zadane údaje prenášajú a spracovávajú. Istou obdobou sú tzv. filter units, ktoré predstavujú položky umožňujúce meniť vyhľadávacie kritéria a tým obsah celého zobrazovaného listu.

Vo všeobecnosti, Data a Multidata units prezentujú obsah istého objektu, zatiaľ čo všetky ostatné unity umožňujú tieto objekty lokalizovať.

Na obrázku 17 je vidieť stránku *Užívateľské nastavenie*, ktorá sa skladá zo štyroch formulárov editujúcich nastavenia domovskej stránky, počtu záznamov na stránke a podobne. Stránka ma vpravo dole označenie **L**, ktoré naznačuje, že ide o Landmark - stránka vždy prístupná, z ktoréhokoľvek miesta webovej aplikácie. Prakticky to znamená, že odkaz na túto stránku sa nachádza niekde v hlavnom menu aplikácie.

Area *Nový časový záznam* na obrázku 15 zahrňuje stránky *Vyplnenie tabuľky*, *Pridanie záznamu*, *Import záznamu*. Písmeno **D** v pravej dolnej časti stránky *Vyplnenie tabuľky* znázorňuje, že táto stránka je nastavená ako predvolená pre túto sekciu.

Úlohou hypertextového diagramu je okrem iného aj zachytiť business logiku aplikácie. K tomu slúžia tzv. operation units - ich cieľom je predovšetkým aktualizovať údaje v databáze alebo uskutočniť ľubovoľnú akciu. Metodika WebML definuje množstvo unitov, napríklad:

- Create unit - vytvára nový záznam v databáze.
- Delete unit - vymazáva záznam z databázy.
- Modify unit - aktualizuje databázu.
- Send mail unit - posielá automaticky vygenerovaný e-mail.

- Connect a Disconnect unity.

Predchádzajúce unity sa v modeloch spájajú s OK a KO linkami (viď. obrázok 15 - vľavo). OK link zaznamenáva prechod na iný unit (resp. stránku) v prípade úspešnej operácie (napríklad modifikovanie záznamu). KO pravé naopak - kam bude užívateľ navigovaný, ak sa pri modifikácii vyskytne problém. Operácie nezobrazujú žiaden obsah - sú preto v modeloch pozicované mimo stránky.

1.2.2 Navigačný model

Cieľom navigačného modelu je zachytiť navigáciu medzi jednotlivými stránkami webového sídla. Stránky sú prepojené odkazmi, ktoré môžu byť kontextové alebo bezkontextové. Kontextové odkazy si medzi sebou prenášajú informácie. Odkazom rozumieme orientovane spojenie medzi dvoma unitmi (zdrojový a cieľový unit) - prakticky sú na stránkach reprezentované odkazmi alebo tzv. submit buttonmi.

Cieľom odkazov je umožniť užívateľovi presun z jedného miesta webového sídla na iné, prenášať informácie medzi stránkami, alebo aktivovať nejakú činnosť (napr. výpočet).

Kompozičný a navigačný model sa zakresľuje do jedného diagramu a vytvárajú tak spolu hypertextový model. Hypertextový model je značne zrozumiteľný, prehľadný a ma vysokú vypovedajúcu hodnotu. Z diagramu je okamžite jasné, čo sa na ktorej web stránke nachádza a kam sa môže užívateľ "prekliknúť".

1.3 Prezentačný model

Prezentačný model vyjadruje vzhľad a rozmiestnenie jednotlivých prvkov nezávisle na výstupnom formáte (HTML, WAP, ...). Prezentácia sa modeluje vo WebRatiu, čo je jedným z mala programov priamo podporujúci metodiku WebML. Podstatou prezentačného modelu je usporiadanie entít z konceptuálneho modelu do tabuľky (viď. obrázok 22) tak, ako ich chceme na stránke zobrazovať. Tvorbu prezentačného modelu si ukážeme v ďalšom texte na praktickom príklade.

Všetky modely ukladá WebRatio do XML dokumentu. Transformáciu z XML do Javy alebo ASP.NET prevádza XSLT (eXtensible Stylesheet Language Transformation).

1.4 Implementačné nástroje

Najlepším programom podporujúcim metodiku WebML je s najväčšou pravdepodobnosťou komerčný program WebRatio. Ďalšími sú napríklad:

- Site Designer - základný nástroj umožňujúci vytvárať štrukturálny, hypertextový a užívateľský model.
- Presentational Designer - dokáže navrhovať vzhľad a rozmiestnenie grafických prvkov na webovej stránke.
- Site Manager - umožňuje spravovať webové sídlo vytvorené metodológiou WebML.
- BEA WebLogic Workshop - dokáže modelovať web-stránky, navigáciu, podporuje webové služby (tzv. Web Services) a generovanie kódu.

1.5 Modelovanie veľkých systémov

Pri rozsiahlych systémoch je pre zvýšenie prehľadnosti nutné zoskupovať homogénne stránky do tzv. areas. V nich je povolené vnorovanie, takže sub-area vnorená v inej arei nie je nič neobvyklého.

V systémoch s rozdelenými prístupovými právami (užívateľ, účtovník, manager, administrátor, ...) sa vytvárajú tzv. site-views. Sú to modely zachytávajúce stránky z pohľadov jednotlivých typov užívateľov. Napríklad bežný užívateľ nemá prístup ku stránkam s rôznymi nastaveniami, preto v "jeho" site-view (rozumej: v jeho pohľade na celé webové sídlo) sa takéto stránky vôbec nevyskytnú.

1.6 Nedostatky metodiky WebML

Myslím si, že metodika sa nezaobrá v dostatočnej miere prvou etapou vývoja aplikácie, ktorou je analýza požiadaviek. Túto fázu by značne zjednodušil jeden alebo viac modelov jazyka UML - napríklad Use Case Diagram alebo Business Process Model. Use Case modeluje interakciu užívateľa so systémom pomocou scenárov a sekvenčných diagramov pre určité skupiny užívateľov. Pre pochopenie systému môže byť veľmi dôležitý aj Business Process Model znázorňujúci chovanie systému a procesy prebiehajúce v ňom.

2 Metodiky podobne WebML

V tejto kapitole sa zameriam na metodiky, z ktorých čerpali autori WebML, ale aj na ostatné metodiky používajúce sa pri analýze a návrhu webovo orientovaných systémov. Medzi najznámejšie metodiky patri Relationship Management Methodology (RMM), Object-Oriented Hypermedia Design Method (OOHDM), alebo Enhanced Object Relationship Methodology (EORM).

Výber ostatných metodik bol prevažne náhodný, pretože v súčasnosti existuje veľké množstvo metodik zaoberajúcich sa návrhom hypermediálnych aplikácií.

2.1 RMM

Podľa [Isak95], návrh aplikácií metodológiou RMM (Relationship Management Methodology) pozostáva zo siedmych krokov:

1. Konštrukcia ER diagramu.
2. Slice design - je to unikátny krok v oblasti návrhu hypermediálnych aplikácií, ktorý navrhuje, ako sa budú informácie v určitých entitách zobrazovať užívateľom a ako k nim budú títo užívatelia pristupovať.
3. Navigačný design.
4. Návrh užívateľského prostredia.
5. Conversion protocol design.
6. Run-time behaviour design.
7. Konštrukcia aplikácie a jej testovanie.

Základom RMM je RMDM (Relationship Management Data Model). Slice design definuje pohľady na RMDM, ktoré odpovedajú uzlom navigačného modelu. RMCASE je CASE nástroj podporujúci túto metodológiu. RMM diagramy je možné integrovať aj do MS Visio (www.visio.com). [Isak95, Isak].

M-slice - nová verzia Slice designu - je založená na možnosti dekompozície prezentačnej jednotky, napríklad web-stranky, na menšie jednotky, ktoré môžu byť opakovane použité. Na podobnom princípe pracujú aplikácie založené na MVC (Model-view-controller), ktoré rozdeľujú stránku na menšie moduly.

”Jednou z hlavných výhod RMM je to, že separuje dáta, štruktúru a užívateľské prostredie. S adresármi a súbormi sa ľahko manipuluje a vývojári dokážu pracovať oddelene, nezávislé na sebe.” [Isak]

Ďalšími výhodami, ktoré [Isak95] prezentuje, sú škálovateľnosť a znovupoužiteľnosť zdrojového kódu, jednoduchá údržba, flexibilita.

2.2 EORM

EORM - Enhanced Object Relationship Methodology - bola prvou objektovo orientovanou metodikou návrhu hypermediálnych aplikácií. Zaujímavosťou metodológie EORM je to, že modeluje vzťah medzi dvoma objektmi ako objekt. Pre modelovanie vzťahov je teda možné využiť výhody objektového paradigmatu. EORM je postavený na 3 základných rámcoch:

- Modelovanie tried, objektov - bežný Class diagram.
- Model kompozície - špecifikuje triedy odkazov (napr. simpleLink - trieda jednoduchých odkazov).
- Model GUI (Graphical User Interface) - navrhuje sa užívateľské rozhranie.

Ako uvádza [Molh], EORM je sémanticky bohatšia v porovnaní s RMM metodológiou.

2.3 HDM

HDM (Hypermedia Design Method) je jednou z prvých metód na návrh hypermediálnych aplikácií. Tato metóda bola neskôr nahradená metodológiou OOHDM, vid' text nižšie.

2.4 OOHDM

S vývojom objektov v posledných rokoch bola prepracovaná aj metóda HDM do objektovej podoby - odtiaľ názov Object-Oriented HDM. Išlo v podstate o kompletne prepracovanie tejto metódy. Návrh metodikou OOHDM sa podľa [Schw] skladá zo štyroch krokov:

1. Vytvorenie konceptuálneho návrhu, pri ktorom sa využívajú diagramy UML. Modelujú sa triedy a vzťahy medzi nimi, čo je vlastne podstatou objektovo orientovaného modelovania.
2. Návrh navigácie. V hypermediálnej aplikácii je dôležité zachytiť vzťahy, resp. prechody, medzi jednotlivými stránkami webového sídla. Vzhľadom k tomu, že drvivá väčšina webových informačných systémov obmedzuje prístupové práva, vznikajú určité skupiny užívateľov. Pre každú z týchto skupín (napr. administrátori, bežní užívatelia) je nutné modelovať tzv. site views - pohľady na stránky. Skupiny užívateľov sa využívajú aj pri modelovaní užívateľsky zameraných scenárov.
3. Návrh abstraktného rozhrania.
4. Implementácia - vytvorenie a zavedenie aplikácie.

[Schw] uvádza, že vyššie uvedeným krokom predchádza tzv. requirements gathering, t.j. proces získavania požiadaviek na daný software. V tejto fáze je dôležité modelovať interakciu užívateľa so systémom. K tomu slúži User Interaction Diagram, scenáre a Use Case modely. Cieľom tejto prípravnej fázy je zachytiť a presne definovať požiadavky, ktoré na danú aplikáciu majú zadávatelia. Tento proces musí:

- identifikovať aktérov a ich interakciu so systémom (User Interaction Diagram).
- zachytiť scenáre pre každú činnosť a typ aktéra daného systému.

Výstupom procesu je grafická reprezentácia interakcií medzi systémom a užívateľom počas vykonávania istej činnosti.

”Následníkom tejto metodológie je SHDM (Semantic Hypermedia Design Method), ktorá podobne ako OOHDM, umožňuje zhustiť špecifikáciu a implementáciu hypermediálnych webových aplikácií.” [Schw]

Obdobnou metódou je SOHDM - Scenario-based Object-oriented HDM, ktorá sa zameriava na procesne orientované hypermediálne systémy s cieľom podporovať firemné procesy.

2.5 SOHDM

Scenario-based Object-oriented Hypermedia Design Methodology pozostáva zo šiestich etáp, ktoré sú podrobnejšie rozpisane v dokumente [\[SOHDM\]](#):

- Analýza domény. V počiatočnej fáze návrhu je nutne namodelovať všetky objekty, ktoré by mohli (ale nemusia) mať niečo spoločné s vyvíjaným systémom v rámci domény. Doménou môže byť napríklad celá firma.
- Objektové modelovanie - modelovanie scenárov a transformácia do objektov v podobe kartičiek s presne definovanou štruktúrou - vid'. obrázok 20.
- Modelovanie pohľadov (view design). V tejto fáze sa transformujú objekty z objektového modelu do navigačných jednotiek. Niekoľké navigačné jednotky vytvárajú objektovo orientovaný pohľad.
- Modelovanie navigácie. V dobre navrhnutých hypermediálnych systémoch je veľmi dôležité to, ako užívatelia prechádzajú zo stránky na stránku a aké informácie sa im zobrazujú - s cieľom predísť redundantným informáciám a tiež so snahou nedovoliť užívateľom stratiť sa v obrovskej záplave informácií.
- Modelovanie implementácie. Fáza implementácie generuje užívateľské prostredie, štruktúru stránok a databázové schéma pre konkrétne implementačné prostredie (závislé na použitej platforme).
- Konštrukcia. V tejto fáze vývojári implementujú hypermediálny systém do cieľového prostredia.

”Scenáre sa používajú na analyzovanie a modelovanie užívateľských požiadaviek. Bližšie špecifikovanie scenárov pomáha udržať flexibilitu systému už od počiatočných fáz vývojového procesu. Scenáre sa transformujú do objektovo orientovaných pohľadov. Tie sa používajú na modelovanie hypermediálnych stránok a navigačných spojení.” [\[SOHDM\]](#)

2.6 VHDM

V dokumente [SOHDM] sa autor v krátkosti venuje aj metodike VHDM, kde uvádza, že celá metodika View-based Hypermedia Design Methodology (VHDM) je založená na pohľadoch (tzv. views), ktoré sa používajú na reprezentáciu užívateľského vnímania hypermediálnych požiadaviek. VHDM napomáha pri skúmaní funkcionality pohľadov v hypermediálnych aplikáciách.

2.7 eW3DT

”Extended World Wide Web Design Technique, skrátene eW3DT, je metodika pozostávajúca z dvoch častí. Za prvé, definuje jednoduchú modelovaciu techniku, ktorá umožňuje namodelovať štruktúru webovej prezentácie alebo informačného systému. Za druhé, pozostáva z návrhového prostredia zvaného WebDesigner, ktorý umožňuje návrhárom vytvoriť funkčný prototyp aplikácie vrátane HTML šablón a CGI skriptov.” [W3DT]

[W3DT] ďalej uvádza, že diagram pozostáva z minimálne jednej stránky (element Page), s voliteľnými elementmi Links a Layout. Účelom Layoutu je mať vopred preddefinovaný set formátovacích špecifikácií, ako napríklad vzhľad záhlavia, zápätia, pozadie obrázku a podobne. Page, Index, Form a Menu sú základnými elementmi hypermediálnych aplikácií modelovaných pomocou tejto metodiky.

2.8 WSDM

Popis metodiky na jej domovských stránkach [WSDM] a v dokumente [Troy] uvádza, že WSDM (Web Site Design Method) vychádza z chovania potencionálneho užívateľa. Návrh metodikou WSDM je zložený z troch krokov:

1. Modelovanie typického užívateľa a analýza jeho chovania - čo robí užívateľ na webovom sídle, ako sa tam dostal, čo hľadá na stránkach, je možné poskytnúť mu okamžite nejaké informácie? Ak áno, aké?
2. Konceptuálny návrh aplikácie.
3. Návrh implementácie.

2.9 UWE

”Metodológia UML-based Web Engineering (UWE) je objektovo orientovaná, pričom jadro tejto metodológie spočíva v analýze požiadaviek, modelovaní užívateľských aktivít a modelovaní konceptuálneho, navigačného a prezentačného modelu.” [Mont]

”Modely sa rovnako ako u iných metodík ukladajú v podobe XML dokumentov. Automatické generovanie webových aplikácií z modelov zabezpečuje nástroj AgroUML.” [Mont]

2.10 Porovnanie s WebML

Nevýhodou niektorých metodík je nedostatočná dokumentácia, prípadne dokumentácia dostatočne bohatá, avšak nepreložená do anglického jazyka. Stáva sa to predovšetkým pri metodikách vzniknutých v Španielsku alebo Portugalsku. WebML však tento problém rozhodne nemá, pretože je veľmi podrobne zdokumentovaná. Oficiálne stránky WebRatio obsahujú aj niekoľko prípadových štúdií zahŕňajúcich množstvo modelov, ktoré jasne demonštrujú použitie jednak metodiky a jednak samotného nástroja WebRatio.

Rovnako dobre je zdokumentované aj WebRatio. Hneď po nainštalovaní ma užívateľ k dispozícii niekoľko príručiek - rýchly sprievodca, kompletný manuál, popis metodiky WebML a iné.

Myslím si, že veľmi zaujímavá je metodika WSDM vytvárajúca potencionálneho užívateľa aplikácie. Návrhárov to určite prinúti zamyslieť sa nad tým, čo bude užívateľ robiť so systémom a prispôbia tomu napríklad navigáciu systému. Myslím, že metodika WebML by mohla integrovať podobné zamyslenie sa nad užívateľom.

HDM nie je prakticky možné porovnávať s WebML, pretože nie je založená na objektoch. Jej následník, OOHDH, je však veľmi podobný metodike WebML, pretože prakticky WebML je odvodená (aj) z OOHDH. V oboch metodológiách sa vytvára konceptuálny a navigačný model. WebML však už veľmi nespomína User Interaction diagram, ani Use Case modely.

EORM, rovnako ako iné metodológie, podporuje konceptuálny diagram a model kompozície. Typy odkazov sú však špecifikované triedami. WebML tiež rozlišuje niekoľko typov odkazov, nemodeluje ich však ako objekty.

Na rozdiel od W3DT, metodika WebML neurčuje v hypertextovom modle typ Layoutu.

Nahradil ho samostatný model - prezentačný. Väčšinu ostatných elementov, ako napríklad Page, Index, Form a Link majú obe metodiky rovnaké.

Ako bolo vyššie uvedené, podstatou metodiky VHDM sú pohľady na celé webové sídlo. WebML, rovnako ako VHDM, modeluje takéto pohľady. VHDM preto podľa môjho názoru bude nahradená práve metodikou WebML.

Ideálna metodika by mala obsahovať požiadavky definované v [Mont]. Niektoré z týchto požiadaviek, ktoré som považoval za dôležité, uvádzam v nasledujúcej tabuľke (Tab. 1). Na zavedenie do spoločnosti ComAp by boli vhodné predovšetkým metodiky WebML, OOHDM, RMM a UWE, preto som sa v nasledujúcom porovnaní sústredil len na ne a vynechal metodiky ako HDM alebo eW3DT.

Požiadavky na vybrané metodológie		WebML	OOHDM	RMM	UWE
01.	Nástroj podporujúci metodiku	C	P	P	P
02.	Hypermediálne komponenty	C	C	C	P
03.	Štruktúra navigácie	C	C	C	C
04.	Organizácia multimedialného obsahu	P	P	P	P
05.	Modelovanie užívateľov	C	C	N	N
06.	Popis interaktívneho chovania	C	C	N	C
07.	Modelovanie procesov	C	C	C	C
08.	Funkčné a nefunkčné požiadavky	P	P	P	P
09.	Validačné pravidlá	P	N	N	N
10.	Znovupoužitelnosť modelov	P	C	N	N
11.	Popis interaktívneho chovania	C	C	N	C

Tab. 1 Porovnanie metodík. **P** - podporuje, **C** - čiastočne podporuje, **N** - nepodporuje.

Z predchádzajúcej tabuľky je zrejmé, že požadované kritériá najlepšie spĺňa metodika WebML a za ňou v tesnom závесе UWE. V ďalšom texte sa preto budem zameriavať práve na metodiku WebML.

3 Nástroje WebRatio a Enterprise Architect

V nasledujúcom texte popíšem program WebRatio, ktorý je prakticky jediným nástrojom podporujúcim metodiku WebML. Ide o komerčný produkt, ktorý je možné vyskúšať v demo verzii. Enterprise Architect je program hojne používaný vo veľkých spoločnostiach, čo svedčí o kvalitách tohto programu. Jeden z týchto programov bude figurovať v odporúčaní pre spoločnosť ComAp.

3.1 WebRatio

”WebRatio je prvým a v súčasnosti jediným nástrojom, ktorý priamo podporuje metodiku WebML. Práve vďaka existencii tohto programu je pravdepodobné, že metodika WebML neupadne do zabudnutia, ale začne ju využívať množstvo analytikov a návrhárov. Vďaka WebML integruje WebRatio množstvo zložiek (tzv. units), ktoré poskytujú vývojárom špecifické nástroje a metódy umožňujúce sústrediť sa na požiadavky kladené na aplikáciu, namiesto technológii.” [[WebRatio](#)]

3.1.1 Popis

V prvom rade by som pochválil vývojárov za voľbu programovacieho jazyka, v akom WebRatio napísal. Zvolili Javu a za to som im ja ako užívateľ niekoľkých operačných systémov veľmi vďačný. Ďalším plusom WebRatia je ukladanie diagramov do textových dokumentov vo formáte XML. Otvorenosť, transparentnosť a možnosť využitia XSLT je veľkým pozitívom.

Aplikácie navrhované WebRatiom sú postavené na MVC (Model-view-controller), ktorý oddeľuje logickú a prezentačnú časť systému. Pri požiadavkách kladených na súčasné systémy (modularita, aktualizácie) si ani nedokážem predstaviť návrh aplikácie bez MVC.

Návrh aplikácií vo WebRatiu vychádza zo štyroch základných krokov, ktoré popisuje metodika WebML:

1. Analýza a definovanie požiadaviek, ktoré musí aplikácia spĺňať. Začlenenie špecifikácie požiadaviek do vývojového procesu umožňuje analytikom zamerať sa na účel aplikácie a tým podporuje efektívnosť jej návrhu.

2. Vytvorenie vizuálneho modelu back-endu a front-endu aplikácie.
3. Vygenerovanie plne funkčného prototypu webovej aplikácie. Tento prototyp umožňuje overiť, či systém spĺňa všetky požiadavky naň kladené. Počas procesu vývoja aplikácie je vhodné priebežne konzultovať prototyp so zadávateľom systému, čo umožňuje už počas vývoja odhaliť nepresnosti v zadaní, resp. realizácii projektu.
4. Dolad'ovanie aplikácie.

Čo je teda hlavnou výhodou WebRatia? Tento program je jedinečný práve vďaka tomu, že podporuje v súčasnosti najkomplexnejšiu metodiku návrhu webových aplikácií. Veľkým plusom WebRatia je to, že pri dátovom modelovaní sa nesnaží vymýšľať niečo nové, ale vychádza z notácie UML, na ktorú je väčšina analytikov a programátorov zvyknutá. Príde mi zvláštne, že v class diagramoch používajú opačný spôsob značenia vzťahov. Je však veľmi jednoduché toto nastavenie zmeniť a relácie označovať bežným spôsobom.

Po ukončení návrhu aplikácie je možné vygenerovať zdrojový kód aplikácie. Zdrojový kód je pri dobrom, resp. kompletnom návrhu systému, plne funkčný. Predstavme si jednoduchší systém zložený napr. z prihlasovacieho formulára, jedného-dvoch zoznamov a stránok s detailmi. WebRatio automaticky generuje databázové dotazy, Java komponenty na strane serveru a dokonca aj zdrojový kód dynamických internetových stránok. Dynamické generovanie zdrojových kódov umožňuje predchádzať duplicitu jednotlivých častí kódu, ktorej sa často programátori (obzvlášť tí menej skúsení) dopúšťajú.

Dnešnému multilinguálnemu prostrediu sa prispôsobujú aj internetové aplikácie v snahe pokryť čo najširšie spektrum užívateľov. WebRatio využíva technológie, ktoré dovoľujú vytvárať prezentácie s podporou viacerých jazykov.

Vývojové prostredie WebRatia sa skladá zo štyroch hlavných častí, viď. obr. 22. Postupne si všetky popíšeme:

- Časť prvá - vľavo hore, zobrazuje hierarchicky usporiadané entity vybraného modelu. Je možné prepínať tieto pohľady:
 - editing view,
 - mapping view,
 - a presentation view.

- Časť druhá - vpravo hore. Hlavná a zároveň najväčšia plocha, na ktorej modelujeme samotnú aplikáciu, zobrazuje rôzne diagramy - konceptuálny, prezentačný, alebo hypertextový. Pre každý z týchto modelov je možné vytvoriť nový site-view.
- Časť tretia - vpravo dole, je určená na ladenie programu. Slúžia k tomu nasledujúce záložky:
 - Propagation rules.
 - Errors.
 - Warnings.
 - Search results.
 - Bookmarks.
 - Tasks.
 - Console.
- Časť štvrtá - vľavo dole, zobrazuje informácie o označenej entite. Umožňuje editovať údaje o názve entity, jej type a podobne.

Zo zloženia hlavného okna je zrejmé, že WebRatio je jednoúčelový nástroj, ktorý nie je možné, na rozdiel od Enterprise Architectu, použiť na iné účely.

3.1.2 Výhody a nevýhody

WebRatio je bohužiaľ, ako som spomínal, jednoúčelový nástroj - obsahuje výlučné tie diagramy, ktoré sú súčasťou metodiky WebML. Absencia diagramov (napr. Use Case) môže byť eliminovaná použitím iného programu. V ComApe by sme však najradšej používali jeden univerzálny nástroj.

Najväčšou prekážkou zavedenia WebRatia do ComApu je jeho podpora programovacieho jazyka Java, resp. nepodporovanie iných jazykov, konkrétne PHP. WebRatio dokáže z modelov vygenerovať plne funkčný kód umožňujúci napríklad prihlásenie užívateľa, zobrazenie listu (zoznamu produktov), informácií o konkrétnom produkte a editovanie týchto informácií. V ComApe je však takéto generovanie kódu nepoužiteľné práve z dôvodu používania odlišného programovacieho jazyka - PHP.

Nemožnosť vytlačiť diagram hodnotím ako veľké negatívum. Je však možné vygenerovať dokumentáciu v HTML formáte, ktorej súčasťou sú aj modelované diagramy. Bohužiaľ model je uložený ako množstvo menších obrázkov, čo znemožňuje jeho prenesenie do iných programov, resp. tlač. Pri exporte do PDF formátu sa model kvôli zmenšeniu stáva nečitateľným.

Ďalšia vec, ktorá mi pri modelovaní chyba, je pridanie poznámok do diagramu. Je možné pridať komentár k entite, nezobrazí sa však v diagrame.

Jednoduchosť WebRatia sa odzrkadľuje aj v modelovaní vzťahov medzi entitami - nie je možné zachytiť vzťahy iné než generalizáciu a asociáciu. Enterprise Architect ponúka ďaleko širšie spektrum vzťahov, ktoré môžeme v prípade potreby modelovať.

3.1.3 Hodnotenie

Spomínané nedostatky a mnohé ďalšie budú s najväčšou pravdepodobnosťou odstránené v ďalších verziách programu, ktorý je ešte relatívne krátko vyvíjaný. V súčasnosti ma tieto maličkosti bohužiaľ dokážu maximálne odradiť od používania tohto modelovacieho nástroja.

Na druhej strane, veľmi sa mi páči hypertextový model a jeho farebná variabilita, vďaka ktorej je diagram veľmi prehľadný. Modelovanie je jednoduché a intuitívne aj napriek množstvu rôznych unitov.

3.2 Enterprise Architect

Vzhľad hlavného okna Enterprise Architectu je zachytený na obrázku 19. Vidíme, že je zložené zo šiestich menších okien, v ktorých sa zobrazujú rozličné informácie, vďaka čomu je modelovanie prehľadné a rýchle.

3.2.1 Popis

Enterprise Architect umožňuje vytvárať množstvo modelov - ďaleko viac než poskytuje samotný štandard UML. Mnohé z týchto diagramov sú veľmi prínosné nielen vo fáze návrhu systému, ale aj počas implementácie a údržby (napr. Test Model, Maintenance Model).

- Project Model. Znázorňuje celý životný cyklus projektu - podrobnosti o projekte v jeho rôznych verziách.

- Maintenance Model. Modelovanie údržby aplikácie.
- Test Model. Zachytáva proces testovania aplikácie, objavené chyby a popis ich nápravy. Umožňuje spravovať jednoduché Test Reporty.
- Deployment Model.
- Component Model. Všetky komponenty použité v rámci projektu sa znázorňujú do tohto diagramu.
- Class Model. Zobrazuje návrh tried v klasickom ERA diagrame - entity, vzťahy medzi nimi a ich atribúty.
- Data Model. Zachytáva dátový návrh aplikácie - dátovú štruktúru. Pre každú modelovanú tabuľku je možné zvoliť databázu (napr. MySQL). Názvy atribútov tabuľky (jej stĺpce) sa odvíjajú práve od typu zvolenej databázy. Enterprise Architect obsahuje všetky najznámejšie databázy, pričom je dokonca možné definovať vlastnú databázu a jej príslušiacie atribúty, prípadne modifikovať už preddefinované databázy. Nevýhodou je, predovšetkým v spojení s MySQL databázou, že nie je možné rozlíšiť medzi jej verziami (4.0, 4.1, 5.0, ...).
- Domain Model. Tento model zachytáva všetky objekty v rámci celej spoločnosti - či už fyzické, alebo abstraktné veci - predovšetkým objekty, ktoré sú v predmete záujmu daného projektu.
- Use Case. Modeluje interakciu užívateľa so systémom pomocou scenárov a sekvencných diagramov vždy pre určitú skupinu užívateľov.
- Requirements Model. Štruktúrovaný katalóg požiadaviek koncových užívateľov. Požiadavky rozdeľujeme na:
 - funkčné - požiadavky na užívateľské prostredie, novú funkcionálnu.
 - nefunkcionálne - požiadavky na bezpečnosť, škálovateľnosť systému.
- Business Process Model. Znázorňuje chovanie systému a v ňom prebiehajúce procesy.
- User Interface Model. Tento model by mohol byť veľmi užitočný pri návrhu

aplikácie nebyť jedného nedostatku - pri modelovaní užívateľského prostredia je možné zachytiť stránku, avšak na tej stránke už prakticky nie je možné namodelovať rôzne užívateľské prvky.

Modelovanie v Enterprise Architecte je veľmi intuitívne a jednoduché. Každý projekt, model a jeho entity obsahujú množstvo nastavení, v ktorých je možné modifikovať prakticky čokoľvek.

3.2.2 Výhody a nevýhody

V porovnaní s WebRatiom je Enterprise Architect ďaleko rozsiahlejší. Nepochybne viac modelov a rôznych nastavení mu dáva obrovskú konkurenčnú výhodu.

Jedinú nevýhodu, ktorú sme v ComApe objavili, je to, že nie je možné zvoliť konkrétnu verziu MySQL databázy, pre ktorú chceme vygenerovať SQL kód. Je však nutné zdôrazniť, že testovací dátový model vygenerovaný Enterprise Architectom sa takmer vôbec nelíši od modelov generovaných v iných programoch (CASEStudio), kde bolo možné zvoliť verziu MySQL databázy.

Túto (relatívne malú) nevýhodu potláča množstvo pozitív, ktoré ma v Enterprise Architecte príjemne prekvapili:

- jednoduchá tlač diagramov,
- veľmi dobré možnosti importu a exportu modelov,
- generovanie veľmi prehľadnej dokumentácie jedného či viacerých modelov v HTML formáte, ktorá je graficky aj obsahovo dobre prepracovaná.,
- generovanie PHP skriptov (tried) z Class Modelu. Pre každý atribút je možné veľmi jednoducho vygenerovať tzv. *get* a *set* metódy. Je možné zvoliť, pre akú verziu PHP sa trieda generuje. Pre verziu 5 sa vygeneruje aj dokumentácia v štruktúre DocBook (viď. obr. 18),
- množstvo diagramov, v ktorých je možné zachytiť prakticky všetko týkajúce sa projektu v celom jeho životnom cykle,
- prídavný modul do vývojového nástroja Eclipse, ktorý v ňom umožňuje zobrazovať a prechádzať jednotlivé modely,

- veľmi jednoduchá práca s programom napriek jeho rozsiahlosti.

3.2.3 Hodnotenie

Enterprise Architect považujem za veľmi robustný nástroj na modelovanie aplikácií. Vďaka množstvu nastavení, ktorými si návrhár môže zmeniť prakticky čokoľvek, prehľadnosti diagramov a jednoduchosti ovládania ho radím medzi najlepšie nástroje svojho druhu.

Nevýhodou je absencia integrácie hypertextového modelu, pre ktorý neobsahuje prvky (unity). Je možné nahradiť ich inými, univerzálnymi prvkami, avšak model nie je vizuálne natoľko pekný ako vo WebRatiu.

4 Vývoj v spoločnosti ComAp, s.r.o.

4.1 Súčasný stav

V súčasnosti sa návrh informačných systémov v ComApe riadi interným procesom SH4, ktorý je schválený normou ISO9000. Proces je zložený z niekoľkých krokov:

1. Vznik požiadavku na projekt. S tým je spojené vytvorenie MR dokumentu (Marketing Requirements). Súčasťou MR dokumentu je aj stručná PowerPointová prezentácia, v ktorej je na slidoch zachytený vzhľad istých (väčšinou kľúčových) stránok znázorňujúci zobrazenie listov a formulárov.
2. K danému projektu sa vymenuje SA (System Architect) a PE (Product Engineer). PE vypracuje PR (Product Requirements) dokument, v ktorom popíše navrhovaný spôsob realizácie projektu - časový odhad, vývojové náklady, použité technológie, architektúru produktu.
3. Po schválení PR dokumentu sa zavedie projekt do plánovacieho nástroja.
4. Testovanie projektu nasleduje po jeho (úspešnej) realizácii.
5. Výstupom testovacej fázy je TR (Test Report). Na jeho základe sa rozhoduje o ďalšom postupe - vrátiť projekt späť do vývoja alebo požiadať o schválenie OV (oficiálnej

verzie). Po jej schválení sa táto OV uvoľní.

V ComApe však neexistuje proces alebo metodika popisujúca postup návrhu systému. Webové aplikácie sa navrhujú len minimálne, predovšetkým kvôli časovému odhadu. U mnohých projektov preto dochádza k časovým prieťahom. Pri striktnom posudzovaní ich úspešnosti by boli bohužiaľ hodnotené negatívne.

Dátový model sa v súčasnosti modeluje v nástroji CASEStudio, v ktorom sa aj vytvára SQL kód generujúci databázu MySQL. Návrh užívateľského prostredia sa vytvára v Power-Pointových prezentáciách.

Na modelovanie tried neexistuje v ComApe jednotný nástroj, ktorý by používali všetci vývojári. Hypertextový model sa pri návrhu webovo orientovaných aplikácii vôbec nemodeluje.

4.2 Vízia budúceho stavu

V ComApe potrebujeme jednoduchú metodiku, ktorá by nám zjednodušila návrh webových informačných systémov založených na objektovom princípe. Novozavedená metodika by mala byť predovšetkým prínosom a nie bremenom zaťažujúcim programátorov a project managerov, ktorí budú musieť vytvárať a čítať nezmyselne diagramy.

Na druhej strane, mnoho diagramov, ktoré metodika WebML nezahrnuje, môžu byť veľmi prínosné - napríklad User Interaction Diagram alebo Business Process Model. Odporúčaná metodika bude pravdepodobne zahŕňať aj niektoré nové modely.

Nová metodika bude prispôbená potrebám spoločnosti ComAp, pričom bude primárne vychádzať z princípov metodiky WebML. Prispôbením sa rozumie napríklad upravenie prezentačného modelu tak, aby na ňom bolo možné zachytiť (pokiaľ to bude možné) aj vzhľad jednotlivých stránok.

Hypertextový model, ktorý je hlavným prínosom celej WebML metodiky, v ComApe určite chýbať nebude. Je však možné, že sa nebude modelovať vo WebRatiu aj napriek tomu, že je na modelovanie tohto modelu pravdepodobne najlepší.

Súčasťou odporúčanej metodiky bude aj odporúčenie o tom, ktorý z nástrojov bude do ComApu nutné dokúpiť. Do úvahy prichádza jedine WebRatio a Enterprise Architect.

4.3 Ako to dosiahnuť?

Zavedenie nového postupu návrhu internetových aplikácií do ComApu je spojené s nasledujúcimi javmi:

- kúpa odporúčaného nástroja - WebRatio alebo Enterprise Architect. Cena Enterprise Architectu sa pohybuje okolo 200 amerických dolárov za jednu licenciu. V ComApe bude pravdepodobne nutné zakúpiť tri licencie. Cena WebRatia momentálne nie je známa, pravdepodobne však bude obdobná Enterprise Architectu - spolu približne 600 dolárov v závislosti od typu verzie.
- zaškolenie vývojárov. Všetci vývojári musia byť oboznámení s notáciou diagramov UML a WebML, obzvlášť s tými, ktoré budú súčasťou novozavedeného postupu. Vedúci webového oddelenia (project manager) by mal byť aspoň rámcovo oboznámený s diagramami UML a WebML. Zaškolenie by prebehlo internou formou školenia, ktorého náklady by mali byť okolo 1 človekodňa.

Náklady zavedenia novej metodiky odhadujem na 900 amerických dolárov. Prínosy dobrého návrhu webového systému však budú s najväčšou pravdepodobnosťou v dlhodobom horizonte ďaleko vyššie - okrem iného aj v podobe lepšieho časového odhadu pri tvorbe PR dokumentu.

Novozavedená metodika by mala zlepšovať súčasný stav návrhu systémov, nie len generovať zbytočné modely, ktoré nemajú žiaden reálny prínos. Je preto nutné zvážiť, či má význam vytvárať návrh metodikou WebML aj pre malé aplikácie.

5 Praktický príklad

Použitie metodiky WebML demonštrujem na praktickom príklade - projekte IntraTimer, ktorý sa bude v najbližšej dobe realizovať. Ukážeme si jednotlivé modely vytvorené vo WebRatiu, aj v Enterprise Architecte.

IntraTimer bude aplikácia umožňujúca vedúcim pracovníkom definovať pre každú zákazku, resp. pre každý projekt, zoznam činností (úloh) a k nim priradiť zodpovedného pracovníka. Zamestnanec si bude priebežne evidovať počet hodín odpracovaných na konkrétnej úlohe. Cieľom aplikácie nie je definovať zodpovednosť za určitú činnosť, len jednoducho evidovať čas strávený na úlohách. Toto evidovanie by malo byť v budúcnosti podkladom pre plánovaciu činnosť vedúcich pracovníkov. Zamestnanec bude mať v systéme IntraTimer niekoľko možností zadávania údajov:

- vloženie niekoľkých záznamov v jednom formulári reprezentovaného tabuľkou s voliteľnými dátumami (od, do) súčasne pre väčšie množstvo úloh,
- vkladanie záznamu jedného za druhým s možnosťou definovať popis činnosti a jej typ,
- import z .cvs alebo .xls súborov.

Časový záznam je elementárnou jednotkou. Niekoľko časových záznamov od jedného alebo viacerých zamestnancov bude vytvárať jednu úlohu, činnosť. Pre každý projekt bude definovaná minimálne jedna činnosť. Jedna zákazka obsahuje jeden alebo viac projektov. Užívateľ nemôže evidovať časové záznamy priamo k projektu - bude však existovať skrytá činnosť, ku ktorej bude užívateľ priradovať záznamy v prípade, že nebudú definované žiadne úlohy pre ten ktorý projekt.

Vo WebRatiu som zostavil hypertextový model, vid'. obrázky 10, 15, 16 a 17. WebRatio diagram bohužiaľ rozsekal na menšie časti. Z diagramu je vidieť, z akých stránok bude projekt pozostávať, ako budú tieto stránky previazané a aké činnosti bude ktorá stránka vykonávať. Model je veľmi prehľadný aj vďaka pestrosti farieb - zelené OK linky, červené KO linky, čierne klasické linky, biele stránky, svetlomodré "page areas", ružové unity. Model je skutočne veľmi prehľadný a je veľkým prínosom návrhu systémov.

Naopak, prezentačný model vo WebRatiu je podľa môjho názoru pre návrhárov záťažou, za predpokladu nevyužitia vygenerovaných šablón - ak sa aplikácia nevyvíja v prostredí J2EE. Modelovanie je zdĺhavé a nie veľmi intuitívne.

Konceptuálny model vytvorený vo WebRatiu sa značne líši od diagramu z Enterprise Architectu, vid'. obrázky 8, 11 a 12. WebRatio opäť ponúka obmedzenú paletu nastavení a možností - nie je možné vizuálne odlíšiť *get* a *set* metódy od iných metód, pri metódach sa nezobrazujú návratové hodnoty ani parametre funkcií. Z diagramu nie je jasne viditeľné, ktorý atribút je privátny a ktorý verejný. Naopak, Enterprise Architect disponuje širokým spektrom možností a nastavení, vďaka čomu sa modelovanie stáva zábavným a jednoduchým. Z modelu na obrázku 11 je vidieť, že napríklad entita Task obsahuje 13 atribútov. Niektoré z nich majú preddefinovanú hodnotu false, niektorá 0, niektoré z nich sú privátne, niektoré verejné (users a m_TimeCalculation). Z modelu je ďalej zrejmé, že trieda Task obsahuje celkom 26 metód, z toho je 10 *get* a 7 *set* metód - všetky sú verejné.

Vygenerovanie tried je v Enterprise Architecte veľmi jednoduché. Obrázok 18 zobrazuje triedu Task vygenerovanú pre jazyk PHP verzie 5. Komentáre sú typu DocBook. Kód je nutné upraviť len minimálne - odstrániť prvé dva riadky, na ktorých sa volajú iné súbory (ScmpUser.php a TimeCalculation.php). O volanie týchto súborov sa postará automaticky hlavný controller.

Model tried z Enterprise Architectu je prehľadný vďaka širokej škále použitých farieb a aj vďaka rozdeleniu metód do spomínaných troch skupín.

Po definovaní skupín užívateľov, zostavení konceptuálneho a hypertextového modelu a po zostavení jednoduchého návrhu užívateľského prostredia (v MS PowerPoint) je možné prejsť k samotnému procesu programovania aplikácie. To uľahčí Enterprise Architect, z ktorého je možné vygenerovať šablóny tried z Class modelu. Projekt sa bude realizovať v dohľadnej dobe.

6 Odporúčaná metodika pre ComAp, s.r.o.

6.1 Modelovací nástroj

Po zoznámení sa s nástrojmi WebRatio a Enterprise Architect som dospel k záveru, že WebRatio nie je vhodný na zavedenie do ComApu. Naopak, **Enterprise Architect** je v súčasnosti asi najlepšou voľbou modelovacieho nástroja a to aj napriek tomu, že bohužiaľ nepodporuje notáciu WebML.

6.2 Modely

Vnútropodnikové informačné systémy v ComApe nie sú veľmi rozsiahle, takže by človek mohol nadobudnúť dojem, že žiadne modelovanie nie je nutné a môže sa prejsť priamo k vývoju aplikácie. Opak je však pravdou! Návrh a modelovanie systému svedčí, myslím si, o profesionalite vývojárov a ich odbornom prístupe.

Obligatórne modely

V súvislosti s objektovým prístupom potrebujeme v každom projekte **Class Diagram**, v ktorom budú modelované triedy vytvorené v danom projekte. Každý projekt bude so sebou prinášať pravdepodobne aj globálne triedy, využívané vo viacerých projektoch. Tieto globálne triedy, ktorých môže byť časom v rádoch desiatok, by bolo vhodné mať zoskupené v jednom diagrame.

V Class diagramoch budú navrhované privátne a verejné atribúty, metódy a špeciálne *get* a *set* metódy. Metódy a atribúty by mali obsahovať stručný popis aj kvôli tomu, že z diagramu sa budú generovať PHP triedy s komentármi vo formáte DocBook. Zdanlivo zdĺhavé vypisovanie atribútov a metód bude teda kompenzované tým, že nebude nutné vytvárať zdrojový kód triedy "od nuly" (vid. obrázok 18). Hlavným cieľom Class diagramu je **poskytovať prehľad o vzťahoch medzi objektmi v rámci projektu a vygenerovať šablony tried.**

Ďalším modelom, potrebným na vygenerovanie SQL dotazov, je **Dátový model**. Tento model poskytuje informácie o štruktúre databázy - o jednotlivých tabuľkách a ich stĺpcoch. Diagram zobrazuje informácie o primárnych kľúčoch, typoch stĺpcov a pod. Hlavnou úlohou

tohto modelu je **vygenerovanie SQL dotazov na vytvorenie MySQL databázy**. Na návrh dátového modelu sa v ComApe v súčasnosti používa CASEStudio.

Súčasťou metodiky WebML je aj prezentačný model. Cieľom tohto diagramu je znázorniť usporiadanie elementov na stránke a navrhnuť vizuálny vzhľad systému, vďaka čomu je možné v spojení s WebRatiom vygenerovať hotové HTML šablóny. Vzhľadom k tomu, že v ComApe nebudeme používať WebRatio a tiež kvôli nemožnosti generovať HTML šablóny, stráca tento model opodstatnenie. V tomto smere by som ponechal súčasný stav - **vytváranie užívateľského rozhrania v podobe PowerPointových prezentácií**, z nasledujúcich dôvodov:

- navrhnu sa len tie stránky, ktoré sú kľúčové (zvyčajné 3 - 5),
- nie je nutný žiadny špeciálny nástroj, takže návrh môže spraviť zadávateľ projektu, ktorý o *prezentačných modeloch metodiky WebML* nič nevie,
- a v neposlednom rade preto, lebo sa domnievam, že modelovanie vzhľadu by bolo (v intranetových podmienkach) plytvaním času, keďže všetky aplikácie majú podobný vzhľad a vývojári sa držia určitých zásad (napr. umiestňovanie väčšiny formulárov do tagu *fieldset*).

Kľúčovým prvkom metodiky WebML je **hypertextový model zložený z kompozičného a navigačného modelu**. Myslím si, že tieto dva modely (resp. jeden model obsahujúci oba dva) by mali byť súčasťou návrhu každého systému v ComApe. Kompozičný model znázorňuje zloženie stránok z kľúčových unitov, akými sú napríklad List unit, Data unit, MultiData unit, Scroll a podobne. Cieľom tohto modelu je premyslieť a namodelovať, aké webové stránky bude projekt obsahovať a z čoho sa budú skladať. Zdôrazňujem, že ide len o rámcové zloženie, napr. stránka *Oder info* systému Promap na obr. 21 s informáciami o zákazke bude obsahovať jeden Scroll unit (na prechod medzi zákazkami) a dva Data unity. Cieľom tohto "zamyslenia sa" je okrem iného aj **rozhodnutie o tom, ktoré unity (dáta unit, scroll unit, list) sa budú na stránke združovať do submodulov**. V spomínanom príklade *Oder info* sa spojil jeden dáta unit (na obr. 21 v hornej časti) a jeden scroll (obr. 21 vpravo hore) a vytvorili tak spolu tzv. *Submodul_Top_info*. (Pozn.: popis submodulov nie je predmetom bakalárskej práce.)

Do hypertextového modelu sa zakresľuje aj navigácia (tzv. navigačný model), ktorého cieľom je znázorniť vzťahy a prechody medzi webovými stránkami, vid'. obr. 9 a 13. Navigačný model obsahuje unity modelované mimo stránok - sú to predovšetkým unity Create, Edit a Delete. Cieľom tohto modelu je znázorniť, aké akcie môžu dané stránky vykonávať a kam je užívateľ v prípade ich úspešnosti či neúspešnosti presmerovaný.

Fakultatívne modely

Niektoré metodiky, napríklad UWE a čiastočné aj WebML podporujú tvorbu Use Case diagramy (tzv. prípady užívania), prípadne aj procesných modelov a diagramov aktivít. Tieto modely sú v určitých prípadoch veľmi prínosné, napríklad počas analýzy požiadaviek kladených na informačný systém. Na druhej strane, v určitých prípadoch (návrh malého systému, kde je jeho funkčnosť jasná a zrozumiteľná) môžu návrhára/vývojára zbytočne zaťažovať.

Preto si myslím, že spomínané modely by si mali vývojári osvojiť a v prípade potreby neváhať a vytvoriť daný model. Môžu pomôcť pri dekompozícii aktivít, identifikácii firemných procesov a podobne.

Ako bolo vyššie uvedené, z týchto modelov je možné vygenerovať prehľadnú dokumentáciu, ktorú bude možné priložiť k PR dokumentu.

6.3 Čo potrebujeme?

Na zavedenie vyššie uvedených postupov do návrhu aplikácií potrebujeme **licencie Enterprise Architectu** a **zaškoliť vývojárov** v oblasti objektového návrhu v UML a tiež tvorby hypertextového modelu. Zaškolenie môže prebehnúť formou interného - približne 2 hodínového - školenia.

Nakúpenie licencií Enterprise Architectu od firmy SparxSystems je možné na internetovej stránke http://www.sparxsystems.com/products/ea_purchase.html. Cena za jednu licenciu Professional Edition je 162 Euro pri kúpe 4 a menej licencií. Verzia Corporate Edition je drahšia a nie je nutná, pretože jej hlavným prínosom je nastavovanie bezpečnostných práv pri práci s modelmi a to sa týká predovšetkým väčších tímov než toho v ComApe.

6.4 Prínos z WSDM

Domnievam sa, že ako návrhári a vývojári systémov by sme si mali osvojiť hlavnú myšlienku metodiky WSDM, ktorou je **modelovanie typického užívateľa**. Typickým užívateľom môže byť zamestnanec hľadajúci isté informácie - je vhodné stanoviť, za akým účelom navštevujú užívatelia daný systém, aké informácie hľadajú a tie im treba poskytnúť v čo najkratšom čase, bez zbytočného preklikávania a v čo možno najprehľadnejšej podobe.

Vyplýva z toho ponaučenie, že **systém je vytváraný pre užívateľa**.

6.5 Zhrnutie

V ComApe potrebujeme zaviesť novú, jednotnú metodiku návrhu webovo-orientovaných informačných systémov. Cieľom metodiky je skvalitniť návrh aplikácii, čím sa zlepšia časové odhady, nebude dochádzať ku zdvojovaniu zdrojového kódu a tým pádom sa zníži počet chýb v systémoch. Budeme modelovať:

- **model tried,**
- **dátový model,**
- **hypertextový model,**
- **návrh užívateľského prostredia,**
- a voliteľne Use Case model, procesný model.

Oproti súčasnému stavu modelovania to znamená rozšírenie o:

- model tried - nevyhnutný kvôli objektovému prístupu,
- hypertextový model (kompozícia a navigácia),
- v istých prípadoch aj Use Case a procesne modely.

Nezabúdať, že systém je vytváraný pre užívateľa - musí byť jednoduchý, intuitívny a užívateľsky prívetivý.

Záver

V prvej časti tejto práce som popísal metodiku WebML a jej jednotlivé modely - konceptuálny, hypertextový a prezentačný. Konceptuálny model je klasickým Class modelom v notácii UML. Kompozičný model zachytávajúci logické zloženie jednotlivých stránok a navigačný model znázorňujúci vzťahy (prechody) medzi jednotlivými stránkami tvoria spoločne hypertextový model. Prezentačným modelom rozumieme diagram zobrazujúci prezentačnú logiku - to, čo užívateľ uvidí na jednotlivých stránkach. V prípade rozdelenia užívateľov do užívateľských roli je možné (niekedy až nevyhnutné) modelovať viac hypertextových modelov - jeden pre každú skupinu užívateľov.

Následne som popisoval iné metodiky prispôbené návrhu webových aplikácií. Stručne som popísal princípy metodík RMM, OOHDM a ďalších podobných metodike WebML. Najlepšou z metodík podobných WebML je pravdepodobne OOHDM. V krátkosti som popísal, čo majú metodiky spoločné a v čom sa líšia.

Cieľom mojej bakalárskej práce bolo vytvoriť novú, prípadne prispôbiť už existujúcu metodiku návrhu webových informačných systémov pre potreby spoločnosti ComAp, s.r.o. Inšpiráciou mi bola metodika WebML, ktorá má v súčasnosti podľa môjho názoru najväčší potenciál v oblasti návrhu webovo orientovaných aplikácií. Pôvodne som predpokladal, že budeme modelovať v programe WebRatio, avšak po zoznámení sa s týmto nástrojom som dal prednosť Enterprise Architectu.

Odporúčaná metodika sa líši od WebML tým, že neobsahuje prezentačný model, ani odvodený model, ani jednotlivé pohľady. Je to predovšetkým z toho dôvodu, že interné systémy nie sú, podľa môjho názoru, natoľko obsiahle, aby bolo nutné modelovať rôzne pohľady. Dôležité je, aby sa modelovali len modely, ktoré budú počas návrhu a vývoja skutočne prínosné.

Na rozdiel od metodiky WebML odporúčam vývojárom v prípade potreby modelovať Use Case alebo procesný model. Vidíme preto veľké rozdiely medzi jednotlivými metodikami. Najväčší prínos pre návrh aplikácií v ComApe vidím každopádne v hypertextovom modeli a v nástroji Enterprise Architect.

Reference

- [UML] KANISOVÁ, Hana, MULLER, Miroslav. UML srozumitelně. Brno: Computer Press, 2004. 158 str. ISBN 80-251-0231-9.
- [Zele05] ZELENKA, Petr, WebML - proces vývoje webové aplikace (návrh aplikace), 2005, On-line: <http://interval.cz/clanky/webml-proces-vyvoje-webove-aplikace-navrh-aplikace/>
- [Zele04] ZELENKA, Petr, WebML - proces vývoje webové aplikace (specifikace požadavků), 2004, On-line: <http://interval.cz/clanky/webml-proces-vyvoje-webove-aplikace-specifikace-pozadavku/>
- [Zele04a] ZELENKA, Petr, WebML - tvorba aplikací v prostředí WebRatio, 2004, On-line: <http://interval.cz/clanky/webml-tvorba-aplikaci-v-prostredi-webratio/>
- [Zele04b] ZELENKA, Petr, WebML - operace a manipulace s daty, 2004, On-line: <http://interval.cz/clanky/webml-operace-a-manipulace-s-daty/>
- [Zele04c] ZELENKA, Petr, WebML - struktura webové aplikace, 2004, On-line: <http://interval.cz/clanky/webml-struktura-webove-aplikace/>
- [Zele04d] ZELENKA, Petr, WebML - navigační model, 2004, On-line: <http://interval.cz/clanky/webml-navigacni-model/>
- [Zele04e] ZELENKA, Petr, WebML - kompozice webové aplikace, 2004, On-line: <http://interval.cz/clanky/webml-kompozice-webove-aplikace/>
- [Zele04f] ZELENKA, Petr, WebML - datové modelování, 2004, On-line: <http://interval.cz/clanky/webml-datove-modelovani/>
- [Zele03] ZELENKA, Petr, WebML - projektování webových aplikací, 2003, On-line: <http://interval.cz/clanky/webml-projektovani-webovych-aplikaci/>

- [Kunc] KUNC, Michael. Web Application Modelling. 3 str. On-line: http://www.feec.vutbr.cz/EEICT/2006/sbornik/02-Magisterske_projekty/07-Informacni_systemy/07-dj.mike.pdf
- [Molh] MOLHANEC, Martin. Metodiky orientované na tvorbu webových sídel. 14 str. On-line: <http://honor.fi.muni.cz/tsw/2004/173.pdf> 7, 16
- [Kuru] KURUC, Jaroslav, DOLOG, Peter a BIELIKOVÁ, Mária. Prototyping navigation in web-based informational systems using WebML. 10 str. On-line: <http://www2.fiit.stuba.sk/~bielik/publ/abstracts/2004/datakon-brno2004.html> 7
- [Gome] GOMEZ, Jaime, CACHERO, Cristina a PASTOR, Oscar. Conceptual modeling of device-independent web applications. 14 str. On-line: <http://www.umlchina.net/xprogrammer/source/ieeeMultimedia.pdf>
- [Lang] LANG, Michael. Hypermedia systems development: Do we really need new methods? 9 str. On-line: <http://www.is.nuigalway.ie/mlang/research/Lang148Hyper.pdf>
- [Dolo] DOLOG, Peter a BIELIKOVÁ, Mária. Hypermedia modeling using UML. 8 str. On-line: <http://www2.fiit.stuba.sk/~bielik/publ/abstracts/2002/ism2002.pdf>
- [Molh2] MOLHANEC, Martin. WebML. Objektově orientovaná metodologie pro tvorbu webových sídel. 12 str. On-line: <http://objekty.pef.czu.cz/2003/sbornik/Molhanec2003.pdf> 7, 9
- [Schw] SCHWABE, Daniel, ROSSI, Gustavo. The Object-Oriented Hypermedia Design Model (OOHDM). On-line: <http://www.telemidia.puc-rio.br/oohdm/oohdm.html> alebo <http://www.tecweb.inf.puc-rio.br/oohdm/space/summary+of+OOHDM>. 17
- [Isak] ISAKOWITZ, Tomas. What is RMM. 1997. On-line: http://jmis.bentley.edu/rmm/What_is_RMM.html 15, 16

- [Isak95] ISAKOWITZ, Tomas. RMM: a methodology for structured hypermedia design. 1995. On-line: <http://portal.acm.org/citation.cfm?id=208346&coll=portal&dl=ACM> 15, 16
- [Will] WILLS, Gary, WOUKEU, Arouna, CARR, Leslie a HALL, Wendy. The need for deeper design in a semantic web. 8 str. On-line: <http://eprints.ecs.soton.ac.uk/8203/01/HT03workshopv5.pdf>
- [WSDM] Domovská stránka: WSDM Overview. On-line: <http://wsdm.vub.ac.be/Research/overview.php> 19
- [SOHDM] LEE, Heeseok, LEE, Choongseok, YOO Cheonsoo. A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems. On-line: <http://portal.acm.org/citation.cfm?id=795704.798761> 18, 19
- [W3DT] BICHLER, Martin, NUSSER, Stefan, Modular Design of Complex Web-Applications with W3DT, Wirtschafsuniversitat Wien. On-line: <http://portal.acm.org/citation.cfm?id=832310.837304> 19
- [Troy] TROYER, O.M.F., LEUNE, C.J. WSDM: A user centered design method for web sites. 26 str. On-line: <http://wsdm.vub.ac.be/Download/Papers/WISDOM/WWW7.PDF> 19
- [Frat] FRATERNALI, Piero, BRAMBILLA, Marco. Conceptual modeling of web applications featuring e-services and workflows. SEBD, June, 25th, 2003. On-line: http://www.webml.org/webml/upload/ent5/1/sebd03_tutorial.pdf 7
- [Ceri] CERI, Stefano, MANOLESCU, Ioana. Constructing and integrating data-centric web applications: methods, tools and techniques. On-line: <http://www.vldb.org/conf/2003/papers/S38P05.pdf>
- [Mont] MONTERO, Susana, DIAZ, Paloma, AEDO, Ignacio, A framework for the analysis and comparison of hypermedia design methods. On-line: <http://peterpan.uc3m.es/pdfs/MonteroIasted03.pdf> 20, 21

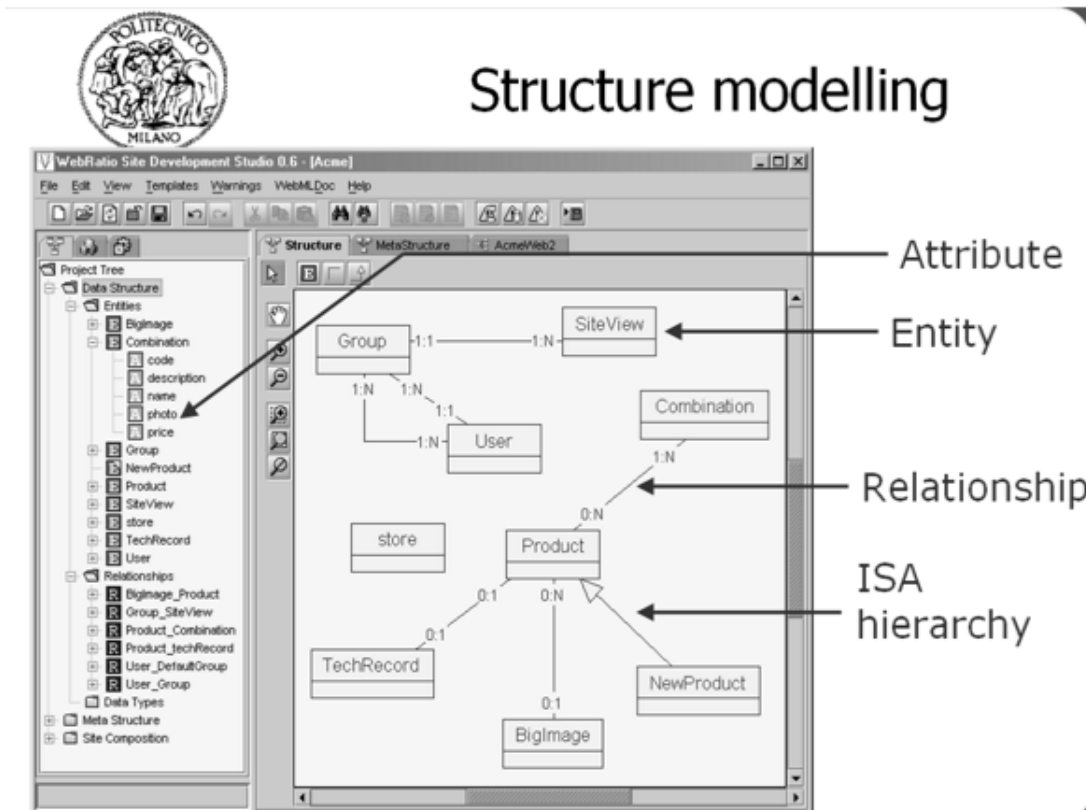
[WebRatio] Domovská stránka: WebRatio. On-line: <http://www.webratio.com/sv1.do> 22

Terminologický slovník

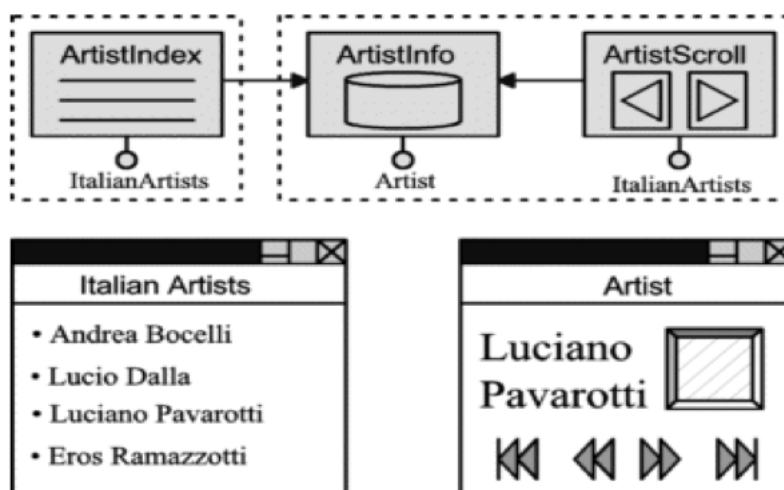
Termín	Význam
Area	Zoskupenie viacerých stránok v hypertextovom modeli.
Enterprise Architect	Komerčný nástroj určený na modelovanie aplikácií.
EORM	Enhanced Object Relationship Methodology. Prvá objektovo orientovaná metodika návrhu hypermediálnych aplikácií.
ERA	Vzťah Entity - Relationship - Attribute. Každá entita v modeli má určité atribúty, entity sú pospájané určitými vzťahmi.
Blob	Dátový typ.
Enum	Dátový typ.
eW3DT	Extended World Wide Web Design Technique. Metodika návrhu webových aplikácií.
Float	Číselný dátový typ.
HDM	Hypermedia Design Method. Metodika návrhy hypermediálnych aplikácií.
Hypertextový model	Model skladajúci sa z kompozičného a navigačného modelu, ktorý znázorňuje zloženie stránok a ich prepojenie.
Kompozičný model	Časť hypertextového modelu zachytávajúca zloženie stránky
GUI	Graphical User Interface - užívateľské rozhranie.
Integer	Číselný dátový typ.
Link	Prepojenie dvoch stránok alebo iných unitov.
MVC	Typ architektúry aplikácií, v ktorej je oddelená prezentačná, dátová a logická vrstva.
Navigačný model	Časť hypertextového modelu zachytávajúca prepojenie jednotlivých stránok.
OOHDM	Object Oriented Hypermedia Design Method. Metodika rozširujúca pôvodné HDM.
OQL	Object Query Language. Objektový jazyk využívajúci sa na dotazovanie v databáze.

PHP	Programovací jazyk.
RMCCase	Nástroj podporující metodologii RMM.
RMDM	Relationship Management Data Model.
RMM	Relationship Management Methodology. Metodika návrhu hypermediálních aplikací.
SHDM	Semantic Hypermedia Design Method. Metodika návrhu hypermediálních aplikací.
SOHDM	Scenario-based Object-oriented Hypermedia Design Method. Metodika návrhu hypermediálních aplikací.
String	Textový datový typ.
Text	Textový datový typ.
Time	Časový datový typ.
UML	Unified Modelling Language.
Unit	Elementární jednotka webové stránky (např. seznam, formulár).
UWE	UML-based Web Engineering. Objektově orientovaná metodika návrhu aplikací.
VHDM	View-based Hypermedia Design Methodology. Metodika návrhu hypermediálních aplikací.
WebML	Web Markup Language.
WebRatio	Komerční nástroj podporující metodiku WebML.
WSDM	Web Site Design Method. Metodika návrhu webových aplikací.
XML	eXtensible Markup Language. Jazyk používající se (v tomto případě) na ukládání modelů do textové podoby.

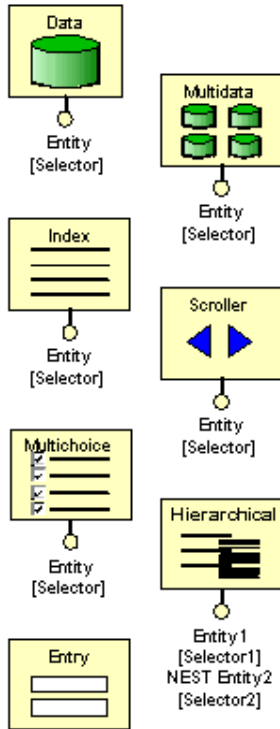
Prílohy



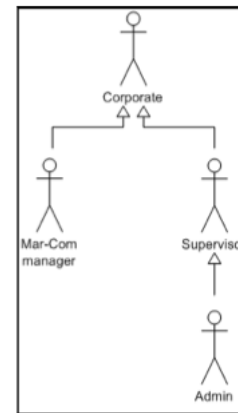
Obr. 1 Štruktúrálly model vo WebRatiu.



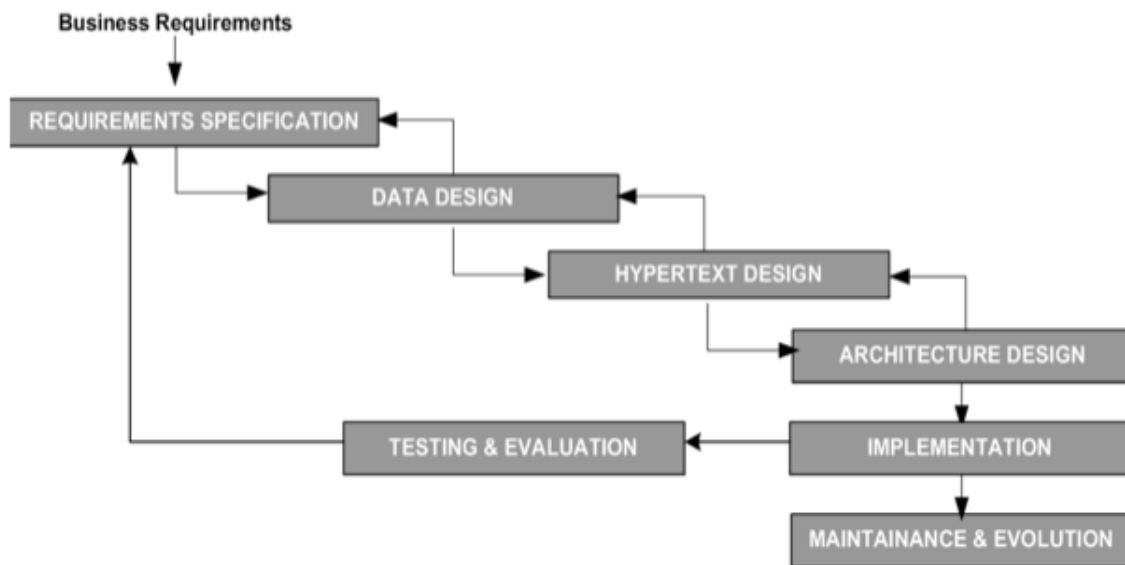
Obr. 2 Jednoduchý hypertextový model.



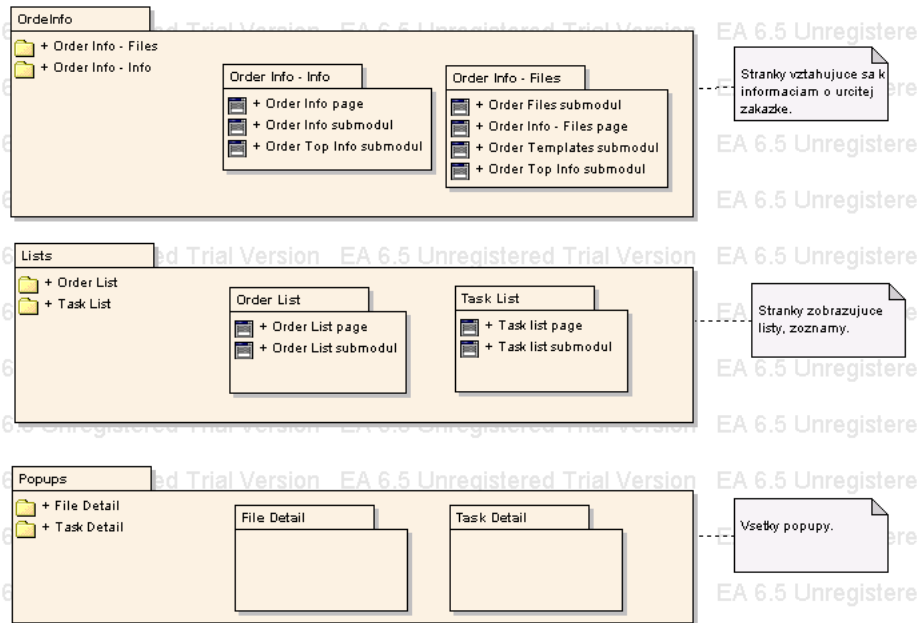
Obr. 3 Typy unitov v hypertextovom modeli.



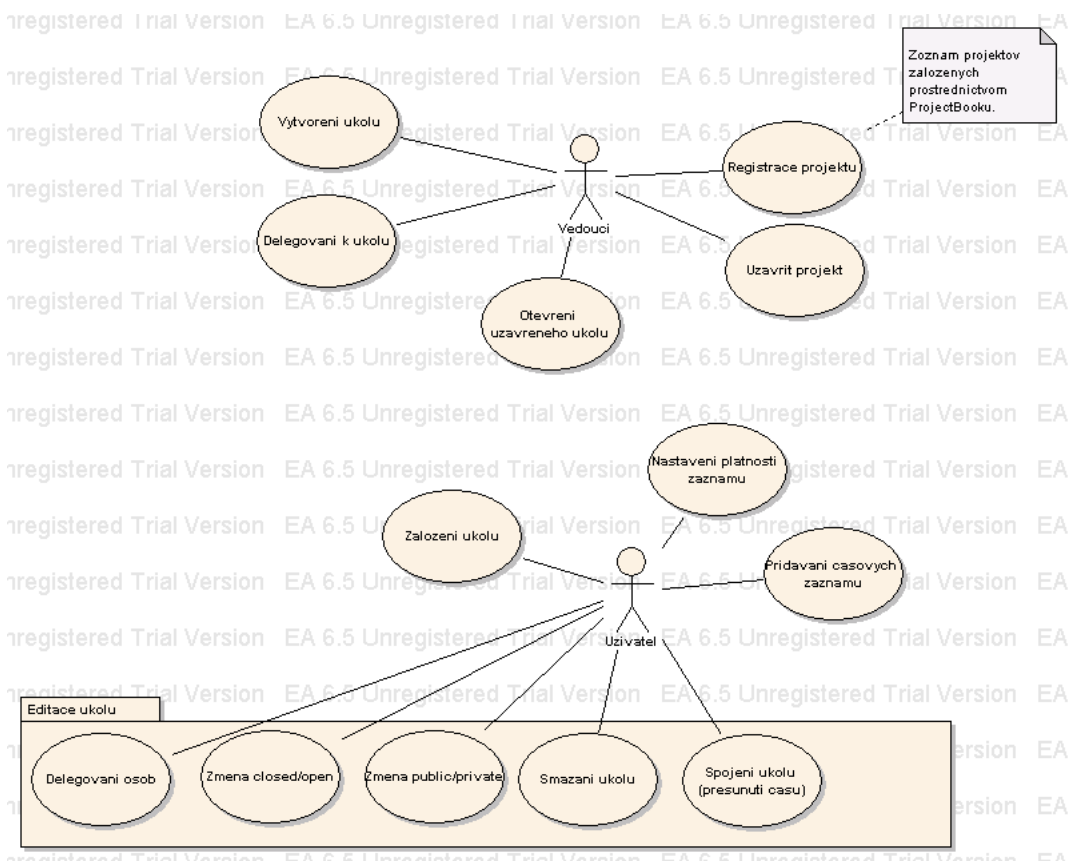
Obr. 4 Užívateľský diagram.



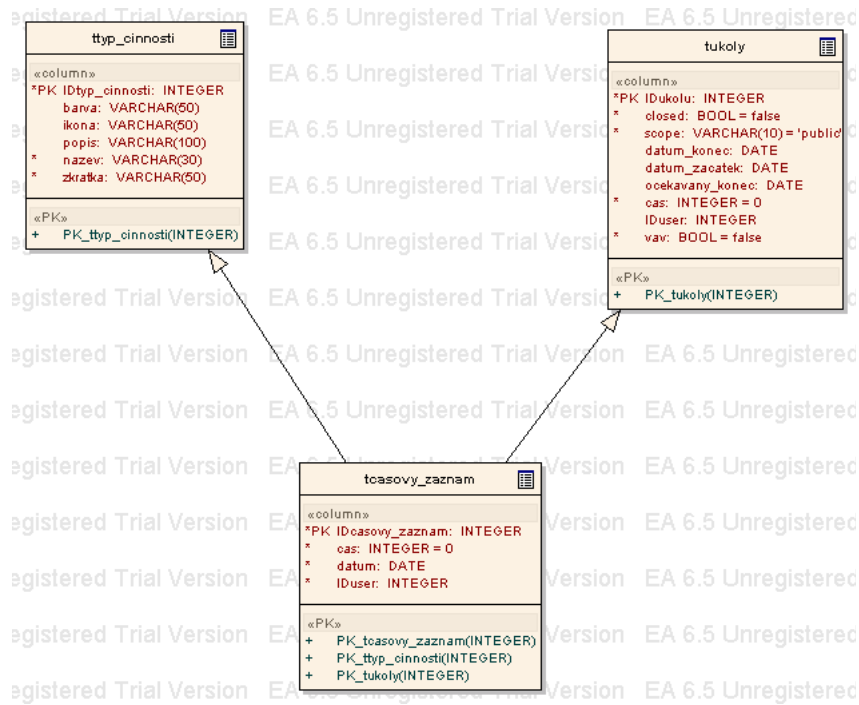
Obr. 5 Web Development Process metodiky WebML.



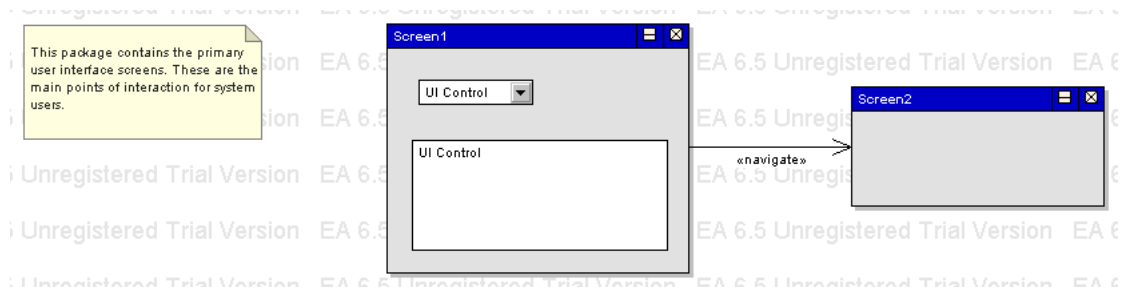
Obr. 6 Zoznam stránok vo forme balíčkov v Enterprise Architecte.



Obr. 7 Typy užívateľov a ich činnosti.



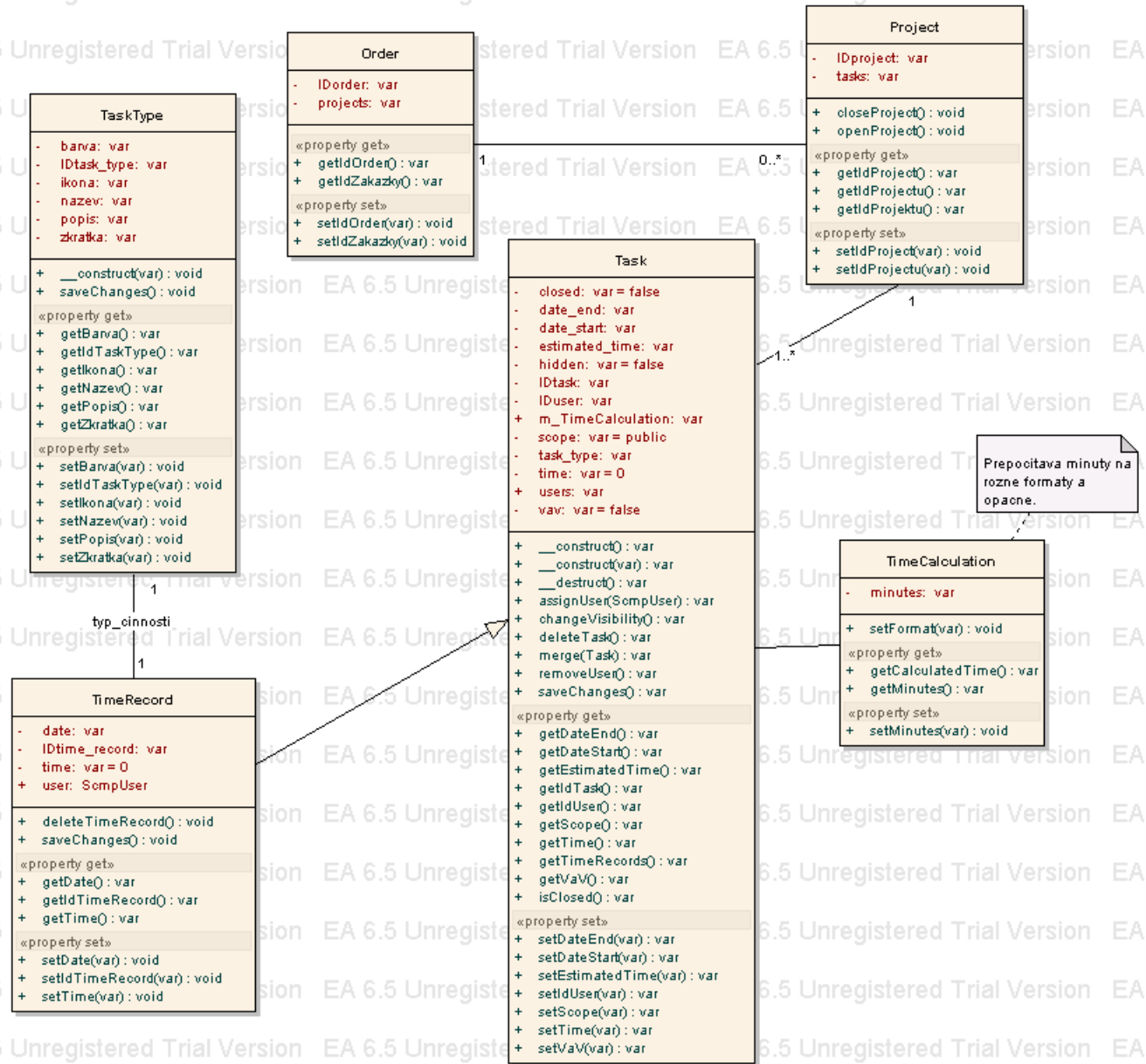
Obr. 8 Dátový model v Enterprise Architecte.



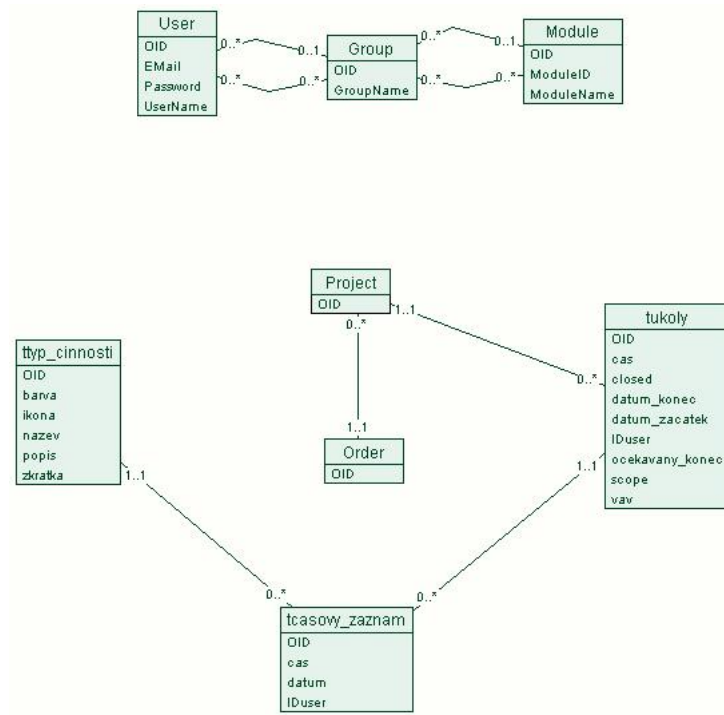
Obr. 9 Prechod medzi dvoma stránkami v Enterprise Architecte (hypertextový model).



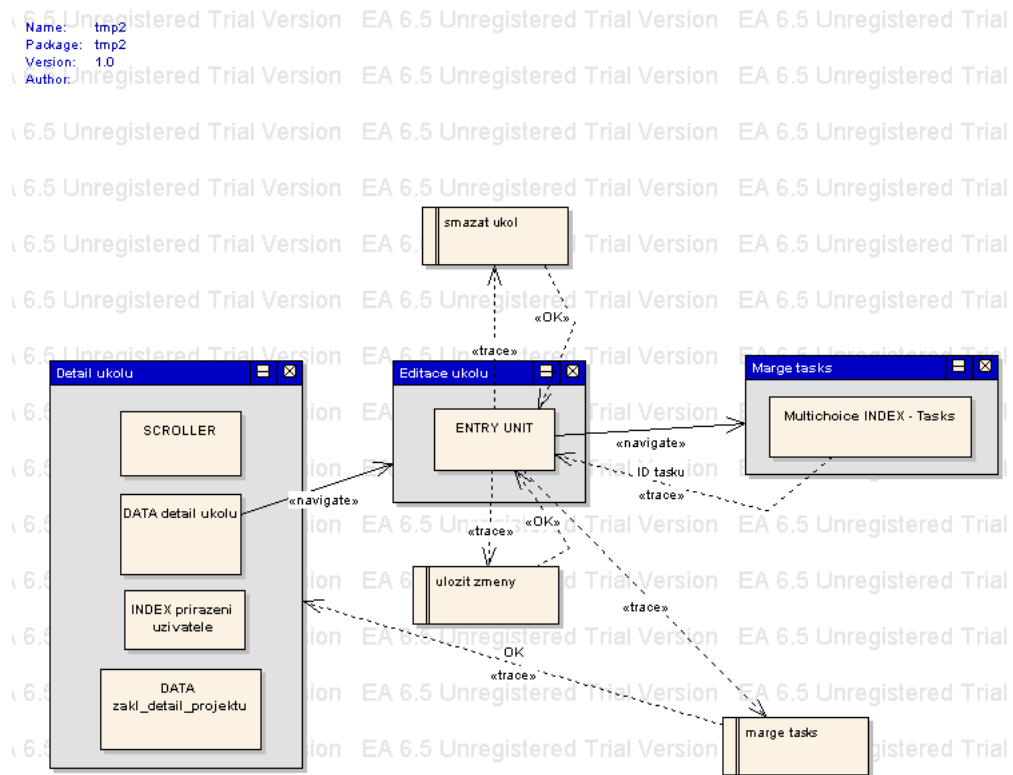
Obr. 10 Stránka Detail projektu.



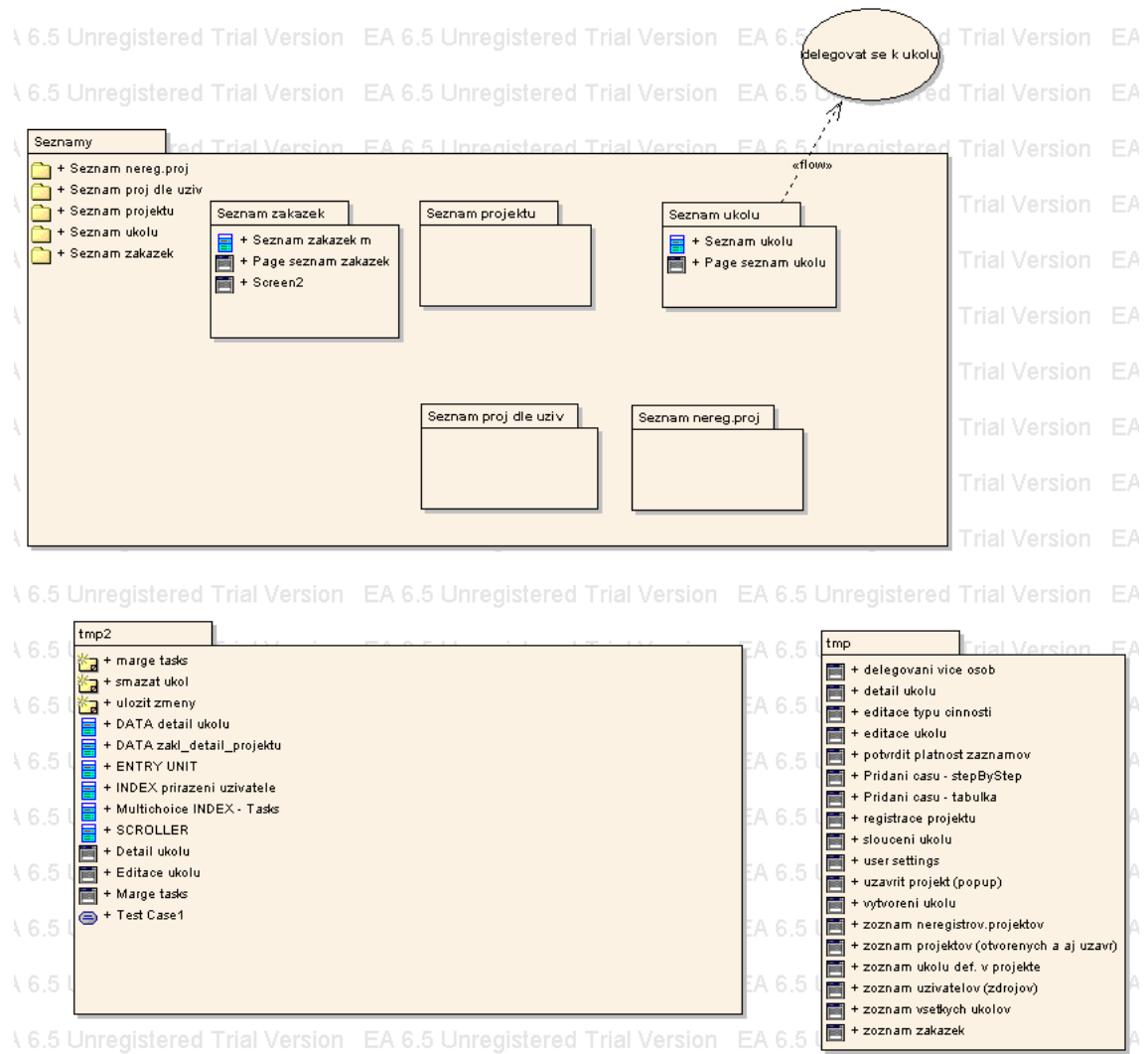
Obr. 11 Class diagram v Enterprise Architecte.



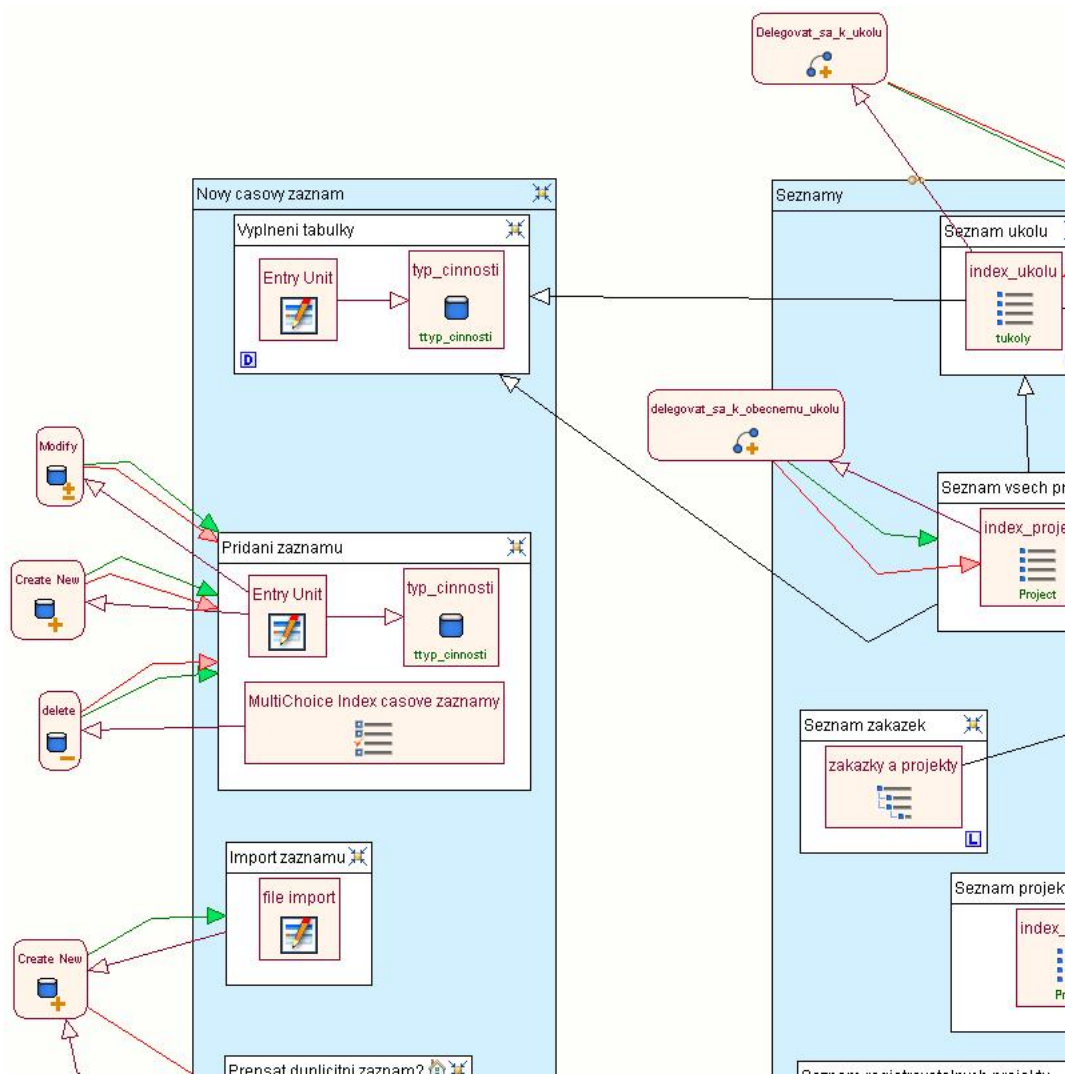
Obr. 12 Jednoduchý class diagram vo WebRatiu.



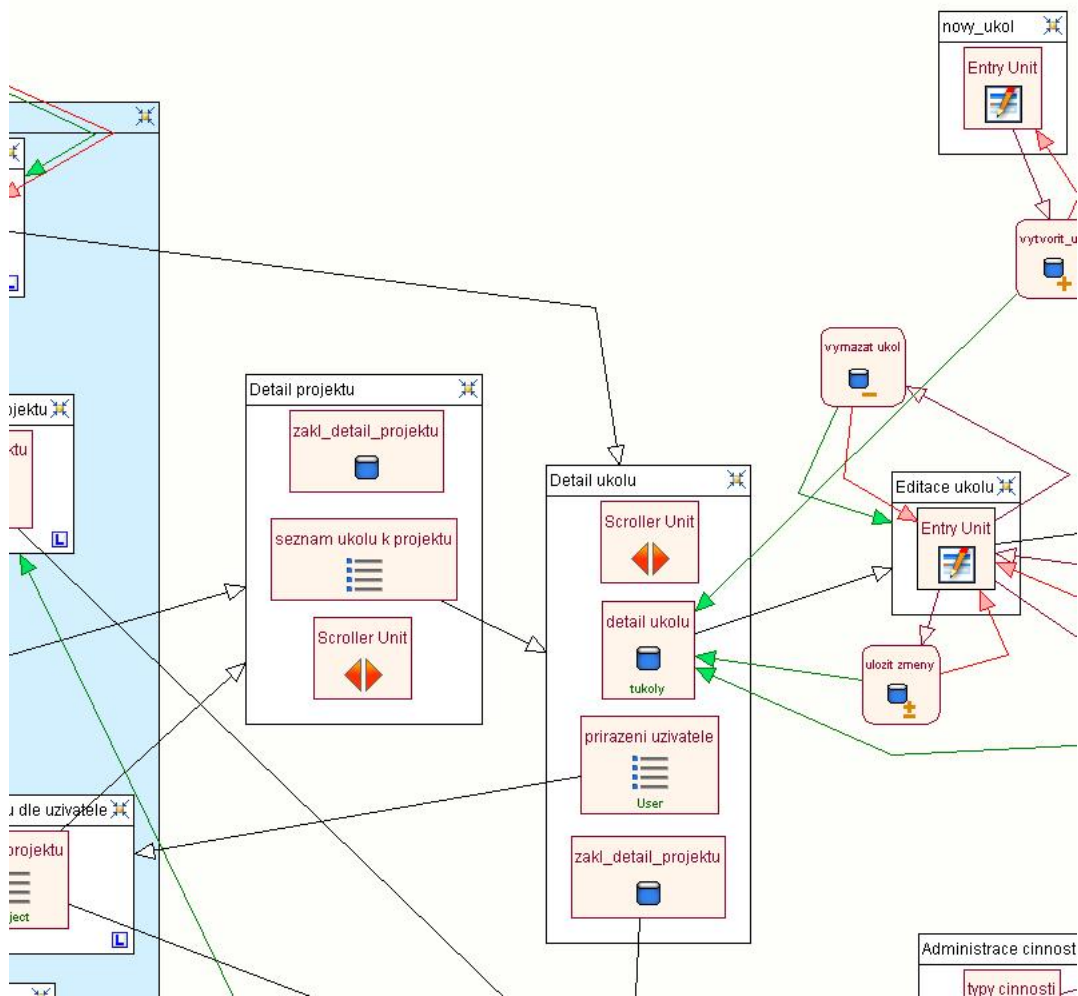
Obr. 13 Hypertextový model v Enterprise Architecte.



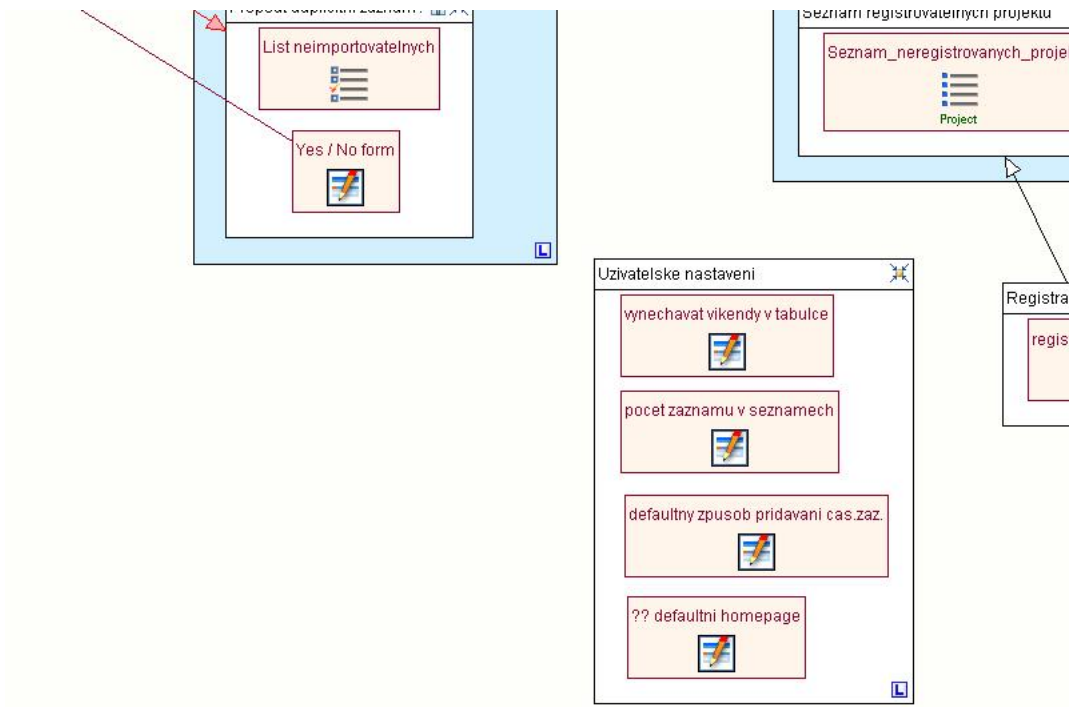
Obr. 14



Obr. 15 Část hypertextového modelu z WebRatia.



Obr. 16 Časť hypertextového modelu z WebRatia.



Obr. 17 Část hypertextového modelu z WebRatia.

```

k?php
require_once ('ScmpUser.php');
require_once ('TimeCalculation.php');

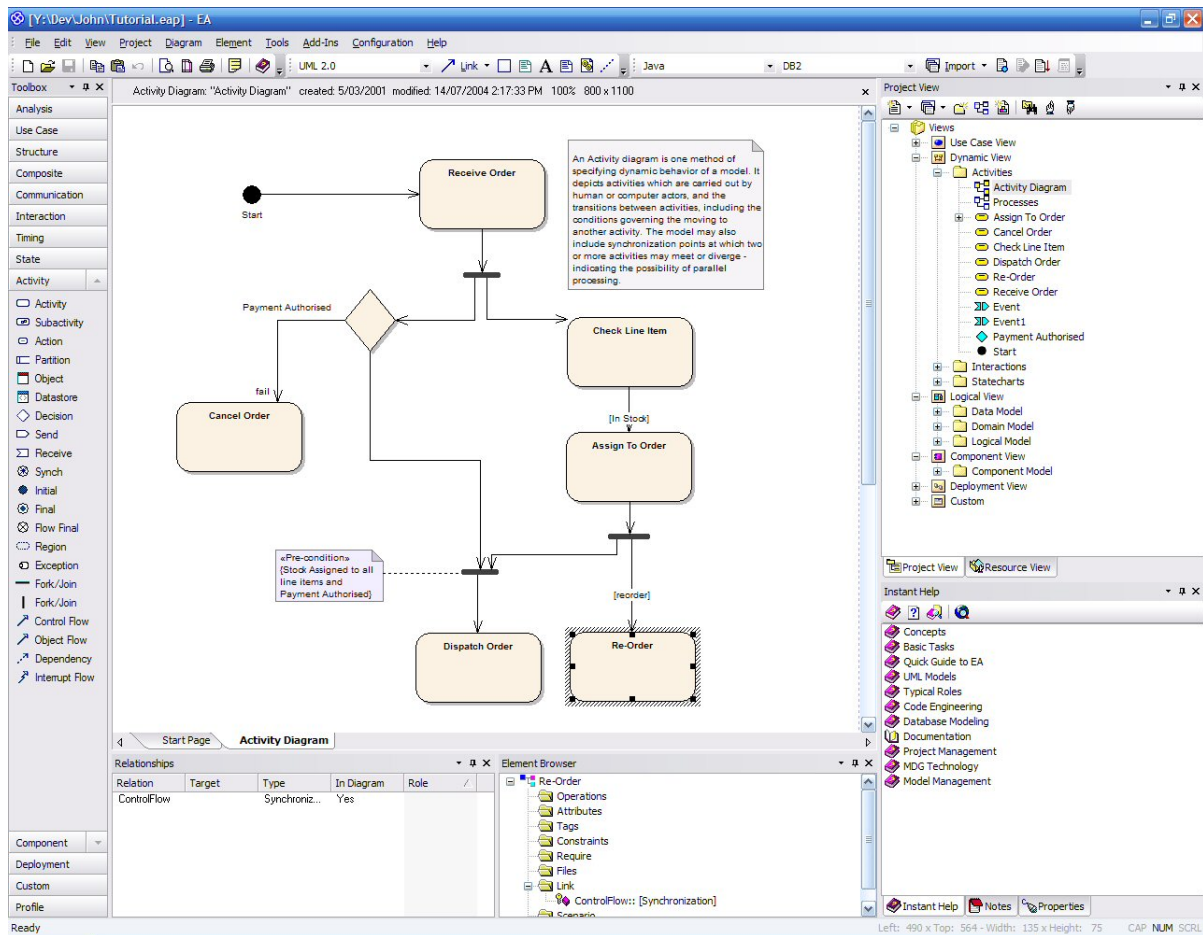
/**
 * @version 1.0
 * @created 20-Dec-2006 11:42:04 AM
 */
class Task
{
    /**
     * uzavreny ukol?
     */
    private $closed = false;
    private $date_end;
    private $date_start;
    private $estimated_time;
    private $hidden = false;
    private $IDtask;
    /**
     * ID uzivatela, ktory ukol zalozil - odpovedny uzivatel
     */
    private $IDuser;
    public $m_TimeCalculation;
    private $scope = public;
    /**
     * typ ukolu, cinnosti
     */
    private $task_type;
    private $time = 0;
    /**
     * pole uzivatelov priradenych k ukolu
     */
    public $users;
    /**
     * vyzkum a vyvoj?
     */
    private $vav = false;

    public function __construct()
    {
    }

    /**
     *
     * @param ITtask    ITtask
     */
    public function __construct($ITtask)
    {
    }
}

```

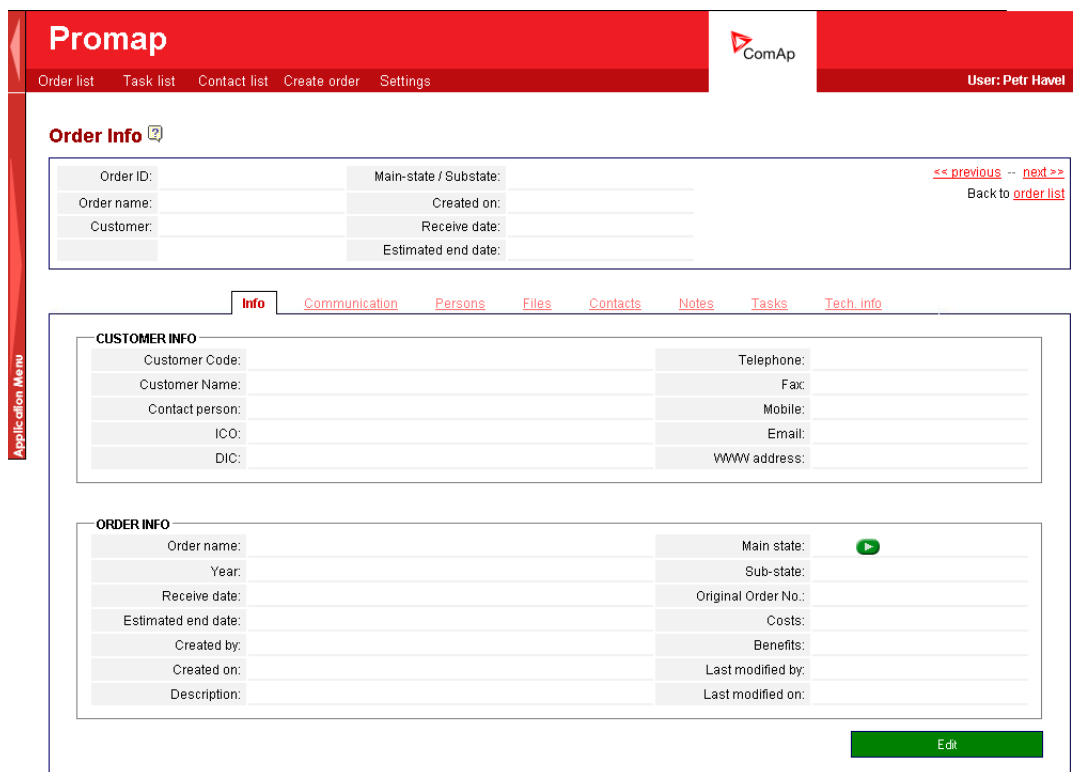
Obr. 18 Trieda Task vygenerovaná Enterprise Architectom v jazyku PHP v5.



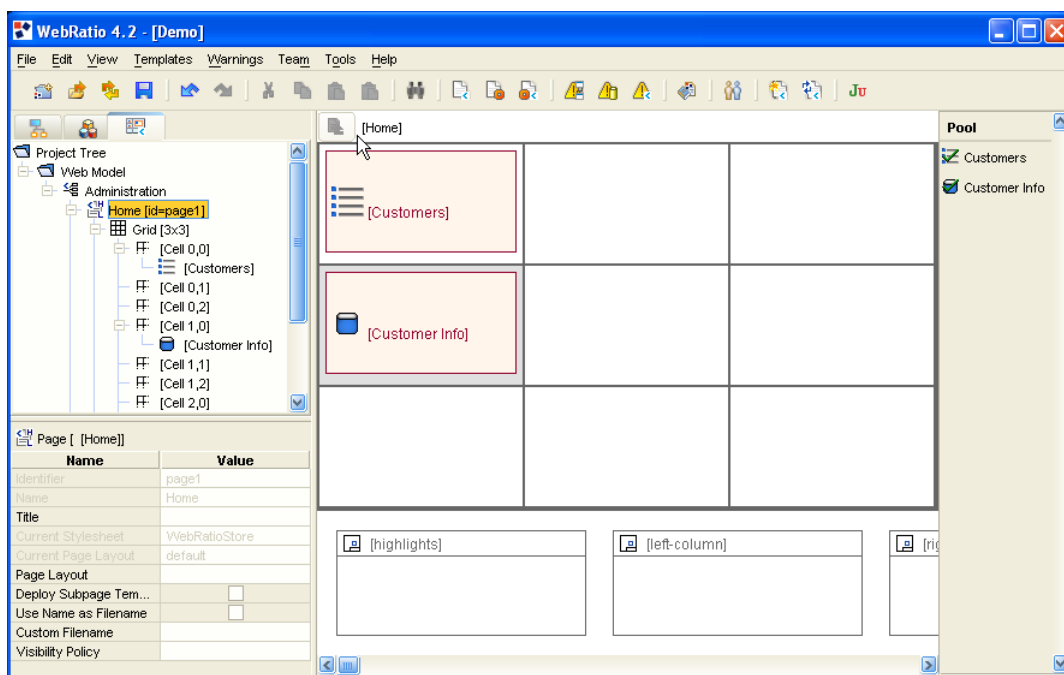
Obr. 19 UML model v Enterprise Architecte.

Class : Customer	Superclass :	Class : Current_Customer	Superclass : Customer
Attributes : Customer_Id Customer_Name Customer_Address Customer_Phone Customer_Credit	Subclass : Current_Customer Prospective_Customer	Attributes : Current_Customer_Initial_Date	Subclass :
	Collaborators:		Collaborators: Account Transaction
	Components :		Components :
Responsibilities : Know_Credit Know_Customer_Information	Associators:	Responsibilities : Check_Account_Balance Check_Account_Transaction Know_Current_Customer	Associators: Account (N) Branch (N)

Obr. 20 Kartyčky s objektmi modelovanými metodikou SOHDM.



Obr. 21 Promap - systém na evidenciu zákaziek v ComAp, s.r.o.



Obr. 22 Prezentačný model vo WebRatiu.