

TESTE DE ARQUÉTIPOS

Archetypes Testing

Alex Mateus Porn¹; Leticia Mara Peres²

Resumo O aumento da interoperabilidade entre sistemas de informação em saúde é possível utilizando o padrão de modelagem *openEHR*. Para verificar a corretude de arquétipos nesse padrão, este trabalho propõe-se convertê-los em ontologias OWL e realizar o teste de mutação. Assim, pretende-se revelar defeitos na modelagem do domínio antes de sua integração com outros sistemas.

Palavras-chave: Cuidado de Saúde Pervasivo, Computação Ubíqua, *openEHR*, Arquétipos, Linguagem Específica de Domínio

Abstract Increasing interoperability among health information systems is possible using *openEHR* standard modeling. In this work it is proposed to convert this pattern archetypes into OWL ontologies and perform mutation testing, to verify their correctness. Thus, we intend to reveal defects in the modeling domain before its integration with other systems.

Keywords: Archetypes, ADL, ontologies, OWL, mutation testing.

1. Profissional do Departamento de Informática do Centro Universitário de União da Vitória - UNIUV - Mestrando em Engenharia de Software pela Universidade Federal do Paraná - UFPR; 2. Professora Adjunta do Departamento de Informática da Universidade Federal do Paraná - UFPR - Doutora em Informática pela Universidade Federal do Paraná - UFPR

Introdução

Visando a eliminar problemas de interoperabilidade entre sistemas de informação em saúde, a modelagem multinível *openEHR*¹ propõe o desenvolvimento de sistemas em dois níveis: Modelo de Referência e Modelo de Arquétipos^[1].

Arquétipos no padrão *openEHR* são definidos em ADL, linguagem que permite a estruturação da informação, mas dificulta a análise semântica do conteúdo representado. A corretude do arquétipo também não é uma atividade facilmente automatizável devido àquela característica da linguagem. Processos de troca e transformação de dados entre sistemas tornam-se mais difíceis do que utilizar modelos orientados à semântica, como ontologias.

A representação de arquétipos ADL em OWL⁷ torna-se uma alternativa ao problema, pois ontologias são a base do desenvolvimento da Web Semântica³.

Como não existe apenas uma forma correta de modelar um domínio específico do conhecimento, defeitos podem ser cometidos ao modelar arquétipos e ontologias. Esses problemas estão relacionados a características próprias da linguagem e geralmente confundem modeladores².

Assim, motivam-se métodos de teste que validem os modelos clínicos representados em ADL.

Este trabalho propõe traduzir arquétipos ADL para ontologias OWL, para realizar testes de validação semântica de arquétipos *openEHR*.

Materiais e Métodos

Foram analisados 5 arquétipos do CKM5, o repositório de arquétipos *openEHR*, sendo um para cada tipo de

Entry. O objetivo desta análise foi verificar a corretude dos arquétipos.

Esses arquétipos foram convertidos para OWL, através de um algoritmo de mapeamento de ADL para OWL³, sendo o processo realizado com a ferramenta *Protégé*⁴, de criação e edição de ontologias.

Para a realização dos testes, aplicou-se o critério Análise de Mutantes⁸, e foram criados 12 operadores de mutação para OWL. Estes operadores geram mutantes, que são as novas estruturas com os defeitos simulados.

Usou-se a ferramenta *Protégé* para gerar os mutantes e interpretar consultas escritas em *DL Query*⁶, que são os dados de entrada do teste.

Resultados

Operadores de Mutação	Descrição
CUC	Uma classe sobe um nível na hierarquia e suas subclasses acompanham a mudança, ficando associadas a ela.
CUP	Uma classe sobe um nível na hierarquia e suas subclasses não acompanham esta mudança, permanecendo associadas à superclasse original.
CDD	Cria uma definição de disjunção entre classes, para todas as subclasses de uma determinada classe.
CDUO	Remove uma das classes da definição de disjunção de um conjunto de três ou mais classes.
CDUA	Elimina completamente uma definição de disjunção entre classes.
CEU	Remove uma definição de equivalência de uma classe.
CEUA	Altera uma definição de equivalência definida por uma expressão lógica (A and B), para (A) ou (B).

[1] Modela conceitos do domínio clínico a partir do modelo de referência.

[2] O ideal é que o escore de mutação seja o mais alto possível. Assim, foi calculado diretamente como a razão entre o número de mutantes mortos e gerados, sendo desconsiderados mutantes equivalentes, objetivando manter os resultados sem a intervenção do usuário.

CEUO	Altera uma definição de equivalência definida por uma expressão lógica (A or B), para (A) ou (B).
PDD	Troca uma definição de domínio de uma propriedade por uma subclasse da classe originalmente definida como sendo este domínio.
PRD	Troca uma definição de intervalo de uma propriedade por uma subclasse da classe originalmente definida como sendo limitadora deste intervalo.
ACOTA	Altera a definição de equivalência definida por uma expressão lógica (A or B), para (A and B).
ACSTA	Altera uma definição de equivalência definida por uma expressão lógica utilizando a restrição de valor existencial <i>SomeValues</i> , substituindo pela restrição universal <i>AllValues</i> .

Quadro 1: Operadores de Mutação

Após a geração dos mutantes e da execução dos dados de teste para revelar o defeito inserido pelo operador de mutação (dizendo-se que o mutante está “morto”), foram obtidos os resultados da Tabela 1, para um arquétipo do tipo *Evaluation*.

Operadores de Mutação	Número de Mutantes			Escore de Mutação ¹
	Gerado	Morto	Vivo	
CUC	12	05	07	42%
CUP	02	01	01	50%
CDD	15	12	03	80%
CDUO	09	00	09	0%
CDUA	02	00	02	0%
CEU	13	11	02	85%
CEUO	30	30	00	100%
PDD	07	07	00	100%
PRD	15	15	00	100%
ACOTA	16	16	00	100%
ACSTA	19	19	00	100%
TOTAL	140	116	24	100%

Tabela 1: Mutantes gerados para cada operador de mutação.

Ao contrário dos dois operadores que não obtiveram escore de mutação (0%), por necessitarem de indivíduos instanciados para produzirem efeitos, os demais apresentaram resultados altamente satisfatórios.

O operador CUC, que altera a estrutura hierárquica da ontologia, mostra possíveis irregularidades de desenvolvimento (Figura 1).

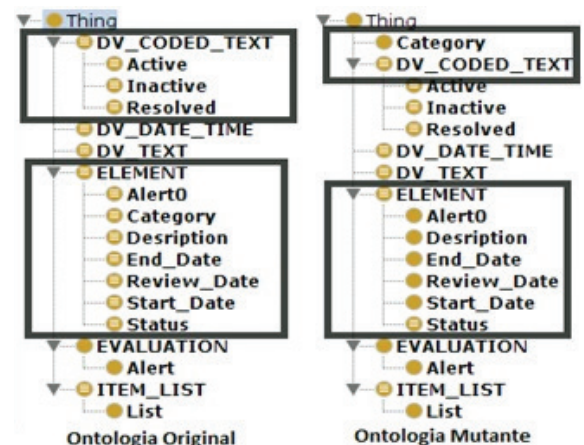


Figura 1: Modelo de arquétipo OWL com aplicação do operador CUC.

Devido à representação semântica do modelo através de axiomas, conforme o arquétipo ADL, ambas as versões desta figura são instanciadas similarmente em 07 dos 12 mutantes gerados, caracterizando-os como equivalentes (Figura 2). Três mutantes mortos com este operador possuem definições de equivalência, podendo caracterizar definições incorretas de axiomas, invalidando a hipótese do operador ser aplicado somente em classes que não possuam definição.

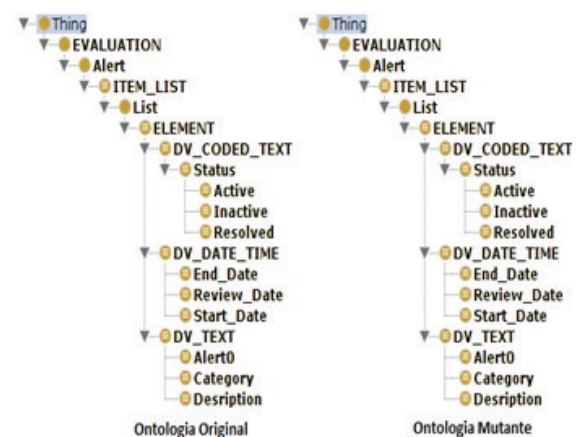


Figura 2: Representação de mutante vivo com o operador CUC.

Para o operador ACOTA, dos 16 mutantes gerados, obteve-se 100% de escore de mutação, i.e., de adequação nos casos de teste.

O mesmo ocorre para o operador ACSTA. A adequação total desses casos de teste significa que o operador ACOTA revela falhas com os operadores AND e OR, e que o operador ACSTA revela a aplicação incorreta em axiomas dos operadores de restrição universal e existencial.

Nota-se também a grande quantidade de mutantes e o alto escore de mutação dos operadores, permitindo validar e adequar o modelo proposto de mutação.

Alguns dos operadores utilizados revelaram defeitos que ocorrem no desenvolvimento ou conversão de arquétipos, de modo que não apresentam falhas na execução, mas retornam resultados insatisfatórios.

Com base nos operadores e testes aplicados, percebe-se a utilidade dos testes para arquétipos.

Conclusão

No desenvolvimento de arquétipos e ontologias cometem-se vários erros, que podem vir a gerar defeitos na utilização de sistemas, conforme descrito na literatura³.

O teste de mutação permite a geração de dados de teste para a identificação de defeitos que não seriam revelados se o teste não fosse executado. Os resultados desse trabalho indicam que o teste de mutação é um excelente critério de teste, adequado para a validação de corretude de arquétipos e ontologias.

Dado que operadores de mutação já foram definidos, mais experiências práticas desse tipo devem ser conduzidas para a validação dessas estruturas, antes de sua implantação em sistemas reais.

Referências

1. Beale T, Kalra D, Heard S, Lloyd D. The openEHR Reference Model: EHR Information Model, agosto de 2008.
2. Poveda-Villalón M, Suárez-Figueroa MC, Gómez-Pérez A. A Double Classification of Common Pitfalls in Ontologies. Workshop on Ontology Quality (OntoQual 2010), Co-located with EKAW 2010, outubro de 2010.
3. Jepsen TC. Just What Is an Ontology, Anyway? IT Professional, 11(5):22-27, 2009.
4. PROTÉGÉ 4.3. Software para ontologias. Disponível em: <http://protege.stanford.edu/download/registered.html>, acesso em: 10 dez. 2013.
5. openEHR. Repositório de arquétipos ADL. Disponível em: <http://www.openehr.org/ckm/>, acesso em: 10 dez. 2013.
6. DL Query. DL Query Plugin. Disponível em: <http://protegewiki.stanford.edu/wiki/DLQueryTab>, acesso em 10 dez, 2013.
7. Bechhofer S, Harmelen F, Hendler J, Horrocks I, McGuinness DL, Schneider PFP, Stein LA. OWL Web Ontology Language Reference. w3c, fevereiro de 2004.
8. DeMillo RA, Lipton RJ, Sayward FG. Hints on Test Data Selection: Help for the Practicing Programmer. Computer, 11(4):34(41), abril de 1978.