

7 Conclusões

Neste capítulo fazemos nossos comentários finais a respeito de trabalhos relacionados, contribuições desta tese e trabalhos futuros, alguns dos quais já fazem parte de nossa agenda.

7.1. Trabalhos Relacionados

O projeto HERA [Frasincar et al., 2002] propõe o uso de RDF e RDF(S) (com pequenas extensões) como linguagem de definição de ontologias. Com esta abordagem, que não faz uso de DAML+OIL, não é possível expressar a mesma riqueza semântica que apresentamos em nosso Modelo Conceitual SHDM. Outros projetos como OntoWebber [Jin et al., 2001] acrescentam ontologias em camadas adicionais ao modelo conceitual para representar a estrutura de *Web-sites*. Em comparação, nossa abordagem manteve o uso de um modelo conceitual do tipo UML, estendendo-o com poucas primitivas como subrelacionamentos (originais de RDF(S)) e classes inferidas definidas a partir de restrições (originais de DAML+OIL).

Mantivemos a abordagem OOHDM ao definirmos o Modelo Navegacional como uma visão do Modelo Conceitual, e adicionamos o uso da linguagem de consulta RQL como parte da especificação. Utilizamos RQL para definir contextos, alcançando um maior grau de expressividade, uma vez que esta linguagem é capaz de consultar de forma transparente tanto as definições do esquema quanto das instâncias.

Os portais OntoWeb³³, Esperonto³⁴, Mindswap³⁵ utilizam uma abordagem diferente do SHDM, uma vez que não fazem uso de modelos de navegação.

³³ <http://ontoweb.aifb.uni-karlsruhe.de/>

³⁴ <http://www.esperonto.net/>

³⁵ <http://www.mindswap.org/>

7.2. Contribuições

A principal contribuição deste trabalho é o desenvolvimento da primeira versão do método SHDM, um método sistemático para desenvolvimento de aplicações para a Web Semântica.

Por exemplo, é possível definir um contexto de `Artefatos por Estilo`, sem saber de antemão qual os possíveis valores (ou subclasses de `Estilo`). Mesmo assim, se no domínio da aplicação surgir uma nova subclasse de `Estilo` (e suas instâncias correspondentes), a mesma especificação de aplicação Web ainda é aplicável. Neste sentido, as especificações SHDM podem ser vistas como especificações de *frameworks* (analogamente a [Schwabe et al., 2001b, 2001c]).

Outro benefício que SHDM proporciona é a habilidade de especificar de forma concisa as estruturas de acesso facetadas. Mostramos como a especificação de facetas é equivalente a grandes enumerações de possíveis caminhos de navegação. Com a crescente disponibilização de taxonomias nos domínios de aplicação, o uso das novas estruturas de acesso que propomos permitirão sempre alcançar conjuntos não vazios, evitando a síndrome da mensagem “sua consulta não retornou nenhum resultado”.

Mais detalhadamente, este trabalho ofereceu as seguintes contribuições:

- um método com abordagem abrangente considerando as diversas etapas de projeto de aplicações Web, integrado com os formalismos da Web Semântica. O método SHDM (*Semantic Hypermedia Design Method*) apresentado é na verdade, uma evolução da versão anterior chamada OOHDM (*Object Oriented Hypermedia Design Method*) – cap3;
- um conjunto de primitivas de modelagem a ser utilizado em cada uma destas etapas de projeto. O SHDM apresenta novas abstrações, que são mais ricas e permitem especificação de um conjunto maior de aplicações Web de forma concisa e não ambígua – caps 4 e 5;
- a especificação dos metadados de aplicações Web através de extensões de linguagens propostas pelo consórcio W3C, como RDF, RDF(S), DAML+OIL e OWL – caps 4 e 5;
- um modelo navegacional conciso e com poder expressivo maior, devido ao uso de uma linguagem de consulta apropriada, que permite que a definição de primitivas navegacionais seja feita com base tanto

nos dados quanto nos metadados. Pelo fato de poder consultar metadados, é possível definir-se famílias de aplicações com características comuns – cap 5;

- uma nova primitiva para a especificação de estruturas de acesso, através da incorporação do conceito de facetas. Esta primitiva permite que se construa índices que dão suporte à seleção incremental de objetos de navegação utilizando múltiplos critérios simultaneamente – cap 5;
- uma arquitetura de *software* que utiliza a própria especificação da aplicação, descrita nos formalismos propostos, para implementar esta aplicação, gerando páginas dinamicamente. Esta arquitetura garante a aderência da implementação à especificação, o que não ocorria anteriormente – cap6.

7.3. Trabalhos Futuros

Diversos trabalhos futuros foram mencionados ao longo do texto e podem ser resumidos aqui:

- Extensões do método SHDM para projetar aplicações Web adaptativas e personalizadas. Neste caso, as regras de adaptação, que podem alterar aspectos de navegação, interface, ou conteúdo, podem ser baseadas não apenas nos modelos de usuário, como também nos próprios modelos conceitual e navegacional, assim como nos dados específicos de determinadas instâncias sendo navegadas;
- Extensão da ontologia SHDM para especificação de Modelos de Interface Abstratas, integrando-os com Modelos de Interação, dando acesso às funcionalidades não-navegacionais da aplicação. Esta extensão deverá ser mapeada para modelos concretos de interface, integrados no ambiente de implementação descrito nesta tese ;
- Projeto Navegacional utilizando Ontologias Conceituais integradas através de mediadores;
- Detalhamento do processo de importação de taxonomias externas no processo de definição de estruturas de acesso baseadas em facetas. Dada a existência de taxonomias definidas nos mais diversos âmbitos, será interessante facilitar ao projetista importar estas taxonomias, atribuindo-as a uma faceta numa estrutura de acesso;

- Revisão das etapas de Levantamento de Requisitos e Projeto de Interface Abstrata do OODHM para adaptá-las a Web Semântica, lidando com as novas abstrações no método SHDM. Para Levantamento de Requisitos será necessário adaptar os UIDs para fazer Engenharia de Ontologias;
- Definição de arquitetura para a Web Semântica contendo Serviços Web (*Web Services*) para:
 - automatizar a oferta de projetos conceituais e ,
 - permitir a realização semi-automática de projetos navegacionais a partir de ofertas de:
 - facetas para o projetista escolher;
 - outras opções de estruturas de acesso e contextos.
- Padronização da arquitetura de implementação em ambiente Java através de uso de *taglibs*;
- Maior quantidade de estudos de caso para obtermos heurísticas diversas como: quais as características das melhores facetas?, qual a quantidade máxima/ideal de facetas (e sub-facetas)?;
- Estudos de caso envolvendo modelagem de métodos.