

8

Discussões Finais e Conclusões

Neste capítulo, apresentamos as conclusões de nosso trabalho. Inicialmente, porém, confrontamos nossa solução com as características da abordagem *RoleEP/Epsilon/J*. Somente esta abordagem procurou dar suporte à introdução transparente de mobilidade em aplicações (Seção 3.4). Procuramos então realizar uma comparação crítica das características de nossa solução com àquelas provenientes da abordagem *RoleEP/Epsilon/J*. Posteriormente, apresentamos as contribuições do trabalho, além de propostas para trabalhos futuros.

8.1. Comparação entre *RoleEP/EpsilonJ* e *ArchM/AspectM*

A abordagem *RoleEP* e o *framework Epsilon/J* de Ubayashi e Tamai (Seção 3.4) fornecem: (1) um mecanismo para separação dos interesses de mobilidade/colaboração das funcionalidades básicas nas aplicações, e (2) um estilo de programação que permite a evolução sistemática e dinâmica de SMAs móveis. Para resolver estes dois problemas, *RoleEP* define dois conceitos principais: um espaço onde agentes se movimentam entre *hosts* e colaboram com outros agentes é denominado “ambiente” e as funções de mobilidade/colaboração que um agente pode assumir em um dado ambiente são denominadas “papéis”. Através destas abstrações, *RoleEP* possibilita que SMAs móveis mudem suas funções dinamicamente a fim de adaptarem-se a contextos externos.

A primeira observação importante na comparação entre a abordagem *RoleEP/EpsilonJ* e aquela proposta por este trabalho diz respeito às diferenças existentes entre os objetivos centrais das duas abordagens. A arquitetura *ArchM* e o *framework AspectM* não possuem entre seus objetivos a busca por uma solução para o problema de evolução sistemática e dinâmica de SMAs móveis. A interseção dos objetivos se restringe ao problema da separação explícita dos interesses de mobilidade no código das aplicações de SMAs móveis. Mesmo assim, veremos adiante que em *RoleEP* a separação de mobilidade é alcançada

apenas parcialmente, uma vez que os interesses de mobilidade são tratados em conjunto com os interesses de colaboração.

Embora a evolução dinâmica de SMAs móveis não figure entre os objetivos centrais deste trabalho, o estilo de programação para evolução sistemática de SMAs de *RoleEP* pode ser comparado ao processo de instanciação de uma aplicação no *framework* AspectM, na medida em que nossa abordagem possui como foco os requisitos de reusabilidade e manutenibilidade de aplicações. De fato, os resultados obtidos com este trabalho permitem afirmar que uma abordagem orientada a aspectos para a separação dos interesses de mobilidade possibilita um bom nível de evolução sistemática das aplicações. Entretanto, não exploramos cenários onde a mobilidade é dinamicamente associada a tipos e/ou papéis de agentes. O objetivo de introdução transparente de mobilidade é tratado somente em tempo de compilação, isto é, não verificamos a influência do processo de combinação dinâmico da mobilidade a tipos e/ou papéis de agentes. Por esta razão, permitimos um nível adequado de evolução sistemática de SMAs móveis, mas a introdução de interesses dinâmicos de mobilidade nas aplicações foi um ponto não estudado completamente.

Outro ponto importante a observar na comparação entre as duas abordagens é que o fato de *RoleEP* se fundamentar sobre conceitos próprios de ambientes e papéis traz distinções importantes face ao uso de abordagens tradicionais e da solução proposta neste trabalho para o problema de separação dos interesses. Em geral, nos SMAs móveis o ambiente de um agente, isto é, o seu contexto de execução, é um espaço limitado pelas configurações de acesso do servidor de cada *host* acessível ao agente. Um ambiente não se estende por mais de um *host*. Por outro lado, o conceito de papéis normalmente está associado à idéia do desempenho de funções em organizações. É possível construir SMAs móveis onde o conceito de papéis não é utilizado diretamente, quando os requisitos das aplicações não impõem a utilização deste conceito. Como estes conceitos não são fundamentais nas abordagens tradicionais, uma aplicação pode ser construída somente a partir da instanciação e manipulação de objetos. Em abordagens baseadas no DSOA como o nosso trabalho, também é possível o desenvolvimento de uma aplicação somente a partir da manipulação de objetos mesmo que não existam aspectos que entrecortem suas propriedades. Ao contrário, em *RoleEP*

uma aplicação não pode ser construída somente com objetos, sem que o conceito de ambientes e papéis próprios da abordagem sejam utilizados.

Quando um ambiente é definido como um contexto de execução limitado pelo servidor de cada *host* e papéis como um conjunto de funções desempenhadas em organizações, a separação dos interesses de mobilidade e a separação dos interesses de colaboração constituem problemas que não necessariamente devem ser tratados em conjunto. *RoleEP*, ao contrário, trata da separação dos interesses de mobilidade e colaboração como um todo, assumindo que tais conceitos necessariamente estão associados em SMAs móveis. O problema que nos propomos a resolver neste trabalho é mais abrangente, uma vez que temos por objetivo a separação explícita dos interesses de mobilidade de quaisquer outros interesses em SMAs. Mais especificamente, propõe-se uma solução que permita não somente a separação das funções de mobilidade/colaboração dentro de um sistema, como também a separação das funções de mobilidade de quaisquer outros interesses, inclusive de colaboração e funcionalidades básicas dos agentes.

Adicionalmente, assumimos que a separação dos interesses de colaboração e mobilidade pode ser tratada por *frameworks* diferentes, apesar de que a mesma abordagem orientada a aspectos possa ser aplicada. A princípio, a instanciação de um papel móvel no *framework* AspectM é suficiente para tratar colaboração e mobilidade em conjunto. Se isto se verificar na implementação de novos estudos onde interesses de colaboração e mobilidade estejam estreitamente relacionados, então um alto nível de flexibilidade é alcançado com o uso do *framework* AspectM. Porém, até o instante presente, no *framework* AspectM um agente adquire papéis e mobilidade apenas em tempo de compilação, enquanto a abordagem *RoleEP* tende a ser mais genérica e efetiva em aplicações com foco na associação dinâmica de papéis a agentes.

O fato de a abordagem ArchM/AspectM ser baseada no DSOA também conduz a importantes diferenças em comparação com a abordagem *RoleEP/EpsilonJ*. Em *RoleEP/EpsilonJ*, o uso da operação de ligação (Seção 3.4) elimina a necessidade do processo de combinação do DSOA (Seção 4.2.1). Por outro lado, as declarações intertipos de *AspectJ* podem ser substituídas pela adição de métodos de papel através da operação de ligação (Seção 3.4). Contudo, adendos de *AspectJ* não possuem construções correspondentes no modelo de *RoleEP*. Este é um ponto fraco desta abordagem, pois reduz a capacidade de

prevenir a replicação de código nas aplicações. No contexto de SMAs móveis, a inexistência de mecanismos correspondentes a adendos se apresenta como uma restrição importante para a separação dos interesses de mobilidade. Não é possível modularizar interesses do protocolo genérico de mobilidade, como a checagem de pré e pós-condições da movimentação de uma agente.

Apresentamos a seguir um confronto entre alguns comentários de Ubayashi e Tamai (2001) quanto ao uso de uma abordagem orientada a aspectos e as soluções propostas neste trabalho para os problemas apontados. O primeiro ponto é que a definição de aspectos é dependente do nome de classes específicas. Por esta característica, um aspecto não poderia ser definido como entrecortando várias classes. Porém, através da utilização de *Container Introduction* (Seção 6.4), foi possível resolver este problema. A idéia geral é introduzir elementos de um aspecto em uma interface genérica através de declarações intertipos de *AspectJ* e, posteriormente, declarar a implementação desta interface em pontos convenientes das aplicações. Em várias classes são, portanto, introduzidos os elementos definidos em um mesmo aspecto.

Outro ponto diz respeito à definição dinâmica de papéis. Como a definição de papéis através de aspectos depende do nome de métodos de classes específicas, para definir novos pontos de junção onde a mobilidade deve ser introduzida, será sempre necessário recompilar os aspectos envolvidos. De fato, na abordagem ArchM/AspectM, a definição de novos pontos de junção onde os interesses de mobilidade são associados a um papel é uma tarefa realizada em tempo de compilação. Já explicamos acima que a influência da utilização do processo de combinação dinâmico na solução para o problema de separação dos interesses de mobilidade não foi tratada completamente neste trabalho. Entretanto, quando precisamos definir um novo ponto de junção onde a mobilidade deve ser introduzida, é provável que seja sempre importante ter um controle estático sobre a característica a ser adicionada face à necessidade de entendimento do comportamento de um tipo e/ou papel de agente como um todo. Este procedimento ajuda a prever erros na execução dos sistemas.

Finalmente, a desvantagem mais clara no uso da abordagem *RoleEP/EpsilonJ* é que os desenvolvedores devem estender várias classes abstratas do *framework Epsilon/J* para instanciar SMAs móveis. Como demonstrado anteriormente, a necessidade de extensão de classes de um

framework, usando mecanismos de orientação a objetos, diminui a modularização dos interesses de mobilidade (Seção 1.1). Além disso, uma limitação importante é o fato de *Epsilon/J* ser dependente de *Aglets*, isto é, outras plataformas de mobilidade não podem ser utilizadas pelos usuários, a menos que *Epsilon/J* seja reimplementado para permitir esta flexibilidade. Entretanto, é importante enfatizar que estas particularidades das abordagens *RoleEP/EpsilonJ* e *ArchM/AspectM* são devidas ao fato de possuírem objetivos diferentes. Nosso trabalho tem como foco principal a reusabilidade e a manutenibilidade de SMAs móveis. A abordagem *RoleEP/EpsilonJ* tem como objetivo central a evolução sistemática de aplicações onde a associação dinâmica de papéis a tipos de agentes é um requisito fundamental.

8.2. Contribuições

O conceito e o modelo de agentes móveis parecem bastante adequados à concepção e construção de sistemas computacionais distribuídos ou heterogêneos. Isto porque, contrariamente ao que acontece nos outros paradigmas da computação distribuída, no paradigma promovido pelos agentes móveis a facilidade de transferência é levada ao extremo. A característica de mobilidade é fornecida não somente ao código que deve ser processado, como nos sistemas baseados em execução remota ou em código sob demanda, mas aos próprios componentes que efetuam o processamento das aplicações, os agentes móveis.

Entretanto, apesar dos benefícios obtidos na utilização do paradigma de agentes móveis para o desenvolvimento de sistemas, evidências empíricas demonstram que abstrações e mecanismos OO não são suficientes para a separação de interesses transversais, como a mobilidade de agentes de software. Sem as abstrações e os mecanismos adequados à separação destes interesses, a mobilidade tende a ficar espalhada e entrelaçada com outros interesses nos SMAs. Portanto, a separação explícita de interesses de mobilidade é fundamental para o desenvolvimento de SMAs reutilizáveis e manuteníveis.

Este trabalho teve como principal objetivo o desenvolvimento de um *framework* orientado a aspectos para mobilidade de agentes de software. Propomos o *framework* *AspectM* como uma solução para os problemas de

espalhamento e entrelaçamento de código de mobilidade em SMAs. As contribuições geradas por nosso trabalho durante o processo de desenvolvimento do *framework* AspectM podem ser traduzidas nos seguintes artefatos para a engenharia de software de SMAs móveis:

- documentação e análise das soluções existentes para o problema de modularização de interesses de mobilidade em SMAs: *frameworks* OO de plataformas de mobilidade, padrões de projeto e abordagem *RoleEP/EpsilonJ*;
- a arquitetura ArchM, cujo processo de abstração se deu a partir do *survey*/análise das outras soluções existentes para o problema de separação de mobilidade e a partir do estudo/extensão da arquitetura orientada a aspectos para mobilidade proposta por Garcia (2004);
- o *framework* AspectM, cujo desenvolvimento foi baseado nas diretivas propostas por ArchM. A utilização de AspectM promove não só a modularização de interesses de mobilidade, como também de flexibilização do uso de plataformas de mobilidade;
- as instâncias de pacotes de classes que realizam a comunicação do *framework* AspectM com as plataformas JADE e *Aglets*;
- o desenvolvimento dos estudos de caso *Expert Committee* e *MobiGrid*, que consistem não só em duas aplicações de natureza diferenciada e que utilizam recursos de diferentes plataformas, mas que também são mantidos por diferentes grupos de pesquisa.

Para o desenvolvimento dos artefatos de software listados acima, foi importante nosso trabalho prévio com SMAs móveis, cujos resultados foram publicados em Lobato e Damasceno (2003). Também é importante mencionar que a primeira versão do *framework* AspectM e da arquitetura ArchM foram publicados em Lobato et al. (2004). Além disso, nosso trabalho foi escolhido recentemente pelo AOSD-Europe (AOSD-Europe) como exemplo de aplicação promissora de POA. O AOSD-Europe é um consórcio de excelência de pesquisa em DSOA, financiado pela Comissão Européia e envolvendo 9 universidades e 2 companhias (IBM e Siemens).

Finalmente, tanto do ponto de vista do problema quanto da solução proposta neste trabalho, a separação explícita dos interesses de mobilidade efetivada por ArchM/AspectM não somente possibilita a melhora da manutenibilidade e do reuso do código de mobilidade, mas também permite um melhor reuso das funcionalidades básicas e dos outros interesses afetados pela ausência de uma separação adequada de mobilidade. Em outras palavras, com a modularização provida por ArchM/AspectM, as classes das funcionalidades básicas e outros interesses de SMAs tornam-se também mais reutilizáveis, pois é possível reusar tais classes sem a preocupação com o código de mobilidade usualmente espalhado e entrelaçado em SMAs.

8.3. Trabalhos Futuros

Como trabalhos futuros, pretendemos:

- implementar novas instâncias do *framework* AspectM, para aplicações desenvolvidas por outros grupos de pesquisa;
- avaliar empiricamente os requisitos do protocolo de mobilidade que não foram tratados nos estudos de caso deste trabalho (Tabela 3);
- implementar novas instâncias do *framework* AspectM, para tornar possível a integração com os *frameworks* OO disponibilizados por outras plataformas de mobilidade;
- inclusive, avaliar e, se possível, integrar o *framework* AspectM com *frameworks* baseados em outros paradigmas, como, por exemplo, o paradigma *blackboard*;
- explorar o uso de combinadores dinâmicos em cenários do protocolo de mobilidade nas novas instâncias de aplicação geradas a partir do *framework* AspectM;
- aplicar métricas, tais como as definidas em Sant'Anna et al. (2003), que permitam a avaliação quantitativa do projeto e da implementação de aplicações instanciadas a partir de AspectM para a modularização dos interesses de mobilidade;
- publicação de artigos em conferências relevantes da área de pesquisa de SMAs móveis.