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 môžme 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 su 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 preto je pohľad a tieto metódy, využívaných vo vývoji webových aplikácií za účelom poukázania na ich silné a slabé stránky. [8]

Čím sa zaoberá?
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 vrámci softvérového inžinierstva. MDD tvrdí, že softvérové systémy musia byť vyvíjané pomocou modelov. MDD proces zvyčajne začína požiadavkovou fázou, v ktorej sa definujú požiadavkové modely, z ktorých následne vzikne jeden alebo viacero koncepčných modelov 2.

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

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 taktiež zaoberá popísaním rôznych MDWE (Model-Driven Web Engineering) metód. Tieto metódy a koncepčné modely článok porovnáva z pohľadu MDD, ako aj z pohľadaju funkcionality a navigácie v rámci webových aplikácií a požiadaviek používateľa na spomenutých stránkach. Metódy, ktorým sa článok primárne venuje sú OOHDM(Obejct-Oriented Hypermedia Design Model), WSDM(Web Site Design Method), OOWS(Object Oriented Web Solutions). Každá z týchto metód predstavuje koncepčné modely, ktoré nám umožnujú 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 porovaniu týchto metód a koncepčným modelom z pohľadu MDD a funkcionality a navigácie v rámci webových aplkácií. Sekcia 5 je venovaná metódam a modelom využivajúcim MDA (Model-Driven Architecture) prístup. Sekcia 6 slúži ako sumarizácia všetkého, čo sme zistili o danej problematike a sekcia 7 poskytuje prehľad bibliografie.

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.

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 elexntov, ktoré sú prepojené linkami an uľahčenie navigácie na web stránke a manipulovania s ňou [4].

3 Prehľad MDWE metód

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 Nejaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna vec
- druhá vec
  - x

[1] Niekedy môžete potrebovať aj poznámku pod čiarou.

- y

Ten istý zoznam, len číslovaný:

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.