

NOVINKY VE WEBOVÝCH METODIKÁCH A METODIKA ONTOWEAVER

Martin Molhanec

České vysoké učení technické – FEL, K-13113; Technická 2, 166 27 PRAHA 6, Dejvice

mailto: molhanec@fel.cvut.cz

http://martin.feld.cvut.cz/~molhanec

ABSTRAKT:

Obsahem příspěvku je informace o současném stavu v oblasti webových metodik a stručný popis a zhodnocení metodiky OntoWeaver určené pro podporu tvorby webových sídel. Jedná se o velice zajímavý počín vytvoření webové metodiky vystavěné na paradigmatu využití ontologií. Tvorba webových sídel je současným trendem v oblasti tvorby informačních systémů, bohužel většina tvůrců a firem prozatím ignoruje její teoretické základy.

1 ÚVOD

Tento příspěvek navazuje na moje předchozí příspěvky na této konferenci v předešlých letech ([1], [2], [4], [5] a [7]) a na konferenci Objekty ([3] a [6]). Velice stručně zopakují několik základních informací o tom co to jsou webové metodiky a k jakému účelu slouží. Webové metodiky jsou speciálním druhem metodik, které jsou primárně určené pro analýzu a návrh webových sídel a webových aplikací. Webové metodiky vznikly na základě o něco starších metodik pro návrh hypermediálních a hypertextových aplikací. Jejich společným základem jsou pochopitelně metodiky pro analýzu a návrh informačních systémů obecně. Znalost těchto webově orientovaných metodik je v ČR minimální, ale podobně je tomu i v dalších zemích. Tato skutečnost je v rozporu se skutečností, že webové aplikace jsou v současné době na vrcholu zájmu všech softwarových vývojářů, uživatelů a obchodníků.

2 NOVINKY V OBLASTI WEBOVÝCH METODIK

První webovou (respektive hypermediální a hypertextovou) metodikou byla metodika HDM (Hypertext Design Model) [8], která vznikla na *Politecnico di Milano* autorů *Garzotto, F., Paolini, P. a Schwabe, D.* Tato metodika měla zásadní vliv na mnoho dalších metodik, které na ní navazovaly ať již přímo nebo nepřímo. Nepřímo na tuto metodiku navázala metodika RMM (Relationship Management Methodology) [9] a OOHDM (Object Oriented HDM) [10]. Ostatně jeden z autorů OOHDM (Schwabe, D.) byl i spoluautor metodiky HDM. Samotná metodika HDM se postupně transformovala do podoby HDM-lite, která již byla speciálně určená pro tvorbu webových sídel a W2000. A touto metodikou se zdá, že vývoj přímých nástupců metodiky HDM vyhasíná. Ale není to pravda tak zcela, skupina spoluautorů, kteří pracovali na této a předešlé metodice, vytváří metodiku WebML [11]. Metodika WebML přináší mnohé nové myšlenky, precizní notaci navigace po webovém sídle a první opravdu komerční nástroj typu CASE pro její podporu – Webratio [12].

Vznikají však i další metodiky, které nejsou až tak úzce příbuzné či navazující na původní metodiku HDM. Mezi ně patří například Araneus [13], OO-H [14], WSDM [15] a metodika

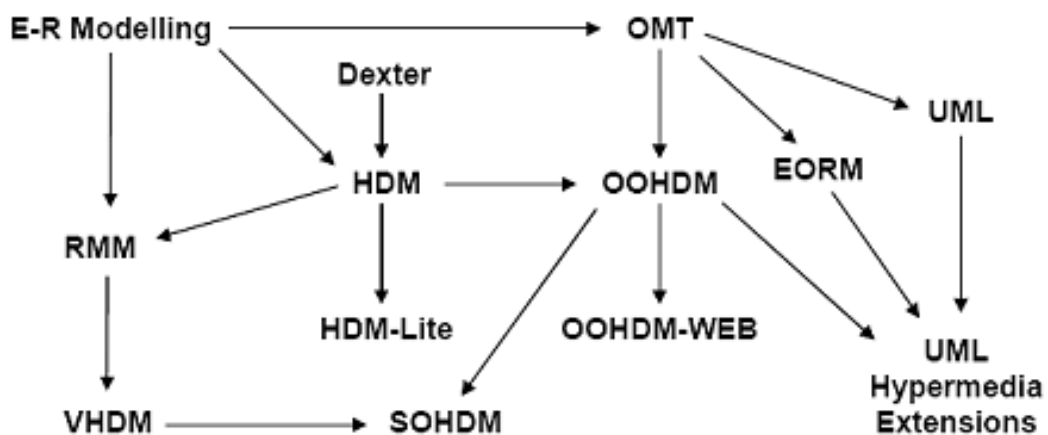
UWE [16]. Každá z výše uvedených metodik přináší nové myšlenky a všechny zápasí s tím, jak se vyrovnat se správným návrhem navigace a prezentace webového sídla.

Mezi živé a stále se rozvíjející metodiky můžeme zařadit následující:

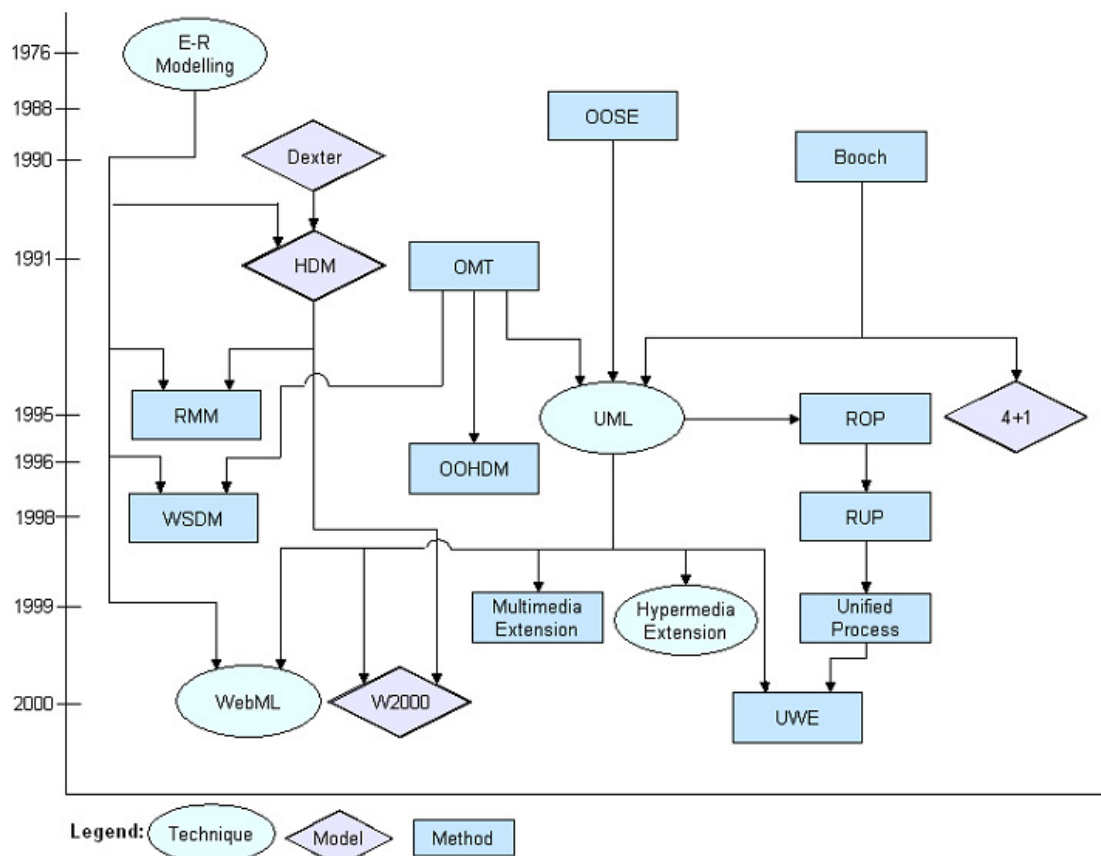
- WSDM – metodika využívající pro modelování ORM a kladoucí důraz na modelování uživatelů.
- HERA [17] – metodika dále rozvíjející myšlenky metodiky RMM a původního Dexterova hypertextového modelu.
- UWE – metodika kladoucí důraz na využití standardního modelovacího jazyka UML.
- WebML – metodika, která se snaží poskytnout jejím uživatelům profesionální (ovšem za peníze) nástroj typu CASE.

Zajímavý obrázek (obr.: 1) ukazující vztahy mezi různými metodikami je převzat z [21]. Jiný podrobnější a novější (obr.: 2) ukazující zase jiné souvislosti je převzat ze [22].

Jednou z nových metodik je i metodika OntoWeaver, podrobněji popisovaná v další části tohoto článku.



obr.: 1: Vztahy mezi různými webovými metodikami.



obr.: 2: Vztahy mezi různými metodikami.

3 OntoWeaver

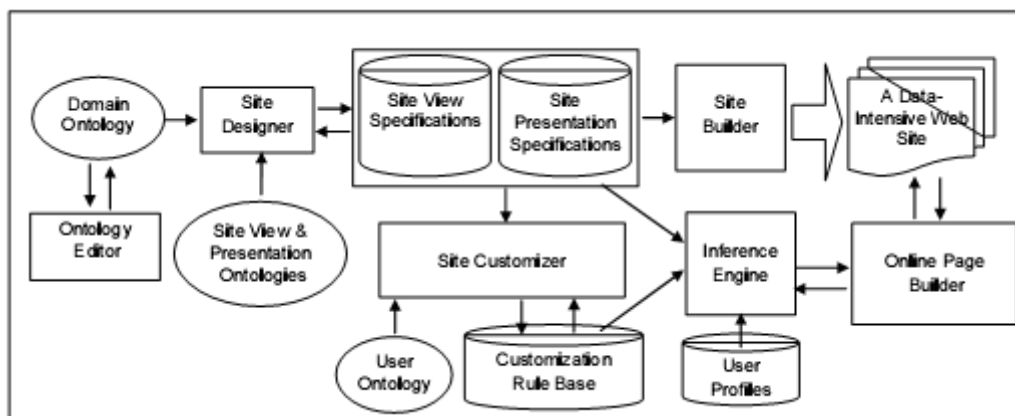
Metodika *OntoWeaver* ([18], [19], [20]) autorů *Yuanguai Lei, Enrico Motta a John Domingue* z *The Open University UK* je jedním z dalších pokusů o vytvoření moderní a ucelené metodiky pro tvorbu webových sídel. Na rozdíl od ostatních metodik, již výše zmíněných, se autoři opírají o dnes moderní a často používaný princip ontologií. Podle autorů metodiky *OntoWeaver* je *ontologie* explicitní formální specifikace pojmů z oblasti našeho zájmu a vztahů mezi těmito pojmy. Autoři metodiky definují následující ontologie:

- *A site view ontology*, pomocí které modelují strukturu webového sídla a rozhraní uživatele.
- *A presentation ontology* pro popis prezentačních stylů a rozvržení webových stránek.
- *A generic customization framework* jako nástroj podpory návrhu přizpůsobeného potřebám uživatele.

Celá specifikace rozsáhlého webového sídla pak zahrnuje následující komponenty:

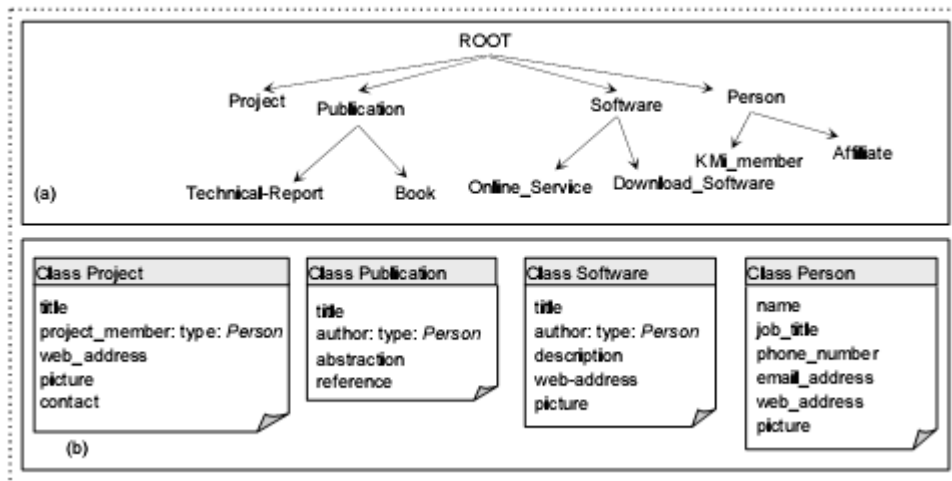
- *A domain ontology*, která je abstrakcí datových zdrojů celého systému. Je tvořena *pojmy a vztahy* mezi těmito pojmy. Datový obsah pak představují instance pojmů. Vztahy mezi pojmy (daty) pak mohou být užity pro odvození navigace uvnitř aplikace.
- *A domain knowledge base*, popisuje datový obsah systému v pojmech *doménové ontologie*.
- *Site view specifications*, které popisují navigační strukturu webového sídla a uživatelského rozhraní webových stránek.
- *Presentation specifications*, které popisují prezentační styly a rozvržení webových stránek.
- *A user ontology*, modelující uživatele systému.
- *A user model*, popisující uživatele systému v pojmech *user ontology*.
- *Customization rules*, definující pravidla pro personalizaci základního vzhledu stránek s ohledem ke konkrétnímu uživateli a jeho prostředí.

Vztahy mezi jednotlivými částmi metodiky OntoWeaver jsou zobrazeny na obr.: 3.



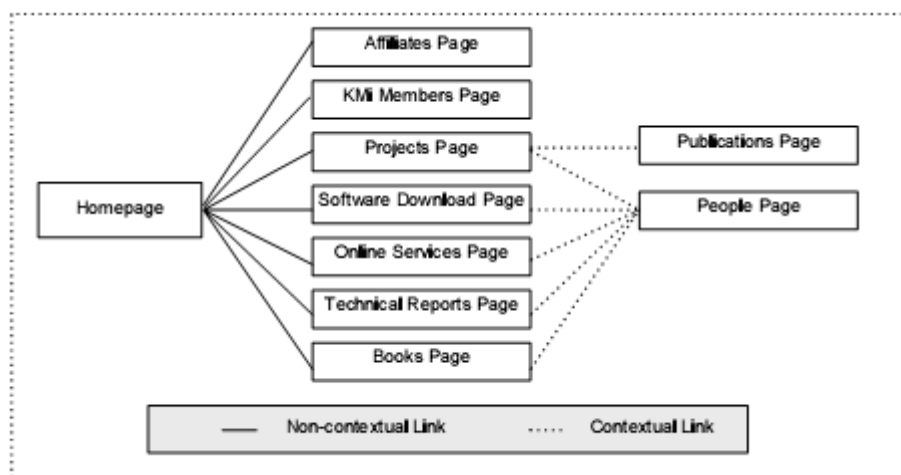
obr.: 3: OntoWeaver Framework.

Ukázka *domain ontology* je na obr.: 4. Autoři využívají dvou vztahů mezi doménovými entitami: *IS-A* vztah mezi jednotlivými entitami a *HAS-A* vztah mezi entitou a jejími atributy. V části (a) je zobrazena hierarchická struktura dat a v části (b) je popis jednotlivých entit (tříd).



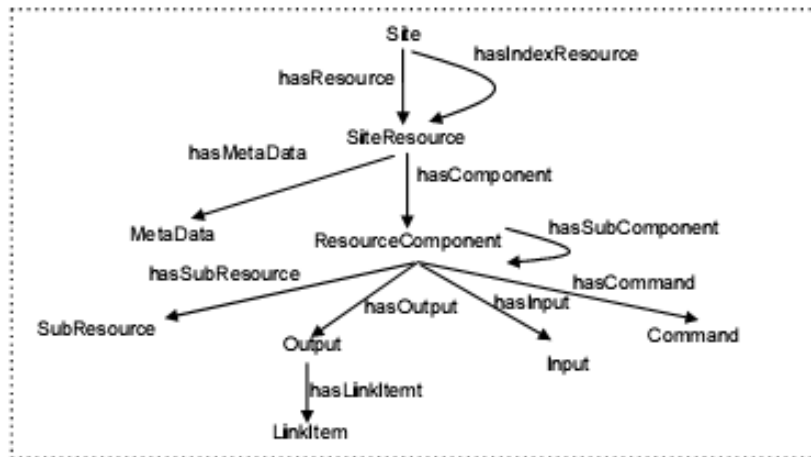
obr.: 4: Doménová ontologie.

Celková struktura webového sídla odpovídajícího výše uvedenému příkladu je pak na obr.: 5. Autoři rozlišují *kontextové*, přenášející kontextovou informaci a *nekontextové* přechody (*link*) mezi stránkami webového sídla.



obr.: 5: Struktura webového sídla.

Další částí metodiky je *Site View Ontology*. Tato ontologie modeluje webové sídlo jako kolekci logických zdrojů. *Site View Ontology* obsahuje množinu navigačních konstruktů, množinu konstruktů pro vyjádření uživatelského rozhraní, které dále zjemňuje na konstrukty atomické (dále nedělitelné) a konstrukty kompozic, které se skládají z dalších jednodušších konstruktů. Na obr.: 6 je přehledně znázorněna hierarchie mezi jednotlivými konstrukty *site view ontology* a v Tab. 1 je jejich podrobný přehled převzatý z [18].



obr.: 6: Přehled *site view ontology*.

Tab. 1

Construct	Sub Constructs	Slots	Description
<i>Composite User Interface Constructs</i>			
Site		<ul style="list-style-type: none"> • hasIndexResource • hasResources • hasDomainURI 	Modelling web sites as a collection of web resources.
SiteResource		<ul style="list-style-type: none"> • hasComponent • hasMetadata 	Modelling web pages.
ResourceComponent	<ul style="list-style-type: none"> • KAComponent • DataComponent • SearchComponent • OutputComponent • InputComponent 	<ul style="list-style-type: none"> • hasSubComponent • hasOutput • hasInput • hasCommand • hasSubResource 	Modelling user interface elements that compose web pages.
KAComponent		<ul style="list-style-type: none"> • hasClassEntity • hasInputComponent • hasCommand 	Modelling components that allow users to input facts to the underlying databases.
DataComponent		<ul style="list-style-type: none"> • hasClassEntity • hasOutputComponent 	Modelling components that publish data content coming from the underlying databases.
SearchComponent		<ul style="list-style-type: none"> • hasClassEntity • hasInputComponent • hasCommand 	Describing components that allow users to make queries over the back-end databases.
OutputComponent		<ul style="list-style-type: none"> • hasOutput • hasDynamicOutput 	Publishing dynamic domain content retrieved from the specified slot of the specified domain class entity.
InputComponent		<ul style="list-style-type: none"> • hasOutput • hasInput 	Presenting interface elements to gather input from end users for a particular slot of the specified class entity.
<i>Atomic User Interface Constructs</i>			
Input		<ul style="list-style-type: none"> • hasClassEntity • hasSlotEntity 	Expressing the basic interface elements that allow end users entering meaningful information to interact with web applications.
Output	<ul style="list-style-type: none"> • DynamicOutput 	<ul style="list-style-type: none"> • hasOutputType • hasOutputValue • hasLinkItem 	Abstracting the basic elements that present a piece of information, either being static/dynamic, text/image, plain or having links associated with it.
DynamicOutput		<ul style="list-style-type: none"> • hasClassEntity • hasSlotEntity 	Displaying dynamic content coming from the specified slot of the specified class entity.
Command		<ul style="list-style-type: none"> • hasService • hasResultPage 	Abstracting the basic interface elements that enable end users to invoke the specified services and bring dynamic content to users.
SubResource		<ul style="list-style-type: none"> • associatedResourceURI • isExternalResource 	Expressing the basic elements, which import specified web pages into web pages.

Navigational Constructs			
LinkItem	• DynamicLinkItem	• associatedResourceURI • isExternalResource • hasParameter	Modelling link relationships.
DynamicLinkItem		• hasClassEntity • hasSlotEntity	Modelling the link items that come from the underlying knowledge bases.
Parameter		• hasParameterClause	Describing contextual information which flows along with links.
ParameterClause		• hasClassEntity • hasSlotEntity • hasRelationOperator • hasValue • hasLogicalOperator	Abstracting parameter clauses, which compose parameters.
MetaData		• hasPageHeadline • hasIntroduction • hasDescription • hasAuthors	Describing meta-information for web pages.

V dalších krocích metodiky probíhá modelování jednotlivých částí webového sídla s využitím RDF [23] specifikace. Existují tři typy atomických (dále nedělitelných) elementů uživatelského rozhraní:

- *output elements*, které reprezentují zobrazení statické nebo dynamické informace;
- *input elements*, které dovolují vstup informace od uživatele;
- *command elements*, které představují aktivaci některých služeb uživatelem.

Vztah mezi zobrazením výše uvedených elementů na stránkách webového sídla a jejich vyjádřením pomocí DRF je na obr.: 7.

The diagram illustrates the mapping between user interface elements and their RDF descriptions. It is divided into two main sections: a top section for a project page and a bottom section for a seminar form.

Top Section: Project Page (AKT)

- Static output element:** Maps to the text description of the AKT project. The RDF code is:


```
<rdf:Description rdf:about=".../static-outputx" >
  <rdf:type rdf:resource="&svo;Output" />
  <svo:hasOutputValueType>text</svo:hasOutputValueType>
  <svo:hasOutputValue>Participants</svo:hasOutputValue>
</rdf:Description>
```
- Dynamic Output element:** Maps to the URL and project name. The RDF code is:


```
<rdf:Description rdf:about=".../url-output" >
  <rdf:type rdf:resource="&svo;DynamicOutput" />
  <svo:hasClassEntity rdf:resource="&do;Project" />
  <svo:hasSlotEntity rdf:resource="&do;has-web-address" />
  <svo:hasLinkItem rdf:resource="#project-url-link" />
</rdf:Description>
```

Bottom Section: Seminar Form (New Seminar)

- Input element:** Maps to the text input field for the seminar title. The RDF code is:


```
<rdf:Description rdf:about=".../seminar-title-input" >
  <rdf:type rdf:resource="&svo;Input" />
  <svo:hasClassEntity rdf:resource="&do;Seminar" />
  <svo:hasSlotEntity rdf:resource="&do;has-title" />
</rdf:Description>
```
- Command element:** Maps to the Submit button. The RDF code is:


```
<rdf:Description rdf:about=".../add-seminar-command" >
  <rdf:type rdf:resource="&svo;Command" />
  <svo:hasClassEntity rdf:resource="&do;Seminar" />
  <svo:hasTask rdf:resource="#new-data-entry" />
</rdf:Description>
```

obr.: 7: Příklad atomických elementů uživatelského rozhraní.

4 ZÁVĚR

Přestože metodika OntoWeaver využívá moderní přístup s využitím ontologií, zdá se mi, že se ve skutečnosti od ostatních metodik, které se v současné době také rozvíjejí, až tak příliš neodlišuje. I ostatní metodiky modelují datové pozadí webového sídla, pro které ovšem používají tradiční názvy, např. konceptuální, objektový nebo ER model. I ostatní metodiky modelují různé typy navigací mezi jednotlivými stránkami aplikace a modelují rozličné typy grafických elementů tvořících webovou stránku. Nicméně důraz na ontologie je znamením, že je zde snaha, postavit webové metodiky na více formální základy, nežli tomu bylo doposud. Autoři této metodiky slibují její další rozvoj včetně vytvoření jí podporujících nástrojů.

Obecně lze konstatovat, že v oblasti webových metodik probíhá neustálý vývoj, který lze charakterizovat následujícími rysy:

- Snaha o přesný a formální popis jednotlivých metod, pojmů a diagramů.
- Snaha o metodický přesný popis přechodu od konceptuálního modelu k modelu navigace a prezentace.
- Snaha o využití standardních nástrojů jaké jsou UML, XML, a další.
- Snaha o pokrytí celého životního cyklu produktu.
- Snaha o poskytnutí profesionálního nástroje typu CASE uživateli.

Jsem přesvědčen, že se dočkáme dalších nových a lepších webových metodik. Otázkou je, kdy budou tyto metodiky vyučovány na školách a obecně akceptovány firmami, které webová sídla vytvářejí.

5 LITERATURA

- [1] Molhanec Martin: „Metodika UWE (UML based Web Engineering)“, In: Tvorba softwaru 2005. Ostrava: VŠB, 2005, s. 143–152. ISBN 80–86840-14X.
- [2] Molhanec Martin: „Metodiky orientované na tvorbu webových sídel“, Tvorba software 2004, TANGER, Ostrava 2004
- [3] Martin Molhanec: „WebML – Objektově orientovaná metodika pro tvorbu webových sídel“, Objekty 2003, VŠB-TU, Ostrava 2003. On line: <http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2003/WebML-CO.pdf>
- [4] Martin Molhanec: „Metodologie orientované na tvorbu webových sídel“, Tvorba software 2003, TANGER, Ostrava 2003. On line: <http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2003/WebMet-co.pdf>
- [5] Molhanec Martin: „Metodologie OOHDm, jazyk Lua a tvorba webových aplikací“, Tvorba software 2002, TANGER, Ostrava 2002. On line: <http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2002/Lua.pdf>
- [6] Molhanec Martin: „The Object-Oriented Hypermedia Design Model (OOHDM)“, Objekty 2001, Česká zemědělská universita, Praha 2001. On line: <http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2001/OOHDM.pdf>
- [7] Molhanec Martin: „Tvorba webových sídel jako inženýrský úkol“, Tvorba software 2001, TANGER, Ostrava 2001. On line: <http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2001/metodika.pdf>

- [8] Franca Garzotto, Paolo Paulini: „HDM - A Model-Based Approach to Hypertext Application Design“, ACM Transactions on Information Systems (TOIS), Volume 11, Issue 1 (January 1993), Pages: 1 - 26, Year of Publication: 1993, ISSN:1046-8188.
- [9] Isakowitz, T., Kamis, A., Koufaris, M.: „The Extended RMM Methodology for Web Publishing“, Working Paper IS-98-18, Center for Research on Information Systems, 1998 (Currently under review at ACM Transactions on Information Systems.).
- [10] Daniel Schwabe, Patricia Vilmin, Rondon Guimarães, Gustavo Rossi: „*A conference Review System in OOHDM*“, IWWOST 2001, Valencia, Spain, 2001.
- [11] Stefano Ceri, Piero Fraternali, Aldo Bongio: "Web Modeling Language (WebML): a modeling language for designing Web sites", WWW9, Amsterdam, May 2000.
- [12] On line: <http://www.webratio.com/>.
- [13] Paolo Merialdo, Paolo Atzeni, Giansalvatore Mecca: „Design and development of data-intensive web sites: The Araneus approach“, February 2003, ACM Transactions on Internet Technology (TOIT), Volume 3 Issue 1, Publisher: ACM Press.
- [14] C. Cachero, J. Gómez, A. Párraga, O.Pastor: „The OO-H Method“, IWWOST 2001, Valencia, Spain, 2001.
- [15] De Troyer, O.M.F., Leune, C.J.: „WSDM: a user centered design method for Web sites“, 7th International World Wide Web Conference, Brisbane 1998.
- [16] Koch N. (2000). Software Engineering for Adaptive Hypermedia Applications. PhD. Thesis.
- [17] R. Vdovjak, F. Frasincar, G.J. Houben, P. Barna, "Engineering Semantic Web Information Systems in Hera", in: Journal of Web Engineering, Vol. 2, No. 1&2, p. 3-26, 2003, Rinton Press.
- [18] Y. Lei, E. Motta, and J. Domingue (2004), Modelling Data-Intensive Web Sites with OntoWeaver, accepted in International Workshop on Web Information Systems Modelling (WISM 2004), Riga, Latvia, 2004
- [19] Y. Lei, E. Motta and J. Domingue (2002).An Ontology-Driven Approach to Web Site Generation and Maintenance. In proceedings of 13th International Conference on Knowledge Engineering and Management, Sigüenza, Spain 1-4 October 2002, pp. 219-234
- [20] Y. Lei, E. Motta, and J. Domingue (2005), OntoWeaver: an Ontology-based Approach to the Design of Data-intensive Web Sites. Journal of Web Engineering, 4(3): 244-262
- [21] Lang, M. (2002). Hypermedia systems development: do we really need new methods? In E. Cohen & E. Boyd (Eds.), Proceedings of the Informing Science + IT Education Conference, Cork, Ireland, 883-891.
- [22] Inge van de Weerd, "WEM: A design method for CMS-based web implementations", technical report UU-CS-2005-043, Institute of Information and Computing Sciences, Utrecht University, 2005
- [23] On line: <http://www.w3.org/TR/rdf-syntax-grammar/>