

5 Conclusões e Trabalhos Futuros

5.1 Conclusões

Neste trabalho foi proposta a utilização do padrão arquitetural Reflective Blackboard [52,53,54] como base para arquiteturas de software para sistemas multi-agentes. Este padrão visa oferecer tratamento de forma separada para as diferentes estratégias de controle incorporadas em sistemas multi-agentes. De forma complementar à proposta do padrão foi desenvolvida uma implementação para este que visa facilitar o desenvolvimento de sistemas multi-agentes baseados na arquitetura proposta.

Através da utilização do padrão alguns benefícios são alcançados:

- Tratamento separado das estratégias de controle;
- Tratamento separado de requisitos não funcionais;
- Maior facilidade de reutilização e manutenção;
- Modularidade.

Além destes benefícios a utilização do padrão traz as vantagens comuns ao emprego de padrões em engenharia de software como vocabulário e estrutura comuns. Desta forma a reutilização da arquitetura e a integração com outros sistemas que implementem o mesmo padrão é facilitado.

Além da definição do padrão foi apresentado o framework T-Rex [50,51,56], desenvolvido para oferecer suporte a sistemas multi-agentes baseados no padrão Reflective Blackboard. Além de facilitar a implementação de estratégias de controle e coordenação de sistemas multi-agentes, T-Rex também oferece suporte às propriedades inter-agentes de mobilidade, persistência e comunicação. Desta forma, ao contrário de outros trabalhos relacionados, a composição destas propriedades inter-agentes através de estratégias de controle é mais simples e transparente.

A utilização de T-Rex foi exemplificada e validada através da aplicação de Marketplace desenvolvida. Nesta aplicação, diferentes tipos de estratégias de controle, relativas a regras específicas do Marketplace, à composição da comunicação com mobilidade e persistência dos agentes e a requisitos não funcionais como segurança de informações foram geradas.

As propostas deste trabalho foram desenvolvidas no contexto do grupo SOC+Agents [58] como a abordagem para o tratamento de propriedades inter-agentes [18,57]. Dentro deste contexto a abordagem apresentada aqui seria composta com o tratamento das propriedades intra-agentes baseado em aspectos [50,21,24,25].

5.2 Contribuições

As principais contribuições deste trabalho foram apresentadas em uma série de publicações em eventos nacionais e internacionais e em um capítulo de livro. Estas publicações são apresentadas abaixo:

- **The Reflective Blackboard Pattern: Architecting Large-Scale Multi-Agent Systems** [53], capítulo do livro “Software Engineering for Large-Scale Multi-Agent Systems”, que apresenta em detalhes o padrão arquitetural Reflective Blackboard. Uma versão resumida este artigo [52] foi apresentada no “1st International Workshop on Software Engineering for Large-Scale Multi-Agent Systems”;
- **T-Rex: A Reflective Tuple Space Environment for Dependable Mobile Agent Systems**[51], artigo apresentado no III Workshop de Comunicação Sem Fio e Computação Móvel, que apresenta a infraestrutura desenvolvida para o tratamento de diferentes propriedades do nível do sistema através da utilização de técnicas de reflexão computacional e de espaços de tuplas;
- **A Unified Software Architecture for System-Level and Agent-Level Dependability in Multi-Agent Object-Oriented Systems** [50], publicado nos anais do “7th ECOOP Workshop on Mobile Objects Systems”, que mostra a utilização da infra-estrutura desenvolvida neste trabalho em conjunto com uma abordagem orientada a aspectos com objetivo de prover tratamento separado para os níveis do agente e do sistema onde ele executa;
- **Promoting Advanced Separation of Concerns in Intra-Agent and Inter-Agent Software Engineering** [18], artigo apresentado no “Workshop on Advanced Separation of Concerns in Object-oriented Systems (ASoC) at OOPSLA'2001” que em conjunto com um pôster [57]apresentado na mesma conferência, mostra como as propostas presentes nesta dissertação podem ser aplicadas em conjunto com outras

técnicas de separação de responsabilidades, no desenvolvimento de sistemas multi-agentes;

- **Um Ambiente Para Aplicações de Agentes Móveis Baseado em Arquiteturas Reflexivas de Espaços de Tuplas** [56], apresentado no Workshop de Teses em Engenharia de Software (WTES) do Simpósio Brasileiro de Engenharia de Software (SBES), que apresenta de forma resumida o conteúdo deste trabalho;

5.3 Trabalhos Futuros

Apesar de alguns resultados e contribuições já terem sido alcançados no que se refere à proposta de arquitetura e ao desenvolvimento de T-Rex existem alguns pontos que devem ser trabalhados para que se atinja um ambiente de desenvolvimento de sistemas multi-agentes robusto.

Em primeiro lugar é necessário o desenvolvimento de mais aplicações que possam mostrar os pontos fracos do padrão arquitetural proposto. Neste sentido, seria interessante utilizar diferentes abordagens que tratem das propriedades intra-agentes de forma a validar a independência entre o padrão e estas abordagens.

Outro ponto importante refere-se a questões específicas relativas à implementação de T-Rex. Um deles se refere ao modelo de transações implementado, uma vez que este controle não está completamente implementado em TSpaces. Na implementação de T-Rex para garantir a atomicidade de desvios para o meta-nível foram utilizados os mecanismos de monitores existentes na linguagem Java [36]. Apesar de através dos testes realizados não terem sido encontrados problemas relativos a deadlocks ou condições de corrida específicas, um melhor modelo transacional deveria ser utilizado.

Outro trabalho importante a ser realizado é a formalização do mecanismo de regras existente em T-Rex. Através deste tipo de formalização poder-se-ia especificar de maneira mais rigorosa as regras que compõem estratégias de controle do sistema multi-agente. Ainda no mecanismo reflexivo é importante definir um mecanismo de especificação da ordem de execução de reações quando múltiplas meta tuplas estão associadas a uma mesma operação. Atualmente este processo deve ser especificado no código das próprias reações pelos programadores do meta-nível. Trabalhos relacionados utilizam uma abordagem que se baseia na ordem de escrita de meta-tuplas

no meta-nível, mas acredita-se que esta abordagem não é satisfatória, levando a necessidade de um mecanismo mais sofisticado de especificação.

É de extrema importância também a especificação de uma metodologia de desenvolvimento de sistemas multi-agentes baseados na arquitetura proposta, desde sua análise até sua implementação. Esta metodologia deve considerar também a modelagem e implementação dos aspectos intra-agentes através de abordagens específicas ou genéricas.