

Porovnanie metód modelovania webových aplikácií*

Patrik Tomčo

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xtomco@stuba.sk

16. októbra 2021

Abstrakt

Modely a modelovacie nástroje sú veľmi často používané softvérovými inžiniermi na vyjadrenie ich myšlienok počas vývoja softvéru. Tieto modely vedú k definícii modelovo-založeného vývojového procesu (MDD: Model-Driven Developement). Počas celej histórie softvérového inžinierstva boli pridávané nové využitia pre modely. Potenciálne benefity využívania modelov sú výrazne väčšie v softvérovej, ako v inej inžinierskej disciplíne. [6]. V MDE (Model-Driven Engineering) sú modely považované za hlavný vývojový artefakt na tvorbu softvéru. [5]

Z toho môžeme vyvodiť, že dôležitosť modelov v MDD je neodmysliteľná a je dôležité vedieť s nimi patrične narábať. Tento článok sa zaoberá popisom a porovnaním MDD metód, ktoré sú esenciálne pre správne a dlhodobé fungovanie softvéru, ako aj pre jeho komplexnosť. Taktiež analyzuje techniky navrhované na špecifikovanie funkčných, dátových a navigačných požiadaviek, ako aj poskytnutých mechanizmov na automatické preloženie týchto požiadaviek do koncepčných modelov. Hlavným cieľom tohto článku je preto pohľad a tieto metódy, využívaných vo vývoji webových aplikácií za účelom poukázania na ich silné a slabé stránky. [8]

Kľúčové slová: koncepčný model, modelmi riadený vývoj, metódy, webová aplikácia, modelmi riadené webové inžinierstvo, softvér

Čo pokladáte za významný problém v tejto oblasti a prečo (opora v literatúre)?

1 Úvod

Modelmi riadený vývoj (MDD: Model-Driven Developement) sa stáva stále viac a viac dôležitou a využívanou metódou rámci softvérového inžinierstva. MDD tvrdí, že softvérové systémy musia byť vyvíjané pomocou modelov. MDD proces zvyčajne začína požiadavkovou fázou, v ktorej sa definujú požiadavkové modely,

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Vladimír Mlynarovič

z ktorých následne vznikne jeden alebo viacero koncepčných modelov 2. Tie majú za úlohu popísať systém bez prihliadnutia na technologické aspekty softvéru a sú neskôr využité v analytickej fáze [8]. Práve týmto modelom a metódam, v ktorých su obsiahnuté, je venovaný tento článok. Presnejšie porovnaniu jednotlivých metód a koncepčných metód z nich pozostavujúcich. Tento článok sa zaoberá popisom rôznych MDWE (Model-Driven Web Engineering) metód. Tieto metódy a koncepčné modely článok porovnáva z pohľadu MDD, ako aj z pohľadu funkcionality a navigácie v rámci webových aplikácií a požiadaviek používateľa na spomenutých stránkach. V dnešnej dobe existuje enormné množstvo metód zaoberajúcich sa vývojom webových aplikácií. Preto by si popísanie všetkých metód dopodrobna vyžadovalo príliš veľké úsilie. Tento článok sa preto venuje len redukovanej množine metód, aby bolo možné sa im viac dopodrobna venovať. Metódy, ktorým sa článok primárne venuje sú OOHDM (Object-Oriented Hypermedia Design Model), OOWS (Object Oriented Web Solutions) a WSDM (Web Site Design Method). Každá z týchto metód predstavuje koncepčné modely, ktoré nám umožňujú popísať webové aplikácie technologicky nezávislým spôsobom. "Tieto metódy boli úspešne aplikované vo vývoji viacerých webových aplikácií, čo je dôkazom toho, že implementácia konštruovania webových aplikácií pomocou koncepčných modelov a ich neskoršie prepísanie do kódu je možné". [8]

Pre porozumenie tejto problematiky je veľmi dôležité vedieť, čo presne koncepčné modely predstavujú. Preto tento článok začne ich popisom. Ďalej bude pokračovať následovne. Sekcia 3 prezentuje prehľad popisovaných MDWE metód a ich bližší popis. Sekcia 4 sa venuje porovnaniu týchto metód a koncepčným modelom z pohľadu MDD a funkcionality a navigácie v rámci webových aplikácií. Sekcia 5 je venovaná metódam a modelom využívajúcim MDA (Model-Driven Architecture) prístup. Sekcia 6 slúži ako sumarizácia všetkého, čo sme zistili o danej problematike a sekcia 7 poskytuje prehľad bibliografie.

Ešte spomenúť MDA prístup

2 Koncepčné modely

Koncepčné modely webových aplikácií špecifikujú jej kompozíciu a navigáciu v nej [4]. Kompozícia web stránky definuje, ktoré stránky tvoria hypertext a ich internú kompozíciu, ako aj možnosti používateľa na zaobchádzanie so systémom. Navigácia definuje rôzne spôsoby, ako môžu byť dané stránky navzájom prepojené linkami, ale aj zobrazenie postupnosti stránok, na základe používania zo strany používateľa, a obsahom vyobrazeným na stránke. Inými slovami, koncepčné modely webových aplikácií špecifikujú jej organizáciu jej front-end rozhrania v podobe stránok, dizajnových elementov, ktoré sú prepojené linkami a uľahčenie navigácie na web stránke a manipulovania s ňou [4].

3 Prehľad MDWE metód

Následuje sekcia venujúca sa popísaniu troch vybraných MDWE (Model-Driven Web Development) metód. Tými sú, ako bolo spomenuté: OOHDM, OOWS , WSDM, Tieto metódy budú popísané chronologicky, podľa toho, v akom roku sa prvýkrát objavili v literatúre. Obrázok č.1 poukazuje na prehľad študovaných metód a chronologické usporiadanie podľa roku prvej publikácie. Tento článok sa venuje hlavne týmto metódam, pretože predstavujú techniky špeciálne vytvorené na špecifikovanie potrieb webových aplikácií. [8]

Vložiť tabuľku s grafmi

3.1 OOHDM: Object Oriented Hypermedia Design Model

OOHDM bolo vyvinuté pánmi Schwabe a Rossi v roku 1994. Bolo to jedno z prvých metodologických riešení pre vývoj webových aplikácií. OOHDM zdôrazňuje separáciu navigačných aspektov softvéru a od iných aspektov ako napríklad koncepčné aspekty a aspekty rozhrania. Ďalšie prístupy boli neskôr inšpirované touto myšlienkou separácií rôznych aspektov. "Nakoniec, je dôležité spomenúť, že OOHDM nie je uzatvorený prístup a je postupne rozširovaný a vylepšovaný." [8]

Proces vývoja tohto prístupu je rozdelený do piatich hlavných fáz.

- Zhromažďovanie požiadaviek. V tejto fáze sú definovaný užívatelia, ktorý používajú webovú aplikáciu, ako aj užívateľské potreby, ktoré musí webová aplikácia podporovať.
- Koncepčný dizajn. Táto fáza pozostáva z koncepcnej schémy, v ktorej su popísané statické systémové aspekty.
- Navigačný dizajn. V tejto fáze musí byť definovaný diagram navigačných tried a diagram navigačnej štruktúry. Prvý reprezentuje statické možnosť navigačného systému. Druhý, na druhej strane, rozširuje prvý diagram vrátane prístupovej štruktúry a navigačného kontextu.
- Abstraktný dizajn rozhrania. Táto fáza definuje opis užívateľského rozhrania abstraktným spôsobom.
- Implementácia. V tejto fáze je webová aplikácia implementovaná. Táto implementácia je založená na predchádzajúcich modeloch.

3.2 OOWS: Object Oriented Web Solutions

Popísanie OOWS

3.3 WSDM: Web Site Design Method

Popísanie WSDM

4 Porovnanie koncepčných modelov

5 Porovnanie koncepčných modelov vyžívajúcich MDA prístup

6 Zhrnutie

7 Bibliografia

8 Úvod1

Toto je moj uvod

Prvy clanok [8]. A druhy clanok [4].

Motivujte čitateľa a vysvetlite, o čom píšete. Úvod sa väčšinou nedelí na časti.

Uveďte explicitne štruktúru článku. Tu je nejaký príklad. Základný problém, ktorý bol naznačený v úvode, je podrobnejšie vysvetlený v časti ?? . Dôležité súvislosti sú uvedené v častiach 10 a 11. Záverečné poznámky prináša časť 12.

Z obr. 1 je všetko jasné.

Aj text môže byť prezentovaný ako obrázok. Stane sa z neho označný plávajúci objekt. Po vytvorení diagramu zrušte znak % pred príkazom

`\includegraphics` označte tento riadok ako komentár (tiež pomocou znaku %).

Obr. 1: Rozhodujúci argument.

9 Nejaka cast

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 9.1), a potom na ešte nejaké (časť 9.1).¹

Môže sa zdať, že problém vlastne nejestvuje [1], ale bolo dokázané, že to tak nie je [2,3]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [7]. Dôležité veci možno *zdôrazniť kurzívou*.

9.1 Njaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna vec
- druhá vec
 - x
 - y

Ten istý zoznam, len číslovaný:

¹Niekedy môžete potrebovať aj poznámku pod čiarou.

1. jedna vec
2. druhá vec
 - (a) x
 - (b) y

9.2 Ešte nejaké vysvetlenie

Veľmi dôležitá poznámka. Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

10 Dôležitá časť

11 Ešte dôležitejšia časť

12 Záver

Literatúra

- [1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [2] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [3] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.
- [4] Vassiliki Gkantouna, Athanasios Tsakalidis, and Giannis Tzimas. Mining interaction patterns in the design of web applications for improving user experience. ACM, July 2016.
- [5] Thiago Gottardi and Rosana Teresinha Vaccare Braga. Evaluating the ability of developers to use metamodels in model-oriented development. IEEE, May 2019.
- [6] B. Selic. The pragmatics of model-driven development. 20(5):19–25, September 2003.
- [7] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. http://www.sei.cmu.edu/productlines/frame_report/.
- [8] Pedro Valderas and Vicente Pelechano. A survey of requirements specification in model-driven development of web applications. 5(2):1–51, May 2011.