

Porovnanie metód modelovania webových aplikácií*

Patrik Tomčo

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

14. decembra 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 Development). 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. [8] V MDE (Model-Driven Engineering) sú modely považované za hlavný vývojový artefakt na tvorbu softvéru. [3] Z toho je možné 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, ohodnotenie z MDD perspektívy a popisuje silné a slabé stránky každej metódy. Hlavným cieľom tohto článku je preto pohľad a tieto metódy, využívaných vo vývoji webových aplikácií za účelom poukázania na ich silné a slabé stránky. [9]

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

1 Úvod

Modelmi riadený vývoj (MDD: Model-Driven Development) 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, z ktorých následne vznikne jeden alebo viacero koncepčných modelov [2]. Tie majú

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

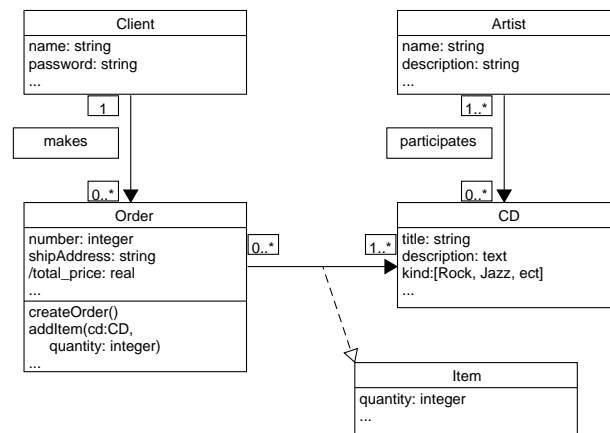
za úlohu popísať systém bez prihliadnutia na technologické aspekty softvéru a sú neskôr využité v analytickej fáze [9]. Práve na tieto modely a metódy, v ktorých su obsiahnuté, sa tento článok zameriava. Presnejšie porovnaniu jednotlivých metód a koncepčných metód z nich pozostavujúcich. Tento článok sa zaoberá popísaním rôznych MDWE (Model-Driven Web Engineering) metód. 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 zaoberá len redukovanou množinou metód, aby bolo možné sa im viac dopodrobna venovať. Metódy, ktorými sa článok primárne zaoberá 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". [9]

Pre porozumenie tejto problematiky je veľmi dôležité vedieť, čo presne koncepčné modely predstavujú. Preto začne tento článok ich popisom. Ďalej bude článok pokračovať nasledovne. Sekcia 3 prezentuje prehľad popisovaných MDWE metód a ich bližší popis. Sekcia 4 sa venuje porovnaniu týchto metód z pohľadu MDD a funkcionality, práce s dátami a navigácie v rámci webových aplikácií. Sekcia 5 sa bližšie analyzuje MDA (Model-Driven Architecture) prístup, ktorý je neskôr využitý v nasledujúcej sekcii. Sekcia 6 je venovaná porovnávaniu metód a modelov využívajúcich spomenutý MDA prístup. V sekcii 7 sa nachádzajú reakcie na prednášky. Sekcia 8 slúži ako sumarizácia všetkého, čo sme zistili o danej problematike a posledná sekcia poskytuje prehľad bibliografie.

2 Koncepčné modely

Ako bolo spomenuté, táto sekcia sa venuje definícii koncepčných modelov. Pre správne porozumenie neskoršej problematiky je znalosť týchto modelov veľmi dôležitá.

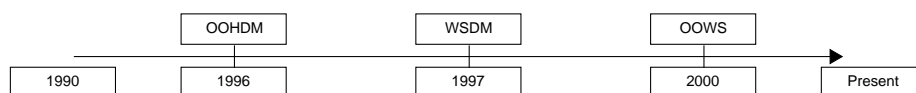
Koncepčné modely webových aplikácií špecifikujú jej kompozíciu a navigáciu v nej [2]. Kompozícia web stránky definuje, ktoré stránky tvoria hypertext a ich internú kompozíciu, ako aj možnosti používateľa na zaobchádzanie so systémom. Navigácia definuje rôzne spôsoby, ako môžu byť dané stránky navzájom 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 organizáciu jej front-end rozhrania v podobe stránok, dizajnových elementov, ktoré sú prepojené linkami na uľahčenie navigácie na web stránke a manipulovania s ňou. [2] Obrázok č.1 predstavuje príklad koncepčného modelu pre obchod s CD.



Obr. 1: Príklad koncepčného modelu pre obchod s CD (upravený [7]).

3 Prehľad MDWE metód

Nasleduje sekcia zaoberajúca sa popisom troch vybraných MDWE (Model-Driven Web Development) metód. Tými sú, ako bolo spomenuté: OOHDM, OOWS, WSDM. Obrázok č.2 predstavuje prehľad študovaných metód a chronologické usporiadanie podľa roku prvej publikácie. Tieto metódy ale nebudú popísané chronologicky, ale podľa ich základnej vnútornej podobnosti. Preto budú najprv popísané objektovo-orientované metódy (OOHDM a OOWS) a potom metóda WSDM. Tento článok analyzuje hlavne tieto metódy, pretože predstavujú techniky špeciálne vytvorené na špecifikovanie potrieb webových aplikácií. [9]



Obr. 2: Chronologický prehľad MDWE metód (upravený [9]).

3.1 OOHDM: Object Oriented Hypermedia Design Model

OOHDM metóda bola vyvinutá pánmi Daniel Schwabe a Gustavo Rossi v roku 1994. Bolo to jedno z prvých metodologických riešení pre vývoj webových aplikácií. OOHDM zdôrazňuje separáciu navigačných aspektov softvéru 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.", tvrdí Pedro Valderas, autor článku o porovnaní požiadaviek MDWE metód. [9]

Proces vývoja tohto prístupu je rozdelený do piatich hlavných fáz:

- Zhromažďovanie požiadaviek. V tejto fáze sú definovaní používatelia, ktorí používajú webovú aplikáciu, ako aj používateľské potreby, ktoré musí webová aplikácia podporovať.
- Koncepčný dizajn. Táto fáza pozostáva z koncepcnej schémy, v ktorej sú popísané statické systémové aspekty.
- Navigačný dizajn. V tejto fáze musí byť definovaný diagram navigačných tried a diagram navigačnej štruktúry. Prvý diagram reprezentuje statické možnosti navigačného systému. Druhý, na druhej strane, rozširuje prvý diagram o prístupové štruktúry a navigačný kontext.
- Abstraktný dizajn rozhrania. Táto fáza definuje opis používateľského rozhrania abstraktným spôsobom.
- Implementácia. V tejto fáze je webová aplikácia implementovaná. Táto implementácia je založená na predchádzajúcich modeloch. [9]

3.2 OOWS: Object Oriented Web Solutions

Ďalej nasleduje novšia metóda s názvom OOWS. Síce sa nejedná o nasledujúcu metódu rámci chronologickej postupnosti, ale článok ju popisuje priamo po OOHDm metóde, pretože obidve tieto metódy majú rovnaký základ, a to objektovo-orientovaný prístup.

Táto metóda bola prvýkrát prezentovaná pánmi Joan Fons a Oscar Pastor v roku 2000. Ide o rozšírenie objektovo-orientovaného prístupu pri vývoji webových aplikácií. Narozdiel od OO-H (object-oriented hypermedia) prístupu, táto metóda je založená iba na objektovo-orientovanej báze. "Toto robí OOWS jednu z mála MDD metód, ktoré poskytujú podporu pre automatické generovanie rozhraní webových aplikácií, ako aj plne operatívnu funkčnosť vyjadrenú z modelov." [9]

Vývojový proces obsiahnutý v tomto prístupe je rozdelený na tri hlavné fázy:

- Analýza požiadaviek. V tejto fáze sú špecifikované požiadavky webovej aplikácie pomocou modelov, ktoré sú založené na koncepte úloh (concept of task). Je dôležité poznamenať, že táto fáza bola uvedená do praxe až po pridaní rôznych rozšírení do tejto metódy.
- Špecifikácia systému. Táto časť pozostáva z popisu webovej aplikácie na koncepcnej úrovni. Na dosiahnutie tohto výsledku sú navrhnuté rôzne modely:

1. Objektový model na predpísanie statickej štruktúry webovej aplikácie
2. Funkčné a dynamické modely na predpísanie správania sa webovej aplikácie.
3. Navigačné a prezentačné modely na predpísanie používateľského rozhrania webovej aplikácie

- Generácia riešení. V poslednej fáze je webová aplikácia automaticky generovaná z modelov spomenutých v predchádzajúcich fázach. Na dosiahnutie tohto cieľa sú ale potrebné rôzne nástroje. [9]

3.3 WSDM: Web Site Design Method

Nakoniec nasleduje posledná metóda s názvom WSDM. Táto sa od predošlých líši, mimo iného, aj v tom, že nie je založená na objektovo-orientovanom prístupe.

WSDM metóda bola vyvinutá pánmi De Troyer a Leune v roku 1998. Ide teda o nepatrne novšiu metódu ako OOHDM ale staršiu ako OOWS.

WSDM definuje webové aplikácie popisovaním požiadaviek rôznych skupín používateľov, ktorí s ňou zaobchádzajú. Ide teda o metódu, ktorá definuje používateľa ako centrum tohto prístupu. Je to jedna z prvých metód, ktorá prihliada na problém vysokej rôznorodosti používateľov vo webových aplikáciách. [9]

Vývojový proces tohto prístupu je rozdelený do piatich hlavných fáz:

- Špecifikácia poslanca. V tejto fáze je potrebné definovať účel a predmet webovej aplikácie. Tatiež musí byť poukázané na cieľové publikum, pre ktoré je daná aplikácia určená.
- Modelovanie publika. V druhej fáze sú špecifikovaní používatelia a sú následne rozdelení do skupín. Toto je realizované za účelom študovania systémových požiadaviek na základe každej skupiny používateľov.
- Koncepčný dizajn. V tretej fáze je vytvorený diagram tried a navigačný model. Diagram tried reprezentuje statický model systému a navigačný model popisuje možnosti navigácie vo webovej aplikácii.
- Implementačný dizajn. Počas štvrtej fázy sú definované koncepčné dizajnové modely. Tieto modely sú následne doplnené o informácie, potrebné na samotnú implementáciu. Takéto informácie sú napríklad model štruktúry stránky alebo prezentačný model. [9]

4 Porovnanie MDWE metód

Táto časť je dedikovaná porovnaniu analyzovaných metód a to OOHDM, OOWS a WSDM. Článok sa zameriava na špecifikovanie piatich hlavných elementov pri každej metóde. Týmto elementami sú funkčné požiadavky, dátové požiadavky, navigačné požiadavky, ohodnotenie z MDD perspektívy a ohodnotenie silných a slabých stránok danej metódy. Funkčné, dátové a navigačné požiadavky budú analyzované na základe špecifických kritérií, ktoré sú rôzne pre všetky typy požiadaviek. Tieto kritériá sú nasledovné:

Funkčné požiadavky:

- overenie vstupov - FP1
- presná postupnosť operácií - FP2
- vzťahy medzi vstupmi a výstupmi - FP3

Dátové požiadavky:

- možnosti prístupu k dátam - DP1
- dátové entity - DP2

Navigačné požiadavky:

- vyobrazenie informácií - NP1
- možnosti navigácie k informáciám - NP2
- funkčnosť spojená s informáciami - NP3

Tieto kritériá môžu byť buď podporované, čiastočne podporované alebo nepodporované. Prehľad týchto kritérií sa nachádza vo forme tabuľky na konci každej metódy. Toto porovnanie je redukované porovnanie toho, ktoré sa nachádza v článku "A Survey of Requirements Specification in Model-Driven Development of Web Applications" od autorov Pedro Valeras a Vicente Pelechano [9].

4.1 OOHDM

Funkčné požiadavky sú explicitne podporované touto metódou. Navrhovaná technika na ich špecifikovanie je takzvaný "use case" diagram. Ďalšie špecifikovanie činností spojených s týmto diagramom umožňuje popis postupnosti činností, ktoré vykoná používateľ alebo systém. Čiže táto metóda zachytáva presnú postupnosť činností, ktoré sa vykonávajú. Preto je kritérium FP2 považované za podporované. Tento "use case" diagram môže byť taktiež rozšírený o pred-podmienky a po-podmienky, ktoré poskytujú podporu pre zachytenie vstupov a definovanie vzťahov medzi vstupmi a výstupmi. Preto sú kritériá FP1 a FP3 považované za podporované. [9]

Dátové požiadavky tejto metódy sú čiastočne podporené UID diagramami. UID (User interaction diagram) diagramy graficky popisujú interakcie medzi používateľom a systémom bez prihliadnutia na špecifické aspekty používateľského rozhrania. Tiež presne popisujú dáta, ku ktorým má používateľ prístup, aj keď dáta, ktoré môžu byť sprístupnené systému nie sú správne popísané. Rovnako, UID špecifikuje údaje, ktoré systém musí uložiť. Ide práve o tie, ktoré sú prístupné používateľovi. Dáta, ktoré existujú iba pre systémové operácie, nie sú špecifikované. Preto je kritérium DP1 čiastočne podporované a kritérium DP2 nepodporované. [9]

Navigačné požiadavky sú explicitne podporované touto metódou. Navrhovaná technika na ich definovanie je znova založená na UID diagramoch. Rôzne interakcie medzi používateľom a systémom popisujú informácie, ktoré sa majú zobraziť. O čo viac, tieto interakcie a vzťahy medzi nimi povoľujú používateľovi jednoduchú navigáciu medzi týmito informáciami. Preto môžu byť kritériá NP1 a NP2 považované za podporované. Nakoniec, interakcie sú doplnené o mechanizmy, ktoré poukazujú na možnosť aktivovania rôznych funkcií. Preto je kritérium NP3 považované za podporované. [9]

Ohodnotenie z MDD perspektívy: Táto metóda prezentuje usmernenia za účelom pomoci pri definícii koncepcnej schémy z UID. Tie sú definované zo skupiny pravidiel, ktoré boli prednesené autormi Patrícia Vilain a Daniel Schwabe. Tieto pravidlá poukazujú na možnosť definovania koncepcnej schémy z UID diagramu. Nevýhodou je, že sú popísané pomocou klasického jazyka, bez predom

definovaných formálnych pravidiel. Preto často dochádza k mierne nejednoznačnosti. [9]

OOHDM			
	Funkčné požiadavky	Dátové požiadavky	Navigačné požiadavky
Explicitná podpora	Áno	Nie	Áno
Technika	Případy použití	-	UID
Kritérium:	FP	DP	NP
	Overenie vstupov ✓✓	Prístup k dátam ✓	Vyobrazenie informácií ✓✓
	Postupnosť operácií ✓✓	Dátové entity X	Navigácia k informáciám ✓✓
	Vzťahy medzi vstupmi a výstupmi ✓✓	-	Funkčnosť a informácie ✓✓

✓✓ - Podporované ✓ - Čiastočne podporované X - Nepodporované

Obr. 3: Prehľad kritérií OOHDM metódy (upravený [9]).

Obrázok č. 3 slúži ako prehľad analyzovaných kritérií pre OOHDM metódu.

4.2 OOWS

Funkčné požiadavky sú plne podporované touto metódou. Tie sú špecifikované pomocou takzvaných "listových úloh". V tejto metóde, overenie vstupov a vzťahy medzi vstupmi a výstupmi môžu byť definované ako ohraničenie v takzvanej charakterizačnej šablóne. Preto sú kritériá FP1 a FP3 považované za podporované. Na druhej strane, postupnosť operácií môže byť popísaná ako postupnosť systémových aktivít. Tie sú definované v diagrame aktivít. To znamená, že kritérium FP2 je taktiež považovaná za podporované. [9]

Dátové požiadavky sú explicitne podporované touto metódou. Na špecifikovanie sú využité informačné šablóny. Šablóny určené rôznym entitám povolujú poukazovať na znaky, ktoré musí systém uložiť pre každú entitu a vzťahy medzi nimi. Tento bod je podložený pripojením typu atribútu na ďalšiu entitu. Preto sú kritériá DP1 a DP2 považované za podporované. Dátová dostupnosť je popísaná ako spojenia systémových aktivít definovaných v diagrame aktivít. To znamená, že dáta môžu byť sprístupnené systémovým aktivitám. Dátová dostupnosť je taktiež charakterizovaná používateľmi pridruženými ku každej úlohe z charakterizačnej šablóny. [9]

Navigačné požiadavky sú taktiež explicitne podporované OOWS metódou. Tie sú popísané v diagrame aktivít. Na popis navigačných požiadaviek je dôležité zaviesť pojem IPs (Interaction Point). Tieto interakčné body reprezentujú moment počas výkonu úlohy, v ktorom si systém a používateľ vymieňa informácie. Na jednej strane, časové vzťahy tejto metódy všeobecne opisujú navigáciu používateľa za informáciami. Toto je zabezpečené vďaka diagramom aktivít, ktorý predpisuje navigáciu ako postupnosť interakčných bodov. Preto je krité-

Silné a slabé stránky: Hlavná silná stránka tohto prístupu je založená na využívaní dobre-známeho "use case" popisu, ktorý povoľuje definovanie funkčných požiadaviek webových aplikácií vďaka pomoci UID diagramov. Tie povoľujú vizuálny popis navigačných a dátových požiadaviek. OOHDM je tiež jedna z mála metód, ktoré poskytujú spomenutý postup špeciálne sústredený na webové aplikácie. Silná stránka tejto metódy je taktiež jej slabou stránkou, a to v tom zmysle, že iba UID technika je definovaná pre MDD použitie. Usmernenia na preberanie koncepčných modelov sú taktiež zahrnuté iba v UID diagramoch. [9]

rium NP2 považované za podporované. Zobrazené informácie sú definované ako entity pridružené ku každému výstupu interakčných bodov. Navyše sú tieto informácie detailne popísané pomocou šablón pripojených k týmto interakčným bodom. Z toho môže usúdiť, že kritériá NP1 a NP3 sú taktiež podporované. [9]

Ohodnotenie z MDD perspektívy: Techniky navrhované OOWS metódou na špecifikovanie požiadaviek sa skladajú z diagramov, ako napríklad diagram aktivít a z textovej šablóny. No aj keď ide o textovú šablónu, tak tá má striktné definovanú štruktúru. Ide o vylepšenie nedostatkov, s ktorými je možné stretnúť sa v OOHDM metóde. Toto precízne definovanie dovoľuje celému OOWS požiadavkovému modelu zapadnúť do presného metamodelu, v ktorom sú popísané všetky koncepty potrebné na špecifikovanie podporovaných požiadaviek. [9]

OOWS			
	Funkčné požiadavky	Dátové požiadavky	Navigačné požiadavky
Explicitná podpora	Áno	Áno	Áno
Technika	Diagram aktivít	Informačná šablóna	Diagram aktivít
Kritérium:	FP	DP	NP
	Overenie vstupov ✓✓	Prístup k dátam ✓✓	Vyobrazenie informácií ✓✓
	Postupnosť operácií ✓✓	Dátové entity ✓✓	Navigácia k informáciám ✓✓
	Vzťahy medzi vstupmi a výstupmi ✓✓	-	Funkčnosť a informácie ✓✓

✓✓ - Podporované ✓ - Čiastočne podporované X - Nepodporované

Obr. 4: Prehľad kritérií OOWS metódy (upravený [9]).

Silné a slabé stránky: Hlavnou silnou stránkou tejto metódy je, že je špeciálne vytvorená na špecifikovanie webových aplikácií. Taktiež ponúka nové techniky orientované na podporu aspektov súvisiacich s webovými aplikáciami, ako napríklad navigácia. Ďalšia výhoda tejto metódy je, že navigačné schopnosti sú popísané globálne. Toto sa líši od iných techník, ktoré sú založené na takzvaných prípadoch použitia (use cases), ktoré podporujú iba obmedzený pohľad na aspekt navigácie, keďže navigačné požiadavky sú popísané jednotlivo pre všetky prípady využitia. Neposlednou výsadou tejto metódy je to, že je plne orientovaná na podporu MDD prístupu

webových aplikácií.

Hlavnou nevýhodou tejto metódy je ale možnosť prívelkej zložitosti požiadavkových modelov. Na dôkladné popísanie týchto modelov je potrebné mnoho zložitých elementov. Taktiež je vhodné spomenúť, že diagram aktivít je doplnený o grafické elementy, ktoré nepochádzajú priamo zo štandardného UML diagramu aktivít. Preto tento diagram môže vyzeráť zložito pre netrénovaných ľudí. [9] Obrázok č. 4 slúži ako prehľad analyzovaných kritérií pre OOWS metódu.

4.3 WSDM

Funkčné požiadavky tejto metódy sú bezprostredne podporované. Sú špecifikované pomocou neformálneho písomného popisu, ktorý je priradený každej triede publika, ale aj pomocou diagramu úloh. Tento diagram úloh povoľuje špecifikáciu postupnosti úloh aplikácie. Čiže kritérium FP2 je považované za podporované. Ďalšia časť kritérií nie je podporovaná diagramami úloh, ale môžu byť definované písomným popisom. Autori tejto metódy ďalej ale približujú, ako tento popis využiť na popísanie nepodporovaných kritérií. Preto ich môžeme považovať za čiastočne podporované. [9]

Dátové požiadavky sú explicitne podporované WSDM metódou. Tie sú taktiež špecifikované pomocou písomných popisov s využitím klasického jazyka. Preto nie je presne definované, aké konkrétne dátové požiadavky táto metóda vyžaduje. Záleží na softvérovom analytikovi, aby využil voľnosť tohto jazyka a popísal tieto požiadavky. Preto sú všetky kritériá považované za čiastočne podporované. [9]

Navigačné požiadavky sú presne podporované touto metódou. Rozoberané kritériá tohto článku môžu byť definované opätovne pomocou písomného popisu a diagramu úloh. Diagramy úloh môžu byť využité na špecifikovanie postupnosti interakcií, ktoré vykonáva používateľ v súvislosti so systémom. Po definitívnom pomenovaní týchto častí interakcií je možné popísať, ako dokáže používateľ vyhľadať informácie. Preto je kritérium NP2 považované za podporované. Vďaka týmto diagramom je taktiež realizovateľné popísanie toho, aký môže mať používateľ prístup k daným informáciám a funkciám. Preto je kritérium NP3 považované za podporované. Aspekt vyobrazovania informácií môže byť doasiahnutí využitím písomného popisu. Kritérium NP1 je preto len čiastočne podporované. [9]

Ohodnotenie z MDD perspektívy: Špecifikácia požiadaviek navrhnutá WSDM metódou je založená na klasifikácii používateľov a na popise požiadaviek, spätých s každým používateľom. Definícia používateľských požiadaviek je realizovaná pomocou písomného popisu. Na vylepšenie tohto problému sú taktiež navrhnuté diagramy úloh. Ten je využitý na ďalšie špecifikovanie a to navigačného modelu a diagramu tried na základe predom spomenutých požiadaviek. [9]

WSDM			
	Funkčné požiadavky	Dátové požiadavky	Navigačné požiadavky
Explicitná podpora	Áno	Áno	Áno
Technika	Klasický jazyk	Klasický jazyk	Klasický jazyk
Kritérium:	FP	DP	NP
	Overenie vstupov ✓	Prístup k dátam ✓	Vyobrazenie informácií ✓
	Postupnosť operácií ✓✓	Dátové entity ✓	Navigácia k informáciám ✓✓
	Vzťahy medzi vstupmi a výstupmi ✓✓	-	Funkčnosť a informácie ✓✓

✓✓ - Podporované ✓ - Čiastočne podporované X - Nepodporované

Obr. 5: Prehľad kritérií WSDM metódy (upravený [9]).

Obrázok č. 5 slúži ako prehľad analyzovaných kritérií pre WSDM metódu.

Silné a slabé stránky: Hlavnou silnou stránkou WSDM metódy je, že berie veľký ohľad na používateľa a na jeho požiadavky. Toto je veľmi dôležitý aspekt v oblasti vývinu webových aplikácií. Táto metóda taktiež popisuje definíciu požiadaviek od najzákladnejších po konkrétnejšie (charakterizácia publika). Využitie písomného popísanie požiadaviek je veľmi flexibilný spôsob, ktorý je doplnený o podrobnejší diagram úloh, za účelom definovania navigačného modelu a diagramu tried.

Nanešťastie, písomný popis s využitím klasického jazyka znovu spadá aj pod nevýhody tejto metódy. Z MDD pohľadu je možné stretnúť sa s určitou mierou nepresnosti, ktorá vychádza z daného neformálneho popisu. [9]

Porovnanie Táto časť predstavuje porovnanie popísaných metód. Tieto boli vyššie popísané na základe funkčných, dátových a navygačných požiadaviek, ako aj z MDD perspektívy. Na konci každej metódy sa taktiež nachádza prehľad ich silných a slabých stránok.

Na začiatok je nutné povedať, že všetky z predošle spomenutých metód boli v minulosti využité na vývoj webových aplikácií a sú na ne aj zamerané. Všetky metódy väčšinou podporujú všetky analyzované kritériá, aj keď niektoré len čiastočne. Prehľad týchto kritérií sa nachádza v tabuľke na konci popisu každej metódy. Samozrejme každá zo spomenutých metód má svoje plusy a mínusy, a sú zamerané na odlišné aspekty vrámci vývinu webových aplikácií.

OOHDM je najstaršia metóda zo všetkých analyzovaných. Je založená na objektovo-orientovanom prístupe a predstavuje techniku separácie určitých aspektov webových aplikácií od ostatných. Pre podporu analyzovaných požiadaviek využíva prípady použitia (use cases) a UID diagramy. Väčšinu kritérií podporuje a zostáva iba pri dátových požiadavkách.

OOWS je o niečo mladšia metóda. Ide o rozšírenie objektovo-orientovaného prístupu pri webových aplikáciách a podporuje automatické generovanie webových rozhraní. Na podporu analyzovaných požiadaviek využíva hlavne diagram aktivít a tiež informačnú šablónu. Táto metóda plne podporuje všetky analyzované kritériá. Ide o veľmi užitočnú metódu, ktorou jedinou nevýhodou je zložitosť požiadavkových modelov.

Poslednou analyzovanou metódou je WSDM. Táto metóda vznikla medzi OOHDM a OOWS metódou a nie je založená na objektovo-orientovanom prístupe. WSDM sa zameriava na používateľa a považuje ho za centrum svojho prístupu. Na podporu analyzovaných požiadaviek využíva klasický jazyk. Taktiež podporuje väčšinu analyzovaných kritérií, no niektoré iba čiastočne. Jej výhodou je práve spomínaný ohľad na používateľa a na jeho požiadavky. Nevýhodou je ale využitie klasického neformálneho jazyka, ktorý môže často viesť k miere nepresnosti pri definovaní požiadaviek. [9]

Ďalej nasleduje predstavenie a popis MDA prístupu.

5 Popis MDA prístupu

MDA prístup je široko využívaný softvérovými inžiniermi pri vyvíjaní webových aplikácií.

Častokrát, pri MDD metódach, je interakcia medzi používateľom a systémom nie presne špecifikovaná. Často je rozhranie systému generované pre rozličné platformy (stolný počítač, web, mobilný telefón...) z rovnakého modelu. Tým sa vytvára priestor pre rozličné komplikácie spojené so systémovým rozhraním. V tom prípade je nutné klásť väčší dôraz na presnejšie špecifikovanie koncepcných modelov. Preto je v tomto prípade MDA prístup veľmi zaujímavým riešením. [1] MDA (Model-Driven Architecture) je prístup využívaný v modelovo založenom softvérovom vývojárstve a predstavuje viacero modelov. Tieto modely majú za úlohu vylepšiť a spresniť proces vývoja softvéru. Prvým z nich je CIM (Computation-Independent model), v preklade model nezávislý na výpočte. Tento model neberie ohľad na výpočtové aspekty spojené s modelovaním systému a je zameraný výlučne na požiadavky systému a jeho prostredia. Ďalším modelom je PIM (Platform-Independent Model), ktorý popisuje interakciu s danou platformou, bez toho, aby uvažoval technologické aspekty platformy.

Tie špecifikuje posledný model PSM (Platform Specific Model). Tento model vychádza z predošlého modelu a už presne popisuje technologické požiadavky rôznych platforém, na ktoré bude webová aplikácia následne implementovaná. Nakoniec je vytvorený model kódu z PSM modelu. Transformácia týchto modelov je automatická alebo polo-automatická. To závisí od MDA prostredia, ktoré ich podporuje. [5]

Ďalšia časť sa venuje porovnaniu MDWE metód vyžívajúcich spomenutý MDA prístup.

6 Porovnanie MDWE metód vyžívajúcich MDA prístup

Postupom rokov sa začali prejavovať slabšie stránky predošle spomenutých metód. Jedna z nich bola spomenutá v predchádzajúcej sekcii. Preto za účelom odstránenia týchto nedokonalostí sa začal využívať MDA prístup v spojení so spomenutými metódami. Tak vznikli hybridné metódy, ktoré využívajú techniky a postupy obsiahnuté v základných metódach, ale sú obohatené o postupy definované v MDA paradigme. Takýmito hybridnými metódami sú napríklad OOHDMDA a WSDMDA. Vďaka implementovaniu týchto postupov sa stávajú dané metódy efektívnejšie v oblasti vývoja webových aplikácií. OOWS metóda je v tejto časti vynechaná, pretože implementácia MDA je pri objektovo-orientovaných metódach čiastočne podobná (napr. obidve využívajú externé nástroje), a preto bude spomedzi objektovo-orientovaných metód popísaná len OOHDMD metóda.

Ďalej budú tieto metódy bližšie špecifikované a porovnané.

6.1 OOHDMDA

OOHDMDA je modelovacia metóda, ktorá kombinuje vlastnosti OOHDMD metódy a prvky MDA prístupu. Jej hlavným cieľom je transformácia koncepčného modelu základnej metódy na PIM a následná transformácia na PSM. Tieto transformácie sú pomerne zložité procesy. Preto budú popísané o niečo jednoduchším spôsobom. Prvou časťou je generovanie PIM modelu. PIM model je vytvorený z dizajnového modelu obsiahnutého v základnej metóde OOHDMD, ktorý je potom rozšírený o ďalšie aspekty (Behavioral Semantics Model). Ďalej nasleduje transformácia na PSM model. Tá je sprevádzaná rôznymi špecializovanými nástrojmi ako napríklad XMINavigationalTransformer, ktorý je spomenutý v článku od Hansa Alberta Schmida [6]. Konečným výsledkom tejto metódy je webová aplikácia založená na sevrlet programe. Ide jednoducho o program, ktorý odpovedá na sieťové požiadavky, najčastejšie HTTP požiadavky. Čiže OOHDMDA generuje servlet-založené webové aplikácie z tradičnej metódy. [6]

6.2 WSDMDA

WSDMDA je MDA-založená metóda na vývoj webových aplikácií. Ako z názvu vyplýva, tak je založená na existujúcej WSDM metóde spomenutej vyššie v článku. WSDMDA zvyšuje efektivitu základnej metódy tým, že poskytuje vyššiu rýchlosť generovania komplexného kódu ako tradičná metóda. Výsledkom

tejto hybridnej metódy oproti základnej je taktiež schopnosť narábať aj s dynamickou webovou aplikáciou, namiesto narábania jedine so statickou. Toto je dosiahnuté vďaka tomu, že vrchná časť webovej stránky sa mení v závislosti od používateľských záujmov, kým druhá časť ostáva taká, aká bola definovaná webovým dizajnérom. Profil týchto používateľských záujmov je využívaný počas chodu webovej aplikácie, za cieľom zobrazovania špeciálnych položiek, ktoré môžu byť využité ako promócia nového produktu alebo ako reklama špecifickej témy v danej oblasti.

Táto metóda prerába koncepčný model tradičnej metódy, za účelom využitia tohto modelu ako PIM model, a to vďaka pridaniu používateľského záujmového profilu. Ďalej kladie veľký dôraz na následnú transformáciu na PSM model, spresnením PIM modelu. [4]

Porovnanie Spomenuté metódy sa od seba líšia ako v základnom, tak aj v tomto rozšírenom tvare. Tu je ale možné presne vidieť ich rozdielne prístupy k MDA paradigme. Prvá fáza je rovnaká, transformácia koncepčného modelu na PIM model. Kým OOHMDA metóda ku koncepčnému modelu pridáva ďalší model, špecificky BMS (Behavioral Semantics Model), tak WSDMDA sa snaží upriamiť ešte väčšiu pozornosť na používateľa pridaním používateľského profilu. Neskôr nasleduje transformácia na PSM model. Tá sa taktiež u oboch kandidátov líši. OOHMDA využíva rôzne nástroje za účelom prerobenia PIM sa sevrlet založenú platformu. Tieto transformácie taktiež zahŕňujú navigačné a koncepčné transformácie [4]. Na druhej strane WSDMDA vytvára všeobecný PSM model, ktorý presne špecifikuje technologické požiadavky danej platformy. Obrázok č.6 slúži ako tabuľka na grafické zhrnutie tohto porovnania.

Kritérium	OOHMDA	WSDMDA
Základná metóda	OOHDM	WSDM
PIM	Prerába objektový model OOHDM na PIM po pridaní BMS (Behavioral Semantics Model)	Prerába koncepčný model WSDM na PIM po pridaní používateľského profilu
PSM	Zameraný na sevrlet-založenú platformu	Všeobecný PSM
Modelová transformácia	Navigačná a koncepčná transformácia	Prerobený koncepčný model (PIM) na všeobecný PSM model

Obr. 6: Porovnanie OOHMDA a WSDMDA metódy (prevzatý [4]).

7 Reakcia na témy

Udržateľnosť a etika. Udržateľnosť a etika sú dve zásadné slová v rámci softvérového inžinierstva. Udržateľnosť je neodmysliteľnou súčasťou života softvérového inžiniera. Ten mus vedieť vyvíjať softvér, ktorý je efektívny, spoľahlivý a univerzálny. Udržateľný rozvoj je taký, ktorý je funkčný v súčasnosti a dokáže

byť využitý aj v budúcnosti.

Na druhej strane etika predkladá to, ako softvérový inžinier koná alebo by mal konať. Softvérový inžinier by sa mal správať podľa verejného záujmu. Taktiež musí prihliadať na to, aké správanie je najvhodnejšie pri práci s klientom a zamestnávateľom. No najdôležitejším princípom etiky je celoživotné vzdelávanie sa a preukazovanie ochoty učiť sa nové veci.

Spoločenské súvislosti. Medzi spoločenské súvislosti spojené s informatikou môže byť rozhodne zahrnutý Agile inžinierstvo a koncept SCRUM-u. Agile engineering prístupuje k vývinu softvéru spôsobom, ktorý je efektívny a flexibilný. Srdcom tohto prístupu sú štyri medzi sebou prepojené piliere. Zamyslenie, vylepšenie, kolaborácia a dodanie. Ďalším základným prvkom agilného inžinierstva je SCRUM. Ide o skupinu inžinierov, ktorí spolu pracujú na danom projekte. Robotu delia na takzvané šprinty a každý deň sumarizujú čo sa už spravilo a čo je ešte potrebné spraviť. Taktiež riešia novovzniknuté problémy a to im povoľuje pracovať nezávisle.

Historické súvislosti. Informatika je síce mladá veda, no jej história je plná rôznych objavov, ktoré napomohli jej rozvoju. O tie sa pričínili veľké množstvo inžinierov. Pretože bez nich by informatika akú poznáme dnes, určite nevznikla. Prvým dôležitým menom je Fernando Corbato, ktorý prišiel s myšlienkou využitia hesla na prístup do počítačov. Ďalším míľnikom bol vynález myši, bez ktorej si nevieme predstaviť prácu s počítačmi. Bola to práve zásluha Bill-a English-a. Taktiež treba spomenúť prvú laserovú tlačiareň, za ktorú ďakujeme človeku, ktorý musel bojovať za svoje ciele. Jeho meno je Gary Starkweather. História informatiky je ale taká rozsiahla, že nie je možné spomenúť všetky významné mená. No je dôležité študovať históriu, pretože tá je základom pre budúci rozvoj.

8 Zhrnutie

Primárnou úlohou tejto práce je popísať a porovnať metódy využívané pri vývoji webových aplikácií. Analyzované metódy boli OOHDM, OOWS a WSDM. Spomenutý bol taktiež populárny MDA prístup spolu s jeho implementáciou na vylepšenie spomenutých metód. Dozvedeli sme sa ako jednotlivé metódy fungujú, aké majú požiadavky a čo je úlohou hybridných metód využívajúcich MDA prístup. Taktiež boli osobitne porovnané základné a hybridné metódy. Z týchto porovnaní vyplývajú viaceré fakty. Je zrejmé, že všetky metódy sú zamerané na vývoj webových aplikácií a líšia sa iba v prístupe realizácie tohto cieľa. Rovnako sme zistili, aký užitočný je MDA prístup a aké je jeho využitie. Ide mimovšetkého o transformáciu modelov za účelom presnejšieho špecifikovania technologických požiadaviek systému. Na to slúžia hybridné metódy ako napríklad OOHMDA a WSDMDA, ktoré kombinujú techniky tradičných metód a obohacujú ich o tie, ktoré sú spomenuté v MDA paradigme.

Tento článok preto predstavuje prehľad týchto metód spolu s MDA sprístupom a poukazuje na silné a slabé stránky každej metódy.

Budúce práce budú zamerané na popis a porovnanie ďalších metód, využívaných pri vývoji webových aplikácií, ako aj na nové techniky a technológie využívané v tomto obore softvérového vývinu.

Literatura

- [1] Oscar Pastor Francisco Valverde, Ignacio Panach. An abstract interaction model for a mda software production method. In *Tutorials, Posters, Panels and Industrial Contributions at ER 2007*. ACM, November 2007.
- [2] Vassiliki Gkantouna, Athanasios Tsakalidis, and Giannis Tzimas. Mining interaction patterns in the design of web applications for improving user experience. ACM, July 2016.
- [3] Thiago Gottardi and Rosana Teresinha Vaccare Braga. Evaluating the ability of developers to use metamodels in model-oriented development. IEEE, May 2019.
- [4] Mohammed Abdalla Osman Mukhtar, Mohd Fadzil B Hassan, Jafreezal Bin Jaafar, and Lukman A. Rahim. Enhanced approach for developing web applications using model driven architecture. In *2013 International Conference on Research and Innovation in Information Systems (ICRIIS)*. IEEE, November 2013.
- [5] Oscar Pastor, Sergio España, José Ignacio Panach, and Nathalie Aquino. Model-driven development. 31(5):394–407, August 2008.
- [6] Hans Albrecht Schmid and Oliver Donnerhak. OOHDMDA – an MDA approach for OOHDM. In *Lecture Notes in Computer Science*, pages 569–574. Springer Berlin Heidelberg, 2005.
- [7] Daniel Schwabe, Gustavo Rossi, and Mark Douglas de Azevedo Jacyntho. A software architecture for structuring complex web applications. 1(1):38–60, January 2002.
- [8] B. Selic. The pragmatics of model-driven development. 20(5):19–25, September 2003.
- [9] Pedro Valderas and Vicente Pelechano. A survey of requirements specification in model-driven development of web applications. 5(2):1–51, May 2011.