

3

Trabalhos relacionados

Neste capítulo são apresentados trabalhos relacionados ao apresentado nesta tese, separados pelas áreas de análise de modelos baseada em ontologias e de verificação de modelos.

3.1.

Preâmbulo

Como já observado anteriormente, o método Observed-MAS está inserido no contexto de ESSMA, cujo foco de pesquisa tem sido o desenvolvimento de linguagens de modelagem, métodos e metodologias para o desenvolvimento de SMAs.

Dadas as características do método Observed-MAS, e o fato de que a ESSMA ainda é uma disciplina em desenvolvimento, os trabalhos relacionados ao método proposto encontram-se em áreas diversas de conhecimento. Dentre tais áreas, podemos citar a análise de modelos baseada em ontologias e a verificação de modelos, em especial modelos descritos em UML. Assim, este capítulo é dividido em duas seções, cada uma delas destinada a descrever e relacionar trabalhos nas áreas citadas ao método proposto nesta tese.

3.2.

Análise de modelos baseada em Ontologias

3.2.1.

Evermann & Wand, 2005.

Em (Evermann & Wand, 2005), os autores definem um método, baseado em ontologias, para construir modelos conceituais mais fidedignos quando confrontados com o domínio da aplicação modelada. Neste artigo os autores propõem a restrição da linguagem de modelagem para a descrição de possíveis

configurações do domínio da aplicação. Para tanto, primeiro é feita uma associação entre a semântica do domínio da aplicação para a linguagem de modelagem para, em seguida, restringir a sintaxe da linguagem de modelagem com respeito às propriedades e restrições consideradas no domínio da aplicação. No exemplo descrito foram usadas a ontologia de Bunge (Bunge, 1977 e 1979) e a linguagem UML, como prova de conceito do método. Dado um compromisso ontológico firmado pela associação da semântica contida na ontologia de Bunge a algumas abstrações de UML, os autores definiram um conjunto de regras de modelagem a partir de um pequeno conjunto de axiomas da ontologia. Estas regras servem para guiar o uso de UML durante a construção de modelos conceituais de um domínio de aplicação, porém estas regras não servem para guiar o uso de UML durante a atividade de projeto de sistemas.

O método proposto por Evermann e Wand faz o uso de ontologias no sentido oposto ao uso feito pelo método Observed-MAS, pois ele mapeia parte das ontologias de domínio para uma linguagem de modelagem, enquanto o Observed-MAS mapeia o metamodelo da linguagem de modelagem para ontologias. Ao definir regras de modelagem conceitual de aplicações num determinado domínio, baseadas em axiomas da ontologia do domínio, o método provê suporte para a construção de modelos conceituais que estão em conformidade com o domínio da aplicação. O método Observed-MAS define consultas baseadas nas propriedades intra-diagramas e inter-diagramas especificadas no metamodelo da linguagem e, posteriormente, descritas na ontologia a fim de apoiar a construção de modelos de SMAs que estão em conformidade com o metamodelo da linguagem de modelagem.

3.2.2.

Dong et al, 2004.

Em (Dong et al, 2004), Dong e colaboradores usam a linguagem formal Z e o provador de teoremas Z/EVES para fazer a verificação de modelos conceituais especificados como ontologias de domínio descritas em DAML+OIL (DAML+OIL, 2004).

Para proceder à verificação, as primitivas da linguagem DAML+OIL tiveram sua semântica descrita em Z, bem como as restrições associadas a estas primitivas. Desta forma, os autores puderam usar as ferramentas disponíveis para verificação de modelos descritos em Z para checar se a definição da

ontologia segue a especificação da sua linguagem de descrição, ou seja, se a estrutura da ontologia está correta, o que inclui a checagem de consistência da hierarquia de conceitos (*subsumption*) e, conseqüentemente, a análise da instanciação destes conceitos. Além deste tipo de verificação, também são realizadas verificações que envolvem conceitos e propriedades da ontologia, a partir da definição de teoremas usados pelo Z/EVES.

O método Observed-MAS possui uma abordagem próxima à utilizada por Dong e seus colaboradores, pois descreve a semântica de metamodelos de linguagens de modelagem para SMAs e dos diagramas associados à linguagem fim, usando uma linguagem formal (DL). Além disso, a partir desta definição é possível traduzir modelos descritos na linguagem de modelagem fim para DL e, conseqüentemente, utilizar os sistemas baseados em DL como ferramentas de suporte à análise de consistência destes modelos. Esta análise é apoiada pela descrição formal da especificação da linguagem prevista no metamodelo através de ontologias, as quais embasam a estruturação das bases de conhecimento formadas pelo mapeamento dos diagramas previstos na linguagem para instâncias da ontologia. A manipulação e consulta a estas bases de conhecimento permite a checagem de propriedades referentes à estrutura dos diagramas estáticos e dinâmicos previstos na linguagem, assim como as relações de interdependência entre eles.

3.2.3.

Kalfoglou & Robertson, 1999.

Kalfoglou & Robertson (1999) usaram ontologias para verificar a conformidade de especificações de domínios de aplicações. Para tanto, eles consideraram a conformidade da especificação de uma aplicação quando confrontada com uma descrição formal do domínio da aplicação. Os autores propuseram o uso de uma ontologia que descrevesse o domínio da aplicação para guiar o engenheiro de requisitos. Conseqüentemente, ao utilizar uma especificação formal da aplicação que é baseada numa ontologia que descreve o domínio desta aplicação, é possível verificar automaticamente a existência de inconsistências ontológicas na especificação. A fim de localizar a relação entre este trabalho e o método proposto nesta tese, considera-se o modelo de quatro camadas que descreve a arquitetura de metadados MOF (*Meta-Object Facility*) proposta pela OMG (OMG, 2003). O trabalho proposto por Kalfoglou e Robertson transita entre a camada de modelos de domínio de aplicações (M1),

onde está descrita a ontologia de domínio que guia o engenheiro de requisitos, e a camada de instâncias destes modelos (M0), onde está descrita a especificação formal da aplicação cuja conformidade é verificada considerando-se a ontologia da camada M1.

O Observed-MAS propõe uma abordagem similar, porém num nível de abstração mais elevado, ou seja, o método transita entre a camada de metamodelos (M2) e a camada de modelos de domínio de aplicações (M1). Esta diferença é significativa, pois o Observed-MAS propõe a descrição de um metamodelo através de ontologias cujas instâncias descrevem aplicações pertencentes à classe de sistemas multiagentes, independentemente do domínio de aplicação de tais sistemas.

3.3.

Verificação de modelos

Recentemente, devido ao porte e à complexidade dos modelos descritos em linguagens de modelagem, em geral UML, que representam sistemas de aplicações reais, alguns esforços de pesquisa foram direcionados para a análise sistemática destes modelos (Kim & Carrington, 2000; Berardi et al, 2001; Berardi, 2002; Straeten et al, 2003). Considerando que a inserção de agentes de software neste contexto acrescenta complexidade aos modelos, o Observed-MAS está relacionado a estes esforços no sentido de prover a análise sistemática de modelos de SMAs descritos em linguagem de modelagem.

3.3.1.

Ekenberg & Johannesson, 2004.

Ekenberg & Johannesson (2004) definiram um arcabouço baseado em lógica de primeira ordem (*First Order Logic* - FOL) para determinar a correção de modelos de projeto. O arcabouço proposto provê informações sobre a tradução de diagramas UML para FOL assim como sobre meios de detectar algumas inconsistências nos modelos. Trata-se de um arcabouço de caráter geral e o uso das regras que permitem a tradução dos modelos UML depende fortemente do conhecimento do desenvolvedor em lógica de primeira ordem, uma vez que ainda não há suporte automático para efetuar esta atividade. Os autores descrevem a análise dos modelos baseados na definição de conjuntos de

propriedades relacionadas a “restrições de integridade”, que devem ser verificadas, para modelos estáticos e para modelos dinâmicos. No contexto do Observed-MAS, estas propriedades são chamadas propriedades intra-diagramas ou propriedades internas aos diagramas. Não é definida análise inter-diagramas, isto é, a análise entre modelos estáticos e dinâmicos.

O método Observed-MAS é baseado em ontologias descritas em lógica de descrição (*Description Logic* – DL), que é um subconjunto decidível da FOL. A tradução dos modelos de SMAs para a linguagem de descrição de ontologias pode ser feita automaticamente, usando sistemas como RICE (RICE, 2004) ou Protégé (Protégé, 2004) associados ao aparato dedutivo fornecido pelo sistema RACER. Além disso, a análise dos modelos é automatizada pelo uso do aparato dedutivo do sistema RACER. O arcabouço proposto por Ekenberg e Johannesson foi definido para verificação de modelos de aplicações que usam a abordagem de orientação a objetos, e, portanto, não prevê nenhum recurso para análise das abstrações relacionadas a agentes e SMAs. O método Observed-MAS considera análises relativas às propriedades internas de diagramas estáticos e dinâmicos, além das propriedades de interdependência entre diagramas estáticos ou entre diagramas estáticos e dinâmicos.

3.3.2.

Straeten, Simmonds e Mens, 2003.

Straeten e colaboradores (Straeten et al, 2003) apresentam o uso de DL para detectar inconsistências em diagramas UML, assim como para manter a consistência entre diagramas UML, no contexto da evolução de software. Dadas as características do contexto do trabalho, os autores consideram apenas a consistência entre diagramas estáticos e dinâmicos, mais precisamente, entre diagramas de classe e diagramas de sequência. Os autores definem o conflito *classless instance*, como a inconsistência entre diagramas de sequência e diagramas de classe onde uma instância utilizada num diagrama de sequência é instância de uma classe que não aparece em nenhum diagrama de classe da modelagem.

No método Observed-MAS também é feita a checagem de consistência entre modelos, porém no contexto de modelos de sistemas multiagentes. Esta diferença de contexto implica na checagem de consistência não só entre diagramas estáticos e dinâmicos, mas também entre diagramas estáticos e estáticos, além da checagem de cada diagrama considerado isoladamente. Além

disso, estendeu-se o conceito *classless instance* para instâncias de outros tipos de classes definidas pela linguagem de modelagem utilizada na prova de conceito, tais como `agent-role-class`.

3.3.3.

Berardi, 2001 e 2002.

Berardi e colaboradores (Berardi et al, 2001; Berardi, 2002) investigam o uso de DL para descrever formalmente diagramas de classe UML e os sistemas FaCT e RACER para classificar tais diagramas relativamente à sua consistência. A abordagem é similar ao Observed-MAS, pois a descrição feita em DL pode ser considerada uma ontologia para diagramas de classe UML, apesar de não haver menção ao termo ontologia em nenhum dos artigos. Entretanto, mesmo que fosse considerada a ontologia que descreve diagramas de classe UML, ela seria baseada num metamodelo orientado a objetos enquanto no Observed-MAS a ontologia usada para descrever os modelos é baseada num metamodelo orientado a SMAs. O método Observed-MAS provê suporte para a análise das propriedades internas não apenas dos diagramas estáticos, mas também dos diagramas dinâmicos previstos no metamodelo da linguagem de modelagem orientada a SMAs. Além disso, também apóia a análise das propriedades de interdependência entre os diagramas estáticos e entre os diagramas estáticos e dinâmicos.

3.3.4.

Kim & Carrington, 2000.

Kim & Carrington (2000) descreveram a tradução de um diagrama de classe UML para uma especificação formal descrita em Object-Z (Smith, 1999). Apesar de apresentar a tradução do diagrama para uma linguagem formal, os autores não descrevem como seria feita a verificação automática destes modelos. Esta também foi uma limitação pela qual nos deparamos durante as primeiras investigações sobre análise de modelos de SMAs, quando foi definida uma extensão de Object-Z para analisar SMAs descritos em MAS-ML, chamada AgentZ (Brandão et al, 2004; 2005). Por este motivo buscou-se novas formas de prover a análise destes modelos, culminando no uso de ontologias e sistemas

baseados em conhecimento para analisar modelos de SMAs descritos em linguagens de modelagem.

O Observed-MAS provê suporte para a descrição formal de modelos de SMAs através do uso de ontologias que descrevem o metamodelo da linguagem de modelagem para SMAs. A verificação de consistência destes modelos é automatizada, e utiliza técnicas de raciocínio baseado em conhecimento através do uso de consultas a bases de conhecimento, proporcionadas pelo aparato dedutivo de um sistema de representação de conhecimento, o sistema RACER.

3.4.

Discussão

Neste capítulo foram descritos trabalhos relacionados ao apresentado nesta tese considerando-se duas abordagens: a verificação de modelos baseada em ontologias e a verificação de modelos UML.

Quando considerada a abordagem relacionada a verificação de modelos baseada em ontologias observa-se que a literatura pertinente é focada na descrição de domínios de aplicação através de ontologias para então verificar se o modelo conceitual da aplicação satisfaz os requisitos especificados pelo engenheiro de requisistos. Ao considerar um nível de abstração acima, ou seja, usar ontologias para verificar se os modelos de uma classe de aplicações (desenvolvidas no domínio de sistemas multiagentes) estão bem formados, aumenta-se o alcance de uso das ontologias sem impedir o uso de ontologias no nível de domínio da aplicação.

As abordagens anteriores não propõem um método que dê suporte para a construção da ontologia que será usada na verificação, nem indica se a verificação é declarativa (indicando onde os problemas ocorrem) ou simplesmente afirmativa (informando que o modelo apresenta problemas). Nesta tese é proposto um método para a análise de modelos de SMAs baseado em ontologias que descrevem o domínio de SMAs e cuja aplicação não só indica ao projetista onde estão os problemas de má formação dos modelos como também indica pontos onde o modelo pode ser melhorado.

Quando considerada a abordagem relacionada a verificação de modelos descritos em UML observa-se o uso de formalismos vários, como FOL, Object-Z e DL. Quando a verificação é baseada em FOL ou Object-Z, sua aplicabilidade depende fortemente do conhecimento destes formalismos por parte do desenvolvedor. Além disso, o suporte para que tais verificações sejam feitas

automaticamente, quando disponível, ainda é precário. O uso de DL para verificação destes modelos supre a lacuna da verificação automática, pois existem vários sistemas eficientes e disponíveis (RACER, 2004; FaCT, 2003, Powerloom, 2005), porém não é suficiente para analisar modelos descritos em linguagens que estendem UML para o domínio de SMAs e nem há indicação de como isto poderia ser feito. Ao considerar o uso de DL associado a uma ontologia que descreve o domínio de SMAs apresenta-se uma direção a ser seguida quando se deseja verificar modelos de uma classe de aplicações descritos em linguagens que estendem UML. Neste sentido, uma possível aplicação da abordagem proposta nesta tese seria, por exemplo, analisar a classe de aplicações descritas usando uma linguagem orientada a aspectos e que estende UML, como proposta por Chavez (2004).