

6 Conclusão

Neste capítulo, procuramos contextualizar nosso trabalho em um quadro mais amplo de trabalhos de ESSMA para domínios complexos de aplicação. Após a seção de trabalhos relacionados, apresentamos um resumo das contribuições do trabalho, além de propostas para trabalhos futuros.

6.1. Trabalhos Relacionados

O gerenciamento da complexidade durante as diferentes fases do processo de desenvolvimento de software depende da disponibilidade de métodos e técnicas adequados de estruturação de sistemas. Na Engenharia de Software, a questão da complexidade está diretamente relacionada à modularização e à decomposição dos sistemas (Parnas, 1972). Os sistemas de software complexos devem ser decompostos em unidades menores e claramente modularizadas, cada uma lidando com uma única característica (Parnas, 1972).

A Orientação a Objetos (OO) foi durante muitos anos o paradigma de desenvolvimento dominante e seus benefícios são atualmente amplamente reconhecidos. As classes, os objetos e os métodos são exemplos de abstrações clássicas do paradigma OO. Por exemplo, um único conceito do sistema pode ser modularizado como uma classe ou um método. A herança e o polimorfismo, por sua vez, são exemplos de mecanismos que permitem a decomposição de conceitos complexos de software.

Nos últimos anos, surgiram algumas técnicas para melhorar a modularidade dos sistemas OO, como padrões de projeto (Gamma et al, 1995) e *frameworks* (Fayad et al, 1999). Entretanto, com o aumento da complexidade, tem se tornado cada vez mais difícil conseguir um nível adequado de modularização e decomposição dos sistemas apenas tendo como base abstrações e mecanismos OO. Algumas técnicas estão surgindo para oferecer melhor suporte à

modularidade, as quais não são diretamente capturadas *apenas* com abstrações e mecanismos da Orientação a Objetos.

Por outro lado, a Engenharia de Software para Sistemas Multi-Agentes tem se apresentado como paradigma promissor para o desenvolvimento de aplicações distribuídas, abertas e extensíveis (Jennings, 1999; Lind, 2000; Garcia, 2004; Silva, 2004; Garcia, 2005). A usabilidade de ESSMA tem sido verificada em vários domínios complexos, como tráfego aéreo, telecomunicações, redes sem-fio, medicina, reputação e computação autônoma (SISTEMAS MULTI-AGENTES), em que, notadamente, a OO não dá suporte a um bom nível de modularização e decomposição dos sistemas.

De acordo com o nosso conhecimento, somente o trabalho de Harroud et al (2004) teve por objetivo a verificação da usabilidade das abstrações e mecanismos de decomposição propostos por ESSMA a fim de diminuir a complexidade do desenvolvimento de ASCs. Entretanto, do ponto de vista da Engenharia de Software, a principal limitação do estudo de Harroud et al (2004) é que sua contribuição é apresentada de tal forma que se constitui como *solução alternativa* àquela proposta por outros *middlewares* usados no desenvolvimento de ASCs (Chen, 2004; Khedr & Karmouch, 2004; Rubinsztein et al, 2004).

Em comparação com Harroud et al (2004), pensamos que a relevância deste trabalho consiste no fato de que, através de nossas contribuições (Seção 6.2), os engenheiros de software são providos de abstrações e mecanismos de decomposição para o desenvolvimento de ASCs mais próximos dos negócios no mundo real, além de permitir o desenvolvimento de ASCs em conjunto com outras ferramentas freqüentemente utilizadas, como JADE (Bellifemine et al, 1999) e MoCA (Rubinsztein et al, 2004).

Portanto, podemos afirmar que, em vez de propor uma solução alternativa às soluções atualmente existentes, nosso trabalho deve ser visto como um *complemento* ao uso de *middlewares publish-subscribe* para desenvolvimento de ASCs e a partir do reuso da infra-estrutura oferecida por plataformas de agentes.

6.2. Contribuições do Trabalho

As abstrações e mecanismos de ESSMA parecem bastante adequados à concepção e construção de sistemas distribuídos ou heterogêneos como as ASCs. Isto porque, contrariamente ao que acontece na OO, quando usamos ESSMA, os requisitos relacionados à distribuição dos sistemas podem ser mais satisfatoriamente contemplados. De fato, partindo das abstrações e mecanismos de decomposição propostos por ESSMA, apresentamos, neste trabalho, soluções complementares às atualmente utilizadas para o desenvolvimento de ASCs, as quais podem ser consideradas como aplicações *inerentemente* distribuídas.

Nossas contribuições podem ser resumidas nos seguintes tópicos:

1. levantamento de requisitos de ASCs especificamente relacionados à sensibilidade ao contexto e a análise dos conceitos potenciais de ESSMA que podem ser utilizados para a satisfação de tais requisitos;
2. engenharia reversa das ASCs *Virtual Lines* e *Wireless Marketing Service* para a obtenção de documentação em UML inicialmente indisponível no site do LAC⁷ da PUC-Rio (LAC);
3. reengenharia de *Virtual Lines* e *Wireless Marketing Service* usando as abstrações e mecanismos de decomposição propostos por ESSMA;
4. proposta de CAAF, um *framework* desenvolvido para facilitar o desenvolvimento de ASCs em conjunto com outras ferramentas existentes, como *middlewares* para ASCs e plataformas de agentes;
5. instanciação da aplicação *Health Care* a partir de CAAF;
6. análise quantitativa de *Virtual Lines* e *Wireless Marketing Service* codificadas *com* e *sem* o uso do paradigma de ESSMA;
7. análise qualitativa dos resultados encontrados a partir da instanciação de *Health Care*;
8. demonstração dos benefícios do uso de abstrações e técnicas de ESSMA em ASCs, através da medição de atributos internos de software, como acoplamento, coesão e tamanho.

⁷ Do inglês: *Laboratory for Advanced Collaboration*.

6.3. Trabalhos Futuros

Como trabalhos futuros, pretendemos:

- levantar novos requisitos de ASCs especificamente relacionados à sensibilidade ao contexto aos quais possam ser aplicadas as abstrações e os mecanismos de decomposição propostos por ESSMA;
- investigar a existência de outros conceitos de ESSMA que possam ser aplicados para o aumento da modularidade e reuso de ASCs, como outros tipos de agentes, por exemplo, colaborativos e regulados;
- aumentar o nível de “variabilidade” de CAAF a partir de requisitos complexos de ASCs existentes e a fim de que novas instâncias do *framework* possam ser obtidas;
- comparar outras ASCs existentes com novas versões obtidas a partir de sua reengenharia utilizando CAAF;
- a partir de um número adequado de estudos de caso, realizar uma análise mais profunda de custos e benefícios obtidos quando do uso de ESSMA no desenvolvimento de ASCs;
- publicar artigos em conferências relevantes das áreas de pesquisa de Computação Ubíqua e Sistemas Multi-Agentes.