

1

Introdução

Atualmente, o desenvolvimento de sistemas complexos, dinâmicos e interativos é uma realidade em nossa sociedade. Com um alto nível de penetração de sistemas de software no dia a dia de pessoas e empresas, a integração com qualidade entre eles torna-se necessária e um desafio técnico interessante para pesquisa em engenharia de software. Integração demanda comunicação entre os sistemas que interagem em prol do atendimento de um objetivo compartilhado, a partir da execução conjunta de procedimentos interrelacionados [Castelfranchi, 1992; Odell et al., 2002]. Este cenário é um ambiente propício para a aplicação da tecnologia de agentes de software [Mathieu et al., 2002; Zambonelli, 2002; Sycara et al., 2003; Zambonelli et al., 2003].

Em um sistema multiagente aberto, agentes de software podem entrar e sair e não existe um controle único e centralizado sobre o desenvolvimento das partes que compõem o sistema [Agha, 1997]. Estas características podem acarretar riscos que tornam o ambiente imprevisível [Friedrikson et al., 2003; Avizienis et al., 2004].

Para limitar o grau de imprevisibilidade destes ambientes é possível desenvolver entidades que observam os comportamentos dos agentes e os comparam com o comportamento esperado para o sistema [Minsky et Ungureanu, 2000]. Esta observação corresponde à interceptação de mensagens da interação entre os agentes e a verificação de se a interação está ocorrendo de acordo com o previsto. A partir desta percepção é possível impor uma diversidade de tratamentos a estas mensagens como bloqueio da mensagem, registro, aviso ou mesmo penalidades como o isolamento de determinado agente do contexto de colaboração, dentre outros.

A técnica resumida acima é chamada de governança de sistemas multiagentes abertos. Agentes de software precisam estar integrados para realizar suas atividades a partir da interação por troca de mensagens [Huhns et Stephens, 1999]. Esta interação pode ser especificada em protocolos de interação bem definidos. No entanto, esta especificação não limita comportamentos indesejados

no sistema multiagente aberto. Para que o comportamento observado seja o mais próximo possível do previsto é preciso descrever leis de interação que são interpretadas por um mediador responsável pela interceptação de mensagens e observação da conformidade com as leis [Paes et al., 2007]. Este mediador também é conhecido como mecanismo de governança de leis de interação. Um mecanismo é customizado a partir da interpretação das leis de interação especificadas [Paes et al., 2005].

Ao observar a natureza das interações é possível perceber que as leis utilizadas em diferentes instâncias de um mesmo domínio apresentam características comuns e pequenas variações, que se bem identificadas, podem ser isoladas de forma a promover a reutilização do projeto e implementação destas leis de interação. Como exemplo, a especificação de uma lei de interação pode ser feita de forma abstrata para ampliar as possibilidades de sua aplicação [Smith, 1980].

O projeto e a implementação de um mecanismo de governança de leis requerem um esforço de aplicação e amadurecimento de técnicas para limitar com eficácia imprevisibilidades ou comportamentos não desejados. Desenvolver um mecanismo como este, a partir do zero, pode não ser um esforço trivial. Desenvolver estes mecanismos com qualidade, é mais difícil ainda. Por fim, ao se falar de uma gama crescente de aplicativos desta natureza em um mesmo domínio, este esforço pode ser multiplicado. Apoiar este cenário é o objetivo principal desta tese.

A técnica de frameworks orientados a objeto (OO) [Fayad et al., 1999] é aplicada com sucesso na produção de famílias de aplicativos complexos orientados a objeto em um mesmo domínio. O princípio básico do framework OO é prever como resultado do desenvolvimento um artefato projetado e implementado, de forma semi-completa, que identifique claramente os componentes que serão reutilizados nas instâncias (núcleo) e as customizações que precisam ser feitas (pontos de extensão). A partir de uma etapa de materialização dos pontos de extensão, chamada de instanciação, um novo aplicativo é gerado.

Como hipótese e proposta deste trabalho, pretende-se aplicar e adaptar conceitos de frameworks OO para uma melhor engenharia na produção e manutenção de famílias de mecanismos de governança de leis de interação. Esta

proposta foi chamada de g-frameworks, ou frameworks de governança de sistemas multiagentes abertos. As duas principais preocupações desta proposta são a flexibilidade e a reusabilidade de leis de interação, isto é, a possibilidade de modificar o mecanismo de governança e a possibilidade da aplicação dos artefatos produzidos em diferentes contextos.

É importante destacar desde já algumas diferenças entre frameworks OO e g-frameworks. A primeira delas está na natureza da abstração de primeira ordem utilizada pela técnica, enquanto frameworks OO lidam com objetos, g-frameworks lidam com interações. Esta diferença terá impacto sobre o núcleo, pontos de extensão, fluxo de execução, modularização e sobre alternativas de especialização possíveis. Por exemplo, se a definição de pontos de extensão em frameworks OO são classes ou interfaces, g-frameworks definem seus pontos de extensão sobre elementos de leis. Se a instância gerada por um framework OO é uma aplicação, em um g-framework são customizadas leis de interação para instruir mecanismos de governança para verificar a conformidade de agentes com o comportamento esperado para o sistema multiagente; dentre outros detalhes que serão discutidos adiante na tese.

Esta tese pretende contribuir para um melhor projeto de leis de interação onde seja possível reutilizar as partes e customizar a lei com base em um suporte explícito para a flexibilidade de seus elementos. A codificação destas regras em uma linguagem própria que possa ser mapeada para um mecanismo de governança está associada ao projeto de leis de interação. Para este propósito, o XMLaw [Paes et al., 2005] foi aprimorado como linguagem de codificação do comportamento previsto dos agentes. Estes agentes serão governados pelo mediador que interpretará a lei de interação. Este mediador é um mecanismo de governança que será implementado pelo middleware MLaw [Paes et al., 2007]. Discutiremos estas duas abordagens com detalhes mais adiante.

Com o objetivo de facilitar a construção de leis de interação e mecanismos de governança, os desenvolvedores das leis possuem um método de referência que pode ser utilizado para as fases de requisitos, análise e projeto e implementação. Cada fase possui uma técnica auxiliar associada que será descrita ao longo desta tese. Para reutilizar de forma efetiva uma solução é preciso documentar o seu projeto de forma que os usuários potenciais do g-framework possam entender a

proposta e o que devem fazer para utilizá-las nos seus contextos. Guias de referência foram propostos para registrar esta natureza de informação.

1.1.

Problema

O desenvolvimento de leis de interação pode ser uma tarefa complexa. Muitos elementos de leis podem ser necessários, os domínios de observação podem ser complexos e cheios de regras, variações podem ser requeridas devido a especificidades do domínio ou alterações constantes nas regras. O investimento para a produção de um conjunto de soluções a partir do zero (sem reutilização) em um mesmo domínio pode ser proibitivo se não houver um bom projeto de software. Portanto, associar flexibilidade e reusabilidade é importante.

O problema a ser resolvido nesta tese é aprimorar a manutenibilidade (flexibilidade e reutilização) de leis de interação que são utilizadas para instruir como mecanismos de governança devem observar e se comportar frente à troca de mensagens entre agentes. A sistematização do desenvolvimento de mecanismos de governança é uma necessidade em uma área crescente de pesquisa e aplicação em nossa sociedade.

O objetivo de prover flexibilidade corresponde à possibilidade de alterar uma solução. Este conceito é fundamental para qualquer customização. É nossa meta dar apoio a esta finalidade com instrumentos que facilitam o projeto e a implementação de um mecanismo de governança flexível.

1.2.

Limitações de Abordagens Existentes

Existem diferentes modelos para governança de leis de interação em sistemas multiagentes abertos. Associados a estes modelos, diversas implementações apóiam a garantia da conformidade a estas regras. Estes modelos [Castelfranchi et al. 1999; Dignum, 2001; Dignum & Dignum 2001; Esteva 2003; Esteva et al. 2001; Jones & Sergot, 1993; Kollingbaum & Norman, 2003; Martin et al., 1999; Mineau, 2003; Minsky & Ungureanu, 2000; Rodriguez-Aguilar, 2001; Schumacher et Ossowski, 2005; Vazques-Salceda et al. 2005] variam quanto a diferentes granularidades de elementos, diferentes abstrações e diferentes formalismos associados. No entanto, não existem técnicas associadas a estes modelos que apóiem de forma efetiva a produção de leis de interação preparadas para a reutilização. Falta ainda uma orientação sobre como é possível projetar

aplicações com este propósito. Por fim, não existem abordagens que facilitem a reutilização de leis de interação de forma sistemática.

1.3. Solução Proposta

A abordagem de g-frameworks visa promover a reutilização de leis de interação de forma sistemática para a produção de uma família de mecanismos de governança. Ao trabalhar com dois níveis de abstração (geral + específico) separa-se o que há de comum em uma lei de interação em um domínio específico na parte geral, ou núcleo da solução, e associada a esta parte devem ser especificados pontos sujeitos a alteração, que corresponderão à parte específica da solução. Eventualmente, a complexidade associada a este domínio pode ser diminuída para quem estiver instanciando a aplicação. A técnica de g-frameworks foi desenvolvida para este propósito baseada em frameworks OO. No entanto, a natureza do problema é diferente e customizações foram necessárias para adequar essa técnica a natureza de problema em discussão.

1.4. Histórico da Abordagem

Esta pesquisa teve início com a proposição de uma primeira versão da linguagem (XMLaw) [Paes et al., 2004] e de uma infra-estrutura para a governança de sistemas multiagentes abertos (MLaw) [Paes et al., 2007], propostos por um grupo de pesquisadores do Laboratório de Engenharia de Software (LES) da PUC-Rio. Ao longo de três anos de experimentação, tanto a linguagem, quanto o projeto e implementação da infra-estrutura amadureceram. Dentro da linha de amadurecimento da proposta, buscamos entender a possibilidade e ganhos com a aplicação de métodos formais para a identificação de inconsistências estruturais em leis de interação [Carvalho et al., 2006b]. Um primeiro modelo conceitual formalizado foi proposto, mas, no entanto, não foi dado prosseguimento a este trabalho.

Esta tese especificamente, desvinculada do trabalho do grupo, teve início a partir de uma abordagem de análise de riscos para identificar requisitos de fidedignidade para o desenvolvimento de sistemas multiagentes abertos [Carvalho et al., 2004]. Esta análise deu origem ao estudo de cadeias de causas e conseqüências que deveriam ser mitigadas por leis de interação [Carvalho et al. 2005a]. Este modelo documentava os requisitos em uma estrutura de cenários e

explorava basicamente a cadeia de eventos de elementos de leis definida pelo modelo conceitual de XMLaw. Este trabalho foi seminal para a proposta de Casos de Leis, que foi inicialmente aplicada como apoio à customização de estratégias de replicação em sistemas multiagentes abertos [Gatti et al., 2006b].

A partir da análise e estudo de sistemas multiagentes abertos, pode-se perceber que as abordagens existentes não lidavam de forma adequada com o tema “projeto de leis de interação flexíveis e reutilizáveis”. A necessidade prática por esta natureza de técnica ficou mais clara a partir de um estudo inicial de versões do Trading Agent Competition – Supply Chain Management (TAC SCM) [Arunachalan et al. 2004; Collins et al. 2004; Collins et al. 2005; Sadeh et al., 2003] e de abordagens correlacionadas à flexibilização de interação em sistemas multiagente na literatura [Haddadi 1996; Koning & Huget 2000; Kubawara et al. 1995; Singh 1998]. Propusemos uma primeira abordagem para facilitar a customização do mecanismo de governança a partir de leis de interação semi-completas. Neste experimento, evoluímos a noção de pontos de extensão associados à especificação da interação de sistemas multiagentes abertos [Carvalho et al. 2005b]. Este experimento foi muito importante devido ao cuidadoso registro das variações e evoluções de regras de interação dos agentes de software em edições passadas desta competição [Collins et al. 2004; Collins et al. 2005; Sadeh et al., 2003].

Em um primeiro protótipo do TAC SCM, percebemos que era possível sistematizar o desenvolvimento de mecanismos de governança a partir da definição de soluções específicas para domínios de aplicação [Carvalho et al. 2006a]. Foi neste momento que a metáfora de frameworks orientadas a objeto foi estudada e adaptada para o contexto específico de leis de interação. Isto deu origem ao trabalho de frameworks de governança de sistemas multiagentes regulados por leis [Carvalho et al. 2007a], re-batizado de g-frameworks. Esta proposta foi apresentada no Doctoral Mentoring do AAMAS [Carvalho 2006c], onde obteve grande receptividade junto à comunidade de governança e junto a um dos autores do trabalho seminal de electronic-institutions [Esteva 2003], Carlos Sierra, referência em governança de SMAs abertos.

A partir do entendimento de que era necessário um maior apoio por parte da linguagem de especificação de leis (XMLaw), propusemos operadores de refinamento [Carvalho et al. 2006d] como instrumento para aprimorar a forma

como registrávamos elementos de leis abstratos e o seu respectivo processo de especialização.

Em paralelo, porém de forma relacionada à manutenibilidade de leis, a técnica de g-frameworks foi aprimorada e ganhou identidade a partir da observação de um outro nível de evolução, o da própria linguagem de especificação de leis para o propósito de testes de integração [Rodrigues et al. 2005] e replicação de agentes de software [Gatti et al. 2006a; Gatti et al., 2006b]. Nestas abordagens, novos elementos de leis foram propostos. Durante estes experimentos buscamos aplicar e amadurecer a técnica de governança, a modularização dos elementos de leis e distinguir claramente a natureza de nosso trabalho. G-frameworks dá ênfase ao apoio ao projeto e implementação de mecanismos de governança customizáveis a partir de leis preparadas para a reutilização, de forma complementar a evolução do próprio modelo conceitual de XMLaw.

Avaliamos ainda ao longo dos estudos a possibilidade de evolução dinâmica não antecipada [Almeida et al., 2006; Ebraert et al., 2005], e propusemos uma abordagem que foi experimentada para este propósito [Carvalho et al., 2006e]. No entanto, optamos por deixar este tópico fora da abordagem de g-frameworks. A pesquisa nesta linha continua a ser realizada em cooperação com a UFCG.

Quanto aos estudos de caso, o TAC SCM é uma iniciativa acadêmica para explorar a técnica de agentes de software em um domínio específico de cadeia de suprimentos. Uma experimentação mais extensa com a técnica era necessária. Neste sentido começamos uma busca por uma aplicação real que pudesse inspirar um estudo de caso pragmático, complexo e que conseguisse demonstrar a aplicabilidade da técnica a ambientes não controlados, como não era o caso do TAC SCM. O Sistema de Liquidação e Custódia de Títulos Públicos do Banco Central do Brasil (SELIC) [MUS, 2005] foi o cenário escolhido para a aplicação da técnica de g-frameworks. A complexidade, natureza e características do SELIC justificavam a aplicação de g-frameworks para o desenvolvimento de mecanismos de governança customizáveis. O SELIC é, na verdade, um mecanismo de governança de leis de interação que regula a negociação de títulos públicos entre instituições financeiras (agentes autônomos). Este experimento demonstrou a aplicabilidade da técnica de g-frameworks em um cenário do mundo real, com complexidade e tamanho superiores ao TAC SCM. O relato sobre a aplicação da

abordagem ao estudo de caso SELIC foi aceito pelo journal da SBC [Carvalho et al., 2007b]. Este estudo de caso será descrito ao final deste documento.

Ao final do trabalho, complementamos a abordagem de Casos de Leis com guias de referência, para facilitar a documentação e re-aproveitamento das soluções desenvolvidas com g-frameworks.

1.5. Contribuições

Casos de leis são uma técnica de documentação de decisões de análise (racional) derivada a partir da especificação de requisitos para sistemas multiagentes abertos. Esta documentação apresenta o registro de requisitos, que serão mapeados para leis de interação. Estas leis são interpretadas por infra-estruturas capazes de verificar, em tempo de execução, se os requisitos, ou melhor, se as regras são obedecidas pelos agentes. Esta técnica foi proposta, experimentada e adaptada para documentar o raciocínio em prol da generalização de leis de interação, onde uma estrutura de hipóteses e evidências justificaria a preocupação com a flexibilidade e a reutilização em projetos em leis de interação.

Técnicas de Governança de SMAS Abertos foram propostas ao longo da pesquisa. Isto incluiu primeiramente amadurecimentos quanto aos elementos pertencentes ao **modelo conceitual de leis de interação**. Contribuímos também com a proposição da **linguagem de especificação de leis (XMLaw)**, com destaque para os **operadores de refinamento**, que são elementos que possibilitam caracterizar um elemento como abstrato, completar detalhes de elementos definidos como abstratos, e especializam elementos inspirados na técnica de herança de orientação a objeto. Por fim, participamos ativamente da evolução, projeto, melhoria e implementação do **mecanismo de observação e monitoramento de leis de interação (MLaw)** que é capaz de monitorar interações em um sistema multiagente e de interpretar as especificações das leis para verificar se elas estão sendo efetivamente cumpridas. Este mecanismo foi aprimorado ao longo de três anos para a versão corrente e inclui o suporte a interpretação de leis de interação flexíveis e customizáveis.

G-Frameworks prevê a reutilização de design e de código semi-acabados e prontos para usar, referente a leis de interação para a produção de uma família de mecanismos de governança. Esta contribuição corresponde também à proposição das técnicas para análise, projeto e implementação de mecanismos de governança.

Associados a g-frameworks, **pontos de extensão em interação** foram propostos para que em projetos de lei sejam identificados pontos de variação a partir da análise da variabilidade dos elementos de leis propostos. Além disto, merece destaque a proposta de documentação de g-frameworks que também será apresentada nesta tese.

Experimentações com a abordagem de governança foram feitas em dois cenários (um acadêmico e outro real) de aplicação e a técnica de g-frameworks se mostrou adequada para a solução dos problemas em questão.

1.6.

Organização da Tese

A tese está organizada da seguinte forma. O Capítulo 2 aborda a apresentação das técnicas básicas utilizadas por g-frameworks para a governança de sistemas multiagentes abertos. Este capítulo descreve para isto o problema exemplo, TAC SCM; o modelo conceitual e a linguagem de especificação de leis de interação XMLaw; o middleware de governança de sistemas multiagentes MLaw; e uma visão geral da técnica de casos de leis customizada para o contexto de reutilização de leis.

O Capítulo 3 aborda técnicas de apoio à reutilização de leis de interação. Nele descrevemos o que é variabilidade de leis de interação, apresentando os operadores de refinamento em XMLaw, a noção de pontos de extensão, e a técnica de g-frameworks. O Capítulo 4 apresenta o método de desenvolvimento de g-frameworks a partir da apresentação da solução e desenvolvimento do g-framework de cadeias de suprimentos, TAC SCM.

O Capítulo 5 apresenta o estudo de caso do g-framework de liquidação e custódia de títulos públicos (SELIC) baseado em uma aplicação real e complexa. Para apoio a análise dos requisitos e do projeto da variabilidade neste exemplo, desenvolvemos e aplicamos uma técnica de processamento de linguagem natural para identificar contextos contendo os pontos de variação da solução. Descrevemos o projeto e a implementação do núcleo do g-framework e de seus pontos de extensão associados.

O Capítulo 6 apresenta trabalhos relacionados à tese e os compara com a técnica de g-frameworks. Algumas dimensões de avaliação são usadas para facilitar o entendimento das diferenças. Por fim, o Capítulo 7 apresenta as conclusões, benefícios e trabalhos futuros decorrentes desta tese de doutorado.