UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CAMILA DASSO THOMASI

OrIAs: Uma Infraestrutura de Nível Micro-Organizacional Baseada em Artefatos para Sistemas Multiagentes

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Antônio Carlos da Rocha Costa
Orientador

Porto Alegre, março de 2014

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Thomasi, Camila Dasso

OrIAs: Uma Infraestrutura de Nível Micro-Organizacional Baseada em Artefatos para Sistemas Multiagentes / Camila Dasso Thomasi. – Porto Alegre: PPGC da UFRGS, 2014.

120 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2014. Orientador: Antônio Carlos da Rocha Costa.

1. Organização de sistemas multiagentes. 2. Infraestrutura de gerenciamento organizacional. 3. Infraestrutura normativa. 4. Interação organizacional. 5. Relação de dependência. 6. Reputação. 7. Normas. 8. Reorganização. I. Costa, Antônio Carlos da Rocha. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do PPGC: Prof. Luigi Carro
Bibliotecário-chefe do Instituto de Informática: Alexandre Borges Ribeiro

“A medida da confiança é a medida da realização.”

AGRADECIMENTOS

Na conclusão de mais esta etapa especial em minha vida, na qual mais um sonho se torna realidade, é indispensável agradecer a todos aqueles que estiveram ao meu lado no decorrer desta caminhada.

Meu muito obrigado aos meus amados pais Paulo e Lidia, que desde o início quando decidi ingressar no mestrado me apoiaram incondicionalmente e não mediram esforços para que mais esse sonho fosse concretizado. Obrigada por entenderem os motivos de minhas ausências e ao mesmo tempo darem tanta força e amor para que eu pudesse enfrentar os obstáculos desta jornada.

Agradeço ao meu amado namorado Alex, por sempre me compreender, me amar e por ter me dado força e me incentivado no momento mais difícil e crucial durante estes dois anos que estive no mestrado.

Obrigada a todos os mestres que conheci ao longo destes dois anos que contribuíram para meu aperfeiçoamento na área acadêmica. Um agradecimento mais que especial ao meu orientador Rocha que despendeu grande parte do seu tempo para me auxiliar na realização deste trabalho, sempre disposto a ouvir minhas ideias indicando sempre o caminho correto a seguir.

Meu agradecimento aos membros da banca examinadora por terem aceitado o convite e pela avaliação deste trabalho.

Obrigada aos colegas e amigos que tive a oportunidade de conhecer e conviver nestes dois anos. Com vocês mais que a troca de conhecimento pude vivenciar experiências que me tornaram um ser humano melhor.

Meu agradecimento a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa durante todo o período de realização deste mestrado.

Por fim, obrigado a Deus por ter me dado a fé, a força, a coragem e a determinação para realização deste e dos outros sonhos que virão.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 7

LISTA DE FIGURAS 8

LISTA DE TABELAS 10

LISTA DE CÓDIGOS 11

RESUMO 12

ABSTRACT 13

1 INTRODUÇÃO 14
1.1 Motivação 15
1.2 Objetivos 16
1.3 Metodologia 16
1.4 Organização da Dissertação 16

2 REVISÃO BIBLIOGRÁFICA 18
2.1 Organização de Sistemas Multiagentes 18
2.2 Metamodelo de Agentes e Artefatos 22
2.2.1 CArtAgO 24
2.3 Organizações Incorporadas 26
2.4 Relação de Dependência entre os Agentes 28
2.4.1 Mecanismos de Racíocinio Social 31
2.4.2 Coalizões Baseadas em Dependências 32
2.5 Reputação 33
2.6 Normas 35
2.7 Comunicação em Sistemas Multiagentes 37
2.7.1 Teoria dos Atos de Fala 38
2.7.2 KQML 39
2.7.3 FIPA 39
2.7.3.1 FIPA Request Protocol 40
2.7.3.2 FIPA Contract Net Protocol 41

3 ORIAS 43
3.1 Etapa Organizacional 44
3.1.1 Dimensão Estrutural 44
3.1.2 Dimensão Funcional 46

3.1.3 Dimensão Dialógica 49
3.1.4 Dimensão Normativa 50
3.2 Etapa de Modelagem dos Artefatos 50
3.2.1 Artefato GeneralORG 51
3.2.2 Artefato Roles 53
3.2.3 Artefato Group 54
3.2.4 Artefatos de Dependência 55
3.2.5 Artefato Norm 56
3.2.6 Artefatos de Comunicação 57
3.2.7 Artefatos de Reputação 60
3.3 Etapa de Descrição dos Serviços 64
3.4 Etapa de Desenvolvimento da Infraestrutura 75

4 EXPERIMENTO REALIZADO 82
4.1 Estudo de Caso 82
4.2 Agentes Utilizados na Simulação 83
4.2.1 Descrição dos Agentes 83
4.2.2 Implementação dos Agentes 84
4.3 Simulação Realizada 86

5 COMPARATIVO COM TRABALHOS RELACIONADOS 93
5.1 ORA4MAS 93
5.2 JaCaMo 95
5.3 OrIAs X ORA4MAS X JaCaMo 97

6 CONCLUSÕES E TRABALHOS FUTUROS 100

REFERÊNCIAS 102

APÊNDICE A CÓDIGOS DOS AGENTES 106

APÊNDICE B ESPECIFICAÇÃO ORGANIZACIONAL DO EXPERIMENTO REALIZADO 111

APÊNDICE C ESQUEMA DA ESPECIFICAÇÃO ORGANIZACIONAL . . 115
C.1 Descrição da Dimensão Estrutural 115
C.2 Descrição da Dimensão Funcional 117
C.3 Descrição da Dimensão Dialógica 118
C.4 Descrição da Dimensão Normativa 119

LISTA DE ABREVIATURAS E SIGLAS

ACL	Agent Communication Language
A&A	Agents & Artifacts
AGR	Agent/Group/Role
BDI	Belief Desire Intentions
CArtAgO	Common Artifact infrastructure for Agent Open environment
EAs	Environmental Artifacts
EOA	Embodied Organizational Artifacts
FIPA	Foundation for Intelligent Physical Agents
KQML	Knowledge Query and Manipulation Language
NOPL	Normative Organisation Programming Language
OAs	Organizational Artifacts
OMI	Organizational Management Infrastructure
OML	Organisation Modeling Language
OperA	Organizations per Agents
ORA4MAS	Organizational Artifacts for Multi-Agent Systems
OrIAs	Organizational Infrastructure for Agents
RBAC	Role-Based Access Control
SMA	Sistema Multiagente
STEAM	Shell for TEAMwork
TAEMS	Task Analysis, Environmental Modeling and Simulation
UML	Unified Modeling Language
XML	eXtensible Markup Language

LISTA DE FIGURAS

Figura 2.1: Representação das três abstrações que caracterizam o metamodelo A&A. 23
Figura 2.2: Localização de um *work environment* em um sistema multiagente extraído de Ricci et al.(2009). 25
Figura 2.3: Representação de um artefato, figura adaptada de Ricci, Piunti e Viroli (2011). 25
Figura 2.4: Classificação das dependências quanto a direção. 29
Figura 2.5: Classificação das dependências em relação ao número de parceiros. . 30
Figura 2.6: Processo de formação e transmissão da reputação, figura adaptada de Conte e Paolucci (2002). 34
Figura 2.7: Protocolo *Request* conforme definido pela FIPA, figura extraída de FIPA (2006) 40
Figura 2.8: Protocolo *Contract Net* conforme definido pela FIPA, figura extraída de FIPA(2002) 41

Figura 3.1: Etapas de criação de infraestrutura. 44
Figura 3.2: Organização geral. 45
Figura 3.3: Objetivo individual na dimensão funcional. 47
Figura 3.4: Objetivo coletivo na dimensão funcional. 47
Figura 3.5: Artefatos presentes na infraestrutura. 51
Figura 3.6: Representação do artefato GeneralOrg. 52
Figura 3.7: Representação do artefato Roles. 53
Figura 3.8: Representação do artefato Groups. 54
Figura 3.9: Representação do artefato RoleDEP. 55
Figura 3.10: Representação do artefato GroupDEP. 56
Figura 3.11: Representação do artefato Norm. 56
Figura 3.12: Propriedade observável `norm`. 56
Figura 3.13: Diagrama que permite ilustrar como novos artefatos de comunicação podem ser criados , adaptado de Rodrigues, Rocha Costa e Dimuro (2013). 58
Figura 3.14: Digrama de classe referente aos protocolos *Request* e *ContractNet*. . . 59
Figura 3.15: Representação dos artefatos de comunicação acessados diretamente pelos agentes. 60
Figura 3.16: Representação do artefato NormREP. 61
Figura 3.17: Representação do artefato DelegationREP. 63
Figura 3.18: Processo de adoção um papel por um agente na organização. 65

Figura 3.19: Diagrama de atividades do ponto de vista do artefato quando um agente adota um papel. 65
Figura 3.20: Diagrama de atividades correspondente aos passos adicionados a um plano quando detectadas etapas das quais o agente é dependente. . . . 66
Figura 3.21: Diagrama de atividades do ponto de vista do agente quando um sinal referente a um novo plano é recebido. 67
Figura 3.22: Diagrama de atividades do ponto de vista do agente quando um sinal referente a um objetivo inicial (*start goal*) é recebido. 67
Figura 3.23: Processo de atualização das reputações de um agente quando o mesmo recebe a delegação de uma tarefa. 68
Figura 3.24: Processo que ocorre quando um agente executa a operação `leaveRole` 68
Figura 3.25: Diagrama de atividades correspondente a execução da operação `leaveRole`. 69
Figura 3.26: Diagrama de atividades relativo ao agente ao receber o sinal `removePlansOfRole`. 69
Figura 3.27: Processo de criação de um grupo por um agente na organização. . . . 70
Figura 3.28: Diagrama de atividades sob o ponto de vista do artefato para formação de um grupo. 70
Figura 3.29: Gerente buscando agentes para formação do grupo através do protocolo *Contract Net*. 70
Figura 3.30: Diagrama de atividades sob o ponto do agente ao receber o sinal sobre os membros do grupo. 71
Figura 3.31: Diagrama de atividades referente a execução do plano coletivo. . . . 72
Figura 3.32: Processo para finalizar um grupo na organização. 73
Figura 3.33: Diagrama de atividade relativo a finalização de um grupo na organização. 73
Figura 3.34: Diagrama de atividade do ponto de vista do agente quando recebe um sinal relativo a finalização de um grupo. 73
Figura 3.35: Criação de normas na infraestrutura. 74
Figura 3.36: Controle realizado pelo agente organizacional em relação do cumprimento/descumprimento das normas. 74
Figura 3.37: Diagrama de atividades correspondente ao controle realizado pelo pelo agente organizacional em relação cumprimento/descumprimento das normas. 75
Figura 3.38: Representação do processo de reorganização. 75
Figura 3.39: Diagrama de atividades correspondente ao processo de reorganização. 76
Figura 3.40: Diagrama de classe referente as dimensões estrutural e funcional. . . 79
Figura 3.41: Diagrama de classe referente a dimensões dialógica. 80
Figura 3.42: Diagrama de classe referente a dimensões normativa. 80
Figura 3.43: Arquitetura lógica da infraestrutura. 80

Figura 4.1: Representação dos artefatos de aplicação utilizados. 87
Figura 4.2: Passos do plano coletivo `realizar eleição`. 88
Figura 4.3: Passos do plano coletivo `contabilizar processo`. 89

LISTA DE TABELAS

Tabela 3.1: Estrutura utilizada para armazenar as dependências em relação aos papéis da organização. 48
Tabela 3.2: Estrutura utilizada para armazenar as dependências em relação aos grupos da organização. 48
Tabela 3.3: Estrutura utilizada para armazenar a reputação quanto ao cumprimento de normas dos agentes. 61
Tabela 3.4: Estrutura utilizada para armazenar a reputação de delegação dos agentes. 62

Tabela 4.1: Operações restritas em cada artefato e os respectivos papéis que podem acessar as operações. 89
Tabela 4.2: Operações restritas em cada artefato e os respectivos papéis que podem acessar as operações. 90

LISTA DE CÓDIGOS

2.1 Exemplo de uma mensagem KQML, código extraído de Wooldridge (2009). 39
3.1 Trecho de código da descrição organizacional referente a dimensão estrutural. 77
3.2 Trecho de código da descrição organizacional referente a dimensão funcional. 77
3.3 Trecho de código da descrição organizacional referente a dimensão dialógica. 78
3.4 Trecho de código da descrição organizacional referente a dimensão normativa. 79
4.1 Padrão utilizado no Jason para representar um objetivo de manutenção, código extraído de Bordini, Hübner e Wooldridge (2007). 85
4.2 Plano original referente ao objetivo `ganhar eleição` do agente político. 90
4.3 Plano modificado pelo artefato referente ao objetivo `ganhar eleição` do agente político. 90
4.4 Plano de contingência para o objetivo `ganhar eleição`. 90
4.5 Plano referente ao objetivo de manutenção do agente político. 91
4.6 Plano referente a adição de crença. 91
4.7 Plano referente ao objetivo `organizar eleição`. 91
A.1 Trecho de código do agente organizacional referente a criação dos artefatos da infraestutura. 106
A.2 Trecho de código que o agentes que participam da organização devem incluir para o tratamento dos eventos advindos dos artefatos da infraestrutura. 106
A.3 Código de um agente que deseja participar da organização. 110
B.1 Especificação organizacional do estudo de caso *processo político de lobby através de contribuições para campanhas eleitorais*. 111
B.2 Especificação organizacional utilizada para o processo de reorganização do estudo de caso *processo político de lobby através de contribuições para campanhas eleitorais*. 113
C.1 Descrição do papel político e do grupo processo eleitoral na especificação organizacional com suas respectivas características. 116
C.2 Descrição dos planos para os objetivos do papel político e do plano para o objetivo do grupo processo eleitoral na especificação organizacional. . . 117
C.3 Descrição dos protocolos utilizados para delegar as tarefas quando os agentes são dependentes e para formação de grupos. 118
C.4 Descrição das normas para criação do grupo processo eleitoral e para utilização do artefato GeneralOrg. 120

RESUMO

Neste trabalho é apresentada uma infraestrutura de nível micro-organizacional para sistemas multiagentes. A infraestrutura permite definir e gerenciar papéis e grupos de papéis organizacionais, que caracterizam o nível micro-organizacional das sociedades de agentes, em termos de quatro dimensões organizacionais: estrutural, funcional, dialógica e normativa. As principais características micro-organizacionais suportadas pela infraestrutura são: definição de papéis em termos de objetivos de realização (*achievement goals*) e objetivos de manutenção (*maintenance goals*), e seus respectivos planos; definição de normas organizacionais e um mecanismo para imposição das normas (*norm enforcement*); relações de dependências organizacionais entre papéis e grupos de papéis; mecanismo de reputação para auxiliar os agentes na busca de parceiros para seus objetivos individuais e selecionar parceiros para atingir metas coletivas; processo de reorganização não institucionalizado realizado por iniciativa dos agentes. A infraestrutura é baseada no metamodelo de agentes e artefatos e no conceito de organizações incorporadas (*embodied organizations*). Como resultado deste trabalho é disponibilizada uma infraestrutura de gerenciamento organizacional, independente de modelo organizacional, desenvolvida utilizando a plataforma CArtAgO que permite tornar a organização acessível aos agentes.

Palavras-chave: Organização de sistemas multiagentes, Infraestrutura de gerenciamento organizacional, Infraestrutura normativa, Interação organizacional, Relação de dependência, Reputação, Normas, Reorganização.

OrIAs: An Artifact-Based Micro-Organizational Level Infrastruture for Multiagent Systems

ABSTRACT

In this work, we present an artifact-based micro-organizational level infrastructure for multi-agent systems. It provides a way to define and manage the organizational roles and groups of roles that characterize the micro-organizational level of agent societies, in terms of the four organizational dimensions: structural, functional, dialogical and normative. The main micro-organizational features supported by the infrastructure are: definition of roles in terms of achievement and maintenance goals, and their corresponding plans; organizational norms and corresponding sanction-based mechanism of norm enforcement; organizational protocols to support the achievement of role goals; relations of organizational dependences between roles and groups of roles; reputation mechanism to help agents to find partners for their individual goals and to select partners to achieve collective goals; reorganization process not institutionalized performed by the initiative of agents. The infrastructure is based on the A & A meta-model and in the concept of embodied organizations. As a result of this work, we provide an organizational management infrastructure, independent of organizational model, developed using the CArtAgO platform that makes the organization accessible to agents.

Keywords: Multi-agent systems organization, Organizational Management Infrastructure, Normative infrastructure, Organizational interaction, Dependence relations, Reputation, Norms, Reorganization .

1 INTRODUÇÃO

Uma das formas de conciliar a autonomia dos agentes e os requisitos de funcionamento é atribuir uma organização ao sistema. Dessa forma o comportamento dos agentes pode ser regulado por restrições que podem ser definidas a partir de uma estrutura formal permitindo ao sistema atingir um determinado propósito.

Para incorporar uma estrutura organizacional em um sistema multiagente é necessário definir papéis e grupos que devem estar presentes na organização, suas características e seus relacionamentos. Essas características definem o nível micro-organizacional de um sistema multiagente (COSTA; DIMURO, 2009).

Frequentemente, é conveniente definir as operações do nível micro-organizacional em termos de padrões de interação, mecanismos de monitoramento do comportamento dos agentes e mecanismos que auxiliem os agentes na correta escolha de seus parceiros organizacionais para melhor atingirem seus objetivos. Dessa forma é conveniente não apenas especificar formalmente a organização como apoiar de forma dinâmica os agentes na utilização dessa organização. Uma forma de disponibilizar a organização para os agentes é manter a representação das características organizacionais através de uma infraestrutura de gerenciamento organizacional (*organizational management infrastructure* (OMI)).

Este trabalho apresenta OrIAs (***Or**ganizational **I**nfrastructure for **A**gents*) uma infraestrutura organizacional baseada em artefatos que permite representar o nível micro-organizacional de sistemas multiagentes.

A infraestrutura OrIAs suporta quatro dimensões organizacionais: estrutural, funcional, dialógica e normativa. O inter-relacionamento entre essas dimensões permite restringir o comportamento dos agentes, estabelecer padrões de interação, distribuir tarefas e coordenar as atividades dos agentes. A infraestrutura permite a definição tanto objetivos de realização (*achievement goals*) como objetivos de manutenção (*maintenance goals*). OrIAs fornece suporte para o raciocínio social dos agentes em termos de relações de dependência entre os papéis organizacionais que os agentes assumem e entre grupos. A infraestrutura utiliza um mecanismo de reputação para auxiliar os agentes tanto na busca de parceiros para o alcance dos objetivos individuais e coletivos como também para auxiliar no processo de aplicação das normas. A infraestrutura permite que normas sejam criadas, atualizadas e excluídas dinamicamente. Além disso, uma infraestrutura de comunicação é integrada a infraestrutura OrIAs de forma a facilitar a interação entre os agentes. OrIAs permite ainda modificar a organização a partir de um processo de reorganização não institucionalizado realizado por iniciativa dos agentes. A infraestrutura permite também que

artefatos de ambiente sejam utilizados como intermediários entre organização e ambiente.

OrIAs é baseada no metamodelo de agentes e artefatos A&A e no conceito de organizações incorporadas (*embodied organizations*). A infraestrutura é implementada utilizando a plataforma CArtAgO [1].

1.1 Motivação

Recentes trabalhos apresentados na literatura, como ORA4MAS (HÜBNER et al., 2010) e JaCaMo (BOISSIER et al., 2011), têm mostrado as vantagens da utilização organizações incorporadas ao ambiente para o desenvolvimento de infraestruturas de gerenciamento organizacional.

Organização incorporada (*Embodied organizations*) (PIUNTI et al., 2009) (PIUNTI et al., 2010) é uma abordagem que permite integrar organização, agentes e ambientes utilizando o metamodelo de agentes e artefatos (A&A). Nessa abordagem o ambiente é visto não apenas como uma fonte de percepções ou alvo das ações para os agentes, mas também capaz de encapsular funcionalidades e serviços organizacionais.

Tanto ORA4MAS como JaCaMo incorporam artefatos organizacionais ao ambiente específicos para o modelo organizacional Moise (HÜBNER; SICHMAN; BOISSIER, 2002). Apesar de essas infraestruturas utilizarem normas para regular o comportamento dos agentes, essas normas estão fortemente atreladas às dimensões normativa do modelo Moise. Além disso, não existe uma dinâmica que permite alterar e remover normas existentes assim como criar novas normas. Apesar do conceito de organização incorporada permitir que artefatos de ambiente sejam considerados como mediadores entre a organização e o ambiente, isso não chega a ser explorado nesses trabalhos.

Como o projeto dos artefatos de ambos os trabalhos estão atrelados ao modelo Moise e este modelo possui três dimensões organizacionais (estrutural, funcional e normativa), aspectos relacionados a dimensão dialógica não são abordados nos trabalhos. Além disso, não são abordados aspectos relacionados a dependências que podem surgir entre agentes e grupos em decorrência de objetivos os quais os agentes não são autossuficientes. Nem são tratados outros tipos de objetivos além de objetivos de realização (*achievement goals*).

No trabalho desenvolvido por Hübner, Boissier e Vercouter (2008), os autores propõe integrar aos artefatos organizacionais do ORA4MAS um artefato relacionado à reputação. Contudo, esse artefato seria responsável por publicar as avaliações das performances dos agentes em relação ao seu comportamento dentro da organização, para que a partir disso a reputação pudesse ser construída. Além disso, o trabalho não trata aspectos de como a reputação influência no processo decisório dos agentes.

[1] http://cartago.sourceforge.net/

1.2 Objetivos

O objetivo deste trabalho é propor uma infraestrutura de gerenciamento micro- organizacional baseada no conceito de organizações incorporadas que permita suportar as quatro dimensões organizacionais (estrutural, funcional, dialógica e normativa), cujo projeto dos artefatos seja mais genérico, ou seja, que não estejam atrelados a nenhum modelo organizacional específico, e que ao mesmo tempo possa suprir algumas lacunas presentes nas abordagens propostas anteriormente.

1.3 Metodologia

Para realização deste trabalho, foram estudadas primeiramente as organizações de sistemas multiagentes, onde foram identificadas as características principais, dimensões, classificação, além de serem estudados alguns modelos organizacionais existentes. Na sequência foram estudadas as infraestruturas organizacionais que utilizam o conceito de organizações incorporadas presentes na literatura sendo identificadas suas principais características e problemas em aberto que poderiam ser trabalhados. Além disso, foram propostos novos elementos a serem agregados na infraestrutura de gerenciamento organizacional que permitem melhor conciliar a autonomia dos agentes e suprir suas limitações individuais.

Em função disso foram estudados os conceitos referentes a esses novos elementos agregados. Foi estudada a teoria relacionada à relação de dependência entre os agentes assim como os mecanismos de raciocínio social capazes de representar essas relações. Também foram estudados os conceitos relativos à reputação e normas. O próximo passo foi identificar como essas teorias poderiam ser modeladas a partir de artefatos organizacionais e como integrar uma dimensão dialógica a infraestrutura proposta.

Após o estudo da parte teórica, foi feita a modelagem dos artefatos segundo o metamodelo A&A e a infraestrutura foi implementada utilizando a plataforma CArtAgO.

1.4 Organização da Dissertação

A dissertação está organizada em seis capítulos:

- **Capítulo 1:** apresenta uma introdução do trabalho proposto, assim como a motivação e objetivo principal do mesmo.
- **Capítulo 2:** apresenta uma revisão bibliográfica das teorias utilizadas para proposta da infraestrutura OrIAs. São apresentadas as características: de uma organização em sistemas multiagentes, o metamodelo de Agentes e Artefatos (A&A) e a plataforma CArtAgO que implementa este modelo, a definição e características das organizações incorporadas (*Embodied Organizations*), os conceitos de dependência e poder social entre os agentes, a definição de reputação utilizada neste trabalho, as características da utilização de normas em sistemas multiagentes e por fim, a comunicação em sistemas multiagentes.

- **Capítulo 3:** apresenta a infraestrutura organizacional proposta denominada OrIAs (*Organizational Infrastructure for Agents*).
- **Capítulo 4:** mostra como a infraestrutura proposta apresentada no Capítulo 3 é utilizada pelos agentes na simulação de um processo político de lobby. Este estudo de caso foi escolhido por ser um exemplo que permite testar todas as funcionalidades providas pela infraestrutura como relações de dependência, reputação, normas, reorganização, dentre outros.
- **Capítulo 5:** compara as semelhanças e diferenças existentes entre a abordagem ORA4MAS e a plataforma JaCaMo com a infraestrutura proposta OrIAs.
- **Capítulo 6:** apresenta as conclusões e trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma revisão das teorias utilizadas para proposta da infraestrutura OrIAs. Na Seção 2.1 são apresentadas as características de uma organização em sistemas multiagentes. Na Seção 2.2 é apresentado o metamodelo de Agentes e Artefatos (A&A) e a plataforma CArtAgO que implementa este modelo. Na Seção 2.3 é apresentada a definição e as características das organizações incorporadas (*Embodied Organizations*). Na Seção 2.4 são apresentados os conceitos de dependência e poder social entre os agentes, mecanismos de raciocínio social e coalizões baseadas em dependência. Na Seção 2.5 é mostrada a definição de reputação utilizada neste trabalho. Na Seção 2.6 são apresentadas as características da utilização de normas em sistemas multiagentes. Por fim, a Seção 2.7 apresenta a comunicação em sistemas multiagentes.

2.1 Organização de Sistemas Multiagentes

Em um sistema multiagente, os agentes são capazes de agir de forma autônoma para o alcance de seus objetivos possuindo capacidade de decisão própria e interesses distintos. Uma das formas de conciliar a autonomia dos agentes e suprir suas limitações individuais é atribuir uma organização ao sistema. Dessa forma é possível coordenar os agentes de forma a obter um comportamento desejado.

Segundo Ferber, Gutknecht e Michel (2004) não existe uma única definição de organização aceita. Contudo existem algumas características principais que são compartilhadas. Uma organização é constituída de agentes. Esses agentes apresentam um comportamento que está funcionalmente relacionado à atividade global da organização. A organização pode ser dividida em partições, denominadas grupos, que podem se sobrepor. Outra característica importante em uma organização é o conceito de papel (*role*).

Um papel refere-se à descrição de um comportamento abstrato de um agente. Nessa descrição estão as responsabilidades associadas ao papel, as restrições e os padrões de interação que um agente deve seguir quando adota esse papel.

Uma organização, além de conciliar a autonomia dos agentes, existe também para suprir as limitações individuais dos mesmos. Essas limitações podem ser cognitivas, físicas, temporais ou institucionais (CARLEY; GASSER, 1999).

A limitação cognitiva está relacionada à racionalidade limitada dos agentes para realizar determinadas tarefas, sendo necessária a união com outros agentes para que possam

alcançar um melhor desempenho. A limitação física está relacionada à necessidade dos agentes em se coordenarem para realizar determinadas ações em detrimento da distribuição dos recursos no ambiente. A limitação temporal está relacionada à necessidade do agente se unir a outros agentes para realização de um objetivo de longo prazo, em função do seu tempo de vida no sistema que pode ser inferior a conclusão deste objetivo. A limitação institucional está relacionada à restrição política ou legal de um agente individual que precisa atingir um status organizacional com intuído de atingir determinados objetivos.

Neste trabalho uma organização é definida como um conjunto de restrições adotadas pelos agentes para que suas atividades sigam os padrões organizacionais. Essas restrições acontecem a partir do momento que um agente decide participar da organização ao assumir determinado papel. Assumir um papel requer que o agente atinja determinados objetivos, sejam eles individuais ou coletivos. Os objetivos devem ser atingidos cumprindo determinadas normas estabelecidas na sociedade e a interação entre os agentes deve seguir os protocolos estabelecidos na organização.

Na adoção de um papel, os agentes devem alcançar os objetivos referentes ao papel assumido. Na organização, definida nesse trabalho, os objetivos podem ser de dois tipos: objetivos de realização (*achievement goals*) e objetivos de manutenção (*maintenance goal*).

Objetivos de realização são objetivos adotados pelos agentes para alcançar um estado particular, ou seja, um estado que o agente deseja que se torne verdadeiro. Objetivos de manutenção referem-se a um estado de mundo que o agente precisa manter, ou seja, um estado que seja mantido verdadeiro indefinidamente. Um exemplo de um objetivo de manutenção seria, por exemplo, garantir que a temperatura de um cômodo esteja abaixo de determinada temperatura. Ao contrário dos objetivos de realização, objetivos de manutenção têm vida longa e não podem ser abandonados após serem alcançados com sucesso. O sucesso de um objetivo de manutenção está em continuamente satisfazer a condição que se deseja manter (DUFF; THANGARAJAH; HARLAND, 2012).

Objetivos de manutenção podem ser utilizados de duas formas. A primeira denominada reativa, espera até determinada condição se tornar falsa antes de tomar alguma medida para restaurar esse estado. A segunda denominada proativa antecipa quando uma condição de manutenção será violada e age apropriadamente para evitar essa violação (DUFF; THANGARAJAH; HARLAND, 2012).

Além disso, de acordo com Ferber, Gutknecht e Michel (2004), uma organização possui dois aspectos: aspecto estrutural (*structural aspect*) e aspecto dinâmico (*dynamic aspect*). O aspecto estrutural, considerado estático, é dividido em duas partes: estrutura de particionamento e estrutura de papéis. A estrutura de particionamento indica como agentes são reunidos em grupos e como grupos estão relacionados. A estrutura de papéis é definida para cada grupo, por um conjunto de papéis e seus relacionamentos. O aspecto estrutural define ainda o conjunto de restrições que os agentes devem satisfazer ao assumir determinado papel. O aspecto dinâmico está relacionado aos padrões de interação definidos para os papéis (criação de grupos, controle de permissões e obrigações, dentre outros).

As organizações podem ainda serem classificadas em: formais e emergentes. Nas organizações formais, as interações entre os agentes são impostas e a situação de interação é pré-estabelecida resultando em uma colaboração simples ou coordenada. Nas organizações emergentes, são os próprios agentes que dinamicamente formam a organização para resolver um determinado problema. Inicialmente esses agentes não tem um objetivo comum a ser atingido. A interação entre os mesmos surge em decorrência da tentativa de alcançar algum objetivo. O termo coalizão é utilizado para caracterizar as organizações emergentes (SICHMAN, 2003) (DIGNUM; DIGNUM; SONENBERG, 2004).

Para que um sistema multiagente possa adotar uma organização formal é necessário que a mesma seja estruturada, especificada e então disponibilizada aos agentes. Isso requer respectivamente um modelo organizacional, uma especificação da organização e uma infraestrutura organizacional.

Um modelo organizacional é uma técnica de modelagem para representar as restrições que modelam as ações e interações de um sistema multiagente (COUTINHO et al., 2005). O foco dessa modelagem é prescrever padrões de atividades conjuntas que irão caracterizar uma dada organização de agentes (COUTINHO, 2009) (ARGENTE et al., 2013).

Alguns modelos organizacionais estão associados a arquiteturas de softwares genéricas, denominadas infraestruturas de gerenciamento organizacional (*Organisation Management Infrastructure* (OMI)), capazes de gerenciar organizações de agentes a partir de uma especificação organizacional. A especificação organizacional permite descrever o modelo de acordo com uma sintaxe usando um formato específico que pode ser manipulado pela infraestrutura. A OMI recebe uma especificação da organização como entrada, interpreta a mesma e provem aos agentes uma organização de acordo com a especificação fornecida (COUTINHO, 2009) (ARGENTE et al., 2013).

Os modelos organizacionais podem ser ainda caracterizados quanto as suas dimensões de modelagem. De acordo com Coutinho et al. (2005) os conceitos organizacionais são distribuídos ao longo de quatro dimensões: estrutural, funcional, dialógica e normativa.

A dimensão estrutural está ligada a especificação de papéis, grupos e relacionamentos entre os papéis. A dimensão funcional se caracteriza pela especificação de metas (individuais ou coletivas) e decomposição de metas que estão ligadas a satisfação do propósito da organização de agentes. A dimensão dialógica é caracterizada pela especificação e estruturas de interação direta entre papéis por troca de mensagens tendo em vista a realização de objetivos organizacionais. A dimensão normativa define normas que inter-relacionam e regulamentam os elementos funcionais, estruturais e dialógicos.

Alguns modelos organizacionais, presentes na literatura, são definidos em termos algumas ou todas as dimensões organizacionais citadas. Para exemplificar são mostrados alguns desses modelos organizacionais e quais dimensões caracterizam os mesmos.

- **Dimensão estrutural:** AGR (*Agent/Group/Role*) (FERBER; GUTKNECHT; MICHEL, 2004) é uma evolução do modelo AALAADIN (FERBER; GUTKNECHT, 1998) utilizado para descrever organizações de sistemas multiagentes. Este modelo, que define a dimensão estrutural de uma organização, é baseado em três principais

conceitos: agentes, grupos e papéis. Um agente é definido como uma entidade ativa e comunicante que assume papéis dentro de grupos. Um grupo pode ser definido como agregações de agentes compartilhando características comuns. Um papel é uma representação abstrata de uma função, serviço ou identificação do agente dentro de um grupo.

- **Dimensão funcional:** TAEMS ((*Task Analysis, Environmental Modeling and Simulation*) (DECKER, 1996) é uma linguagem de modelagem para descrever estruturas de tarefas de agentes. TAEMS permite a representação explicita e quantitativa do inter-relacionamento de tarefas. Uma estrutura de tarefas em TAEMS é uma árvore de decomposição de tarefas. Os nodos mais altos são tarefas que representam os objetivos que o agente tentará alcançar. Tarefas podem ser decompostas em sub-objetivos. Uma tarefa individual que não possui subtarefas é denominada método. Pode existir mais de um método para realizar uma tarefa, e cada método possui uma quantidade de tempo e produz um resultado com alguma qualidade.

- **Dimensão estrutural e funcional:** STEAM (Shell for TEAMwork) (TAMBE, 1997) é um modelo, que possui as dimensões estrutural e funcional, para o desenvolvimento de times no qual a organização é concebida como um time de agentes. Este modelo é baseado em na teoria das intenções conjuntas (*joint intentions theory*). O modelo possui duas hierarquias ligadas através de papéis: *Team organization hierarchy and roles* e *Team activity hierarchy*. Essas hierarquias representam a estrutura e o funcionamento de um time. A organização neste modelo é representada em termos de times, sub-times, papéis, objetivos e planos compartilhados. Os agentes que entram em um time assumem papéis e devem seguir determinados planos.

- **Dimensões estrutural, funcional e normativa:** MOISE+ (HÜBNER; SICHMAN; BOISSIER, 2002) é uma extensão do modelo MOISE (HANNOUN et al., 2000), assim como seu antecessor é composto por duas noções principais: especificação organizacional (estática) e entidade organizacional (criação de entidades que seguem a especificação organizacional). Neste modelo a especificação organizacional é dividida nas dimensões: estrutural, funcional e deôntica. A dimensão estrutural especifica os papéis, grupos e links de comunicação da organização. A dimensão funcional é baseada no conceito de missões e planos globais. Essa dimensão especifica como os objetivos globais podem ser atingidos a partir da decomposição em planos, que agrupam um conjunto de missões que são atribuídas aos agentes. A decomposição de planos globais resulta em uma árvore de decomposição de metas (*goal-tree*) denominada esquema (*scheme*) onde os objetivos das folhas da árvore podem ser alcançados pelos agentes. As especificações funcionais e estruturais são definidas independentemente, contudo o relacionamento entre elas se dá através da especificação deôntica que especifica no nível individual as permissões e obrigações de um papel sobre uma missão.

- **Dimensões estrutural, funcional, normativa e dialógica:** OperA (*Organizations per Agents*) é um modelo organizacional que possui as quatro dimensões organizacionais. O modelo é formado por quatro estruturas conceituais: estrutura social, estrutura de interação, estrutura de normas e estrutura de comunicação. A estrutura social consiste em uma lista de definição de papéis e grupos e um grafo de dependência para assumir um papel. Neste modelo um papel pode ser refinado em

sub-papéis que juntos realizam os objetivos de um dado papel. O refinamento de um papel define as dependências entre os sub-papéis. As possíveis ações de um papel determinam as interações com os outros agentes, dessa forma a estrutura de interação é responsável por estabelecer os padrões de interações que determinam os possíveis diálogos entre os agentes. A estrutura normativa define como os indivíduos devem se comportar. Normas são especificadas em logica deôntica e podem estar relacionadas a papéis, cenas ou transições entre cenas. Na estrutura de comunicação existem mecanismos que incluem tanto representação do domínio de conhecimento quanto protocolos de comunicação. A especificação do conteúdo de comunicação é feito usando ontologias e linguagens de comunicação de agentes.

Apesar da atribuição de uma organização ao sistema permitir especificar como os agentes devem trabalhar em conjunto, auxiliando na coordenação eficaz em um sistema multiagente, muitas vezes é necessário modificar essa organização, seja para otimizar seu funcionamento, seja para adaptações devido a mudanças no ambiente. Esse processo de modificação é denominado reorganização. A reorganização em um sistema multiagente define um processo de modificação seja no aspecto estrutural ou no aspecto dinâmico da organização (DIGNUM; DIGNUM; SONENBERG, 2004) (HÜBNER, 2003).

Segundo Hübner (2003) o processo de reorganização pode ocorrer de três formas distintas: pré-definida, controlada e emergente. Na forma de reorganização pré-definida, o processo de reorganização já está previsto no modelo organizacional, ou seja, modificações a serem realizadas já estão programadas. Na forma de reorganização controlada, o processo de reorganização é conhecido, contudo não se sabe quando o processo ocorrerá. Nessa forma o processo de reorganização é dividido nas seguintes etapas: monitoração, projeto, seleção e implementação. Cada etapa pode ser executada de modo endógeno (agente(s) do sistema realiza(m) a etapa) ou exógeno (usuário do SMA realiza a etapa). Na forma de reorganização emergente, não existe um controle do sistema sobre o processo de reorganização. Esse processo pode acontecer por iniciativa de um agente segundo seus próprios critérios e métodos.

2.2 Metamodelo de Agentes e Artefatos

Esta seção descreve as principais características do metamodelo A&A (*Agents & Artifacts*) sendo baseada nos trabalhos desenvolvidos por Omicini, Ricci e Viroli (2008) e Piunti e Ricci (2009).

A&A é um metamodelo caracterizado em termos de três abstrações: agentes, artefatos e *workspaces*, Figura 2.1. Segundo este modelo um sistema multiagente é projetado dentro de um ambiente comum onde agentes não se comunicam apenas através de uma linguagem de comunicação de alto nível, mas também ao utilizarem diferentes tipos de artefatos organizados no ambiente de trabalho. O comportamento desses sistemas resulta de uma interação de entidades autônomas (agentes) e entidades reativas e funcionais (artefatos).

Neste modelo, agentes representam os componentes proativos do sistema tendo como principal característica a autonomia. Os agentes autônomos encapsulam o controle e autogovernam o curso de suas próprias ações não possuindo interfaces para que possam ser

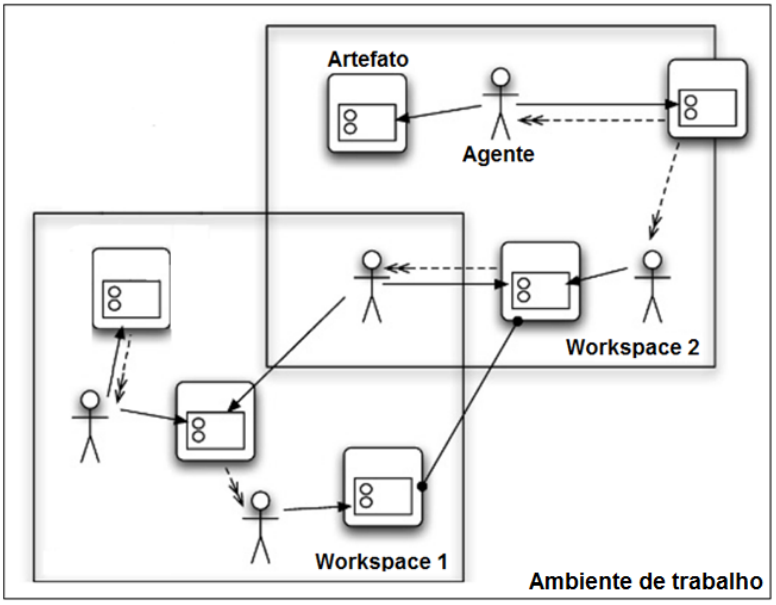


Figura 2.1: Representação das três abstrações que caracterizam o metamodelo A&A.

invocados e controlados.

Os artefatos representam componentes passivos do sistema construídos para determinada função. Essas entidades não possuem autonomia e são reativas em termos de controle. Artefatos podem ser considerados como ferramentas nas mãos dos agentes que são capazes de manipular, compartilhar e utilizar essas entidades para auxiliar e facilitar na realização de suas tarefas.

O comportamento dos artefatos emerge apenas quando usados pelos agentes. Assim para que possam ser utilizados, artefatos devem disponibilizar operações (*operations*) que modificam o estado do artefato fazendo com que o mesmo produza determinados efeitos no ambiente. Um artefato possui uma interface que contém uma coleção de operações que são disponibilizadas para serem utilizadas pelos agentes.

Artefatos exercem um importante papel na forma como agentes cognitivos executam suas tarefas e alcançam seus objetivos. Um agente com habilidades cognitivas pode avaliar artefatos disponíveis, selecionar o artefato que se adapta a suas necessidades utilizando o mesmo para alcançar seus objetivos de forma mais eficiente e eficaz.

Os *workspaces* são containers utilizados para estruturar e organizar o conjunto de agentes e artefatos definindo assim a topologia do ambiente. Esses ambientes de trabalho permitem que sistemas multiagentes complexos possam ser organizados como de áreas de trabalho distribuídas entre vários nós da rede sendo permitido ainda que agentes pertençam a múltiplos *workspaces* ao mesmo tempo.

Um sistema multiagente baseado no metamodelo A&A é composto por agentes e artefatos. Dessa forma, podem ser destacadas quatro potenciais interações fundamentais entre essas entidades: comunicação (*communication*), operação (*operation*), composição

(*composition*) e apresentação (*presentation*). Comunicação denota a troca de informações entre agentes. Operação está relacionada à utilização dos artefatos pelos agentes. Composição refere-se à ligação que pode existir entre diferentes artefatos. Apresentação relaciona-se à manifestação dos artefatos para os agentes.

2.2.1 CArtAgO

As características da plataforma CArtAgO descritas nessa subseção são baseadas nos trabalhos de Ricci (2009) e Ricci, Piunti e Viroli (2011).

CArtAgO é uma plataforma para criação de ambientes de trabalho (*work environments*) baseado na noção de artefato e *workspace*, definidos pelo metamodelo de agentes e artefatos. O modelo de desenvolvimento introduzido pelo metamodelo A&A e adotado em CArtAgO visa captar a natureza orientada a funções de artefatos que são usados (observados e controlados) por agentes.

A plataforma inclui uma API de código aberto [1] em Java para programação dos artefatos. Além disso, possui uma API para os agentes composta por conjuntos de funções básicas para criação e interação com artefatos, além de gerenciamento e ingresso em ambientes de trabalho. A plataforma possui ainda um ambiente de execução e ferramentas relacionadas que suportam a distribuição de ambientes de trabalho, gerenciamento do *workspace* e gerenciamento do ciclo de vida dos artefatos.

Os ambientes de trabalho são programados por desenvolvedores para serem utilizados por agentes como uma entidade que auxilia os mesmos na realização de suas tarefas. Em um sistema multiagente, o ambiente de trabalho é composto por um conjunto de recursos e ferramentas denominadas artefatos que são compartilhados e utilizados de forma cooperativa pelos agentes inseridos no *workspace*.

Um *workspace*, quando criado, contém um conjunto de artefatos pré-definidos que são criados juntamente com o mesmo. O conjunto de artefatos pré-definidos inclui: *factory artifact* (utilizado para instanciar artefatos), *registry artifact* (utilizado para controlar o conjunto de artefatos disponíveis no *workspace*) e *security-registry* (utilizado para gerenciar o conjunto de políticas definidas no *workspace*).

Um ambiente de trabalho pode ser visto como uma camada computacional extra do sistema multiagente, Figura 2.2, localizada entre os agentes e o ambiente externo, mediando a interação entre os mesmos, funcionando como um canal de comunicação. Nos ambientes baseados em artefatos, os agentes podem selecionar quais os artefatos que deseja explorar. O modelo *Role-Based Access Control* (RBAC) é adotado para especificar e gerenciar os aspectos relacionados à segurança controlando a entrada e saída de agentes do ambiente de trabalho, assim como o acesso e interação dos agentes com os artefatos.

A representação de um artefato é mostrada na Figura 2.3.

A interação entre agentes e artefatos acontece por meio de duas atividades fundamen-

[1] Disponível em http://cartago.sourceforge.org.

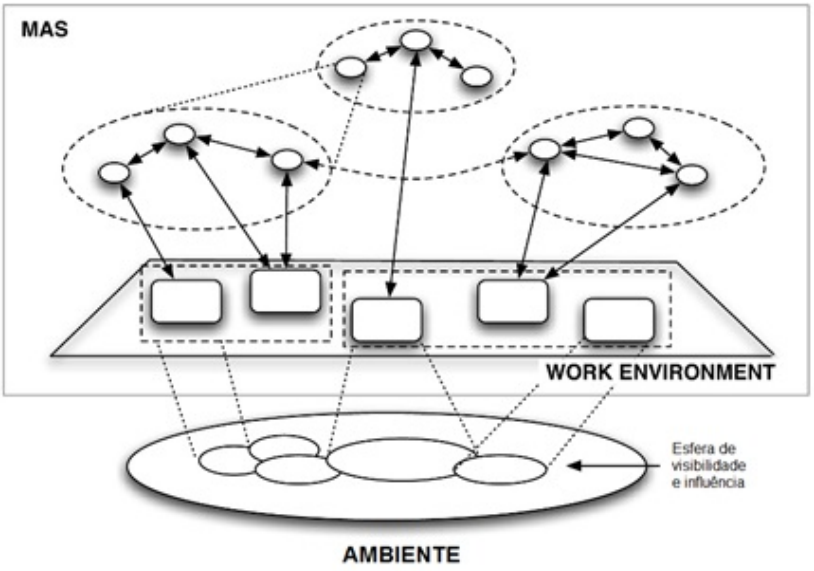


Figura 2.2: Localização de um *work environment* em um sistema multiagente extraído de Ricci et al.(2009).

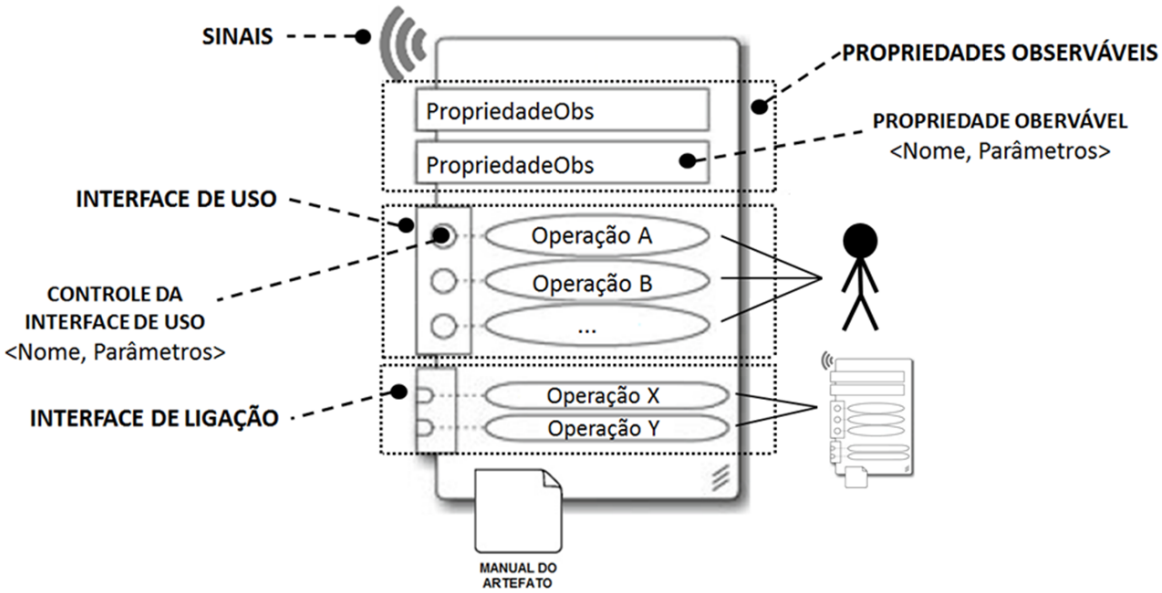


Figura 2.3: Representação de um artefato, figura adaptada de Ricci, Piunti e Viroli (2011).

tais desempenhadas pelos agentes no ambiente: ação e percepção. Os agentes utilizam os artefatos a partir da interface de uso (*Usage Interface*) onde podem disparar as operações e recebem o resultado pela percepção dos eventos gerados pelo funcionamento do artefato e pelas propriedades observáveis.

Operações representam processos computacionais que podem ser disparados por um agente. A execução de uma operação pode resultar tanto em mudanças do estado interno do artefato (não-observável) quanto na geração de um fluxo de eventos observáveis que

podem ser percebidos pelos agentes que utilizam ou observam o artefato. Ao ser executada, a operação gera um fluxo de eventos percebidos tanto pelo agente que é responsável pelo disparo da operação como por todos os agentes que estão observando este artefato.

Propriedades observáveis representam variáveis de estado que podem ser percebidas pelos agentes. Os valores dessas propriedades podem variar dinamicamente como resultado da execução de alguma operação no artefato. A execução de uma operação pode ainda gerar sinais (*signals*) que diferentemente de propriedades observáveis são úteis para representar eventos não persistentes do artefato que podem conter alguma informação. Para que esses eventos sejam percebidos pelos agentes é necessário estar observando o artefato. Para isso o agente deve executar a operação *focus* sobre o artefato. Um agente pode observar mais de um artefato ao mesmo tempo. Percepções relacionadas a valores de propriedades observáveis podem ser diretamente mapeadas para as crenças dos agentes desde que os mesmo estejam observando o artefato.

Artefatos podem ser equipados com um manual. Esse manual descreve a função específica do artefato, sua interface de uso e instruções de funcionamento do artefato. Manuais podem ser usados pelos agentes, em tempo de execução, para verificar se o artefato pode auxiliá-lo no alcance de seus objetivos.

Com intuito de habilitar a interação entre artefatos, os mesmos podem ser ligados a partir de interfaces de ligação (*link interfaces*). Interfaces de ligação permitem que um artefato possa executar uma operação sobre outro artefato. Operações de ligação não podem ser acessadas por agentes somente pelos artefatos ligados.

CArtAgO foi projetado para ser explorado por qualquer plataforma de programação de agentes permitindo que agentes com modelos, plataformas e arquiteturas heterogêneas interajam como parte de um sistema multiagente compartilhando um ambiente comum baseado em artefatos. CArtAgO disponibiliza algumas pontes (*bridges*) para que as plataformas de programação de agentes possam utilizar essa infraestrutura. Essas pontes estendem o repertório de ações providas pela plataforma de programação do agente com um novo conjunto de ações para serem usadas dentro do ambiente do artefato. Algumas ações podem ser destacadas: entrar e sair de *workspaces*, criação e utilização de artefatos, observação dos artefatos, dentre outras.

2.3 Organizações Incorporadas

Organizações incorporadas (*Embodied organizations*) (PIUNTI et al., 2009) (PIUNTI et al., 2010) é uma abordagem que tem como objetivo a integração entre organização, agentes e ambientes. Essa abordagem permite que os desenvolvedores foquem separadamente em cada uma dessas dimensões estabelecendo diferentes estilos de interação entre as mesmas.

Quando uma organização é atribuída a um sistema multiagente, os agentes devem lidar com dois tipos de interação: com a organização virtual (onde agentes adotam/deixam papéis, coordenam suas ações) e um ambiente físico (onde agentes executam ações, percebem eventos, se comunicam, acessam recursos, etc). A ideia da abordagem é integrar

organização, agentes e ambiente, a partir do metamodelo de agentes e artefatos (A&A), introduzindo o conceito de organização incorporada ao ambiente.

Nesta abordagem, uma organização baseada em artefatos é composta por um conjunto heterogêneo de artefatos organizacionais incorporados (*Embodied Organizational Artifacts* (EOA)) que podem ser de dois tipos: artefatos de ambiente (*Environmental Artifacts (EAs)*) e artefatos organizacionais (*Organizational Artifacts* (OAs)).

Os artefatos de ambiente podem ser considerados como recursos ou ferramentas utilizados para determinado propósito com intuito de auxiliar os agentes em suas atividades individuais e coletivas. Já os artefatos organizacionais, por sua vez, encapsulam funções organizacionais que podem ser exploradas pelos agentes para cumprir os propósitos organizacionais, como por exemplo, definição dos papéis dos agentes, operações para adotar e deixar um papel, informar alcance de objetivos do papel, normas a serem seguidas por esses agentes, dentre outros.

A vantagem dessa abordagem está no fato de que a utilização de artefatos organizacionais incorporados ao ambiente exime dos agentes a necessidade de lidar com estruturas complexas descritas na especificação organizacional, além de não haver a necessidade dos mesmos terem uma representação explicita da organização em mente para explorar seus serviços.

Nesta proposta, artefatos de ambiente podem ser utilizados também como mediadores entre organização e ambiente. Essa característica é definida por relações do tipo *count-as* e *enact-as*.

Count-as é uma relação entre os efeitos de uma ação executada por uma entidade (ação de um agente sobre um artefato) em um contexto específico e os efeito produzidos no nível organizacional do sistema. Nesse tipo de relação a ação de um agente pode adquirir o efeito de uma ação institucional. Um exemplo de uma relação do tipo *count-as* poderia ser em função de operações realizadas sobre artefatos de ambiente que podem produzir, por exemplo, um evento que sob o ponto de vista da organização pode contar como uma violação de normas.

Enact-as se refere a aspectos relativos a regulamentação da organização. Para tal são especificados dois mecanismos principais para aplicação das normas: regimentação (*regimentation*) e imposição (*enforcement*). Enquanto o mecanismo de regimentação impede que agentes realizem ações que não sejam permitidas por uma norma, o mecanismo de imposição é utilizado quando alguma violação é permitida. Esse mecanismo é utilizado então para detecção da violação de alguma norma.

Assim como a infraestrutura OrIAs, que é baseada no conceito de *embodied organizations*, a abordagem ORA4MAS (HÜBNER et al., 2010) e a plataforma JaCaMo (BOISSIER et al., 2011) propõe infraestruturas baseada em artefatos, onde elementos organizacionais são modelados em artefatos. ORA4MAS e JaCaMo são apresentadas com maior detalhamento no Capítulo 5.

2.4 Relação de Dependência entre os Agentes

A dependência é uma das fontes principais de interação social e cooperação entre os agentes. Podemos dizer que um agente ag1 é dependente socialmente de um agente ag2 caso entre eles exista uma interferência social. Por exemplo, para que ag1 atinja determinado objetivo é necessário que ag2 realize alguma ação que ag1 não é capaz de realizar. Nessa situação ag2 adquire certo poder social sobre ag1, visto que sua ação pode facilitar ou inviabilizar a obtenção dos objetivos de ag1. Exercer um poder social é um dos fatores que possibilitam que um agente consiga que outros agentes adotem suas tarefas (CASTELFRANCHI, 2003).

Os agentes que adotam requisições de outros agentes esperam obter também uma adoção como retorno. A cooperação e as trocas sociais é que tornam possível obter auxílio e também auxiliar os demais agentes na sociedade para que todos possam alcançar suas metas.

A teoria das dependências foi proposta originalmente por Castelfranchi, Miceli e Cesta (1992). Esta seção descreve resumidamente essa teoria com base nos trabalhos de SICHMAN et al. (1994) e SICHMAN (2003).

Segundo a teoria das dependências, um agente ag1 deve raciocinar sobre a maneira correta de delegar, por exemplo, uma ação t que necessita para um agente ag2, garantindo que essa ação seja realizada e que o agente que recebe essa ação esteja consciente que está realizando a mesma para que ag1 alcance seu objetivo. Uma possibilidade para que ag2 aceite realizar essa tarefa é que ag1 ofereça algo em troca, como uma ação para que ag2 atinja o mesmo objetivo (cooperação) ou outro objetivo (escambo social).

A dependência pode ser dividida em dois tipos: dependência objetiva e dependência subjetiva. A dependência objetiva é a dependência real que existe entre todos os agentes que compõem a sociedade, mostrando quem necessita de quem e para que. A dependência subjetiva é a dependência sob o ponto de vista de cada agente (parcial), ou seja, é quem o agente acredita que depende dele e quem ele acredita ser dependente (CASTELFRANCHI; FALCONE, 1998).

De acordo com a teoria da dependência social, um agente pode ser considerado a-autônomo (*a-autonomous*), r-autônomo (*r-autonomous*) e s-autônomo (*s-autonomous*) (CONTE; CASTELFRANCHI; VENEZIANO, 1998). Um agente ag1 é a-autonômo para um dado objetivo g de acordo com um conjunto de planos p se existe um plano que alcança este objetivo e todas ações que pertencem a este plano pertence ao seu próprio conjunto de ações.

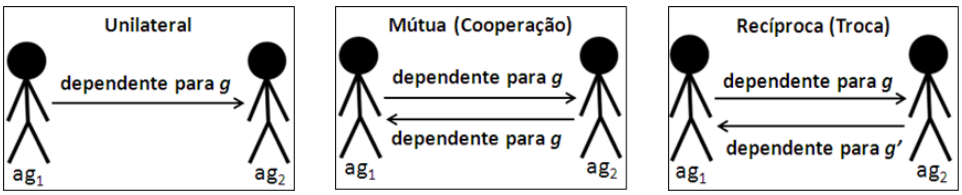
Um agente é r-autônomo para um dado objetivo g de acordo um conjunto de planos p se existe um plano que alcança este objetivo e todos os recursos necessários para execução deste plano pertence ao conjunto de recursos do agente. Por fim, um agente é dito s-autônomo, ou seja, socialmente autônomo, se é ao mesmo tempo a-autônomo e r-autonômo.

Similarmente, se um agente não possui todas as ações para execução de um determinado plano e obtenção de determinado objetivo, o agente é considerado a-dependente

(*a-depends*). Assim como, se não possui todos os recursos, o agente será r-dependente (*r-depends*). No caso de o agente ser ao mesmo tempo a-dependente e r-dependente, o agente é considerado socialmente dependente, denominado s-dependente (*s-depends*).

Assim, existem quatro situações que relacionam um agente a um determinado objetivo: *no goal* (NG), *no plan* (NP), *autonomous* (AUT) e *dependent* (DEP). *No goal* é quando um agente não tem determinado objetivo em sua lista de objetivos. *No plan* é quando um agente tem um objetivo mas não tem um plano para alcançar o mesmo. *Autonomous* é quando um agente tem o objetivo e planos para alcançar o mesmo sendo autossuficiente para realização de todas etapas desse plano. *Dependent* é quando um agente tem um objetivo e o plano para realizar esse objetivo, contudo não é capaz de realizar sozinho todos os passo desse plano (SICHMAN, 1998).

Em relação à direção, a dependência pode ser classificada em unilateral e bilateral, Figura 2.4. Dependência unilateral ocorre quando um agente ag1 depende de um agente ag2 em relação a alguma ação *a* para atingir algum objetivo, mas não existe uma dependência de ag2 em relação a ag1 para uma dada ação. Dependência bilateral ocorre quando ag1 depende de ag2 para uma ação *a* para atingir algum objetivo *g* e ag2 depende de ag1 para uma dada ação *a'* para atingir um objetivo *g'*. Em relação aos objetivos, a dependência pode ser mútua ou recíproca. Dependência mútua ocorre quando os agentes ag1 e ag2 reconhecem que dependem um do outro para o mesmo objetivo, e dependência recíproca ocorre quando os agentes reconhecem que dependem um do outro para diferente objetivos.

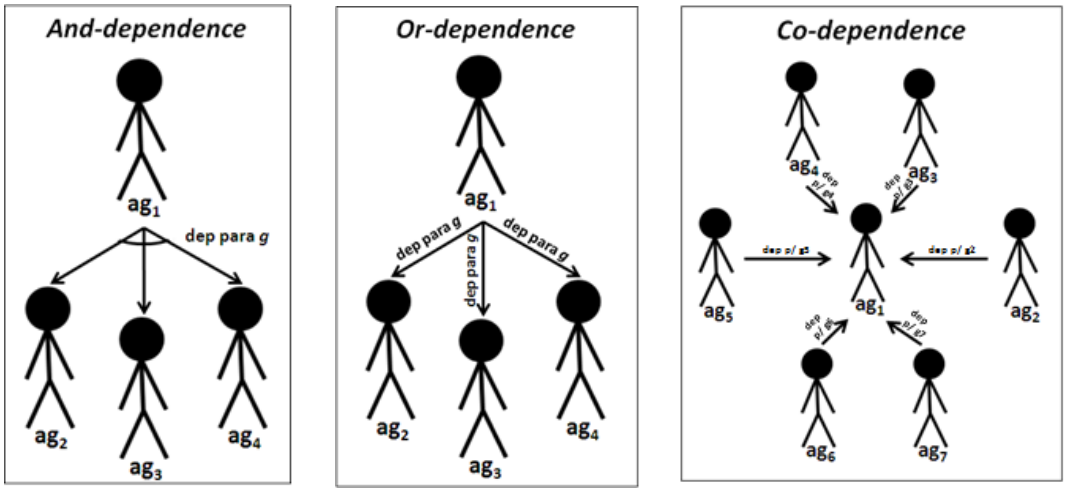


(a) Dependência unilateral. (b) Dependência mútua. (c) Dependência recíproca.

Figura 2.4: Classificação das dependências quanto a direção.

De acordo com Conte, Castelfranchi e Veneziano (1998), em relação ao número de parceiros, Figura 2.5, existem as relações que podem ser ainda de e-dependência (*and-dependence*), ou-dependência (*or-dependence*) e co-dependência (*co-dependence*). Uma e-dependência ocorre quando existem muitos agentes (parceiros) que precisam realizar diferentes ações para que o agente ag1 possa alcançar um objetivo *g*. Já uma ou-dependência ocorre quando existem muitos agentes (parceiros) que podem realizar uma determinada ação *a* para que o agente ag1 possa alcançar seu objetivo. Uma co-dependência ocorre quando um conjunto de agentes depende de determinado agente ag1 cada para seu próprio objetivo, e a menos que ag1 consiga executar as ações simultaneamente, será disputado pelos demais agentes.

Situações de dependência são utilizadas para auxiliar os agentes na escolha de parceiros mais propensos a aceitar uma proposta de coalizão, sendo apresentadas no trabalho de Sichman et al. (1994). Para o raciocínio social, o agente utiliza sua base de planos com



(a) *And-Dependence*. (b) *Or-Dependence*. (c) *Co-Dependence*.

Figura 2.5: Classificação das dependências em relação ao número de parceiros.

intuito de determinar suas relações de dependências. Contudo, se o agente tem acesso também aos planos dos demais agentes da sociedade e existe, por exemplo, uma dependência entre ag1 e ag2, pode ser útil para ag1 saber, a partir dos planos de ag2, se esse também está ciente dessa dependência.

As inferências em relação as dependências podem ser locais e mútuas. Um agente infere uma dependência localmente se utiliza apenas seus próprios planos para constatar esta relação. Por outro lado, se o agente utiliza além dos seus próprios planos, os planos do outro agente para inferir esta relação estamos diante de uma dependência mútua. Existem 6 situações de dependência que podem ocorrer entre dois agentes ag1 e ag2. Considera-se para as definições que ag1 é o agente que raciocina socialmente.

1. ***Independence* (IND):** O agente ag1 infere a partir de seus próprios planos que não depende do agente ag2 para realização de um objetivo g1.

2. ***Locally Believed Mutual Dependence* (LBMD):** O agente ag1 infere a partir de seus próprios planos que existe uma dependência mútua entre ele e ag2 para um objetivo g1, mas não pode constatar o mesmo utilizando os planos de ag2.

3. ***Mutually Believed Mutual Dependence* (MBMD):** O agente ag1 infere a partir de seus próprios planos que existe uma dependência mútua entre ele e ag2 para um objetivo g1, constatando o mesmo ao utilizar os planos de Ag2.

4. ***Locally Believed Reciprocal Dependence* (LBRD):** O agente ag1 infere a partir de seus próprios planos que existe uma dependência recíproca entre ele e ag2 para os objetivos g1 e g2, mas não pode constatar o mesmo utilizando os planos de ag2.

5. ***Mutually Believed Reciprocal Dependence* (LBRD):** O agente ag1 infere a partir de seus próprios planos que existe uma dependência recíproca entre ele e ag2 para os objetivos g1 e g2, constatando o mesmo ao utilizar os planos de ag2.

6. ***Unilateral Dependence* (UD):** O agente ag1 infere a partir de seus próprios planos que depende de ag2 para um objetivo g1, mas ag2 não depende de ag1 para nenhum de seus objetivos.

2.4.1 Mecanismos de Racíocinio Social

Raciocinar sobre os outros agentes presentes na sociedade é uma importante característica de um agente inteligente (SICHMAN et al., 1994). Para que isso seja possível, um agente necessita de um mecanismo de raciocínio social. Esse tipo de mecanismo permite representar explicitamente algumas propriedades dos outros agentes, explorar essa representação, além de monitorar e revisar essa representação com o intuito de evitar inconsistências (DEMAZEAU; SICHMAN, 2001).

O mecanismo de raciocínio social permite que um agente avalie a susceptibilidade de outros agentes na adoção de seus objetivos. Este mecanismo é baseado na teoria da dependência e poder social e permite uma melhor adaptação do agente no ambiente, uma vez que melhora o desempenho do mesmo no alcance de seus objetivos. O mecanismo auxilia o agente no processo de formação de parcerias ou coalizões para realização de tarefas que sozinho o agente estaria impossibilitado de realizar (SICHMAN, 2003). A seguir são apresentados alguens mecanismos de raciocínio social encontrados na literatura.

O trabalho desenvolvido por Sichman et al. (1994) é um dos primeiros a definir um mecanismo de raciocínio social que utiliza a noção de dependência e poder social em sistemas multiagentes. O modelo é baseado em formação dinâmica de coalizões onde agentes dependem uns dos outros para realização de tarefas. Para que seja possível o raciocínio social, os agentes mantem a informação dos demais agentes da sociedade em uma estrutura denominada descrição externa (*external description*) utilizada pelos agentes para verificação das dependências. A descrição externa é composta de várias entradas, onde cada uma corresponde a um determinado agente, contendo as ações que este pode realizar, os recursos controlados por ele e os planos utilizados para obtenção dos objetivos. Redes de dependência *dependence networks*) são utilizadas para representar as relações de dependência de um agente. Esse trabalho apresenta ainda a noção de situação de dependência (*dependence situations*). O modelo apresentado é utilizado para implementar o simulador DEPNET que busca relações e situações de dependência e gera uma rede de dependência para o agente.

Em Conte, Castelfranchi e Veneziano (1998) o simulador DEPNET é estendido para o sistema denominado MICROdep. Enquanto DEPNET gera redes de dependência baseado nas propriedades individuais dos agentes, MICROdep busca possíveis parcerias a partir dessas redes. São utilizados os conceitos de *and*, *or* e *co* dependência para definir o conceito de poder de negociação (*negotiation power*) .

No trabalho de Sichman (1998) é apresentado um sistema denominado DEPINT para formação de coalizão entre agentes, utilizando o mecanismo de raciocínio social proposto em Sichman (1994). Este sistema permite que um agente avalie, durante as interações, se a representação que tem em relação aos demais agentes está consistente. O trabalho introduz o conceito de *goal situation* (*No Goal*, *No Plans*, *Autonomous* e *Dependent*) que relaciona um agente a determinado objetivo.

O trabalho desenvolvido por Alonso (1998) é o primeiro mecanismo a considerar custo e benefícios na formação de coalizões baseadas em dependência social, combinando conceitos de dependência social e racionalidade baseada em utilidade. São utilizadas árvores *and*/*or* para representação dos objetivos/subobjetivos dos agentes. Além disso, é utilizado o conceito de dependência fraca na qual um agente é autônomo na realização de seus objetivos, mas prefere cooperar para obtenção do mesmo.

O trabalho de David, Sichman e Coelho (1999) também combina as abordagens quantitativa e qualitativa para escolha dos parceiros. Para tal, a descrição externa apresentada em Sichman (1994) é estendida para incluir uma propriedade de peso que está relacionada a importância de um objetivo para um agente e uma propriedade de custo para cada uma de suas ações. O trabalho ainda apresenta o conceito de dependência inversa considerada como uma ferramenta de raciocínio estratégico sobre as necessidades dos possíveis parceiros.

O trabalho de Morgado e Gaspar (2000) considera que o raciocínio sobre os outros pode estar além dos agentes que determinado agente está diretamente relacionado resultando em uma cadeia de coalizões aninhadas. O raciocínio de um agente sobre os outros para estabelecer uma coalizão, pode adotar duas diferentes perspectivas: global (raciocinar sobre as relações de dependência em relação às ações dos agentes da sociedade que podem participar da coalizão) e local (agente raciocina somente sobre os agentes que depende diretamente).

Em Sichman e Conte (2002) é apresentada uma estrutura denominada grafos de dependência (*dependence graphs*). Essa estrutura é uma extensão da noção de redes de dependência, proposta em Sichman (1994), que permite representar todas as dependências existentes em um sistema multiagente. O autor apresenta ainda o conceito de grafos reduzidos. O sistema PART-NET ++, proposto por Monteiro e Sichman (2006), estende o sistema PART-NET (CONTE; PEDONE, 1998), utilizado para estudar parcerias entre pares de agentes, para parcerias formadas por múltiplos agentes. O sistema utiliza grafos de dependência para representação das relações entre os agentes da sociedade. Esse sistema trabalha apenas com dependências recíprocas e utiliza agentes que seguem diferentes estratégias de escolha (utilitaristas, substancialistas e avarenta) para busca de parcerias. O trabalho introduz ainda os conceitos de estratificação e tolerância social.

Nos artefatos de dependência projetados na infraestrutura, é utilizada uma modificação da descrição externa proposta no mecanismo de raciocínio social desenvolvido por Sichman et al. (2004). Esse mecanismo foi escolhido como estrutura para armazenar as características das dependências dos agentes por ser um mecanismo simples que permite uma consulta rápida dos possíveis parceiros e também por permitir representar todas as características desejadas das dependências no protótipo inicial da infraestrutura.

2.4.2 Coalizões Baseadas em Dependências

Uma Coalizões Baseadas em Dependências (CBD) (SICHMAN, 2003) consiste em utilizar um mecanismo de raciocínio social para resolução de problemas de forma cooperativa, detectando possíveis escolhas para formação da coalizão.

Esta técnica é composta por alguns passos que seguem. Ao representar suas relações de dependências existentes em alguma estrutura, o agente deve determinar um objetivo a atingir e um plano para realização do mesmo. A partir disso, o agente deve calcular sua situação para realização desse objetivo. Caso sua situação seja autônoma, as ações necessárias para execução do objetivo são realizadas. Caso o agente constate que é dependente para realização deste objetivo ele calcula sua situação de dependência para tal objetivo levando em consideração os agentes os quais depende, escolhendo um parceiro que possa ajudá-lo.

Ao efetuar a escolha de um parceiro, uma proposta de coalizão é enviada solicitando ao mesmo a realização da ação necessária para obtenção de seu objetivo oferecendo em troca uma outra ação que esse agente necessite. Caso o parceiro aceite a proposta, a coalizão é formada, caso contrário, outro parceiro deve ser escolhido. Caso não existam mais parceiro, outro plano deve ser escolhido. Caso não exista outro plano, escolhe-se outro objetivo. Caso não exista outro objetivo, o processo é finalizado.

2.5 Reputação

A reputação é uma importante propriedade a ser considerada em sistemas multiagentes. No contexto organizacional reputação pode ser utilizada para auxiliar tanto na aplicação de normas, como um incentivo para que os agentes se comportem como esperado, como também para auxiliar na busca por possíveis parceiros para uma interação sendo utilizada como um auxílio no processo decisório entre as várias opções de parcerias.

Nesta seção é descrito o modelo cognitivo de reputação de Conte e Paolucci (2002) utilizado como base para o projeto dos artefatos de reputação utilizados na infraestrutura proposta. Este modelo é baseado na noção de estado mental dos agentes (crença e meta-crença) sendo definido a partir de dois conceitos principais inter-relacionados: imagem e reputação.

A reputação consiste na transmissão de crenças sobre como os agentes são avaliados em relação a uma conduta desejada socialmente. A reputação se espalha como um efeito de propagação de crenças (Figura 2.6). A formação da reputação acontece seguindo algumas etapas. Primeiramente um agente Ag1 realiza uma avaliação sobre um dado agente Ag2 durante alguma interação ou observação (experiência) com esse agente. Essa experiência contribui para a formação da imagem social sobre o agente Ag2, que nada mais é do que um conjunto de crenças resultantes da avaliação sobre esse agente. Já a reputação é definida como o efeito causado pela transmissão desta imagem (avaliação social). Conhecer a reputação de um agente, devido a difusão da reputação, pode influenciar nas interações futuras com esse agente. Esse fenômeno é denominado contágio.

Assim, a imagem é uma crença de avaliação que diz se um agente é bom ou ruim em relação a dado comportamento. Já a reputação está um nível acima da imagem, sendo uma meta-crença sobre a mente dos outros, ou seja, representa a avaliação dos outros em relação a um determinado agente.

O processo de formação da imagem e difusão da reputação envolve quatro tipos de

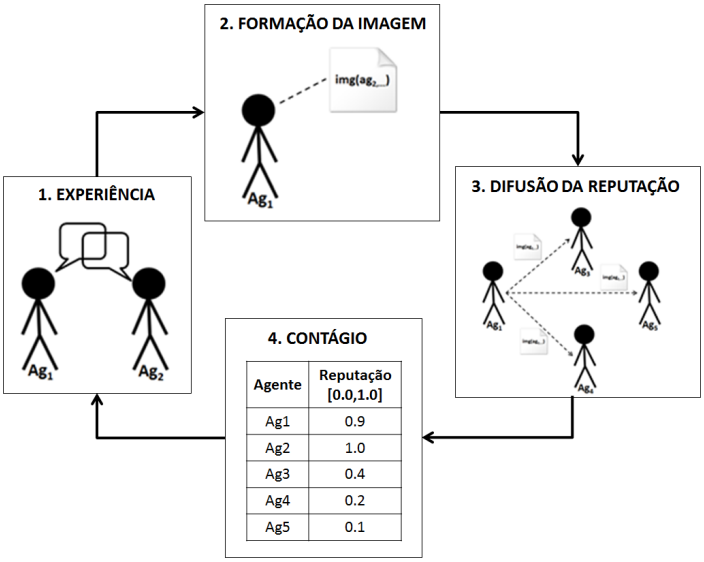


Figura 2.6: Processo de formação e transmissão da reputação, figura adaptada de Conte e Paolucci (2002).

entidades (agentes, grupos, organizações): avaliadores (*evaluators*), alvos (*targets*), beneficiários (*beneficiaries*) e transmissores (*third party*). Avaliadores são aqueles capazes de formar uma crença positiva ou não em decorrência de uma avaliação a partir de uma interação ou observação. Alvos são os as entidades objetos da avaliação. Beneficiários são entidades que se beneficiam da avaliação de outras entidades. Transmissores são aquelas entidades capazes de difundir as informações em relação à reputação de outras entidades (mesmo que não conheça essa entidade ou acredite que essa reputação seja falsa).

Como a imagem se refere à crença (avaliação) enquanto que a reputação se refere à meta-crença (transmissão da crença de avaliação), essas informações podem não serem iguais. A imagem em relação a uma entidade não implica que esse valor seja o mesmo valor da reputação dessa entidade, e vice versa. A imagem de um alvo está relacionada à verdadeira informação obtida pela avaliação. Entretanto a reputação pode não condizer com este mesmo valor. Como não é um atributo estático a reputação pode ser submetida a distorções e manipulação e pode ser usada para elogiar ou envergonhar indivíduos, em função de seu comportamento. Dessa forma, a reputação pode ser perdida tão facilmente quanto adquirida.

Existe uma preocupação em função da necessidade de a reputação de um índividuo ser condizente com seu comportamento, ou seja, a reputação estar correta. Visto que isso pode interferir nas relações sociais do agente. Uma reputação boa pode manter parceiros por um longo tempo enquanto que uma fama ruim não merecida causa imediatamente isolamento social. Apesar disso, a reputação pode ser utilizada como um mecanismo de autodefesa onde os agentes podem isolar parceiros não confiáveis.

Além de a imagem poder auxiliar um agente a decidir quando é conveniente ou não interagir com determinado agente. O conhecimento dessa imagem pode ser conveniente para outros agentes para avaliar possíveis interações com um determinado alvo, visto que transmitindo essa imagem um agente pode influenciar outros agentes em suas tomadas de

decisão.

Transmitir a reputação de alguém não necessariamente implica no comprometimento do transmissor com a veracidade da informação transmitida. Além disso, existem quatro aspectos principais relacionados à decisão de transmitir uma reputação que devem ser considerados:

- Porque (ou não) transmitir a reputação de alguém: o porquê está relacionado a diversos fatores que auxiliam o agente na decisão de transmitir ou não uma reputação. Dentre esses fatores podem ser destacados aceitação, reputação da fonte, responsabilidade social e benevolência. Aceitação refere-se ao grau no qual o casamento da reputação com a imagem que o candidato a transmissor tem do alvo. A meta-crença pode ou não corresponder com sua crença (imagem), mesmo assim o transmissor pode repassar essa informação sobre o alvo. Reputação da fonte está relacionada a decisão de transmitir ou não a reputação ser afetada pela reputação da fonte dessa informação ou pelo menos da imagem que o agente tem dessa fonte. Responsabilidade Social está relacionada a integrar uma forma de controle ao sistema para prevenir que agentes espalhem uma reputação negativa sobre outro agente com intuito de prejudicar o mesmo. Com esse controle o sistema pode obrigar o agente que espalhou a reputação distorcida a reparar o mal causado. A benevolência está relacionada à identidade do endereçado. Se agente é benevolente a um agente alvo a reputação transmitida é precisa (seja essa reputação boa ou má). Se o agente é um beneficiário a transmissão da reputação será mais cautelosa ao transmitir uma boa reputação ao invés de uma ruim.

- Para quem transmitir a reputação: a escolha para quem transmitir a reputação é influenciada por fatores como: reciprocidade, discrição da fonte, usabilidade, persuasão e cointeresse. Reciprocidade está relacionada a probabilidade dos beneficiários em retornar informação. Discrição da fonte está relacionada a proteção da identidade da fonte da informação com intuito de evitar retaliação no caso de transmissão de uma má reputação. A usabilidade, esta relacionada à probabilidade dos beneficiários forneceram informações relevantes. A persuasão está relacionada à probabilidade dos beneficiários de aceitar informações sobre a reputação. Por fim, cointeresse está relacionado ao interesse e curiosidade do beneficiário sobre a informação.

- Sobre quem é essa informação (o alvo da reputação): está relacionado a fatores como fama (visibilidade dos alvos), distância (física e social) existente entre os agentes e aceitabilidade (credibilidade das avaliações).

- Como e quando espalhar a reputação: esta relacionada à forma de transmissão da reputação que geralmente acontece a partir da interação entre os agentes onde um agente recebe uma informação e transmite aos demais.

2.6 Normas

Normas são utilizadas para orientar, controlar ou regular um comportamento. Além disso, normas são indispensáveis para superar problemas relacionados à coordenação de

sistemas complexos e heterogêneos. Dessa forma, normas podem ser utilizadas em sistemas multiagentes para lidar com a heterogeneidade, a autonomia e a diversidade de interesses entre os agentes autônomos (BOELLA; VAN DER TORRE; VERHAGEN, 2006) (LóPEZ; LUCK; DINVERNO, 2006).

Segundo López, Luck e Dinverno (2006) normas especificam padrões de comportamento para um conjunto de agentes. Esses padrões podem ser ações a serem executadas, restrições sobre algumas ações dos agentes, objetivos que devem ser satisfeitos ou evitados pelos agentes, dentre outras definições. Além disso, normas podem ser usadas como sinônimo para obrigação, proibição, permissão, leis sociais, dentre outros. Dessa forma, é difícil encontrar uma definição única para normas.

Contudo, de acordo com López, Luck e Dinverno (2006), independente da definição, uma norma pode ser caracterizada pelas seguintes propriedades: prescrição (*prescriptiveness*), sociabilidade (*sociality*) e pressão social (*social pressure*). Prescrição está relacionada as normas informarem a um agente como se comportar. Sociabilidade está relacionada a situações em que mais de um agente está envolvido. Pressão social está relacionada a mecanismos que forçam os agentes a cumprirem as normas.

Normas são sempre endereçadas a um conjunto de agentes e devem sempre especificar as situações que os agentes devem cumprir as mesmas, visto que as normas geralmente não são aplicadas a todo momento, mas apenas em circunstâncias particulares. Como agentes são entidades autônomas, os mesmos possuem a liberdade na decisão no cumprimento ou não de uma norma. Dessa forma, são necessários mecanismos para promover o cumprimento de normas ou evitar o descumprimento das mesmas. Os agente podem receber recompensas (*rewards*) quando normas são cumpridas e punições (*punishments*) quando normas são descumpridas.

O cumprimento ou descumprimento de normas pode ser monitorado de várias formas. A forma mais simples verifica se um agente cumpriu ou violou determinada norma. No entanto, em algumas situações é preciso distinguir casos de quando um agente não cumpriu determinada norma na íntegra, contudo fez um esforço para não violá-la.

As normas introduzidas em um sistema multiagente funcionam como um meio para garantir a ordem social, além de evitar conflitos entre os agentes. Contudo nem todos os conflitos existentes podem ser antecipados. Assim um sistema multiagente normativo deve possuir um mecanismo que permita a criação de novas normas, alteração de normas existentes ou exclusão de normas que se tornaram obsoletas. Normas podem ser criadas tanto pelo projetista de agentes quanto pelos próprios agentes desde que os mesmos possuam um papel que que os habilite para essa função.

Dessa forma, um agente pode interagir com um sistema normativo de várias formas. Seja criando novas normas, modificando ou excluindo normas existentes, seja controlando o cumprimento das normas, recompensando agentes que cumprem as normas e punindo agentes que violam as mesmas (BOELLA; VAN DER TORRE; VERHAGEN, 2006).

Quando uma norma é adotada por um agente, uma cópia dessa norma é criada na mente do agente, isso representa a internalização de uma norma por um agente. No tra-

balho de López, Luck e Dinverno (2006), essa situação é denominada instância de uma norma (*norm instance*).

Não apenas a relação entre normas e agentes deve ser estudada, mas também a relação entre normas e outros conceitos sociais (BOELLA; VAN DER TORRE; VERHAGEN, 2006). Neste trabalho, por exemplo, normas estão inseridas na organização e estão relacionadas ao conceito de reputação.

Como normas permitem estabelecer como agentes devem se comportar em uma determinada situação, as normas podem ser uma importante fonte de informação sobre a habilidade ou boa vontade dos agentes para realizar certas tarefas. Uma forma de avaliar o comportamento dos agentes em relação ao cumprimento ou violação das normas seria utilizar alguma métrica capaz de caracterizar esse comportamento. Dessa forma, quando as normas são utilizadas como um mecanismo para regular as atividades, por exemplo, em uma organização de agentes, o monitoramento das mesmas pode prover importantes informações sobre os agentes em relação aos seus compromissos sociais (FALCONE et al., 2013).

Em um sistema regido por normas, utilizando alguma métrica que avalie um agente quanto ao cumprimento/descumprimento de normas, como por exemplo, a reputação, é possível obter informações sobre as interações passadas relacionadas a diferentes formas que esse agente respondeu às normas. Se em um sistema essa reputação influência nas relações sociais desse agente, como por exemplo, para escolha de parceiros, uma reputação ruim quanto ao cumprimento de normas pode levar esse agente a um isolamento social. Neste caso a reputação pode ser usada como um mecanismo de auxílio na imposição das normas.

2.7 Comunicação em Sistemas Multiagentes

Em um sistema multiagente os agentes estão inseridos em um ambiente comum e dinâmico. Um ambiente comum cria dependências entre os agentes, como por exemplo, causando interferência no nível de planejamento e execução de ações. Um ambiente dinâmico leva a modificações imprevisíveis nos objetivos dos agentes e a necessidade da execução eficaz dos planos.

Dessa forma, assim como podem existir objetivos que são difíceis de serem alcançados individualmente havendo a necessidade de cooperação e coordenação entre os agentes, existem também situações conflitantes sob as quais os agentes precisam chegar a um acordo. Nessas situações citadas, para que os agentes possam se coordenar e resolver alguns conflitos os mesmos necessitam de habilidades sociais e de comunicação.

Para que a comunicação ocorra é necessário uma linguagem de comunicação comum entre os agentes. As linguagens de comunicação de agentes (*Agent Communication Language* (ACL)) mais utilizadas são: KQML (FININ et al., 1994) e FIPA[2]. Ambas as linguagens de comunicação foram desenvolvidas baseadas na teoria dos atos de fala (*speech acts theory*).

[2]http://www.fipa.org/

Essa seção tem como objetivos apresentar as características gerais relacionadas a teoria dos atos de fala e as linguagens KQML e FIPA. São apresentados ainda dois protocolos de comunicação (*Request* e *Contract Net*) que são utilizados para os desenvolvimento dos artefatos da infraestrutura de comunicação integrada a infraestrutura OrIAS.

2.7.1 Teoria dos Atos de Fala

A teoria dos atos de fala (*speech acts theory*), considera que o ato de enunciar uma frase tem características similares a uma ação física, visto que ambas alteram o estado de mundo.

Nessa teoria foram verificados verbos performativos que corresponde a diferentes tipos de atos de fala. Exemplos desses verbos são: *request*, *inform* e *promisse*. Além disso, são identificados três diferentes aspectos dos atos de fala: ato locucionário, ato ilocucionário e ato perlocucionário. Um ato locucionário se refere ao ato de pronunciar uma frase. Um ato ilocucionário se refere a uma ação posterior ao enunciado da frase. Por fim, um ato perlocucionário se refere ao efeito da ação.

Existem algumas propriedades que devem ser asseguradas em um ato de fala entre um locutor e um ouvinte. Considerando o exemplo da performativa *request* entre um agente locutor e um agente ouvinte são apresentadas as propriedades:

- Condição normal de I/O: Um agente ouvinte deve estar habilitado a receber a requisição do agente locutor.
- Condição preparatória: O agente ouvinte de estar habilitado a realizar a ação solicitada através da performativa e o agente locutor deve acreditar que o agente ouvinte esta habilitado para a ação.
- Condição de sinceridade: Distingue uma performativa de *request* sincera quando o agente locutor realmente deseja que a ação referente a essa performativa seja realizada de uma performativa de *request* não verdadeira.

Existe ainda uma classificação sistemática de possíveis tipos de atos de fala:

- Representativos: Um ato deste tipo se refere a informação do emissor em relação a um determinado estado de mundo.
- Diretivos: representa a tentativa (requisição) do locutor para que o ouvinte realize alguma coisa.
- Comissivos: Compromete o locutor a realizar alguma ação para o ouvinte. Relativo a uma promessa.
- Expressivos: Expressa algum estado psicológico, como gratidão.
- Declarativos: Efetua mudanças no estado institucional.

2.7.2 KQML

KQML (*Knowledge Query and Manipulation Language*) é uma linguagem baseada em mensagens para comunicação de agentes que define um formato comum para mensagens. Cada mensagem possui uma performativa e um número de atributos. Um exemplo de uma mensagem em KQML é apresentada no Código 2.1. A performativa *ask-one* utilizada na mensagem indica a um pedido de informação direcionado a um agente. Na mensagem o remetente está solicitando informações sobre um determinado preço para o destinatário stock-server (WOOLDRIDGE, 2009).

Código 2.1: Exemplo de uma mensagem KQML, código extraído de Wooldridge (2009).

```
    : content (PRICE IBM ?price)
    : receiver stock-server
    : language LPROLOG
    : ontology NYSE-TICKS
)
```

Os demais componentes da mensagem representam seus atributos. O atributo *content* indica o conteúdo da mensagem. O atributo *receiver* define o destinatário da mensagem. O atributo *language* a linguagem que o conteúdo da mensagem está escrito. O atributo *ontology* informa a terminologia do conteúdo da mensagem.

A ideia do uso do KQML é que agentes possam se comunicar mesmo sendo implementados usando diferentes linguagens de programação e paradigmas. Contudo apesar dessa ideia inicial, KQML é criticado em função de uma serie de fatores. De acordo com Wooldridge (2009), o conjunto básico de performativas KQML não foi fortemente restringido, nem as formas de obtenção de mensagens foram definidas precisamente. Logo diferentes implementações de KQML foram desenvolvidos e a comunicação entre os agentes foi dificultada logo não foi possível obter a interoperabilidade esperada. Essas críticas, dentre outras, levaram ao desenvolvimento de outra linguagem pelo consórcio FIPA.

2.7.3 FIPA

FIPA (*Foundation for Intelligent Physical Agents*) é uma associação sem fins lucrativos responsável por desenvolver padrões para sistemas de agentes. O objetivo principal dessa iniciativa foi o desenvolvimento de uma linguagem de comunicação de agentes (ACL). Essa associação define além de um conjunto de performativas de fala, alguns protocolos de interação e alguns tipos de leilões (WOOLDRIDGE, 2009).

A ACL desenvolvida pela FIPA é similar ao KQML. A estrutura das mensagens é a mesma e os atributos são muito similares. Contudo a grande diferença entre as duas está na coleção de performativas providas e, devido às criticas em relação ao KQML, FIPA-ACL define uma semântica para seus atos de fala em termos de uma linguagem formal denominada SL. Resumidamente o que a FIPA-ACL faz é mapear cada mensagem ACL para uma formula SL que define a restrição que o remetente da mensagem deve satisfazer para estar de acordo com o padrão FIPA-ACL, além de mapear a mensagem para uma formula SL que define o propósito da mensagem (WOOLDRIDGE, 2009).

Nesta subseção não são apresentadas todas as performativas providas pela linguagem, nem todos os protocolos, ou leilões definidos. Como a infraestrutura de comunicação que é integrada a infraestrutura OrIAs utiliza o padrão FIPA para implementação dos ar-

tefatos de comunicação são apresentados com maior detalhamento apenas os protocolos utilizadas na infraestrutura (*Request Protocol* e *Contract Net Protocol*) e as performativas utilizadas nesses protocolos.

2.7.3.1 FIPA Request Protocol

O FIPA *Request Protocol* é utilizado para que um agente possa requisitar para outro a execução de alguma tarefa. A representação desse protocolo é apresentada na Figura 2.7. Qualquer interação utilizando este protocolo deve possuir um identificador. Assim cada mensagem trocada deve ser identificada por um (`conversation-id`). Esse identificador permite que o agente gerencie suas estratégias de comunicação e atividades.

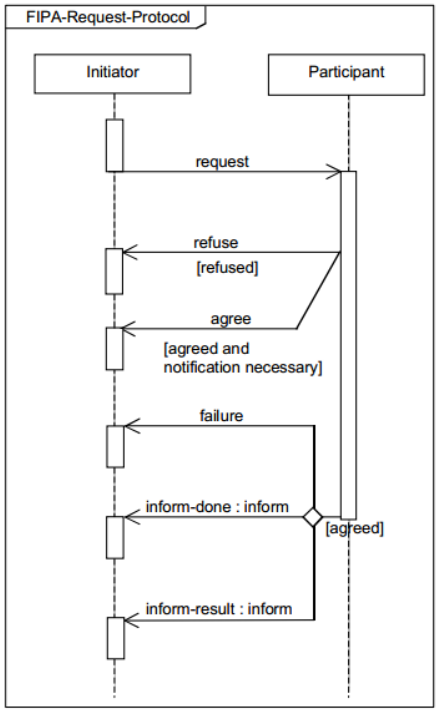


Figura 2.7: Protocolo *Request* conforme definido pela FIPA, figura extraída de FIPA (2006)

Nesse protocolo, um agente denominado `initiator` requisita a realização de uma tarefa para um agente denominado `participant`. O agente `participant` processa a requisição e decide se aceita ou recusa a solicitação. Se a requisição é aceita, uma mensagem do tipo `agree` é enviada para o agente `initiator`. Caso contrário uma mensagem `refuse` é enviada.

A mensagem do tipo `agree` é opcional, sendo somente obrigatória em casos onde uma aceitação explicita para realização e uma tarefa é necessária. Uma vez que o agente aceite a requisição, depois de concluída a tarefa, o agente `participant` deve enviar uma mensagem informando que a tarefa foi realizada (`inform-done`), ou informando que a tarefa foi realizada e os resultados (`inform-result`), ou informando que houve uma falha ao atender a requisição (`failure`).

2.7.3.2 *FIPA Contract Net Protocol*

O FIPA *Contract Net Protocol* é uma pequena modificação do protocolo *Contract Net* proposto originalmente por (SMITH, 1980). Essa modificação adiciona atos de comunicação de confirmação e de recusa. A representação desse protocolo é apresentada na Figura 2.8. Assim como no FIPA *Request Protocol*, cada mensagem trocada deve ser identificada por um `conversation-id`.

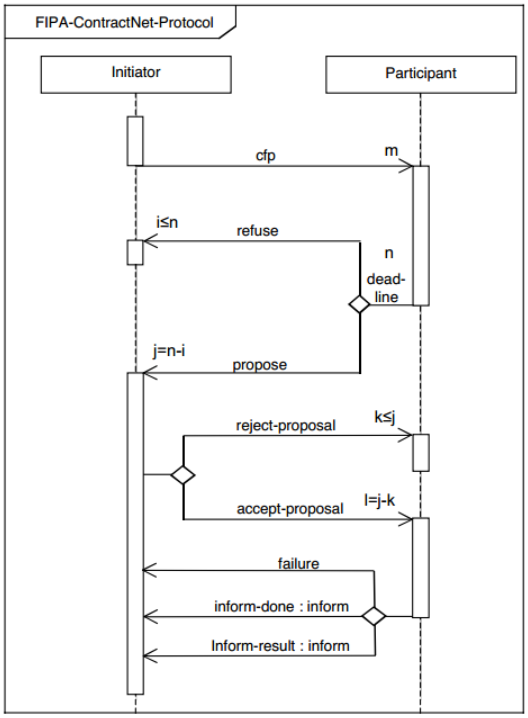


Figura 2.8: Protocolo *Contract Net* conforme definido pela FIPA, figura extraída de FIPA(2002)

Neste protocolo um agente denominado `initiator` deseja que uma tarefa seja realizada por uma ou mais agentes. Esse agente então assume o papel de gerente e solicita *m* propostas para outros agentes a partir da performativa `call for proposals (CFP)`. Essa mensagem especifica a tarefa e as condições para sua execução.

Os agentes denominados `participants` recebem as mensagens do tipo CFP respondem com mensagens do tipo `propose`. Nessa mensagem estão as ofertas dos agentes `participants` para realização da tarefa. Essas ofertas podem estar relacionadas a preço, tempo para tarefa ser realizada, dentre outros. Agentes `participants` podem também recusar a tarefa.

Existe um tempo (*deadline*) para que as propostas sejam realizadas. Após esse tempo o agente gerente (`initiator`) avalia a proposta e seleciona um ou mais agentes para realização da tarefa. Aos agentes selecionados será enviada uma mensagem do tipo `accept-proposal`. Os agentes que foram rejeitados recebem uma mensagem do tipo

`reject-proposal`.

Quando um agente `initiator` aceita uma proposta, o agente `participant` que recebe a mensagem se compromete a realizar a tarefa. Ao completar a tarefa, o agente `participant` envia uma mensagem que a tarefa foi concluída. Essa mensagem pode ser do tipo `inform-done` para informar a realização da tarefa ou do tipo `inform-result` que seria uma versão mais completa da mensagem `inform-done`. Se o agente `participant` falha na conclusão da tarefa, uma mensagem do tipo `failure` é enviada.

3 ORIAS

Nesta seção é apresentada a infraestrutura micro-organizacional proposta denominada OrIAs (*Organizational Infrastructure for Agents*). A infraestrutura é baseada no metamodelo de Agente e Artefatos A&A (OMICINI; RICCI; VIROLI, 2008) e no conceito de organizações incorporadas (*Embodied Organizations*) (Seção 2.3) que permite integrar organizações, agentes e ambiente. A infraestrutura é composta de artefatos organizacionais e artefatos de comunicação sendo implementada utilizando a plataforma CArtAgO que é baseada no metamodelo A&A.

Para que OrIAs torne a organização disponível aos agentes é necessário descrever as características organizacionais em um arquivo com uma sintaxe definida que pode ser interpretada pela infraestrutura. A infraestrutura permite especificar a organização a partir das quatro dimensões organizacionais: estrutural, funcional, dialógica e normativa.

Diferentemente de outras infraestruturas como JACAMO (BOISSIER et al., 2011) cujos artefatos são projetados para um modelo organizacional específico, no caso o MOISE (HÜBNER; SICHMAN; BOISSIER, 2002), a infraestrutura apresentada não é projetada exclusivamente para um modelo organizacional. A ideia é prover uma infraestrutura genérica na qual a especificação dos modelos organizacionais existentes possa ser traduzida para a especificação aceita pela infraestrutura.

A infraestrutura suporta organizações especificadas com apenas uma dimensão organizacional (como em TAEMS (DECKER, 1996) (funcional) e AGR (FERBER; GUTKNECHT; MICHEL, 2004) (estrutural)), duas dimensões organizacionais (como em STEAM (TAMBE, 1997) (funcional e estrutural)), três dimensões organizacionais(como em MOISE (HÜBNER; SICHMAN; BOISSIER, 2002) (funcional, estrutural e normativa)) e quatro dimensões organizacionais (como em OperA (DIGNUM et al., 2002)(funcional, estrutural, normativa e dialógica)). Assim com pequenas modificações ou extensões a infraestrutura OrIAs possa funcionar de acordo com algum modelo organizacional existente.

O projeto da infraestrutura é estruturado em quatro etapas. Na descrição de cada etapa é possível tratar separadamente: aspectos relacionados aos conceitos organizacionais presentes na infraestrutura, a modelagem desses conceitos a partir da utilização de artefatos, a modelagem dos serviços disponibilizados pela infraestrutura e por fim detalhes de implementação. As etapas do projeto da infraestrutura, ilustradas na Figura 3.1, são:

1. Etapa Organizacional: Esta etapa se propõe a descrever as características das quatro

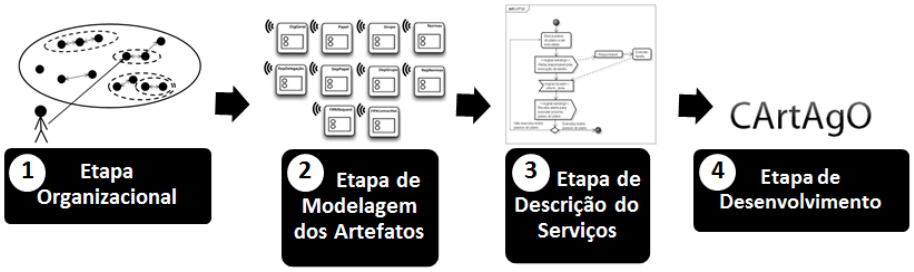


Figura 3.1: Etapas de criação de infraestrutura.

dimensões organizacionais presentes na infraestrutura: estrutural, funcional, dialógica e normativa.

2. Etapa de Modelagem dos Artefatos: Nesta etapa os conceitos organizacionais e o funcionamento da infraestrutura definidos a partir das quatro dimensões organizacionais são modelados em artefatos.

3. Etapa de Descrição dos Serviços: Nesta etapa é apresentada a dinâmica de funcionamento da infraestrutura sob a perspectiva dos serviços disponibilizados pelos artefatos.

4. Etapa de Desenvolvimento da Infraestrutura: Nesta etapa são apresentados alguns detalhes relativos à implementação da infraestrutura e a especificação organizacional utilizada como entrada para infraestrutura que torna a organização disponível para os agentes.

3.1 Etapa Organizacional

A infraestrutura organizacional suporta quatro dimensões organizacionais: estrutural, funcional, dialógica e normativa. Nas seções subsequentes são explicadas detalhadamente as características de cada uma das dimensões presentes na infraestrutura.

3.1.1 Dimensão Estrutural

A dimensão estrutural permite definir os papéis e grupos existentes na organização. Na infraestrutura a adoção de um papel permite que os agentes passem a fazer parte da organização ao mesmo tempo em que passam a ter seus comportamentos restringidos. Os grupos na infraestrutura correspondem à agregação de agentes que possuem papéis que se complementam para realização de uma tarefa mais complexa que exige agentes com diferentes capacidades.

Papéis definidos nesta infraestrutura são caracterizados pelos objetivos que devem ser alcançados pelos agentes, os planos para alcançar tais objetivos, a cardinalidade, compatibilidade e relações de dependência de cada papel.

Na infraestrutura um agente que assume o papel, além de receber o objetivo a ser atingido, recebe o plano para alcançar este objetivo. Dessa forma não é necessário que o agente que deseje assumir determinado papel possua um plano para todos os objetivos que deve alcançar. A cardinalidade se refere ao número máximo de agentes que podem adotar

determinado papel. Um agente pode assumir mais de um papel logo a compatibilidade se refere a quais papéis que um agente pode assumir caso já tenha assumido determinado papel. Por exemplo, um agente não pode assumir o papel de policial e bandido ao mesmo tempo.

Contudo apesar de autônomos, os agentes que adotaram determinado papel estão inseridos em um ambiente comum e dinâmico e dificilmente serão autossuficientes, dependendo de agentes que possuem outros papéis para o alcance de algum de seus objetivos ou ações. Esta infraestrutura permite que sejam definidas as relações de dependência entre os papéis. Por exemplo, é possível definir que um agente que assume um papel de aluno de mestrado depende de um agente que assume um papel orientador para realização de sua pesquisa.

A infraestrutura possui uma organização geral, Figura 3.2, onde estão definidos os papéis da organização. Essa organização geral pode ser particionada sendo criados grupos na organização. Esses grupos são criados com a finalidade de atingir objetivos que necessitem da colaboração e coordenação de agentes com capacidades distintas.

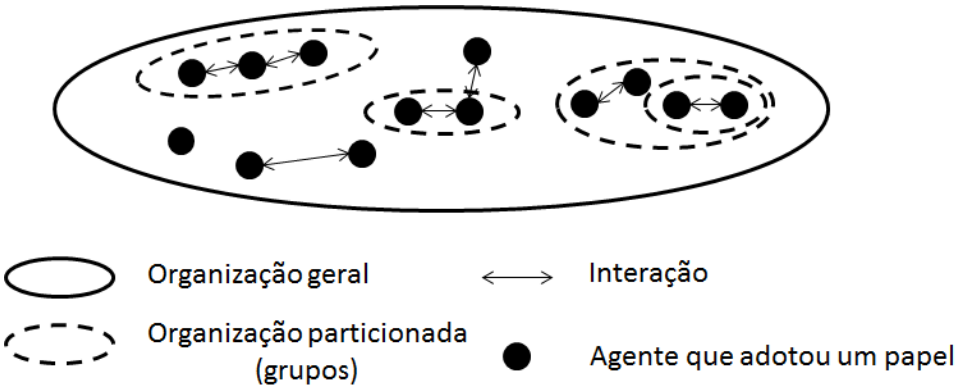


Figura 3.2: Organização geral.

Um grupo possui um papel dentro da organização. Este papel está relacionado ao objetivo global que envolve a colaboração de vários agentes que possuem diferentes papéis. Ou seja, um grupo é formado pela agregação de vários agentes que adotaram determinado papel na organização geral.

O grupo possui um objetivo global decomposto em subobjetivos atribuídos aos agentes com papéis específicos ou atribuídos a grupos com papéis específicos, ou seja, é possível definir que um grupo depende de outro grupo para realização de determinado subobjetivo.

Assim um grupo é caracterizado por um objetivo global que determina seu papel na organização e um conjunto de papéis que podem assumir os subobjetivos do plano global.

3.1.2 Dimensão Funcional

A dimensão funcional é responsável pelo comportamento da organização, ou seja, toda a dinâmica sobre a dimensão estrutural. Na infraestrutura os agentes podem assumir papéis da organização geral e participarem de grupos para realização de objetivos coletivos.

Os papéis possuem objetivos e planos para o alcance desses objetivos. Assim quando um agente assume um papel deve satisfazer os objetivos referentes a este papel. Quando um agente adota um papel, recebe os objetivos e planos desse papel. Um desses objetivos é denominado objetivo inicial. Esse objetivo possui um plano contendo subobjetivos que são os objetivos do papel na ordem devem ser alcançados. Conforme cada subobjetivo é alcançado, o agente tentará atingir o próximo subobjetivo do objetivo inicial.

Dentro da organização geral podem ser criados grupos. Um agente pode participar de um ou mais grupos. Um grupo funciona como um único agente possuindo determinado papel. Um grupo possui um objetivo global. Este objetivo possui um plano composto por subobjetivos que podem ser realizados por agentes ou grupos.

Para auxiliar no entendimento, considere como exemplo uma versão modificada do problema do mundo dos blocos. O problema original considera um conjunto de blocos situados em uma mesa sendo que apenas um bloco pode ser movido por vez. Os blocos podem ser tanto colocados sobre a mesa quanto colocados uns sobre os outros.

Algumas modificações foram feitas sobre o problema original. Aqui, considera-se que os blocos estão fora da mesa, e que existem duas ações a mais que o problema original. Além de sobrepor os blocos, são disponibilizadas as ações de colocar os blocos sobre a mesa e pintar os blocos. Nessa versão modificada, não existe a restrição de apenas um bloco poder ser movido por vez. Além disso, para que os blocos sejam sobrepostos ou pintados os mesmos devem ser colocados sobre a mesa.

Nesta organização existem três tipos de papéis que podem ser assumidos: transportador, empilhador e pintor. O transportador é o responsável por colocar os blocos sobre a mesa, o empilhador é responsável por sobrepor os blocos e o pintor responsável por mudar a coloração dos blocos. Na figura 3.3 é mostrado um exemplo de quando um agente Ag2 assume o papel de empilhador e recebe os objetivos e planos deste papel. Nessa árvore o nodo raiz representa o objetivo inicial que organiza todos os objetivos do papel. Os nodos filhos representam os objetivos do papel. Esses objetivos estão no primeiro nível da árvore como subobjetivos do objetivo inicial. Os nodos folhas representam os passos do plano para alcançar o objetivo desejado.

A organização pode ter um objetivo que necessite de agentes com diferentes papéis para sua realização. Um exemplo seria o objetivo de organizar os blocos. Para tal um grupo pode ser criado para realização deste objetivo global. O grupo é criado por um agente que passa a assumir o papel de gerente do grupo. Esse processo é similar ao que acontece no modelo AALAADIN (FERBER; GUTKNECHT, 1998). O agente gerente quando cria um grupo tem acesso ao plano para realização do objetivo global. Cada passo do plano deve ser atribuído a agentes que possuem o papel especificado ou a um grupo capaz de realizar tal tarefa. O gerente do grupo deve encontrar na organização ge-

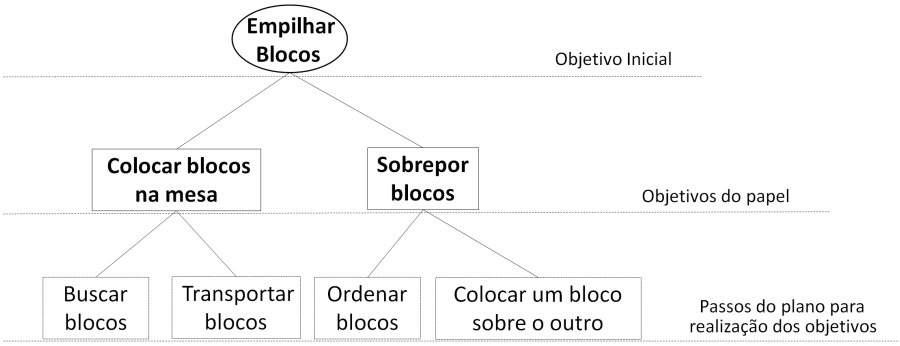


Figura 3.3: Objetivo individual na dimensão funcional.

ral agentes/grupos que possuam os requisitos necessários para realização de algum dos subobjetivos do plano coletivo na organização. Este plano coletivo possui um objetivo global, Figura 3.4, (nodo da árvore) que é decomposto em subobjetivos. Esses subobjetivos podem ser alcançados por agentes que possuem determinado papel, ou por grupos capazes de realizar determinada tarefa.

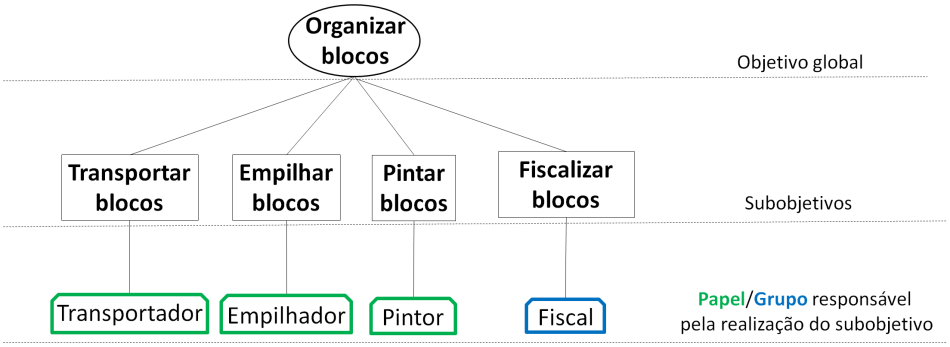


Figura 3.4: Objetivo coletivo na dimensão funcional.

A infraestrutura permite ainda definir objetivos de manutenção para agentes que assumem determinados papéis. Pode ser criado um objetivo de manutenção, no exemplo anterior, para agentes que assumem o papel de pintor possam garantir uma quantidade suficiente de tinta em seu estoque.

Como agentes que adotam determinado papel e grupos podem não serem autossuficientes para realização de suas tarefas é necessário um mecanismo de raciocínio social que permita aos agentes raciocinar sobre a capacidade dos demais agentes e grupos da organização para obtenção de vantagens sobre as habilidades dos mesmos. Esta infraestrutura trabalha apenas com dependências em relação a objetivos. Com esse propósito são definidas duas estruturas que permitem o raciocínio social tanto para grupos quanto para agentes. Essas estruturas são baseadas na descrição externa proposta em (SICHMAN et al., 1994).

A estrutura de dependência, Tabela 3.1, se refere as dependências em relação aos objetivos dos papéis na organização. Essa estrutura é formada por Papel, Agentes, Objetivo

dependente e Passos do plano. Papel refere-se ao papel o qual serão descritas as características em relação às dependências. Agentes refere-se aos agentes que estão ocupando o referente papel num determinado momento. Objetivo dependente refere-se ao objetivo que possui dependências. Passos do plano refere-se as etapas do plano do objetivo dependente, e o papel capaz de realizar tal passo.

Tabela 3.1: Estrutura utilizada para armazenar as dependências em relação aos papéis da organização.

Papel	Agentes	Objetivo dependente	Passos do plano (*passo: responsável*)
transportador	{Ag1, Ag3}	-	-
empilhador	{Ag2, Ag4}	empilharBlocos	colocarBlocosNaMesa: transportador [1] sobreporBlocos: empilhador
pintor	{Ag5, Ag6}	pintarBlocos	colocarBlocosNaMesa: transportador [2] sobreporBlocos: empilhador [3] pintarBloco: pintor

Similar à estrutura de dependência dos papéis da organização, a estrutura de dependência dos grupos, Tabela 3.2, é composta por Papel de grupo, Grupo, Objetivo dependente e Passos do plano. Papel de grupo refere-se a denominação do grupo o qual serão descritas as características em relação às dependências. Grupo refere-se aos grupos criados capazes de realizar os objetivos de determinado papel de grupo. Objetivo dependente refere-se ao objetivo que possui dependências. Passos do plano refere-se as etapas do plano do objetivo dependente, e o papel/grupo capaz de realizar tal passo.

Tabela 3.2: Estrutura utilizada para armazenar as dependências em relação aos grupos da organização.

Papel de grupo	Grupo	Objetivo dependente	Passos do plano (*passo: responsável*)
organizarBlocos	G1	organizarBlocos	transportarBlocos: transportador [4] empilharBlocos: empilhador [5] pintarBlocos: pintor [6] fiscalizarBlocos: fiscal [7]
fiscal	G2	-	-

A infraestrutura trabalha com dois tipos de reputação: reputação de delegação e reputação de normas. Reputação de delegação refere-se a reputação de um agente em relação a

[1]Passo do plano que o agente empilhador é dependente do agente transportador

[2]Passo do plano que o agente pintor é dependente do agente transportador

[3]Passo do plano que o agente pintor é dependente do agente empilhador

[4]Passo do plano coletivo em que o grupo organizarBlocos é dependente de agentes com papel transportador

[5]Passo do plano coletivo em que o grupo organizarBlocos é dependente de agentes com papel empilhador

[6]Passo do plano coletivo em que o grupo organizarBlocos é dependente de agentes com papel pintor

[7]Passo do plano coletivo em que o grupo organizarBlocos é dependente de um grupo com papel de fiscal

adoção de tarefas delegadas. Reputações de normas referem-se a um agente ter cumprido ou não com as normas estabelecidas.

Quando um agente não consegue realizar um objetivo esse verifica se o motivo é em decorrência de uma ação ou objetivo que o agente não consegue realizar sozinho. Caso esse seja o motivo, o agente irá utilizar a estrutura de armazenamento de dependências para verificar de qual papel é dependente e quais agentes possuem este papel. Para escolha do parceiro o agente levará em consideração a reputação do mesmo quanto a adoção de tarefas delegadas.

Para alcançar seus objetivos um agente pode precisar ter acesso a determinados recursos. A utilização desses recursos são regulados por normas. O cumprimento/descumprimento das normas reflete diretamente na reputação do agente quanto ao cumprimento de normas.

Esta infraestrutura permite ainda trabalhar com reorganização de sistemas multiagentes no nível da especificação organizacional. Essa reorganização compreende mudanças na estrutura relacionadas a criação de novos grupos e papéis, mudanças nas relações de dependências, mudanças na parte funcional no tocante a planos individuais e coletivos além de mudanças relacionadas a dimensão normativa. A forma que esse processo de reorganização ocorre acontece de forma emergente onde não existe um controle explicito do sistema em relação ao processo de reorganização. A reorganização acontece por iniciativa de um agente não existindo uma reorganização institucionalizada. Esse agente depende da aceitação de um agente organizacional para que a iniciativa de modificar a organização seja bem sucedida.

3.1.3 Dimensão Dialógica

A dimensão dialógica define os protocolos de comunicação utilizados para interação entre os agentes. Na infraestrutura protocolos são definidos para comunicação dos agentes para formação dos grupos e também em decorrência das relações de dependência.

Quando um grupo é criado o agente gerente do grupo precisa buscar agentes ou grupos que realizem partes da tarefa coletiva. Com esse objetivo é utilizado o protocolo de contrato de rede (*Contract Net Protocol*) (SMITH, 1980) (FIPA, 2002) para determinar qual agente/grupo que será responsável determinada parte do plano coletivo.

Na busca por agentes, o gerente do grupo verifica quais agentes estão ocupando determinado papel que é capaz de realizar determinada tarefa no plano coletivo. Após essa etapa o agente gerente solicita a realização das tarefas para os agentes selecionados. Os agentes selecionados enviam ao gerente uma média entre suas reputações de delegação e de cumprimento de normas. Ao receber as respostas o gerente do grupo avalia qual dos agentes possui maior média das reputações e informa a escolha ao agente selecionado.

Na busca por grupos, o gerente do grupo solicitante verifica quais grupos são capazes de realizar determinada tarefa no plano coletivo. O gerente então solicita aos gerentes desses grupos a realização da tarefa desejada. O gerente do grupo capaz de realizar a tarefa desejada envia ao gerente solicitante uma proposta contendo o tempo até que o grupo

possa começar a executar a tarefa desejada. Ao receber as respostas o gerente do grupo solicitante avalia qual dos grupos iniciará a execução da tarefa no menor tempo e informa a escolha ao gerente desse grupo.

Quando um agente que adotou determinado papel não consegue realizar alguma tarefa de seu plano individual, por motivos de dependência, é utilizado o protocolo Request (FIPA, 2006). Neste protocolo as tarefas a serem realizadas são requisitadas pelos agentes solicitantes e a resposta em relação ao comprometimento com a tarefa delegada assim como o resultado é informado pelos destinatários.

3.1.4 Dimensão Normativa

Normas podem ser utilizadas para orientar, regular ou controlar um comportamento. Assim a dimensão normativa pode ser usada para regulamentar os elementos funcionais, estruturais e dialógicos. As normas podem ser usadas na dimensão estrutural, por exemplo, para regulamentar quais papéis que podem ser assumidos ao mesmo tempo pelos agentes assim como quais papéis que podem criar determinados grupos. Na dimensão funcional normas podem ser utilizadas, por exemplo, restringir determinadas ações que os agentes podem executar. Na dimensão dialógica normas podem ser usadas, por exemplo, para controlar os padrões de interação entre os agentes.

Nesta infraestrutura as normas serão utilizadas para controlar quais papéis podem criar quais grupos e também para controlar o acesso aos recursos utilizados pelos agentes para o alcance de seus objetivos. Os recursos são os artefatos organizacionais e os artefatos de ambiente que auxiliam os agentes na realização de suas tarefas individuais e coletivas.

3.2 Etapa de Modelagem dos Artefatos

A infraestrutura é composta por artefatos organizacionais que incorporam as dimensões definidas na Seção 3.1. O conjunto de artefatos organizacionais disponíveis no ambiente provem operações e propriedades observáveis para os agentes facilitando a interação com a organização. Esses artefatos permitem que os agentes adotem papéis, participem de grupos para realização de tarefas coletivas, busquem as dependências individuais e de grupos, atualizem e percebam modificações nas normas e utilizem a reputação como auxílio no processo de parcerias, formação de grupos e imposição de normas.

Como já mencionado anteriormente a infraestrutura é baseada no metamodelo de Agente e Artefatos A&A (OMICINI; RICCI; VIROLI, 2008) e no conceito de organizações incorporadas (*Embodied Organizations*) (Seção 2.3) sendo composta de artefatos organizacionais e artefatos de comunicação. Artefatos de ambiente específicos da aplicação podem ser criados e podem interagir com os artefatos organizacionais. Como exemplo, podem ser criados artefatos de ambiente que representem algum recurso que podem ter seu acesso controlado pela utilização de normas.

Primeiramente será apresentada uma visão geral da infraestrutura com todos os artefatos presentes, a partir disso são apresentadas as características gerais de cada artefato, suas operações e se existente as ligações com os demais artefatos organizacionais. Como

mostra a Figura 3.5, a infraestrutura é composta pelos seguintes artefatos: GeneralORG, Roles, Group (um para cada grupo criado), Norm, RoleDEP, GroupDEP, DelegationREP, NormREP, FIPAContractNET e FIPARequest.

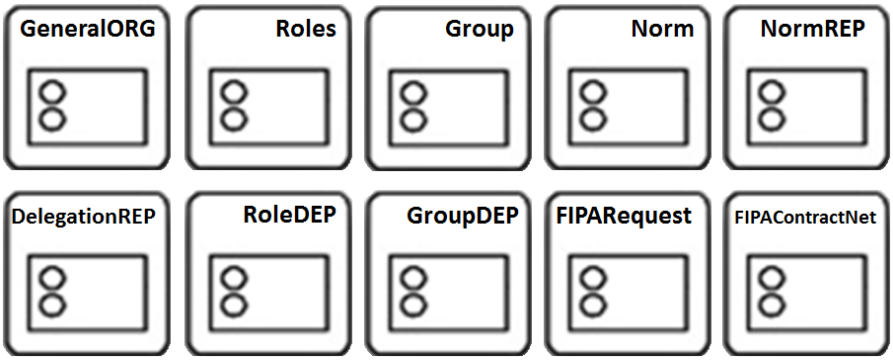


Figura 3.5: Artefatos presentes na infraestrutura.

Alguns artefatos possuem interfaces de ligação com outros artefatos permitindo a interação entre os mesmos com intuito de compartilhar algumas informações necessárias para o correto funcionamento da infraestrutura. Juntamente com a descrição dos artefatos, apresentadas nas próximas seções, serão apresentadas também suas ligações com os demais artefatos organizacionais.

A infraestrutura possui um agente com papel organizacional. Este agente não é definido através da dimensão estrutural, ele já está implicitamente definido na organização. Este é o agente responsável por criar os artefatos da infraestrutura que estarão disponíveis aos demais agentes. Além disso, esse agente tem como funções fiscalizar o cumprimento de normas e aceitar ou não o processo de reorganização solicitado por um agente.

3.2.1 Artefato GeneralORG

O primeiro artefato a ser apresentado é o GeneralORG. Este artefato representa a organização geral descrita na Subseção 3.1.1. Este é o artefato que permite que os agentes adotem e deixem papéis, crie e finalize grupos, e modifiquem a organização a partir de um processo de reorganização.

GeneralORG é um artefato que funciona como uma ponte entre o agente que deseja participar da organização com a especificação organizacional. Utilizando as operações disponíveis do artefato o agente pode verificar os papéis existentes e então adotar um papel nessa organização.

Cada agente que adota um papel passa a fazer parte da organização geral, assim como cada grupo criado passa a ser um grupo que faz parte da organização geral. Um agente, quando adota um papel, executa a operação sobre o artefato GeneralORG que por sua vez realiza a operação de adoção de papel no artefato Roles. O artefato GeneralORG também é responsável por atualizar a informação sobre um novo membro da organização nos artefatos de reputação de normas (NormRep) e reputação de delegação (DelegationRep).

Ou seja, o artefato GeneralORG funciona também como uma interface para os agentes que desejam entrar na organização com os demais artefatos que compõem a organização. Quando um grupo é criado este artefato cria um novo artefato para esse grupo atribuindo o agente que solicitou a criação como gerente do grupo.

A representação do artefato GeneralOrg é mostrada na Figura 3.6. A interface de uso do artefato possui as seguintes operações:

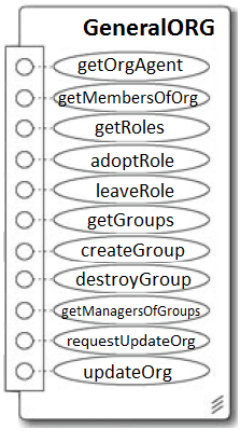


Figura 3.6: Representação do artefato GeneralOrg.

- `getOrgAgent`: Operação responsável por retornar o agente organizacional.
- `getMembersOfOrg`: Esta operação retorna todos os agentes que fazem parte da organização, ou seja, todos os agentes que adotaram algum papel na organização.
- `getRoles`: Esta operação é utilizada para verificar quais os papéis existentes na organização.
- `adoptRole`: Operação utilizada por um agente que deseja fazer parte da organização para adotar um papel.
- `leaveRole`: Operação utilizada por um agente para deixar um papel na organização.
- `getGroups`: Operação que retorna todos os grupos criados na organização.
- `createGroup`: Operação utilizada por um agente para criar um grupo. O agente que cria o grupo passa ser o gerente do mesmo.
- `destroyGroup`: Operação utilizada para finalizar um grupo.
- `getManagersOfGroups`: Operação utilizada para buscar os agentes responsáveis pela gerencia de um grupo com determinado papel.
- `requestUpdateOrg`: Operação utilizada por um agente para solicitar modificações na organização atual.
- `updateOrg`: Operação utilizada pelo agente organizacional para modificar a organização atual.

3.2.2 Artefato Roles

O artefato Roles é o responsável por toda dinâmica decorrente da adoção de um papel pelo agente. Na infraestrutura um agente quando adota um papel, recebe os objetivos referentes ao papel e os planos para alcançar esses objetivos. Além disso, como mostrado na Seção 3.1.2 um desses objetivos pode ser denominado objetivo inicial. O objetivo inicial possui um plano contendo os subobjetivos que são os objetivos do papel na ordem devem ser alcançados.

Este artefato é o responsável além das operações para adotar e deixar um papel também pelo envio dos objetivos e respectivos planos do papel adotado pelos agentes. O artefato possui uma estrutura que armazena quais agentes que estão adotando quais papéis da organização em determinado momento.

Os agentes não adotam ou deixam papéis diretamente sobre este artefato. Estas operações são realizadas sobre o artefato GeneralOrg e este por sua vez realiza essas operações sobre o artefato Roles utilizando a interface de ligação entre os mesmos. Essa operações não são diretamente realizadas sobre o artefato Roles, pois quando agentes entram ou deixam a organização outros artefatos são atualizados a respeito dessas informações. O responsável por essas atualizações é então o artefato GeneralOrg.

A representação do artefato Roles é mostrada na Figura 3.7. A interface de uso do artefato possui as seguintes operações:

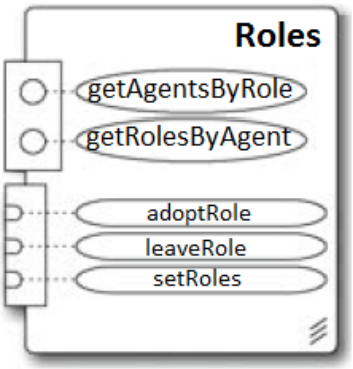


Figura 3.7: Representação do artefato Roles.

- `getAgentsByRole`: Esta operação retorna todos os agentes que estão adotando determinado papel. A operação recebe como parâmetro o papel que se deseja conhecer os agentes.
- `getRolesByAgent`: Esta operação é o inverso da anterior. Recebe como parâmetro um agente e retorna todos os papéis adotados por este agente.

O artefato possui as seguintes operações de ligação com os demais artefatos:

- `adoptRole`: Esta é a operação executada pelo artefato GeneralOrg quando um agente deseja adotar um papel na organização.

- `leaveRole`: Esta é a operação executada pelo artefato GeneralOrg quando um agente deseja deixar um papel na organização.
- `setRoles`: Esta é a operação executada pelo artefato GeneralOrg quando ocorre o processo de reorganização. Esta operação atualiza na estrutura do artefato,que armazena os papéis da organização e os agentes que estão adotando esses papéis em determinado momento, os novos papéis que foram inseridos na organização.

3.2.3 Artefato Group

Um artefato Group é criado para cada grupo que é criado na organização. Este artefato é criado a partir do artefato GeneralOrg quando os agente executam a operação para criação de um grupo.

O agente que solicita a criação de um grupo é nomeado como o gerente deste grupo. O artefato Group auxilia o agente gerente na busca por agentes ou grupos capazes de realizar determinadas tarefas do plano coletivo. Este artefato auxilia ainda no processo de coordenação na execução do plano. O artefato possui uma estrutura que armazena que agente(s)/grupo(s) são responsáveis pela execução de determinada tarefa do plano coletivo.

A representação do artefato Group é mostrada na Figura 3.8. A interface de uso do artefato possui as seguintes operações:

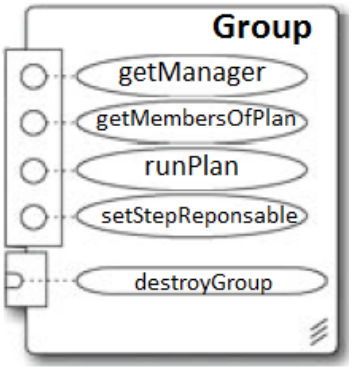


Figura 3.8: Representação do artefato Groups.

- `getManager`: Operação que permite buscar o agente gerente do grupo.
- `getMembersOfPlan`: Operação que retorna ao agente gerente quais papéis/grupos são os responsáveis pela execução de cada parte do plano coletivo. Com essa informação o agente gerente pode buscar quais agentes/grupos, que possuam determinado papel responsável pela execução de determinada parte do plano, podem fazer parte do grupo.
- `setStepResponsable`: Quando o agente gerente escolhe o agente/grupo com determinado papel, responsável por determinada parte do plano, utiliza essa operação para armazenar na estrutura presente na infraestrutura os responsáveis pelas etapas do plano coletivo.

– `runPlan`: Quando o agente gerente conseguiu buscar todos os agentes/grupos com determinado papel para realização de todas as etapas do plano, executa essa operação sobre o artefato. Essa operação é responsável por auxiliar o gerente na coordenação da execução do plano coletivo.

O artefato possui uma operação de ligação com os demais artefatos:

– `destroyGroup`: Esta é a operação executada pelo artefato GeneralOrg quando o agente gerente deseja finalizar a formação de um grupo.

3.2.4 Artefatos de Dependência

A infraestrutura possui dois artefatos de dependência: RoleDEP e GroupDEP. RoleDEP contém a estrutura descrita na Tabela 3.1 da Seção 3.1.2. Já o artefato GroupDEP contém a estrutura Tabela 3.2 da Subseção 3.1.2. Ambos os artefatos auxiliam os agentes na busca por parceiros quando estes não são autossuficientes para execução e algum de seus objetivos.

A representação do artefato RoleDEP é mostrada na Figura 3.9. A interface de uso do artefato possui as seguintes operações:

Figura 3.9: Representação do artefato RoleDEP.

– `getRoleDependence`: Operação que recebe o objetivo para o qual o agente é dependente como parâmetro e retorna os papéis que o agente depende para cada passo do plano que o agente não seja autossuficiente.

– `getPartner`: Operação que recebe como parâmetro o passo do plano que o agente é dependente e retorna o parceiro com a melhor reputação de delegação para realizar a tarefa.

A representação do artefato GroupDEP é mostrada na Figura 3.10. A interface de uso do artefato possui a seguinte operação:

– `getGroupDependence`: Operação que recebe como parâmetro o plano coletivo que um grupo é dependente e retorna o grupo capaz de realizar este objetivo.

Figura 3.10: Representação do artefato GroupDEP.

3.2.5 Artefato Norm

O artefato Norm permite que sejam criadas normas para regular algum comportamentos dos agentes que adotaram determinado papel. Na infraestrutura as normas são utilizadas para controlar quais papéis podem criar grupos e também controlar o acesso dos agentes aos recursos. Os recursos são considerados os artefatos e o controle está relacionado ao acesso às operações providas pelos artefatos. Ou seja, existem normas que regulam quais operações podem ser executadas por quais agentes na organização.

O artefato Norms é uma modificação do artefato de normas desenvolvido em (SANTOS; ROCHA COSTA, 2012). Neste artefato as normas podem ser criadas, alteradas ou removidas. A criação de normas pode acontecer a partir de um agente ou a partir da especificação das normas na dimensão normativa da especificação organizacional. A representação do artefato Norm é mostrada na Figura 3.11.

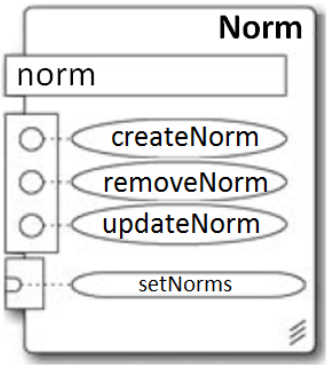


Figura 3.11: Representação do artefato Norm.

O artefato possui a propriedade observável `norm`, Figura 3.12, que contém os seguintes argumentos:

```
norm(type, operation, action, roles, punishment/reward)
```

Figura 3.12: Propriedade observável `norm`.

- `type`: Refere-se ao tipo da norma (permissão, obrigação,...).
- `operation`: Refere-se a operação do artefato que tem seu acesso controlado.

- `action`: Refere-se a ação que deve ser realizada caso a norma seja violada.
- `roles`: Refere-se aos papéis que podem realizar a operação do artefato que é controlada.
- `punishment/reward`: Refere-se a punição/recompensa aplicada quando a norma é violada ou cumprida.

A interface de uso do artefato possui as seguintes operações:

- `createNorm`: Operação que permite criar uma nova norma.
- `removeNorm`: Operação que permite remover uma norma existente.
- `updateNorm`: operação que permite alterar uma norma existente.

O artefato possui uma operação de ligação com os demais artefatos:

- `setNorms`: Esta é a operação executada pelo artefato GeneralOrg quando ocorre o processo de reorganização. Esta operação cria novas normas ou altera normas já existentes a partir da nova especificação organizacional.

3.2.6 Artefatos de Comunicação

Para comunicação dos agentes foi utilizada uma infraestrutura de comunicação baseada em artefatos desenvolvida por Rodrigues, Rocha Costa e Dimuro (2013). Essa infraestrutura de comunicação é integrada a infraestrutura OrIAs permitindo que a mesma incorpore os aspectos da dimensão dialógica da infraestrutura.

A infraestrutura de comunicação foi originalmente proposta para plataforma JaCaMo (BOISSIER et al., 2011). Contudo a infraestrutura utiliza artefatos para mediar a comunicação entre os agentes implementados utilizando a plataforma CArtAgO não sendo uma infraestrutura intrinsicamente ligada a plataforma JaCaMo. Dessa forma, a infraestrutura pode facilmente ser integrada a infraestrutura OrIAs.

Em função dos artefatos serem usados como mediadores da comunicação, é possível especificar de forma modular as interações do sistema encapsulando as mesmas em artefatos. A criação de novos protocolos e meios de comunicação se torna possível a partir da implementação dos respectivos artefatos.

A infraestrutura de comunicação é constituída de artefatos divididos em dois grupos: artefatos de fala e artefatos de protocolo. Artefatos de fala executam atos de fala simples e são utilizados simplesmente para o envio de mensagens. Esses protocolos gerenciam conversas em andamento que usam artefatos deste tipo, armazenando os participantes das conversas assim como a fila de mensagens de cada um. Artefatos de protocolo são responsáveis pelo encapsulamento de algum protocolo de comunicação. Esse artefato concentra os detalhes envolvidos em conversas baseadas em protocolo como o autômato que controla o fluxo de mensagens do protocolo e o estado da conversa de cada agente envolvido. Dessa forma, enquanto os artefatos de fala definem performativas de comunicação e são

utilizados apenas para enviar mensagens, os artefatos de protocolo encapsulam a lógica de algum protocolo e são utilizados para definir uma sequencia autorizada de mensagens a serem enviadas.

A comunicação entre os agentes ocorre a partir da execução das operações disponíveis nos artefatos. Os artefatos têm como função encaminhar as mensagens recebidas aos respectivos destinatários além de fiscalizar a ordem de execução do envio caso algum protocolo esteja sendo utilizado.

As operações nos artefatos de comunicação possuem o mesmo nome das performativas correspondentes. Por exemplo, uma operação denominada *inform* envia uma mensagem do tipo *inform*.

SpeechActArtifact e *ProtocolArtifact* 3.13, Figura 3.13, são os artefatos base da infraestrutura. A partir da extensão das classes correspondentes a esses artefatos podem ser criados os artefatos de comunicação utilizados pelos agentes.

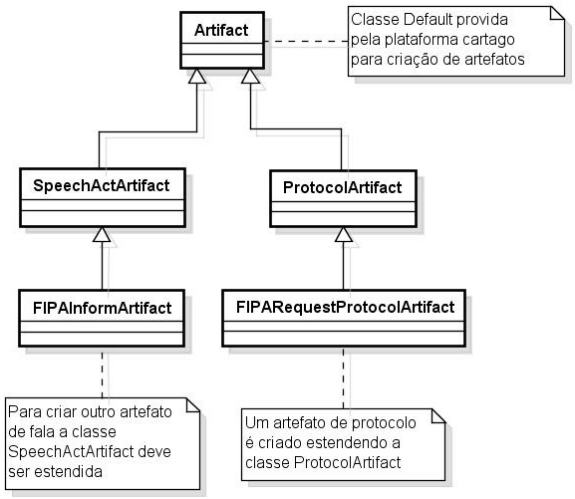


Figura 3.13: Diagrama que permite ilustrar como novos artefatos de comunicação podem ser criados , adaptado de Rodrigues, Rocha Costa e Dimuro (2013).

O artefato `SpeechActArtifact` é o responsável por definir o comportamento básico de um artefato de fala, enquanto o artefato `ProtocolArtifact` é responsável por definir o comportamento de um artefato de protocolo.

Para criar um artefato de atos de fala deve ser criada uma classe que estenda a classe `SpeechActArtifact` e deve ser implementada apenas a operação que executa este ato de fala. Por exemplo, para um artefato que implemente a performativa *inform* deve ser implementada apenas a operação relativa a essa performativa.

A criação de um artefato de protocolo é similar à criação de um artefato de atos de fala. Uma classe estendendo a classe `ProtocolArtifact` deve ser criada, assim como o autômato relativo ao fluxo de execução do protocolo. Todas as operações que enviam mensagens possuem o parâmetro de identificação da conversa. Além disso, é mantido um

registro das conversas em andamento.

A infraestrutura de comunicação permite então que sejam criados os protocolos desejados e suas respectivas performativas de comunicação desde que sejam criados os artefatos correspondentes. Nessa infraestrutura já estão implementados os protocolos Request e ContractNet como mostrados no diagrama da Figura 3.14.

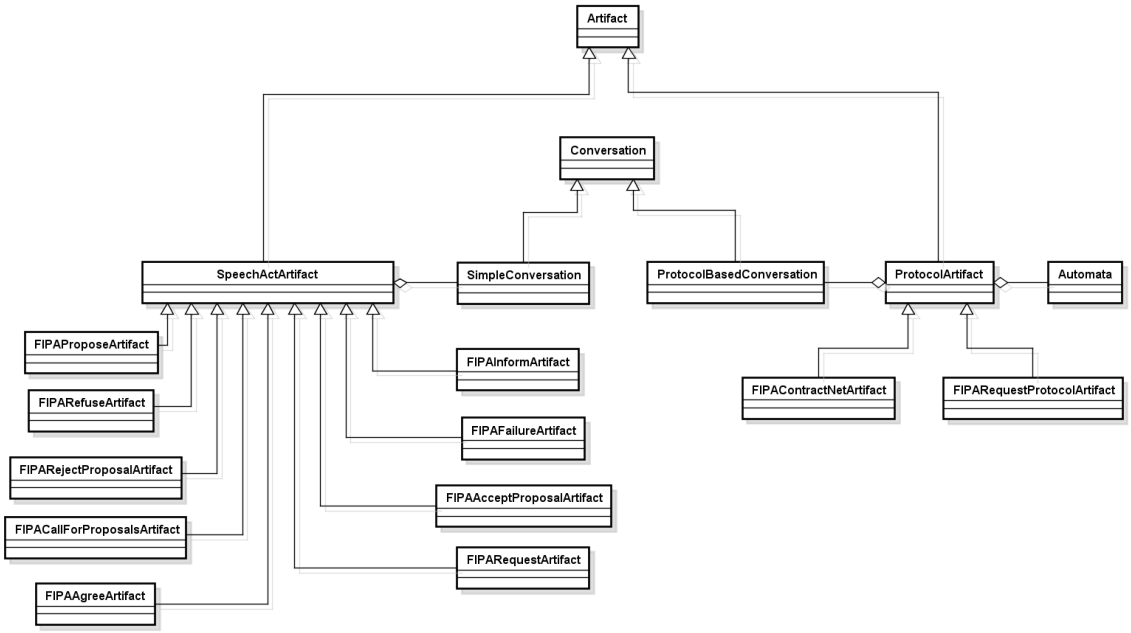


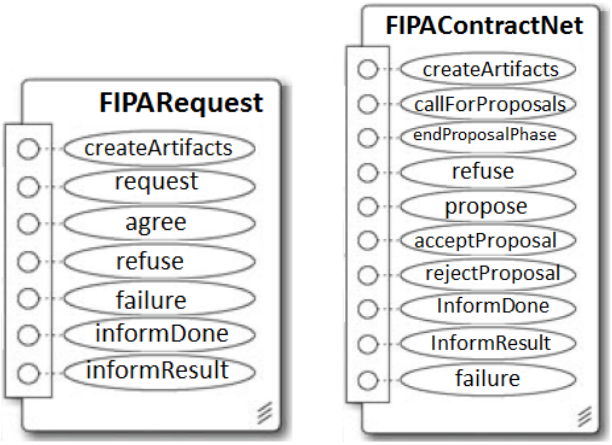
Figura 3.14: Digrama de classe referente aos protocolos *Request* e *ContractNet*.

A infraestrutura OrIAs utiliza o protocolo Request para comunicação dos agentes quando os mesmo são dependente e o protocolo ContractNet para formação de grupos. Abstraindo os demais artefatos e classes que compõem a infraestrutura de comunicação, a representação dos artefatos utilizados diretamente pelos agentes para comunicação na infraestrutura são: o FIPARequest, e o FIPAContractNet, Figura 3.15.

A interface de uso dos artefatos possuem em comum a operação `createArtifacts`, essa operação é executada pelo agente organizacional depois da criação dos artefatos no *wokspace*. Esse operação tem como função criar os artefatos de atos de fala das performativas do respectivo protocolo.

Quando a operação `createArtifacts` do artefato FIPARequest é executada são criados os artefatos de atos de fala: Request, Agree, Refuse, Failure, InformDone e InformeResult. O mesmo ocorre para criação dos artefatos de atos de fala correspondentes as performativas do protocolo *Contract Net* com a execução da operação `createArtifacts` no artefato FIPAContractNet.

As operações de ambos os artefatos correspondem a tipos de mensagens enviadas seguindo o fluxo de mensagens do protocolo. Cada operação tem como nome o tipo de mensagem enviada. O significado de cada tipo de mensagem dentro do protocolo é



(a) Representação do artefato FIPARequest.

(b) Representação do artefato FIPAContractNet.

Figura 3.15: Representação dos artefatos de comunicação acessados diretamente pelos agentes.

mesmo do já explicado nas Subseções 2.7.3.1 e 2.7.3.2.

3.2.7 Artefatos de Reputação

A infraestrutura utiliza um mecanismo de reputação para auxiliar os agentes tanto na busca de parceiros para o alcance dos objetivos individuais e coletivos como também para auxiliar no processo de aplicação das normas. O modelo trabalha com dois tipos de reputação: reputação quanto ao cumprimento de normas e reputação de delegação de tarefas. O valor para as reputações dos agentes varia de 0.0 a 1.0.

O projeto dos artefatos de reputação é baseado no modelo cognitivo de reputação de Conte e Paolucci apresentado na Seção 2.5. Seguindo este modelo, quando um agente Ag1 delega uma tarefa para um agente Ag2 e recebe o resultado dessa delegação, Ag1 forma uma imagem sobre Ag2. Se o resultado da tarefa delegada foi de acordo com o esperado pelo agente Ag1 uma imagem positiva em relação a Ag2 é formada, caso contrário é formada uma imagem negativa. Quando um agente Ag3 realiza uma operação sobre um artefato que é regulado por normas, o agente organizacional observa que essa operação foi realizada e consegue detectar se o agente violou ou não determinada norma. A partir dessa observação o agente organizacional pode formar uma imagem positiva ou negativa em relação ao agente Ag3. Se o agente Ag1 ou o agente organizacional decide compartilhar essa imagem com os demais membros da organização são utilizados respectivamente os artefatos de: reputação de delegação e reputação quanto ao cumprimento de normas.

Para reputação quanto ao cumprimento de normas é criado o artefato NormREP. Este artefato contém uma estrutura que armazena os agentes membros da organização e o respectivo valor de sua reputação. A Tabela 3.3 mostra um exemplo de como ficaria estrutura utilizada para o armazenamento das reputações para um determinado estudo de caso. As reputações dos agentes no artefato NormREP levam em consideração o cumprimento/-descumprimento das normas por parte dos agentes.

Tabela 3.3: Estrutura utilizada para armazenar a reputação quanto ao cumprimento de normas dos agentes.

Agente	Reputação quanto ao cumprimento de normas
Ag1	0.5
Ag2	0.4
Ag3	0.6
Ag4	0.8

Através do artefato Norm os agentes recebem as normas e sabem quais papéis que podem criar determinados grupos e acessar determinados recursos. Através da propriedade observável `norm` os agentes sabem também os valores de aumento/diminuição da reputação pelo parâmetro `punishment/reward`. Assim no cálculo de reputação utilizado, quando um agente cumpre com uma norma, sua reputação quanto ao cumprimento de normas aumenta o valor do parâmetro `punishment/reward`, caso a norma seja descumprida sua reputação diminui o valor do parâmetro `punishment/reward`.

Para o cálculo do aumento da reputação temos:

$$reputação = reputaçãoAtual + punishment/reward \tag{3.1}$$

Para o cálculo da diminuição da reputação temos:

$$reputação = reputaçãoAtual - punishment/reward \tag{3.2}$$

A representação do artefato NormREP é mostrada na Figura 3.16.

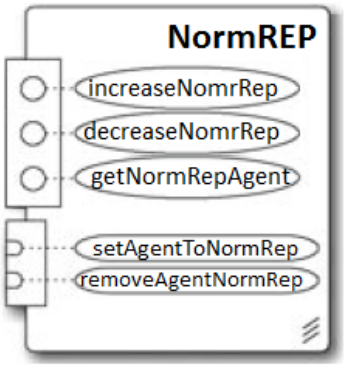


Figura 3.16: Representação do artefato NormREP.

A interface de uso do artefato possui as seguintes operações:

- `increaseNomrRep`: Operação executada pelo agente organizacional para aumentar o valor da reputação de um agente.
- `decreaseNomrRep`: Operação executada pelo agente organizacional para diminuir o valor da reputação de um agente.
- `getNormRepAgent`: Operação utilizada pelos agentes na organização para buscar o valor de reputação quanto ao cumprimento de normas de um agente.

O artefato possui as seguintes operações de ligação com os demais artefatos:

- `setAgentToNormRep`: Esta é a operação executada pelo artefato GeneralOrg para inserir um novo agente na estrutura que armazena as reputações, quando esse agente passa a fazer parte da organização.
- `removeAgentNormRep`: Esta é a operação executada pelo artefato GeneralOrg pa atualizar a estrutura que armazena as reputações quando um agente deixa um papel saindo da organização.

Para reputação quanto ao cumprimento de tarefas delegadas é criado o artefato DelegationREP. A reputação quanto à delegação é subdividida ainda em dois tipos de reputação: reputação de delegação quanto ao cumprimento do protocolo e reputação de delegação quanto ao resultado da tarefa delegada. Este artefato contém uma estrutura que armazena os dois tipos de reputação de delegação para cada papel que o agente ocupa. Ou seja, se uma agente Ag1 ocupa o papel de pintor e também o papel de empilhador ele terá suas reputações de delegação para cada papel. A Tabela 3.4 mostra um exemplo de como ficaria estrutura utilizada em para o armazenamento das reputações para um determinado estudo de caso.

Tabela 3.4: Estrutura utilizada para armazenar a reputação de delegação dos agentes.

Papel	Reputação de delegação		
	Agente	Cumprimento protocolo	Resultado da tarefa delegada
Pintor	Ag1	0.4	1.0
	Ag2	0.3	0.5
Transportador	Ag3	0.5	1.0
	Ag4	0.7	0.8
Empilhador	Ag5	0.5	1.0
	Ag1	0.6	0.9

Quando um agente depende de outro para realização de um objetivo que não é capaz de realizar sozinho precisa solicitar a realização desse objetivo para um agente que possa realizar. Para tal, o agente dependente precisa se comunicar com o agente capaz de realizar a tarefa. Para a comunicação dos agentes em situação de dependência é estabelecido um protocolo. Na infraestrutura, o protocolo utilizado para que os agentes se comuniquem é o Request. O protocolo possui fluxo no qual as mensagens devem seguir uma

determinada ordem. Utilizando o Request um agente dependente deve primeiro solicitar o alcance de um objetivo para outro agente. O agente que recebe a solicitação deve aceitar ou rejeitar e por fim informar o resultado.

A troca de mensagens é realizada utilizando os artefatos de comunicação apresentados na Seção 3.2.6. Nessa infraestrutura, os artefatos de protocolo possuem um autômato que controla a ordem de envio dessas mensagens. Caso o agente tente enviar uma mensagem fora de ordem o artefato do protocolo detecta este erro. Quando um agente não respeita a ordem das mensagens de acordo com o protocolo, o artefato responsável pelo protocolo Request executa uma operação sobre o artefato DelegationREP que decrementa a reputação do agente quanto ao cumprimento do protocolo. Caso o agente capaz de realizar a tarefa respeite o fluxo do protocolo a reputação de delegação quanto ao cumprimento do protocolo do agente é aumentada.

Quando um agente dependente solicita a outro a execução de uma tarefa o resultado dessa delegação pode não ser o que o agente dependente esperava. Se o resultado da execução da tarefa delegada não é satisfatório a reputação do agente que realizou a tarefa quanto ao resultado da tarefa delegada diminui, caso contrário a reputação aumenta.

O cálculo utilizado para o aumento das reputações de delegação de protocolo é o mesmo utilizado para o aumento/diminuição do valor de reputação quanto ao cumprimento de normas (Equações 3.1 e 3.2), porém é fixado o valor 0.1 para o aumento/diminuição do valor de reputação dos agentes. A representação do artefato DelegationREP é mostrada na Figura 3.17.

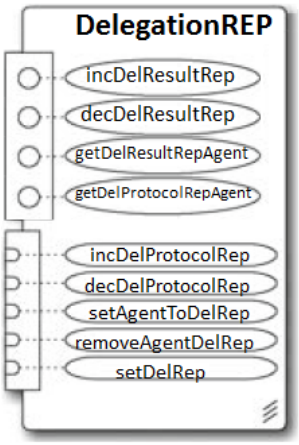


Figura 3.17: Representação do artefato DelegationREP.

A interface de uso do artefato possui as seguintes operações:

- `incDelResultRep`: Operação que aumenta a reputação do agente em relação ao resultado de uma tarefa delegada.
- `decDelResultRep`: Operação que diminui a reputação do agente em relação ao resultado de uma tarefa delegada.

- `getDelResultRepAgent`: Operação que retorna a reputação de um agente em relação ao resultado de uma tarefa delegada.

- `getDelProtocolRepAgent`: Operação que retorna a reputação de um agente em relação ao cumprimento do protocolo.

O artefato possui as seguintes operações de ligação com os demais artefatos:

- `incDelProtocolRep`: Operação realizada pelo artefato FIPARequest quando o agente cumpre corretamente o fluxo de mensagens do protocolo.

- `decDelProtocolRep`: Operação realizada pelo artefato FIPARequest quando o agente não cumpre o fluxo de mensagens do protocolo.

- `setAgentToDelRep`: Operação executada pelo artefato GeneralOrg que atualiza a estrutura que armazena os agentes com suas respectivas reputações de delegação. Executada quando um agente adota um papel na organização.

- `removeAgentDelRep`: Operação executada pelo artefato GeneralOrg que atualiza a estrutura de reputação dos artefatos quando um agente deixa um papel na organização.

- `setDelRep`: Esta é a operação executada pelo artefato GeneralOrg quando ocorre o processo de reorganização. Esta operação insere novos papéis na estrutura que armazena os papéis, os agentes que adotaram os papéis e sua respectiva reputação.

3.3 Etapa de Descrição dos Serviços

Esta seção apresenta a etapa de modelagem dos serviços. Nesta etapa é mostrada toda dinâmica de funcionamento da infraestrutura sob a perspectiva dos serviços disponibilizados pelos artefatos.

Para um melhor entendimento do funcionamento da infraestrutura, assim como da ligação entre os artefatos, são mostradas os eventos que ocorrem quando agente realiza alguma operação sobre determinado artefato. A dinâmica de funcionamento é mostrada tanto do ponto de vista dos agentes, ou seja, o que os eventos disparados interferem no processo cognitivo dos agentes quanto do ponto de vista do funcionamento interno dos artefatos quando a operação é executada. Para representação dos aspectos dinâmicos da infraestrutura são utilizados diagramas de atividades definido pela Linguagem de Modelagem Unificada (UML) (BOOCH; RUMBAUGH; JACOBSON, 2006).

Primeiramente é mostrado na Figura 3.18 o processo quando um agente adota um papel. Para adotar um papel um agente ag1 deve executar a operação `adoptRole` sobre o artefato GeneralOrg que por sua vez executa a operação sobre o artefato Roles. Ao adotar um papel o agente ag1 passa a fazer parte da organização geral.

A Figura 3.19 mostra o diagrama de atividades do artefato Role quando um agente adota um papel.

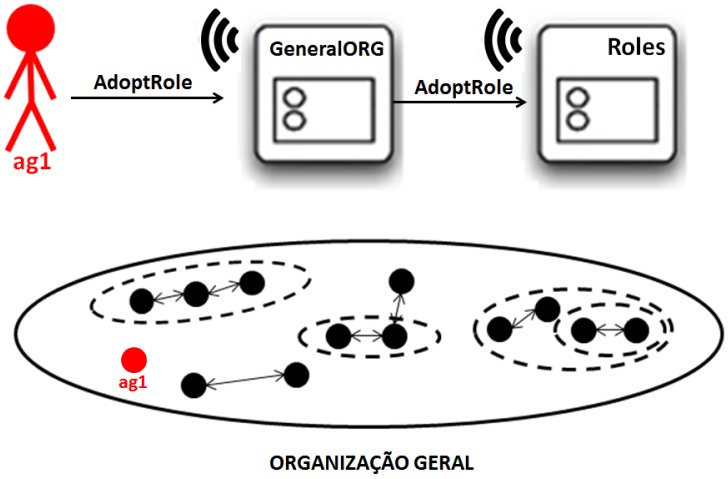


Figura 3.18: Processo de adoção um papel por um agente na organização.

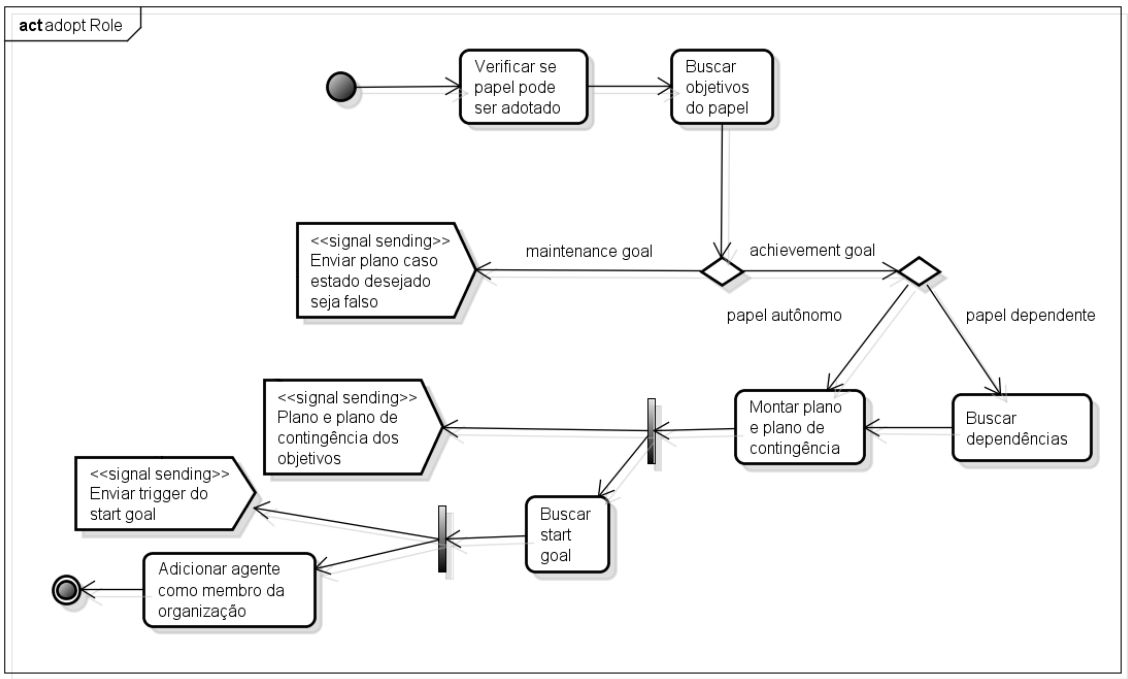


Figura 3.19: Diagrama de atividades do ponto de vista do artefato quando um agente adota um papel.

Quando um agente adota um papel, primeiramente é verificado se o papel pode ser adotado. Um papel pode ser adotado se está disponível, se ainda não foi adotado pelo agente, e, caso o agente já tenha adotado outros papéis na organização, o papel que o agente deseja adotar deve ser compatível com os demais papéis adotados. Satisfeitas as restrições no próximo passo o artefato Roles irá buscar os objetivos referentes ao papel.

Caso o objetivo do papel seja do tipo *maintenance* é enviado um plano para o agente através de um sinal. Este plano é executado sempre que a crença relativa a um estado de mundo desejado varia. Este plano verifica se as condições de mundo desejado são

verdadeiras. Caso o estado de mundo desejado se torne falso o agente deve alcançar um objetivo de manutenção com intuito de manter o estado de mundo almejado.

Caso o objetivo do papel seja do tipo *achievement* primeiramente é verificado se o agente é autossuficiente para alcançar o objetivo. Caso o agente seja autossuficiente são buscados os passos do plano sendo montado um plano para alcançar o objetivo e um plano alternativo caso o plano falhe. Esses planos são montados e enviados através de sinais para os agentes. Caso o agente seja dependente para realização de alguma das etapas do plano, primeiramente são verificadas quais etapas que o agente é dependente. Realizado esse passo, é montado o plano. Para cada etapa que o agente é dependente são acrescentados passos que o agente deve seguir para satisfazer as dependências. A Figura 3.20 mostra como são acrescidos passos ao plano original quando é constatado que o agente é dependente para alcançar determinado objetivo. Esses passos incluem ações sobre o artefato RoleDEP para busca de parceiros e ações sobre o artefato de FIPARequest para solicitar auxílio de outros agentes para realização das tarefas. Tendo o plano e o plano alternativo sido montados, os mesmos são enviados através de sinais para os agentes.

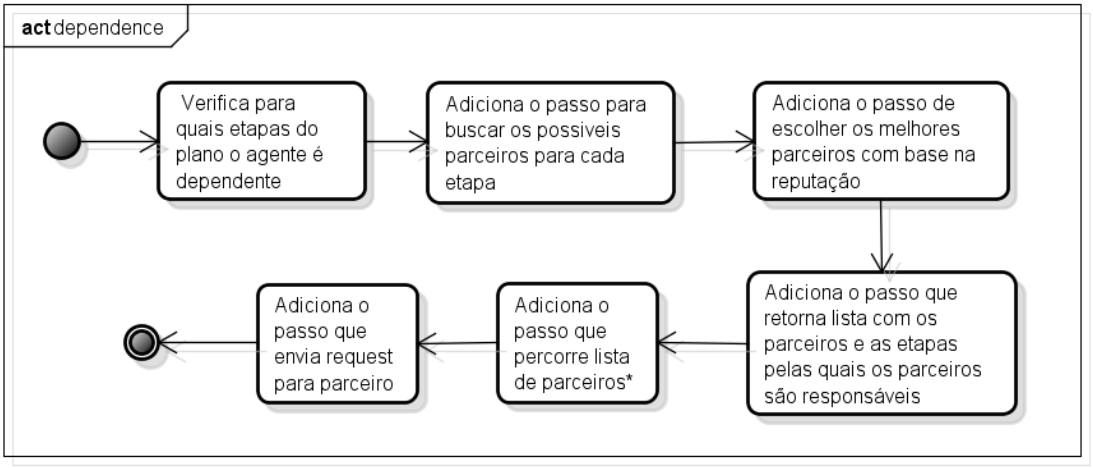


Figura 3.20: Diagrama de atividades correspondente aos passos adicionados a um plano quando detectadas etapas das quais o agente é dependente.

Um agente, quando recebe um conjunto de planos, pode fazer uma adoção parcial ou total. Caso o agente já possua em sua base de planos um plano para o objetivo do papel não é necessário que o plano que esta sendo enviado seja adotado. Na infraestrutura é considerado apenas o caso de adoção total dos planos do papel. Assim, quando um agente recebe um sinal referente a um plano, Figura 3.21, ele deve adicionar esse plano em sua biblioteca de planos.

Depois desse passo, são buscados os objetivos dos agentes classificados como *start goal*. Esses objetivos considerados como iniciais são passados através de sinais aos agentes. Os objetivos iniciais referem-se aos objetivo que o papel deve alcançar na ordem que devem ser atingidos. Quando um agente recebe um sinal de um *start goal*, Figura 3.22, o agente de disparar o objetivo inicial sinalizado.

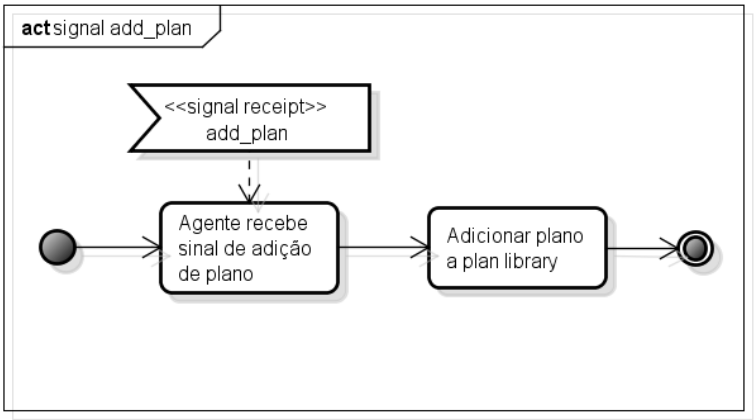


Figura 3.21: Diagrama de atividades do ponto de vista do agente quando um sinal referente a um novo plano é recebido.

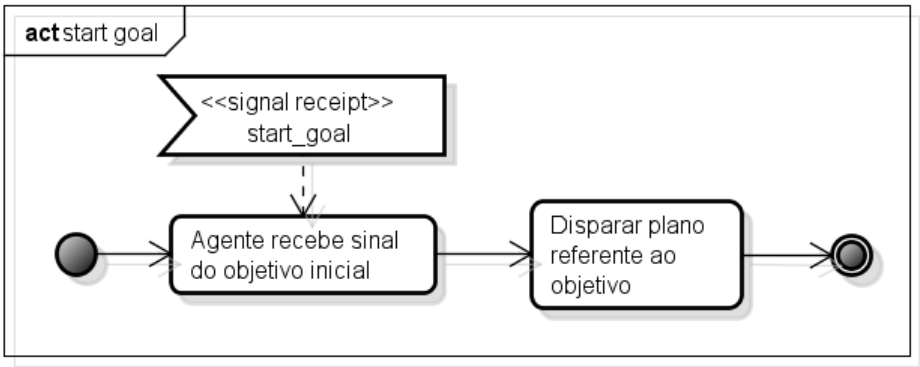


Figura 3.22: Diagrama de atividades do ponto de vista do agente quando um sinal referente a um objetivo inicial (*start goal*) é recebido.

Após todas essas etapas o agente é finalmente adicionado como membro da organização no artefato GeneralOrg.

Lembrando que quando um agente ag2 delega um objetivo a um agente ag3, Figura 3.23, isso é feito a partir do protocolo *Request*. Caso o agente ag2 que recebeu a delegação siga corretamente o fluxo de mensagens do protocolo, sua reputação de delegação quanto ao cumprimento do protocolo aumenta, caso contrário diminui. Depois da tarefa realizada o agente ag2 que delegou a tarefa recebe o resultado da tarefa. Se o resultado foi o esperado ag2 atualiza a reputação de ag3 aumentando a mesma, caso contrário a reputação é diminuída.

Para deixar um papel, um agente deve realizar a operação `leaveRole` sobre o artefato GeneralOrg que por sua vez realiza a operação `leaveRole` sobre o artefato Role através da interface de ligação entre os artefatos.

Como pode ser observado pela Figura 3.24 e pelos diagrama de atividade da Figura 3.25, quando a operação `leaveRole` é executada sobre o artefato GeneralOrg é verificado se o agente possui outros papéis na organização além do que ele está querendo

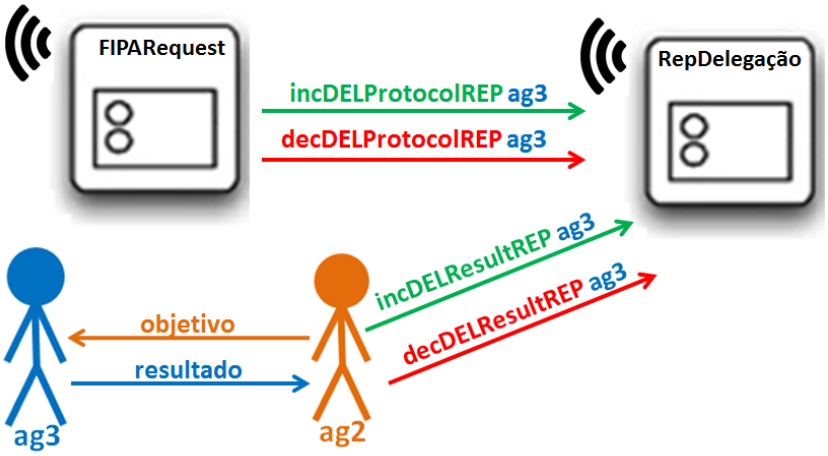


Figura 3.23: Processo de atualização das reputações de um agente quando o mesmo recebe a delegação de uma tarefa.

deixar. Caso o agente não possua outro papel ele deixa de fazer parte da organização sendo removido da estrutura que armazena os membros da organização no artefato. Além disso, o agente também é removido das estruturas que armazenam as reputações nos artefatos NormREP e DelegationREP. A operação `leaveRole` é executada então sobre o artefato Role. Essa operação remove o agente da estrutura que armazena quais agentes estão ocupando determinado papel no artefato e envia para o agente que deixou o papel um sinal para que os planos relativos ao papel sejam removidos, Figura 3.26.

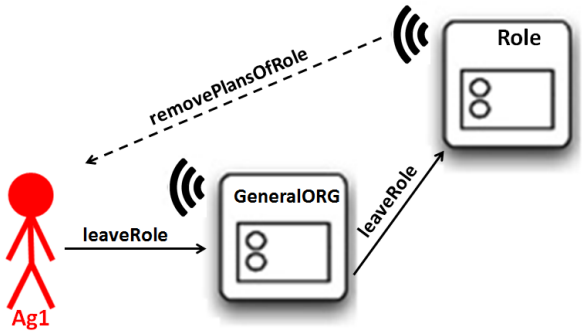


Figura 3.24: Processo que ocorre quando um agente executa a operação `leaveRole`

Na Figura 3.27 é mostrado o processo que ocorre quando um agente cria um grupo na organização. Para criar um grupo um agente ag1 que já é membro da organização deve executar a operação `createGroup` sobre o artefato GeneralOrg que por sua vez cria um artefato de grupo e atribui o agente ag1 como o gerente do grupo. Após criar um grupo é necessário formar o mesmo buscando agentes capazes de realizar determinadas etapas do plano. Para isso o agente gerente realiza a operação `getMembersOfPlan`, que retorna quais papéis são responsáveis por cada etapa do plano coletivo.

No diagrama da Figura 3.28 é mostrado o que ocorre quando um agente executa a

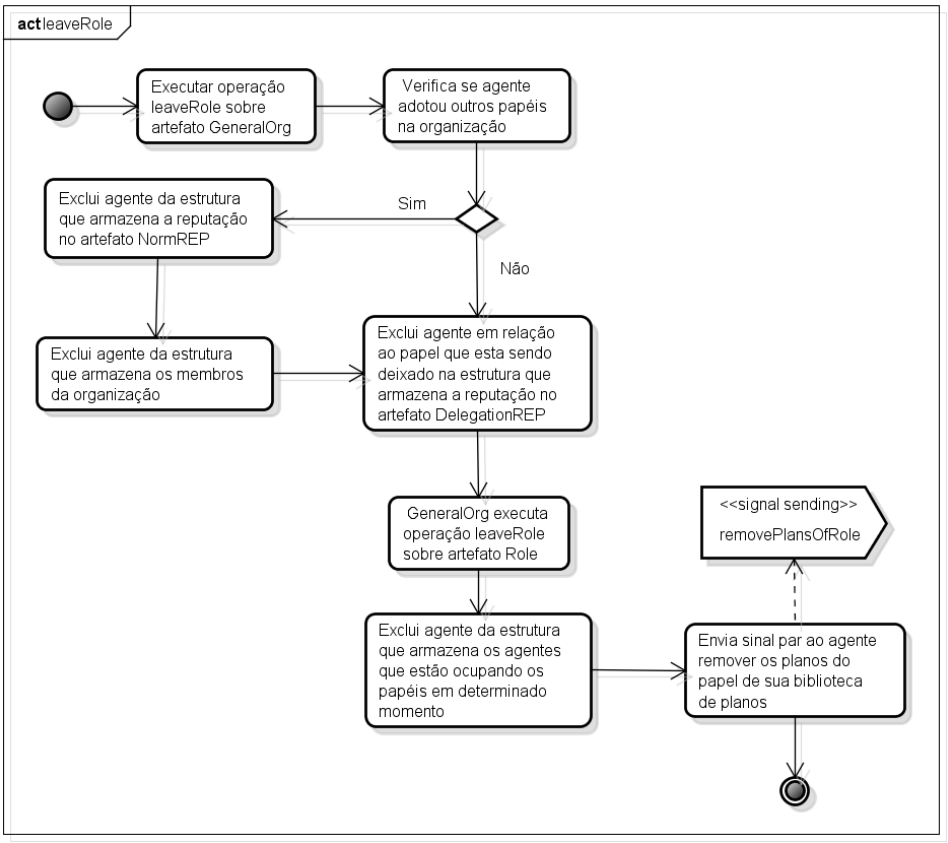


Figura 3.25: Diagrama de atividades correspondente a execução da operação `leaveRole`.

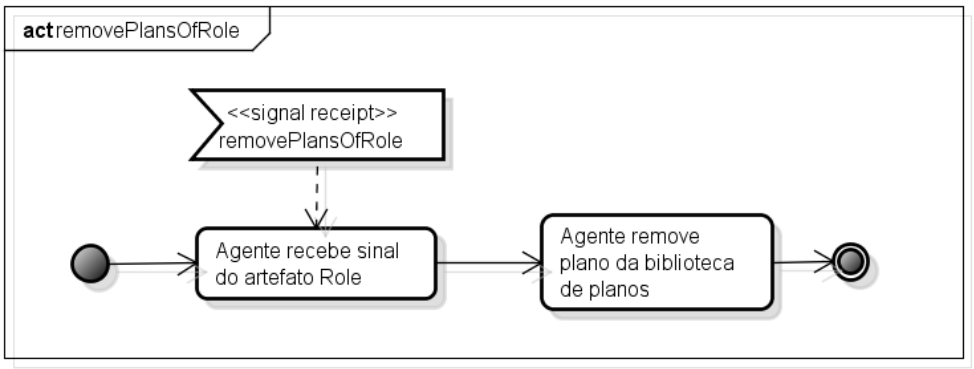


Figura 3.26: Diagrama de atividades relativo ao agente ao receber o sinal `removePlansOfRole`.

operação `getMembersOfPlan`.

Sabendo quais papéis são responsáveis por qual etapa do plano o gerente do grupo passa para próxima etapa que é buscar os agentes para formar o grupo. Para isso o agente gerente utiliza o protocolo *Contract Net* para selecionar os melhores agentes para cada etapa como mostrado na Figura 3.29.

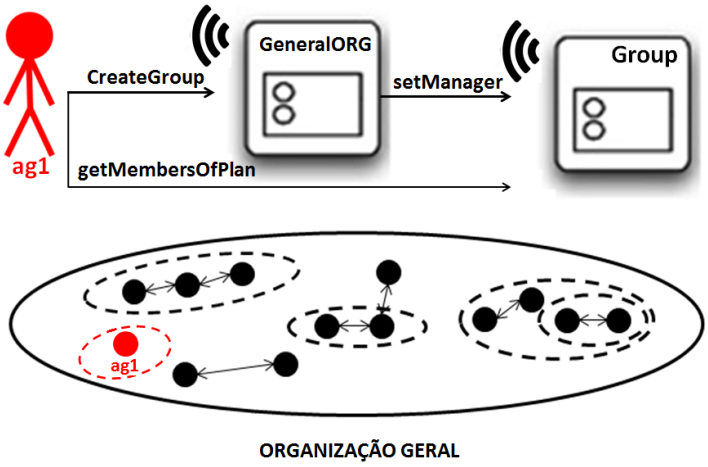


Figura 3.27: Processo de criação de um grupo por um agente na organização.

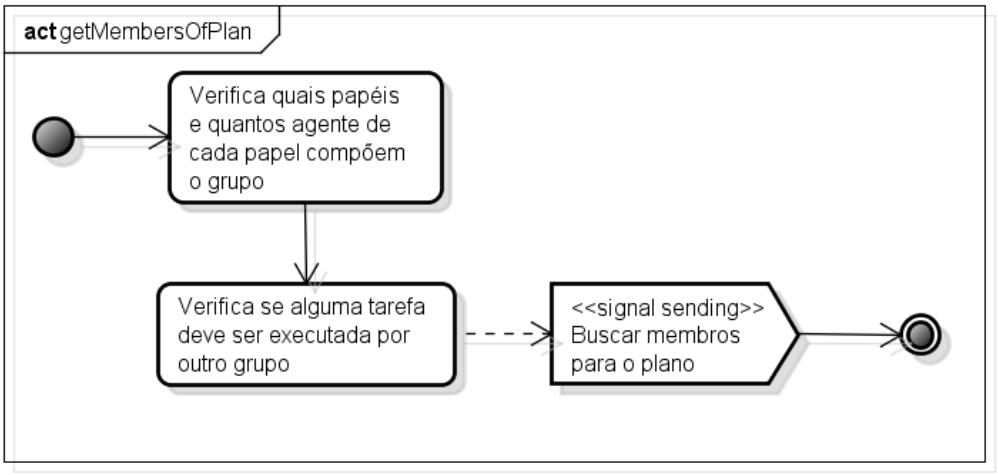


Figura 3.28: Diagrama de atividades sob o ponto de vista do artefato para formação de um grupo.

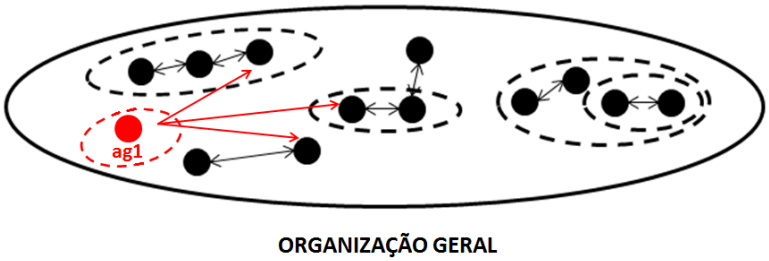


Figura 3.29: Gerente buscando agentes para formação do grupo através do protocolo *Contract Net*.

Primeiramente, o artefato do grupo irá verificar as etapas que tenham agentes como responsáveis. A partir disso é verificado qual papel é responsável e quantos agentes que possuem esse papel são necessários para realizar este passo do plano. Após são verifica-

das as etapas que tenham grupos como responsáveis. Feito isso é enviado um sinal para o agente gerente. Este sinal contém uma lista com as etapas e os papéis/grupos responsáveis.

Quando o agente gerente recebe o sinal sobre os membros do grupo, Figura 3.30, utilizando o protocolo *Contract Net*, para cada etapa do plano o agente gerente irá enviar uma solicitação para todos os agentes que possuam determinado papel capaz de realizar determinado passo do plano coletivo. Agentes que recebem a solicitação enviam uma proposta ao gerente. Essa proposta corresponde à média das reputações de delegação e de normas do agente.

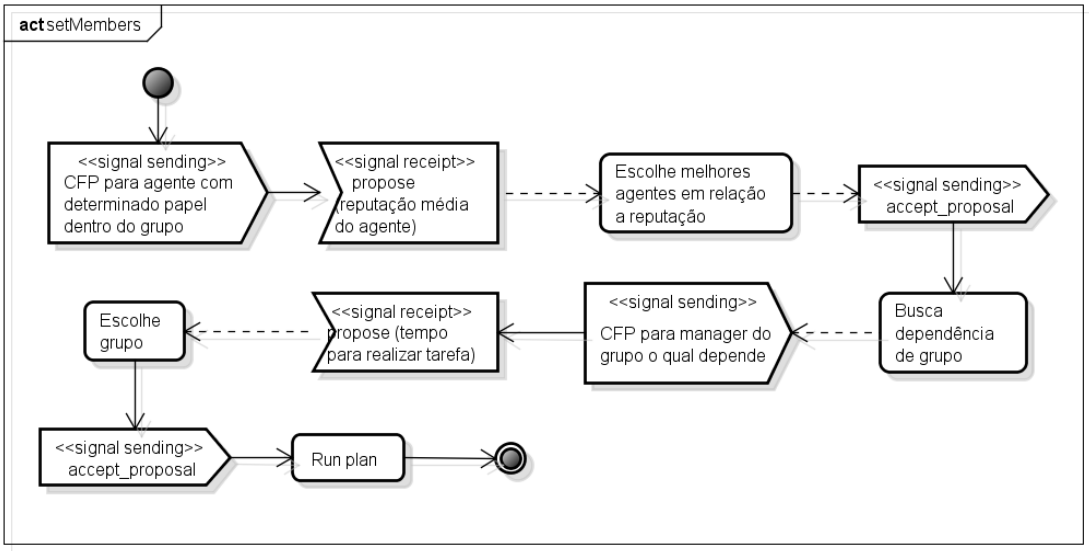


Figura 3.30: Diagrama de atividades sob o ponto do agente ao receber o sinal sobre os membros do grupo.

Após a etapa de propostas o agente gerente escolhe os melhores agentes capazes de realizar determinada etapa do plano. Os agentes escolhidos recebem uma mensagem de aceitação e passam a observar o artefato do grupo.

A próxima etapa consiste na busca por grupos capazes de realizar determinadas etapas do plano coletivo. O agente verifica qual grupo com determinado papel é capaz de realizar o passo do plano. Feito isso é buscado no artefato de dependência de grupo quais grupos criados são capazes de realizar determinado objetivo. O gerente então busca quem são os agentes gerentes desses grupos e envia, usando o *Contract Net*, solicitações para realização desse objetivo.

Os agentes gerentes dos grupos que recebem a solicitação enviam uma proposta ao agente gerente do grupo solicitante. A proposta contém o tempo necessário para que o grupo possa alcançar o objetivo solicitado. O gerente ao receber as propostas seleciona o grupo que irá levar menos tempo para alcançar o objetivo e envia uma mensagem de aceitação. O gerente do grupo que recebe a mensagem passa a observar o artefato do grupo.

Após a formação do grupo já é possível executar o plano coletivo. O diagrama de atividade da Figura 3.31 mostra a dinâmica da execução do plano coletivo.

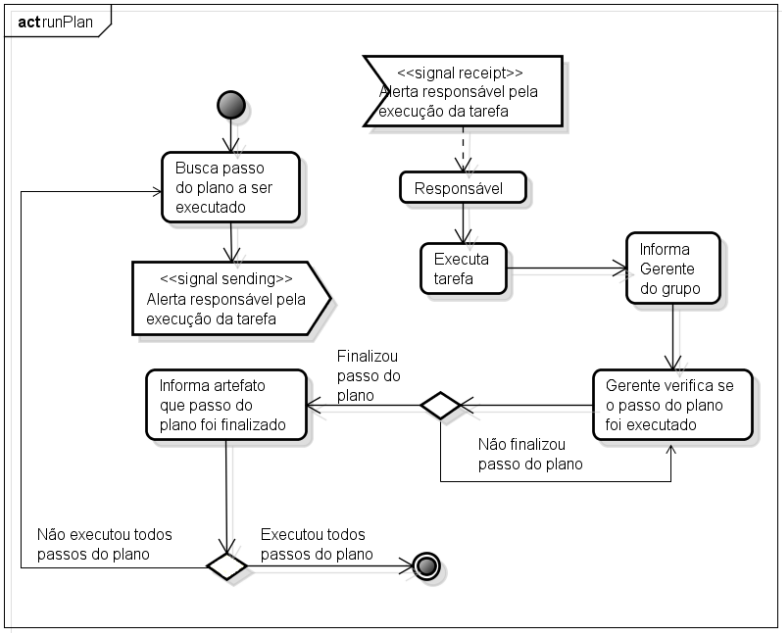


Figura 3.31: Diagrama de atividades referente a execução do plano coletivo.

O controle da execução do plano coletivo pode ser realizada pelo agente que gerencia o grupo, pelo agente gerente em conjunto com o artefato ou delegado totalmente para o artefato. Na infraestrutura, o controle da execução do plano realizado pelo agente gerente em conjunto com o artefato. O agente gerente, quando deseja iniciar a execução de um objetivo coletivo, executa a operação `runPlan`.

Quando essa operação é invocada pelo gerente a primeira coisa feita pelo artefato é buscar o passo do plano a ser executado. Feito isso, o artefato busca os agentes responsáveis por executar esse passo e envia um sinal avisando que tal objetivo deve ser alcançado. O responsável então executa esta tarefa e informa ao agente gerente que a tarefa foi executada.

Cada passo do plano coletivo pode ser executado por um ou mais agentes. O gerente então verifica se todos responsáveis pela etapa do plano executaram suas tarefas. Caso todos agentes tenha executado o gerente sinaliza para o artefato que o passo do plano foi concluído. O processo é repetido até que todos passos do plano tenham sido realizados.

Quando um agente gerente decide finalizar um grupo existente executa a operação `destroyGroup` sobre o artefato GeneralOrg. Essa operação exclui o grupo da estrutura que armazena os grupos da organização. Após é executada a operação `destroyGroup` através da interface de ligação sobre o artefato Group.

Como pode ser observado pela Figura 3.32 e pelo diagrama de atividade da Figura 3.33, quando a operação `destroyGroup` é executada sobre o artefato Group são en-

viados sinais para os agentes responsáveis pelos passos do plano. Quando os agentes recebem esses sinais, Figura 3.34, param de observar o artefato, ou seja, deixam de acompanhar os eventos referentes ao grupo.

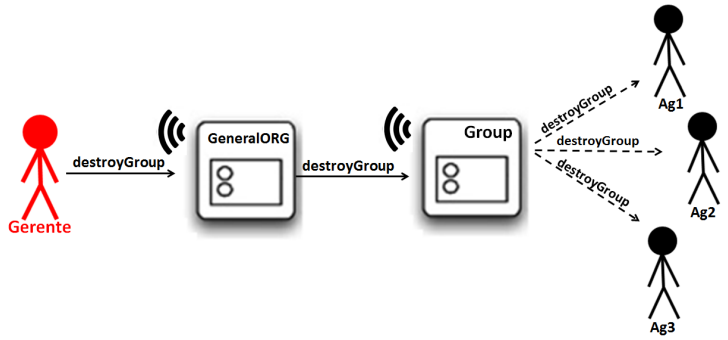


Figura 3.32: Processo para finalizar um grupo na organização.

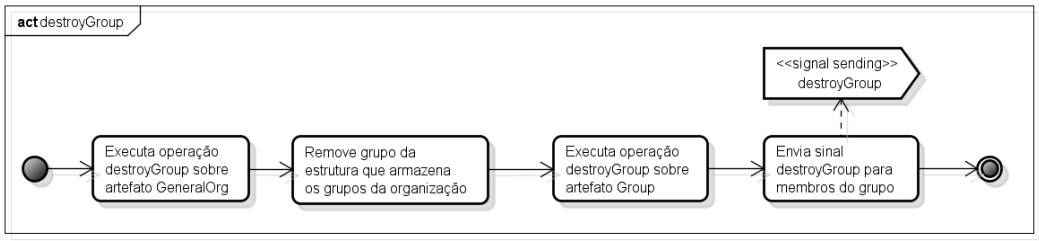


Figura 3.33: Diagrama de atividade relativo a finalização de um grupo na organização.

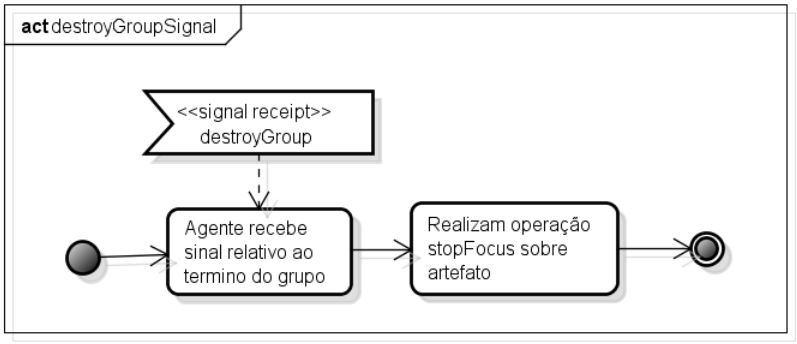


Figura 3.34: Diagrama de atividade do ponto de vista do agente quando recebe um sinal relativo a finalização de um grupo.

Na infraestrutura as normas são utilizadas para controlar o acesso aos artefatos e também para controlar quais papéis podem criar grupos. Como mostra a Figura 3.35, as normas podem ser criadas tanto a partir de um agente organizacional quanto a partir de uma especificação da dimensão normativa da infraestrutura.

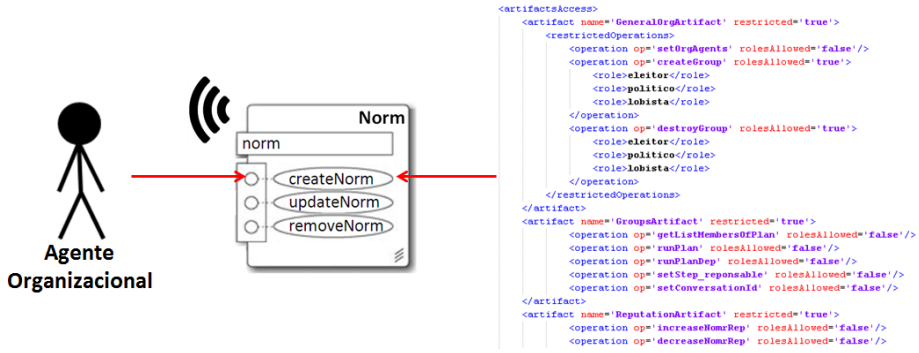


Figura 3.35: Criação de normas na infraestrutura.

O cumprimento/descumprimento das normas é controlado pelo agente organizacional. O cumprimento/descumprimento influencia diretamente na reputação de normas dos agentes. Nos artefatos organizacionais e também nos artefatos de ambiente que vierem a ser criados em função da aplicação existem operações que são restritas a determinados papéis e outras operações que são permitidas a todos os papéis.

Quando uma operação é restrita cada vez que um agente executa uma operação é enviado um sinal para o agente organizacional. Este sinal informa a operação e qual agente realizou a mesma. Quando recebe este sinal o agente organizacional verifica quais papéis que poderiam realizar a operação e verifica que o agente que executou a operação possuía um dos papéis permitidos. Caso o agente estivesse autorizado a realizar a operação sua reputação quanto ao cumprimento de normas aumentaria, caso contrário a reputação diminuiria. A Figura 3.36 e o diagrama da Figura 3.37 mostram como é feito este controle.

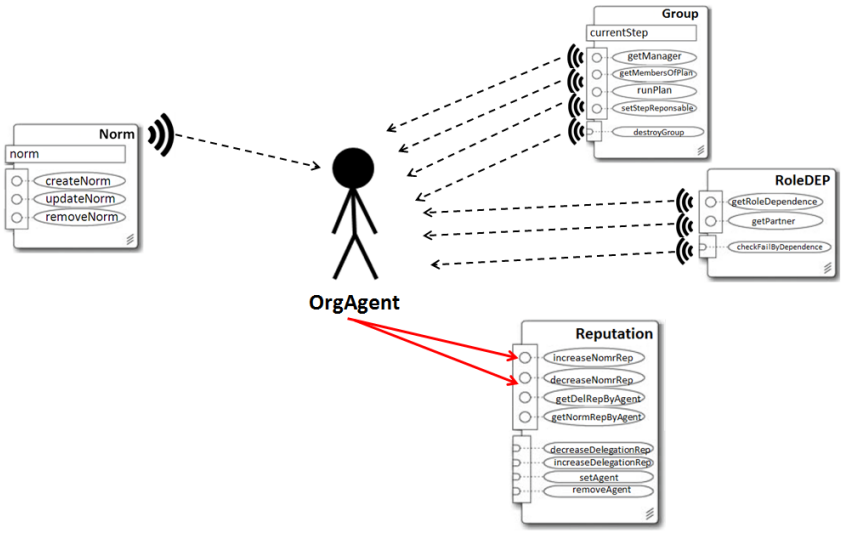


Figura 3.36: Controle realizado pelo agente organizacional em relação do cumprimento/-descumprimento das normas.

O processo de reorganização é mostrado na Figura 3.38 e no Diagrama de atividades da Figura 3.39. Um agente que deseje modificar a organização excuta a operação

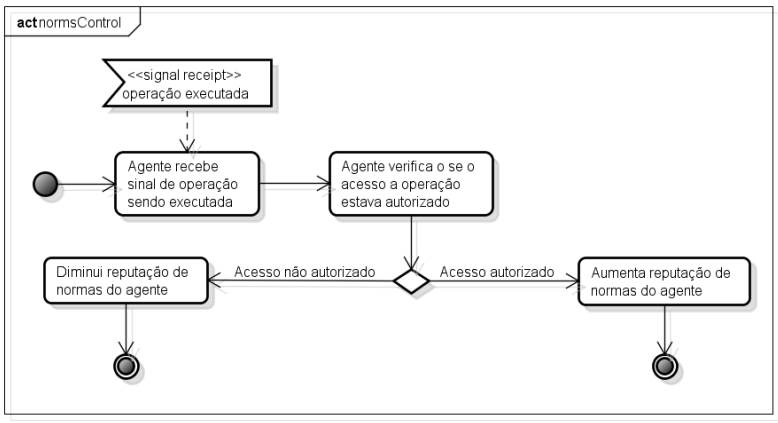


Figura 3.37: Diagrama de atividades correspondente ao controle realizado pelo pelo agente organizacional em relação cumprimento/descumprimento das normas.

`requestUpdateOrganization` sobre o artefato . Essa operação gera um sinal para o agente organizacional. Ao receber o sinal o agente organizacional avalia se modifica ou não a organização. Caso o agente organizacional aceite modificar a organização a operação `updateOrganization` é realizada sobre o artefato GeneralOrg.

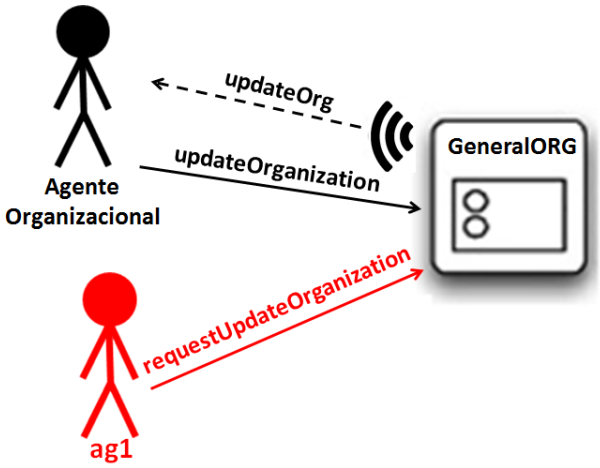


Figura 3.38: Representação do processo de reorganização.

3.4 Etapa de Desenvolvimento da Infraestrutura

Nesta etapa são apresentados os detalhes relativos ao desenvolvimento da infraestrutura. Além de alguns detalhes relativos à implementação é mostrada a especificação organizacional utilizada como entrada para infraestrutura que torna a organização disponível para os agentes.

A infraestrutura é implementada utilizando a plataforma CArtAgO, apresentada na Seção 2.2.1. Como já mencionado, a infraestrutura de comunicação, apresentada na Seção

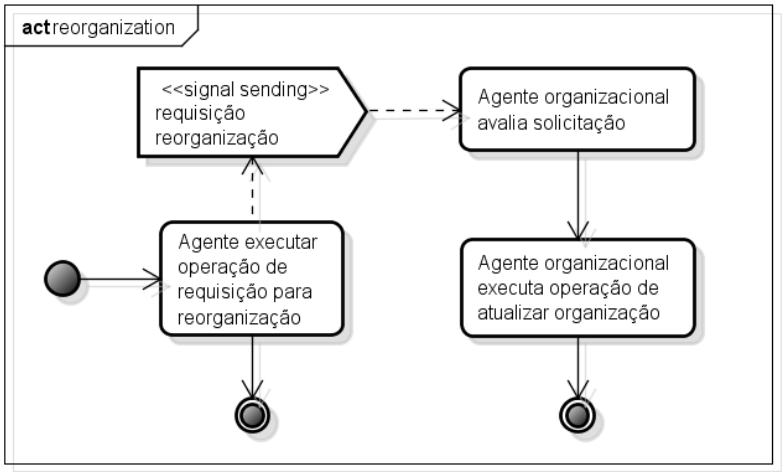


Figura 3.39: Diagrama de atividades correspondente ao processo de reorganização.

3.2.6, também desenvolvida utilizando a plataforma CArtAgO, é integrada a infraestrutura OrIAs.

CArtAgO possui pontes (*bridges*) que fornecem APIs especificas para criar, manipular e acessar artefatos. Atualmente existem *bridges* para as linguagens de programação orientada a agentes Jason[1], Jadex[2] e SimpA[3]. Além disso, estão sendo desenvolvidas bridges para 2APL[4] e JADE[5]. Assim a ideia é que a infraestrutura possa permitir que agentes implementados em diferentes linguagens e plataformas possam interagir e adotar a mesma especificação organizacional. Contudo em um primeiro momento os agentes utilizados para os testes com a infraestrutura foram implementados apenas usando Jason.

Para que a organização fique disponível aos agentes, a infraestrutura OrIAs recebe uma descrição da organização como entrada, interpreta a mesma e prove aos agentes a organização de acordo com a especificação fornecida.

A descrição que serve como entrada para OrIAs é escrita em um arquivo xml e permite descrever as características de cada dimensão presente na infraestrutura. Xml (eXtensible Markup Language) [6] foi escolhido, pois em função de seguir determinadas regras de sintaxe, os documentos deste tipo podem ser gerenciados por uma infraestrutura organizacional que pode estar habilitada a entender e manipular o conteúdo de um documento XML sem a necessidade da intervenção humana. Além disso, a plataforma CArtAgO, utilizada para implementação da infraestrutura, é totalmente desenvolvida em Java. Java por sua vez possui diversas bibliotecas [7] [8] [9] que facilitam a manipulação de XML. Outro fator, que contribui para utilização de xml, é que alguns modelos organizacionais, como

[1]http://jason.sourceforge.net/
[2]http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/About/Overview
[3]http://apice.unibo.it/xwiki/bin/view/SimpA/
[4]http://apapl.sourceforge.net/
[5]http://jade.tilab.com/
[6]http://www.w3schools.com/xml/
[7]http://www.jdom.org/
[8]http://xstream.codehaus.org/
[9]http://www.saxproject.org/

Moise (HÜBNER; SICHMAN; BOISSIER, 2002), utilizam este tipo de arquivo para escrever a especificação organizacional.

A descrição da infraestrutura no arquivo permite descrever separadamente os detalhes das dimensões: estrutural, funcional, dialógica e normativa. A seguir são apresentados trechos do arquivo onde são mostrados como são descritas as características das dimensões para uma aplicação qualquer.

Na descrição da dimensão estrutural, Código 3.1, são descritos os papéis e grupos presentes na organização.

Para os papéis são descritos: o nome que identifica o papel, objetivos, cardinalidade e compatibilidade. Os objetivos do papel descrevem o nome do objetivo, o tipo (*achievement* ou *maintenance*) e se o objetivo é inicial (*startGoal*) ou não. A cardinalidade refere-se ao número de agentes mínimo e máximo de cada papel auxiliando no controle do número máximo de agentes que podem adotar determinados papéis. A compatibilidade serve para controlar e impedir que agentes adotem papéis que um agente não pode assumir ao mesmo tempo.

Código 3.1: Trecho de código da descrição organizacional referente a dimensão estrutural.

```
<structuralDimension>
   <roleDescription>
      <role role='politico'>
         <roleCardinality min='1' max='2'/>
         <goals>
            <goal goal='ganharEleicao' type='achievement' startGoal='false'/>
            <goal goal='conseguirVotos' type='achievement' startGoal='false'/>
            <goal goal='manterApoioFinaceiro' type='maintenance' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
   </roleDescription>
   <groupDescription>
      <group name='processoEleitoral'>
         <goal goal='realizarEleicao' type='collective' startGoal='true'/>
         <groupMembers>
            <role role='politico' goal='ganharEleicao'/>
         </groupMembers>
         </group>
   </groupDescription>
</structuralDimension>
```

Para os grupos são descritos: o nome do grupo, objetivo e membros. O objetivo do grupo se refere ao nome do objetivo, o tipo (coletivo ou individual) e se o objetivo é inicial(*startGoal*) ou não.

Na descrição da dimensão funcional, Código 3.2, são descritos os planos individuais e coletivos para os objetivos descritos na dimensão estrutural.

Código 3.2: Trecho de código da descrição organizacional referente a dimensão funcional.

```
<functionalDimension>
   <individualGoals>
      <plans>
         <plan trigger='ganharEleicao' context='true'>
            <step type='goal' responsible='lobista'>financiarCampanha</step>
            <step type='goal' responsible='politico'>conseguirVotos</step>
         </plan>
      </plans>
   </individualGoals>
   <collectiveGoals>
      <plans>
         <plan trigger='realizarEleicao' context='true'>
            <step type='individualGoal' responsible='politico'numberOfAgentsByRole='2'>
                ganharEleicao</step>
```

```
            <step type='collectiveGoal' responsible='resultadoEleitoral'>
                contabilizarProcesso</step>
         </plan>
      </plans>
   </collectiveGoals>
</functionalDimension>
```

Para os planos individuais são descritos o *trigger* responsável pelo disparo do plano, o contexto que o plano se aplica e os passos do plano. Para os passos do plano são descritos o tipo do passo (ação, objetivo,...) e o responsável que é o papel capaz de executar o passo.

Para os planos coletivos são descritos o *trigger* responsável pelo disparo do plano, o contexto que o plano se aplica e os passos do plano. Para os passos do plano são descritos o tipo do passo (objetivo individual ou coletivo), o responsável que é o papel/grupo capaz de executar o passo. Caso o responsável seja um agente é especificado o número de agentes necessários para realizar o passo do plano.

Na descrição da dimensão dialógica, Código 3.3, são descritos os protocolos de comunicação que devem ser utilizados para comunicação dos agentes em situação de dependência e para formação de grupos.

Código 3.3: Trecho de código da descrição organizacional referente a dimensão dialógica.

```
<dialogicalDimension>
   <roleDependence>
      <protocol name='request'>
         <artifact artifactName='FIPARequestArtifact'/>
         <protocolFlow>
            <phase>
               <operation name='request'/>
            </phase>
            <phase>
               <operation name='agree'/>
                  <operation name='refuse'/>
            </phase>
            <phase>
               <operation name='failure'/>
               <operation name='inform_done'/>
               <operation name='inform_result'/>
            </phase>
         </protocolFlow>
      </protocol>
   </roleDependence>
   <groupFormation>
      <protocol name='contractNet'>
         <artifact artifactName='FIPAContractNetArtifact'/>
         <protocolFlow>
            <phase>
               <operation name='call_for_proposals'/>
            </phase>
            <phase>
                  <operation name='propose'/>
               <operation name='refuse'/>
            </phase>
               <phase>
               <operation name='accept_proposal'/>
               <operation name='reject_proposal'/>
            </phase>
            <phase>
               <operation name='failure'/>
               <operation name='inform_done'/>
               <operation name='inform_result'/>
            </phase>
         </protocolFlow>
      </protocol>
   </groupFormation>
</dialogicalDimension>
```

Na descrição da dimensão dialógica são descritos: o protocolo, qual artefato que implementa o protocolo, o fluxo de mensagens do protocolo em cada fase e as operações que devem ser executadas no artefato para o envio de determinado tipo de mensagem.

Na descrição de dimensão normativa, Código 3.4; são descritos: que papel pode criar qual grupo e os artefatos organizacionais e de aplicação que são controlados por normas.

Código 3.4: Trecho de código da descrição organizacional referente a dimensão normativa.

```
<normativeDimension>
   <createGroup>
      <group groupName='processoEleitoral'>
         <allowedRoles>
            <role>TSE</role>
         </allowedRoles>
      </group>
   </createGroup>
   <artifactsAccess>
         <artifact name='GeneralOrgArtifact' type='organizatonal' restricted='true'>
         <operation op='createGroup' rolesHaveAccess='true'>
            <role>eleitor</role>
            <role>politico</role>
            <role>lobista</role>
            <role>TSE</role>
         </operation>
         <operation op='destroyGroup' rolesHaveAccess='true'>
            <role>manager</role>
         </operation>
      </artifact>
   </artifactsAccess>
</normativeDimension>
```

Para criação de grupos são descritos o nome do grupo e quais papéis que tem permissão para criar o grupo. Para controlar o acesso dos artefatos são descritos os artefatos e para cada artefato é descrito o nome, o tipo (organizacional ou aplicação) e se o artefato tem suas operações restritas. Caso o artefato tenha suas operações restritas é descrita cada operação e identificado os papéis que podem ter acesso a essas operações.

Quando o arquivo xml é dado como entrada para infraestrutura a descrição organizacional é mapeada para um conjunto de classes que contém estruturas que se tornam acessíveis aos artefatos organizacionais. O diagrama dessas classes é mostrado nas Figuras 3.40, 3.41 e 3.42.

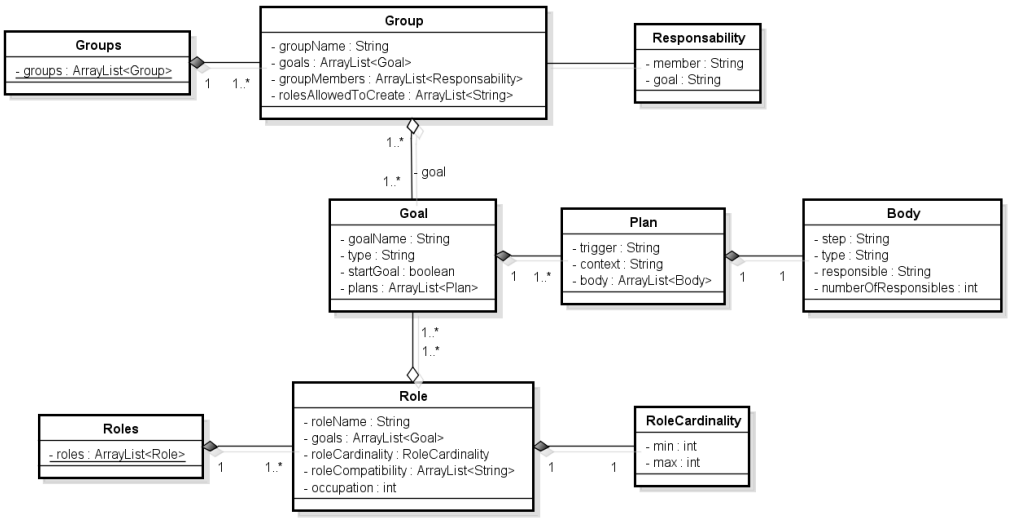


Figura 3.40: Diagrama de classe referente as dimensões estrutural e funcional.

A arquitetura lógica da infraestrutura é mostrada a partir do diagrama de pacotes da Figura 3.43. No diagrama o pacote `io` contém as classes responsáveis pela manipulação da especificação funcional. O pacote `classes` contém as classes das Figuras 3.40, 3.41 e 3.42 onde é mapeada a especificação organizacional. O pacote `organizationalArtifacts` contém as classes dos artefatos da infraestrutura. Os pacotes `protocolArtifacts`,

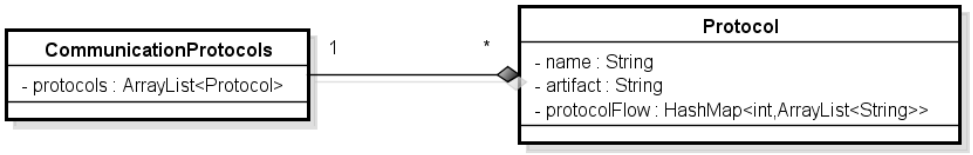


Figura 3.41: Diagrama de classe referente a dimensões dialógica.

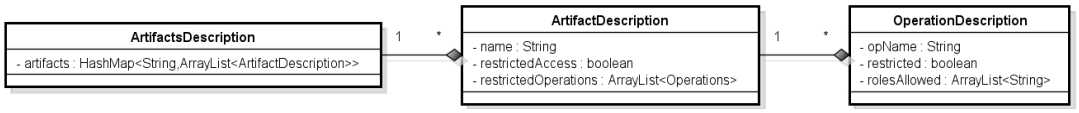


Figura 3.42: Diagrama de classe referente a dimensões normativa.

`communicationArtifacts` e `common` contém as classes da infraestrutura de comunicação utilizada.

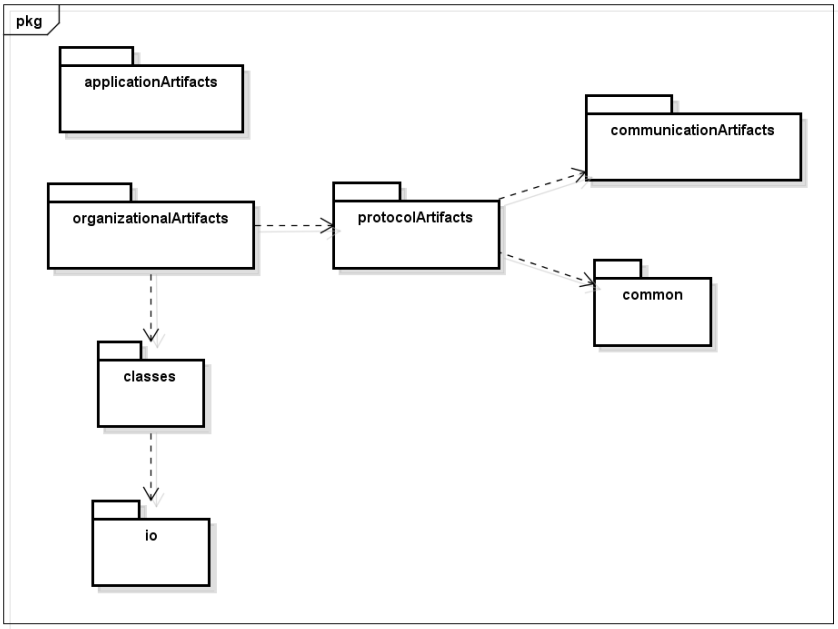


Figura 3.43: Arquitetura lógica da infraestrutura.

Para ser utilizada pelos agentes a infraestrutura precisa primeiramente ser criada. Isso é feito pelo agente organizacional. Esse agente é o responsável por criar os artefatos organizacionais, de comunicação e de aplicação. Além disso, o agente organizacional também é o responsável por fazer a ligação entre artefatos que possuam interface de ligação entre si.

Para que os agentes possam utilizar a infraestrutura de comunicação é necessário que essa seja criada e que os agentes se cadastrem nos artefatos de comunicação e passem a observar os mesmos. Assim, quando o agente organizacional cria os artefatos comuni-

cação envia um aviso aos agentes para que os mesmo possam se cadastrar nos artefatos criados.

Os agentes que passam a integrar a organização devem incluir um arquivo com códigos específicos para o tratamento dos sinais advindos dos artefatos organizacionais e dos artefatos de comunicação. Tendo esses códigos incluídos, agentes que desejam participar da organização precisam apenas realizar a operação `adoptRole`. A partir disso começa todo o processo de recebimento dos planos, objetivos e demais eventos já explicados nas seções anteriores.

Os códigos referentes à criação da infraestrutura pelo agente organizacional e ao código que deve ser incluído pelo agentes que desejam participar da organização é mostrado no Apêndice A.

Com a organização descrita, de acordo com a sintaxe especificada no arquivo xml, e os artefatos organizacionais, de comunicação e de aplicação sido criados pelo agente organizacional a infraestrutura pode ser utilizada para o gerenciamento organizacional.

4 EXPERIMENTO REALIZADO

Nesta seção será mostrado como os artefatos apresentados nas seções anteriores são utilizados pelos agentes na simulação de um *processo político de lobby*. São mostrados como os papéis, grupos, objetivos, planos, normas e dependências advindos do estudo de caso são descritos na especificação organizacional ficando disponíveis aos agentes através infraestrutura OrIAs. Na seção é mostrado ainda um exemplo do processo de reorganização que pode ser realizado na infraestrutura.

Primeiramente é apresentado o estudo de caso escolhido, após são apresentadas as características dos agentes utilizados na simulação e a linguagem que são implementados. Por fim é apresentada a simulação realizada.

4.1 Estudo de Caso

O estudo de caso utilizado é o *processo político de lobby através de contribuições para campanhas eleitorais*. Este estudo de caso foi escolhido por ser um exemplo que permite testar todas as funcionalidades providas pela infraestrutura como relações de dependência, reputação, normas, reorganização, dentre outros.

A modelagem do processo político de lobby é baseado no trabalho de (RODRIGUES, 2003) contudo algumas características da modelagem original foram modificadas com intuito testar todas funcionalidades da infraestrutura. Tais modificações incluem inserção de novos tipos de agentes, modificação de alguns objetivos do tipo *achievement* com intuito de criar algumas relações de dependências entre os agentes e criação de alguns objetivos do tipo *maintenance*.

No cenário são identificados 6 tipos (papéis) de agentes: político, lobista, eleitor, TSE, fiscal financeiro e fiscal dos votos. O contexto social o qual os agentes estão inseridos é uma eleição política. Cada agente tem objetivos e interesses próprios. Contudo podem não serem autossuficientes para realização de determinados objetivos sendo dependentes de outros papéis para realização dos mesmos. Além disso, existem algumas normas que esses agentes devem cumprir. Cada agente que participa da organização ao adotar algum dos papéis pode ser avaliado pelos demais membros da sociedade a partir de sua reputação relacionada ao cumprimento de normas e a delegação de tarefas.

O político tem como objetivo ganhar a eleição e para alcançar este objetivo deve conseguir apoio financeiro para campanha e a maioria dos votos dos eleitores. O lobista

representa um interesse particular (como por exemplo, grandes empresas) e deseja apoiar algum politico visando um benefício futuro com sua vitória. Assim o objetivo desses agentes seria financiar a campanha de algum político. O eleitor tem como interesse escolher um político que melhor represente seus interesses e os da sua comunidade. Assim o objetivo do eleitor é votar. O TSE corresponde ao tribunal superior eleitoral e tem como objetivo organizar o processo eleitoral. O fiscal financeiro é o agente que tem como objetivo verificar se um candidato possui o financiamento mínimo para sua campanha eleitoral. Já o fiscal dos votos tem como objetivo contabilizar os votos e informar o agente eleito pela maioria.

4.2 Agentes Utilizados na Simulação

Nesta seção são descritas as características e número de agentes utilizados na simulação e também como esses agentes são implementados.

4.2.1 Descrição dos Agentes

A simulação possui 13 agentes: 2 políticos, 4 lobistas, 3 eleitores, 1 TSE, 1 fiscal financeiro e 1 fiscal de votos. Os objetivos de cada agente e suas dependências são descritos com maior detalhamento a seguir.

- **Agente Político:** Este agente possui dois objetivos do tipo *achievement* e um objetivo do tipo *maintenance*.

  Um dos objetivos do tipo *achievement* é `ganhar eleição`. Esse objetivo é formado por dois subobjetivos: `financiar campanha` e `conseguir votos`. Para que o subobjetivo `financiar campanha` seja alcançado o agente político depende de agentes com o papel lobista. O subobjetivo `conseguir votos` possui como uma de suas etapas o objetivo `votar`. Para que o objetivo `votar` seja atingido o agente político depende de agentes com papel de eleitor.

  O objetivo de manutenção do agente político é `manter apoio financeiro`. O agente político precisa de no mínimo dois apoios para o financiamento da campanha. Caso o número de apoios se torne inferior ao desejado o agente politico deve disparar este objetivo de manutenção com intuito de retomar o estado desejado.

- **Agente Lobista:** Este tipo de agente possui um único objetivo do tipo *achievement* que é `financiar campanha`. Cada lobista deve financiar apenas uma campanha de algum dos agentes que adotaram o papel político.

- **Agente Eleitor:** Este tipo de agente possui um único objetivo do tipo *achievement* que é `votar` . Cada eleitor deve votar apenas uma vez e em um único político.

- **Agente TSE:** Este tipo de agente é o responsável por organizar todo o processo eleitoral. Tem como objetivo `organizar eleicao`. Esse objetivo cria um grupo com o papel de processo eleitoral com agentes responsáveis pela campanha eleitoral (políticos) e com agentes responsáveis pelo resultado das eleições (fiscal financeiro e fiscal de votos).

- **Agente Fiscal financeiro:** Este tipo de agente possui um único objetivo do tipo *achievement* que é `contabilizarFinanciamento`. Esse objetivo verificar se os políticos que se candidataram possuem o requisito mínimo de apoio financeiro.

- **Agente Fiscal dos votos:** Este tipo de agente possui um único objetivo do tipo *achievement* que é `contabilizar Votos`. Esse objetivo contabiliza os votos da eleição e informar o agente político que foi o vencedor.

4.2.2 Implementação dos Agentes

Apesar da possibilidade de utilizar as pontes (*bridges*) existentes no CArtAgO para que agentes implementados em diferentes linguagens possam adotar a mesma especificação organizacional, em um primeiro momento foram utilizados um conjunto de agentes implementados apenas em Jason (BORDINI; HÜBNER; WOOLDRIDGE, 2007).

Jason é um interpretador da linguagem *AgentSpeak* (RAO, 1996), utilizado para o desenvolvimento de sistemas multiagentes. Jason possui um código aberto[1] implementado na linguagem java, portanto multiplataforma.

Jason estende a linguagem *AgentSpeak(L)* que é uma extensão natural de programação em lógica para arquitetura de agentes BDI, sendo um modelo abstrato para programação de agentes.

A arquitetura BDI (*Belief, Desire, intentions*) é baseada no entendimento do raciocínio prático em humanos utilizado para decidir momento a momento qual ação executar em um próximo passo. A arquitetura possui este nome pelo fato de considerar crenças, desejos e intenções para execução de uma ação de forma racional. Essas atitudes mentais representam respectivamente os estados informacionais, motivacionais e deliberativos dos agentes (WOOLDRIDGE, 2000).

Na arquitetura BDI, um agente pode ser completamente especificado pelos eventos que ele pode perceber, as ações que pode executar, as crenças que detém, os objetivos que adota e por seus planos (KINNY; GEORGEFF; RAO, 1996). As crenças de um agente correspondem à informação que esse tem sobre o mundo. Desejos representam estados que podem ser atingidos. Intenções representam desejos os quais o agente se compromete a alcançar. Essa arquitetura apresenta ainda a ideia de planos que seriam os passos a serem seguidos que surgem a partir de uma intenção. Ao obter um plano o agente tentará executá-lo para atingir um estado desejado. Se o plano for executado com sucesso o estado desejado será atingido (WOOLDRIDGE, 2000).

Um agente em Jason é definido por conjunto de crenças, objetivos, e um conjunto de planos que formam uma biblioteca de planos (*plan library*). O agente reage a eventos que estão relacionados a mudanças em suas crenças devido à percepção do ambiente ou mudança nos objetivos do agente originados pela execução de planos disparados por outros eventos.Um agente está constantemente percebendo o ambiente e raciocinando sobre como alcançar seus objetivos.

[1] Disponível em http://jason.sourceforge.net

Um agente possui uma base de crenças (*belief base*). Essas crenças expressam uma propriedade particular de um objeto ou individuo. Jason utiliza anotações (*anotations*) para prover detalhes que estão associados a uma crença particular. Essas anotações permitem deixar a base de crenças mais legível ao mesmo tempo em que facilita o gerenciamento desta base. Outra característica de Jason é *strong negation* usada para expressar que um agente acredita explicitamente que algo é falso.

Objetivos representam estados de mundo que agente deseja alcançar. Jason possui dois tipos de objetivos: objetivos a serem realizados (*achievement goals*) e objetivos de teste (*test goals*). *Achievement goals* são formados por uma fórmula atômica prefixada com o operador '!' e representam o estado que o agente deseja alcançar. *Test goals* são prefixados pelo operador '?' sendo utilizados para recuperar informações disponíveis na base de crenças do agente.

Em Jason não existem estruturas que permitam representar objetivos mais complexos. Dessa forma, Jason não suporta diretamente objetivos de manutenção (*maintenance goals*). Contudo são disponibilizadas diretivas pré-processadas que transformam planos, de acordo com certos padrões, para obter o comportamento do objetivo de interesse.

No caso de um objetivo de manutenção é necessário que um agente possua planos caso um estado de mundo desejado não seja verdadeiro. O padrão que define um objetivo de manutenção no Jason é mostrado no Código 4.1.

Código 4.1: Padrão utilizado no Jason para representar um objetivo de manutenção, código extraído de Bordini, Hübner e Wooldridge (2007).

```
g[source(percept)].
-g : true <- !g.
```

A primeira linha do código indica que quando o agente inicia sua execução assume que um estado de mundo (crença) desejado é verdadeiro. Caso este estado de mundo se torne falso, ou seja, a crença seja removida da base de crenças, o plano da linha seguinte é disparado este plano tem como passo um objetivo que permite novamente obter o estado de mundo desejado.

Em Jason um plano contém três partes distintas: *trigger*, *context* e *body*.

```
trigger :  context <- body.
```

Um *trigger* existe para dizer ao agente qual dos planos em uma *plan library* deve ser utilizado para um determinado evento. O *context* de um plano é utilizado para verificar se determinado plano, entre várias alternativas, é aquele que obterá sucesso para o tratamento de determinado evento, levando em consideração a informação corrente que o agente tem sobre o ambiente. O *body* de um plano é uma sequencia de ações ou (sub)objetivos que o agente deve realizar quando o plano é disparado.

Planos são disparados em função da decorrência de eventos. Eventos representam mudanças nas crenças ou devido a objetivos. As mudanças podem ser de dois tipos adição ('+') ou remoção ('-'). Planos podem ainda conter rótulos (*labels*) que podem ser usados para elaborar funções de seleção.

Eventos para o tratamento de falhas em planos também são disponibilizados pelo Jason. Se uma ação falha, ou não existe um plano aplicável para um objetivo, o plano causador da falha é removido das intenções e um evento interno associado ao plano que falhou associado a mesma intenção é gerado. Se o programador provê um plano para o tratamento desta falha, este plano é colocado no topo das intenções. Caso não seja disponibilizado um plano pelo programador, a intenção é descartada e este evento é informado com uma mensagem de alerta.

Outra funcionalidade provida são as ações internas usadas tanto no contexto quanto no corpo dos planos. Qualquer ação que inicie por '.' denota uma ação interna. Essas ações são definidas por usuários e são executadas internamente pelos agentes. Essas ações diferem-se das demais ações encontradas nos planos que são executadas com intuito de modificar o ambiente compartilhado.

Jason inclui ainda comunicação entre agentes baseada na teoria de atos de fala (*speech-act*) provida por performativas de comunicação (*tell, untell, unachieve, askOne, askAll, tellHow, untellHow, askHow*) baseadas na linguagem KQML. Existe ainda a possibilidade de executar de forma distribuída um sistema multiagente utilizando Saci (HÜBNER; SICHMAN, 2000) ou Jade (BELLIFEMINE; CAIRE; GREENWOOD, 2007).

Jason oferece ainda suporte para o desenvolvimento de ambientes que são programados em Java e um IDE na forma de um plugin jEdit ou Eclipse que inclui um *inspector mind* que auxilia no debugging.

4.3 Simulação Realizada

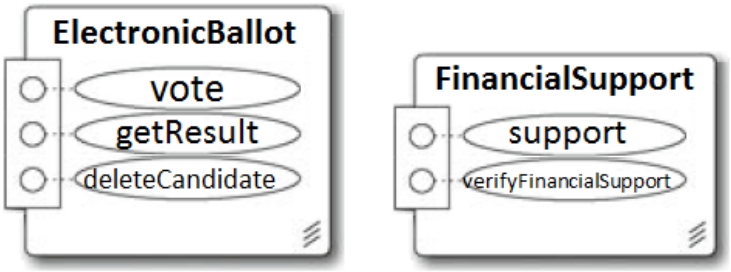
Para simulação foram criados dois artefatos de aplicação: ElectronicBallot e FinancialSupport. ElectronicBallot representa a urna que é utilizada pelos eleitores para realizar a votação e utilizada pelos fiscais de voto para contabilizar os votos. FinancialSupport é o artefato utilizado pelos lobistas para apoiar financeiramente os candidatos e utilizado pelos ficais financeiros para verificar se os candidatos receberam o apoio financeiro mínimo.

As representações dos artefatos ElectronicBallot e FinancialSupport são mostradas na Figura 4.1.

O artefato ElectronicBallot possui uma estrutura que armazena os agentes que com papel político e seus respectivos números de votos. O artefato FinancialSupport possui uma estrutura que armazena também os políticos candidatos e seus respectivos apoiadores (lobistas).

A interface de uso do artefato ElectronicBallot possui as seguintes operações:

- `vote`: Operação utilizadas pelos agentes com papel de eleitor para votarem nos candidatos desejados.
- `getResult`: Operação utilizada pelo agente com papel de fiscal de voto para obter o resultado da eleição.



(a) Representação do artefato ElectronicBallot.

(b) Representação do artefato FinancialSupport.

Figura 4.1: Representação dos artefatos de aplicação utilizados.

– `deleteCandidate`: Operação utilizada pelo agente fiscal financeiro, para retirar um agente com papel político da eleição, quando é constatado que o mesmo não possui o apoio mínimo para o financiamento da campanha.

A interface de uso do artefato FinancialSupport possui as seguintes operações:

– `support`: Operação utilizada pelos lobistas quando desejam apoiar um agente com papel político.

– `verifyFinacialSupport`: Operação utilizada pelos agentes com papel fiscal financeiro para verificar se os agentes com papel político receberam apoio financeiro suficiente para candidatura.

Antes de iniciar a simulação é necessário especificar a organização em um arquivo xml conforme mostrado na Seção 3.4. Nesta especificação serão descritas as características de cada uma das quatro dimensões organizacionais utilizadas na simulação. No Apêndice C é mostrado como especificar no arquivo xml, que serve como entrada para infraestrutura, uma organização utilizando o estudo de caso adotado neste trabalho como exemplo.

Na simulação são utilizados dois arquivos xml com a descrição organizacional. O primeiro arquivo é utilizado para a descrição inicial da organização. Este arquivo é passado, pelo agente organizacional na criação dos artefatos, para o artefato responsável por prover a organização aos agentes. O segundo arquivo é utilizado para demonstrar o processo de reorganização.

Para o primeiro arquivo, na dimensão estrutural são descritos os papéis e grupos da organização juntamente os seus objetivos. Para os papéis é descrita também as cardinalidades e as compatibilidades. Os papéis descritos neste arquivo são: político, lobista, eleitor e TSE. Os objetivos correspondentes a esses papéis são os mesmos descritos na Subseção 4.2.1. O grupo descrito no arquivo é o processo eleitoral cujo objetivo é `realizar eleição` que tem o papel político e o grupo resultado eleitoral como membros do plano coletivo.

Na dimensão funcional são descritos os planos para cada um dos objetivos descritos na dimensão estrutural. Os passos para os planos dos agentes segue o que foi descrito no

na Subseção 4.2.1. Os passos do objetivo coletivo são mostrados na Figura 4.2. Neste plano é possível observar que existe uma dependência de grupo. O grupo processo eleitoral depende do grupo resultado eleitoral para realização do objetivo coletivo.

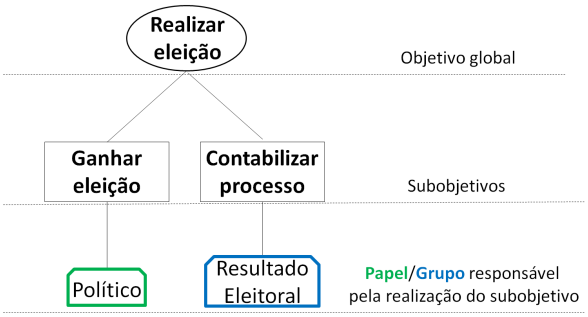


Figura 4.2: Passos do plano coletivo `realizar eleição`.

Na dimensão dialógica são descritos os protocolos utilizados em situações de dependência entre os agentes e para formação de grupos. São utilizados o protocolo *Request* para dependências entres os agentes e o protocolo *Contract Net* para formação de grupos.

Na dimensão normativa são especificadas normas para criação de grupo e para acesso as operações dos artefatos organizacionais e dos artefatos de aplicação. A Tabela 4.1 mostra as operações restritas em cada artefato e os papéis que podem acessar as operações. As operações que não mostradas na tabela podem ser acessadas por todos os agentes. As operações que nenhum agente, que adote um papel na organização, pode acessar são operações que apenas agentes organizacionais ou artefatos, através da interface de ligação, têm acesso.

O arquivo completo utilizado para especificação organizacional do estudo de caso pode ser encontrado no Código B.1 Apêndice B.

O arquivo que irá demonstrar o processo de reorganização irá descrever mais dois papéis, outro grupo para organização e novas normas para os artefatos de aplicação.

Na dimensão estrutural são definidos os papéis: fiscal financeiro e fiscal de votos. Os objetivos referentes aos papéis são os mesmos mencionados na Subseção 4.2.1. O grupo definido tem o papel de resultado eleitoral, esse grupo tem como objetivo coletivo `contabilizar processo`.

Na dimensão funcional são descritos os planos para cada um dos objetivos descritos na dimensão estrutural. Os passos para os planos dos agentes segue o que foi descrito no na Subseção 4.2.1. Os passos do objetivo coletivo são mostrados na Figura 4.3.

Nesta descrição a dimensão dialógica não é alterada. Na dimensão normativa são inseridas restrições às operações dos artefatos de aplicação referentes aos novos papéis descritos, como mostra a Tabela 4.2.

Tabela 4.1: Operações restritas em cada artefato e os respectivos papéis que podem acessar as operações.

Artefato	Norma	
	Operação	Papéis com acesso permitido
GeneralOrgArtifact	createGroup	eleitor, político, lobista e TSE
	destroyGroup	gerente do grupo
GroupsArtifact	getListMembersOfPlan	gerente do grupo
	runPlan	gerente do grupo
	setStepReponsable	gerente do grupo
NormReputationArtifact	increaseNomrRep	nenhum
	decreaseNomrRep	nenhum
NormArtifact	createNorma	nenhum
	removeNorm	nenhum
	updateNorm	nenhum
DelegationReputationArtifact	increaseDelegationProtocolRep	nenhum
	decreaseDelegationProtocolRep	nenhum
GroupDEPArtifact	todas operações	todos papéis
RolesArtifact	todas operações	todos papéis
RoleDEPArtifact	todas operações	todos papéis
FIPARequestArtifact	todas operações	todos papéis
FIPAContractNetArtifact	todas operações	todos papéis
electronicBallot	vote	eleitor
financialSupport	support	lobista

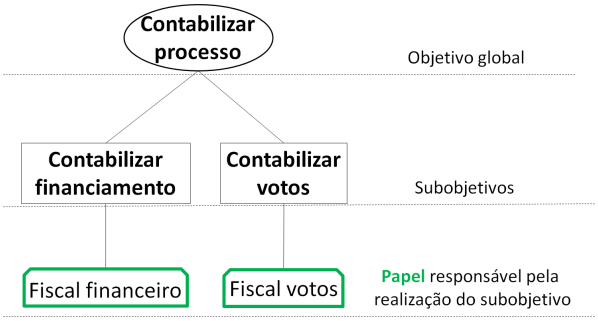


Figura 4.3: Passos do plano coletivo `contabilizar processo`.

O arquivo completo utilizado para especificação organizacional do processo de reorganização do estudo de caso pode ser encontrado no Código B.2 do Apêndice B.

Na simulação primeiramente toda a infraestrutura (artefatos organizacionais, artefatos de comunicação e artefatos de aplicação) é criada pelo agente organizacional, como mostra o Código A.1 do Apêndice A.

As agentes utilizados na simulação possuem todos o mesmo Código A.3 do Apêndice A. O que varia é o papel que desejam adotar.

Tabela 4.2: Operações restritas em cada artefato e os respectivos papéis que podem acessar as operações.

Artefato	Norma	
	Operação	Papéis com acesso permitido
electronicBallot	getResult	fiscalVotos
	deleteCandidate	fiscalFinanceiro
financialSupport	verifyFinancialSupport	fiscalFinanceiro

Depois de criada a infraestrutura, os agentes começam a adoção dos papéis. Como mencionado anteriormente, dois agentes adotam o papel político, quatro agentes adotam o papel lobista, três agentes adotam o papel eleitor e um agente adota o papel TSE.

Ao adotar um papel uma agente recebe os objetivos referentes ao mesmo e os planos para alcançar estes objetivos. Os objetivos tem a característica de serem iniciais ou não.

Um agente que adota o papel político recebe os objetivo e planos referentes ao mesmo. Contudo estes objetivos não são iniciais, assim o agente apenas recebe os planos e aguarda até o momento que esses objetivos devam ser alcançados.

Quando um agente irá receber um plano, no momento que o plano está sendo montado existe a verificação se o agente é autossuficiente para a realização do mesmo. No caso do político, para o objetivo `ganhar eleição` ,esse papel é dependente de dois outros papéis: lobista e eleitor. Dessa forma ao invés do agente receber o plano do Código 4.2, o artefato se encarrega de buscar as dependências e entrega um plano completo para o agente, Código 4.3, no qual em tempo de execução é possível buscar os parceiros para as tarefas que o agente necessita. Além disso, o agente também recebe um plano de contingência, caso ocorra alguma falha no plano Código4.4.

Código 4.2: Plano original referente ao objetivo `ganhar eleição` do agente político.

```
+!ganharEleicao[source(Source)] : true <-
   !financiarCampanha;
   !conseguirVotos.
```

Código 4.3: Plano modificado pelo artefato referente ao objetivo `ganhar eleição` do agente político.

```
+!ganharEleicao[source(Source)] : true <-
   getPartner("ganharEleicao","financiarCampanha",Partner);
   internalAction.int_random(1000000,RandConv);
   .concat("C",RandConv,IdConv);
   request(Partner,IdConv,financiarCampanha)["FIPARequestArtifact"];
   .wait(1000);
   ?newagreeMessage(_,IdConv);
   -newagreeMessage(_,IdConv);
   .wait(1000);
   ?newinformMessage(_,IdConv);
   -newinformMessage(_,IdConv);
   !conseguirVotos.
```

Código 4.4: Plano de contingência para o objetivo `ganhar eleição`.

```
-!ganharEleicao[source(Source)]: true <- !ganharEleicao.
```

O papel político possui um objetivo de manutenção denominado `manter apoio financeiro`. Para que o plano referente a este objetivo seja executado é necessário que

o mesmo esteja inserido em um contexto. O contexto especificado é que a crença que o agente tem em relação ao apoio financeiro seja maior ou igual a dois apoios. O artefato se encarrega de montar este plano, Código 4.5, e repassar ao agente que adotou o papel. Ao mesmo tempo, um plano referente à alteração da crença (estado de mundo desejado) é montado. Neste plano, Código 4.6, cada vez que a crença referente ao apoio financeiro se altera é verificado se este número é o desejado. Caso o estado de mundo não seja o desejado é disparado o objetivo de manutenção `manter apoio financeiro`.

Código 4.5: Plano referente ao objetivo de manutenção do agente político.

```
+!manterApoioFinaceiro[source(Source)] : apoioFinanceiro(X) & X < 2 <-
   getPartner("manterApoioFinaceiro","financiarCampanha",PartnerZDM);
   internalAction.int_random(1000000,RandConvXXO);
   .concat("C",RandConvXXO,IdConvIIL);
   request(PartnerZDM,IdConvIIL,financiarCampanha)["FIPARequestArtifact"];
   .wait(1000);
   ?newagreeMessage(_,IdConvIIL);
   -newagreeMessage(_,IdConvIIL);
   .wait(1000);
   ?newinformMessage(_,IdConvIIL);
   -newinformMessage(_,IdConvIIL).
```

Código 4.6: Plano referente a adição de crença.

```
-apoioFinanceiro(X) : X < 2<-!manterApoioFinaceiro.
```

Já o agente quando adota o papel TSE possui o objetivo `organizarEleicao` que tem como característica ser um objetivo inicial. Assim quando o agente recebe o plano, Código 4.7, referente a esse objetivo, um sinal do objetivo inicial é enviado ao agente. O recebimento desse sinal implica no agente iniciar a execução do plano para tentar alcançar este objetivo.

Código 4.7: Plano referente ao objetivo `organizar eleição`.

```
+!organizarEleicao[source(Source)] : true <-
   requestUpdateOrganization("updateElection.xml");
   .wait(10000);
   !createGroup("resultadoEleitoral");
   .wait(5000);
   !createGroup("processoEleitoral").
```

Como agente TSE é o único dos agentes que possui um plano com a característica de inicial toda a dinâmica da simulação se da a partir deste agente.

A primeira coisa que o agente com papel TSE faz é solicitar o processo de reorganização enviando o arquivo xml com a descrição organizacional para que a organização seja modificada. Caso o agente organizacional aceite esta mudança, há uma atualização e uma integração das características descritas nos dois arquivos. A atualização está relacionada a modificação de algo já definido anteriormente, como por exemplo, a modificação das restrições sobre artefatos já definidas anteriormente. A integração está relacionada à inserção de novas características na organização, como por exemplo, novos papéis ou grupos sendo inseridos na organização.

No caso da simulação o agente organizacional irá aceitar a solicitação do agente TSE. Com a modificação da organização o agente TSE pode solicitar a criação do grupo resultado eleitoral, que antes não era possível por não estar disponível anteriormente na organização pode solicitar posteriormente a criação do grupo processo eleitoral que depende do grupo resultado eleitoral.

Assim como nos objetivos individuais dos agentes, os objetivos coletivos são definidos também como iniciais ou não. Quando um objetivo é definido como inicial, após o

agente gerente criar o grupo e buscar os membros, é enviado um sinal para o gerente com intuído de iniciar a execução do plano coletivo. Quando um plano coletivo não tem a característica de inicial, o gerente cria o grupo, busca os membros para a realização das etapas, contudo não inicia a imediatamente execução do plano.

Na simulação, após o agente TSE criar o grupo processo eleitoral são buscados os membros (agentes e grupos com determinado papel) que farão parte do grupo. Como o objetivo é do tipo inicial o plano será executado logo após a formação do grupo. Seguindo os passo do plano descritos na Figura 4.2, primeiramente os políticos serão sinalizados para alcançarem o objetivo ganhar eleição. Após a etapa ser concluída por todos agentes comprometidos com a mesma, o próximo passo do plano coletivo é executado. Esse passo consiste no objetivo `contabilizar processo` cujo responsável é o grupo resultado eleitoral anteriormente criado pelo agente TSE. O gerente deste grupo recebe então um sinal para que o objetivo coletivo do grupo resultado eleitoral seja iniciado. Terminada todas as etapas do plano o objetivo coletivo do grupo processo eleitoral está finalizado.

Para os objetivos individuais e coletivos da simulação existem relações de dependência entre papéis e entre grupos. Para formação de grupos é necessária a busca por agentes. Para os dois casos citados existem critérios para escolha dos agentes e grupos. Essas situações mencionadas acarretam: na comunicação entre os agentes de acordo com o protocolo definido, na atualização da reputação dos agentes quanto a tarefas delegadas e também quanto ao cumprimento de normas, e na busca pelas possíveis dependências. Toda a dinâmica, na simulação, relacionada a esses aspectos é a mesma que está descrita na Seção 3.3 que mostra como os artefato se comportam mediante sua utilização pelos agentes e o como os eventos disparados pelos artefatos afetam o processo cognitivo desses agentes.

5 COMPARATIVO COM TRABALHOS RELACIONADOS

Esta seção destina-se a realizar um comparativo da infraestrutura proposta com os dois trabalhos que mais se relacionam com a mesma: a abordagem ORA4MAS (HÜBNER et al., 2010) e a plataforma JaCaMo (BOISSIER et al., 2011). Primeiramente são apresentadas as características de cada um desses trabalhos e por fim é feito um comparativo com a infraestrutura OrIAs.

5.1 ORA4MAS

ORA4MAS *Organizational Artifacts for Multi-Agent Systems* (HÜBNER et al., 2010) é uma abordagem baseada no metamodelo de agentes e artefatos (A&A) (OMICINI; RICCI; VIROLI, 2008) que considera o ambiente não apenas como uma fonte de percepções para os agentes ou alvo das ações dos mesmos, mas também capaz de encapsular funcionalidades e serviços. O trabalho demonstra a utilização de artefatos organizacionais com intuito oferecerem serviços necessários para o gerenciamento de uma organização.

Na abordagem apresentada por ORA4MAS, uma infraestrutura para o gerenciamento organizacional é composta por agentes organizacionais e por artefatos organizacionais. Agentes organizacionais são responsáveis pela criação e gerenciamento dos artefatos da organização. Artefatos organizacionais são utilizados pelos agentes da aplicação, que participam da organização, de forma a facilitar suas atividades, e por agentes organizacionais para gerenciar e adaptar a organização em tempo de execução.

Segundo esta abordagem, para que uma organização obtenha sucesso é necessário que as restrições impostas pela especificação organizacional sejam garantidas pela infraestrutura organizacional. Essas restrições podem ser a partir de normas estabelecidas pela especificação organizacional. Em ORA4MAS as normas se relacionam a obrigações, permissões ou interdições para realizar alguma ação ou alcançar algum objetivo. Contudo para que sejam uteis normas precisam de um mecanismo para aplicação das mesmas. ORA4MAS apresenta dois mecanismos para aplicação das normas: regimentação (*regimentation*) e imposição (*enforcement*).

Regimentation é um mecanismo que impede que um agente realize ações que não são permitidas por normas de forma a preservar importantes características da organização. Este mecanismo não permite violação por parte dos agentes. *Enforcement* é um mecanismo aplicado após detectada a violação de alguma norma. O cumprimento/descumprimento de uma norma deve ser detectado e avaliado para verificar se é passível de

punição ou recompensa. Assim *regimentation* deve ser usada para restringir a ação dos agentes enquanto que *enforcement* deve ser utilizada quando alguma violação é permitida.

Nessa abordagem mecanismos de regimentação são inseridos nos artefatos que os agentes da aplicação utilizam para acessar a organização. Isso permite, por exemplo, controlar quando e se um agente pode adotar um papel. Dessa forma a autonomia do agente é limitada pelo artefato. Em funções da violação das normas ser permitida, para o mecanismo de aplicação de normas são utilizadas funcionalidades providas pelos artefatos para detectar e mostrar o cumprimento/descumprimento de uma norma.

O trabalho desenvolvido por Hübner, boissier e Vercouter (2008) argumenta que a reputação pode ser considerada como um instrumento de auxilio no cumprimento das normas. Assim os autores propõe integrar a proposta dos artefatos organizacionais do ORA4MAS artefatos relacionados à reputação. Esses artefatos seriam responsáveis por publicar avaliações das performances dos agentes em relação ao seu comportamento dentro da organização. A avaliação dos agentes seria em relação tanto a normas quanto em relação à contribuição para realização de tarefas coletivas executadas na organização. A ideia da proposta é que um agente seja constantemente avaliado pela organização em relação aos papéis que ele assume e o resultado desta avaliação fique disponível aos demais membros.

Com essa avaliação, os demais agentes membros podem construir a reputação de determinado agente que participa da organização. Em função de a reputação influenciar no processo de decisão, os agentes tenderiam a manter uma reputação satisfatória. O trabalho, no entanto descreve apenas o processo de tornar a avaliação de determinado agente disponível aos demais membros da organização utilizando um artefato de reputação e não como a reputação de determinado agente é construída e utilizada no processo decisório.

Para exemplificar a utilização do artefato de reputação, os autores utilizam a abordagem ORA4MAS para modelar um conjunto de artefatos segundo o modelo Moise+ apresentado na Seção 2.1. Na modelagem existem três tipos de artefatos: *group*, *schema* e *reputation*. Artefatos do tipo *group* mantém o estado de uma instância de um grupo disponibilizando operações relacionadas a este grupo. Artefatos do tipo *schema* provem operações relacionadas ao esquema, como por exemplo, comprometimento com missões. O artefato do tipo *reputation* criado possui ligações com todos os artefatos organizacionais e pode ser observado pelos agentes que participam da organização. Os demais artefatos organizacionais informam ao artefato de reputação o estado corrente da organização e essa informação permite calcular a avaliação dos agentes. Essa avaliação é disponibilizada aos membros da organização através de propriedades observáveis. A avaliação dos agentes é calculada em relação ao contexto dos papéis e missões que estão engajados.

A avaliação utiliza os seguintes critérios: obediência, proatividade e resultado. Obediência está relacionado ao número de objetivos que o agente é obrigado a alcançar. Proatividade está relacionado número de objetivos que o agente alcança em determinado esquema sem ser obrigado. Resultado esta relacionado ao número de esquemas que o agente participa que são executados com sucesso. Contudo este artefato de reputação prove uma avaliação que permite calcular a reputação de um agente e não a reputação do agente em si.

Segundo (HÜBNER et al., 2010), a proposta do ORA4MAS foi avaliada apenas para MOISE+ a ideia é que a proposta possa ser avaliada para outros modelos como: OperA(DIGNUM et al., 2002) , AGR (FERBER; GUTKNECHT; MICHEL, 2004), dentre outros. Outro ponto ainda a ser estudado é o processo de reorganização utilizando esta abordagem e o impacto desse processo sobre os artefatos organizacionais. Além disso, os autores pretendem estudar a definição de uma meta-organização para ORA4MAS onde existiriam papéis especiais para agentes organizacionais acessarem artefatos organizacionais.

5.2 JaCaMo

A plataforma JaCaMo (BOISSIER et al., 2011) é o resultado da integração das seguintes tecnologias: Jason[1] para programação dos agentes, CArtAgO[2] para programação do ambiente baseado no metamodelo A&A e Moise [3] para definição da organização.

As três dimensões: agente (*agent*), ambiente (*environment*) e organização (*organization*), são integradas de modo a simplificar o modelo de programação adotado no desenvolvimento de sistemas multiagentes complexos. Para interação entre os agentes, a plataforma JaCaMo utiliza a comunicação da plataforma Jason baseada em atos de fala.

A conexão entre agente e ambiente (A-E) acontece a partir do mapeamento das ações externas dos agentes para as operações nos artefatos, e das propriedades observáveis dos artefatos para percepção dos agentes. Agentes podem realizar ações, caso exista um artefato que disponibiliza essa operação. As propriedades observáveis dos artefatos que o agente observa são mapeadas diretamente para crença dos agentes.

A conexão entre organização e ambiente (O-E) acontece em função da infraestrutura organizacional ser parte do ambiente baseado em artefatos no qual os agentes estão situados. Nesta abordagem os artefatos organizacionais presentes no ambiente provem operações usadas por um agente da aplicação para fazer parte da organização e também para que agentes organizacionais possam gerenciar essa organização.

A conexão entre agente e organização (A-O) acontece a partir do mapeamento de objetivos definidos na dimensão organizacional para os objetivos individuais do agente. A delegação dos objetivos da organização para os agente é feito a partir de obrigações.

A dimensão organizacional do JaCaMO é modelada em artefatos organizacionais segundo a abordagem ORA4MAS. Esses artefatos organizacionais são responsáveis por encapsular o comportamento organizacional especificado em Moise. Os artefatos organizacionais em JaCaMo são: OrgBoard, GroupBoard e SchemeBoard.

OrgBoard é utilizado para acompanhar o estado atual da entidade organizacional. GroupBoard é utilizado para gerenciar o ciclo de vida de um grupo. Esse artefato prove

[1] http://jason.sourceforge.net/
[2] http://cartago.sourceforge.net/
[3] http://moise.sourceforge.net/

operações que permite que os agentes adotem e deixem papéis, adicione ou remova esquemas sociais. SchemeBoard é um artefato que é conectado ao seu correspondente artefato GroupBoard. Os agentes que participam deste grupo estão envolvidos em alcançar os objetivos gerenciados por seu esquema. Esse artefato prove operações que permitem aos agentes tanto se comprometerem e deixarem missões como informar o alcance de algum objetivo.

Além das ações os artefatos organizacionais possuem propriedades observáveis que permitem que os agentes tenham acesso ao estado da organização. O artefato GroupBoard, por exemplo, permite que os agentes, a partir das propriedades observáveis verifiquem os papéis disponíveis em um grupo.

Os artefatos organizacionais adotam um comportamento organizacional especificado em Moise. Para que essa adoção seja possível as especificações em Moise devem ser traduzidas para uma linguagem especifica denominada NOPL (*Normative Organisation Programming Language*). Os artefatos organizacionais possuem um interpretador para essa linguagem capaz de prover a infraestrutura organizacional necessária para gerenciar a organização.

A proposta inicial da plataforma JaCaMo não contempla o processo de reorganização. Contudo o trabalho desenvolvido por Sorici (2012) apresenta uma infraestrutura que é uma extensão da plataforma JaCaMo para gerenciar o processo e reorganização.

A extensão da plataforma JaCaMo para reorganização utiliza uma especificação organizacional definida para coordenar e supervisionar o processo de reorganização. Essa especificação usa Moise OML para especificar as dimensões: estrutural, funciona e normativa do processo de reorganização.

A execução do processo de reorganização é gerenciada por um artefato do tipo GroupBoard denominado *Reorganziation Group* que permite que agentes adotem/deixem papéis definidos no grupo para reorganização. Também é criado um artefato do tipo SchemeBoard denominado *Reorganization Social Scheme* para que os agentes possam se comprometer com as missões e alcançar objetivos de acordo com a especificação. Em suma esses artefatos provem operações para o gerenciamento da execução do processo de reorganização. As operações providas por esses artefatos que modificam tanto a especificação organizacional (OS) como a entidade organizacional (OE) precisam ser coordenadas. Essa coordenação refere-se a agentes adotando determinado papel ou comprometido com alguma missão que devem deixar o papel ou abandonar a missão em algum momento. A sequência que essas ações devem ser realizadas é dependente ou da aplicação, ou de características decorrentes da organização.

Em função disso, para auxiliar os agentes no alcance dos objetivos referentes ao processo de reorganização é criado um artefato especial denominado ReorgBoard. Esse artefato auxilia no processo de reorganização coordenando os passos que devem ser seguidos para que mudanças sejam realizadas nos níveis de especificação e de entidade organizacional. Esses passos constituem o que os autores denominam de *implementation plan*. Este plano é armazenado no artefato ReorgBoard sendo acessado pelos agentes, que fazem parte do grupo responsável pela reorganização, através das propriedades observáveis do

artefato.

5.3 OrIAs X ORA4MAS X JaCaMo

Este capítulo destina-se a comparar as semelhanças e diferenças entre a abordagem ORA4MAS e a plataforma JaCaMo com a infraestrutura proposta OrIAs.

Assim como a dimensão organizacional da plataforma JaCaMo, que é baseada na abordagem ORA4MAS, OrIAs propõe uma infraestrutura de gerenciamento organizacional baseada no conceito de *embodied organizations* que permite integrar o conceito de agente, ambiente e organização.

Seguindo o proposto na abordagem ORA4MAS, OrIAs também trabalha com dois tipos de agentes: agentes organizacionais (responsáveis pelo gerenciamento e criação da infraestrutura) e agentes de aplicação (agentes adotam papéis na organização).

Apesar de existir uma infraestrutura de comunicação proposta para a plataforma JaCaMo apresentada na Subseção 3.2.6, na proposta original da plataforma, para interação dos agentes é utilizada a mesma comunicação baseada em atos de fala do Jason. Em contrapartida OrIAs integra aos artefatos organizacionais uma infraestrutura de comunicação que permite que agentes implementados em diferentes linguagens sejam capaz de se comunicarem desde que tenham acesso aos artefatos da infraestrutura de comunicação. Além disso, os artefatos de comunicação da infraestrutura permitem que sejam criados protocolos específicos para comunicação dos agentes.

Assim como a abordagem ORA4MAS, OrIAs garante que as restrições da especificação organizacional sejam seguidas a partir da utilização de normas. Os artefatos desenvolvidos em OrIAs impedem que um agente realize ações que não são permitidas (como por exemplo, adotar papéis incompatíveis, adotar papéis que não estão mais disponíveis, entre outras) ao mesmo tempo fornece mecanismos que permite ao que um agente organizacional consiga detectar a violação das normas.

Ao contrário das normas utilizadas na plataforma JaCaMo que estão diretamente relacionadas ao modelo organizacional Moise (normas relacionadas a permissões e obrigações entre papéis e missões), as normas em OrIAs são mais genéricas e estão relacionadas ao acesso as operações providas pelos artefatos organizacionais, o que permite controlar quais agentes tem acesso a quais operações dos artefatos. Além disso, OrIAs permite que esse controle também seja estendido aos artefatos de aplicação e de comunicação. Ou seja, basta especificar quais são as restrições de acesso no arquivo de descrição da organização, fornecido para infraestrutura, para que esses artefatos sejam controlados por agentes organizacionais.

Outro ponto a ser mencionado é que OrIAs permite que as normas sejam criadas tanto a partir da especificação organizacional como pelo agente organizacional a partir do artefato de normas disponibilizado. Na versão inicial da infraestrutura são criadas normas apenas para o controle de acesso aos artefatos, contudo é possível que sejam criados normas para controlar outras características.

Para auxiliar tanto no cumprimento de normas quanto na escolha de parceiros para realização das tarefas dos agentes, OrIAs utiliza artefatos de reputação. Ao contrário do artefato de reputação proposto para a abordagem ORA4MAS, que disponibiliza uma avaliação dos agentes para que então possa ser construída a reputação sobre essa avaliação, os artefatos de reputação da infraestrutura OrIAs permitem que os membros da organização tenham acesso a reputação de determinado agentes em relação as normas e delegação de tarefas. Além disso, no projeto da infraestrutura é mostrado como os agentes podem utilizar o valor de reputação disponível dos demais membros da organização no seu processo decisório.

JaCaMo possui artefatos organizacionais que contemplam as características do modelo Moise, No artigo que propõe ORA4MAS é apresentada uma modelagem da abordagem em artefatos. Contudo o projeto desses artefatos está atrelado ao modelo Moise. Um dos trabalhos que os autores se propõem a realizar é avaliar a proposta da utilização dessa abordagem para outros modelos organizacionais. Contudo para avaliação em outros modelos, é necessário projetar novos artefatos na infraestrutura que sejam compatíveis com as características de cada modelo que se deseja avaliar.

OrIAs é uma infraestrutura que se baseia em quatro dimensões organizacionais: estrutural, funcional, dialógica e normativa, para o projeto dos artefatos. Além disso, permite que organização seja descrita em função de cada dimensão. Dessa forma OrIAs permitiria descrever uma organização que fosse baseada em determinados modelos organizacionais existentes. Mesmo que estes modelos não contemplem as quatro dimensões organizacionais citadas.

Por exemplo, OrIAs permitiria descrever uma organização baseada no modelo Steam (que possui as dimensões estrutural e funcional). A descrição da infraestrutura permitiria descrever os papéis e objetivos de um time. Já um o plano compartilhado de um time Steam, pode ser descrito em OrIAs como um grupo que possui um plano coletivo. Sub-times podem ser descritos como subgrupos ligados a este grupo, entre outras características. Para AGR (que possui apenas a dimensão estrutural) podem ser especificadas na dimensão estrutural da descrição da organização de OrIAs os papéis que podem ser adotados e os grupos que podem ser criados. Para o modelo Moise (que possui as dimensões estrutural, funcional, dialógica e normativa) missões podem ser descritas em OrIAs como os objetivos individuais de determinado papel, esquemas podem ser mapeados para grupos, a dimensão deontica pode ser mapeada para OrIAs a partir da criação de normas (que podem ser criadas no artefato de normas presente na infraestrutura) para definir as permissões e obrigações de um papel sobre uma missão (definida como objetivo individual em OrIAs).

Diferentemente das outras abordagens, na adoção de um papel em OrIAs um agente recebe os objetivos e planos para os objetivos do papel. O agente não precisa ter em sua base de planos um plano para determinado objetivo que adotou. Contudo caso o agentes possua o plano para o objetivo pode optar por utilizar este plano. OrIAs permite também fazer a distinção entre objetivos de realização e objetivos de manutenção, além de permitir a definição de relações de dependências entre papéis e grupos.

Em OrIAs um agente quando adota o papel na organização passa a fazer parte de um grande grupo denominado organização geral. Portanto a infraestrutura permite que agentes possam executar objetivo específicos do papel adotado sem necessariamente estarem inseridos em um grupo especifico criado para realização de uma tarefa coletiva.

Diferentemente de JaCaMo, onde os agentes são implementados em Jason, a proposta de OrIAs é permitir que agentes implementados em diferentes plataformas possam adotar a mesma organização, se comunicando utilizando a infraestrutura de comunicação integrada a infraestrutura, desde que existam pontes para essas plataformas disponibilizadas pelo CArtAgO.

A infraestrutura OrIAs permite ainda realizar o processo de reorganização. O processo é menos complexo do que o realizado pela extensão da plataforma JaCaMo apresentada que realiza o processo de reorganização. Em OrIAs não existe um artefato dedicado ao processo de reorganização, nem a antiga especificação da organização é modificada. O agente com o papel organização que tem como função aceitar ou não a solicitação de mudança organizacional por um agente da aplicação. Caso o agente organizacional aceite é necessário apenas integrar e/ou modificar a organização que fica disponível aos agentes a partir dos artefatos.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou OrIAs (***Or**ganizational **I**nfrastructure for **A**gents***) uma infraestrutura de nível micro-organizacional baseada em artefatos para sistemas multiagentes.

OrIAs suporta as quatro dimensões organizacionais: estrutural, funcional, dialógica e normativa. A dimensão estrutural da infraestrutura permite definir os papéis e grupos presentes na organização com suas respectivas características. A dimensão funcional permite descrever toda dinâmica sobre a dimensão estrutural (comportamento da organização). A dimensão dialógica permite estabelecer padrões de interação entre os agentes onde são estabelecidos os protocolos de comunicação utilizados. A dimensão normativa permite restringir o comportamento dos agentes a partir da utilização de normas para o controle de acesso aos recursos e criação de grupos.

No projeto da infraestrutura de gerenciamento organizacional OrIAS foram propostos novos elementos a serem agregados para que fosse possível melhor conciliar a autonomia dos agentes e também suprir suas limitações individuais. Assim a infraestrutura fornece suporte para o raciocínio social dos agentes em termos de relações de dependência entre os papéis organizacionais que os agentes assumem e entre grupos de papéis. A infraestrutura utiliza um mecanismo de reputação para auxiliar os agentes tanto na busca de parceiros para o alcance dos objetivos individuais e coletivos como também para auxiliar no processo de aplicação das normas permitindo que normas sejam criadas, atualizadas e excluídas dinamicamente. A infraestrutura permite ainda distinguir objetivos de alcance (*achievement goals*) de objetivos de manutenção (*maintenance goals*). Além disso, a infraestrutura de comunicação proposta por Rodrigues, Rocha Costa e Dimuro (2013) é integrada a infraestrutura OrIAs de forma a facilitar a interação entre os agentes. OrIAs permite realizar um processo de reorganização não institucionalizado realizado por iniciativa dos agentes. Além disso, na infraestrutura, os artefatos de ambiente são utilizados como intermediários entre a organização e o ambiente.

Ainda existem muitos aspectos que podem ser explorados a partir do protótipo inicial da infraestrutura de gerenciamento organizacional proposta. Dentre as melhorias que podem ser realizadas em trabalhos futuros são listadas algumas sugestões agrupadas de acordo com teorias presentes na infraestrutura:

- Relação de dependência entre os agentes: na infraestrutura são consideradas apenas dependências em relação a objetivos. Existe a possibilidade de estender os mecanismos de dependência utilizados na infraestrutura para trabalhar com dependências

relacionadas a ações e recursos. Além disso, para disponibilizar a estrutura que armazena as dependências dos agentes o artefato precisa ter acesso aos planos dos agentes, ou seja, necessita ter acesso a informações privadas desses agentes[1].

- Reputação: na infraestrutura é considerado que os agentes são honestos e transmitem sempre uma reputação correta dos agentes os quais interagiram. Contudo é possível que seja simulada a difusão de reputações distorcidas permitindo analisar como uma reputação errônea sobre determinado agente influência nas suas relações sociais.

- Reorganização: na infraestrutura é apresentado um processo inicial de reorganização, onde a mesma ocorre de forma emergente e pela solicitação dos próprios agentes ao agente organizacional. No entanto, podem ser estudados meios para realizar uma reorganização controlada, onde a organização possa ser monitorada identificando quando o processo de reorganização é necessário. Alguns elementos presentes na infraestrutura podem auxiliar nesse processo. A reputação de delegação, por exemplo, pode ser utilizada para identificar quando os agentes não estão mais conseguindo satisfazer seus objetivos, assim como a reputação quanto ao cumprimento de normas pode auxiliar na detecção de normas que precisam ser modificadas.

- Agentes heterogêneos: apesar de CArtAgO fornecer pontes (*bridges*) para diferentes plataformas de agentes, permitindo que agentes heterogêneos adotem a mesma organização, os testes realizados com infraestrutura foram realizados apenas com agentes implementados utilizando a plataforma Jason.

- Mapeamento dos modelos organizacionais: Apesar de ser possível que a especificação de alguns modelos organizacionais, presentes na literatura, possa ser mapeada para a especificação que serve como entrada para a infraestrutura OrIAs, os testes realizados foram apenas para explorar as funcionalidades providas pela infraestrutura não tendo sido realizados testes cuja especificação organizacional segue algum modelo definido.

[1]Essa característica poderia, então, ser uma limitação para aplicações onde essa informação privada não pode ser consultada pelos demais agentes da sociedade.

REFERÊNCIAS

ALONSO, E. How individuals negotiate societies. In: INTERNATIONAL CONFERENCE ON MULTI AGENT SYSTEMS, 1998. P18-25. **Proceedings...**, 1998. [S.l.:s.n].

ARGENTE, E.; BOISSIER, O.; ESPARCIA, S. Describing Agent Organisations. In: **Agreement Technologies**. Springer, 2013. p.253–275.

BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. **Developing multi-agent systems with JADE**. Wiley. com, 2007. v.7.

BOELLA, G.; VAN DER TORRE, L.; VERHAGEN, H. Introduction to normative multiagent systems. **Computational & Mathematical Organization Theory**, v.12, n.2-3, p.71–79, 2006.

BOISSIER, O.; BORDINI, R. H.; HÜBNER, J. F.; RICCI, A.; SANTI, A. Multi-agent oriented programming with JaCaMo. **Science of Computer Programming**, v.78, n.6, p.747-761, 2011.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **Uml: guia do usuário**. Elsevier Brasil, 2006.

BORDINI, R.; HÜBNER, J.; WOOLDRIDGE, M. **Programming multi-agent systems in AgentSpeak using Jason**. Wiley-Interscience, 2007. v.8.

CARLEY, K. M.; GASSER, L. Computational organization theory. **Multiagent systems: A modern approach to distributed artificial intelligence**, p.299–330, 1999.

CASTELFRANCHI, C. The micro-macro constitution of power. Protosociology, v.30, p.208-268, 2003.

CASTELFRANCHI, C.; FALCONE, R. Principles of trust for MAS: cognitive anatomy, social importance, and quantification. In: MULTI AGENT SYSTEMS, 1998. PROCEEDINGS. INTERNATIONAL CONFERENCE ON. 1998. p.72–79.

CASTLEFRANCHI, C.; MICELI, M.; CESTA, A. Dependence relations among autonomous agents. **ACM SIGOIS Bulletin**, v.13, n.3, p.14, 1992.

CONTE, R.; CASTELFRANCHI, C.; VENEZIANO, V. The computer simulation of partnership formation. **Computational & Mathematical Organization Theory**, v.4, n.4, p.293–315, 1998.

CONTE, R.; PAOLUCCI, M. **Reputation in artificial societies**: social beliefs for social order. Springer, 2002. v.6.

CONTE, R.; PEDONE, R. Finding the best partner: the part-net system. In: MULTI-AGENT SYSTEMS AND AGENT-BASED SIMULATION. 1998. p.156–168.

COSTA, A. C. R.; DIMURO, G. P. A Minimal Dynamical Organization Model. In: DIGNUM, V. (Ed.). **Hanbook of Multi-Agent Systems**: semantics and dynamics of organizational models. Hershey: IGI Global, 2009. p.419–445.

COUTINHO, L. R. **Interoperabilidade organizacional em sistemas multiagentes abertos baseada em engenharia dirigida por modelos.** 2009. 253p. Tese (Doutorado em Ciência da Computação) — Escola Politécnica da Universidade de São Paulo.

COUTINHO, L. R.;SICHMAN, J. S.; BOISSIER, O. Modeling organization in mas: a comparison of models. In: WORKSHOP ON SOFTWARE ENGINEERING FOR AGENT-ORIENTED SYSTEMS, SEAS, 1, 2005. **Proceedings...** 2005. [S.l.:s.n].

DAVID, N.; SICHMAN, J. S.; COELHO, H. Extending social reasoning to cope with multiple partner coalitions. In: **Multi-Agent System Engineering**. Springer, 1999. p.175–187.

DECKER, K. S. **Task environment centered simulation**. Simulating Organizations: Computational Models of Institutions and Groups. AAAI Press/MIT Press, 1996.

DEMAZEAU, Y.; SICHMAN, J. S. On social reasoning in multi-agent systems. **Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial**, v.5, n.13, p.68–84, 2001.

DIGNUM, V.; DIGNUM, F.; SONENBERG, L. Towards dynamic reorganization of agent societies. In: WORKSHOP ON COORDINATION IN EMERGENT AGENT SOCIETIES. ECAI, 2004. P.68-84. **Proceedings...** 2004. [S.l.:s.n].

DIGNUM, V.; MEYER, J-J.; WEIGAND, H.; DIGNUM, F. An Organization-oriented Model for Agent Societies. In: RASTA, 2002. **Proceedings...** [S.l.:s.n].

DUFF, S.; THANGARAJAH, J.; HARLAND, J. Maintenance goals in intelligent agents. **Computational Intelligence**, 2012.

FALCONE R.; CASTELFRANCHI, C.; CARDOSO, HL.; JONES, A.; OLIVEIRA, E. Norms and Trust. In: **Agreement Technologies**. Springer, 2013. p.221–231.

FERBER, J.; GUTKNECHT, O. A meta-model for the analysis and design of organizations in multi-agent systems. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, 1998. p. 128-135. **Proceedings...** 1998, [S.l.:s.n].

FERBER, J.; GUTKNECHT, O.; MICHEL, F. From agents to organizations: an organizational view of multi-agent systems. In: **Agent-Oriented Software Engineering IV**. Springer, 2004. p.214–230.

FININ, T.; FRITZSON, R.; MCKAY, D.; MCENTIRE, R. KQML as an agent communication language. In: INFORMATION AND KNOWLEDGE MANAGEMENT. 1994. p.456–463.

FIPA. **FIPA Contract Net Interaction Protocol Specification**. [Online; Acessado em janeiro de 2014], http://www.fipa.org/specs/fipa00029/.

FIPA. **FIPA Request Interaction Protocol Specification**. [Online; Acessado em janeiro de 2014], http://www.fipa.org/specs/fipa00026/.

HANNOUN, M.; BOISSIER, O.; SICHMAN, J. S. MOISE: an organizational model for multi-agent systems. In: **Advances in Artificial Intelligence**. Springer, 2000. p.156–165.

HÜBNER, J. F. **Um modelo de reorganização de sistemas multiagentes**. 2003. 246p. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.

HÜBNER, J. F.; BOISSIER, O.; VERCOUTER, L. Instrumenting multi-agent organisations with reputation artifacts. In: COORDINATION, ORGANIZATION , INSTITUTIONS, AND NORMS, 2008. p. 17-24. **Proceedings...** 2008. [S.l.:s.n].

HÜBNER, J. F.; BOISSIER, O.; KITIO, R.; RICCI, A. Instrumenting multi-agent organisations with organisational artifacts and agents. **Autonomous Agents and Multi-Agent Systems**, v.20, n.3, p.369–400, 2010.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: **Advances in artificial intelligence**. Springer, 2002. p.118–128.

HÜBNER, J.; SICHMAN, J. SACI: uma ferramenta para implementação e monitoração da comunicação entre agentes. In: INTERNATIONAL JOINT CONFERENCE IBERO-AMERICANA ON AI, 7; BRAZILIAN SYMPOSIUM ON AI, 15, 2000. p. 47-56. **Proceedings...** 2000. [S.l.:s.n].

KINNY, D.; GEORGEFF, M.; RAO, A. A methodology and modelling technique for systems of BDI agents. **Agents breaking away**, p.56–71, 1996.

LÓPEZ, F. L. y; LUCK, M.; DINVERNO, M. A normative framework for agent-based systems. **Computational & Mathematical Organization Theory**, v.12, n.2-3, p.227–250, 2006.

MORGADO, L.; GASPAR, G. A social reasoning mechanism based on a new approach for coalition formation. In: OF THE . Department of Informatics: University of Lisbon, 2000.

OMICINI, A.; RICCI, A.; VIROLI, M. Artifacts in the A&A meta-model for multi-agent systems. **Autonomous Agents and Multi-Agent Systems**, v.17, n.3, p.432–456, 2008.

PIUNTI, M. ; RICCI, A.; BOISSIER, O. Embodied organisations in MAS environments. In: **Multiagent System Technologies**. Springer, 2009. p.115–127.

PIUNTI, M.; RICCI, A.; BOISSIER, O.; HÜBNER, J. F. Programming Open Systems with Agents, Environments and Organizations. In: WOA, 2010. **Proceedings...** [S.l.:s.n]

PIUNTI, M.; RICCI, A. Cognitive use of artifacts: exploiting relevant information residing in mas environments. In: **Knowledge Representation for Agents and Multi-Agent Systems**. Springer, 2009. p.114–129.

RAO, A. S. AgentSpeak(L): bdi agents speak out in a logical computable language.) In: MAAMAW, 1996. p. 42-55 **Proceedings...** [S.l.:s.n]

REGO MONTEIRO, J. de Lima do; SICHMAN, J. PartNET++: simulating multiple agent partnerships using dependence graphs. **Multi-Agent-Based Simulation VI**, p. 14–23, 2006.

RICCI, A.; PIUNTI, M.; VIROLI, M.; OMICINI, A. Environment programming in CArtAgO. **Multi-Agent Programming: Languages, Tools and Applications**, p.259–288, 2009.

RICCI, A.; PIUNTI, M.; VIROLI, M. Environment programming in multi-agent systems: an artifact-based perspective. **Autonomous Agents and Multi-Agent Systems**, v.23, n.2, p.158–192, 2011.

RODRIGUES, M. R. **Um sistema de valores de troca para suporte às interações em sociedades artificiais.** 2003. 134p.

RODRIGUES, T. F.; ROCHA COSTA, A. C. da; DIMURO, G. P. A Communication Infrastructure Based on Artifacts for the JaCaMo Platform. In: INTERNATIONAL WORKSHOP ON ENGINEERING MULTI-AGENT SYSTEMS, EMAS/AAMAS, 2013. **Proceedings...** [S.l.:s.n]

SANTOS, I. A. d. S. dos; ROCHA COSTA, A. C. da. Toward a Framework for Simulating Agent-Based Models of Public Policy Processes on the Jason-CArtAgO Platform. In: INTERNATIONAL WORSHOP ON AGENT-BASED MODELING FOR POLICY ENGINEERING, AMPLE, 2012. p. 45. **Proceedings...** 2012. [S.l.:s.n].

SICHMAN, J. S. Depint: dependence-based coalition formation in an open multi-agent scenario. **Journal of Artificial Societies and Social Simulation**, v.1, n.2, p.1998, 1998.

SICHMAN, J. S. **Raciocínio Social e Organizacional em Sistemas Multiagentes: avanços e perpectivas.** 2003. 258p. Tese (Tese de Livre-Docência — Escola Politécnica da Universidade de São Paulo.

SICHMAN, J. S.; CONTE, R. Multi-Agent Dependence by Dependence Graphs. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS, AAMAS, 2002. p. 483-490. **Proceedings...**: [S.l], Acm Press, 2002.

SICHMAN, J. S. et al. A Social Reasoning Mechanism Based On Dependence Networks. In: ACM PRESS. John Wiley and Sons, 1994. p.188–192.

SMITH, R. G. The contract net protocol: high-level communication and control in a distributed problem solver. **IEEE Transactions on Computers**, v.100, n.12, p.1104–1113, 1980.

SORICI, A.; PICARD, G.; BOISSIER, O.; SANTI, A.; HÜBNER]. Multi-Agent Oriented Reorganisation within the JaCaMo infrastructure. In: INTERNATIONAL WORKSHOP ON INFRAESTRUCTURES AND TOOLS FOR MULTIAGENT SYSTEMS, ITMAS,3, 2012. p. 135-148. **Proceedings...** [S.l.:s.n].

TAMBE, M. Towards Flexible Teamwork. **Journal of Artificial Intelligence Research**, v.7, p.83–124, 1997.

WOOLDRIDGE, M. **Reasoning about rational agents**. MIT press, 2000.

WOOLDRIDGE, M. **An introduction to multiagent systems**. Wiley, 2009.

APÊNDICE A CÓDIGOS DOS AGENTES

Código A.1: Trecho de código do agente organizacional referente a criação dos artefatos da infraestutura.

```
/* Initial beliefs and rules */

/* Initial goals */
!start.

/* Plans */
+!start : true <-
     .print("hello_world.");
     !createOrganizationalArtifacts;
     !createApplicationArtifacts.

+!createOrganizationalArtifacts: true <-
     makeArtifact("generalOrg","organizationalArtifacts.GeneralOrgArtifact",["eleicao.xml"],IdGenOrg)
          ;
     focus(IdGenOrg); makeArtifact("roles","organizationalArtifacts.RolesArtifact",[],IdRoles);
     focus(IdRoles);
     makeArtifact("roleDep","organizationalArtifacts.RoleDEPArtifact",[],IdRoleDep);
     focus(IdRoleDep);
     makeArtifact("groupDep","organizationalArtifacts.GroupDEPArtifact",[],IdGroupDep);
     focus(IdGroupDep);
     makeArtifact("FIPARequestArtifact", "protocolArtifacts.FIPARequestArtifact",[],IdReqArt);
     focus(IdReqArt);
     makeArtifact("delegationReputation","organizationalArtifacts.DelegationReputationArtifact",[],
          IdDelRep);
     focus(IdDelRep);
     makeArtifact("normReputation","organizationalArtifacts.NormReputationArtifact",[],IdNormRep);
     focus(IdNormRep);
     createArtifacts[artifact_name("FIPARequestArtifact")];
     subscribe[artifact_name("FIPARequestArtifact")];
     .broadcast(tell,subscribeArtifact("FIPARequestArtifact"));
     makeArtifact("FIPAContractNetArtifact","protocolArtifacts.FIPAContractNetArtifact",[],
          IdContractNetArt);
     focus(IdContractNetArt);
     createArtifacts[artifact_name("FIPAContractNetArtifact")];
     subscribe[artifact_name("FIPAContractNetArtifact")];
     .broadcast(tell,subscribeArtifact("FIPAContractNetArtifact"));
     linkArtifacts(IdRoleDep,"out-roleDep-roles",IdRoles);
     linkArtifacts(IdGenOrg,"out-genOrg-delegationRep",IdDelRep);
     linkArtifacts(IdGenOrg,"out-genOrg-normRep",IdNormRep);
     linkArtifacts(IdGenOrg,"out-genOrg-roles",IdRoles);
     linkArtifacts(IdRoleDep,"out-roleDep-rep",IdDelRep);
     linkArtifacts(IdReqArt,"out-request",IdDelRep);
     linkArtifacts(IdDelRep,"out-delegationReputation",IdRoles);
     .broadcast(tell,focusArtifact("roles"));
     .broadcast(tell,focusArtifact("generalOrg"));
     makeArtifact("NormArtifact", "organizationalArtifacts.NormArtifact", [],
     IdNormArtifact); focus(IdNormArtifact);
     linkArtifacts(IdGenOrg,"out-genOrg-norms",IdNormArtifact);
     .broadcast(tell,focusArtifact("NormArtifact")).

+!createApplicationArtifacts: true <-
     makeArtifact("electronicBallot","applicationArtifacts.ElectronicBallot",[],IdElectronicBallotg);
     .broadcast(tell,focusArtifact("electronicBallot"));
     makeArtifact("financialSupport","applicationArtifacts.FinancialSupport",[],IdFinancialSupport);
     .broadcast(tell,focusArtifact("financialSupport")).
```

Código A.2: Trecho de código que o agentes que participam da organização devem incluir para o tratamento dos eventos advindos dos artefatos da infraestrutura.

```
//common for all agents

/*adop/leave role */
+!adoptNewRole(Role) : true <-
    .wait(3000);
     adoptRole(Role)[artifact_name("generalOrg")].

-!adoptNewRole(Role)[error_msg(Msg)] : true <-
```

```
     .puts("#{Msg}");
     !adoptNewRole(Role).

+!leaveRole(Role) : true <-
     leaveRole(Role)[artifact_name("generalOrg")].

-!leaveRole(Role)[error_msg(Msg)] : true  <-
     .puts("#{Msg}").

+addPlanOfRole(Plan) : true <-
     .add_plan(Plan).

+removePlanOfRole(PlanTrigger): true <-
     .concat("{ ",PlanTrigger," }",S);
     .term2string(T,S);
     .relevant_plans(T, _, L);
     .remove_plan(L).

+startGoal(Goal): true <-
     .term2string(T,Goal);
     !T.

+add_beliefBase(Belief) : true <-
     .term2string(T,Belief);
     +T.

+add_beliefUpdate(B) : true <-
     .add_plan(B).

/*recognize artifacts */
+subscribeArtifact(ArtName) : true <-
     subscribe[artifact_name(ArtName)];
     lookupArtifact(ArtName,Id);
     focus(Id).

+focusArtifact(ArtName) : true <-
     lookupArtifact(ArtName,Id);
     focus(Id).

//// communication ////
//REQUEST PROTOCOL used for DEPENDENCE RELATION
//informs the agent about a new Request message (DELEGATION REQUEST), on his/her queue of the
     conversation
//this event is created on the request protocol

+newrequestMessage(S, C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     .count(newrequestMessage(_,_),N);
     if(N == 0){
          !request(S,C,M,A);
     }.

+!request(S,C,M,A) : true <-
     +newrequestMessage(S, C);
     .term2string(Msg,M);
     .puts("DELEGATION REQUEST FOR GOAL #{M} from: #{S}");
     agree(S,C,M);
     !Msg[source(S)];
     inform_done(S,C,M)[artifact_id(A)];
     .wait(2000);
     .abolish(newrequestMessage(_,_));.

-!request(S,C,A)[error_msg(Msg)] : true  <- .puts("#{Msg}").

//informs the agent about a new Agree message (ACCEPTANCE OF DELEGATION), on his/her queue of the
     conversation

+newagreeMessage(S, C)[artifact_id(A)] : true <-
     +newagreeMessage(S, C);
     getMessage(C, M)[artifact_id(A)];
     .puts("ACCEPTANCE OF DELEGATION from: #{S} FOR GOAL #{M}").

//informs the agent about a new Inform message (INFORMATION ABOUT GOAL ACHIEVEMENT), on his/her queue
      of the conversation

+newinformMessage(S, C)[artifact_id(A)] : true <-
     +newinformMessage(S, C);
     getMessage(C, M)[artifact_id(A)];
     .puts("GOAL #{M} ACHIEVED from #{S}");
     internalAction.int_random(10,R);
     if(R>=5){
          increaseDelegationResultRep(S,M);
     }
     else{
          decreaseDelegationResultRep(S,M);
     }.

+newrefuseMessage(S, C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     .puts("GOAL #{M} REFUSED from #{S}").

/*group creation, formation and running */

+!createGroup(GroupName): true <-
     createGroup(GroupName,Id);
     focus(Id);
     getListMembersOfPlan(GroupName)[artifact_id(Id)];.

+groupInfo(GroupName,IdGroup) : true <- +manager(GroupName,IdGroup).
```

```
+execute(Step,Type,Responsable,ConversationId,Manager,ArtifactName)[artifact_id(A)] : true <-
     if(Type == "individualGoal"){
          .term2string(T,Step);
          !T;
          .concat(Step,";",ArtifactName,Msg);
          inform_done(Manager,ConversationId,Msg)[artifact_name("FIPAContractNetArtifact")];
     }
     if(Type == "collectiveGoal"){
          .count(committedWithGroup(ArtifactName,Responsable,Step),N);
          .concat(Step,";",ArtifactName,Msg);
          if(N > 0){
               runPlanDep[artifact_name(Responsable)];
               inform_done(Manager,ConversationId,Msg)[artifact_name("FIPAContractNetArtifact")];
          }
     }.

+executedStep(Group, Step, NumberOfAgents) : true <- !executed(Group,Step, NumberOfAgents).

+!executed(Group,Step, NumberOfAgents) : true <-
     .count(informDone(_,Step,Group),N);
     if(NumberOfAgents == N){
          .abolish(informDone(_,Step,Group));
          .wait(10000);
          runPlan[artifact_name(Group)];
     }
     if(NumberOfAgents > N){
          .wait(3000);
          !executed(Group,Step, NumberOfAgents);
     }.

+endGroupPlan(Group) : true <- .puts("Plan of  group #{Group} finalized!").

+currentStep(Step,StepName,NumberOfSteps) : true <- +currentStep(Step,StepName,NumberOfSteps).

+numberOfGroupSteps(Group,NumberOfSteps): true <- +numberOfSteps(Group,NumberOfSteps).

+destroyGroup(GroupName) : true <-
     lookupArtifact(GroupName,GroupNameId);
     stopFocus(GroupNameId);
     .count(manager(GroupName,_),N);
     if(N>=1){
          -manager(GroupName,_);
     }.

//contract net protocol used for group formation

+setMembersOfPlan(ArtifactName,Roles,GroupDep,StartGoal)[artifact_id(ArtifactId)] : true <-
     .length(Roles,MaxRole);
     for(.range(I,0,MaxRole-1)){
          .nth(I,Roles,R);
          internalAction.split(R,";",StrList);
          .nth(0,StrList,Role);
          .nth(1,StrList,Step);
          .nth(2,StrList,NumberOfAgents);
          .term2string(Number,NumberOfAgents);
          !callForProposals(ArtifactName,Role,Step,Number);
     }
     if(GroupDep \== ""){
          !groupDependence(ArtifactName,GroupDep);
     }
     if(StartGoal){
          runPlan[artifact_id(ArtifactId)]
     }.

+!groupDependence(DependentGroup,Goal) : true <-
     getGroupDependence(Goal, GroupDependence);
     .length(GroupDependence,MaxGroup);
     for(.range(I,0,MaxGroup-1)){
          .nth(I,GroupDependence,Group);
          internalAction.split(Group,";",StrList);
          .nth(0,StrList,GroupName);
          .nth(1,StrList,GoalName);
          !callForProposalsOfGroup(DependentGroup,GroupName,GoalName);
     }.

+!callForProposalsOfGroup(DependentGroup,GroupName,GoalName): true <-
     getManagersOfGroups(GroupName,Managers);
     .my_name(Me);
     .term2string(Me,Name);
     if(.member(Name,Managers)){
          .puts("eu sou o manager");
          runPlan[artifact_name(GroupName)];
     }
     else{
          .length(Managers,MaxManagers);
          internalAction.int_random(1000000,R);
          .concat("C",R,ConversationId);
          .concat(GroupName,";",GoalName,GroupDep);
          for(.range(I,0,MaxManagers-1)){
          .nth(I,Managers,Ag);
          call_for_proposals(Ag,ConversationId,GroupDep);
          .puts("Send cfp of GROUP for GROUP : #{GroupName} with MANAGER : #{Ag} for theGOAL : #{
               GoalName}  ");
     }
     !waitProposals(ConversationId);
     .count(groupPropose(_,_,_,ConversationId),N);
     if(N == 0){
          .abolish(groupPropose(_,_,_,ConversationId));
```

```
          !callForProposalsOfGroup(DependentGroup,GroupName,GoalName);
     }
     if(N >= 1){
          !acceptGroupProposals(DependentGroup,GroupName,GoalName,ConversationId);
          .abolish(groupPropose(_,_,_,ConversationId));
     }
}.

+!acceptGroupProposals(DependentGroup,GroupName,GoalName,ConversationId) : true <-
     .findall(Ag,groupPropose(Ag,_,_,ConversationId),AgentList);
          .findall(Time,groupPropose(_,_,Time,ConversationId),TimeList);
     internalAction.array_multisort(TimeList,AgentList,SortTimeList,SortAgentList);
     lookupArtifact(DependentGroup,ArtifactId);
     .nth(0,SortAgentList,S);
     .nth(0,SortTimeList,T);
     .wait(T);
     .concat(DependentGroup,";",GroupName,";",GoalName,GroupDep);
     setStep_reponsable(GoalName,GroupName)[artifact_id(ArtifactId)];
     setConversationId(GroupName,ConversationId)[artifact_id(ArtifactId)];
     accept_proposal(S, ConversationId, GroupDep);.


+!callForProposals(ArtifactName,Role,Goal,NumberOfAgents): true <-
     getAgentsByRole(Role,Agents);
     .length(Agents,Max);
     internalAction.int_random(1000000,R);
     .concat("C",R,ConversationId);
     for(.range(I,0,Max-1)){
          .nth(I,Agents,Ag);
          call_for_proposals(Ag,ConversationId,Role);
          .puts("Send cfp to #{Ag}");
     }
     !waitProposals(ConversationId);
     .count(propose(_,_,ConversationId),N);
     if(N < NumberOfAgents){
          .abolish(propose(_,_,ConversationId));
          !callForProposals(ArtifactName,Role,Goal,NumberOfAgents);
     }
     if(N >= NumberOfAgents){
          !acceptProposals(Goal,ConversationId,ArtifactName);
          .abolish(propose(_,_,ConversationId));
     }.

+!waitProposals(Id) : true <-
     .wait(3000);
     end_proposal_phase(Id).

-!waitProposals(Id) : true <-  !waitProposals(Id).

+!acceptProposals(Goal,ConversationId,ArtifactName) : true <-
     .findall(Ag,propose(Ag,_,ConversationId),AgentList);
     .findall(Rep,propose(_,Rep,ConversationId),RepList);
     internalAction.array_multisort(RepList,AgentList,SortRepList,SortAgentList);
     .length(SortAgentList,Max);
     lookupArtifact(ArtifactName,ArtifactId);
     for(.range(I,0,Max-1)){
          .nth(I,SortAgentList,S);
          accept_proposal(S, ConversationId, ArtifactName);
          setStep_reponsable(Goal,S)[artifact_id(ArtifactId)];
          setConversationId(S,ConversationId)[artifact_id(ArtifactId)];
     }.

+newcall_for_proposalsMessage(S,C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     internalAction.split(M,";",StrList);
     .length(StrList,Size);
     //individual group proposal
     if(Size == 1){
          .puts("New call for proposals from #{S} relates to the role #{M}");
          .my_name(Name);
          getNormRepAgent(Name,NormReputation);
          getDelegationResultRepAgent(M, Name,DelResReputation);
          getDelegationProtocolRepAgent(M, Name,DelProtReputation);
          Reputation = (NormReputation + DelResReputation + DelProtReputation)/3;
          .puts("#{Reputation}");
          .concat("",Reputation,Rep);
          propose(S, C, Rep);
     }
     //group proposal to manager
     if(Size == 2){
          .nth(0,StrList,GName);
          .nth(1,StrList,GoalName);
          .puts("New call for proposals from #{S} relates to the Group #{GName} for theGoal: #{
               GoalName}");
          ?currentStep(Step,_,NumberOfSteps);
          if(Step == NumberOfSteps){
               .concat(GName,";",0,Timeout);
          }
          if(Step < NumberOfSteps){
               Time = (Step+2) * 10000;
               .concat(GName,";",Time,Timeout);
          }
          propose(S, C, Timeout);
     }.

+newproposeMessage(S,C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     internalAction.split(M,";",StrList);
     .length(StrList,Size);
```

```
     if(Size == 1){
          .puts("Propose message from #{S} with reputation: #{M}");
          .term2string(T1,S);
          .term2string(T2,M);
          +propose(T1,T2,C);
     }
     if(Size == 2){
          .nth(0,StrList,GName);
          .nth(1,StrList,TimeOut);
          .puts("Propose message from #{S} for group: #{GName} with timeout#{TimeOut}");
          .term2string(T1,S); .term2string(T2,GName);
          .term2string(T3,TimeOut);
          +groupPropose(S,GName,T3,C);
     }.

+newaccept_proposalMessage(S,C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     internalAction.split(M,";",StrList);
     .length(StrList,Size);
     if(Size == 1){
          .puts("Receive accept proposal from #{S} to join group artifact #{M}");
          +focusArtifact(M);
     }
     if(Size == 3){
          .nth(0,StrList,DepGroup);
          .nth(1,StrList,MyGroupName);
          .nth(2,StrList,GoalName);
          .puts("Receive accept proposal from #{DepGroup} to carry out a GOAL
          #{GoalName} through the group #{MyGroupName} "); +focusArtifact(DepGroup);
          +committedWithGroup(DepGroup,MyGroupName,GoalName);
     }.

+newreject_proposalMessage(S,C)[artifact_id(A)] : true <-   .puts("Reject proposal from #{S}").

+newinformCNMessage(S,C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     .puts("Inform done from #{S} about #{M}");
     internalAction.split(M,";",StrList);
     .nth(0,StrList,Step);
     .nth(1,StrList,Group);
     +informDone(S,Step,Group).

+informDone(S,Step,Group): true <- .puts("inform done add S #{S}, Step #{Step} ,Group #{Group}").

+newfailureCNMessage(S,C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     .puts("Failure from #{S} about #{M}").

+newrefuseCNMessage(S,C)[artifact_id(A)] : true <-
     getMessage(C, M)[artifact_id(A)];
     .puts("Refuse from #{S} about #{M}").
```

Código A.3: Código de um agente que deseja participar da organização.

```
{ include("common.asl") }

/* Initial goals */
!start.

/* Plans */
+!start : true <- !adoptNewRole("eleitor").
```

APÊNDICE B ESPECIFICAÇÃO ORGANIZACIONAL DO EXPERIMENTO REALIZADO

Código B.1: Especificação organizacional do estudo de caso *processo político de lobby através de contribuições para campanhas eleitorais*.

```
<?xml version="1.0" encoding="UTF-8"?>
<organizationalInfrastructure>
<structuralDimension>
   <roleDescription>
      <role role='politico'>
         <roleCardinality min='1' max='2'/>
         <goals>
            <goal goal='ganharEleicao' type='achievement' startGoal='false'/>
            <goal goal='conseguirVotos' type='achievement' startGoal='false'/>
            <goal goal='manterApoioFinaceiro' type='maintenance' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
      <role role='eleitor'>
         <roleCardinality min='1' max='5'/>
         <goals>
            <goal goal='votar' type='achievement' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='politico'/>
            <role role='lobista'/>
         </roleCompatibility>
      </role>
      <role role='lobista'>
         <roleCardinality min='2' max='4'/>
         <goals>
            <goal goal='financiarCampanha' type='achievement' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
      <role role='TSE'>
         <roleCardinality min='1' max='1'/>
         <goals>
            <goal goal='organizarEleicao' type='achievement' startGoal='true'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
   </roleDescription>
   <groupDescription>
      <group name='processoEleitoral'>
          <goal goal='realizarEleicao' type='collective' startGoal='true'/>
         <groupMembers>
            <role role='politico' goal='ganharEleicao'/>
         </groupMembers>
         </group>
   </groupDescription>
</structuralDimension>

<functionalDimension>
   <individualGoals>
      <plans>
         <plan trigger='ganharEleicao' context='true'>
            <step type='goal' responsible='lobista'>financiarCampanha</step>
            <step type='goal' responsible='politico'>conseguirVotos</step>
         </plan>
         <plan trigger='manterApoioFinaceiro' context='apoioFinanceiro(X) &amp; X &lt; 2'>
            <step type='action' responsible='politico'>getAgentsByRole("lobista",Agents)</step>
            <step type='action' responsible='politico'>.length(Agents,MaxAgents)</step>
            <step type='command' responsible='politico'>if(MaxAgents > 0){</step>
            <step type='goal' responsible='lobista'>financiarCampanha</step>
            <step type='command' responsible='politico'>}</step>
         </plan>
```

```
         <plan trigger='conseguirVotos' context='true'>
            <step type='action' responsible='politico'>getAgentsByRole("eleitor",Agents)</step>
            <step type='action' responsible='politico'>.length(Agents,MaxAgents)</step>
            <step type='command' responsible='politico'>if(MaxAgents > 0){</step>
            <step type='goal' responsible='eleitor'>votar</step>
            <step type='goal' responsible='politico'>conseguirVotos</step>
            <step type='command' responsible='politico'>}</step>
         </plan>
         <plan trigger='votar' context='true'>
            <step type='action' responsible='eleitor'>vote(Source)</step>
            <step type='action' responsible='eleitor'>leaveRole("eleitor")</step>
         </plan>
         <plan trigger='financiarCampanha' context='true'>
            <step type='action' responsible='lobista'>support(Source)</step>
            <step type='action' responsible='lobista'>leaveRole("lobista")</step>
         </plan>
         <plan trigger='organizarEleicao' context='true'>
            <step type='action' responsible='TSE'>requestUpdateOrganization("updateElection.xml")</
                step>
            <step type='action' responsible='TSE'>.wait(10000)</step>
            <step type='goal' responsible='TSE'>createGroup("resultadoEleitoral")</step>
            <step type='action' responsible='TSE'>.wait(5000)</step>
            <step type='goal' responsible='TSE'>createGroup("processoEleitoral")</step>
         </plan>
      </plans>
   </individualGoals>
   <collectiveGoals>
       <plans>
         <plan trigger='realizarEleicao' context='true'>
            <step type='individualGoal' responsible='politico' numberOfAgentsByRole='2'>ganharEleicao
                </step>
            <step type='collectiveGoal' responsible='resultadoEleitoral'>contabilizarProcesso</step>
         </plan>
      </plans>
   </collectiveGoals>
</functionalDimension>

<dialogicalDimension>
<roleDependence>
   <protocol name='request'>
      <artifact artifactName='FIPARequestArtifact'/>
      <protocolFlow>
         <phase>
            <operation name='request'/>
         </phase>
         <phase>
            <operation name='agree'/>
            <operation name='refuse'/>
         </phase>
         <phase>
            <operation name='failure'/>
            <operation name='inform_done'/>
            <operation name='inform_result'/>
         </phase>
      </protocolFlow>
   </protocol>
</roleDependence>
<groupFormation>
   <protocol name='contractNet'>
      <artifact artifactName='FIPAContractNetArtifact' initialOperation='call_for_proposals' />
      <protocolFlow>
         <phase>
            <operation name='call_for_proposals'/>
         </phase>
         <phase>
            <operation name='propose'/>
            <operation name='refuse'/>
         </phase>
         <phase>
            <operation name='accept_proposal'/>
            <operation name='reject_proposal'/>
         </phase>
         <phase>
            <operation name='failure'/>
            <operation name='inform_done'/>
            <operation name='inform_result'/>
         </phase>
      </protocolFlow>
   </protocol>
</groupFormation>
</dialogicalDimension>

<normativeDimension>
    <createGroup>
      <group groupName='processoEleitoral'>
         <allowedRoles>
            <role>TSE</role>
         </allowedRoles>
      </group>
    </createGroup>
   <artifactsAccess>
      <artifact name='GeneralOrgArtifact' type='organizatonal' restricted='true'>
            <operation op='createGroup' rolesHaveAccess='true'>
               <role>eleitor</role>
               <role>politico</role>
               <role>lobista</role>
               <role>TSE</role>
            </operation>
            <operation op='destroyGroup' rolesHaveAccess='true'>
```

```
            <role>manager</role>
         </operation>
      </artifact>
      <artifact name='GroupsArtifact' type='organizational' restricted='true'>
         <operation op='getListMembersOfPlan' rolesHaveAccess='true'>
            <role>manager</role>
         </operation>
         <operation op='runPlan' rolesHaveAccess='true'>
            <role>manager</role>
         </operation>
         <operation op='runPlanDep' rolesHaveAccess='true'>
            <role>manager</role>
         </operation>
         <operation op='setStep_reponsable' rolesHaveAccess='true'>
            <role>manager</role>
         </operation>
         <operation op='setConversationId' rolesHaveAccess='true'>
            <role>manager</role>
         </operation>
      </artifact>
      <artifact name='NormReputationArtifact' type='organizational' restricted='true'>
         <operation op='increaseNomrRep' rolesHaveAccess='false'/>
         <operation op='decreaseNomrRep' rolesHaveAccess='false'/>
      </artifact>
      <artifact name='NormArtifact' type='organizational' restricted='true'>
         <operation op='createNorma' rolesHaveAccess='false'/>
         <operation op='removeNorm' rolesHaveAccess='false'/>
         <operation op='updateNorm' rolesHaveAccess='false'/>
      </artifact>
      <artifact name='DelegationReputationArtifact' type='organizational' restricted='true'>
         <operation op='increaseDelegationProtocolRep' rolesHaveAccess='false'/>
         <operation op='decreaseDelegationProtocolRep' rolesHaveAccess='false'/>
      </artifact>
      <artifact name='GroupDEPArtifact' type='organizational' restricted='false'/>
      <artifact name='RolesArtifact' type='organizational' restricted='false'/>
      <artifact name='RoleDEPArtifact' type='organizational' restricted='false'/>
      <artifact name='FIPARequestArtifact' type='organizational' restricted='false'/>
      <artifact name='FIPAContractNetArtifact' type='organizational' restricted='false'/>
      <artifact name='electronicBallot' type='application' restricted='true'>
         <operation op='vote' rolesHaveAccess='true'>
            <role>eleitor</role>
         </operation>
      </artifact>
      <artifact name='financialSupport' type='application' restricted='true'>
         <operation op='support' rolesHaveAccess='true'>
            <role>lobista</role>
         </operation>
      </artifact>
   </artifactsAccess>
</normativeDimension>
</organizationalInfrastructure>
```

Código B.2: Especificação organizacional utilizada para o processo de reorganização do estudo de caso *processo político de lobby através de contribuições para campanhas eleitorais*.

```
<?xml version="1.0" encoding="UTF-8"?>
<organizationalInfrastructure>
<structuralDimension>
   <roleDescription>
      <role role='fiscalVotos'>
         <roleCardinality min='1' max='1'/>
         <goals>
            <goal goal='contabilizarVotos' type='achievement' startGoal='false'/>
         </goals>
         <roleCompatibility>
         </roleCompatibility>
      </role>
      <role role='fiscalFinanceiro'>
         <roleCardinality min='1' max='1'/>
         <goals>
            <goal goal='contabilizarFinanciamento' type='achievement' startGoal='false'/>
         </goals>
         <roleCompatibility>
         </roleCompatibility>
      </role>
   </roleDescription>
   <groupDescription>
      <group name='resultadoEleitoral'>
          <goal goal='contabilizarProcesso' type='collective' startGoal='false'/>
         <groupMembers>
            <role role='fiscalVotos' goal='contabilizarVotos'/>
            <role role='fiscalFinanceiro' goal='contabilizarFinanciamento'/>
         </groupMembers>
         </group>
   </groupDescription>
</structuralDimension>

<functionalDimension>
   <individualGoals>
      <plans>
         <plan trigger='contabilizarVotos' context='true'>
            <step type='action' responsible='own'>getResult(Winner)</step>
            <step type='command' responsible='own'>.puts("Politico #{Winner} venceu a eleição!")</
                step>
```

```
         </plan>
         <plan trigger='contabilizarFinanciamento' context='true'>
            <step type='action' responsible='own'>verifyFinancialSupport(2,Candidates)</step>
            <step type='action' responsible='own'>.length(Candidates,MaxCandidates)</step>
            <step type='command' responsible='own'>if(MaxCandidates > 0){</step>
            <step type='command' responsible='own'>for(.range(I,0,MaxCandidates-1)){</step>
            <step type='action' responsible='own'>.nth(I,RolesCandidates,C)</step>
            <step type='action' responsible='own'>deleteCandidate(C)</step>
            <step type='command' responsible='own'>}</step>
            <step type='command' responsible='own'>}</step>
         </plan>
      </plans>
   </individualGoals>
   <collectiveGoals>
       <plans>
         <plan trigger='contabilizarProcesso' context='true'>
            <step type='individualGoal' responsible='fiscalFinanceiro' numberOfAgentsByRole='1'>
                 contabilizarFinanciamento</step>
            <step type='individualGoal' responsible='fiscalVotos' numberOfAgentsByRole='1'>
                 contabilizarVotos</step>
         </plan>
      </plans>
   </collectiveGoals>
</functionalDimension>

<dialogicalDimension>
   <roleDependence/>
   <groupFormation/>
</dialogicalDimension>

<normativeDimension>
    <createGroup>
      <group groupName='resultadoEleitoral'>
         <allowedRoles>
            <role>TSE</role>
         </allowedRoles>
      </group>
    </createGroup>
   <artifactsAccess>
      <artifact name='electronicBallot' type='application' restricted='true'>
            <operation op='getResult' rolesHaveAccess='true'>
               <role>fiscalVotos</role>
            </operation>
            <operation op='deleteCandidate' rolesHaveAccess='true'>
               <role>fiscalFinanceiro</role>
            </operation>
      </artifact>
      <artifact name='financialSupport' type='application' restricted='true'>
            <operation op='verifyFinancialSupport' rolesHaveAccess='true'>
               <role>fiscalFinanceiro</role>
            </operation>
      </artifact>
   </artifactsAccess>
</normativeDimension>
</organizationalInfrastructure>
```

APÊNDICE C ESQUEMA DA ESPECIFICAÇÃO ORGANIZACIONAL

Neste apêndice são mostrados os passos para criação do arquivo xml, que descreve a organização, que é fornecido como entrada para a infraestrutura OrIAs. Para tal é utilizado o estudo de caso *processo político de lobby através de contribuições para campanhas eleitorais* para exemplificar o processo. Para facilitar o entendimento é mostrado como descrever cada uma das dimensões presentes no arquivo separadamente.

No arquivo xml as dimensões da organização são descritas entre as tags `<organizationalInfrastructure>` e `<\organizationalInfrastructure>`.

C.1 Descrição da Dimensão Estrutural

Esta dimensão é responsável por descrever os papéis e grupos da organização e suas respectivas características. Para exemplificar como descrever um papel e um grupo são utilizados o papel político e o grupo processo eleitoral.

No arquivo xml as características da dimensão estrutural são descritas entre as tags `<structuralDimension>` e `<\structuralDimension>`. As características dos papéis da organização são descritos entre as tags `<roleDescription>` e `<\roleDescription>` enquanto que as características dos grupos são descritos entre as tags `<groupDescription>` e `<\groupDescription>`.

Para descrever o papel político deve ser criada uma tag `<role role='politico'>` entre as tags `<roleDescription>` e `<\roleDescription>`. A tag `role` possui como atributo o nome que define o papel, no caso o papel político. Entre as tags `<role role='politico'>` e `<\role>` são descritas as características do respectivo papel.

A tag `<roleCardinality min='1' max='2'/>` define o número máximo (`max`) e mínimo (`min`) de agentes que podem assumir o papel. Entre as tags `<goals>` e `<goals>` são definidos os objetivos do papel. Para cada objetivo deve ser criada uma tag `<goal>` com seus respectivos atributos. Para o objetivo ganhar eleicao `<goal goal='ganharEleicao' type='achievement' startGoal='false'/>` é definido: o nome que caracteriza o objetivo no atributo `goal`, o tipo do objetivo no atributo `type` (o tipo pode ser *achievement* ou *maintenance*) e o se o objetivo é inicial ou não no atributo `startGoal` (*true* ou *false*). Entre as tags `<roleCompatibility>`

e <\roleCompatibility> são definidas as compatibilidades do papel com os demais papéis da organização. Para cada papel compatível de ser criada uma tag <role role='eleitor'/> com o nome que caracteriza o respectivo papel.

Para descrever o grupo processo eleitoral deve ser criada uma tag <group name='processoEleitoral'> entre as tags <groupDescription> e <\groupDescription>. A tag group possui como atributo o nome que define o papel, no caso o grupo processoEleitoral. Entre as tags <group name='processoEleitoral'> e <\group> são descritas as características do respectivo grupo.

A tag <goal goal='realizarEleicao' type='collective' startGoal='true'/> define o objetivo do grupo. O atributo goal define o nome que caracteriza o objetivo do grupo. O atributo type especifica o tipo do objetivo (*individual* ou *collective*) O atributo startGoal indica se esse objetivo é inicial ou não.

Entre as tags <groupMembers> e <\groupMembers> são definidos os membros do grupo. Para cada membro é criada uma tag <role role='politico' goal='ganharEleicao'/> que define o papel (atributo role) responsável por cada subobjetivo (atributo goal) do plano coletivo.

No Código C.1 é mostrada a descrição da dimensão estrutural onde são descritas as características do papel político e do grupo processo eleitoral.

Código C.1: Descrição do papel político e do grupo processo eleitoral na especificação organizacional com suas respectivas características.

```
<structuralDimension>
   <roleDescription>
      <role role='politico'>
         <roleCardinality min='1' max='2'/>
         <goals>
            <goal goal='ganharEleicao' type='achievement' startGoal='false'/>
            <goal goal='conseguirVotos' type='achievement' startGoal='false'/>
            <goal goal='manterApoioFinaceiro' type='maintenance' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
      <role role='eleitor'>
         <roleCardinality min='1' max='5'/>
         <goals>
            <goal goal='votar' type='achievement' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='politico'/>
            <role role='lobista'/>
         </roleCompatibility>
      </role>
      <role role='lobista'>
         <roleCardinality min='2' max='4'/>
         <goals>
            <goal goal='financiarCampanha' type='achievement' startGoal='false'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
      <role role='TSE'>
         <roleCardinality min='1' max='1'/>
         <goals>
            <goal goal='organizarEleicao' type='achievement' startGoal='true'/>
         </goals>
         <roleCompatibility>
            <role role='eleitor'/>
         </roleCompatibility>
      </role>
   </roleDescription>
   <groupDescription>
      <group name='processoEleitoral'>
          <goal goal='realizarEleicao' type='collective' startGoal='true'/>
         <groupMembers>
            <role role='politico' goal='ganharEleicao'/>
```

```
        </groupMembers>
        </group>
   </groupDescription>
</structuralDimension>
```

C.2 Descrição da Dimensão Funcional

Na dimensão funcional é definida toda a dinâmica sobre a dimensão estrutural. Nessa dimensão são definidos os planos para os objetivos dos papéis e grupos definidos na dimensão estrutural. No arquivo xml as características da dimensão funcional são descritas entre as tags `<functionalDimension >` e tags `<\functionalDimension>`.

Para exemplificar como é descrita essa dimensão são especificados os planos para os objetivos do papel político e o plano do grupo processo eleitoral.

Os planos relativos aos objetivos dos papéis são descritos entre as tags `<plans>` e `<\plans>` que estão entre as tags `<individualGoals>` e `<\individualGoals>`. Para cada plano deve ser criada um tag `<plan trigger='ganharEleicao' context='true'>` cujo atributo `trigger` especifica o *trigger* do plano e o atributo `context` define o contexto que o plano é aplicado. Entre as tags `<plan trigger='ganharEleicao' context='true'>` e `<\plan>` são definidos os passos do plano. Para cada passo do plano é criada uma tag `<step type='command' responsible='politico'>`. O atributo `type` representa o tipo do passo do plano (`command` (correspondente a algum comando da linguagem de agentes utilizadas), `action` (correspondente a alguma operação executada sobre um artefato) e `goal` (correspondente a algum objetivo)). No atributo `responsible` é definido o papel responsável pela etapa do plano. Entre as tags `<step type='command' responsible='politico'>` e `<\step>` é especificado o passo do plano.

Os plano relativo ao objetivo de um grupo é descrito entre as tags `<plans>` e `<\plans>` que estão entre as tags `<collectiveGoals>` e `<\collectiveGoals >` . Para cada plano deve ser criada um tag `<plan trigger='realizarEleicao' context='true'>` cujo atributo `trigger` especifica o *trigger* do plano e o atributo `context` define o contexto que o plano é aplicado. Entre as tags `<plan trigger='realizarEleicao' context='true'>` e `<\plan>` são definidos os passos do plano. Para cada passo do plano é criada uma tag `<step type='individualGoal' responsible= 'politico' numberOfAgentsByRole='2'>ganharEleicao</step>`. O atributo `type` representa o tipo do passo do plano (`individualGoal` ou `collectiveGoal`). O atributo `responsible` permite definir o responsável pela etapa do plano (agente ou grupo). O atributo `numberOfAgentsByRole` define o número de agentes responsável pela execução da etapa do plano coletivo. Esse atributo é definido apenas quando os responsáveis pelos passos do plano são agentes. Entre as tags `<step type='individualGoal' responsible='politico' numberOfAgentsByRole='2'>` e `<\step>` é especificado o passo do plano.

No Código C.2 é mostrada a descrição dos planos para os objetivos do papel político e do plano para o objetivo do grupo processo eleitoral na especificação organizacional.

Código C.2: Descrição dos planos para os objetivos do papel político e do plano para o objetivo do grupo processo eleitoral na especificação organizacional.

```
<functionalDimension>
   <individualGoals>
      <plans>
         <plan trigger='ganharEleicao' context='true'>
            <step type='goal' responsible='lobista'>financiarCampanha</step>
            <step type='goal' responsible='politico'>conseguirVotos</step>
         </plan>
         <plan trigger='manterApoioFinaceiro' context='apoioFinanceiro(X) &amp; X &lt; 2'>
            <step type='action' responsible='politico'>getAgentsByRole("lobista",Agents)</step>
            <step type='action' responsible='politico'>.length(Agents,MaxAgents)</step>
            <step type='command' responsible='politico'>if(MaxAgents > 0){</step>
            <step type='goal' responsible='lobista'>financiarCampanha</step>
            <step type='command' responsible='politico'>}</step>
         </plan>
         <plan trigger='conseguirVotos' context='true'>
            <step type='action' responsible='politico'>getAgentsByRole("eleitor",Agents)</step>
            <step type='action' responsible='politico'>.length(Agents,MaxAgents)</step>
            <step type='command' responsible='politico'>if(MaxAgents > 0){</step>
            <step type='goal' responsible='eleitor'>votar</step>
            <step type='goal' responsible='politico'>conseguirVotos</step>
            <step type='command' responsible='politico'>}</step>
         </plan>
   </individualGoals>
   <collectiveGoals>
       <plans>
         <plan trigger='realizarEleicao' context='true'>
            <step type='individualGoal' responsible='politico' numberOfAgentsByRole='2'>ganharEleicao
                </step>
            <step type='collectiveGoal' responsible='resultadoEleitoral'>contabilizarProcesso</step>
         </plan>
      </plans>
   </collectiveGoals>
</functionalDimension>
```

C.3 Descrição da Dimensão Dialógica

Na dimensão dialógica são definidos os protocolos utilizados para interação dos agentes para delegação de tarefas, quando os mesmos são dependentes, e para formação dos grupos. No arquivo xml as características da dimensão dialógica são descritas entre as tags `<dialogicalDimension>` e tags `<\dialogicalDimension>`.

O protocolo utilizado para delegação de tarefas deve ser especificado entre as tags `<roleDependence>` e `<\roleDependence>`. Já o protocolo utilizada para formação dos grupos deve ser especificada entre as tags `<groupFormation>` e `<\groupFormation>`.

Para exemplificar como é descrita essa dimensão é especificado o fluxo do protocolo Request utilizado para interação dos agentes quando os mesmos são dependentes para realização de algum objetivo.

O protocolo é especificado entre as tags `<protocol name='request'>` e `<\protocol>`. O atributo `name` define o protocolo que está sendo descrito. Entre essas tags é definida a tag `<artifact artifactName='FIPARequestArtifact' \>` que permite definir o artefato criado para o protocolo. Entre as tags `<protocolFlow>` e `<\protocolFlow>` são definidas as fases do protocolo entre as tags `<phase>` e `<\phase>` . Para cada fase são definidas as operações do protocolo (correspondente ao nome das operações disponibilizadas no artefato do protocolo) que podem ser executadas pela tag `<operation name='call-for-proposals'/>`.

No Código C.3 é mostrada a descrição dos protocolos utilizados para delegar as tarefas quando os agentes são dependentes e para formação de grupos.

Código C.3: Descrição dos protocolos utilizados para delegar as tarefas quando os agentes são dependentes e para formação de grupos.

```
<dialogicalDimension>
<roleDependence>
   <protocol name='request'>
      <artifact artifactName='FIPARequestArtifact'/>
      <protocolFlow>
         <phase>
            <operation name='request'/>
         </phase>
         <phase>
            <operation name='agree'/>
            <operation name='refuse'/>
         </phase>
         <phase>
            <operation name='failure'/>
            <operation name='inform_done'/>
            <operation name='inform_result'/>
         </phase>
      </protocolFlow>
   </protocol>
</roleDependence>
<groupFormation>
   <protocol name='contractNet'>
      <artifact artifactName='FIPAContractNetArtifact' initialOperation='call_for_proposals' />
      <protocolFlow>
         <phase>
            <operation name='call_for_proposals'/>
         </phase>
         <phase>
            <operation name='propose'/>
            <operation name='refuse'/>
         </phase>
         <phase>
            <operation name='accept_proposal'/>
            <operation name='reject_proposal'/>
         </phase>
         <phase>
            <operation name='failure'/>
            <operation name='inform_done'/>
            <operation name='inform_result'/>
         </phase>
      </protocolFlow>
   </protocol>
</groupFormation>
</dialogicalDimension>
```

C.4 Descrição da Dimensão Normativa

Na dimensão normativa são definidas as normas para criação de grupos e para acesso aos recursos (artefatos) controlados por normas. No arquivo xml as características da dimensão normativa são descritas entre as tags `<normativeDimension>` e tags `<\normativeDimension>`.

Para exemplificar como é descrita essa dimensão são especificadas as normas para criação do grupo processo eleitoral e para o acesso ao artefato GeneralOrg.

As normas para criação dos grupos são definidas entre as tags `<createGroup>` e tags `<\createGroup>`. Entre essas tags é definida a tag `<group groupName='processoEleitoral'>` o grupo que tem sua criação restringida pelo atributo `groupName`. Entre essas tags estão as tags `<allowedRoles>` e `<\allowedRoles>` são definidos os papéis que podem criar o respectivo grupo entre as tags `<role>` e `</role>`.

As normas para o acesso aos artefatos são definidas entre as tags `<artifactsAccess>` e `<\artifactsAccess>`. Para cada artefato é criada uma tag `<artifact name='GeneralOrgArtifact' type='organizatonal' restricted='true'>`. O atributo `name` corresponde o nome do artefato. O atributo `type` define o tipo do artefato (*organizatonal* ou *application*). O atributo `restricted` define se o artefato tem suas operações restritas.

Caso o artefato possua operações que sejam restritas deve ser descrita entre as tags `<operation op='getResult' rolesHaveAccess='true'>` e `<\operation>` as características dessas operações, onde o atributo `op` define o nome da operação o atributo `rolesHaveAccess` indica se algum papel tem acesso a essa operação, caso o valor desse atributo seja *true* deve ser definido entre as tags `<role>` e `<\role>` papel que pode acessar essa operação.

No Código C.4 é mostrada a descrição das normas para criação do grupo e para utilização do artefato GeneralOrg.

Código C.4: Descrição das normas para criação do grupo processo eleitoral e para utilização do artefato GeneralOrg.

```
<normativeDimension>
    <createGroup>
      <group groupName='processoEleitoral'>
         <allowedRoles>
            <role>TSE</role>
         </allowedRoles>
      </group>
    </createGroup>
   <artifactsAccess>
      <artifact name='GeneralOrgArtifact' type='organizatonal' restricted='true'>
            <operation op='createGroup' rolesHaveAccess='true'>
               <role>eleitor</role>
               <role>politico</role>
               <role>lobista</role>
               <role>TSE</role>
            </operation>
            <operation op='destroyGroup' rolesHaveAccess='true'>
               <role>manager</role>
            </operation>
      </artifact>
   </artifactsAccess>
</normativeDimension>
```