

Luis Gustavo Nardin

**Uma Arquitetura de Apoio à
Interoperabilidade de Modelos de
Reputação de Agentes**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Luis Gustavo Nardin

Uma Arquitetura de Apoio à Interoperabilidade de Modelos de Reputação de Agentes

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Jaime Simão Sichman

Este exemplar foi revisado e alterado em relação a versão original, sob responsabilidade única do autor com a anuência de seu orientador.

São Paulo, 19 de Abril de 2009.

Assinatura do autor

Assinatura do orientador

Ficha Catalográfica

Nardin, Luis Gustavo

Uma Arquitetura de Apoio à Interoperabilidade de Modelos de Reputação de Agentes. São Paulo, 2009. 146 p. DVD-ROM em Anexo.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Inteligência artificial. 2. Sistemas multiagentes. 3. Reputação. 4. Arquitetura baseada em Serviços. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. Título.

Agradecimentos

Agradeço primeiramente ao meu orientador, Prof. Dr. Jaime Simão Sichman, por suas diretrizes sempre claras e criteriosas que possibilitaram a realização deste trabalho.

Aos meus colegas do Laboratório de Técnicas Inteligentes (LTI), em especial à: Dra. Anarosa Brandão, por suas sugestões e apoio no direcionamento deste trabalho, bem como na sua disponibilidade na revisão técnica de diversos artigos; e Priscilla Avegliano, pela sua contribuição na disponibilização de material de apoio a este trabalho.

Agradeço ao Prof. Dr. Laurent Vercouter por sua disponibilidade de tempo para responder minhas inúmeras questões a respeito do funcionamento das plataformas de simulação.

Gostaria de agradecer ao Guillaume Müller e ao Jomi Hübner pelos esclarecimentos quanto ao funcionamento da plataforma de simulação, bem como pelo apoio no desenvolvimento dos experimentos.

Agradecimentos especiais aos meus pais e irmãs, por todo seu apoio despendido durante o período de elaboração deste trabalho. Agradeço ainda uma pessoa especial que foi essencial para que atingisse mais este objetivo.

Gostaria ainda de agradecer a todos meus amigos, que próximos ou à distância, foram de extrema importância para que eu atingisse mais este objetivo.

Resumo

Em sistemas multiagentes (SMA), a impossibilidade de controle do comportamento interno dos agentes propicia a emergência de dilemas sociais. Uma das possíveis abordagens para resolver tais dilemas é a abordagem social, que tem no conceito de reputação seu principal mecanismo de controle. Para agilizar a obtenção da reputação, torna-se essencial que os agentes cooperem, trocando informações a respeito de reputação. Assim, a partir de diversas observações, avaliações e troca de informações sobre reputação pode-se observar o surgimento da ordem social no sistema global. Entretanto, a heterogeneidade semântica entre os diferentes modelos de reputação torna a interação de agentes um problema, principalmente em SMA abertos. Este trabalho propõe uma arquitetura baseada em serviços, denominada SOARI, para o tratamento deste problema. Esta arquitetura apóia o mapeamento e tradução de conceitos entre ontologias de modelos de reputação distintos através do uso de uma ontologia comum no domínio da reputação. Para ilustrar o uso da arquitetura, foram realizados diversos experimentos nos domínios de comércio eletrônico e de avaliação de obras de arte. Neste último caso, analisou-se os impactos causados nos agentes como consequência da habilitação de uma maior expressividade e heterogeneidade na comunicação sobre reputação.

Abstract

In Multiagent Systems (MAS), controlling the internal agent's behavior is impossible and therefore social dilemma may emerge. One possible approach to solve the problem is the social approach, which considers reputation as its main control mechanism. In order to accelerate the reputation evaluation, it is essential that the cooperating agents could exchange information about reputation. Thus, by using observations, evaluations and information exchange about reputation the social order in the global system may occur. However, the semantic heterogeneity of different reputation models makes agent interaction a problem, especially in open MAS. This work proposes a service oriented architecture, named SOARI, to deal with this problem. This architecture supports concept mapping and translation among different reputation model ontologies by using a common ontology. In order to illustrate its deployment, several experiments in the electronic commerce and art appraisal domains were performed. In the latter it was analyzed the effects caused on the agents as a consequence of a more expressive and heterogeneous communication about reputation.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

1	Introdução	1
1.1	Motivação	3
1.2	Objetivos	5
1.3	Metodologia	6
1.4	Organização do Texto	7
2	Modelos de Reputação em SMA	8
2.1	Introdução	8
2.2	Aspectos da Reputação	9
2.3	Principais Modelos de Reputação	10
2.4	Repage	12
2.4.1	Arquitetura e Funcionamento Interno	12
2.4.2	Avaliações Sociais	14
2.4.3	Representação da Avaliação	15
2.5	L.I.A.R.	16
2.5.1	Arquitetura e Funcionamento Interno	16
2.5.2	Avaliações Sociais	19
2.5.3	Representação da Avaliação	20
2.6	Considerações Finais	20

3	Interoperabilidade Semântica Baseada em Ontologias	22
3.1	Introdução	22
3.2	Ontologias	23
3.2.1	Definição	23
3.2.2	Linguagens de Representação de Ontologias	24
3.2.3	Ambientes para Raciocínio Automático com Ontologias	25
3.2.4	Operações com Ontologias	27
3.3	Ontologias de Domínio da Reputação	28
3.3.1	Ontologia de Chang, Hussain e Dillon (2006)	28
3.3.2	FORe	29
3.4	Abordagens para Interoperabilidade Semântica	31
3.5	Considerações Finais	32
4	Arquiteturas Orientadas a Serviços	33
4.1	Introdução	33
4.2	Características Principais	35
4.3	Web Services	35
4.3.1	XML	36
4.3.2	SOAP	36
4.3.3	WSDL	37
4.3.4	UDDI	39
4.4	Considerações Finais	40
5	SOARI	41
5.1	Introdução	41
5.2	Arquitetura Genérica de Agente para Interação sobre Reputação	42
5.3	Arquitetura Orientada a Serviços para Interação sobre Reputação	44
5.3.1	Serviço de Mapeamento de Ontologias	46
5.3.2	Tradutor	49

5.4	Fases de Funcionamento	54
5.4.1	Inicialização	54
5.4.2	Operação	58
5.5	Integração a Sistemas Legados	60
5.6	Considerações Finais	60
6	Implementação e Funcionamento da Arquitetura	62
6.1	Introdução	62
6.2	Implementação	63
6.2.1	Serviço de Mapeamento de Ontologias	63
6.2.2	Tradutor	63
6.3	Funcionamento	70
6.3.1	Inicialização	71
6.3.2	Operação	75
6.4	Considerações Finais	79
7	Exemplo de Aplicação da Arquitetura	80
7.1	Introdução	80
7.2	Ambientes de Simulação	81
7.2.1	Plataforma de Simulação ART	81
7.2.2	Plataforma de Simulação FOrEART	83
7.3	Modelo de Agente	84
7.4	Experimentos Realizados	86
7.4.1	Experimento 1: ART/L.I.A.R	89
7.4.2	Experimento 2: ART/Repage	90
7.4.3	Experimento 3: ART/Misto	91
7.4.4	Experimento 4: FOrEART/L.I.A.R.	95
7.4.5	Experimento 5: FOrEART/Repage	96
7.4.6	Experimento 6: FOrEART/Misto	98

7.5	Análise dos Resultados	102
7.5.1	Metodologia de Análise	102
7.5.2	Efeito da expressividade na comunicação	103
7.5.3	Efeito da heterogeneidade de modelos de reputação	106
7.6	Considerações Finais	111
8	Conclusões	112
	Referências	115
	Apêndice A – Resultados dos Experimentos	120
	Apêndice B – Testes Estatísticos	144

Lista de Figuras

1.1	Abordagem híbrida para interoperabilidade de modelos de reputação	5
2.1	Arquitetura do modelo Repage (SABATER-MIR; PAOLUCCI; CONTE, 2006)	12
2.2	Exemplo de Avaliação do modelo Repage (SABATER-MIR; PAOLUCCI, 2007)	15
2.3	Componentes do modelo L.I.A.R. (MULLER; VERCOUTER, 2008) .	16
2.4	Processo de Raciocínio do modelo L.I.A.R. (MULLER; VERCOUTER, 2008)	18
4.1	Arquitetura Orientada a Serviços (MCGOVERN et al., 2003)	34
4.2	Exemplo de arquivo em formato XML	36
4.3	Exemplo de mensagem SOAP	37
4.4	Exemplo de definição do elemento <i>types</i> do WSDL	38
4.5	Exemplo de definição do elemento <i>message</i> do WSDL	38
4.6	Exemplo de definição do elemento <i>port type</i> do WSDL	39
4.7	Exemplo de definição do elemento <i>binding</i> do WSDL	39
4.8	Exemplo de definição do elemento <i>service</i> do WSDL	39
5.1	Arquitetura genérica de agentes com modelo de reputação	41
5.2	Arquitetura genérica de agentes para a interação de reputação (VERCOUTER et al., 2007)	42
5.3	Arquitetura Orientada a Serviços para Interação de Reputação . .	45
5.4	Componentes do Serviço de Mapeamento de Ontologias	46
5.5	Componentes do TRADUTOR	50
5.6	Diagrama de seqüência da fase de Inicialização do SMO	57
5.7	Diagrama de seqüência da fase de Operação do SMO	59

6.1	Consulta em SPARQL utilizando os conceitos da ontologia do modelo de reputação Repage	75
6.2	Mensagem SOAP para tradução do conceito <i>Reputation</i>	76
6.3	Consulta em SPARQL utilizando os conceitos da <i>Ontologia Comum no Domínio da Reputação</i> FORe	77
6.4	Mensagem FIPA-ACL	77
6.5	Consulta em SPARQL utilizando os conceitos da ontologia do modelo de reputação L.I.A.R.	78
6.6	Resultado da consulta SPARQL no modelo de reputação L.I.A.R.	78
6.7	Resultado da consulta em formato XML	79
7.1	Experimento 1: ART/L.I.A.R. - Valor médio do <i>DIbRp</i> , <i>IbRp</i> e <i>RpRcbRp</i> para o agente <i>Desonesto</i>	90
7.2	Experimento 2: ART/Repage - Valor médio da Imagem e Reputação para o agente <i>Desonesto</i>	91
7.3	Experimento 3.1: ART/Misto-D/L.I.A.R. - Valor médio da Imagem e Reputação para o agente <i>Desonesto</i>	92
7.4	Experimento 3.1: ART/Misto-D/L.I.A.R. - Valor médio do <i>DIbRp</i> , <i>IbRp</i> e <i>RpRcbRp</i> para o agente <i>Desonesto</i>	93
7.5	Experimento 3.2: ART/Misto-D/Repage - Valor médio da Imagem e Reputação para o agente <i>Desonesto</i>	94
7.6	Experimento 3.2: ART/Misto-D/Repage - Valor médio do <i>DIbRp</i> , <i>IbRp</i> e <i>RpRcbRp</i> para o agente <i>Desonesto</i>	95
7.7	Experimento 4: FOReART/L.I.A.R. - Valor médio do <i>DIbRp</i> , <i>IbRp</i> e <i>RpRcbRp</i> para o agente <i>Desonesto</i>	96
7.8	Experimento 5: FOReART/Repage - Valor médio da Imagem e Reputação para o agente <i>Desonesto</i>	97
7.9	Experimento 6.1: FOReART/Misto-D/L.I.A.R. - Valor médio da Imagem e Reputação para o agente <i>Desonesto</i>	99
7.10	Experimento 6.1: FOReART/Misto-D/L.I.A.R - Valor médio do <i>DIbRp</i> , <i>IbRp</i> e <i>RpRcbRp</i> para o agente <i>Desonesto</i>	99
7.11	Experimento 6.2: FOReART/Misto-D/Repage - Valor médio da Imagem e Reputação para o agente <i>Desonesto</i>	101

7.12 Experimento 6.2: FOrEART/Misto-D/Repage - Valor médio do
DlRp, *IbRp* e *RpRcbRp* para o agente *Desonesto* 101

Lista de Tabelas

2.1	Resumo da análise dos modelos de reputação Repage e L.I.A.R.	21
5.1	Estrutura de dados da tabela ONTOLOGY	47
5.2	Estrutura de dados da tabela ONTRELATION	48
5.3	Estrutura de dados da tabela ONTTRANSLATION	48
6.1	Campos dos arquivos de configuração do CT	64
6.2	Tipos de Mensagem SOARI x Performativa FIPA-ACL	65
6.3	Conceitos da ontologia do modelo de reputação Repage	72
6.4	Conceitos da ontologia do modelo de reputação L.I.A.R.	73
6.5	Alinhamento do modelo de reputação Repage com a FORe	73
6.6	Alinhamento do modelo de reputação L.I.A.R. com a FORe	74
6.7	Mapeamento do modelo de reputação Repage para FORe	74
6.8	Mapeamento do modelo de reputação L.I.A.R. para FORe	75
6.9	Mapeamento parcial dos conceitos do Repage para FORe	76
6.10	Mapeamento parcial dos conceitos da FORe para L.I.A.R.	78
7.1	Tabela de experimentos realizados	87
7.2	Tempo médio de execução das simulações	88
7.3	Experimento 1: ART/L.I.A.R. - Valor dos atributos do agente <i>Desonesto</i>	89
7.4	Experimento 2: ART/Repage - Valor dos atributos do agente <i>De-</i> <i>sonesto</i>	90
7.5	Experimento 3.1: ART/Misto-D/L.I.A.R. - Valor dos atributos do agente <i>Desonesto</i>	92
7.6	Experimento 3.2: ART/Misto-D/Repage - Valor dos atributos do agente <i>Desonesto</i>	93

7.7	Experimento 4: FOReART/L.I.A.R. - Valor dos atributos do agente <i>Desonesto</i>	95
7.8	Experimento 5: FOReART/Repage - Valor dos atributos do agente <i>Desonesto</i>	97
7.9	Experimento 6.1: FOReART/Misto-D/L.I.A.R. - Valor dos atributos do agente <i>Desonesto</i>	98
7.10	Experimento 6.2: FOReART/Misto-D/Repage - Valor dos atributos do agente <i>Desonesto</i>	100
7.11	Hipóteses utilizadas para verificar o efeito da expressividade de comunicação	105
7.12	Valores de p para verificar o efeito da expressividade de comunicação	105
7.13	Hipóteses para verificar o efeito da heterogeneidade na plataforma ART	108
7.14	Valores de p para verificar o efeito da heterogeneidade na plataforma ART	108
7.15	Hipóteses para verificar o efeito da heterogeneidade na plataforma FOReART	110
7.16	Valores de p para verificar o efeito da heterogeneidade na plataforma FOReART	111
A.1	Resultados do Experimento 1: ART/L.I.A.R.	120
A.2	Resultados do Experimento 2: ART/Repage	123
A.3	Resultados do Experimento 3.1: ART/Misto-D/L.I.A.R.	126
A.4	Resultados do Experimento 3.2: ART/Misto-D/Repage	129
A.5	Resultados do Experimento 4: FOReART/L.I.A.R.	132
A.6	Resultados do Experimento 5: FOReART/Repage	135
A.7	Resultados do Experimento 6.1: FOReART/Misto-D/L.I.A.R.	137
A.8	Resultados do Experimento 6.2: FOReART/Misto-D/Repage	140
B.1	Valores de p para verificar o efeito da expressividade de comunicação	144
B.2	Valores de p para verificar o efeito da heterogeneidade	145
B.3	Valores de p para Teste Shapiro-Wilk	145

B.4	Valores de p para Teste F	146
-----	---------------------------------------	-----

Lista de Abreviaturas e Siglas

ACL *Agent Communication Language*

AFRAS *A Fuzzy Reputation Agent System*

API *Application Programming Interface*

ART *Agent Reputation and Trust Testbed*

CT *Controlador do Tradutor*

DAML *DARPA Agent Markup Language*

DL *Description Logic*

DTD *Document Type Definition*

FaCT *Fast Classification of Terminologies*

FIPA *Foundation for Intelligent Physical Agents*

FORe *Functional Ontology of Reputation*

FOReART *Functional Ontology of Reputation in ART Testbed*

HSQldb *Hyperthreaded Structured Query Language Database*

HTML *HyperText Markup Language*

HTTP *Hypertext Transfer Protocol*

ICSMO *Interface Cliente do Serviço de Mapeamento de Ontologias*

IET *Interface de Estratégias de Tradução*

IMI *Interface do Módulo de Interação*

IML *Interface do Módulo de Linguagem*

IMTR *Interface do Módulo de Tratamento de Reputação*

ITV *Interface de Transformação de Valores*

JADE *Java Agent DEvelopment Framework*

KIF *Knowledge Interchange Format*

L.I.A.R. *Liar Identification for Agent Reputation*

MI *Módulo de Interação*

MMR *Módulo de Mapeamento de Reputação*

MTR *Módulo de Tratamento de Reputação*

OIL *Ontology Inference Layer*

OMS *Ontology Mapping Service*

OWL *Web Ontology Language*

RDF *Resource Description Framework*

Repage *REputation and imAGE*

RACER *Renamed ABox and Concept Expression Reasoner*

SGML *Standard Generalized Markup Language*

SMA *Sistemas Multiagentes*

SMO *Serviço de Mapeamento de Ontologias*

SOA *Service Oriented Architecture*

SOAP *Simple Object Access Protocol*

SOARI *Service Oriented Architecture for Reputation Interaction*

SPARQL *Simple Protocol and RDF Query Language*

SQL *Structure Query Language*

W3C *World Wide Web Consortium*

UDDI *Universal Description, Discovery, and Integration*

UML *Unified Modelling Language*

UML-OCL *Unified Modelling Language-Object Constraint Language*

URL *Uniform Resource Locator*

WSDL *Web Services Description Language*

XML *eXtensible Markup Language*

1 Introdução

Seres humanos normalmente agem em interesse próprio. Quando em ambientes compartilhados, realizam ações coletivas para satisfazerem seus objetivos e desejos, através de diferentes tipos de interações, como cooperação, competição e negociação. Tais interações criam certas relações de dependência, que fazem emergir situações denominadas de dilemas sociais.

Os dilemas sociais são caracterizados por serem situações em que um agente, ao tomar uma decisão para maximizar o seu próprio ganho de curto prazo, produz resultados desfavoráveis para os outros agentes da sociedade (OSTROM, 1998). O exemplo mais típico e conhecido de dilema social é representado pelo chamado Dilema do Prisioneiro (AXELROD, 1984), que consiste em um problema clássico da Teoria dos Jogos.

Os chamados Sistemas Multiagentes (SMA)¹ são compostos por um conjunto de agentes autônomos, situados em um ambiente compartilhado, que interagem uns com os outros com o intuito de atingirem seus objetivos (WOOLDRIDGE, 2002). Em SMA, a emergência de dilemas sociais advém da impossibilidade de se controlar o comportamento interno de cada um dos agentes.

Para ilustrar a emergência de dilemas sociais em sociedades de agentes, considera-se um cenário adaptado do *Agent Reputation and Trust Testbed* (ART) proposto em (FULLAM et al., 2005b). O SMA é composto por dois tipos de agentes: (i) agentes avaliadores, que avaliam pinturas de diferentes épocas, e (ii) agentes clientes, que contratam os avaliadores para avaliarem suas pinturas. Cada agente avaliador possui conhecimento variado sobre pinturas de diferentes épocas, e para tanto, cobra um valor fixo (Va) para realizar a avaliação. Os agentes clientes solicitam aos agentes avaliadores a avaliação de pinturas, sendo a escolha do agente avaliador definida pelo histórico de precisão de suas avaliações anteriores.

Quando um agente cliente solicita a avaliação de uma pintura de uma determinada época a um agente avaliador, este último analisa o pedido e determina

¹Do inglês *Multiagent Systems*.

se possui conhecimento suficiente para avaliá-la. Caso decida que possui o conhecimento necessário, realiza a avaliação e responde ao agente cliente. Em caso contrário, solicita a opinião de outros agentes avaliadores da sociedade. Estas opiniões, requerem o pagamento de um valor fixo (Vo) aos agentes provedores da opinião.

O objetivo de cada agente avaliador é acumular o maior valor possível no decorrer de sua vida. Desta forma, quando solicitado por um agente cliente para avaliar uma pintura, ele tenta inicialmente avaliá-la sem requisitar a opinião de outros agentes, o que lhe proporciona o maior ganho possível (Va pago pelo agente cliente). No caso de solicitar a opinião de outro agente avaliador, seu ganho é reduzido pelo valor pago pela opinião (Vo) ao outro agente avaliador ($Va - Vo$). Porém, como o agente pode requisitar a opinião de diversos agentes (n), seu ganho poderá ser ainda menor ($Va - (n * Vo)$).

Apesar de um agente requisitar a opinião de outros agentes, nada garante ou obriga que estes últimos respondam à requisição, e se respondê-la, que a opinião seja a correta. Racionalmente, se um agente prover uma opinião correta a outro, ele estará auxiliando o primeiro a obter um maior ganho. Conseqüentemente, tal comportamento atrairá um maior número de clientes ao agente solicitante, que por sua vez possibilitará um maior ganho e assim sucessivamente. Analisando deste ponto de vista, a tendência racional de cada agente avaliador é não responder às requisições de opinião de outros agentes avaliadores, ou respondê-las incorretamente. Este não estímulo à cooperação entre os agentes da sociedade caracteriza um dilema social.

A possibilidade de existência de dilemas sociais em SMA aumenta o risco deste não permanecer em um estado estável. Para Castelfranchi (2000), este estado estável é identificado como ordem social, que estabelece uma relação de equilíbrio entre a autonomia dos agentes e o controle do sistema.

Uma das possíveis abordagens para solucionar o problema da obtenção da ordem social é a criação de instituições que realizem a supervisão do comportamento dos agentes na sociedade. Além da supervisão, as instituições são responsáveis pela aplicação de sanções aos agentes que não cumprirem as normas estabelecidas. Tais instituições são entidades normalmente centralizadas e requerem uma visão completa do sistema para que possam desempenhar suas funções.

Outra possível abordagem é a abordagem social, na qual cada um dos agentes possui internamente mecanismos sociais e distribuídos que representam outros agentes e lhe permite o controle adaptativo e auto-organizado da sociedade. Para

Castelfranchi (2000), tais mecanismos sociais são denominados controle social.

Ostrom (1998) identificou que a reciprocidade é a principal norma utilizada por muitos indivíduos em situações que envolva dilema social, e propôs que o principal mecanismo a ser implantado nos agentes para a obtenção da ordem social é o conceito de reputação.

Segundo esta proposta, cada agente observa e avalia uma parte do sistema e identifica os agentes que se comportam de acordo com as normas e os que as violam, sendo atribuído aos agentes observados um valor de reputação. Baseados neste valor, que representa a noção de reputação do agente, os agentes têm como decidir como interagirão com outros agentes.

Com o intuito de agilizar a obtenção da reputação, os agentes cooperam, trocando informações a respeito da reputação dos outros. Assim, a partir de diversas observações, avaliações e troca de informações sobre reputação pode-se observar o surgimento da ordem social no sistema global.

1.1 Motivação

Os SMA são considerados cada vez mais uma opção tecnológica viável ao desenvolvimento de sistemas complexos e abertos. Com uma maior utilização desta tecnologia, aumenta-se a possibilidade do aparecimento de problemas relacionados à obtenção da ordem social em sociedades de agentes.

Aparentemente, dentre as abordagens citadas, a abordagem social é a mais adequada para auxiliar na solução do problema da obtenção da ordem social em SMA. Segundo Ostrom (1998), isto ocorre pois ela se adapta melhor às características destes sistemas e reflete melhor a realidade das sociedades humanas na resolução deste tipo de problema.

No domínio de SMA, o controle social pode ser efetivado através da utilização de sistemas e modelos de reputação. Entretanto, como ainda não existe um consenso na definição do conceito de reputação, os diversos sistemas e modelos de reputação já propostos representam aspectos distintos da noção de reputação. Exemplos de sistemas e modelos de reputação propostos são Histos e Sporas (ZACHARIA; MAES, 2000), Regret (SABATER-MIR; SIERRA, 2002), Yu-Singh (YU; SINGH, 2002), AFRAS (*A Fuzzy Reputation Agent System*) (CARTER; MOLINA; MURO, 2003), L.I.A.R. (*Liar Identification for Agent Reputation*) (MULLER; VERCOUTER, 2008) e Repage (*REputation and imAGE*) (SABATER-MIR; PAOLUCCI; CONTE, 2006).

Devido às características de SMA abertos, não há como determinar, de maneira não intrusiva, o modelo de reputação utilizado pelos agentes do sistema. Conseqüentemente, existe uma grande possibilidade de que os agentes utilizem modelos de reputação distintos. Porém, para que haja um controle efetivo da sociedade de agentes através da utilização da abordagem social, conforme sugere Ostrom (1998), todos os agentes têm que ser capazes de interagir uns com os outros sobre reputação.

Conseqüentemente, a co-existência desta diversidade de modelos de reputação em um mesmo sistema dificulta a interoperabilidade sobre reputação entre agentes, já que eles se baseiam em semânticas distintas para representar os conceitos de reputação (BRANDÃO et al., 2007).

Desta forma, suponha a instanciação do cenário de avaliação de pinturas com três agentes avaliadores, sendo eles Ana, Beto e Clara. Suponha ainda que a agente Ana receba uma solicitação de um agente cliente para avaliar uma pintura da época impressionista. Como ela identifica que seu conhecimento não é suficiente para avaliar uma pintura desta época, ela decide solicitar a opinião de outro agente avaliador. Devido a interações passadas, Ana sabe que Clara possuía conhecimento para avaliar pinturas da época impressionista, mas como já se passou muito tempo desde a última interação entre elas, Ana não está certa se Clara ainda tem boa reputação em relação à avaliação de pinturas desta época. Contudo, Ana conhece o agente Beto, e Ana sabe que tal agente teve interações recentes com Clara. Assim, Ana resolve consultar Beto sobre a reputação de Clara como avaliadora de pinturas desta época. Ana consulta Beto, que responde informando que Clara possui uma boa reputação como avaliadora de pinturas da época impressionista. Desta forma, Ana atualiza sua reputação sobre Clara como avaliadora de pinturas desta época e resolve solicitar sua opinião. Clara responde à solicitação e recebe em troca o pagamento por sua opinião a Ana, que por sua vez responde ao agente cliente e recebe o pagamento pelo serviço prestado.

Para que este cenário possa ocorrer, todos os agentes têm que conhecer a semântica dos conceitos utilizados para a troca de informação, dentre os quais encontram-se os conceitos relacionados à noção de reputação. Considerando um ambiente dinâmico e com heterogeneidade de agentes e modelos de reputação, como garantir que estes agentes se comuniquem sobre reputação sem que haja necessidade de conhecimento explícito sobre os modelos usados pelos outros agentes?

1.2 Objetivos

O presente trabalho tem como objetivo propor uma arquitetura orientada a serviços, denominada SOARI, que apoie a interoperabilidade semântica entre agentes com modelos heterogêneos de reputação. Esta arquitetura segue a abordagem híbrida proposta por Visser et al. (2000), o que possibilita que as interações ocorram sem que haja a necessidade dos agentes conhecerem explicitamente o modelo de reputação utilizado pelos outros agentes da sociedade com os quais interagem. Além disto, ela habilita os agentes a se comunicarem mais expressivamente a respeito de reputação.

Na Figura 1.1 é ilustrado o uso da abordagem híbrida na interoperabilidade entre dois agentes com modelos heterogêneos de reputação, utilizando uma arquitetura orientada a serviços.

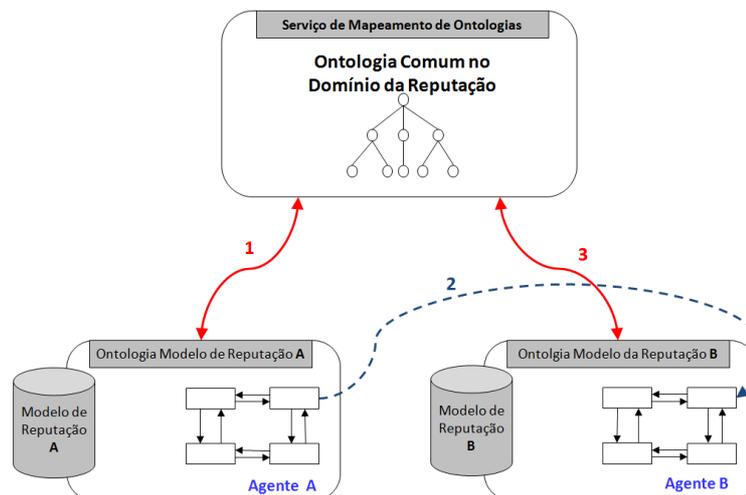


Figura 1.1: Abordagem híbrida para interoperabilidade de modelos de reputação

Considerando esta abordagem e o cenário de avaliação de pinturas, quando Ana (Agente A) solicita a reputação de Clara a Beto (Agente B), ela elabora a solicitação utilizando os conceitos de seu modelo interno de reputação (Modelo de Reputação A). Porém, antes de encaminhá-la a Beto, solicita a tradução dos conceitos de sua ontologia interna pelos conceitos de uma *Ontologia Comum no Domínio da Reputação* ao Serviço de Mapeamento de Ontologias (Indicação 1). Uma vez traduzidos, substituem-se os conceitos na mensagem, que é encaminhada para Beto (Indicação 2). Ao receber a mensagem, Beto solicita a tradução dos conceitos da *Ontologia Comum no Domínio da Reputação* pelos conceitos de seu modelo interno de reputação (Modelo de Reputação B) (Interação 3), substitui-os

na mensagem, que pode então ser processada a fim de responder a Ana.

Para ilustrar o uso da arquitetura, pretende-se realizar alguns experimentos que possibilitem demonstrar os benefícios de uma sociedade composta por agentes com modelos de reputação heterogêneos. Supondo uma sociedade composta por agentes honestos e desonestos, almeja-se responder as seguintes perguntas:

1. *Expressividade de comunicação*: Existe uma melhora de exatidão na identificação dos agentes desonestos por agentes que utilizam uma maior expressividade em suas interações sobre reputação?
2. *Heterogeneidade de modelos de reputação*: Pode-se afirmar que sociedades compostas por agentes com modelos de reputação distintos têm uma melhora de exatidão na identificação dos agentes desonestos em relação as sociedades homogêneas?

1.3 Metodologia

Os métodos utilizados na elaboração deste trabalho envolveram, primeiramente, um levantamento bibliográfico de temas relacionados à reputação. Como o conceito de reputação é interdisciplinar, inicialmente foi realizado um estudo sobre a relevância do conceito de reputação nas áreas de ciências sociais, econômicas e da computação, sendo a análise aprofundada com o estudo da aderência do conceito à área de SMA.

A segunda etapa consistiu no estudo e seleção de dois dos diversos modelos de reputação existentes para servirem de base para os experimentos a serem realizados: *Liar Identification for Agent Reputation* (L.I.A.R.) e *REputation and imAGE* (Repage). Esta etapa envolveu ainda a descrição dos modelos de reputação selecionados em termos ontológicos. As premissas para a seleção destes dois modelos de reputação serão apresentadas mais adiante neste trabalho.

Na terceira etapa foi feito um estudo sobre a interoperabilidade semântica entre agentes, considerando para tal a utilização de ontologias. Assim, um estudo mais detalhado sobre ontologias foi realizado e identificadas duas ontologias que poderiam servir de ontologias comum no domínio da reputação: a ontologia proposta por Chang, Hussain e Dillon (2006) e a Ontologia Funcional da Reputação (FORe)² (CASARE; SICHMAN, 2005a). Devido a maior abrangência de tratamento do tema reputação, a FORe foi estudada em mais detalhes e selecio-

²Do inglês *Functional Ontology of Reputation*.

nada para servir como interlíngua na interoperabilidade entre agentes no domínio da reputação.

Após esta etapa, foi desenvolvida a arquitetura detalhada para permitir a interação sobre reputação entre agentes. De posse desta arquitetura, foi realizada a programação de seus componentes e do modelo de reputação L.I.A.R..

Esta etapa foi seguida pelo mapeamento semântico das ontologias dos modelos de reputação selecionados com a FORe, bem como pela verificação de funcionamento da arquitetura. Tal verificação foi realizada com o uso de um exemplo simplificado de comércio eletrônico.

Por fim, para possibilitar responder às questões formuladas na seção 1.2, a arquitetura foi testada no âmbito das plataformas de simulação ART (FULLAM et al., 2005a) e *Functional Ontology of Reputation in ART Testbed* (FOReART) (BRANDÃO et al., 2007).

1.4 Organização do Texto

O restante deste documento está dividido em mais sete capítulos e dois apêndices. No capítulo 2, são apresentados os dois modelos de reputação utilizados nos experimentos práticos deste trabalho. O capítulo 3 apresenta uma introdução sobre interoperabilidade semântica, sendo o componente principal desta seção as ontologias. As características e componentes principais da Arquitetura Orientada a Serviços (SOA) são apresentadas no capítulo 4. O capítulo 5 descreve detalhadamente a arquitetura proposta, denominada SOARI (*Service Oriented Architecture for Reputation Interaction*). A implementação e demonstração do funcionamento da arquitetura SOARI é apresentada no capítulo 6, através de um exemplo simplificado de comércio eletrônico. No capítulo 7, a arquitetura é testada nas plataformas de simulação ART e FOReART, através da realização e análise de alguns experimentos. Finalmente, as conclusões e trabalhos futuros são apresentados no capítulo 8.

No apêndice A são apresentados os resultados obtidos nos experimentos realizados nas plataformas de simulação ART e FOReART. No apêndice B, diferentes testes estatísticos são realizados sobre tais resultados com a finalidade de selecionar o teste estatístico mais adequado para analisá-los.

2 Modelos de Reputação em SMA

Este capítulo tem por objetivo apresentar as características dos modelos de reputação Repage e L.I.A.R. que serão utilizados nos experimentos realizados neste trabalho. Na seção 2.1 é apresentada uma introdução sobre sistemas de reputação em SMA. A seção 2.2 descreve os aspectos fundamentais envolvidos na noção de reputação. Na seção 2.3, alguns dos principais modelos de reputação são sucintamente apresentados. A seção 2.4 descreve o modelo de reputação Repage, seguida da descrição do modelo de reputação L.I.A.R. na seção 2.5. Por fim, algumas considerações finais sobre os modelos de reputação e a justificativa de sua escolha são apresentadas na seção 2.6.

2.1 Introdução

Um sistema de reputação é um sistema computacional que, segundo Resnick et al. (2000), é responsável por coletar, distribuir e agregar informações a respeito do comportamento passado de seus participantes, auxiliando-os no processo de tomada de decisão para interação com outros participantes. Para que estes sistemas operem de forma efetiva, os participantes têm que satisfazer três propriedades: (1) permanecer no sistema por um longo período de tempo para que possam ocorrer interações futuras, (2) capturar e distribuir informações a respeito das interações atuais, e (3) utilizar as informações capturadas para auxiliar no processo de tomada de decisão.

A base para a construção de sistemas de reputação são os modelos de reputação. Tal modelo é a representação do conhecimento do domínio de reputação em forma de conceitos.

O grande interesse, na atualidade, pelo estudo e a modelagem de reputação possibilitou o surgimento de uma grande quantidade de modelos e sistemas. Estes modelos e sistemas se distinguem uns dos outros por oferecerem formas distintas

de representar a reputação e os aspectos envolvidos.

Segundo Casare e Sichman (2005a), alguns conceitos são utilizados para identificar os diversos aspectos envolvidos na representação da reputação, como descrito a seguir.

2.2 Aspectos da Reputação

Reputação pode ser considerada tanto uma propriedade social quanto um processo social. Ela é uma propriedade social quando considerada como uma representação mental que um agente possui a respeito de outro agente (crença). No entanto, ela é um processo social quando considerada como o resultado da propagação de crenças (meta-crença) na rede social de agentes (CONTE; PAOLUCCI, 2002).

Os conceitos usados para identificar os diversos aspectos de uma reputação são natureza da reputação, papéis dos agentes envolvidos na sua formação e propagação, fontes de informação para sua geração, valor da reputação, fator de avaliação, manutenção e distribuição (CASARE; SICHMAN, 2005a):

Natureza da Reputação - permite identificar a qual tipo de entidade a reputação está associada, por exemplo, trata-se da reputação de um indivíduo, de um grupo, de um produto, de um serviço ou de um evento;

Papel Reputativo - permite identificar quais são os papéis dos agentes envolvidos na formação da reputação. Conte e Paolucci (2002) identificam quatro papéis na formação da reputação: alvo, avaliador, beneficiário e propagador. Muller e Vercouter (2008) estendem esta classificação e definem sete papéis: alvo, participante, observador, avaliador, julgador, beneficiário e propagador;

Tipo da Reputação - permite classificar a reputação baseada na fonte de informação utilizada na sua formação. Uma reputação primária se baseia no contato direto com o agente avaliado, enquanto uma reputação secundária se baseia em informações transmitidas por terceiros;

Valor da Reputação - permite identificar o tipo do valor da reputação em uma palavra ou conceito, ou através de exemplos e justificativas;

Fator de Avaliação - permite identificar quais são os fatores além do comportamento do agente que afetam a avaliação da reputação. No caso de

reputações propagadas, além do valor da reputação também é considerada a reputação do agente que a propagou;

Manutenção da Reputação - permite identificar o método utilizado para agregar os diversos valores da reputação ao longo do tempo;

Distribuição da Reputação - permite identificar o método utilizado para a propagação da reputação, por exemplo, a existência ou não de uma entidade especializada nesta propagação.

2.3 Principais Modelos de Reputação

Conforme mencionado na seção 1.1, existem vários modelos de reputação distintos na literatura. Alguns dos principais são descritos a seguir:

Sporas foi um dos primeiros trabalhos em SMA que propuseram um modelo de cálculo de reputação de um agente. Trata-se de um mecanismo de cálculo de reputação para comunidades *online* fracamente conectadas. Neste modelo, todos os agentes possuem um único valor de reputação, conhecido por todos os membros da comunidade, que é calculado por meio das avaliações sobre as interações diretas fornecidas pelos envolvidos (ZACHARIA; MAES, 2000);

Histos é um mecanismo de cálculo de reputação para comunidades *online* altamente conectadas. Este modelo considera não somente as interações diretas, como também as relações sociais existentes entre os agentes, para o cálculo da reputação. No modelo a reputação é uma propriedade interna de cada agente a respeito de outros agentes (ZACHARIA; MAES, 2000);

Regret valoriza os aspectos sociais envolvidos no cálculo da reputação, considerando para tanto três noções de reputação: *Individual*, *Social* e *Ontológica*. A reputação *Individual* refere-se a como um agente é avaliado pelos outros agentes da sociedade. A reputação *Social* refere-se à avaliação do indivíduo em relação ao grupo social ao qual pertence. A reputação *Ontológica* refere-se à natureza multi-contextual da reputação e depende do contexto da avaliação (SABATER-MIR; SIERRA, 2002);

Yu-Singh propõem um modelo baseado na teoria de evidências de *Dempster-Shafer* para o cálculo da reputação. Neste modelo, a reputação é calculada com base na reputação direta e propagada. Para o cálculo da reputação direta, utiliza-se um conjunto de informações de qualidade de serviço obtida

das interações diretas entre dois agentes. A reputação propagada é inferida com base nas avaliações propagadas por outros agentes, as quais são consideradas proporcionalmente à reputação do agente propagador (YU; SINGH, 2002);

AFRAS tem como principal característica representar os valores da reputação como um conjunto *fuzzy*. Um único valor de reputação, denominado *re-membrance*, é armazenado para cada agente, e sua atualização é realizada usando o cálculo de agregação ponderada. Tal atualização ocorre quando o agente realiza uma nova interação direta com o agente ou quando recebe uma recomendação de um agente que considera confiável (CARTER; MOLINA; MURO, 2003);

Repage é um sistema computacional que emprega uma abordagem cognitiva da reputação proposta por Conte e Paolucci (2002). *Imagem e Reputação* são os dois conceitos centrais deste modelo e representam avaliações sociais. Enquanto a *imagem* é considerada uma crença (avaliação baseada na experiência direta), a *reputação* é considerada uma meta-crença, pois se baseia na informação propagada anonimamente na sociedade (SABATER-MIR; PAOLUCCI; CONTE, 2006).

L.I.A.R. é um modelo não intrusivo para o controle da interação entre agentes em um ambiente aberto e descentralizado. Este modelo se baseia na abordagem social para atingir o controle social. Para tanto, cada agente representa as interações observadas, determina se elas são aceitáveis ou não em relação as normas e pune os agentes para forçá-los a realizarem as interações aceitáveis (MULLER; VERCOUTER, 2008);

Dentre os modelos e sistemas de reputação sucintamente descritos, foram selecionados para utilização neste trabalho, os modelos de reputação Repage e L.I.A.R.. A seleção destes dois modelos justifica-se basicamente pela heterogeneidade dos modelos, o que proporcionará melhores condições para a demonstração e validação do funcionamento da arquitetura proposta. Tal justificativa é apresentada em maiores detalhes na seção 2.6.

A seguir, os dois modelos de reputação selecionados são descritos com maiores detalhes.

2.4 Repage

Como mencionado na seção anterior, Repage é um modelo de reputação, proposto por Conte e Paolucci (2002), que se baseia no modelo de imagem, reputação e seus inter-relacionamentos. Sua arquitetura e funcionamento são apresentados sucintamente a seguir.

2.4.1 Arquitetura e Funcionamento Interno

A arquitetura do modelo Repage, conforme ilustrado na Figura 2.1, é composta de três elementos principais: Memória, Detectores e Analisador (SABATER-MIR; PAOLUCCI; CONTE, 2006).

A memória é o componente principal da arquitetura e é composta de um conjunto de predicados, que co-existem na memória de uso geral do agente e na memória Repage. Esta última, por sua vez, é um subconjunto da memória de uso geral do agente. Qualquer mudança em uma das memórias é imediatamente refletida na outra (SABATER-MIR; PAOLUCCI, 2007).

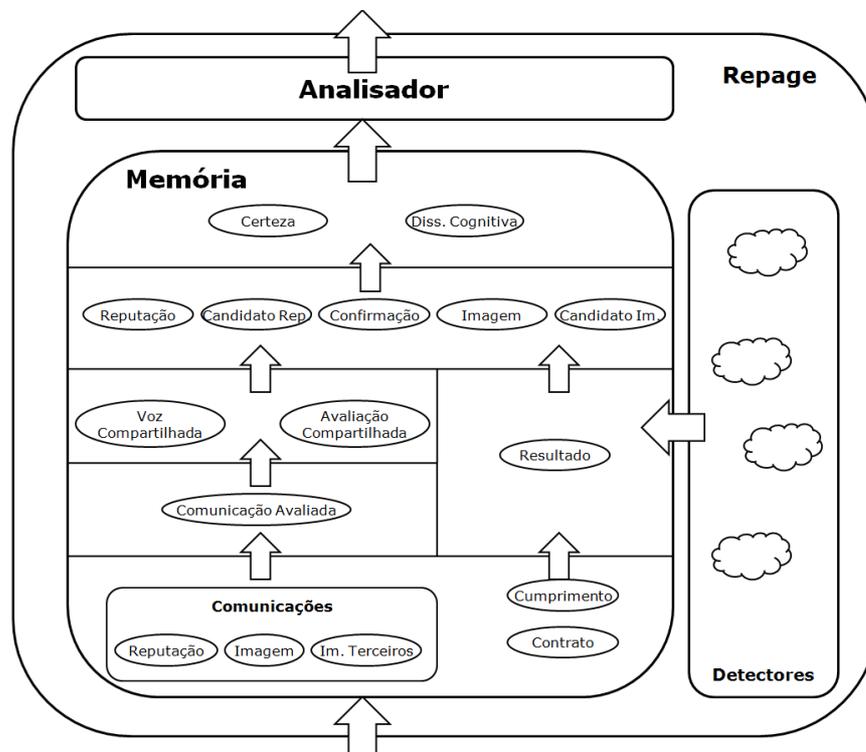


Figura 2.1: Arquitetura do modelo Repage (SABATER-MIR; PAOLUCCI; CONTE, 2006)

Predicados são objetos que contém avaliações sociais. Os predicados são organizados em níveis na memória Repage, e interconectados para refletir sua

interdependência. Cada predicado possui um conjunto de antecedentes e conseqüentes, assim, no caso de algum de seus antecedentes sofrer alguma alteração, o predicado é notificado, recalcula o valor de sua avaliação e grau de confiança, e notifica seus conseqüentes.

A memória do modelo Repage é dividida em níveis. No primeiro nível encontram-se: os *contratos*, que representam os acordos firmados entre os agentes avaliador e avaliado; os *cumprimentos*, que representam os resultados do atendimento dos contratos firmados; e as informações provenientes de agentes informantes como *reputação* - informação anônima sobre o agente avaliado, *imagem* - imagem do agente avaliado frente ao agente avaliador, e *imagem de terceiros* - imagem recebida de outros agentes referente ao agente avaliado. Neste nível, os predicados ainda não foram avaliados pelo sistema.

No segundo nível existem dois predicados: *comunicação avaliada* e *resultado*. A *comunicação avaliada* é o resultado da ponderação das avaliações recebidas de outros agentes com base na imagem que o agente possui do agente informante. O *resultado* contém as avaliações subjetivas (imagem) do agente avaliador com relação às experiências diretas construídas a partir dos *contratos* e *cumprimentos*.

No nível subsequente existem dois predicados: *voz compartilhada* e *avaliação compartilhada*, que são conseqüentes do predicado *comunicação avaliada*. O predicado *voz compartilhada* é formado a partir da ponderação entre a *comunicação avaliada* e a *reputação* do agente propagador, enquanto a *avaliação compartilhada* é constituída da ponderação entre o predicado *comunicação avaliada* e a *imagem* do agente propagador. Os predicados *voz compartilhada* e *avaliação compartilhada* são responsáveis pela formação, respectivamente, dos predicados *reputação candidata* e *imagem candidata*. Estes predicados, quando atingem um determinado limite, transformam-se nos predicados *reputação* e *imagem*, respectivamente. O predicado *confirmação* representa a qualidade da última informação.

O último nível possui dois predicados: *certeza* e *dissonância cognitiva*. O predicado *certeza* representa informações que são totalmente confiáveis, enquanto a *dissonância cognitiva* representa informações que se contradizem, por exemplo, quando a *imagem* e *reputação* a respeito de um mesmo agente em um mesmo contexto são contraditórias.

A propagação das informações dos predicados antecedentes para os conseqüentes entre os níveis da arquitetura é realizada pelos detectores. Eles são unidades de inferência que atuam sobre os predicados da memória Repage, e responsáveis por (1) inferir novos predicados a partir dos já existentes, (2) remover

predicados desnecessários, e (3) formar a rede de dependência entre os predicados (SABATER-MIR; PAOLUCCI, 2007).

Ao atingir o último nível, o analisador é acionado para se comunicar com o módulo planejador do agente. Ele é responsável por propor ações para melhorar a precisão dos predicados e resolver problemas de dissonância na memória Repage (SABATER-MIR; PAOLUCCI; CONTE, 2006).

2.4.2 Avaliações Sociais

O desenvolvimento deste modelo foi baseado na noção de estado mental do agente como crença e meta-crença, que correspondem respectivamente às avaliações sociais *imagem* e *reputação*.

A formação da *imagem* e *reputação* envolve diferentes tipos de agentes, sendo eles, *alvo* - indivíduos, grupos ou artefatos que são avaliados; *avaliador* - indivíduos que formam uma avaliação social sobre outros agentes com base em sua crença; *transmissor* - indivíduos que propagam suas avaliações sociais para outros agentes; e *beneficiário* - indivíduos que usam as avaliações sociais propagadas por outros agentes.

Portanto, no cenário apresentado na seção 1.1 em que a agente Ana solicita ao agente Bob a reputação de Clara, a agente Ana desempenha o papel de *beneficiária*, Bob desempenha os papéis de *avaliador* e *transmissor* e Clara desempenha o papel de *alvo*.

Imagem é a crença individual do agente avaliador a respeito do agente alvo. Quando esta mesma crença é comum a um conjunto de agentes, ela é denominada de *imagem compartilhada*. A *imagem* é formada com base tanto no comportamento do agente alvo e nas normas e padrões aceitos pelo agente avaliador, quanto em informações sobre a *imagem* do agente alvo originárias de outros agentes (SABATER-MIR; PAOLUCCI; CONTE, 2006).

Por outro lado, *reputação* é uma avaliação impessoal que emerge da propagação das avaliações realizadas por outros agentes avaliadores a respeito do agente alvo. Portanto, a *reputação* é uma meta-crença, já que é uma crença a respeito da avaliação realizada por outro agente. (SABATER-MIR; PAOLUCCI; CONTE, 2006).

Enquanto a *imagem* representa o que o agente transmissor realmente acredita e se compromete socialmente, a *reputação* representa a informação que circula na sociedade; neste caso, o agente transmissor não se compromete com a veracidade

e com as conseqüências de sua propagação. Sendo assim, seus conteúdos não necessariamente coincidem.

O valor da *imagem* e da *reputação* é associado ao agente alvo em conjunto com o papel desempenhado por este último. Assim, por exemplo, um agente avaliador de pinturas pode possuir uma boa reputação na avaliação de pinturas impressionistas, no entanto, uma reputação muito ruim como avaliador de pinturas modernistas.

2.4.3 Representação da Avaliação

Neste modelo, a representação das avaliações sociais é dada por um modelo nebuloso. Cada avaliação é composta por uma tupla de 5 valores positivos, os quais representam o grau de aderência da avaliação a cada um dos conjuntos nebulosos associados (*muito ruim* (mr), *ruim* (r), *neutro* (n), *bom* (b), *muito bom* (mb)), além de um valor (s) que representa o grau de confiança que o agente possui nesta avaliação. Tal representação habilita uma maior expressividade ao modelo e é formalmente expressa conforme segue

$$\{[w_{mr}, w_r, w_n, w_b, w_{mb}], s\}$$

onde, $w_i \in [0, 1]$ representa o grau de aderência da avaliação classificada sob os rótulos $i \in \{mr, r, n, mb, b\}$, cuja somatória é igual a 1, e $s \in [0, 1]$ representa o grau de confiança associado a esta avaliação.

Por exemplo, a Figura 2.2 ilustra uma avaliação na qual o agente possui 80% ($s = 0.8$) de confiança na avaliação, que indica que o comportamento do agente avaliado é quase sempre muito ruim (mr) e às vezes ruim (r).

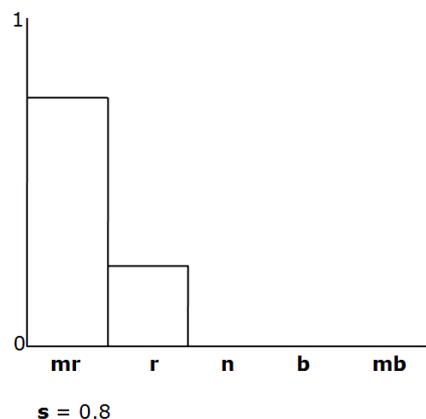


Figura 2.2: Exemplo de Avaliação do modelo Repage (SABATER-MIR; PAOLUCCI, 2007)

2.5 L.I.A.R.

L.I.A.R. é um modelo para a implementação do controle social das interações entre agentes, que possibilita a construção de agentes que podem (1) raciocinar sobre as interações de outros agentes, (2) detectar se eles violaram as regras de comunicação vigentes no sistema, e (3) formar e atualizar reputação a respeito deles. Para tanto, este modelo assume como premissa que todos os agentes possuem conhecimento homogêneo a respeito das regras de comunicação vigentes no sistema (MULLER; VERCOUTER, 2008).

2.5.1 Arquitetura e Funcionamento Interno

Os componentes que formam um agente que utiliza o modelo L.I.A.R. são ilustrados na Figura 2.3, sendo os retângulos a representação dos processos e os ovais a representação dos dados. Além de ilustrar os componentes, o diagrama também determina o comportamento do agente do ponto de vista do fluxo da informação, desde a observação das interações até a atuação através de interações, que é representada pelas arestas direcionadas que interconectam os módulos (MULLER; VERCOUTER, 2008).

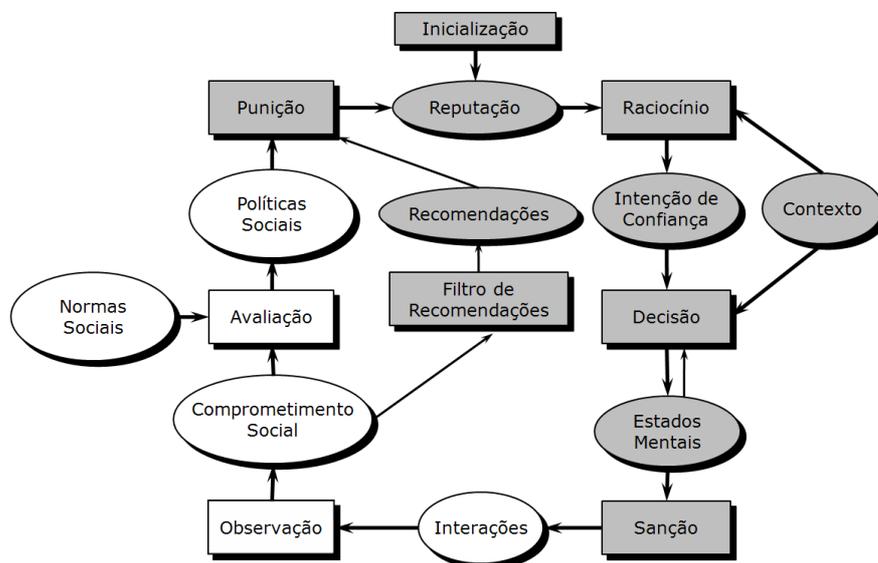


Figura 2.3: Componentes do modelo L.I.A.R. (MULLER; VERCOUTER, 2008)

O diagrama dos componentes do modelo L.I.A.R., ilustrado na Figura 2.3, pode ser subdividido em duas partes, sendo elas denominadas de *Supervisão das Interações* (componentes com fundo branco) e *Modelo de Reputação* (componentes com fundo cinza). Enquanto a *Supervisão das Interações* realiza a avaliação

das interações observadas e compreende os componentes: *Interações*, *Observação*, *Comprometimento Social*, *Normas Sociais*, *Avaliação* e *Políticas Sociais*; o *Modelo de Reputação* utiliza o resultado das avaliações para estimar, no tempo, o comportamento do agente com relação às *Normas Sociais*. O *Modelo de Reputação* é integrado pelos componentes: *Filtro de Recomendações*, *Recomendações*, *Punição*, *Inicialização*, *Reputação*, *Raciocínio*, *Intenção de Confiança*, *Decisão*, *Contexto*, *Estados Mentais* e *Sanção*.

O comportamento de um agente que utiliza o modelo L.I.A.R. inicia-se com a observação das *Interações* com/entre outros agentes no ambiente, as quais são transformadas em *Comprometimentos Sociais* durante o processo de *Observação*. Assim, um *Comprometimento Social* representa uma interação observada pelo agente.

Em seguida, o processo de *Avaliação* compara os *Comprometimentos Sociais* às *Normas Sociais* originando as *Políticas Sociais*.

Enquanto as *Normas Sociais* são regras que os agentes têm que respeitar durante suas interações e são um conhecimento *a priori* dos agentes, as *Políticas Sociais* são uma especificação do conteúdo da norma do ponto de vista do agente avaliador.

Inicialmente, cinco tipos de *Reputação*¹ são criadas pelo processo de *Inicialização*. Posteriormente, estas *Reputações* são atualizadas pelo processo de *Punição*, que utiliza como entrada um conjunto de *Políticas Sociais* geradas durante o processo de *Avaliação* das *Interações*.

Uma vez finalizado o processo de *Inicialização* ou *Punição*, os valores das reputações são utilizados pelo processo de *Raciocínio* do agente que consiste em deduzir a intenção de confiança do agente alvo com base nos valores das reputações. O processo de *Raciocínio* segue um processo em cascata, conforme ilustrado na Figura 2.4. Assim, os valores dos diferentes tipos de reputação são comparados sequencialmente a valores limite pré-definidos superiores e inferiores, sendo a ordem de comparação determinada pela relevância do tipo de reputação. Os valores limites superiores e inferiores, e o valor de relevância são representados, respectivamente, por θ_X^{trust} , $\theta_X^{distrust}$, onde X representa um dos tipos de reputação. Além destes valores limites, existe o valor de relevância que representa o número de interações já feitas com o agente, representado por $\theta_X^{relevance}$, onde X representa um dos tipos de reputação. O valor de relevância também é utilizado

¹Estes tipos são denominados DIbRp, IIBRp, ObsRcbRp, EvRcbRp e RpRcbRp, e serão detalhados na seção 2.5.2.

para determinar se o tipo de reputação deve ou não ser considerado no processo.

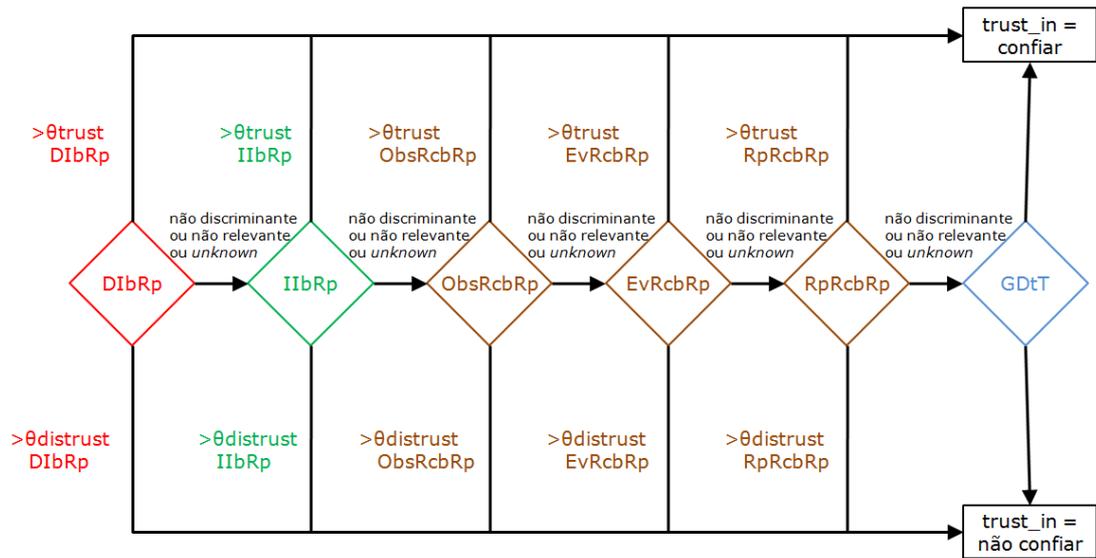


Figura 2.4: Processo de Raciocínio do modelo L.I.A.R. (MULLER; VERCOUTER, 2008)

O processo de *Raciocínio* retorna dois valores: *Intenção de Confiança* ($trust_in$), que é um valor lógico, verdadeiro ou falso, que representa se o agente deve ou não confiar no agente alvo; e o *Valor da Confiança*, que é um valor numérico que representa a confiança na intenção do agente alvo.

Inicialmente, o agente compara o valor do primeiro tipo de reputação ($DIbRp$) com os valores limite correspondentes. Se o valor da reputação for maior que o limite superior, o agente beneficiário tem a intenção de confiar no agente alvo e retornará este valor como resultado do processo. Se, por outro lado, o valor da reputação for menor que o limite inferior, o agente beneficiário tem a intenção de não confiar no agente alvo e retornará este valor como resultado do processo. Entretanto, se o valor da reputação ficar entre os dois limites ou for *unknown*, então o agente verificará o próximo tipo de reputação. Caso nenhum dos tipos de reputação consiga determinar se o agente alvo é confiável ou não, o processo retorna um valor denominado *Disposição Geral de Confiança* ($GDtT$). Tal valor é definido globalmente para o sistema e representa a pré-disposição do agente em confiar ou não em outro agente quando não é possível determinar a confiabilidade do agente alvo através dos valores de reputação.

Por fim, a *Intenção de Confiança* e o *Valor da Confiança* são utilizados pelo processo de *Decisão*. Este processo é responsável por realizar alguma ação sobre

os *Estados Mentais* do agente, sendo este influenciado pelo *Contexto* no qual o agente está inserido.

Além destes processos, existe ainda o processo de *Propagação* e de *Recomendações* que consiste em definir porque, quando, como e para quem as recomendações serão transmitidas. Estas recomendações, quando recebidas por um agente, são avaliadas através do processo de *Filtro de Recomendações* e mantidas para utilização pelo processo de *Punição*, caso sejam consideradas confiáveis.

Assim, imagine a situação em que a agente Ana, dona de uma galeria de artes, estipula uma regra que proíbe que seus funcionários avaliem pinturas impressionistas com um valor inferior a R\$ 30.000,00 (*Norma Social*). Observando o ambiente, a agente Ana observa que o agente Beto, um de seus funcionários, avaliou uma pintura deste tipo de um cliente no valor de R\$ 27.000,00 (*Comprometimento Social*). A agente Ana, com base nesta observação identifica que o agente Beto violou uma das regras da galeria (*Política Social*). Ela então reavalia a reputação do agente Beto (processo de *Punição* atualiza a avaliação social *IbRp*) e com base nesta reavaliação resolve despedi-lo (processo de *Raciocínio e Decisão*).

2.5.2 Avaliações Sociais

Neste modelo, os papéis que os agentes podem desempenhar são: (1) *alvo* - agente que está sendo avaliado; (2) *participante* - agente que interage com o agente alvo; (3) *observador* - agente que observa a mensagem e a transforma em um *Comprometimento Social*; (4) *avaliador* - agente que gera as *Políticas Sociais* a partir dos *Comprometimentos Sociais* e suas *Normas Sociais*; (5) *jugador* - agente que calcula as reputações a partir de um conjunto de *Políticas Sociais*; (6) *beneficiário* - agente que raciocina sobre a reputação e toma alguma ação baseada em seu valor; e (7) *propagador* - agente que transmite informações a respeito de mensagens observadas, *Políticas Sociais* ou reputação.

Diferentemente do modelo Repage, o modelo L.I.A.R. não diferencia as avaliações sociais em *imagem* e *reputação*, sendo elas classificadas de acordo com os papéis desempenhados pelos agentes. Neste modelo, a reputação é classificada em cinco tipos distintos: *Reputação baseada em Interação Direta* (DIbRp), *Reputação baseada em Interação Indireta* (IIbRp), *Reputação baseada em Recomendações Observadas* (ObsRcbRp), *Reputação baseada em Recomendações Avaliadas* (EvRcbRp) e *Reputação baseada em Recomendação de Reputação* (RpRcbRp).

Quando a agente Ana, baseada em interações indiretas que teve com Beto, determina que ele não é confiável e resolve despedi-lo, ela se baseou na *Reputação baseada em Interação Indireta* que possui de Beto. No entanto, se esta avaliação se baseasse na interação direta com o agente Beto, por exemplo quando Beto chegou atrasado às reuniões semanais, sua decisão teria sido tomada com base na *Reputação baseada em Interação Direta*.

Apesar de valiosos, estes dois tipos de reputação são limitados, já que se restringem a um único ponto de vista, da agente Ana. Sendo assim, para enriquecer sua avaliação sobre Beto, Ana poderia consultar outros funcionários da galeria para averiguar se sua decisão é acertada ou se ele merece uma nova chance. Seguindo o modelo L.I.A.R., Ana poderia solicitar que os funcionários lhe transmitam três tipos de informação diferentes: (1) suas observações a respeito do comportamento de Beto formando assim a *Reputação baseada em Recomendações Observadas* sobre Beto; (2) suas *Políticas Sociais* formando assim *Reputação baseada em Recomendações Avaliadas*; ou (3) as reputações que possuem de Beto, possibilitando assim que Ana forme sua *Reputação baseada em Recomendação de Reputação*.

2.5.3 Representação da Avaliação

No modelo L.I.A.R. a representação do valor da reputação é representado por um único valor numérico, no domínio $[-1, +1] \cup unknown$, sendo -1 a representação de uma reputação ruim e $+1$ a representação de uma reputação boa. *Unknown* é o termo usado para representar o desconhecimento sobre a reputação de determinado agente, sendo este o valor inicialmente atribuído à reputação dos agentes.

2.6 Considerações Finais

Este capítulo apresentou os diferentes aspectos de reputação identificados por Casare e Sichman (2005a). Também foram descritas as características e o funcionamento dos sistemas de reputação Repage e L.I.A.R., os quais são utilizados na realização dos experimentos deste trabalho.

Conforme descrito nas seções 2.4 e 2.5, ambos os modelos de reputação realizam avaliações diretas e indiretas de outros agentes. Entretanto, além desta característica são poucas as similaridades entre os dois modelos. Enquanto o modelo Repage possibilita a avalia outros agentes através dos conceitos de *imagem*

e *reputação*, o modelo L.I.A.R. habilita a avaliação através de cinco tipos de reputação. Esta diferença além de quantitativa, também é semântica, não havendo uma relação direta entre os conceitos de ambos. Além desta diferença semântica, outra diferença importante entre os modelos é a forma de representação. O modelo Repage representa as avaliações sociais usando conjuntos nebulosos enquanto o modelo L.I.A.R. utiliza uma representação numérica delimitada. Esta diferença, além de influenciar na forma de representação, também influencia nos métodos de atualização da reputação.

Esta diferença motivou a escolha destes dois modelos para a realização deste trabalho, possibilitando assim a verificação e demonstração do funcionamento da arquitetura proposta de forma mais efetiva e clara.

O critério da breve análise das diferenças apresentada está embasado nos aspectos da reputação, apresentados na seção 2.2. Um resumo do resultado desta análise é apresentada na Tabela 2.1.

Tabela 2.1: Resumo da análise dos modelos de reputação Repage e L.I.A.R.

Critério Modelo	Natureza	Papel	Tipo da Reputação	Valor da Avaliação	Fator de Avaliação	Manutenção	Distribuição (Propagação)
Repage	Reputação de Indivíduo	Alvo, Avaliador, Beneficiário, Transmissor	Primária (<i>imagem</i>) e Secundária (<i>reputação</i>)	Valor Consolidado	Reputação do Avaliador	Método <i>Fuzzy</i>	Agente <i>Transmissor</i>
L.I.A.R.	Reputação de Indivíduo	Alvo, Participante, Observador, Avaliador, Julgador, Beneficiário, Propagador	Primária (<i>DlBp</i> , <i>IIBp</i>) e Secundária (<i>ObsRcbp</i> , <i>EvRcbp</i> , <i>RpRcbp</i>)	Valor Detalhado (baseado em Normas Sociais Políticas Sociais)	Reputação do Avaliador	Método estatístico	Agente Propagador

Diante das diferenças existentes entre modelos de reputação e da diversidade de modelos propostos para SMA, bem como da utilidade de seu uso no controle social, a importância da troca de informação sobre reputação entre agentes com modelos de reputação heterogêneos aumenta. Este problema de interoperabilidade possui diversas formas de solução, uma delas é através do uso de ontologias, conforme apresentado no próximo capítulo.

3 Interoperabilidade Semântica Baseada em Ontologias

Neste capítulo, são apresentados alguns dos elementos fundamentais que possibilitam a interoperabilidade semântica entre agentes, sendo o principal elemento as ontologias. Inicialmente, o significado de interoperabilidade e seus níveis de integração são apresentados na seção 3.1. Na seção 3.2 é apresentado uma introdução sobre ontologias e os elementos a elas relacionados. A descrição de ontologias comum no domínio da reputação são apresentadas na seção 3.3. Em seguida, a seção 3.4 apresenta as possíveis abordagens da aplicação de ontologias na interoperabilidade de sistemas. Finalmente, na seção 3.5 são apresentadas as justificativas quanto às escolhas referenciadas nesta seção.

3.1 Introdução

Interoperabilidade é a habilidade de dois ou mais sistemas ou componentes de compartilhar informação e de utilizar a informação compartilhada (IEEE, 1991). Para tanto, é necessário que os sistemas sejam capazes de acessar, processar e interpretar esta informação.

Em sociedades de agentes, caracterizadas por serem abertas e descentralizadas, estas atividades podem estar comprometidas devido à heterogeneidade dos agentes. Visser et al. (2000) propõem três diferentes níveis de integração para a obtenção da interoperabilidade entre agentes, sendo eles:

- **Integração Sintática** - Envolve a especificação da informação no nível sintático. *eXtensible Markup Language* (XML) e *HyperText Markup Language* (HTML) são exemplos de padrões que satisfazem este nível de integração;
- **Integração Estrutural** - Envolve a re-formatação de diferentes estruturas de dados para uma estrutura homogênea. Normalmente, neste nível de

integração são utilizados *middlewares* que definem regras de mapeamento entre diferentes estruturas de informação;

- **Integração Semântica** - Envolve a resolução de conflitos semânticos gerados em decorrência da impossibilidade de mapeamento um para um de conceitos entre os modelos envolvidos.

Este trabalho adota o nível de integração semântica, a fim de viabilizar a interoperabilidade semântica entre noções de reputação trocadas entre agentes heterogêneos.

Entende-se por interoperabilidade semântica a capacidade de dois ou mais sistemas heterogêneos e distribuídos trabalharem em conjunto, compartilhando as informações com entendimento comum de seu significado (BURANARACH, 2001).

No entanto, para que a interoperabilidade semântica seja possível, alguns requisitos precisam ser atendimentos: o estabelecimento de uma representação semântica comum e a criação de uma relação de correspondência entre os conceitos de diferentes sistemas.

Segundo Visser et al. (2000), o uso de ontologias é uma das possibilidades mais promissoras para garantir a interoperabilidade semântica de sistemas. Em primeiro lugar, elas fornecem um vocabulário comum com uma semântica bem definida e possibilitam que as informações sejam expressas de forma estruturada. Além disso, possibilitam a definição de relações de correspondência entre os conceitos dos sistemas através de diferentes mecanismos (KALFOGLOU; SCHORLEMMER, 2003).

3.2 Ontologias

3.2.1 Definição

Conhecida no ramo da filosofia como uma *teoria sobre a natureza da existência*, as ontologias têm sido encaradas de maneira consideravelmente diferente no universo computacional.

Segundo Gruber (1993), uma ontologia é uma especificação explícita de uma conceitualização, sendo considerada conceitualização uma representação formal de um domínio do conhecimento. Tal representação é composta por objetos e seus inter-relacionamentos.

As ontologias podem ser classificadas considerando diferentes características.

Guarino (1998), por exemplo, as classifica de acordo com o seu nível de generalidade:

- **Ontologia de Alto Nível** - descreve conceitos gerais ou conhecimento de senso comum que são independentes de um problema ou domínio particular;
- **Ontologia de Domínio** - descreve um vocabulário geral relacionado a um determinado domínio;
- **Ontologia de Tarefa** - descreve conceitos relacionados a execução de uma tarefa ou atividade específica;
- **Ontologia de Aplicação** - descreve conceitos que dependem tanto de um domínio específico como de uma tarefa específica, e geralmente são uma especialização de ambos.

Neste trabalho é utilizada uma Ontologia de Domínio para viabilizar a interoperabilidade entre agentes sobre reputação.

3.2.2 Linguagens de Representação de Ontologias

O elemento chave para a construção de ontologias é a definição de uma linguagem de representação adequada. A linguagem deve ter semântica bem-definida, ser expressiva o suficiente para descrever inter-relacionamentos complexos e restrições entre os objetos, e ser capaz de manipular e inferir automaticamente com limites aceitáveis de tempo e de recursos (MCILRAITH; MARTIN, 2003).

Algumas das linguagens desenvolvidas para expressar formalmente as ontologias são brevemente descritas na seqüência:

KIF (*Knowledge Interchange Format*) (GENESERETH; FIKES, 1992) baseia-se em uma extensão do cálculo de predicados e seu principal objetivo é permitir a troca de conhecimento entre sistemas heterogêneos. Ela é uma linguagem com alto poder expressivo, no entanto, é uma linguagem de baixo nível para a representação de ontologias;

OntoLingua (FARQUHAR; FIKES; RICE, 1996) foi desenvolvida pela Universidade de Stanford. O termo pode tanto referenciar a linguagem de ontologia baseada em *Knowledge Interchange Format* (KIF) e *Frame Ontology* (GRUBER, 1993), como o ambiente *Ontolingua Server*. A Ontolingua, enquanto uma linguagem, permite descrever ontologias de forma a ser compatível com

outras linguagens de representação através da definição de classes, relações, funções, indivíduos e axiomas (GRUBER, 1993);

DAML+OIL (HORROCKS, 2002) é uma versão atualizada de *DARPA Agent Markup Language* (DAML) e *Ontology Inference Layer* (OIL) que disponibiliza um conjunto variado de construtores para o desenvolvimento de ontologias. Ela segue uma abordagem orientada a objetos e possibilita a descrição da ontologia em termos de *classes* e *propriedades*;

OWL (*Web Ontology Language*) (MCGUINNESS; HARMELEN, 2004) é uma revisão da DAML+OIL, sendo recomendada pela *World Wide Web Consortium* (W3C)¹ para a representação de ontologias para a Web Semântica. Ela se subdivide em três diferentes linguagens, que têm expressividades distintas e o propósito de atender a diferentes grupos de usuários:

OWL Lite - atende aos usuários que necessitam de hierarquia de classificação e restrições simples;

OWL DL - atende aos usuários que necessitam de máxima expressividade sem perda da completude computacional (i.e., garante-se que todas as conclusões são computáveis), e decidibilidade (i.e., todos os cálculos terminam em tempo finito);

OWL Full - atende aos usuários que desejam máxima expressividade e a liberdade sintática da *Resource Description Framework* (RDF) (KLYNE; CARROLL, 2004) sem garantias computacionais.

Neste trabalho, é adotada a linguagem OWL DL devido a sua expressividade e seu amplo uso pela comunidade acadêmica. Além disso, ela possibilita a inferência lógica por mecanismos de raciocínio automático que implementam um algoritmo correto e completo para raciocínio com classes. Tais mecanismos de raciocínio são implementados em motores de inferência, descritos a seguir.

3.2.3 Ambientes para Raciocínio Automático com Ontologias

Apesar do propósito mais aparente das ontologias ser a representação de conhecimento de um domínio específico, o seu maior benefício origina-se da possibilidade de extração de informações implícitas a partir das informações explicitamente representadas. A esta extração de informações é dado o nome de inferência lógica, a qual pode ser automatizada com o uso de motores de inferência.

¹<http://www.w3c.org/>.

A seguir são brevemente descritos alguns dos principais motores de inferência existentes na atualidade para linguagem OWL DL.

FaCT++² foi desenvolvido na Universidade de Manchester. Trata-se de uma nova geração do motor de inferência *Fast Classification of Terminologies* (FaCT) (HORROCKS, 1999) para lógica descritiva. Reutiliza os algoritmos corretos e completos do tipo *tableaux* usados pelo FaCT, porém, possui uma arquitetura interna diferente, que em conjunto com o desenvolvimento em C++, melhorou seu desempenho e portabilidade. Uma de suas características é solucionar problemas de classificação aplicando algoritmos do tipo *tableaux* após transformar tais problemas em problemas de satisfação;

RACERPro (*Renamed ABox and Concept Expression Reasoner*) (RACER SYSTEMS GMBH AND CO. KG, 2007) é um motor de inferência para lógica descritiva, desenvolvido em *Common Lisp*, mantido e comercializado pela empresa alemã *Racer Systems GmbH & Co. KG*. Suas principais funcionalidades são a verificação de consistência de classes, classificação de classes, teste de instâncias e recuperação de instâncias. Apesar de possuir algoritmos corretos e completos para o raciocínio com classes, o mesmo não ocorre para o raciocínio com instâncias;

Pellet (SIRIN et al., 2007) foi desenvolvido na Universidade de Maryland, sendo o primeiro motor de inferência desenvolvido para raciocinar exclusivamente em OWL. Dessa forma, pode explorar as funcionalidades da OWL, oferecendo raciocínio com instâncias e consultas sobre instâncias. Ele também possibilita o raciocínio automático sobre OWL Full; entretanto, antes de realizar tal raciocínio transforma essa ontologia em OWL DL utilizando heurísticas. Outra funcionalidade disponível é a identificação da causa raiz de uma contradição.

Dentre os motores de inferência referenciados, neste trabalho é utilizado o motor de inferência *Pellet*. As razões que guiaram esta escolha foram sua aderência natural ao raciocínio sobre ontologias descritas em OWL DL, ser desenvolvido em linguagem de programação Java, possuir integração com o ambiente de edição *Protégé-OWL Plugin* (HORRIDGE et al., 2004), ter sua forma de licenciamento condizente ao propósito acadêmico e ser amplamente utilizado pela comunidade científica.

²<http://owl.man.ac.uk/factplusplus/>.

3.2.4 Operações com Ontologias

Algumas das terminologias usadas para referenciar operações com ontologias podem ser encontradas em (KALFOGLOU; SCHORLEMMER, 2003). No entanto, para evitar possíveis desentendimentos em relação ao seu significado, segue a descrição das principais definições usadas neste trabalho:

- **Alinhamento** - processo que estabelece as relações binárias entre os conceitos de duas ontologias;
- **Mapeamento** - processo que determina a relação de equivalência entre os conceitos de duas ontologias. Consiste em um conjunto de funções que associam os conceitos e relações de uma ontologia aos conceitos e relações de outra ontologia. Assume-se que, se dois conceitos são equivalentes, eles significam a mesma coisa ou algo similar (DOU; MCDERMOTT; QI, 2003);
- **Tradução** - processo que aplica as funções de mapeamento na tradução de sentenças de uma ontologia para outra.

As operações de mapeamento e alinhamento podem ser desempenhadas de três diferentes maneiras (BOCKTING, 2005):

- **Manual** - O mapeamento ou alinhamento é totalmente realizado por especialistas no domínio, ou por pessoas com conhecimento na captura e transcrição da informação para ontologias. Este é um processo tedioso, sujeito a falhas e impossível de ser realizado em grande escala;
- **Semi-Automática** - O mapeamento ou alinhamento é realizado por um sistema, no entanto, há a necessidade da intervenção humana para realização de algumas tarefas. Normalmente, estes sistemas além de realizar suas tarefas, são capazes de auxiliar o usuário na identificação e correção de inconsistências inseridas involuntariamente por este último. Exemplos destes sistemas são PROMPT (NOY; MUSEN, 2000) e Chimaera (MCGUINNESS et al., 2000);
- **Automática** - O mapeamento ou alinhamento é totalmente realizado por um sistema, sem a intervenção humana, que requer técnicas especiais de mapeamento. Uma taxonomia destas técnicas é disponibilizada em Madhavan, Bernstein e Rahm (2001). Apesar disso, na atualidade não é possível a utilização desta forma de mapeamento quando se deseja obter um resultado

com alto grau de qualidade, pois são necessárias mais informações que aquelas disponibilizadas nas ontologias. Mesmo com estas limitações, alguns sistemas se propõem a desempenhar estas operações automaticamente, dentre os quais, Cupid (MADHAVAN; BERNSTEIN; RAHM, 2001) e GLUE (DOAN et al., 2003).

Neste trabalho, conforme mostrado na seção 6.3.1.2, a operação de alinhamento é realizada de modo manual.

3.3 Ontologias de Domínio da Reputação

Verifica-se através da análise da literatura que existe pouco esforço na definição da reputação em termos ontológicos, apesar do conhecimento dos ganhos que este estudo traria ao entendimento do conceito de reputação e identificação de seus diversos componentes. Um dos motivos que possivelmente ocasiona esta falta de interesse é a não existência de um consenso único sobre a definição do conceito de reputação.

Apesar de poucos, alguns esforços na definição da reputação em termos ontológicos foram realizados por Casare e Sichman (2005a) e Chang, Hussain e Dillon (2006). A seguir é feita uma breve descrição de suas características.

3.3.1 Ontologia de Chang, Hussain e Dillon (2006)

Na proposta de Chang, Hussain e Dillon (2006), identifica-se dois tipos de reputação, sendo a distinção entre eles realizada com base no grau de sofisticação do processo de agregação das recomendações. A reputação *básica* é definida como sendo a agregação de recomendações solicitadas a terceiros. Já a reputação *avançada* estende a definição *básica* e considera para a agregação das recomendações não somente o valor da recomendação, mas também a reputação dos fornecedores desta recomendação. Analisando esta proposta, verifica-se que ela é bastante simples, pois considera somente o aspecto *Tipo da Reputação* descrito na seção 2.2. Além disso, a ontologia proposta não está representada formalmente através de uma linguagem comumente utilizada na representação de ontologias (vide alguns exemplos na seção 3.2.2), mas sim através da linguagem *Unified Modelling Language-Object Constraint Language* (UML-OCL).

3.3.2 FORe

A ontologia de reputação proposta por Casare e Sichman (2005a), denominada Ontologia Funcional da Reputação (FORe) representa grande parte do conhecimento científico sobre reputação originado nas áreas das Ciências Humanas e da Inteligência Artificial. Ela considera todos os aspectos da reputação descritos na seção 2.2 e encontra-se formalmente descrita em OWL DL.

A FORe foi concebida baseada na Ontologia Funcional do Direito (VALENTE, 1995), da qual utiliza os conceitos do mundo legal para modelar os conceitos existentes no mundo social. Para tanto, a modelagem da FORe estende alguns aspectos da Ontologia Funcional do Direito, quais sejam (CASARE, 2005):

- ampliação do conceito de norma jurídica para norma social;
- substituição da reação jurídica (sanção ou compensação) pela reação social (má reputação ou boa reputação);
- interiorização do mecanismo de controle social, incorporando-o a “mente” do agente.

Desta última extensão, pode-se identificar que a FORe foi modelada seguindo a abordagem social para a resolução dos problemas de obtenção da ordem social.

Segundo (CASARE, 2005), a FORe é composta de duas grandes categorias: *Conhecimento de Reputação* e *Conhecimento de Senso Comum*. A categoria de *Conhecimento de Senso Comum* representa os elementos de senso comum envolvidos diretamente com a noção de reputação. Já a categoria de *Conhecimento de Reputação* representa o conhecimento do sistema reputativo e é subdividida em quatro outras categorias: *Conhecimento Reputativo*, *Conhecimento de Responsabilidade*, *Conhecimento Normativo* e *Conhecimento do Mundo*.

Como a FORe não foi construída somente para atender ao propósito de interoperabilidade semântica, sua utilização neste trabalho limita-se aos conceitos necessários para o mapeamento de modelos de reputação. Um maior detalhamento da ontologia pode ser obtido em (CASARE, 2005).

No que concerne à categoria *Conhecimento Reputativo*, serão utilizados os seguintes conceitos:

Tipo da Reputação - distingue uma reputação com base na origem da informação que possibilitou sua formação. Por exemplo, quando a agente

Ana avalia a reputação da agente Clara, baseada nas suas experiências diretas ou nas observações das interações de Clara, a reputação formada é denominada *Reputação Primária*. No entanto, se ela avaliar a agente Clara baseada nas informações recebidas do agente Beto, a reputação formada é denominada *Reputação Secundária*.

Papel Reputativo - representa os papéis desempenhados por uma entidade envolvida em um processo reputativo. Por exemplo, Ana, desempenhando o papel de *Receptor*, solicita informações sobre a agente Clara (*Alvo*) ao agente Beto (*Avaliador*). Beto além de avaliar a reputação da agente Clara também desempenha os papéis de *Transmissor*, quando transmite a reputação de Clara para Ana e *Mantenedor*, já que ele atualiza a reputação que possui de Clara devido à recepção de novas informações.

Componente de Cálculo da Reputação - agrupa os aspectos envolvidos na geração do valor da reputação. Os aspectos representam as características consideradas na avaliação, bem como a relevância associada a cada uma delas.

Já os conceitos utilizados da categoria *Conhecimento de Senso Comum* são os seguintes:

Entidade Reputativa - representa as entidades que são capazes de desempenhar ao menos um papel reputativo, podendo a entidade ser um *Agente* ou *Não Agente*. Entende-se por *Agente* a entidade que pode agir por si própria, enquanto por *Não Agente* aquela que não desempenha nenhum papel reputativo, assim sendo somente alvo da avaliação da reputação, como uma empresa.

Tempo - representa “as sucessões dos anos, dias e horas que envolvem a noção de presente, passado e futuro”.

Fonte de Informação - representa os elementos que podem atuar como fonte da informação, sendo elas: *Experiência Direta* - consiste nas informações adquiridas através das interações diretas; *Observação* - consiste nas informações obtidas através de observações das interações; *Herança de Grupo* - consiste nas informações adquiridas das características de um grupo ao qual o agente pertence; *Informação Recebida de Terceiro* - consiste nas informações transmitidas na rede social; e *Preconceito* - consiste nas informações baseadas em preconceitos.

Devido a suas características, a FORe é adotada neste trabalho para servir como ontologia no domínio da reputação.

3.4 Abordagens para Interoperabilidade Semântica

De acordo com Visser et al. (2000), existem três abordagens distintas para a aplicação de ontologias na obtenção da interoperabilidade semântica, sendo elas:

- **Centralizada** - todos os agentes do sistema utilizam uma única ontologia comum de domínio para representar a informação internamente e durante a interação com outros agentes;
- **Descentralizada** - cada agente do sistema utiliza sua própria ontologia de domínio para representar a informação internamente e durante a interação com outros agentes. Assim, para que haja interoperabilidade completa entre todos os agentes do sistema, é necessário que cada agente possua o mapeamento de sua ontologia para todas as outras ontologias existentes no sistema;
- **Híbrida** - cada agente do sistema utiliza uma ontologia de domínio para representar a informação internamente, porém, utiliza uma ontologia comum de domínio durante a interação com outros agentes. Assim, cada agente deve possuir ao menos um único mapeamento de sua ontologia para a ontologia comum de domínio para que seja possível a interoperabilidade.

Segundo Casare e Sichman (2005b), a abordagem híbrida é a mais vantajosa para a integração semântica de modelos de reputação. Em primeiro lugar, ela não se limita a heterogeneidade de modelos de reputação em SMA, dado que cada agente pode operar com seu próprio modelo. Em segundo lugar, não envolve um procedimento de alinhamento complexo entre ontologias em tempo de execução, já que para entrar na sociedade a ontologia do modelo de reputação do agente deve estar alinhada a ontologia comum no domínio da reputação.

Outra vantagem da abordagem híbrida no domínio da reputação não mencionada é o fato dos agentes não conhecerem o modelo de reputação uns dos outros. Como o funcionamento dos modelos de reputação é conhecido, tal ignorância dificulta que agentes maliciosos propaguem valores de reputação com a intenção de manipular o funcionamento do sistema, alterando os resultados do cálculo da reputação a seu favor.

3.5 Considerações Finais

Segundo Uschold et al. (1996), as ontologias possuem três propósitos de aplicação, sendo eles, comunicação, engenharia de sistemas e interoperabilidade. Destas aplicações a interoperabilidade é identificada por Uschold et al. (1996) como a principal delas.

Neste trabalho propõe-se uma arquitetura para possibilitar a interoperabilidade semântica entre agentes com modelos de reputação heterogêneos. Para tanto, a abordagem híbrida de ontologias é utilizada. Esta abordagem requer a existência de uma ontologia comum no domínio da reputação para servir de interlíngua. Devido às suas características mais abrangentes, a FORe é adotada.

A arquitetura proposta neste trabalho segue o paradigma de orientação a serviços, sendo suas principais características descritas a seguir.

4 Arquiteturas Orientadas a Serviços

Este capítulo tem por finalidade apresentar o modelo de Arquitetura Orientada a Serviços (SOA)¹ que é adotada na arquitetura proposta neste trabalho. Primeiramente, o modelo SOA é introduzido na seção 4.1. A seguir, a seção 4.2 apresenta as principais características deste modelo. A seção 4.3 apresenta uma visão geral sobre Web Services. Finalmente, algumas considerações finais são apresentadas na seção 4.4.

4.1 Introdução

Arquitetura de software é a descrição de um sistema, na qual são definidos seu propósito, funções, propriedades e interfaces. Segundo (IEEE, 2000), ela também descreve como os componentes fundamentais de um sistema se relacionam entre si e com o ambiente, bem como os princípios que direcionam seu projeto, operação e evolução.

Existem dois tipos de acoplamento entre componentes de um sistema: *fraco* e *forte*. No acoplamento *fraco*, os componentes possuem nenhuma ou algumas dependências. No acoplamento *forte*, estes possuem muitas dependências.

A Arquitetura Orientada a Serviços (SOA) é um modelo arquitetural que surgiu da necessidade de colaboração e integração de componentes funcionais de forma interoperável e com fraco acoplamento. Neste contexto, os componentes funcionais são denominados *serviços*.

Apesar de não existir uma definição única de *serviço*, pode-se defini-lo como um componente de software, fisicamente independente, que está disponível em um ponto particular de uma rede de computadores, recebe e envia mensagens, e comporta-se de acordo com sua especificação (ERL, 2007).

¹Do inglês, *Service Oriented Architecture*.

Os aspectos funcionais dos serviços são descritos utilizando metadados, os quais são publicados para que seus consumidores possam descobri-los e obter toda a informação necessária para requisitá-los.

Um dos principais aspectos do modelo SOA é a distinção entre a implementação e a interface do serviço. Do ponto de vista do consumidor, um serviço é simplesmente uma entidade que responde a um formato particular de requisição, não sendo de seu interesse os detalhes sobre as ações executadas para o seu atendimento (MCGOVERN et al., 2003).

O modelo SOA é composto por diversas entidades que integradas sustentam o paradigma “publicar/descobrir/requisitar” (Figura 4.1).

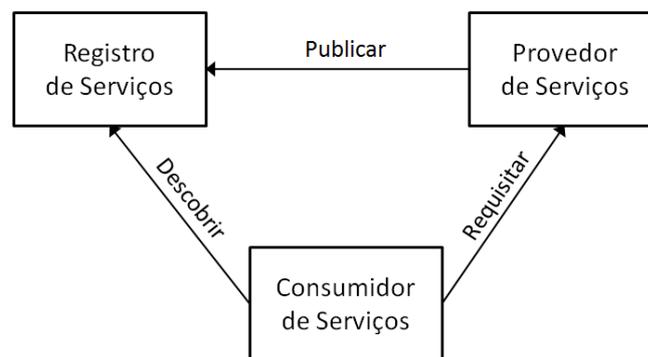


Figura 4.1: Arquitetura Orientada a Serviços (MCGOVERN et al., 2003)

Segundo este paradigma, as principais entidades que o sustentam são:

Provedor de Serviços - componente de software que publica o *Contrato de Serviço* no *Registro de Serviços* e o executa quando requisitado pelo *Consumidor de Serviços*;

Consumidor de Serviços - aplicação, módulo de software ou outro serviço que requisita um serviço a um *Provedor de Serviços*. Deve descobrir um serviço usando o *Registro de Serviços* e requisitá-lo enviando uma mensagem formatada conforme especificado no *Contrato de Serviço*;

Registro de Serviços - local de armazenamento da descrição dos serviços publicados pelos *Provedores de Serviços* e disponíveis para requisição pelos *Consumidores de Serviços*;

Contrato de Serviço - definição do serviço, onde se especifica como o *Consumidor de Serviços* deve requisitar um serviço ao *Provedor de Serviços*. O *Contrato de Serviços* é publicado pelo próprio *Provedor de Serviços* no *Registro de Serviços*.

Além das entidades que atendem a esse paradigma, podem existir duas outras entidades no modelo SOA, sendo elas: o *Roteador de Serviços* e a *Alocação de Serviços* (MCGOVERN et al., 2003).

4.2 Características Principais

As *arquiteturas de software* refletem os diferentes princípios e premissas assumidas por seus idealizadores. Dentre as características identificadas por McGovern et al. (2003), as principais características de uma arquitetura orientada a serviços são as seguintes:

- **Dinamismo** - refere-se a possibilidade dos *Provedores de Serviços* publicarem ou retirarem a publicação de seus serviços em tempo de execução, bem como a possibilidade de os *Consumidores de Serviços* descobrirem e requisitarem um serviço também em tempo de execução;
- **Interoperabilidade** - refere-se a possibilidade de permitir a comunicação entre sistemas que utilizam diferentes plataformas ou linguagens. Tal característica é possível devido a separação entre a implementação e a descrição do serviço;
- **Fraco Acoplamento** - O acoplamento entre os *Provedores de Serviços* e os *Consumidores de Serviços* é *fraco*, já que os consumidores não dependem diretamente da implementação dos serviços, mas de sua interface, que pode ser implementada por diferentes *Provedores de Serviços*.

4.3 Web Services

Web Services representa uma das concepções mais bem sucedidas de SOA. Ele consiste em um conjunto de padrões e protocolos largamente difundidos e aceitos pela comunidade de computação, os quais possibilitam a interoperabilidade entre sistemas independentemente de plataforma, sistema e linguagem (HASS; BROWN, 2004).

Tal interoperabilidade é possibilitada com o uso de padrões baseados em XML, que inclui: *Simple Object Access Protocol* (SOAP), *Web Services Description Language* (WSDL), *Universal Description, Discovery, and Integration* (UDDI) e outros protocolos, que são brevemente descritos a seguir.

4.3.1 XML

eXtensible Markup Language (XML) é uma metalinguagem baseada na *Standard Generalized Markup Language* (SGML) (ISO, 1986), que foi projetada para facilitar a troca de documentos estruturados na Internet.

Um documento XML é basicamente constituído de elementos definidos por *marcadores* que identificam e delimitam porções deste documento. Esses elementos são estruturados de forma hierárquica e podem conter atributos que fornecem informações adicionais a seu respeito (Figura 4.2).

```
<?xml version="1.0" ?>
<reputacao>
  <agente nome="A">
    <DIbRp>0.8</DIbRp>
    <IIbRp>0.7</IIbRp>
  </agente>
  <agente nome="B">
    <DIbRp>-0.6</DIbRp>
    <IIbRp>-0.1</IIbRp>
  </agente>
</reputacao>
```

Figura 4.2: Exemplo de arquivo em formato XML

Neste exemplo, a primeira linha identifica o arquivo como sendo um documento XML. O primeiro elemento do documento é o elemento *<reputação>*. Este elemento possui outros dois elementos *<agente>*, que contêm o atributo *nome* que os distinguem (A e B). Cada um dos elementos *<agente>* possui ainda outros dois subelementos, *<DIbRp>* e *<IIbRp>*, componentes do modelo L.I.A.R..

Normalmente, um documento XML é associado a uma especificação, denominada *esquema*, que define um conjunto de regras sobre a formação e estrutura do documento XML, bem como a ordem, local e tipo de elementos e atributos que podem fazer parte deste documento. Com base nesta especificação é possível determinar se um documento XML está *bem-formado*. Por documento *bem-formado* entende-se todo documento que segue as regras definidas pelo esquema ao qual está associado.

4.3.2 SOAP

Simple Object Access Protocol (SOAP) é um protocolo baseado em XML para a troca de informações estruturadas entre sistemas em ambientes descentralizados e distribuídos (MITRA; LAFON, 2007).

O item básico de transmissão deste protocolo é uma mensagem composta

por três partes: *envelope*, *header* e *body*. O *envelope* define regras específicas de encapsulamento dos dados que serão transmitidos e é obrigatório. O *header* é opcional e pode ser utilizado para indicar a realização de algum processamento adicional por alguma entidade intermediária entre o cliente solicitante e o serviço solicitado. Isto é possível já que uma mensagem SOAP passa através de diversos nós intermediários no caminho entre o cliente solicitante e o serviço solicitado. O *body* carrega o conteúdo da mensagem destinada ao receptor final.

Por exemplo, a Figura 4.3 ilustra uma mensagem SOAP solicitando a tradução do conceito *PropagatedReputation* da *Functional Ontology of Reputation* (FORE) para a ontologia Repage.

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns2:translateConcept xmlns:ns2="http://webservice.mapper.otservices">
      <ns2:fromConcept>PropagatedReputation</ns2:fromConcept>
      <ns2:ontFrom>
        <ns1:type xmlns:ns1="http://translationrepository.mapper.otservices/xsd">1</ns1:type>
        <ns1:uri xmlns:ns1="http://translationrepository.mapper.otservices/xsd">FORE</ns1:uri>
        <ns1:version xmlns:ns1="http://translationrepository.mapper.otservices/xsd">1</ns1:version>
      </ns2:ontFrom>
      <ns2:ontTo>
        <ns1:type xmlns:ns1="http://translationrepository.mapper.otservices/xsd">2</ns1:type>
        <ns1:uri xmlns:ns1="http://translationrepository.mapper.otservices/xsd">REPAGE</ns1:uri>
        <ns1:version xmlns:ns1="http://translationrepository.mapper.otservices/xsd">1</ns1:version>
      </ns2:ontTo>
    </ns2:translateConcept>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 4.3: Exemplo de mensagem SOAP

4.3.3 WSDL

Web Services Description Language (WSDL) é uma linguagem baseada em XML utilizada para descrever serviços. Trata-se de um documento escrito em XML que, além de descrever o serviço, especifica como acessá-lo e quais as operações ou métodos estão disponíveis (CHRISTENSEN et al., 2001).

Um documento WSDL especifica os aspectos técnicos para requisição de serviços. Ele não pode ser considerado um *Contrato de Serviço*, conforme definido no modelo *Service Oriented Architecture* (SOA), pois não possibilita a definição de atributos não funcionais, tais como sua disponibilidade e desempenho.

Para especificar os aspectos técnicos, o WSDL é organizado em duas seções lógicas que correspondem a uma descrição abstrata do serviço e uma descrição concreta.

A descrição abstrata é composta pelos elementos *types*, *message* e *port type*, enquanto a descrição concreta compõe-se dos elementos *binding* e *service*:

- *types* - descreve todos os tipos de dados utilizados na comunicação. Por exemplo, os tipos complexos de dados *translateConcept* e *translateConceptResponse* são descritos na Figura 4.4.

```

<wsdl:types>
  <xs:schema xmlns:ax223="http://rmi.java/xsd"
    xmlns:ax225="http://translationrepository.mapper.otsservices/xsd"
    xmlns:ax227="http://webservice.mapper.otsservices/xsd" attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://webservice.mapper.otsservices">
    <xs:import namespace="http://rmi.java/xsd" />
    <xs:import namespace="http://translationrepository.mapper.otsservices/xsd" />
    <xs:import namespace="http://webservice.mapper.otsservices/xsd" />
    <xs:element name="translateConcept">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="fromConcept" nillable="true" type="xs:string" />
          <xs:element minOccurs="0" name="ontFrom" nillable="true" type="ax225:Ontology" />
          <xs:element minOccurs="0" name="ontTo" nillable="true" type="ax225:Ontology" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="translateConceptResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="return" nillable="true" type="ax227:MapperList" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>

```

Figura 4.4: Exemplo de definição do elemento *types* do WSDL

- *message* - descreve a mensagem de requisição ou de resposta. Neste elemento é informado o nome da mensagem que pode conter zero ou mais elementos que representam os parâmetros da mensagem ou de valor de retorno. Na Figura 4.5 são descritas uma mensagem de requisição *translateConceptRequest* e uma outra mensagem de resposta *translateConceptResponse*, que estão associadas aos elementos *translateConcept* e *translateConceptResponse* respectivamente.

```

<wsdl:message name="translateConceptRequest">
  <wsdl:part name="parameters" element="xsd:translateConcept" />
</wsdl:message>
<wsdl:message name="translateConceptResponse">
  <wsdl:part name="parameters" element="xsd:translateConceptResponse" />
</wsdl:message>

```

Figura 4.5: Exemplo de definição do elemento *message* do WSDL

- *port type* - descreve o conjunto de operações que podem ser executadas por um serviço. Por exemplo, na Figura 4.6 é definida uma única operação denominada *translateConcept* com seus parâmetros de entrada e saída *translateConceptRequest* e *translateConceptResponse*.

```

<wsdl:portType name="MapperServerWebServiceImplPortType">
  <wsdl:operation name="translateConcept">
    <wsdl:input message="xsd:translateConceptRequest" wsaw:Action="urn:translateConcept" />
    <wsdl:output message="xsd:translateConceptResponse"
wsaw:Action="urn:translateConceptResponse" />
  </wsdl:operation>
</wsdl:portType>

```

Figura 4.6: Exemplo de definição do elemento *port type* do WSDL

- *binding* - especifica o protocolo de rede e o formato da mensagem a ser utilizada por um *port type* específico. Ele representa a ligação entre as descrições abstrata e concreta do WSDL. A Figura 4.7 ilustra a definição da associação do protocolo *Hypertext Transfer Protocol* (HTTP) ao *port type translateConcept*.

```

<wsdl:binding name="MapperServerWebServiceImplHttpBinding"
type="xsd:MapperServerWebServiceImplPortType">
  <http:binding verb="POST" />
  <wsdl:operation name="translateConcept">
    <http:operation location="MapperServerWebServiceImpl/translateConcept" />
    <wsdl:input>
      <mime:content type="text/xml" part="translateConcept" />
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml" part="translateConcept" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

Figura 4.7: Exemplo de definição do elemento *binding* do WSDL

- *service* - especifica o endereço *Uniform Resource Locator* (URL) onde o serviço está disponível. No exemplo da Figura 4.8, o serviço está disponível em *http://localhost:8080/axis2/services/MapperServerWebServiceImpl*.

```

<wsdl:service name="MapperServerWebServiceImpl">
  <wsdl:port name="MapperServerWebServiceImplHttpEndpoint"
binding="xsd:MapperServerWebServiceImplHttpBinding">
    <http:address location="http://localhost:8080/axis2/services/MapperServerWebServiceImpl" />
  </wsdl:port>
</wsdl:service>

```

Figura 4.8: Exemplo de definição do elemento *service* do WSDL

4.3.4 UDDI

UDDI é uma especificação técnica utilizada para facilitar a publicação e descoberta de serviços e *Provedores de Serviços*. No contexto do Web Services, o próprio UDDI também é um serviço.

O UDDI é descrito através de duas especificações: *UDDI Schema* e *UDDI Application Programming Interface (API)*. O *UDDI Schema* determina os tipos de estruturas de dados XML que compõem um registro de serviço como o nome do serviço, uma breve descrição do que ele faz, o endereço onde o serviço está disponível e a descrição das interfaces para acesso do serviço. Já o *UDDI API* determina as mensagens SOAP usadas para publicar ou descobrir um serviço no diretório de serviços.

Os *Provedores de Serviços* utilizam o UDDI para publicar seus serviços e torná-los disponíveis aos *Consumidores de Serviços*. Os *Consumidores de Serviços* por sua vez utilizam o UDDI para descobrir os serviços que lhes interessam e obter as descrições necessárias para utilizá-los.

4.4 Considerações Finais

Neste capítulo foram apresentadas de forma resumida os principais conceitos relativos ao modelo de arquitetura baseada em serviços.

SOA, mais especificamente Web Services, é utilizada na concepção da arquitetura SOARI que é apresentada mais detalhadamente no próximo capítulo. Tal uso está relacionado com a abordagem proposta por Dameron et al. (2004) de prover todas as funções relacionadas a ontologia através de Web Services. Esta abordagem visa possibilitar que aplicações da Web Semântica acessem estes serviços de forma idêntica à forma como acessam outros serviços, além de potencializar o uso de funções ontológicas para um maior número de aplicações.

Além dos ganhos identificados em (DAMERON et al., 2004), outros são obtidos com o uso de Web Services. Estes ganhos advêm principalmente das características de fraco acoplamento, dinamismo, e interoperabilidade da SOA, apresentados no próximo capítulo.

5 SOARI

O objetivo deste capítulo é descrever detalhadamente a arquitetura SOARI (*Service Oriented Architecture for Reputation Interaction*), que tem por finalidade possibilitar a interoperabilidade entre agentes que utilizam modelos de reputação heterogêneos. A seção 5.1 apresenta uma arquitetura genérica de agente que implementa um modelo de reputação. Na seção 5.2, a arquitetura usada como base para o desenvolvimento da SOARI é descrita. A arquitetura SOARI e seus módulos são detalhados na seção 5.3. Em seguida, a seção 5.4 apresenta as fases de funcionamento da arquitetura apresentada. Na seção 5.5, são descritos os requisitos necessários para a integração desta arquitetura a sistemas legados. Por fim, a seção 5.6 apresenta algumas considerações finais sobre a arquitetura.

5.1 Introdução

A necessidade por controle social em SMA trouxe a necessidade de interoperabilidade entre agentes que utilizam modelos de reputação. Uma arquitetura genérica de agente, que utiliza modelos de reputação, é ilustrada na Figura 5.1, sendo os retângulos a representação dos módulos de controle e os ovais a representação dos repositórios de dados.

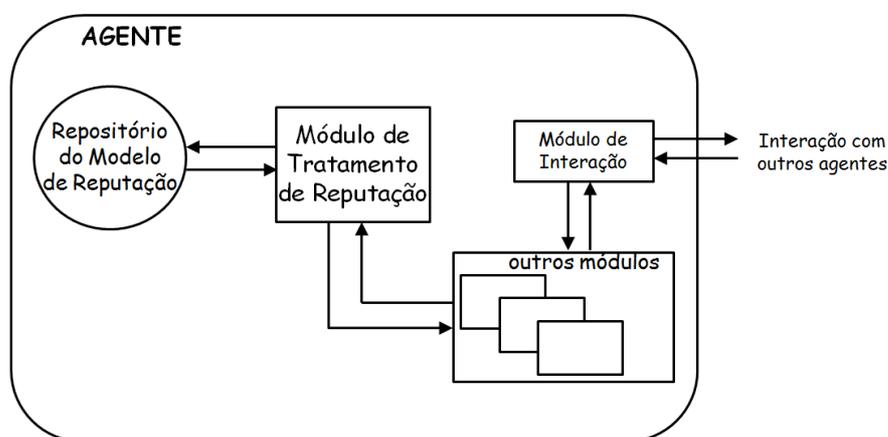


Figura 5.1: Arquitetura genérica de agentes com modelo de reputação

Os agentes que implementam a arquitetura são compostos por diversos módulos, dentre os quais o **Módulo de Interação** e o **Módulo de Tratamento de Reputação** (MTR). O **Módulo de Interação** (MI) é responsável pelo recebimento e envio de mensagens, bem como pelo redirecionamento das mensagens recebidas para os outros módulos internos do agente. O MTR é responsável por processar as mensagens sobre reputação, calcular e atualizar internamente o valor da reputação dos agentes, e solicitar e responder a consultas sobre reputação. Sendo assim, quando o MI recebe uma mensagem relacionada à reputação, ele a redireciona diretamente ao MTR.

Porém, em decorrência da heterogeneidade dos modelos de reputação existentes em SMA, a interoperabilidade entre todos os agentes que utilizam esta arquitetura é impossibilitada. Esta impossibilidade é resultado da incapacidade destes agentes de compreenderem os conceitos de outros modelos de reputação, além do seu próprio.

5.2 Arquitetura Genérica de Agente para Interação sobre Reputação

Para suprir a necessidade por uma arquitetura que viabilize a interoperabilidade entre agentes heterogêneos, Vercouter et al. (2007) propuseram uma arquitetura genérica para a interação de reputação baseada em ontologias (Figura 5.2).

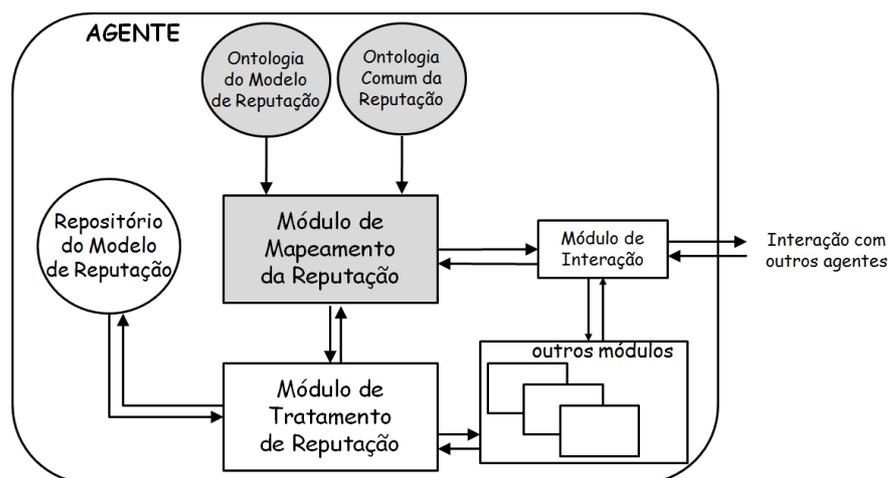


Figura 5.2: Arquitetura genérica de agentes para a interação de reputação (VERCOUTER et al., 2007)

Esta arquitetura, que estende a arquitetura genérica proposta na Figura 5.1, adota a abordagem híbrida de ontologias, conforme descrito na seção 3.4. Com a

adoção desta abordagem, os agentes necessitam desempenhar duas funções inter-relacionadas, porém distintas: *mapeamento* e *tradução*. A definição e distinção existente entre estes conceitos está descrita na seção 3.2.4. Para tanto, são adicionados três novos módulos a arquitetura:

Módulo de Mapeamento de Reputação (MMR) é o componente central desta arquitetura. Ele é responsável por desempenhar as funções de *mapeamento* e *tradução* dos conceitos da ontologia do modelo de reputação interna do agente para a ontologia comum da reputação, e vice-versa;

Ontologia do Modelo de Reputação armazena a representação da ontologia do modelo de reputação usada internamente pelo agente;

Ontologia Comum da Reputação armazena a representação da ontologia comum no domínio da reputação usada como interlíngua na comunicação com outros agentes do sistema.

A inclusão destes três novos módulos altera a dinâmica do fluxo da informação interna do agente. Esta mudança pode ser demonstrada através da apresentação de dois cenários.

No primeiro cenário, o funcionamento da arquitetura é demonstrado a partir do momento em que o agente recebe uma mensagem sobre reputação. Neste caso, a seqüência de tarefas executada internamente pelo agente é:

1. Uma mensagem é recebida pelo MI e seu cabeçalho é verificado. Caso o agente identifique que a mensagem está relacionada à reputação e expressa usando conceitos da **Ontologia Comum da Reputação**, seu conteúdo é redirecionado para o MMR. Caso contrário, ele é redirecionado para algum outro módulo interno do agente;
2. O MMR ao receber a mensagem, identifica os conceitos da **Ontologia Comum da Reputação** e os traduz para os conceitos correspondentes da **Ontologia do Modelo de Reputação**;
3. O próprio MMR, de posse da mensagem traduzida, aciona o MTR para o processamento da mensagem.

No segundo cenário, o funcionamento da arquitetura é demonstrado quando o agente requisita ou transmite informação sobre reputação a outro agente. Neste caso, a seqüência de tarefas executada internamente pelo agente é:

1. O MTR é acionado para elaborar uma mensagem utilizando os conceitos do seu modelo de reputação. Finalizada sua elaboração, a mensagem é encaminhada ao MMR;
2. O MMR identifica os conceitos da **Ontologia do Modelo de Reputação** na mensagem e os traduz para os conceitos correspondentes da **Ontologia Comum da Reputação**;
3. A mensagem é então encaminhada pelo MMR ao MI, que a transmite para seu(s) destinatário(s).

Embora tal arquitetura possa ser considerada como uma boa alternativa para resolver o problema proposto neste trabalho, ela nunca foi implementada. Além disso, ela possui alguns inconvenientes do ponto de vista do agente. Primeiro, ela combina em um único módulo as funções de *mapeamento* e *tradução*, não apresentando uma distinção clara entre estas duas funções. Segundo, requer que todos os agentes possuam internamente representados, em termos ontológicos, seu modelo de reputação e a ontologia comum no domínio da reputação. Finalmente, todos os agentes têm que executar internamente a função de *mapeamento*, função esta que consome muitos recursos computacionais.

Estas são premissas extremamente importantes, já que afetam consideravelmente o desempenho dos agentes. Desta forma, propõe-se na próxima seção uma possível solução para tais inconvenientes, usando para isto a abordagem SOA.

5.3 Arquitetura Orientada a Serviços para Interação sobre Reputação

A principal idéia da arquitetura SOARI (Figura 5.3) é desempenhar as funções de *mapeamento* e *tradução* externamente ao agente, disponibilizando os resultados aos agentes através de um serviço de consulta *on-line*.

A arquitetura SOARI estende a arquitetura proposta por Vercoouter et al. (2007) de duas maneiras distintas:

- subdivide o MMR em dois módulos distintos e especializados, denominados Serviço de Mapeamento de Ontologias (SMO) e TRADUTOR.
- executa as funções de *mapeamento* e *tradução*, através do módulo SMO, externamente ao agente.

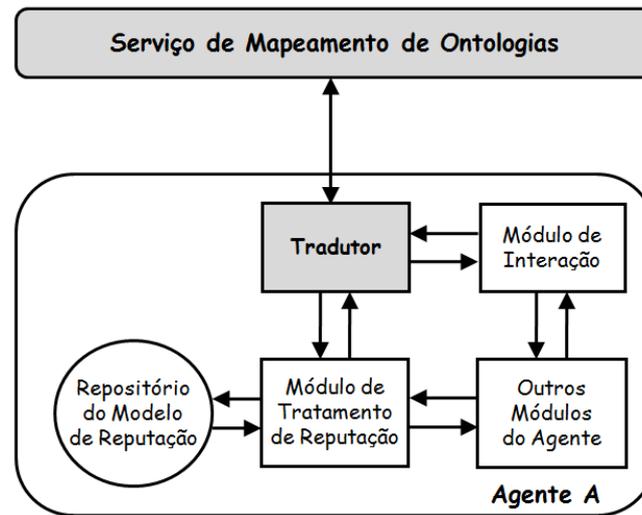


Figura 5.3: Arquitetura Orientada a Serviços para Interação de Reputação

Analisando através da perspectiva de uma abordagem SOA, o SMO é considerado o **Provedor de Serviços** e o módulo **TRADUTOR** é o **Consumidor de Serviços**. Nesta arquitetura não é considerado o **Registro de Serviços**, no entanto, sua inclusão é bastante simples já que refere-se somente à localização dinâmica do módulo SMO pelo módulo **TRADUTOR**.

Segundo a perspectiva de projeto e programação, a vantagem de se utilizar a abordagem SOA é a simplificação dos agentes, já que estes não têm que executar as funções de *mapeamento* e *tradução* internamente, o que reduz o consumo de recursos pelos agentes. Outra vantagem desta abordagem é a possibilidade de reutilização dos resultados gerados por novos agentes, desde que a ontologia do modelo de reputação destes agentes já tenha sido processada pelo SMO.

O SMO é um serviço disponibilizado externamente ao agente. Ele implementa as funções de *mapeamento* e *tradução*, o que se reflete nas suas duas principais atividades:

- identificar o mapeamento existente entre os conceitos de uma ontologia de modelo de reputação e os conceitos de uma ontologia comum no domínio da reputação;
- traduzir conceitos quando solicitado pelo módulo **TRADUTOR**.

O módulo **TRADUTOR** reside internamente nos agentes, e utiliza o resultado das funções desempenhadas pelo SMO para realizar suas diferentes atividades:

- substituir os conceitos da ontologia comum no domínio da reputação pelos

conceitos da ontologia do modelo de reputação do agente nas mensagens recebidas pelo agente;

- substituir os conceitos da ontologia do modelo de reputação do agente pelos conceitos da ontologia comum no domínio da reputação nas mensagens enviadas pelo agente;
- interpretar mensagens de reputação descritas em termos da ontologia de seu modelo de reputação, e acionar as funções correspondentes no MTR;
- criar uma mensagem descrita em termos da ontologia de seu modelo de reputação quando requisitado pelo MTR.

Nas próximas seções, os módulos SMO e TRADUTOR são descritos mais detalhadamente.

5.3.1 Serviço de Mapeamento de Ontologias

O módulo Serviço de Mapeamento de Ontologias (SMO), ilustrado na Figura 5.4, é o principal módulo da arquitetura SOARI. Ele é um serviço externo ao agente implementado como um Web Service (BOOTH et al., 2004) e composto por seis componentes descritos a seguir.

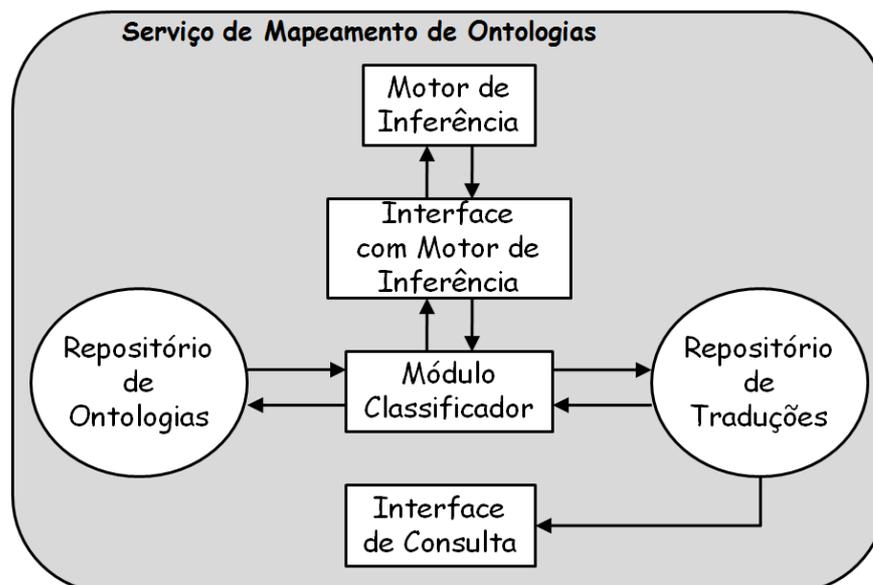


Figura 5.4: Componentes do Serviço de Mapeamento de Ontologias

5.3.1.1 Repositório de Ontologias

Consiste em um depósito de dados de ontologias. As ontologias armazenadas neste componente devem ser o resultado do alinhamento entre uma *Ontologia Comum no Domínio da Reputação* e uma *Ontologia de Modelo de Reputação*. Neste repositório, as ontologias são armazenadas como arquivos em um diretório do sistema e são descritas em linguagem OWL DL. Por exemplo, um arquivo contendo o alinhamento entre as ontologias Repage e FORe conterá ambas as ontologias, bem como um conjunto de relações de alinhamento, tal como mostrado na Tabela 6.5.

5.3.1.2 Repositório de Traduções

Este componente é o depósito de dados onde os resultados dos mapeamentos realizados pelo componente *Módulo Classificador* são armazenados. Sua estrutura é composta por três tabelas em um banco de dados relacional, denominadas *ONTOLOGY*, *ONTRELATION* e *ONTTRANSLATION*.

A tabela *ONTOLOGY* contém o registro de todas as ontologias processadas pelo SMO e sua estrutura de dados é descrita na Tabela 5.1.

Tabela 5.1: Estrutura de dados da tabela ONTOLOGY

Campo	Tipo	Descrição
id	inteiro	Identificador da ontologia
onttype	inteiro	Identifica o tipo de ontologia: 1 - Ontologia Comum no Domínio da Reputação 2 - Ontologia de Modelo de Reputação
onturi	texto	Nome da ontologia
ontversion	inteiro	Versão da ontologia
ontdescription	texto	Descritivo da ontologia

A tabela *ONTRELATION* contém o relacionamento entre uma *Ontologia Comum no Domínio da Reputação* e uma *Ontologia de Modelo de Reputação*, estando sua estrutura de dados descrita na Tabela 5.2.

A tabela *ONTTRANSLATION* contém o mapeamento entre os conceitos de uma *Ontologia Comum no Domínio da Reputação* e uma *Ontologia de Modelo de Reputação*. A Tabela 5.3 descreve sua estrutura de dados.

Tabela 5.2: Estrutura de dados da tabela ONTRELATION

Campo	Tipo	Descrição
id	inteiro	Identificador da relação entre duas ontologias
ontinterchange	inteiro	Identificador da ontologia do tipo <i>Ontologia Comum no Domínio da Reputação</i> armazenada na tabela <i>ONTOLOGY</i>
ontnative	inteiro	Identificador da ontologia do tipo <i>Ontologia de Modelo de Reputação</i> armazenada na tabela <i>ONTOLOGY</i>
ontfilename	texto	Caminho completo do arquivo que contém a ontologia alinhada entre a <i>ontinterchange</i> e <i>ontnative</i>

Tabela 5.3: Estrutura de dados da tabela ONTTRANSLATION

Campo	Tipo	Descrição
id	inteiro	Identificador do mapeamento entre os conceitos de duas ontologias
ontrelid	inteiro	Identificador da relação entre duas ontologias na tabela <i>ONTRELATION</i>
fromconcept	texto	Conceito da <i>Ontologia Comum no Domínio da Reputação</i>
toconcept	texto	Conceito da <i>Ontologia de Modelo de Reputação</i>

5.3.1.3 Interface com Motor de Inferência

Consiste em uma interface que especifica as funções mínimas a serem desempenhadas pelo componente *Motor de Inferência*, a fim de atender as necessidades do componente *Módulo Classificador*. Por ser uma interface, ele habilita a substituição do componente *Motor de Inferência*, caso haja necessidade, sem impactar o funcionamento do restante do módulo.

5.3.1.4 Motor de Inferência

Este componente é um motor de inferência de ontologias OWL DL e deve ter como uma de suas funcionalidades a classificação de ontologias. Classificação é a função que determina a relação hierárquica para todas as classes existentes em uma ontologia, com a finalidade de criar uma hierarquia completa dos conceitos desta ontologia (SIRIN et al., 2007).

5.3.1.5 Módulo Classificador

Este é o componente principal do SMO. Ele é responsável por identificar o mapeamento implícito existente nas ontologias alinhadas armazenadas no *Repositório de Ontologia*. O mapeamento identificado por este módulo é inferido pela simples

classificação da *Ontologia do Modelo de Reputação* alinhada a *Ontologia Comum no Domínio da Reputação*. Para tanto, ele utiliza o componente *Motor de Inferência* através do componente *Interface com Motor de Inferência* para realizar a classificação das ontologias e funciona conforme o Pseudo-Algoritmo 1.

Pseudo-Algoritmo 1 Módulo Classificador

```

Ontology ← readOntologyRepository()
classifyOntology(Ontology)
allAsserted ← getAllAssertedConcepts(Ontology)
for all asserted of allAsserted do
    allInferred ← getAllInferredConcepts(asserted)
    for all inferred of allInferred do
        writeTranslationRepository(asserted, inferred)
    end for
end for

```

5.3.1.6 Interface de Consulta

Este é o componente que responde as solicitações feitas pelos agentes ao SMO. As funções disponibilizadas por este componente são as seguintes:

addOntologyTranslation - Identifica o mapeamento existente entre duas ontologias alinhadas e armazena o resultado no *Repositório de Tradução*.

getOWLFile - Retorna o arquivo de uma ontologia alinhada armazenado no *Repositório de Tradução*.

isOntologyTranslation - Retorna verdadeiro, caso exista no *Repositório de Tradução* o mapeamento entre uma *Ontologia Comum no Domínio de Reputação* e uma *Ontologia de Modelo de Reputação*, ou falso, em caso contrário.

removeOntologyTranslation - Apaga do *Repositório de Tradução* o mapeamento entre uma *Ontologia Comum no Domínio de Reputação* e uma *Ontologia de Modelo de Reputação*.

translateConcept - Retorna a lista de conceitos resultado da tradução de um conceito entre uma ontologia de origem e uma ontologia de destino armazenados no *Repositório de Tradução*.

5.3.2 Tradutor

O módulo TRADUTOR, ilustrado na Figura 5.5, utiliza os mapeamentos armazenados no SMO para substituir os conceitos da *Ontologia Comum no Domínio*

da Reputação pelos conceitos da *Ontologia do Modelo de Reputação* interna do agente em mensagens sobre reputação e vice-versa. Este módulo, conforme apresentado na Figura 5.3, está fortemente integrado a outros módulos do agente, mais especificamente ao *Módulo de Interação* (MI) e ao *Módulo de Tratamento de Reputação* (MTR).

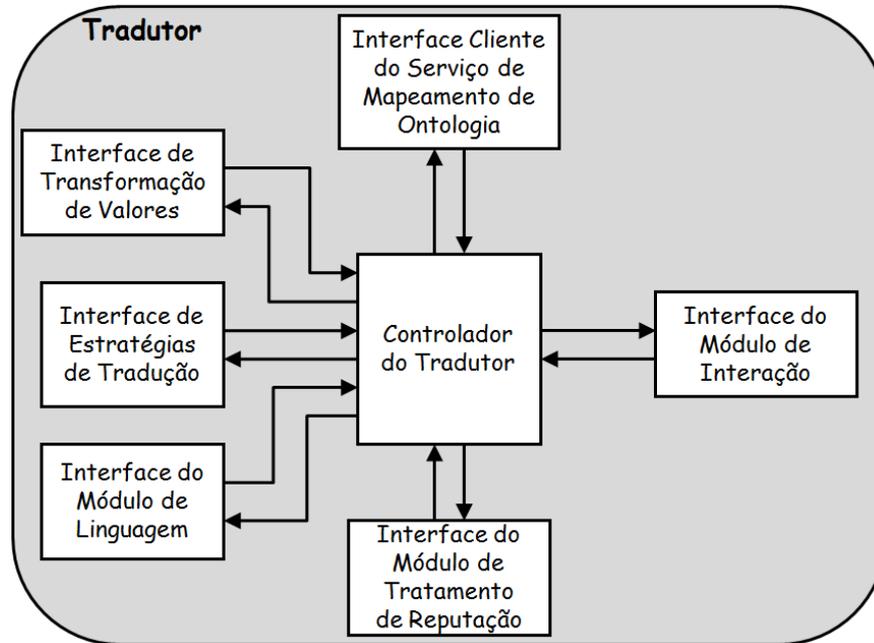


Figura 5.5: Componentes do TRADUTOR

Dentre as funcionalidades desempenhadas por este módulo, uma das mais importantes é o tratamento de *Problemas de Tradução*. Entende-se por *Problemas de Tradução* qualquer tradução de conceito de uma ontologia de origem para uma ontologia de destino cujo mapeamento não seja um para um. Exemplos de *Problemas de Tradução* são *incompletude*, quando um conceito de uma ontologia de origem não é mapeado para nenhum conceito na ontologia de destino, e *ambiguidade*, quando um conceito de uma ontologia de origem é mapeado para mais de um conceito da ontologia de destino.

Flexibilidade e configurabilidade foram identificados como sendo as duas principais características para o projeto deste módulo, já que os agentes não implementam os mesmos MI e MTR, bem como utilizam diferentes estratégias de tradução. A flexibilidade é a característica que possibilita que o módulo TRADUTOR se adapte para integrar-se aos módulos MI e MTR do agente. Já a característica de configurabilidade possibilita a seleção de estratégias de tradução ou funções de transformação de valores diferentes para cada agente.

Para satisfazer estas duas características, o módulo TRADUTOR possui um

componente central, denominado *Controlador do Tradutor* (CT) e seis outros componentes que desempenham funções específicas.

O componente CT é responsável por controlar a comunicação e o sequenciamento de ativação dos outros seis componentes do módulo durante o recebimento e envio de mensagens pelo agente.

Todos os seis componentes possuem funções específicas no módulo TRADUTOR; no entanto, são especificações de interface das funções que devem desempenhar, o que possibilita que sua implementação seja diferente para cada agente.

Os principais componentes deste módulo são descritos a seguir.

5.3.2.1 Interface do Módulo de Interação (IMI)

Consiste na especificação de uma interface com a função de realizar a comunicação entre o MI e o módulo TRADUTOR. Este componente é composto por dois métodos principais: *inMessage* e *outMessage*.

O método *inMessage* deve ser acionado pelo MI do agente quando ele necessita redirecionar uma mensagem sobre reputação para o módulo TRADUTOR.

O método *outMessage* é acionado pelo módulo TRADUTOR quando este necessita enviar uma mensagem sobre reputação para outros agentes. Ele consiste somente em uma especificação de interface, já que sua implementação está intimamente relacionada com o MI do agente e este último varia de acordo com o ambiente de comunicação do sistema.

5.3.2.2 Interface Cliente do Serviço de Mapeamento de Ontologias (ICSMO)

Consiste na especificação de uma interface com a função de realizar a comunicação entre o módulo TRADUTOR e o SMO. Como o SMO está implementado como um Web Service, esta interface está implementada como um cliente de Web Service. Este componente possui os mesmos métodos disponibilizados pelo componente *Interface de Consulta* do SMO, no entanto, implementados do ponto de vista do solicitante da função. Portanto, ele é composto por cinco métodos principais: *addOntologyTranslation*, *getOWLFile*, *isOntologyTranslation*, *removeOntologyTranslation* e *translateConcept*, descritos na seção 5.3.1.

O método mais importante dentre os citados é o *translateConcept*, pois é ele que solicita ao SMO as possíveis traduções de um conceito de uma ontologia de

origem para uma ontologia de destino.

5.3.2.3 Interface de Transformação de Valores (ITV)

Consiste em uma interface com a função de transformar a representação do valor de reputação do modelo de reputação do agente para a representação usada pela SOARI e vice-versa. Segundo Pinyol, Sabater-Mir e Cuní (2007) a representação do valor da reputação é um dos aspectos mais importantes dos modelos de reputação. Eles identificam quatro tipos de representação:

- *Booleana* (BO) - a reputação pode assumir um de dois valores, por exemplo, Bom ou Ruim, Verdadeiro ou Falso;
- *Númérico Delimitado* (ND) - a reputação é um valor entre um intervalo numérico delimitado, por exemplo, $[0, 1]$ delimitado entre 0 e 1;
- *Conjuntos Discretos* (CD) - a reputação pode assumir um dos valores de um conjunto de possibilidades, por exemplo, {Muito Ruim, Ruim, Neutro, Bom, Muito Bom};
- *Distribuição Probabilística* (DP) - Esta representação aplica uma distribuição de probabilidades em um conjunto discreto ordenado. Por exemplo, considere um conjunto ordenado $L = \{\text{Muito Ruim, Ruim, Neutro, Bom, Muito Bom}\}$ onde L_1 é Muito Bom, L_2 é Bom e assim sucessivamente. Se X for a distribuição probabilística sobre L , então, X_i é definida como a probabilidade da avaliação L_i . Dado que X é uma distribuição probabilística, tem-se que $\sum_{i=1..5} X_i = 1$.

A representação do valor da reputação adotada pela arquitetura SOARI é a representação do tipo *Distribuição Probabilística*, a mesma adotada pelo modelo de reputação Repage. Esta representação é adotada pois ela consegue lidar com a falta de precisão inerente à transmissão de uma avaliação subjetiva (AVEGLIANO, 2008). Este componente é composto por dois métodos: *incomingTransformation* e *outgoingTransformation*.

O método *incomingTransformation* é executado quando o CT identifica algum valor na mensagem de reputação recebida do MI do agente. Ele transforma este valor da representação da SOARI para a representação do modelo de reputação usado internamente pelo agente.

O método *outgoingTransformation* é executado quando o CT identifica algum valor na mensagem de reputação a ser enviada para o MI do agente. Ele trans-

forma este valor da representação do modelo de reputação usado internamente pelo agente para a representação da SOARI.

5.3.2.4 Interface de Estratégias de Tradução (IET)

Este componente é a especificação de uma interface com a função de tratar possíveis *Problemas de Tradução* que possam ocorrer. Os dois *Problemas de Tradução* identificados são *ambiguidade* e *incompletude*. Algumas das estratégias para tratar estes problemas são (i) remover o conceito da mensagem, (ii) não substituí-lo na mensagem mantendo o conceito original, ou (iii) utilizar algum tipo de heurística na determinação do conceito mais adequado para substituição na mensagem.

Apesar destas estratégias resolverem alguns *Problemas de Tradução*, elas podem ser as geradoras de novos problemas. Para prevenir o surgimento destes novos problemas, estas estratégias devem ser logicamente equivalentes ($A \Leftrightarrow B$): assim se o conceito A é substituído pelo conceito B, quando o inverso ocorrer, é obrigatório que o conceito B seja substituído pelo conceito A.

5.3.2.5 Interface do Módulo de Linguagem (IML)

Consiste em um componente que tem as funções de realizar as análises léxica e sintática das mensagens sobre reputação, identificar nelas os conceitos relacionados a uma ontologia de reputação, e montar mensagens sobre reputação descrevendo-as em uma linguagem de consulta.

Este componente é composto de duas interfaces, sendo uma representando as funções de *parsing* relacionado a linguagem de consulta (*LanguageInterface*) e outra que consiste na representação do conteúdo da mensagem e em um conjunto de funções para manipulá-lo (*ObjectInterface*).

5.3.2.6 Interface do Módulo de Tratamento de Reputação (IMTR)

Este componente é a especificação de uma interface com a função de interpretar mensagens expressas em linguagem de consulta e acionar operações no MTR do agente baseado na interpretação. Portanto, para cada combinação de linguagem de consulta com modelo de reputação de agente deve existir uma implementação desta especificação.

A próxima seção descreve as fases e a dinâmica de funcionamento da arquitetura SOARI.

5.4 Fases de Funcionamento

Na seção 5.3, a estrutura estática da arquitetura SOARI foi apresentada. Nesta seção, esta mesma arquitetura é apresentada do ponto de vista funcional e dinâmico, sendo seu funcionamento dividido em duas fases: *Inicialização* e *Operação*.

5.4.1 Inicialização

A fase de *Inicialização* corresponde às atividades que devem ser executadas para habilitar o SMO a receber e responder solicitações do módulo TRADUTOR de um agente para um modelo de reputação ainda desconhecido. A seqüência de atividades necessárias nesta fase são:

1. Criar a *Ontologia do Modelo de Reputação* do novo agente, já que normalmente os modelos de reputação não estão descritos em termos ontológicos e o SMO necessita da ontologia do modelo de reputação para operar;
2. Alinhar a *Ontologia do Modelo de Reputação* à *Ontologia Comum no Domínio da Reputação*, já que o SMO processa somente ontologias que estão descritas em termos de uma ontologia comum de domínio (abordagem híbrida);
3. Gerar o mapeamento entre os conceitos da *Ontologia do Modelo de Reputação* e a *Ontologia Comum no Domínio da Reputação*.

A seguir tais atividades são descritas em maiores detalhes.

5.4.1.1 Criar a ontologia do modelo de reputação

Diversas metodologias foram propostas para a construção de ontologias, dentre elas as metodologias de Uschold et al. (1996) e a *Methontology* proposta por Fernández-Lopes, Gómez-Pérez e Jurino (1997). Análises comparativas entre estas metodologias são apresentadas em (CORCHO; FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2003) e (SILVA; SOUZA; ALMEIDA, 2008).

Apesar da existência de diversas metodologias para a construção de ontologias, nenhuma delas é especificamente utilizada neste trabalho. Tais metodologias abrangem etapas não somente de construção das ontologias, mas de todo o seu ciclo de vida, o que não é requerido para atender às necessidades deste trabalho. Sendo assim, utiliza-se um método mais simples, que se subdivide em duas etapas para a construção das ontologias dos modelos de reputação:

1. **Identificar os conceitos do modelo de reputação** - Esta etapa consiste no estudo do modelo de reputação e identificação dos principais conceitos relacionados à reputação. A realização desta etapa ocorre de forma manual, é normalmente desempenhada pelo projetista do sistema e de difícil automatização;
2. **Construir a ontologia do modelo de reputação em OWL DL** - Esta etapa consiste na representação dos conceitos identificados na primeira etapa na linguagem OWL DL. Sua realização também ocorre de forma manual e é desempenhada por um especialista na linguagem OWL DL.

5.4.1.2 Alinhar a ontologia do modelo de reputação

Alinhamento, conforme definido na seção 3.2.4, é o estabelecimento de relações binárias entre conceitos de duas ontologias. Portanto, nesta atividade a finalidade é estabelecer as relações binárias entre cada um dos conceitos gerados na etapa anterior da *Ontologia do Modelo de Reputação* e os conceitos da *Ontologia Comum no Domínio da Reputação*.

Conforme apresentado na seção 3.2.4, esta atividade pode ser realizada de forma *manual*, *semi-automática* ou *automática*. Neste trabalho, o alinhamento entre a *Ontologia do Modelo de Reputação* e a *Ontologia Comum no Domínio da Reputação* é realizada de forma manual, já que a automatização desta atividade não é o foco deste trabalho.

5.4.1.3 Gerar o mapeamento

Esta atividade tem por finalidade gerar o mapeamento entre os conceitos da *Ontologia do Modelo de Reputação* e a *Ontologia Comum no Domínio da Reputação* com base no alinhamento previamente realizado. Este mapeamento é realizado pelo SMO de forma automatizada, conforme o diagrama de seqüência *Unified Modelling Language* (UML) ilustrado na Figura 5.6, que representa a ordem de acionamento dos módulos do SMO para a descoberta do mapeamento.

Inicialmente, o agente, através do módulo ICSMO, realiza uma chamada ao método *addOntologyTranslation* do componente *Interface de Consulta* do SMO, passando como parâmetros o nome e versão da *Ontologia do Modelo de Reputação* e da *Ontologia Comum no Domínio da Reputação*, bem como o arquivo contendo o alinhamento destas duas ontologias. O SMO verifica se o mapeamento entre estas duas ontologias existe no *Repositório de Traduções*. Caso não exista, o

arquivo contendo o alinhamento das ontologias é armazenado no *Repositório de Ontologias*, e em seguida, o *Módulo Classificador* é acionado. Este último, lê a ontologia armazenada no *Repositório de Ontologias*, e aciona o componente *Interface com Motor de Inferência*, que repassa a ação para o *Motor de Inferência*. O *Motor de Inferência* classifica os conceitos das ontologias alinhadas e retorna o resultado da classificação para o *Módulo Classificador*. O *Módulo Classificador* realiza então o processamento dos resultados, conforme o pseudo-algoritmo 1 descrito na seção 5.3.1, a fim de identificar o mapeamento entre os conceitos das ontologias. Identificado o mapeamento, armazena-o no *Repositório de Traduções* para uso durante a fase de operação.

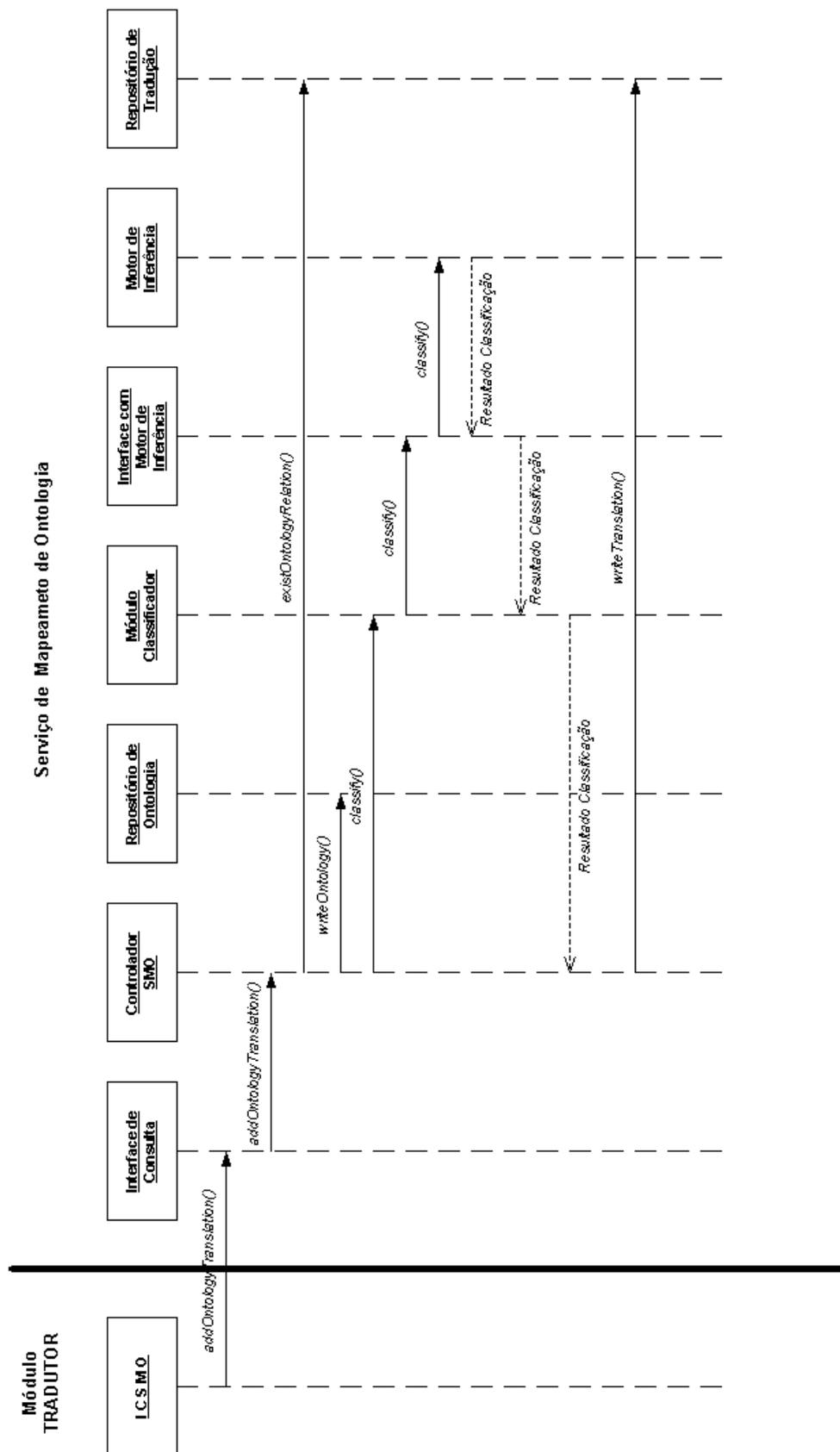


Figura 5.6: Diagrama de seqüência da fase de Inicialização do SMO

5.4.2 Operação

A fase de *Operação* consiste nas atividades executadas por um agente para possibilitar sua interoperabilidade com outros agentes sobre reputação. Esta fase é realizada obrigatoriamente em tempo de execução e a seqüência de acionamentos dos componentes está ilustrada na Figura 5.7.

A seqüência se inicia quando o agente recebe uma mensagem sobre reputação no seu MI. Este módulo analisa o cabeçalho desta mensagem, verifica se ela se refere ao domínio da reputação e está descrita usando conceitos da *Ontologia Comum no Domínio da Reputação*, e em caso afirmativo, encaminha-a ao componente IMI do módulo TRADUTOR. Este módulo encaminha a mensagem ao CT, que então aciona o componente IML para a identificação dos conceitos da *Ontologia Comum no Domínio da Reputação*. Uma vez identificados os conceitos, o CT aciona o ICSMO, que consulta o componente *Interface de Consulta* do SMO, usando uma chamada Web Service ao método *translateConcept*, para a tradução dos conceitos identificados. Nesta consulta é informado o conceito que se deseja traduzir e as ontologias *Ontologia Comum no Domínio da Reputação* e *Ontologia do Modelo de Reputação*. O componente *Interface de Consulta* recebe a solicitação e consulta no *Repositório de Traduções* todos os conceitos da *Ontologia do Modelo de Reputação* relacionados ao conceito encaminhado. O resultado é retornado ao ICSMO que o reencaminha então ao CT. O CT verifica o resultado recebido e caso identifique algum *Problema de Tradução* repassa os resultados para o componente IET. Este último resolve os *Problemas de Tradução* e reencaminha o resultado para o CT. Em seguida, o CT substitui os conceitos na mensagem por seus correspondentes traduzidos. Este mesmo componente verifica se há algum valor de reputação na mensagem que necessita de transformação. Em caso afirmativo, encaminha este valor ao componente ITV para efetuar-lá. Por fim, a mensagem é encaminhada ao componente IMTR para sua interpretação e o componente MTR é finalmente acionado.

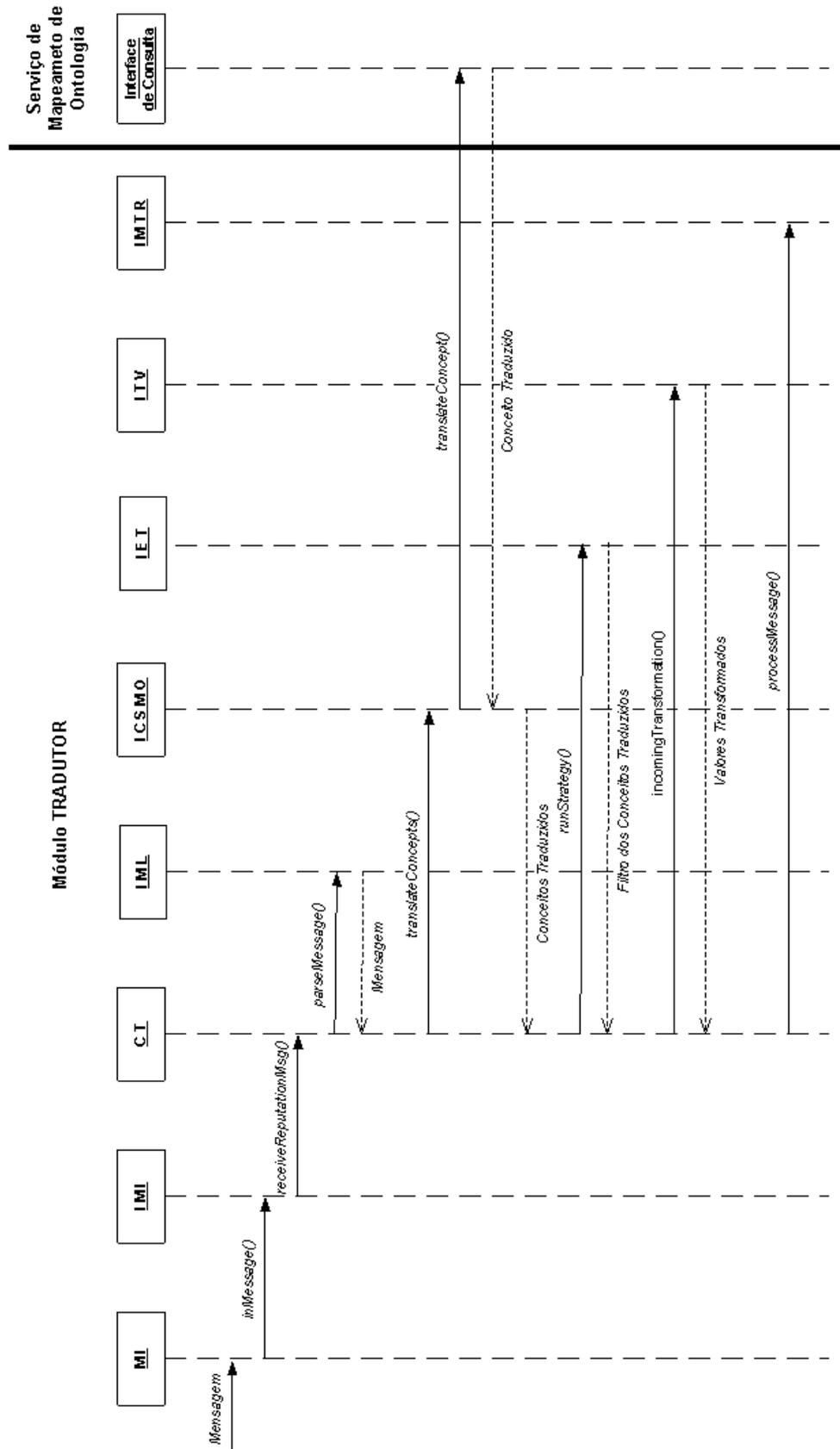


Figura 5.7: Diagrama de seqüência da fase de Operação do SMO

5.5 Integração a Sistemas Legados

Considerando a existência de um agente com a arquitetura genérica com modelo de reputação (Figura 5.1), a sua integração com a arquitetura SOARI (módulo TRADUTOR) requer algumas adaptações em alguns de seus módulos, mais especificamente MI e MTR. Entretanto, devido às características de flexibilidade e configurabilidade usadas como premissa durante o projeto do módulo TRADUTOR, estas adaptações são reduzidas ao mínimo. As adaptações nos módulos MI e MTR necessárias são:

- **Módulo de Interação** - Este módulo deve ser alterado para identificar mensagens sobre reputação e expressas em termos da ontologia definida como *Ontologia Comum no Domínio da Reputação*, extrair seu conteúdo, e enviá-lo para o CT através do IMI. Ele também deverá ser capaz de receber mensagens do CT através do IMI a fim de transmiti-las aos seus respectivos destinatários;
- **Módulo de Tratamento de Reputação** - A única adaptação a ser realizada neste módulo é fazer com que se registre no IMTR para que possa ser acionado por este último. Obrigatoriamente, o acionamento do *Módulo de Tratamento de Reputação* deve ser possível através de chamadas compatíveis com as disponíveis na linguagem de programação na qual a arquitetura está desenvolvida.

Pelas alterações necessárias, percebe-se que a integração da arquitetura SOARI a um agente já existente, apesar de intrusiva, é mínima. Esta facilidade de integração pode ser considerada como um ponto forte da arquitetura, já que os benefícios que ela pode trazer em comparação aos custos de alteração são significativos.

5.6 Considerações Finais

Este capítulo tratou dos aspectos estáticos e dinâmicos da arquitetura SOARI, desenvolvida segundo o modelo arquitetural SOA. Além das vantagens proporcionadas por esta arquitetura em comparação à arquitetura proposta por Vercoouter et al. (2007), o uso de SOA proporciona outras duas vantagens:

- Melhor adequação da arquitetura SOARI para ser utilizada em SMA, já que o componente SMO pode ser descoberto dinamicamente através do

uso de *Registro de Serviços*, facilitando assim a entrada de novos agentes no ambiente;

- Maior escalabilidade e disponibilidade da arquitetura, já que diversos componentes SMO podem ser mantidos ativos no ambiente a fim de dividir a carga de processamento das solicitações e servir como redundância um do outro.

No próximo capítulo detalhes da implementação e um exemplo de funcionamento da arquitetura SOARI num cenário de comércio eletrônico simplificado são apresentados.

6 Implementação e Funcionamento da Arquitetura

Neste capítulo, a arquitetura SOARI é implementada e integrada a agentes que utilizam modelos heterogêneos de reputação a fim de verificar e demonstrar o seu funcionamento. A seção 6.1 apresenta as premissas consideradas para a implementação da arquitetura SOARI. Em seguida, na seção 6.2 é apresentada uma breve descrição da implementação da arquitetura proposta no capítulo 5. A seção 6.3 apresenta um exemplo de funcionamento da arquitetura em um cenário de comércio eletrônico simplificado. Por fim, na seção 6.4 são apresentadas as considerações finais sobre o funcionamento da arquitetura.

6.1 Introdução

A implementação da arquitetura SOARI foi realizada segundo o paradigma de programação orientado a objetos e para isto usou-se a linguagem de programação Java. A escolha por esta linguagem de programação foi motivada por sua portabilidade e por ser utilizada na maioria dos trabalhos e experimentos realizados na área de SMA, principalmente em reputação. Devido a esta última característica, foi possível o reuso de diversos componentes, como a implementação do modelo Repage, o que agilizou seu desenvolvimento e realização de experimentos.

A seguir são descritos detalhes da implementação de cada um dos módulos da arquitetura SOARI.

6.2 Implementação

O ambiente de programação Eclipse versão 3.4.0¹ foi o ambiente de programação utilizado para a implementação da arquitetura SOARI.

6.2.1 Serviço de Mapeamento de Ontologias

O SMO foi implementado como um Web Service, conforme descrito na seção 5.4.1, utilizando a implementação SOAP denominada *Apache Axis versão 1.4*² e *Code Generator Wizard - eclipse Plug-in*³.

O *Repositório de Ontologias* foi implementado como um diretório onde os arquivos das ontologias alinhadas de modelos de reputação são armazenados.

No *Módulo Classificador* foi utilizado o *Protégé-OWL Plugin versão 3.4 beta* (HORRIDGE et al., 2004) para manipular os arquivos OWL das ontologias e facilitar a codificação do pseudo-código descrito na seção 5.3.1. Além disto, este mesmo plugin foi utilizado para realizar a integração com o *Motor de Inferência Pellet versão 1.5*⁴.

Os dados gerados pelo *Módulo Classificador* são armazenados no *Repositório de Traduções* implementado como um banco de dados relacional. O banco de dados utilizado foi o *Hyperthreaded Structured Query Language Database (HSQLDB) versão 1.8.0.8*⁵.

6.2.2 Tradutor

O módulo TRADUTOR foi codificado de forma a ter flexibilidade para simplificar sua integração a sistemas legados. Para tanto, todos os seus componentes foram especificados na arquitetura como interfaces, não havendo nenhuma referência à sua codificação. Assim, o módulo TRADUTOR pode facilmente se adaptar a qualquer ambiente ou características internas dos agentes.

A relação entre a arquitetura e a implementação é estabelecida via um arquivo de configuração descrito em XML e passado como parâmetro no momento da inicialização do CT. Os campos deste arquivo são descritos na Tabela 6.1, os quais devem seguir esta ordem no arquivo de configuração.

¹<http://www.eclipse.org>.

²<http://ws.apache.org/axis/>.

³<http://ws.apache.org/axis2/tools/>.

⁴<http://clarkparsia.com/pellet>.

⁵<http://hsqldb.org/>.

Tabela 6.1: Campos dos arquivos de configuração do CT

Campo	Descrição
id	Nome de identificação do agente
urlMapperServer	URL para acesso ao módulo SMO
ontNativeName	Nome da ontologia do modelo de reputação interno do agente
ontNativeVersion	Versão da ontologia do modelo de reputação interno do agente
ontInterchangeName	Nome da ontologia comum no domínio da reputação considerada pelo CT
ontInterchangeVersion	Versão da ontologia comum no domínio da reputação considerada pelo CT
mapperClientClass	Caminho completo da classe que implementa o componente <i>Interface Cliente do Serviço de Mapeamento de Ontologia</i>
internalLanguageName	Nome da linguagem de consulta considerada pelo CT
internalLanguageClass	Caminho completo da classe que implementa o componente <i>Interface do Módulo de Linguagem</i>
inputTranslationStrategyClass	Caminho completo da classe que implementa o componente <i>Interface de Estratégias de Tradução</i> quando a mensagem é recebida pelo CT
outputTranslationStrategyClass	Caminho completo da classe que implementa o componente <i>Interface de Estratégias de Tradução</i> quando a mensagem é gerada e enviada a partir do CT
valueTransformationConcept	Nome do conceito cujo valor deve ser transformado usando o componente <i>Interface de Transformação de Valores</i>
valueTransformationClass	Caminho completo da classe que implementa o componente <i>Interface de Transformação de Valores</i>
reputationReasonerClass	Nome completo da classe que implementa o componente <i>Interface do Módulo de Modelo de Reputação</i>

Na seqüência são apresentados detalhes de implementação adicionais de cada um dos componentes que integram o módulo TRADUTOR.

6.2.2.1 Interface do Módulo de Interação

Este componente foi implementado utilizando o *framework Java Agent DEvelopment Framework* (JADE)⁶ que é completamente desenvolvido em Java e que simplifica o desenvolvimento de SMA. Ele disponibiliza um *middleware* que atende à especificação da *Foundation for Intelligent Physical Agents* (FIPA)⁷ e uma interface gráfica que facilita a depuração de sistemas.

O método *outMessage* do componente IMI foi codificado para realizar o mapeamento dos tipos das mensagens sobre reputação para as performativas do linguagem de comunicação de agentes FIPA *Agent Communication Language* (ACL), conforme descrito na Tabela 6.2.

Tabela 6.2: Tipos de Mensagem SOARI x Performativa FIPA-ACL

Tipo de Mensagem SOARI	Performativa FIPA-ACL
REQUEST	REQUEST
INFORM	INFORM
RESULT	CONFIRM
FAULT	FAILURE

Este método também complementa o cabeçalho da mensagem FIPA-ACL com as informações sobre a linguagem interna do conteúdo da mensagem (*método setLanguage*) e a ontologia usada (*método setOntology*). Estas duas informações são essenciais para que o MI do agente destinatário da mensagem possa identificá-la como uma mensagem sobre reputação e redirecioná-la para o componente IMI do seu módulo TRADUTOR.

6.2.2.2 Interface de Transformação de Valores

Pinyol, Sabater-Mir e Cuní (2007) sugerem algumas regras de transformação de valores entre tipos de representação distintos. Como é adotada pela arquitetura SOARI a representação *DP* para a interoperabilidade de valores de reputação entre agentes, foram implementados dois componentes para transformação de valores: um para transformação da representação *Distribuição Probabilística* (*DP*)

⁶<http://jade.tilab.com/>.

⁷<http://www.fipa.org/>.

para *Numérico Delimitado* (*ND*) e outro para transformação reversa. Estas transformações aplicam-se, por exemplo, a agentes que usam o modelo de reputação L.I.A.R., que representa os valores da reputação como *ND*.

incomingTransformation - A regra desta transformação corresponde ao centro de massa do valor *DP* de entrada. Assim, dado um valor de entrada representado como *DP*, sua transformação para *ND* é dada pela equação

$$CM(X) = \frac{1}{10} \sum_{i=1}^5 (2i - 1)X_i \quad (6.1)$$

onde $X \in DP$ representa o valor de entrada representado como *DP* e $CM(X)$ representa o valor de centro de massa de X e corresponde ao valor de entrada representado como *ND*.

Por exemplo, a transformação do valor [0.0; 0.0; 0.4; 0.6; 0.0] da representação *DP* para *ND*, conforme a equação 6.1, é ilustrada a seguir.

$$\begin{aligned} ND &= CM(X) \\ ND &= \frac{1}{10} ((1 * 0.0) + (3 * 0.0) + (5 * 0.4) + (7 * 0.6) + (9 * 0.0)) \\ ND &= 0.62 \end{aligned}$$

outgoingTransformation - Este método implementa a regra inversa de transformação descrita em *incomingTransformation*, transformando valores representados em *ND* para *DP*. Esta transformação requer que o centro de massa gerado pelo valor *DP* resultante seja igual ao valor de entrada *ND*. Tal transformação é dada pelas equações que segue:

$$R'(X) = \begin{cases} 1 & \text{se } 0 \leq X \leq 0.2; \\ 2 & \text{se } 0.2 \leq X \leq 0.4 \\ 3 & \text{se } 0.4 \leq X \leq 0.6 \\ 4 & \text{se } 0.6 \leq X \leq 0.8 \\ 5 & \text{se } 0.8 \leq X \leq 1.0 \end{cases} \quad (6.2)$$

$$i_1 = \min\{R'(X), R'(max\{X - 0.1, 0\})\} \quad (6.3)$$

$$i_2 = \min\{5, i_1 + 1\} \quad (6.4)$$

$$p = 1 - \frac{X - CM(i_1)}{CM(i_2) - CM(i_1)} \quad (6.5)$$

onde X representa o valor de entrada $X \in ND$, $CM(i_1)$ e $CM(i_2)$ representam o valor dos centros de massa de i_1 e i_2 , que por sua vez representam

as posições dos elementos de DP , e p representa a probabilidade na posição i_1 .

Assim, as posições i_1 e i_2 de DP são preenchidas com os valores p e $(1 - p)$ respectivamente, enquanto as outras posições são preenchidas com valor 0.

A seguir é apresentada a sequência para realizar a transformação do valor 0,78 da representação ND para DP .

Primeiramente, calcula-se i_1 e i_2 conforme as equações 6.3 e 6.4, respectivamente.

$$i_1 = \min\{R'(0.78), R'(\max\{0.78 - 0.1, 0\})\}$$

$$i_1 = \min\{4, R'(0.68)\}$$

$$i_1 = \min\{4, 4\}$$

$$i_1 = 4$$

$$i_2 = \min\{5, i_1 + 1\}$$

$$i_2 = \min\{5, 4 + 1\}$$

$$i_2 = \min\{5, 5\}$$

$$i_2 = 5$$

Em seguida, é calculado p conforme a equação 6.5.

$$p = 1 - \frac{0.78 - CM(4)}{CM(5) - CM(4)}$$

$$p = 1 - \frac{0.78 - 0.7}{0.9 - 0.7}$$

$$p = 1 - \frac{0.08}{0.2}$$

$$p = 1 - 0.4$$

$$p = 0.6$$

Uma vez finalizado os cálculos, o valor representado como DP correspondente ao valor 0,78 representado como ND é apresentado a seguir.

$$DP = [0.0; 0.0; 0.0; p; (1 - p)]$$

$$DP = [0.0; 0.0; 0.0; 0.6; 0.4]$$

6.2.2.3 Interface de Estratégias de Tradução

A IET é acionada para a resolução de *Problemas de Tradução* como *incompletude* e *ambiguidade*. Na sua especificação, é definido que todas as estratégias recebem

como entrada o conceito da ontologia de origem a ser traduzido e a lista de possíveis traduções para a ontologia de destino, e devem retornar como saída a(s) tradução(ões) apropriada(s).

Algumas estratégias de tradução implementadas neste trabalho são as seguintes:

ZeroEmptyNAll - Se não houver nenhuma tradução possível, a estratégia retorna vazio. Em caso contrário, retorna todos os conceitos da ontologia de destino possíveis.

Por exemplo, suponha o uso desta estratégia para a tradução do conceito *Reputação Propagada* da FORe (ontologia de origem) para a ontologia do modelo de reputação Repage (ontologia de destino). A estratégia retornaria o resultado vazio, já que não existe um conceito correspondente na ontologia do Repage para o conceito *Reputação Propagada*. Porém, se utilizada para tradução da FORe (ontologia de origem) para a ontologia do modelo de reputação L.I.A.R. (ontologia de destino), a estratégia retornaria os três conceitos correspondentes na ontologia de destino: *Reputação baseada em Recomendações Observadas*, *Reputação baseada em Recomendações Avaliadas* e *Reputação baseada em Recomendação de Reputação*.

ZeroEmptyNFirst - Se não houver nenhuma tradução possível, a estratégia retorna vazio. Em caso contrário, retorna o primeiro conceito da ontologia de destino.

Desta forma, aplicando esta estratégia no exemplo descrito anteriormente, a estratégia retornaria o resultado vazio, na tradução do conceito *Reputação Propagada* da FORe para a ontologia do modelo de reputação Repage. Porém retornaria somente o conceito *Reputação baseada em Recomendações Observadas* se utilizada para tradução da FORe para a ontologia do modelo de reputação L.I.A.R..

ZeroInputNAll - Se não houver nenhuma tradução possível, a estratégia retorna o conceito de entrada. Em caso contrário, retorna todos os conceitos da ontologia de destino possíveis.

Portanto, aplicando esta estratégia no exemplo descrito inicialmente, a estratégia retornaria o resultado *Reputação Propagada*, na tradução do conceito *Reputação Propagada* da FORe para a ontologia do modelo de reputação Repage. Porém, se utilizada para tradução da FORe para a ontologia do modelo de reputação L.I.A.R., a estratégia retornaria os três

conceitos correspondentes na ontologia de destino: *Reputação baseada em Recomendações Observadas*, *Reputação baseada em Recomendações Avaliadas* e *Reputação baseada em Recomendação de Reputação*.

ZeroInputNFirst - Se não houver nenhuma tradução possível, a estratégia retorna o conceito de entrada. Em caso contrário, retorna o primeiro conceito da ontologia de destino.

Se aplicá-la ao exemplo, a estratégia retornaria o resultado *Reputação Propagada*, na tradução do conceito *Reputação Propagada* da FORe para a ontologia do modelo de reputação Repage e retornaria o conceito *Reputação baseada em Recomendações Observadas* na tradução da FORe (ontologia de origem) para a ontologia do modelo de reputação L.I.A.R..

HeuristicStrategy - Esta estratégia de tradução é mais complexa, pois ela determina heurísticamente uma possível tradução para os conceitos que não possuem uma tradução direta. Tal heurística utiliza a ontologia alinhada para seu processamento, e caminha na estrutura hierárquica formada em busca de alguma possível tradução. Primeiramente, caminha, a partir do conceito indicado, para seus subconceitos em busca de alguma correspondência entre os conceitos. Caso não encontre nenhuma correspondência, realiza o mesmo processo, porém caminhando para seus superconceitos da hierarquia.

Por exemplo, a tradução do conceito *Reputação Propagada* da FORe para a ontologia do modelo de reputação Repage retornaria como resultado o conceito *Reputação*. Como não existe um conceito correspondente para tradução direta, esta estratégia realiza primeiramente uma busca por subconceitos de *Reputação Propagada* da FORe, que possuam correspondência com algum conceito da ontologia do modelo de reputação Repage. Como não há nenhum subconceito que atenda a este critério, a busca é invertida, realizando o mesmo procedimento com os superconceitos do conceito *Reputação Secundária*. Neste processo, identifica-se o conceito *Reputação Secundária* da FORe com uma correspondência para o conceito *Reputação* da ontologia do modelo de reputação Repage, e este é retornado como resultado.

6.2.2.4 Interface do Módulo de Linguagem

Foi adotada a linguagem de consulta *Simple Protocol and RDF Query Language* (SPARQL) (PRUD'HOMMEAUX; SEABORNE, 2007) para a transmissão e solicitação de informação sobre reputação. A escolha desta linguagem teve inicialmente como

base a possibilidade de se elaborar consultas sobre reputação com maior grau de expressividade. No entanto, no decorrer dos experimentos identificou-se que não haveria necessidade do uso de linguagem com tal expressividade, já que as consultas sobre reputação podem ser expressas em uma linguagem de consulta comum como *Structure Query Language* (SQL).

Este componente implementa as funções relacionadas à análise léxica e sintática, bem como montagem de mensagens com conteúdo descrito em formato SPARQL. Para tanto, foi utilizado o gerador de analisador léxico e sintático denominado *JavaCC versão 4.0*⁸.

A seguir é demonstrado o funcionamento da arquitetura SOARI conforme descrito na seção 5.4.

6.3 Funcionamento

Nesta seção é ilustrado o uso da arquitetura SOARI em um cenário de comércio eletrônico simplificado a fim de verificar e demonstrar seu funcionamento. Neste cenário existem dois tipos de agentes: consumidores e provedores de serviço. Os provedores publicam e provêm serviços, enquanto os consumidores procuram e contratam os serviços disponíveis.

Em ambientes de comércio eletrônico, inerentemente abertos e dinâmicos, diversos riscos são impostos aos agentes, como a realização de transações comerciais com provedores de serviços desconhecidos. Para reduzir estes riscos, cada agente deve incorporar um mecanismo de reputação para auxiliar na decisão de contratar ou não os serviços de determinado provedor.

Considerando este cenário, quando um consumidor deseja contratar um serviço, ele primeiro procura os provedores que têm o serviço desejado disponível. Em seguida, ele troca informações com outros agentes para avaliar a reputação de cada um dos provedores de serviço encontrados. Finalmente, ele escolhe o provedor a contratar com base no seu próprio conhecimento sobre o mesmo e nas avaliações de reputação recebidas dos outros agentes.

Suponha agora um sistema com dois consumidores, denominados Alice e Bob, que utilizam diferentes modelos de reputação, respectivamente, Repage e L.I.A.R.. Utilizando a arquitetura SOARI para possibilitar a comunicação entre os agentes, suponha que a agente Alice queira comprar uma passagem aérea

⁸<https://javacc.dev.java.net/>.

barata para voar de Roma para Paris. Inicialmente, Alice procura as empresas aéreas que têm vôos entre estas duas cidades e elabora um lista por ordem de custo. Porém, como ela nunca utilizou os serviços de nenhuma destas empresas aéreas, ela deseja escolher uma que seja confiável. Entretanto, Alice conhece Bob, que viaja muito pela Europa, então ela decide perguntar a ele sobre a reputação de cada uma das empresas aéreas a fim de obter mais informações que a auxiliarão em sua escolha.

Considerando que optou-se pela FORe como *Ontologia Comum no Domínio da Reputação*, a SOARI necessita estar habilitada para realizar a tradução de conceitos das ontologias dos modelos de reputação Repage e L.I.A.R. para FORe e vice-versa. Isto ocorre na fase de *Inicialização* da arquitetura apresentada na próxima seção.

6.3.1 Inicialização

Conforme apresentado na seção 5.4.1, a fase de *Inicialização* é composta pelas atividades de criação da *Ontologia do Modelo de Reputação* dos agentes, alinhamento desta ontologia com a *Ontologia Comum no Domínio da Reputação*, e geração do mapeamento entre os conceitos destas duas ontologias.

Nas próximas seções, estas três atividades são realizadas para o cenário de comércio eletrônico descrito, considerando os modelos de reputação Repage e L.I.A.R., bem como a *Ontologia Comum no Domínio da Reputação* FORe.

6.3.1.1 Criar a ontologia do modelo de reputação

As ontologias dos modelos de reputação Repage e L.I.A.R. foram construídas usando a linguagem OWL DL no ambiente de edição de ontologias Protégé.

Os conceitos dos modelos de reputação Repage e L.I.A.R., necessários para o mapeamento de modelos de reputação foram citados na seção 3.3 e estão representados na primeira coluna das Tabelas 6.3 e 6.4, respectivamente. Sua implementação em termos ontológicos é mostrada na segunda coluna das respectivas tabelas. Tais conceitos compreendem a ontologia do modelo de reputação e foram identificados usando um processo manual de análise.

Tabela 6.3: Conceitos da ontologia do modelo de reputação Repage

Conceito do Modelo Repage	Classe da Ontologia Repage
Natureza Imagem	AGENTIMAGE
Natureza Reputação	AGENTREPUTATION
Imagem	IMAGE
Reputação	REPUTATION
Papel Alvo	TARGET
Papel Avaliador	EVALUATOR
Papel Beneficiário	BENEFICIARY
Papel Propagador	PROPAGATOR
Valor	VALUE

6.3.1.2 Alinhar a ontologia do modelo de reputação

O processo de alinhamento das ontologias dos modelos de reputação Repage e L.I.A.R. também foi realizado manualmente, sendo o resultado apresentado nas Tabelas 6.5 e 6.6, respectivamente. Neste caso, o alinhamento foi realizado relacionando conceitos da FORe com os conceitos dos dois modelos de reputação, estando as relações binárias definidas na FORe.

6.3.1.3 Gerar o mapeamento

Finalizado o alinhamento, um agente realiza a chamada do método *addTranslationOntology* do SMO, transmitindo para este último o arquivo da ontologia alinhada. O SMO o armazena então no *Repositório de Ontologias* do SMO e aciona o *Módulo Classificador* para realizar o processamento conforme o pseudo-algoritmo 1 descrito na seção 5.3.1. O resultado gerado é então armazenado no *Repositório de Traduções*.

As Tabelas 6.7 e 6.8 apresentam o resultado obtido pelo mapeamento dos modelos de reputação Repage e L.I.A.R., respectivamente.

Tabela 6.4: Conceitos da ontologia do modelo de reputação L.I.A.R.

Conceito do Modelo L.I.A.R.	Classe da Ontologia L.I.A.R.
Faceta da Reputação	FACET
Reputação baseada em Interação Direta	DIBRP
Reputação baseada em Interação Indireta	IIBRP
Reputação baseada em Recomendações Observadas	OBSRCBRP
Reputação baseada em Recomendações Avaliadas	EVRCBRP
Reputação baseada em Recomendações de Reputação	RPRCBRP
Papel Alvo	TARGET
Papel Avaliador	EVALUATOR
Papel Beneficiário	BENEFICIARY
Papel Julgador	PUNISHER
Papel Observador	OBSERVER
Papel Participante	PARTICIPANT
Papel Propagador	PROPAGATOR
Valor da Reputação	REPUTATIONVALUE

Tabela 6.5: Alinhamento do modelo de reputação Repage com a FORe

Classe Repage	Relações Binárias
AGENTIMAGE	$\exists isReputationNatureOf(Activity\ or\ Event\ or\ GroupOfIndividual\ or\ Individual\ or\ Location\ or\ Object\ or\ ReputativeSkill\ or\ Service)$
AGENTREPUTATION	$\exists isReputationNatureOf(Activity\ or\ Event\ or\ GroupOfIndividual\ or\ Individual\ or\ Location\ or\ Object\ or\ ReputativeSkill\ or\ Service)$
IMAGE	$\exists hasInformationSource(DirectExperience\ or\ Observation\ or\ SecondHandInformation)$
REPUTATION	$\exists hasInformationSource(GroupInheritance\ or\ SecondHandInformation)$
TARGET	$\exists hasReputation(ReputationType)$
EVALUATOR	$\exists hasReputationProcess(ReputationMaintenanceProcess)$ $\exists hasReputationProcess(ReputationEvaluationProcess)$
BENEFICIARY	$\exists hasReputationProcess(ReputationReceptionProcess)$
PROPAGATOR	$\exists hasReputationProcess(ReputationTransmissionProcess)$
VALUE	$\exists isOutputOf(ReputationMaintenanceProcess)$ $\exists isOutputOf(ReputationEvaluationProcess)$ $\exists isReputationValueOf(Reputation)$ $\exists isReputationValueOf(Image)$

Tabela 6.6: Alinhamento do modelo de reputação L.I.A.R. com a FORe

Classe L.I.A.R.	Relações Binárias
FACET	$\exists isReputationNatureOf(Activity\ or\ Event\ or\ GroupOfIndividual\ or\ Individual\ or\ Location\ or\ Object\ or\ ReputativeSkill\ or\ Service)$
DIBRP	$\exists hasInformationSource(DirectExperience)$ $\exists hasReputationValue(ReputationValue)$
IIBRP	$\exists hasInformationSource(Observation)$ $\exists hasReputationValue(ReputationValue)$
OBSRCBRP	$\exists hasInformationSource(Observation)$ $\exists hasReputationValue(ReputationValue)$
EVRCBRP	$\exists hasInformationSource(SecondHandInformation)$ $\exists hasReputationValue(ReputationValue)$
RPRCBRP	$\exists hasInformationSource(SecondHandInformation)$ $\exists hasReputationValue(ReputationValue)$
TARGET	$\exists hasReputation(ReputationType)$
EVALUATOR	-
BENEFICIARY	$\exists hasReputationProcess(ReputationReceptionProcess)$
Punisher	$\exists hasReputationProcess(ReputationMaintenanceProcess)$ $\exists hasReputationProcess(ReputationEvaluationProcess)$
OBSERVER	-
PARTICIPANT	-
PROPAGATOR	$\exists hasReputationProcess(ReputationTransmissionProcess)$
REPUTATIONVALUE	$\exists isOutputOf(ReputationEvaluationProcess)$ $\exists isOutputOf(ReputationMaintenanceProcess)$ $\exists isReputationValueOf(DIbRp)$ $\exists isReputationValueOf(IIbRp)$ $\exists isReputationValueOf(ObsRcbRp)$ $\exists isReputationValueOf(EvRcbRp)$ $\exists isReputationValueOf(RpRcbRp)$

Tabela 6.7: Mapeamento do modelo de reputação Repage para FORe

Classe Repage	Classe FORe
AGENTREPUTATION	REPUTATIONNATURE
AGENTIMAGE	
IMAGE	REPUTATIONTYPE
REPUTATION	SECONDARYREPUTATION
EVALUATOR	EVALUATORROLE MAINTAINERROLE
BENEFICIARY	RECIPIENTROLE
TARGET	TARGETROLE
PROPAGATOR	TRANSMITTERROLE
VALUE	REPUTATIONEVALUATIONVALUE REPUTATIONFINALVALUE

Tabela 6.8: Mapeamento do modelo de reputação L.I.A.R. para FORe

Classe L.I.A.R.	Classe FORe
FACET	REPUTATIONNATURE
DIBRP	DIRECTREPUTATION
IIBRP	OBSERVEDREPUTATION
OBSRCBRP EVRCBRP RPRCBRP	PROPAGATEDREPUTATION
PUNISHER	EVALUTORROLE MAINTAINERROLE
BENEFICIARY	RECIPIENTROLE
TARGET	TARGETROLE
PROPAGATOR	TRANSMITTERROLE
REPUTATIONVALUE	REPUTATIONEVALUATIONVALUE REPUTATIONFINALVALUE

6.3.2 Operação

Uma vez terminada a fase de *inicialização*, a arquitetura SOARI está pronta para auxiliar os agentes na troca de informações sobre reputação. Assim, a agente Alice, com o objetivo de obter a reputação das empresas aéreas, aciona seu componente MTR requisitando a ele que consulte o agente Bob sobre tal informação. O componente MTR, através do IMTR, elabora uma mensagem de consulta usando os conceitos do modelo de reputação Repage (modelo de reputação interno do agente). Como já mencionado anteriormente, neste trabalho é utilizada a linguagem de consulta SPARQL.

A mensagem resultante, utilizando os conceitos do modelo de reputação Repage, é mostrada na Figura 6.1.

```

SELECT ?AgentName ?Reputation ?Value
WHERE {?x reputation:AgentName ?AgentName .
       ?x reputation:Value ?Value .
       ?x reputation:Reputation ?Reputation .
       ?x reputation:AgentReputation ?RepNature.
       FILTER ((?Reputation = true) && REGEX(?RepNature,Airline))
}

```

Figura 6.1: Consulta em SPARQL utilizando os conceitos da ontologia do modelo de reputação Repage

A mensagem é então enviada para o CT que aciona o IML. Este último componente identifica os conceitos VALUE, REPUTATION e AGENTREPUTATION

como sendo conceitos da *Ontologia do Modelo de Reputação* que necessitam de tradução. O CT consulta então o ICSMO para obter as possíveis traduções para cada um dos conceitos identificados da ontologia do modelo Repage para a *Ontologia Comum no Domínio da Reputação*. O ICSMO verifica em sua tabela interna se tal tradução já foi realizada anteriormente, em caso afirmativo, reaproveita a tradução. Em caso negativo, consulta o SMO para obter as possíveis traduções, conforme ilustra a Figura 6.2.

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soapenvelope">
  <soapenv:Body>
    <ns2:translateConcept xmlns:ns2="http://webservice.mapper.otservices">
      <ns2:fromConcept>Reputation</ns2:fromConcept>
      <ns2:ontFrom>
        <ns1:type xmlns:ns1="http://translationrepository.mapper.otservices/xsd">2</ns1:type>
        <ns1:uri xmlns:ns1="http://translationrepository.mapper.otservices/xsd">REPAGE</ns1:uri>
        <ns1:version xmlns:ns1="http://translationrepository.mapper.otservices/xsd">1</ns1:version>
      </ns2:ontFrom>
      <ns2:ontTo>
        <ns1:type xmlns:ns1="http://translationrepository.mapper.otservices/xsd">1</ns1:type>
        <ns1:uri xmlns:ns1="http://translationrepository.mapper.otservices/xsd">FORE</ns1:uri>
        <ns1:version xmlns:ns1="http://translationrepository.mapper.otservices/xsd">1</ns1:version>
      </ns2:ontTo>
    </ns2:translateConcept>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 6.2: Mensagem SOAP para tradução do conceito *Reputation*

A Tabela 6.9 apresenta a possível tradução para os conceitos identificados pelo IML.

Tabela 6.9: Mapeamento parcial dos conceitos do Repage para FORe

Classe Repage	Classe FORe
VALUE	REPUTATIONEVALUATIONVALUE REPUTATIONFINALVALUE
REPUTATION	SECONDARYREPUTATION
AGENTREPUTATION	REPUTATIONNATURE

Quando a agente Alice recebe a resposta das consultas do SMO, o ICSMO armazena a tradução em sua tabela interna para uso futuro. Em seguida, a IET é acionada para resolver todos os possíveis *Problemas de Tradução* que possam existir. Neste caso, o único problema está relacionado a tradução do conceito Repage VALUE para FORe, já que o mesmo foi traduzido para dois possíveis conceitos. Usando uma estratégia de tradução heurística, sendo esta a seleção do primeiro conceito da lista, o conceito VALUE é traduzido para REPUTATIONEVALUATIONVALUE. Como nesta consulta não existe nenhum valor numérico para tradução, o CT não aciona o componente ITV, e a mensagem resultante da tradução é mostrada na Figura 6.3, já com a utilização dos conceitos da FORe.

```

SELECT ?AgentName ?SecondaryReputation ?ReputationEvaluationValue
WHERE {?x reputation:AgentName ?AgentName .
      ?x reputation:ReputationEvaluationValue ?ReputationEvaluationValue .
      ?x reputation:SecondaryReputation ?SecondaryReputation .
      ?x reputation:ReputationNature ?RepNature .
      FILTER ((?SecondaryReputation = true) && (REGEX(?RepNature,Airline)))
}

```

Figura 6.3: Consulta em SPARQL utilizando os conceitos da *Ontologia Comum no Domínio da Reputação FORE*

Em seguida, esta mensagem é redirecionada para o IMI que aciona o módulo MI do agente. Este último encapsula a mensagem em um protocolo de comunicação⁹ antes de enviá-la para o agente Bob, conforme ilustrado na Figura 6.4.

```

(REQUEST
 :receiver (set ( agent-identifier :name Bob@machine1:5555/JADE ) )
 :content "SELECT ?agentname ?secondaryreputation ?reputationevaluationvalue WHERE
 {?x reputation:agentname ?agentname .?x reputation:reputationevaluationvalue
 ?reputationevaluationvalue . ?x reputation:secondaryreputation ?secondaryreputation .
 ?x reputation:reputationnature ?repnature . FILTER ((?secondaryreputation = true) &&
 (REGEX(?repnature,Airline))) }"
 :reply-with bob40
 :language SPARQL
 :ontology FORE_v_1 )

```

Figura 6.4: Mensagem FIPA-ACL

O agente Bob recebe a mensagem enviada por Alice através de seu módulo MI e extrai informações da mensagem. Estas informações lhe permitem identificar que a mensagem está relacionada com reputação, assim seu conteúdo é redirecionado para o componente IMI. O CT recebe o conteúdo da mensagem do IMI e o encaminha para o IML para obter a lista de conceitos FORE. O IML reconhece os conceitos REPUTATIONEVALUATIONVALUE, SECONDARYREPUTATION e REPUTATIONNATURE como sendo conceitos FORE. O CT então aciona o ICSMO, que consulta o SMO através do envio de uma mensagem SOAP semelhante àquela mostrada na Figura 6.2, já que não possui a tradução para estes conceitos em sua tabela interna, para obter as possíveis traduções destes conceitos. O resultado da tradução dos conceitos é apresentado na Tabela 6.10.

O CT aciona o IET para que todos os *Problemas de Tradução* sejam resolvidos. Nesta tradução, existe somente um problema relacionado ao conceito SECONDARYREPUTATION, pois não existe nenhum conceito L.I.A.R. que esteja diretamente relacionado com este conceito FORE. Para resolver este problema existem duas possíveis estratégias: remover o conceito da mensagem traduzida, ou usar algum tipo de heurística para determinar uma possível tradução implícita.

⁹Neste caso, adotou-se a linguagem FIPA-ACL (<http://www.fipa.org>).

Tabela 6.10: Mapeamento parcial dos conceitos da FORe para L.I.A.R.

Classe FORe	Classe L.I.A.R.
REPUTATIONEVALUATIONVALUE	REPUTATIONVALUE
SECONDARYREPUTATION	
REPUTATIONNATURE	FACET

Como a primeira alternativa pode invalidar a mensagem, a melhor estratégia é usar uma heurística que consiste em identificar todos os conceitos do modelo de reputação L.I.A.R. que estão relacionados com conceitos da FORe, que são sub-conceitos do SECONDARYREPUTATION na ontologia classificada obtida do SMO. Por exemplo, os conceitos EVRcbRp, OBSRcbRp e RPRcbRp estão relacionados ao conceito PROPAGATEDREPUTATION que é subconceito do conceito SECONDARYREPUTATION. Assim, a mensagem resultante da tradução é mostrada na Figura 6.5.

```

SELECT ?AgentName ?EvRcbRp ?ObsRcbRp ?RpRcbRp ?ReputationValue
WHERE {?x reputation:AgentName ?AgentName .
      ?x reputation:ReputationValue ?ReputationValue .
      ?x reputation:EvRcbRp ?EvRcbRp .
      ?x reputation:ObsRcbRp ?ObsRcbRp .
      ?x reputation:RpRcbRp ?RpRcbRp .
      ?x reputation:Facet ?RepNature .
      FILTER((?EvRcbRp=true || ?ObsRcbRp=true || ?RpRcbRp=true) &&
              (REGEX(?RepNature,Airline)))
}

```

Figura 6.5: Consulta em SPARQL utilizando os conceitos da ontologia do modelo de reputação L.I.A.R.

Depois que a mensagem é traduzida, o CT a envia para o componente IMTR que interpreta a consulta SPARQL e aciona o componente MTR. Este último processa a requisição e retorna uma resposta ao componente IMTR, conforme ilustrado na Figura 6.6.

```

((AgentName=A; ReputationValue=0.7; EvRcbRp=false; ObsRcbRp=false; RpRcbRp=true);
 (AgentName=A; ReputationValue=0.4; EvRcbRp=true; ObsRcbRp=false; RpRcbRp=false);
 (AgentName=B; ReputationValue=0.9; EvRcbRp=false; ObsRcbRp=false; RpRcbRp=true);
 (AgentName=C; ReputationValue=0.1; EvRcbRp=false; ObsRcbRp=true; RpRcbRp=false))

```

Figura 6.6: Resultado da consulta SPARQL no modelo de reputação L.I.A.R.

O IMTR recebe esta informação e a transforma no formato SPARQL Query Results XML Format (BECKETT; BROEKSTRA, 2008), conforme parcialmente ilustrado na Figura 6.7, o qual é enviado de volta a agente Alice, num procedimento semelhante.

```

<?xml version="1.0" encoding="UTF-8"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="?agentname"/>
    <variable name="?reputationvalue"/>
    <variable name="?evrcbrp"/>
    <variable name="?obsrcbrp"/>
    <variable name="?rprcbrp"/>
  </head>
  <results>
    <result>
      <binding name="?agentname">
        <literal>A</literal>
      </binding>
      <binding name="?reputationvalue">
        <literal datatype="http://www.w3.org/2001/XMLSchema#float">0.7</literal>
      </binding>
      <binding name="?evrcbrp">
        <literal datatype="http://www.w3.org/2001/XMLSchema#boolean">>false</literal>
      </binding>
      <binding name="?obsrcbrp">
        <literal datatype="http://www.w3.org/2001/XMLSchema#boolean">>false</literal>
      </binding>
      <binding name="?rprcbrp">
        <literal datatype="http://www.w3.org/2001/XMLSchema#boolean">>true</literal>
      </binding>
    </result>
    . . .
  </results>
</sparql>

```

Figura 6.7: Resultado da consulta em formato XML

6.4 Considerações Finais

A descrição das fases de funcionamento deste capítulo sugere que todas as ontologias de modelos de reputação necessitam ser mapeadas antes do início da fase de *Operação*. No entanto, esta restrição não existe. Quando um novo agente entra no ambiente e não encontra o mapeamento de sua *Ontologia de Modelo de Reputação* para a *Ontologia Comum no Domínio da Reputação*, ele solicita a realização deste mapeamento através do acionamento do método *addOntologyTranslation*, passando ao SMO o arquivo contendo sua ontologia alinhada a FORe.

Este cenário simplificado de comércio eletrônico, apresentado originalmente em (NARDIN et al., 2008b), ilustrou o funcionamento da arquitetura SOARI. O próximo capítulo apresenta uma aplicação da arquitetura nas plataformas de simulação ART e FOReART, que têm como cenário o mercado de avaliações de obras de arte. Esta aplicação visa fornecer subsídios para responder às questões formuladas na seção 1.2.

7 Exemplo de Aplicação da Arquitetura

Inicialmente, na seção 7.1 é fornecida uma breve introdução sobre os objetivos da realização dos experimentos. Na seção 7.2, são apresentadas as plataformas de simulação ART e FOrEART, seguidas da descrição do modelo de agente utilizado na seção 7.3. Os experimentos propriamente ditos e os resultados obtidos encontram-se na seção 7.4. A metodologia e análise dos resultados são apresentados na seção 7.5. Por fim, na seção 7.6, são apresentadas as considerações finais sobre os experimentos realizados.

7.1 Introdução

A plataforma SOARI possibilita a interoperabilidade sobre reputação entre agentes que utilizam modelos heterogêneos de reputação, conforme demonstrado na seção 6.3. Assim, este capítulo tem como objetivo avaliar se o fato de interoperar com agentes heterogêneos tem algum impacto efetivo no comportamento dos agentes. Em particular, pretende-se responder às questões propostas na seção 1.2:

1. Existe uma melhora de exatidão na identificação dos agentes desonestos por agentes que utilizam uma maior expressividade em suas interações sobre reputação?
2. Pode-se afirmar que sociedades compostas por agentes com modelos de reputação distintos têm uma melhora de exatidão na identificação dos agentes desonestos em relação as sociedades homogêneas?

Para tanto, alguns experimentos serão realizados nas plataformas de simulação ART e FOrEART. Nos experimentos, um agente deliberadamente mentirá sobre a reputação de outros agentes e sobre o valor das pinturas. A análise será realizada para determinar a exatidão com que os outros agentes detectam o agente

mentiroso através do uso de reputação.

Deseja-se verificar também se uma comunicação mais expressiva ou a heterogeneidade de modelos de reputação influenciam na exatidão da identificação da reputação do agente mentiroso.

A seguir são descritas as plataformas de simulação usadas na realização dos experimentos.

7.2 Ambientes de Simulação

As plataformas ART e FOrEART resultam de um esforço de vários grupos de pesquisa na área de reputação de agentes em prover um ambiente comum para comparar os diversos modelos e implementações existentes.

Além do fato de serem padrão na área, a adoção das plataformas de simulação ART e FOrEART para a realização dos experimentos foi motivada principalmente pelo conhecimento e bom contato com a equipe que o desenvolveu, facilitando assim o esclarecimento de dúvidas e apoio nas simulações.

7.2.1 Plataforma de Simulação ART

A plataforma ART (FULLAM et al., 2005a) simula uma competição iterativa entre diversos agentes. O cenário simulado da competição é similar ao apresentado no capítulo 1.

Neste cenário existem dois tipos de agentes: os agentes avaliadores e os agentes clientes. Os agentes clientes contratam os agentes avaliadores para avaliarem suas pinturas. Os agentes avaliadores possuem conhecimento variado sobre as pinturas de diferentes épocas e podem requisitar a opinião de outros agentes para auxiliá-los em suas avaliações.

A simulação se inicia com o cadastro dos agentes na plataforma de simulação ART. Em seguida, o motor de simulação executa várias vezes o ciclo de simulação, acionando um conjunto de funções nos agentes de forma síncrona e ordenada, conforme mostra o pseudo-algoritmo 2.

No início de cada ciclo, o simulador aloca para cada agente avaliador uma quantidade específica de clientes (*setMarketShares()*). Tais clientes requisitam aos agentes avaliadores a apreciação de uma pintura de uma época específica e pagam um valor fixo por esta avaliação. O agente avaliador, baseado em seu conheci-

Pseudo-Algoritmo 2 Ciclo de Simulação

```

while  $t < \text{numberOfTimeStepsPerSession}$  do
  setMarketShares()
  manageReputationRequests()
  manageReputationAcceptsAndDeclines()
  manageReputationReplies()
  manageCertaintyRequests()
  manageCertaintyReplies()
  manageOpinionRequests()
  fulfillOpinionCreationOrders()
  collectWeights()
  manageOpinionReplies()
  calculateAppraisals()
  distributeAppraisalData()
end while

```

mento, determina se possui ou não competência para avaliá-la, definindo assim se deve solicitar ou não a opinião de outros agentes avaliadores. Em caso positivo, consulta na sociedade a reputação de possíveis avaliadores de pintura da época desejada (*manageReputationRequests()*, *manageReputationAcceptsAndDeclines()* e *manageReputationReplies()*). Em seguida, define um conjunto de agentes aos quais solicita seu grau de certeza em relação a mesma época de pintura (*manageCertaintyRequests()* e *manageCertaintyReplies()*). Por grau de certeza entende-se quanto o agente conhece ou finge conhecer a respeito de pinturas da época consultada. Para a solicitação de ambas as informações, reputação e grau de certeza, o agente solicitante paga um valor fixo em dinheiro ao fornecedor da informação, o que não garante a veracidade da informação fornecida. Com base na reputação e no grau de certeza informados, o agente avaliador decide então a quais agentes irá solicitar a opinião a respeito da pintura (*manageOpinionRequests()*).

Em seguida, os agentes cuja opinião foi solicitada informam ao simulador quanto recurso pretendem despendar para atender a cada uma das solicitações recebidas (*fulfillOpinionCreationOrders()*). A quantidade de recursos é dada por um valor entre 0 e o custo total da opinião. Simultaneamente, os agentes que solicitaram a opinião informam ao simulador o percentual a ser considerado de cada uma das opiniões solicitadas (*collectWeights()* e *manageOpinionReplies()*). O simulador, de posse destas informações, calcula o valor avaliado final da pintura e o compara com seu valor real (*calculateAppraisals()*). Esta comparação tem a finalidade de determinar a exatidão dos agentes na avaliação de pinturas de determinada época. Por fim, a informação referente à opinião informada ao simulador é enviada aos agentes solicitantes (*distributeAppraisalData()*).

Nos próximos ciclos de simulação, mais ou menos clientes são alocados pelo

simulador a cada um dos agentes avaliadores, segundo a exatidão de suas avaliações passadas. Ao término da competição, o agente mais bem sucedido é aquele que acumula maior valor em dinheiro.

Na plataforma ART, a interoperabilidade entre agentes é obtida através de uma representação comum da reputação que consiste em um valor numérico entre 0 e 1. Apesar de não explicitamente definido, assume-se que o valor 0 refere-se a uma reputação ruim e 1 a uma reputação boa.

Considerando o uso desta plataforma, suponha que os agentes A e B são implementados com os modelos de reputação Repage e L.I.A.R., respectivamente. Considere que o agente A solicita ao agente B informação a respeito da reputação do agente C para a pintura da época modernista. O agente B internamente em seu modelo de reputação possui os seguintes valores de reputação a respeito do agente C referente a época modernista: $DIbRp = 0,8$, $IIbRp = 0,6$, $ObsRcbRp = 0,7$, $EvRcpRp = 0,8$ e $RpRcbRp = 0,4$. Para responder ao agente A, ele utiliza seu processo de raciocínio interno e chega a dois resultados, um indicando a intenção de confiança igual a *verdadeiro* e outro informando o valor de confiança de 0,6. Porém, para responder ao agente A, o agente B necessita representar estes dois valores em um único, já que a forma de representação considerada na plataforma ART é um único número entre 0 e 1. Assim, o agente B verifica que a intenção de confiança é verdadeiro e retorna para o agente A o valor de reputação igual ao valor de confiança, ou seja, 0,6. Ao receber a reputação do agente C, o agente A converte este valor para sua representação interna de reputação e o utiliza conforme necessário.

7.2.2 Plataforma de Simulação FOReART

A plataforma ART, apesar de possibilitar a simulação sobre reputação, possui diversas limitações:

1. Limita a expressividade da comunicação entre os agentes a uma faixa de valores entre 0 e 1;
2. Não disponibiliza informações suficientes para que os agentes utilizem todo o potencial de seus modelos de reputação. Por exemplo, mesmo sendo possível para um agente que utiliza o modelo de reputação L.I.A.R. calcular a *Reputação baseada em Interação Indireta* de outro agente, como a plataforma não possibilita que os agentes observem as interações de outros agentes, este tipo de reputação não pode ser calculado;

3. Gera valores iniciais aleatórios para a realização das simulações, impossibilitando o uso do mesmo conjunto de dados de entrada em diversas simulações a fim de comparar seus resultados.

Diante destas limitações, Brandão et al. (2007) propuseram uma extensão da plataforma ART, denominada FOrEART, para possibilitar a comunicação dos agentes através da transmissão de mensagens simbólicas que representam conceitos presentes na ontologia FOrE. Apesar de eliminar a limitação referente a expressividade da comunicação, esta alteração não atende a todas as necessidades dos experimentos a serem realizados neste trabalho. Sendo assim, duas novas alterações foram implementadas na plataforma FOrEART para atender as necessidades deste trabalho:

1. Substituição das funções aleatórias para a geração de valores iniciais por funções que carregam dados de um arquivo texto. Esta alteração possibilita que o mesmo conjunto de dados de entrada inicial seja utilizado em todas as simulações, possibilitando assim a comparação dos resultados obtidos entre os cenários;
2. Disponibilização da informação referente a todas as interações que ocorreram no ciclo anterior a todos os agentes (*distributeAppraisalData()*). Desta forma, o agente pode identificar que um mesmo agente lhe forneceu um valor de opinião sobre uma época de pintura diferente da fornecida a outro agente.

A seguir é apresentada a estratégia usada pelos agentes na realização dos experimentos deste trabalho.

7.3 Modelo de Agente

Todos os agentes usam a mesma estratégia geral para tomada de decisão para solicitar e responder às solicitações, conforme mostra o pseudo-algoritmo 3. A estratégia é implementada em funções que são acionadas pelo motor de simulação das plataformas de simulação, conforme apresentado na seção 7.2.1.

Inicialmente, o agente atualiza o valor de reputação dos outros agentes com base nas informações do último ciclo da simulação. Em seguida, solicita a todos estes a reputação de possíveis agentes avaliadores de pinturas para as quais foi contratado por um agente cliente (*prepareReputationRequests()*).

Pseudo-Algoritmo 3 Estratégia geral para solicitação de informação

```

prepareReputationRequests(){
    Request reputation of all other agents
}

prepareReputationAcceptsAndDeclines(){
if lie() || (expertise > expertiseThreshold) then
    return true
else
    return false
end if
}

prepareReputationReplies(){
if lie() then
    return reputation - 0,4
else
    return reputation
end if
}

prepareCertaintyRequests(){
    Request certainty of the first in the list
}

prepareCertaintyReplies(){
if lie() then
    return expertise + 0,5
else if (expertise > expertiseThreshold) then
    return experise
end if
}

prepareOpinionRequests(){
if TrustFor() || (certanty > certaintyThreshold) then
    Request opinion
end if
}

prepareOpinionCreationOrders(){
    Inform the simulator how much to spend on the opinion request
}

prepareOpinionProviderWeights(){
    Inform simulator about the weight to consider of each requested opinion
}

prepareOpinionReplies(){
    Send opinion information to the requesting agents
}

```

O agente então responde as solicitações recebidas. Caso seja um mentiroso, sempre aceita responder às solicitações, já que tentará se autopromover, fazendo com que os outros pensem que tem um bom conhecimento de reputação dos outros. No caso dos outros agentes, somente aceitam as solicitações caso sua competência for superior a um limite pré-estabelecido (*prepareReputationAcceptsAndDeclines()*). Para todas as solicitações aceitas, responde com o valor da reputação do agente e época solicitados (*prepareReputationReplies()*), que pode ser incorreta quando o agente for mentiroso: ele sempre tentará desqualificar os outros agentes.

Em seguida, o agente seleciona um grupo de agentes e solicita o grau de certeza que eles possuem a respeito de pinturas de determinada época (*prepareCertaintyRequests()*). Para as solicitações de grau de certeza recebidas, o agente verifica se deve ou não mentir ao agente solicitante. Caso determine que não deve mentir e sua competência for superior ao limiar *expertiseThreshold*, ele responde com o valor de sua competência. No entanto, se determinar que deve mentir para o agente solicitante, então responde com o valor mínimo entre 1 e o seu valor de competência acrescido de 0,5 (*prepareCertaintyReplies()*).

O agente então solicita a opinião somente dos agentes em que confia (i.e., cujo valor de reputação é superior a um determinado limite) ou do qual recebeu o valor de grau de certeza superior a um limiar (*prepareOpinionRequests()*).

Na seqüência, o agente informa ao simulador: (1) quanto recurso pretende despendar para o atendimento às solicitações de opinião aceitas (*prepareOpinionCreationOrders()*), e (2) o percentual, que é calculado baseado no valor da reputação destes agentes, a ser considerado pelo simulador das opiniões solicitadas aos outros agentes (*prepareOpinionProviderWeights()*).

Por fim, envia para os agentes solicitantes os valores das opiniões informadas ao simulador (*prepareOpinionReplies()*).

7.4 Experimentos Realizados

Com o objetivo de se analisar o impacto na exatidão da identificação da reputação de outros agentes devido à interoperabilidade entre agentes com modelos de reputação heterogêneos, diversos experimentos foram realizadas usando as plataformas de simulação ART e FOReART e o modelo de agente descrito na seção 7.3.

Os experimentos se distinguem pela combinação de dois fatores: (1) tipos

de modelo de reputação utilizados pelos agentes na simulação, e (2) forma de comunicação sobre reputação entre os agentes. Foram realizados experimentos com apenas um modelo de reputação (Repage ou L.I.A.R.) ou com ambos. Já as formas possíveis de comunicação entre os agentes são *Numérica* e *Simbólica*. A Tabela 7.1 apresenta os experimentos possíveis com a combinação destes dois fatores.

Tabela 7.1: Tabela de experimentos realizados

ID	Experimento	Modelo de Reputação	Comunicação
1	ART/L.I.A.R.	L.I.A.R.	Numérica
2	ART/Repage	Repage	Numérica
3	ART/Misto	L.I.A.R. e Repage	Numérica
4	FOReART/L.I.A.R.	L.I.A.R.	Simbólica
5	FOReART/Repage	Repage	Simbólica
6	FOReART/Misto	L.I.A.R. e Repage	Simbólica

Nos experimentos existem dois tipos de agentes: *Honesto* e *Desonesto*. Os agentes *Honestos* sempre respondem às solicitações somente se possuem competência sobre a época de pintura solicitada, sendo a informação transmitida coerente com o seu estado interno. Já o agente *Desonesto* responde a todas às solicitações, não importando sua competência; no entanto, nunca responde às solicitações com a informação coerente com o seu estado interno, e sempre tenta denegrir a imagem dos outros agentes. O comportamento do agente desonesto é portanto determinista: ele sempre responde do mesmo modo as requisições, conforme citado na seção 7.3.

O objetivo principal dos experimentos é identificar o valor médio e o desvio padrão da reputação atribuídos pelos agentes do tipo *Honesto* a uma determinada época de pintura dos agentes do tipo *Desonesto*. Estes valores são posteriormente analisados na seção 7.5. A fim de possibilitar tal análise, os dados iniciais de conhecimento de épocas de pinturas e a distribuição inicial de clientes são idênticas para todos os experimentos. Além disto, todos os agentes usam o mesmo modelo, conforme descrito na seção 7.3, e parametrização em todas as simulações de todos os experimentos.

Do ponto de vista formal, considere um conjunto de n agentes, onde $i = \{1, 2, \dots, n - 1\}$ são agentes do tipo *Honesto* e $j = n$ do tipo *Desonesto*. Seja

ainda r_{ij}^{sk} o valor da reputação atribuído pelo agente i ao agente j no ciclo k da simulação s . Tipicamente, o valor da reputação atribuído pelo agente i ao agente j na simulação s corresponde à média dos valores da reputação para um conjunto

de ciclos. Assim, $r_{ij}^s = \frac{\sum_{k=l}^m r_{ij}^{sk}}{m-l+1}$, onde l e m representam os limites inferior e superior de ciclos respectivamente. O valor médio de reputação atribuído então

pelos agentes *Honestos* ao agente *Desonesto* na simulação s é $r_j^s = \frac{\sum_{i=1}^{n-1} r_{ij}^s}{n-1}$.

Finalmente, dado um conjunto de simulações $s = \{1, \dots, p\}$ que compõem um experimento, o valor médio a ser testado é $r_j = \frac{\sum_{s=1}^p r_j^s}{p}$.

Neste trabalho, para todos os experimentos foram executadas 10 simulações distintas ($p = 10$), com 100 ciclos cada. Cada simulação era composta por 21 agentes ($n = 21$), sendo 20 do tipo *Honesto* e 01 do tipo *Desonesto* ($i = [1, 20]$ e $j = 21$). O valor médio e desvio padrão da reputação atribuído ao agente *Desonesto* por cada agente *Honesto* utiliza os valores obtidos no último ciclo da simulação ($l = 100$ e $m = 100$).

O tempo médio das simulações por experimento, considerando uma máquina com processador Intel Core 2 Duo T7500 2.20 GHz e 2 GB de memória RAM, é mostrado na Tabela 7.2, onde nos experimentos mistos é indicado o tipo do agente *Desonesto*.

Tabela 7.2: Tempo médio de execução das simulações

Experimento	Tempo médio de execução
ART/L.I.A.R.	04m 37s
ART/Repage	51m 25s
ART/Misto-D/L.I.A.R.	25m 20s
ART/Misto-D/Repage	27m 49s
FOReART/L.I.A.R.	10h 06m 31s
FOReART/Repage	08h 08m 09s
FOReART/Misto-D/L.I.A.R.	09h 53m 20s
FOReART/Misto-D/Repage	08h 08m 09s

Do ponto de vista experimental, todos os experimentos foram realizados usando a plataforma FOReART. No entanto, neste trabalho usa-se o termo plataforma ART quando a comunicação entre os agentes é limitada através de um único valor na faixa $[0, 1]$. No caso da comunicação entre os agentes ser realizada via a transmissão de mensagens simbólicas, a plataforma é então denominada

plataforma FOrEART.

A seguir são descritos mais especificamente cada um dos experimentos realizados e seus respectivos resultados.

7.4.1 Experimento 1: ART/L.I.A.R

Este experimento foi realizado apenas com agentes homogêneos cujo modelo de reputação é o L.I.A.R.. Além disto, interagem sobre reputação através da transmissão de um único valor na faixa $[0, 1]$.

A média e o desvio padrão dos valores dos atributos $DibRp$, $IibRp$ e $RpRcbRp$ atribuídos pelos agentes do tipo *Honesto* ao agente *Desonesto* são apresentados na Tabela 7.3.

Tabela 7.3: Experimento 1: ART/L.I.A.R. - Valor dos atributos do agente *Desonesto*

Simulação	$DibRp$	$IibRp$	$RpRcbRp$
1	$0,86 \pm 0,46$	$0,81 \pm 0,03$	$1,00 \pm 0,00$
2	$0,46 \pm 0,88$	$0,43 \pm 0,17$	$0,59 \pm 0,15$
3	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$1,00 \pm 0,00$
4	$0,76 \pm 0,62$	$0,72 \pm 0,05$	$0,99 \pm 0,01$
5	$0,70 \pm 0,73$	$0,74 \pm 0,06$	$0,99 \pm 0,01$
6	$0,80 \pm 0,62$	$0,85 \pm 0,03$	$1,00 \pm 0,00$
7	$0,64 \pm 0,73$	$0,58 \pm 0,08$	$1,00 \pm 0,00$
8	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$1,00 \pm 0,00$
9	$0,90 \pm 0,45$	$0,95 \pm 0,01$	$1,00 \pm 0,00$
10	$0,77 \pm 0,62$	$0,57 \pm 0,10$	$0,83 \pm 0,08$
média	$0,79 \pm 0,16$	$0,76 \pm 0,19$	$0,94 \pm 0,13$

Na Figura 7.1 é apresentada, nas ordenadas, a média dos valores dos atributos do modelo de reputação L.I.A.R. para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento, representado nas abcissas.

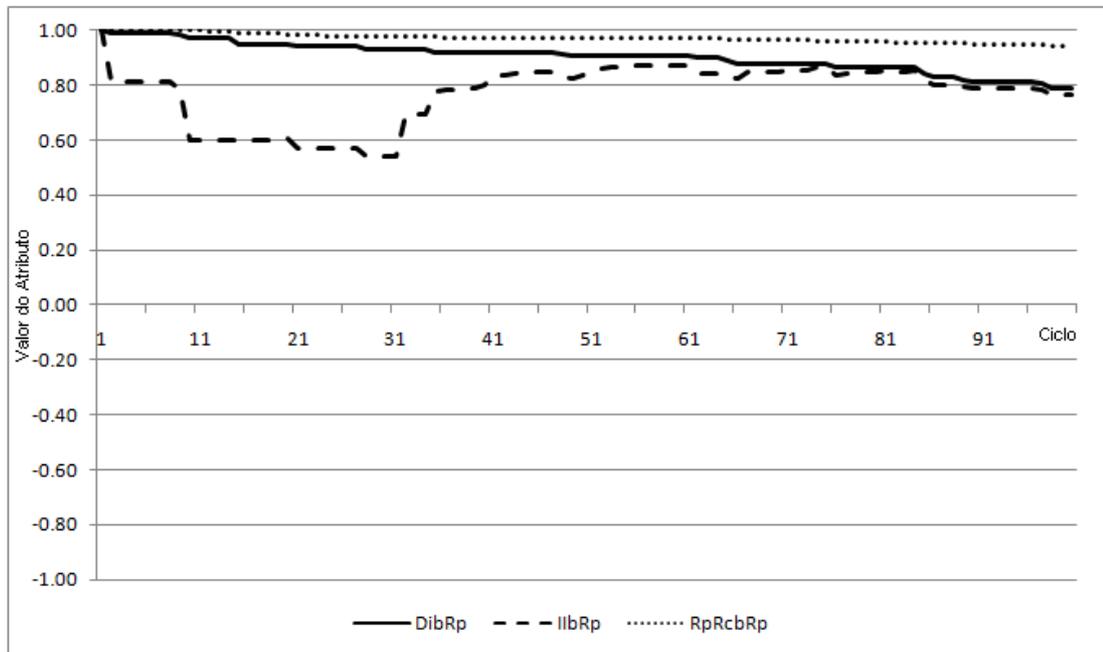


Figura 7.1: Experimento 1: ART/L.I.A.R. - Valor médio do $DIBRp$, $IIBRp$ e $RpRcbRp$ para o agente *Desonesto*

7.4.2 Experimento 2: ART/Repage

Este experimento foi realizado apenas com agentes homogêneos cujo modelo de reputação é o Repage. Além disto, interagem sobre reputação através da transmissão de um único valor na faixa $[0, 1]$.

Na Tabela 7.4, são apresentados a média e o desvio padrão dos valores dos atributos *Imagem* e *Reputação* atribuídos pelos agentes do tipo *Honesto* ao agente do tipo *Desonesto*.

Tabela 7.4: Experimento 2: ART/Repage - Valor dos atributos do agente *Desonesto*

Simulação	Imagem	Reputação
1	0,62 ± 0,11	0,90 ± 0,10
2	0,59 ± 0,03	0,75 ± 0,00
3	0,76 ± 0,01	1,00 ± 0,00
4	0,61 ± 0,04	0,71 ± 0,09
5	0,70 ± 0,05	1,00 ± 0,00
6	0,93 ± 0,03	1,00 ± 0,00
7	0,66 ± 0,08	0,76 ± 0,02
8	0,79 ± 0,05	1,00 ± 0,00
9	0,86 ± 0,01	1,00 ± 0,00
10	0,71 ± 0,04	1,00 ± 0,00
média	0,72 ± 0,11	0,91 ± 0,12

Na Figura 7.2 é ilustrado, nas ordenadas, o valor da média dos atributos do modelo de reputação Repage para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento, representado nas abcissas.

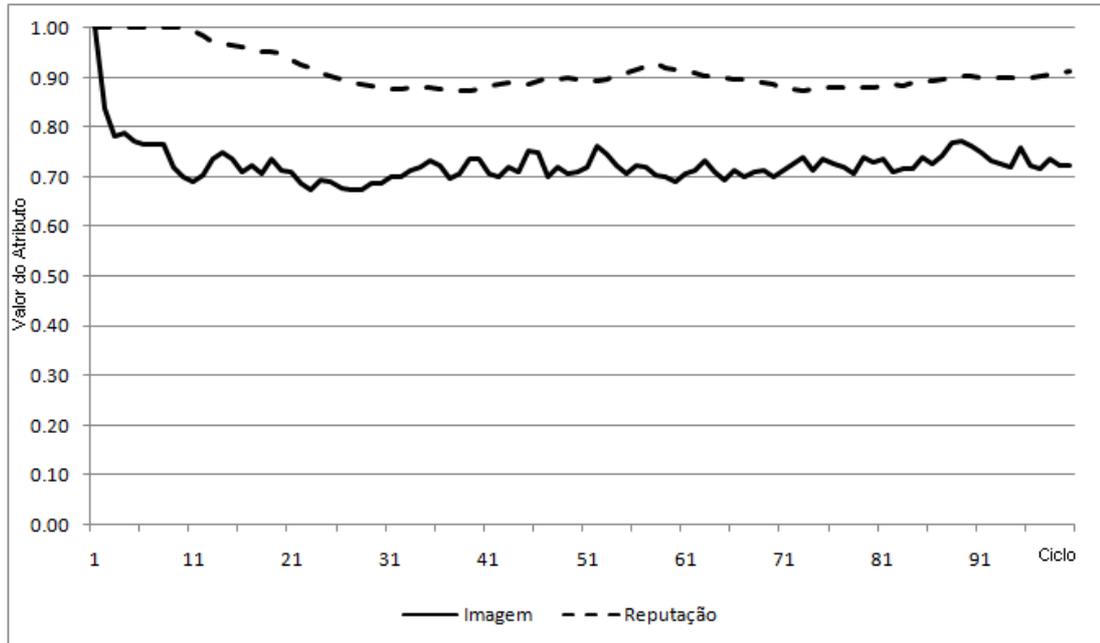


Figura 7.2: Experimento 2: ART/Repage - Valor médio da Imagem e Reputação para o agente *Desonesto*

7.4.3 Experimento 3: ART/Misto

Este experimento foi realizado com agentes que utilizam modelos de reputação heterogêneos, Repage e L.I.A.R.. Além disto, interagem sobre reputação através da transmissão de um único valor na faixa $[0, 1]$.

Este experimento subdividiu-se em dois outros experimentos dependendo do modelo de reputação utilizado pelo agente do tipo *Desonesto*.

7.4.3.1 Experimento 3.1: ART/Misto-D/L.I.A.R.

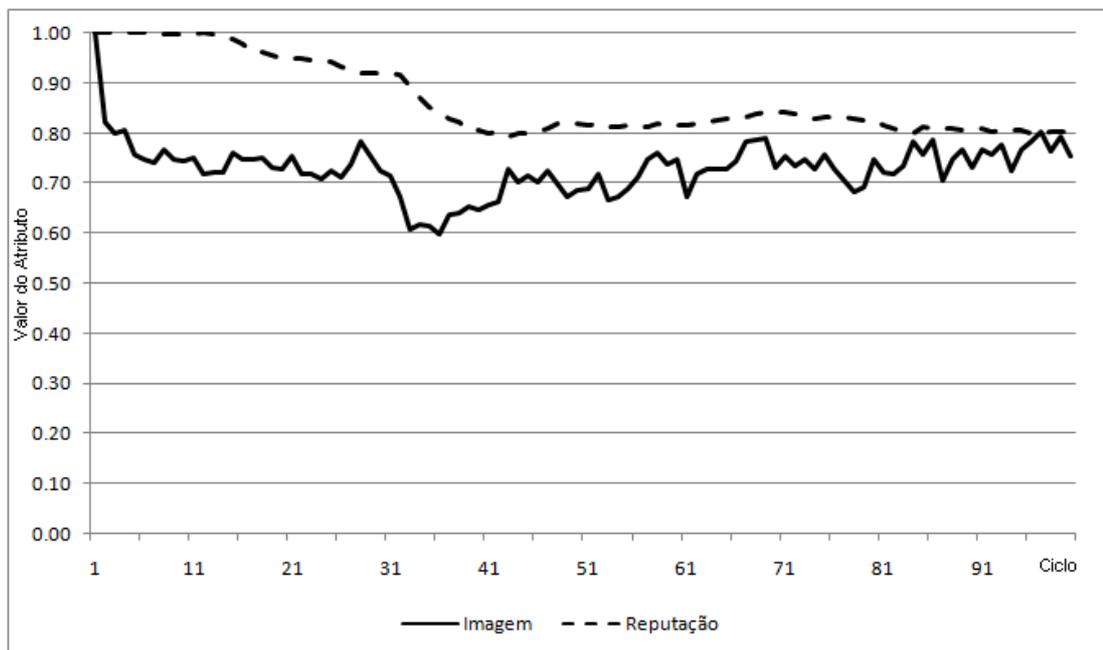
Neste experimento, a simulação é composta por 21 agentes, sendo 20 do tipo *Honesto*. Dentre eles, 10 utilizam o modelo de reputação Repage e 10 utilizam o modelo de reputação L.I.A.R.. O agente do tipo *Desonesto* utiliza o modelo de reputação L.I.A.R..

A Tabela 7.5 mostra o valor médio e desvio padrão dos atributos *Dlbrp*, *Iibrp*, *RpRcbRp*, *Imagem* e *Reputação* atribuídos pelos agentes *Honestos* ao agente *Desonesto*.

Tabela 7.5: Experimento 3.1: ART/Misto-D/L.I.A.R. - Valor dos atributos do agente *Desonesto*

Sim.	DIbRp	IIbRp	RpRcbRp	Imagem	Reputação
1	0,60 ± 0,84	0,71 ± 0,12	0,48 ± 0,12	0,78 ± 0,03	0,88 ± 0,01
2	0,54 ± 0,83	0,22 ± 0,11	0,23 ± 0,12	0,85 ± 0,02	0,17 ± 0,00
3	1,00 ± 0,00	0,92 ± 0,01	0,88 ± 0,01	0,85 ± 0,00	1,00 ± 0,00
4	0,80 ± 0,63	0,77 ± 0,04	0,58 ± 0,06	0,78 ± 0,05	0,55 ± 0,25
5	0,98 ± 0,08	0,77 ± 0,01	0,84 ± 0,01	0,84 ± 0,00	1,00 ± 0,00
6	0,73 ± 0,65	0,77 ± 0,01	0,94 ± 0,01	0,79 ± 0,04	1,00 ± 0,00
7	0,93 ± 0,22	0,57 ± 0,05	0,83 ± 0,03	0,58 ± 0,10	0,99 ± 0,01
8	1,00 ± 0,00	1,00 ± 0,00	0,88 ± 0,01	0,75 ± 0,02	1,00 ± 0,00
9	1,00 ± 0,00	1,00 ± 0,00	0,87 ± 0,02	0,78 ± 0,00	1,00 ± 0,00
10	0,53 ± 0,84	0,82 ± 0,05	0,34 ± 0,08	0,54 ± 0,13	0,44 ± 0,07
média	0,81 ± 0,20	0,75 ± 0,23	0,69 ± 0,26	0,75 ± 0,11	0,80 ± 0,30

O valor médio dos atributos dos modelos de reputação Repage e L.I.A.R. para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento é ilustrado nas Figuras 7.3 e 7.4 respectivamente.

**Figura 7.3:** Experimento 3.1: ART/Misto-D/L.I.A.R. - Valor médio da Imagem e Reputação para o agente *Desonesto*

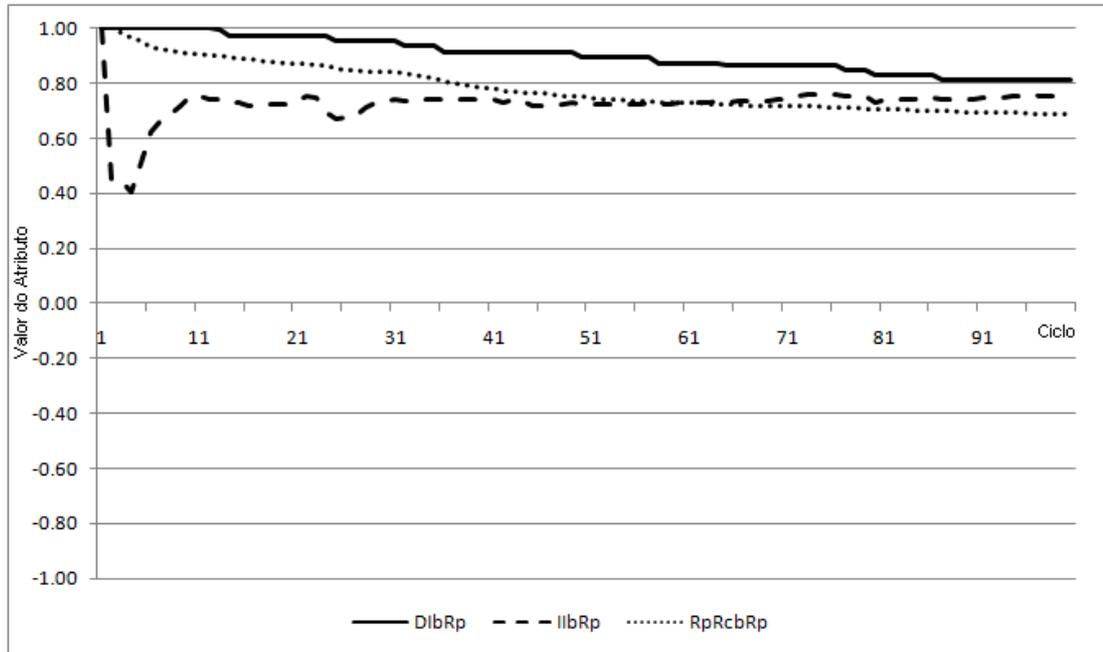


Figura 7.4: Experimento 3.1: ART/Misto-D/L.I.A.R. - Valor médio do $DibRp$, $IibRp$ e $RpRcbRp$ para o agente *Desonesto*

7.4.3.2 Experimento 3.2: ART/Misto-D/Repage

Este experimento é similar ao descrito na seção anterior, sendo que a única diferença reside no fato do agente *Desonesto* utilizar o modelo de reputação Repage.

A Tabela 7.6 mostra o valor médio e desvio padrão dos atributos $DibRp$, $IibRp$, $RpRcbRp$, *Imagem* e *Reputação* atribuídos pelos agentes *Honestos* ao agente *Desonesto*.

Tabela 7.6: Experimento 3.2: ART/Misto-D/Repage - Valor dos atributos do agente *Desonesto*

Sim.	$DibRp$	$IibRp$	$RpRcbRp$	<i>Imagem</i>	<i>Reputação</i>
1	$0,67 \pm 0,66$	$0,41 \pm 0,10$	$0,85 \pm 0,08$	$0,77 \pm 0,02$	$1,00 \pm 0,00$
2	$0,20 \pm 1,03$	$0,47 \pm 0,06$	$0,87 \pm 0,04$	$0,78 \pm 0,05$	$1,00 \pm 0,00$
3	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$0,87 \pm 0,01$	$0,71 \pm 0,03$	$0,98 \pm 0,06$
4	$1,00 \pm 0,00$	$0,95 \pm 0,01$	$0,77 \pm 0,05$	$0,72 \pm 0,03$	$0,98 \pm 0,06$
5	$0,77 \pm 0,63$	$0,77 \pm 0,02$	$0,84 \pm 0,02$	$0,75 \pm 0,03$	$0,89 \pm 0,10$
6	$1,00 \pm 0,00$	$0,87 \pm 0,00$	$0,94 \pm 0,01$	$0,95 \pm 0,02$	$1,00 \pm 0,00$
7	$0,32 \pm 0,94$	$0,30 \pm 0,08$	$0,35 \pm 0,16$	$0,64 \pm 0,05$	$0,86 \pm 0,01$
8	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$0,90 \pm 0,01$	$0,70 \pm 0,00$	$1,00 \pm 0,00$
9	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$0,86 \pm 0,01$	$0,75 \pm 0,03$	$0,98 \pm 0,06$
10	$0,40 \pm 0,97$	$0,72 \pm 0,29$	$0,76 \pm 0,21$	$0,75 \pm 0,08$	$0,90 \pm 0,21$
média	$0,74 \pm 0,32$	$0,72 \pm 0,29$	$0,76 \pm 0,21$	$0,75 \pm 0,08$	$0,90 \pm 0,21$

O valor médio dos atributos dos modelos de reputação Repage e L.I.A.R. para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento é ilustrado nas Figuras 7.5 e 7.6 respectivamente.

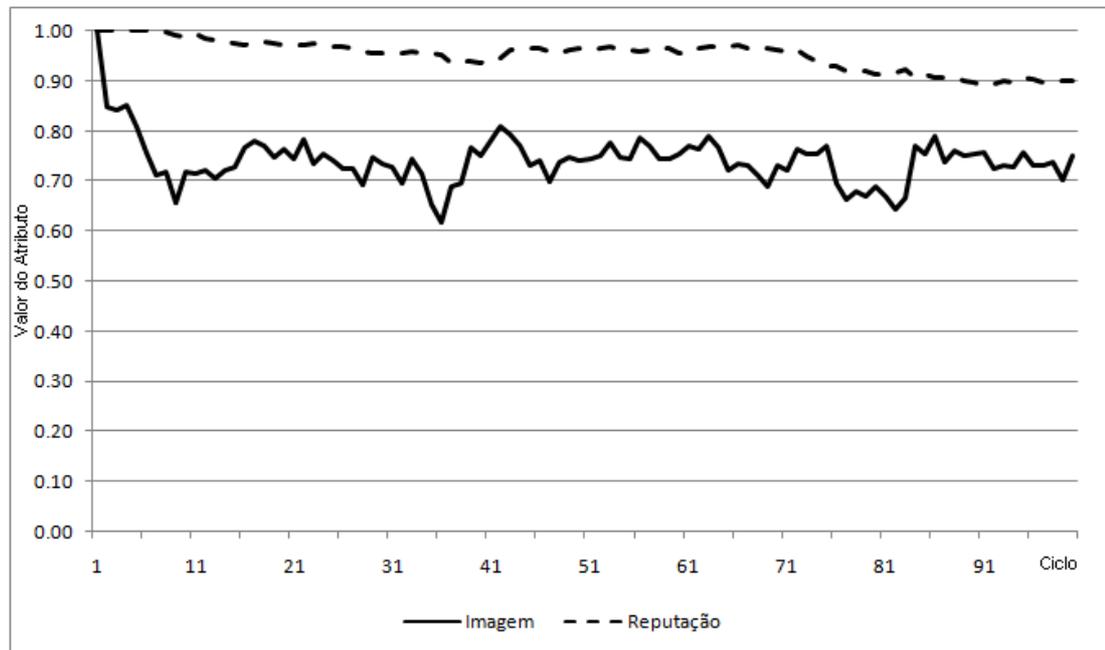


Figura 7.5: Experimento 3.2: ART/Misto-D/Repage - Valor médio da Imagem e Reputação para o agente *Desonesto*

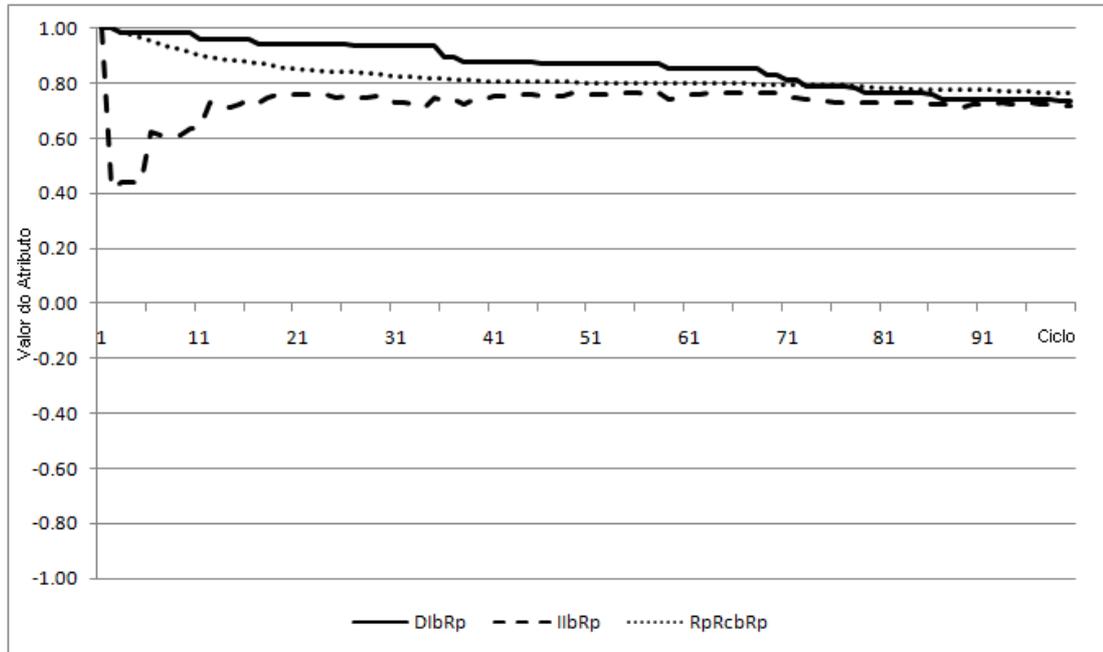


Figura 7.6: Experimento 3.2: ART/Misto-D/Repage - Valor médio do $DlbRp$, $IlbRp$ e $RpRcbRp$ para o agente *Desonesto*

7.4.4 Experimento 4: FOReART/L.I.A.R.

Este experimento foi realizado apenas com agentes homogêneos cujo modelo de reputação é o L.I.A.R.. Além disto, interage sobre reputação através da transmissão de mensagens simbólicas.

A média e o desvio padrão dos valores dos atributos $DlbRp$, $IlbRp$ e $RpRcbRp$ atribuído pelos agentes do tipo *Honesto* ao agente *Desonesto* são apresentados na Tabela 7.7.

Tabela 7.7: Experimento 4: FOReART/L.I.A.R. - Valor dos atributos do agente *Desonesto*

Simulação	$DlbRp$	$IlbRp$	$RpRcbRp$
1	$0,20 \pm 1,01$	$0,08 \pm 0,17$	$0,80 \pm 0,00$
2	$0,64 \pm 0,73$	$0,48 \pm 0,08$	$0,80 \pm 0,00$
3	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$0,80 \pm 0,00$
4	$0,37 \pm 0,93$	$0,21 \pm 0,16$	$0,80 \pm 0,00$
5	$0,87 \pm 0,46$	$0,81 \pm 0,03$	$0,80 \pm 0,00$
6	$0,64 \pm 0,73$	$0,75 \pm 0,03$	$0,80 \pm 0,00$
7	$0,50 \pm 0,89$	$0,76 \pm 0,05$	$0,80 \pm 0,00$
8	$1,00 \pm 0,00$	$1,00 \pm 0,00$	$0,80 \pm 0,00$
9	$0,90 \pm 0,45$	$0,92 \pm 0,02$	$0,80 \pm 0,00$
10	$0,66 \pm 0,73$	$0,68 \pm 0,05$	$0,80 \pm 0,00$
média	$0,68 \pm 0,27$	$0,67 \pm 0,32$	$0,80 \pm 0,00$

Na Figura 7.7 é apresentada a média dos valores dos atributos do modelo de reputação L.I.A.R. para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento.

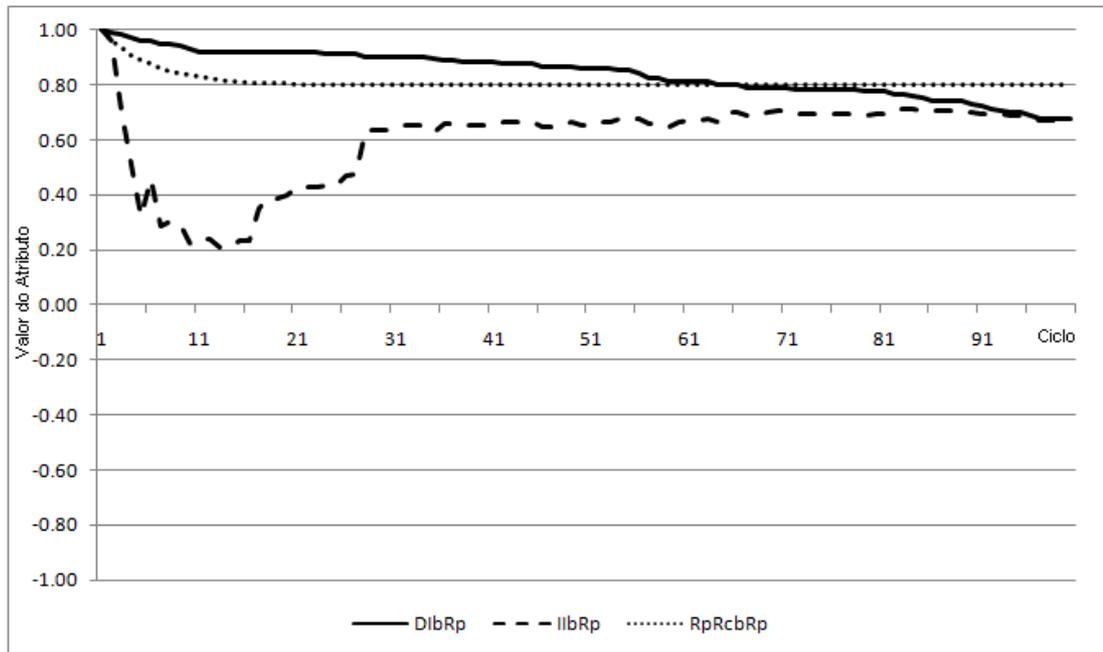


Figura 7.7: Experimento 4: FOReART/L.I.A.R. - Valor médio do *DlbRp*, *IlbRp* e *RpRcbRp* para o agente *Desonesto*

7.4.5 Experimento 5: FOReART/Repage

Este experimento foi realizado apenas com agentes homogêneos cujo modelo de reputação é o Repage. Além disto, interagem sobre reputação através da transmissão de mensagens simbólicas.

Na Tabela 7.8, são apresentados a média e o desvio padrão dos valores dos atributos *Imagem* e *Reputação* atribuídos pelos agentes do tipo *Honesto* ao agente do tipo *Desonesto*.

Na Figura 7.8 é ilustrado, nas ordenadas, o valor da média dos atributos do modelo de reputação Repage para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento, representado nas abcissas.

Tabela 7.8: Experimento 5: FOReART/Repage - Valor dos atributos do agente *Desonesto*

Simulação	Imagem	Reputação
1	0,39 ± 0,02	0,00 ± 0,00
2	0,00 ± 0,00	0,00 ± 0,00
3	0,37 ± 0,07	0,00 ± 0,00
4	0,10 ± 0,07	0,00 ± 0,00
5	0,38 ± 0,20	0,00 ± 0,00
6	0,45 ± 0,24	0,00 ± 0,00
7	0,21 ± 0,12	0,00 ± 0,00
8	0,47 ± 0,04	0,00 ± 0,00
9	0,54 ± 0,04	0,00 ± 0,00
10	0,45 ± 0,06	0,00 ± 0,00
média	0,34 ± 0,18	0,00 ± 0,00

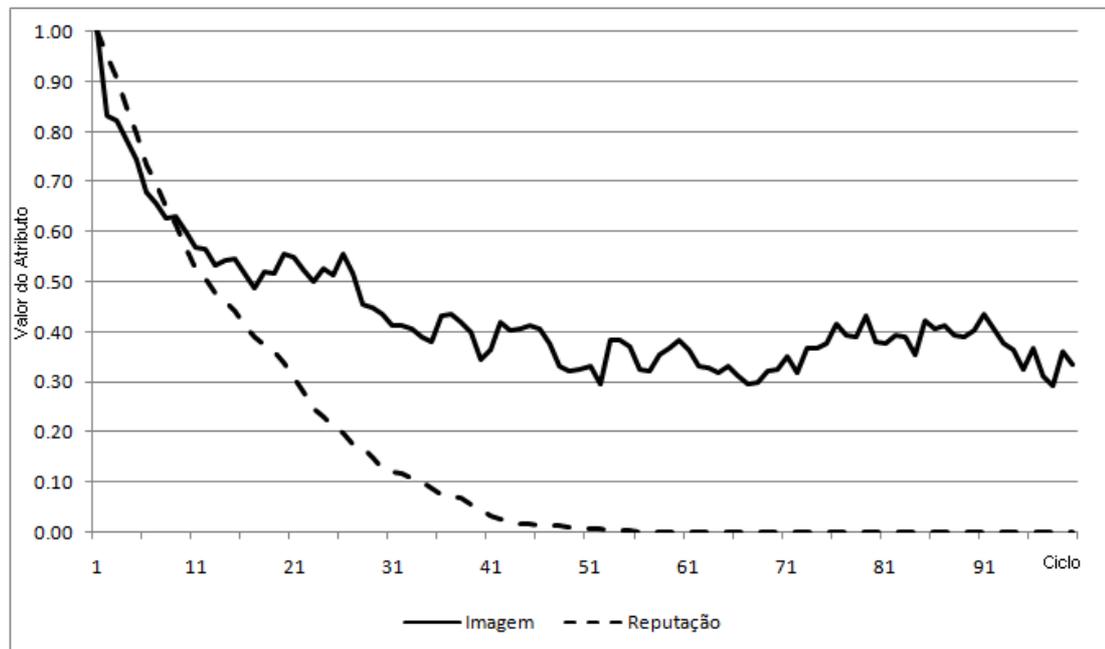


Figura 7.8: Experimento 5: FOReART/Repage - Valor médio da Imagem e Reputação para o agente *Desonesto*

7.4.6 Experimento 6: FOReART/Misto

Este experimento foi realizado com agentes que utilizam modelos de reputação heterogêneos, Repage e L.I.A.R.. Além disto, interagem sobre reputação através da transmissão de mensagens simbólicas.

Este experimento subdividiu-se em dois outros experimentos dependendo do modelo de reputação utilizado pelo agente do tipo *Desonesto*.

7.4.6.1 Experimento 6.1: FOReART/Misto-D/L.I.A.R.

Neste experimento, a simulação é composta por 21 agentes, sendo 20 do tipo *Honesto*. Dentre eles, 10 utilizam o modelo de reputação Repage e 10 utilizam o modelo de reputação L.I.A.R.. O agente do tipo *Desonesto* utiliza o modelo de reputação L.I.A.R..

A Tabela 7.9 mostra o valor médio e desvio padrão dos atributos *DibRp*, *IibRp*, *RpRcbRp*, *Imagem* e *Reputação* atribuídos pelos agentes *Honestos* ao agente *Desonesto*.

Tabela 7.9: Experimento 6.1: FOReART/Misto-D/L.I.A.R. - Valor dos atributos do agente *Desonesto*

Sim.	DibRp	IibRp	RpRcbRp	Imagem	Reputação
1	0,80 ± 0,63	0,48 ± 0,10	0,49 ± 0,05	0,71 ± 0,06	0,50 ± 0,00
2	0,60 ± 0,84	-0,04 ± 0,52	0,44 ± 0,16	0,74 ± 0,04	0,50 ± 0,00
3	1,00 ± 0,00	1,00 ± 0,00	0,41 ± 0,05	0,57 ± 0,04	0,50 ± 0,00
4	0,80 ± 0,63	0,72 ± 0,15	0,45 ± 0,01	0,55 ± 0,02	0,50 ± 0,00
5	0,60 ± 0,84	0,92 ± 0,03	0,40 ± 0,08	0,77 ± 0,02	0,50 ± 0,00
6	0,80 ± 0,63	0,89 ± 0,01	0,48 ± 0,03	0,79 ± 0,02	0,50 ± 0,00
7	0,80 ± 0,63	0,74 ± 0,04	0,43 ± 0,05	0,55 ± 0,03	0,50 ± 0,00
8	1,00 ± 0,00	1,00 ± 0,00	0,50 ± 0,04	0,66 ± 0,04	0,50 ± 0,00
9	1,00 ± 0,00	1,00 ± 0,00	0,47 ± 0,02	0,64 ± 0,05	0,50 ± 0,00
10	0,73 ± 0,64	0,34 ± 0,14	0,38 ± 0,03	0,63 ± 0,02	0,50 ± 0,00
média	0,81 ± 0,15	0,70 ± 0,35	0,44 ± 0,04	0,66 ± 0,09	0,50 ± 0,00

O valor médio dos atributos dos modelos de reputação Repage e L.I.A.R. dos agentes *Honestos* para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento é ilustrado nas Figuras 7.9 e 7.10 respectivamente.

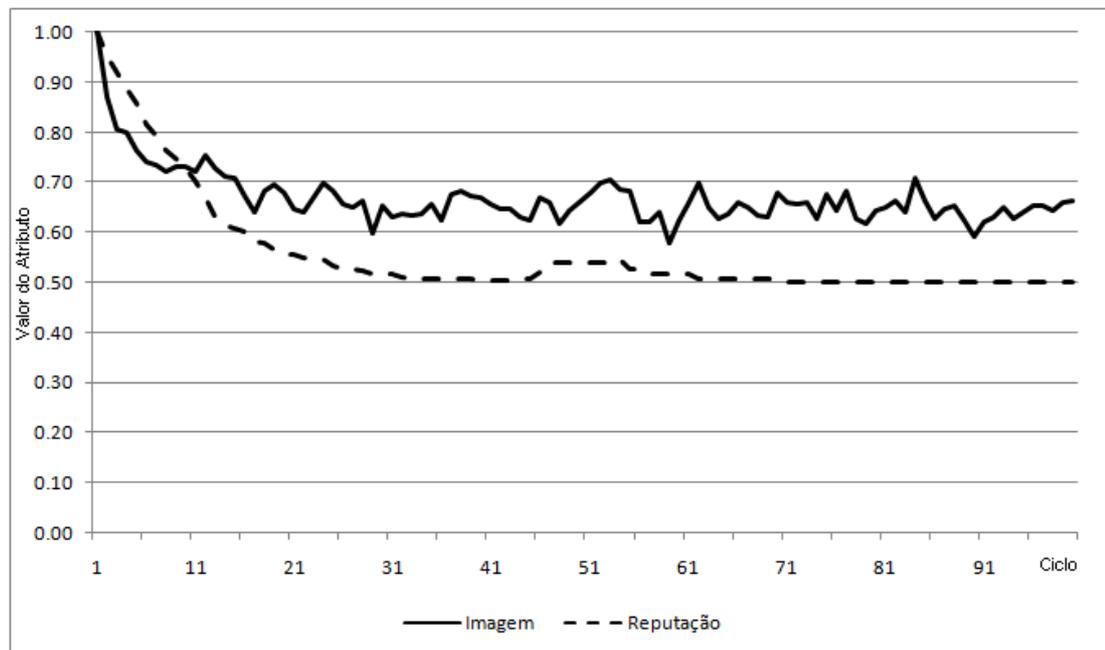


Figura 7.9: Experimento 6.1: FOrEART/Misto-D/L.I.A.R. - Valor médio da Imagem e Reputação para o agente *Desonesto*

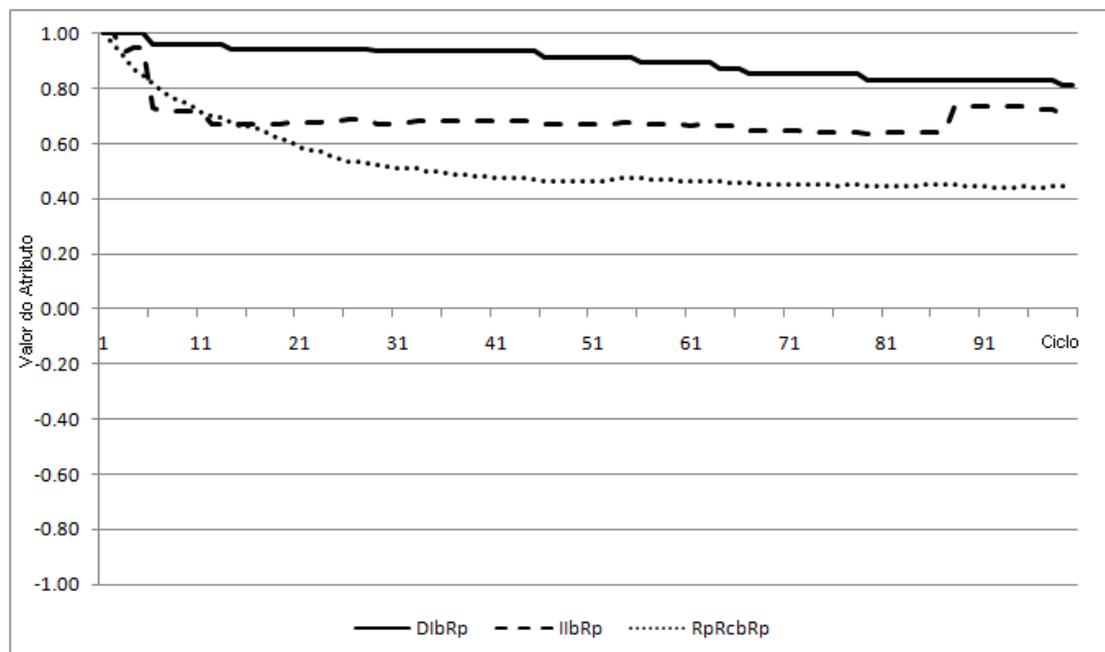


Figura 7.10: Experimento 6.1: FOrEART/Misto-D/L.I.A.R. - Valor médio do *DIBRp*, *IIBRp* e *RpRcbRp* para o agente *Desonesto*

7.4.6.2 Experimento 6.2: FOReART/Misto-D/Repage

Este experimento é similar ao descrito na seção anterior, sendo que a única diferença reside no fato do agente *Desonesto* utilizar o modelo de reputação Repage.

A Tabela 7.10 mostra o valor médio e desvio padrão dos atributos *Dlbrp*, *Iibrp*, *RpRcbRp*, *Imagem* e *Reputação* atribuídos pelos agentes *Honestos* ao agente *Desonesto*.

Tabela 7.10: Experimento 6.2: FOReART/Misto-D/Repage - Valor dos atributos do agente *Desonesto*

Sim.	Dlbrp	Iibrp	RpRcbRp	Imagem	Reputação
1	0,40 ± 0,97	0,37 ± 0,10	0,43 ± 0,11	0,72 ± 0,02	0,50 ± 0,00
2	0,60 ± 0,84	0,46 ± 0,04	0,51 ± 0,05	0,69 ± 0,05	0,50 ± 0,00
3	1,00 ± 0,00	1,00 ± 0,00	0,40 ± 0,04	0,65 ± 0,06	0,50 ± 0,00
4	0,45 ± 0,80	0,51 ± 0,06	0,47 ± 0,02	0,77 ± 0,02	0,50 ± 0,00
5	0,40 ± 0,97	0,26 ± 0,15	0,47 ± 0,04	0,67 ± 0,03	0,50 ± 0,00
6	0,60 ± 0,84	0,00 ± 0,38	0,48 ± 0,03	0,69 ± 0,05	0,50 ± 0,00
7	0,80 ± 0,63	0,64 ± 0,06	0,44 ± 0,04	0,70 ± 0,02	0,50 ± 0,00
8	1,00 ± 0,00	1,00 ± 0,00	0,55 ± 0,04	0,67 ± 0,02	0,50 ± 0,00
9	1,00 ± 0,00	1,00 ± 0,00	0,29 ± 0,12	0,54 ± 0,02	0,25 ± 0,00
10	0,94 ± 0,18	0,64 ± 0,08	0,40 ± 0,03	0,56 ± 0,04	0,50 ± 0,00
média	0,72 ± 0,26	0,59 ± 0,34	0,44 ± 0,07	0,67 ± 0,07	0,48 ± 0,08

O valor médio dos atributos dos modelos de reputação Repage e L.I.A.R. dos agentes *Honestos* para o agente *Desonesto* para cada ciclo das simulações realizadas neste experimento é ilustrado nas Figuras 7.11 e 7.12 respectivamente.

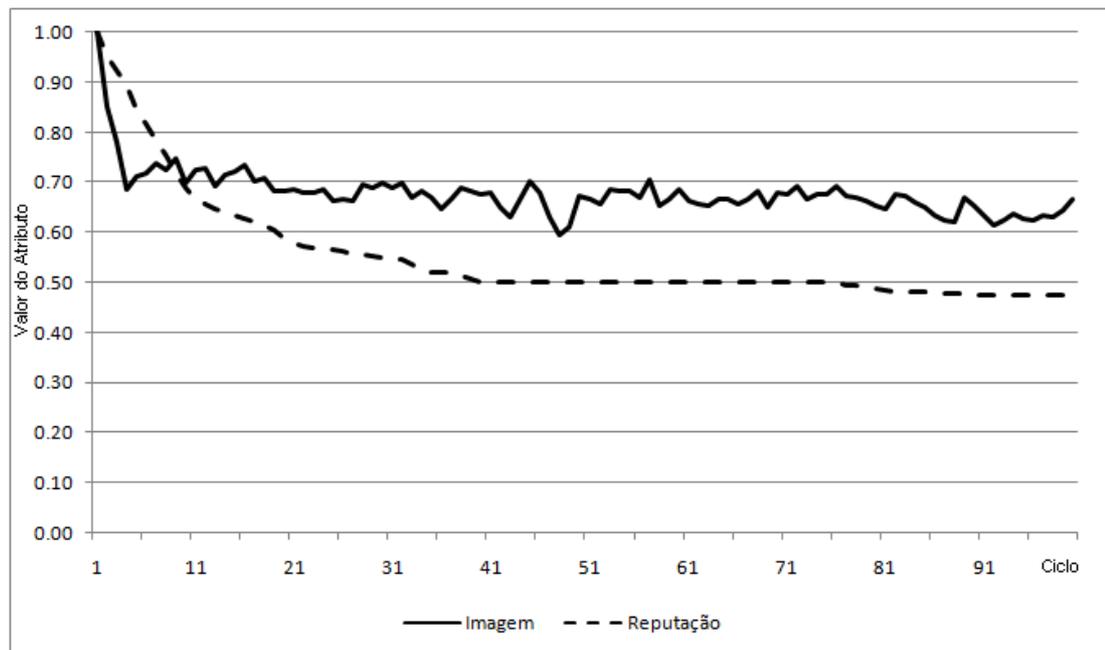


Figura 7.11: Experimento 6.2: FOrEART/Misto-D/Repage - Valor médio da Imagem e Reputação para o agente *Desonesto*

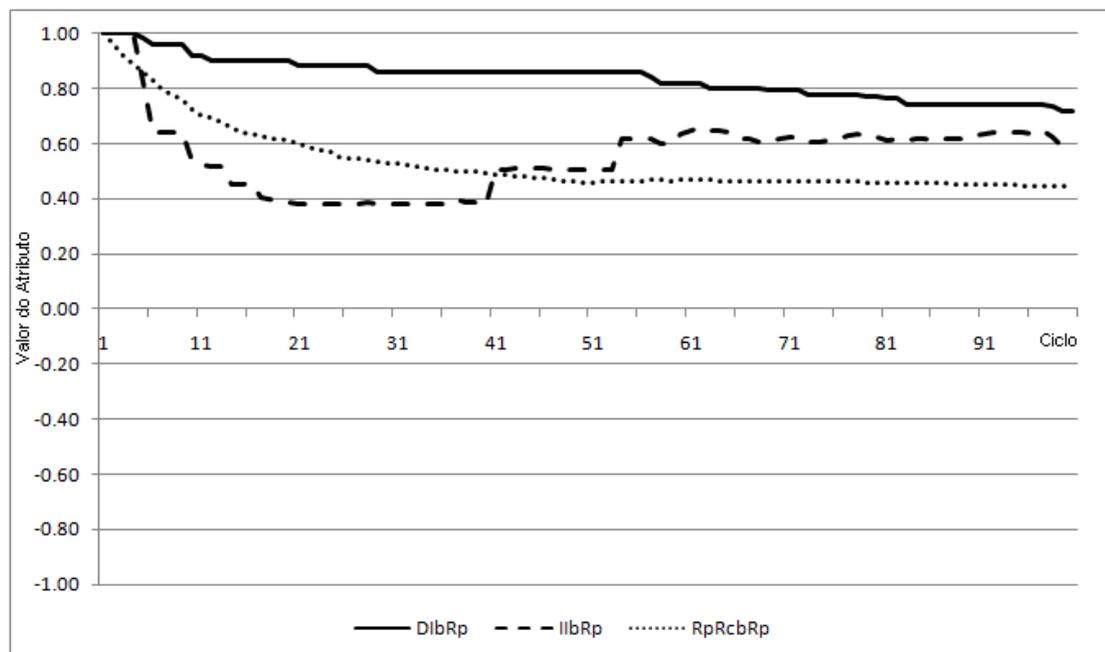


Figura 7.12: Experimento 6.2: FOrEART/Misto-D/Repage - Valor médio do $DIBRp$, $IIBRp$ e $RpRcbRp$ para o agente *Desonesto*

7.5 Análise dos Resultados

Nesta seção, são apresentadas as análises dos resultados obtidos na seção 7.4. Primeiramente, na seção 7.5.1 é apresentada a metodologia adotada para análise dos resultados. Em seguida, nas seções 7.5.2 e 7.5.3, são apresentadas as análises visando responder às questões levantadas na seção 1.2.

7.5.1 Metodologia de Análise

A metodologia utilizada para análise dos resultados obtidos nos experimentos descritos na seção 7.4 baseia-se em um processo de hipóteses estatísticas denominado *teste de hipóteses* (BOSLAUGH; WATTERS, 2008). O processo de teste de hipóteses provê uma regra de decisão usada para aceitar ou rejeitar uma hipótese estatística com base em elementos amostrais. Para tanto, formula-se duas hipóteses denominadas *hipótese nula* (H_0) e *hipótese alternativa* (H_1). Normalmente a hipótese alternativa representa a suposição que se deseja provar, enquanto a hipótese nula é formulada com o propósito de ser rejeitada. Conseqüentemente, ao se rejeitar a hipótese nula, aceita-se a hipótese alternativa.

Neste trabalho adota-se o teste de hipóteses não-paramétrico denominado *Teste da Soma das Ordens* de Wilcoxon (*Wilcoxon's Rank Sum Test*) (BOSLAUGH; WATTERS, 2008). Esta escolha baseou-se no fato de nem todos os dados a serem testados terem distribuição normal, conforme apresentado no apêndice B.

Para o cálculo do *Teste da Soma das Ordens*, considere dois conjuntos de valores, denominados A e B , a serem comparados. Primeiramente, requer-se que os valores destes dois conjuntos sejam unidos e ordenados do menor para o maior valor. Em seguida, para cada valor é atribuído um número de posição, assim, ao menor valor atribui-se o número de posição 1, ao segundo menor valor atribui-se o número de posição 2 e assim sucessivamente. Caso haja valores repetidos, o número de suas posições são somados e o resultado da soma dividido pelo número de repetições existentes, sendo o número resultante deste cálculo atribuído a cada uma das repetições. Finalizada a atribuição do número das posições, realiza-se a somatória dos números da posição dos valores de cada um dos grupos, que correspondem respectivamente a W_A e W_B . A partir do menor valor entre W_A e W_B , calcula-se o valor de Z para determinar se a diferença entre os dois grupos é estatisticamente significativa. Z é calculado pela expressão:

$$z = \frac{W - \mu_W}{\sigma_W} \quad (7.1)$$

onde, μ_W e σ_W representam respectivamente a soma esperada e o desvio padrão de W , cujas expressões são:

$$\mu_W = \frac{n_A(n_A + n_B + 1)}{2} \quad (7.2)$$

$$\sigma_W = \sqrt{\frac{n_A n_B (n_A + n_B + 1)}{12}} \quad (7.3)$$

Utiliza-se então o valor calculado de Z para obter o valor de p através da tabela de distribuição normal. Caso o valor obtido de p seja menor ou igual ao valor de significância (α) estipulado, H_0 é rejeitado, aceitando-se H_1 . Em caso contrário, H_0 não é rejeitado.

7.5.2 Efeito da expressividade na comunicação

Para a análise dos resultados com relação ao efeito da expressividade na comunicação dos agentes, são verificados se os valores dos atributos dos experimentos que se comunicam numericamente (sem expressividade) são maiores do que os que se comunicam simbolicamente (com expressividade). Assim, deseja-se comparar os atributos obtidos nos pares de experimentos (1, 4), (2, 5) e (3, 6). No entanto, como os experimentos 3 e 6 se subdividem em 2 experimentos cada, conforme o tipo da reputação do agente *Desonesto*, o último par de experimentos na verdade se subdivide em 4 pares, (3.1, 6.1), (3.1, 6.2), (3.2, 6.1) e (3.2, 6.2), que correspondem a todas às possíveis combinações entre os experimentos 3 e 6.

Para tanto, assume-se que quanto maior o valor do atributo, maior é a reputação de um agente perante a outro. Assim, como se deseja demonstrar que uma comunicação mais expressiva possibilita uma melhor exatidão na identificação da reputação de agentes, as seguintes hipóteses são formuladas:

Hipótese A O valor do atributo *Dlbrp* do modelo de reputação L.I.A.R. do experimento ART é maior que o valor do mesmo atributo do experimento FOReART. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART}^{Dlbrp} > Q_{FOReART}^{Dlbrp}$$

Para validar a Hipótese A, realiza-se o teste:

$$H_0: Q_{ART}^{Dlbrp} \leq Q_{FOReART}^{Dlbrp}$$

$$H_1: Q_{ART}^{Dlbrp} > Q_{FOReART}^{Dlbrp}$$

Hipótese B O valor do atributo *IbRp* do modelo de reputação L.I.A.R. do experimento ART é maior que o valor do mesmo atributo do experimento FOReART. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART}^{IbRp} > Q_{FOReART}^{IbRp}$$

Para validar a Hipótese B, realiza-se o teste:

$$H0: Q_{ART}^{IbRp} \leq Q_{FOReART}^{IbRp}$$

$$H1: Q_{ART}^{IbRp} > Q_{FOReART}^{IbRp}$$

Hipótese C O valor do atributo *RpRcbRp* do modelo de reputação L.I.A.R. do experimento ART é maior que o valor do mesmo atributo do experimento FOReART. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART}^{RpRcbRp} > Q_{FOReART}^{RpRcbRp}$$

Para validar a Hipótese C, realiza-se o teste:

$$H0: Q_{ART}^{RpRcbRp} \leq Q_{FOReART}^{RpRcbRp}$$

$$H1: Q_{ART}^{RpRcbRp} > Q_{FOReART}^{RpRcbRp}$$

Hipótese D O valor do atributo *Imagem* do modelo de reputação Repage do experimento ART é maior que o valor do mesmo atributo do experimento FOReART. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART}^{Imagem} > Q_{FOReART}^{Imagem}$$

Para validar a Hipótese D, realiza-se o teste:

$$H0: Q_{ART}^{Imagem} \leq Q_{FOReART}^{Imagem}$$

$$H1: Q_{ART}^{Imagem} > Q_{FOReART}^{Imagem}$$

Hipótese E O valor do atributo *Reputação* do modelo de reputação Repage do experimento ART é maior que o valor do mesmo atributo do experimento FOReART. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART}^{Reputacao} > Q_{FOReART}^{Reputacao}$$

Para validar a Hipótese E, realiza-se o teste:

$$H0: Q_{ART}^{Reputacao} \leq Q_{FOReART}^{Reputacao}$$

$$H1: Q_{ART}^{Reputacao} > Q_{FOReART}^{Reputacao}$$

Como nem todas as combinações de experimentos citadas anteriormente geram valores para todos os atributos de reputação dos modelos Repage e L.I.A.R., algumas das hipóteses não se aplicam a algumas das combinações. Na Tabela 7.11 são apresentadas as relações entre par de experimentos e hipóteses.

Tabela 7.11: Hipóteses utilizadas para verificar o efeito da expressividade de comunicação

Pares	Experimentos	Hipóteses
(1, 4)	ART/L.I.A.R. FOReART/L.I.A.R.	A, B, C
(2, 5)	ART/Repage FOReART/Repage	D, E
(3.1, 6.1)	ART/Misto-D/L.I.A.R. FOReART/Misto-D/L.I.A.R.	A, B, C, D, E
(3.1, 6.2)	ART/Misto-D/L.I.A.R. FOReART/Misto-D/Repage	A, B, C, D, E
(3.2, 6.1)	ART/Misto-D/Repage FOReART/Misto-D/L.I.A.R.	A, B, C, D, E
(3.2, 6.2)	ART/Misto-D/Repage FOReART/Misto-D/Repage	A, B, C, D, E

Realizando os cálculos descritos na seção 7.5.1 para cada uma das combinações de experimentos, obtêm-se os valores de p apresentados na Tabela 7.12.

Tabela 7.12: Valores de p para verificar o efeito da expressividade de comunicação

Pares	Hipótese				
	A	B	C	D	E
(1, 4)	0,20	0,36	0,00	n/a	n/a
(2, 5)	n/a	n/a	n/a	0,00	0,00
(3.1, 6.1)	0,59	0,51	0,02	0,02	0,01
(3.1, 6.2)	0,26	0,14	0,01	0,01	0,01
(3.2, 6.1)	0,59	0,52	0,01	0,02	0,00
(3.2, 6.2)	0,40	0,27	0,01	0,01	0,00

Com base em um teste ao nível de significância (α) de 0,05, H_0 é rejeitado se o valor de p for inferior a 0,05.

Analisando a Tabela 7.12, verifica-se que os valores p das hipóteses C, D e E são inferiores ao limite de 0,05. Porém, o mesmo não ocorre para os valores das hipóteses A e B, que não possuem nenhum de seus valores inferior a este limite.

Do ponto de vista dos modelos de reputação, as hipóteses D e E estão associadas aos atributos *Imagem* e *Reputação* do modelo de reputação *Repage*, enquanto as hipóteses A, B e C estão associadas aos atributos *Dlbrp*, *Iibrp* e *RpRcbRp* do modelo de reputação *L.I.A.R.*

Considerando as características dos experimentos, os agentes com modelos de reputação *Repage* atualizam ambos os atributos de seu modelo durante a comunicação, enquanto os agentes com modelos de reputação *L.I.A.R.* atualizam somente o atributo *RpRcbRp*. Os atributos *Dlbrp* e *Iibrp* são atualizados somente com informações observadas pelo agente, e deste modo, eles não são influenciados pela comunicação.

Assim, pode-se concluir que uma maior expressividade na comunicação sobre reputação tem um impacto estatisticamente significativo nos agentes em ambos os modelos, aumentando assim a exatidão na identificação de agentes desonestos.

7.5.3 Efeito da heterogeneidade de modelos de reputação

Para a análise dos resultados do efeito da heterogeneidade de modelos de reputação na exatidão da reputação de agentes, são verificados se os valores dos atributos dos experimentos com modelos de reputação mistos são maiores do que os dos experimentos com apenas um modelo de reputação. Assim, deseja-se comparar os atributos obtidos nos pares de experimentos (1, 3), (2, 3), (4, 6) e (5, 6). No entanto, como os experimentos 3 e 6 se subdividem em 2, os pares expandidos de experimentos são (1, 3.1), (1, 3.2), (2, 3.1), (2, 3.2) (4, 6.1), (4, 6.2), (5, 6.1) e (5, 6.2).

Com o objetivo de facilitar a apresentação das análises, as mesmas serão apresentadas em duas seções a seguir.

7.5.3.1 Efeito da heterogeneidade na plataforma ART

Nesta seção, é apresentada a análise do impacto da heterogeneidade dos pares de experimentos da plataforma ART. Os pares de experimentos que se enquadram nesta classificação são (1, 3.1), (1, 3.2), (2, 3.1) e (2, 3.2). Para a análise destes pares de experimentos são formuladas as seguintes hipóteses:

Hipótese F O valor do atributo *Dlbrp* do modelo de reputação *L.I.A.R.* do experimento ART/*L.I.A.R.* é maior que o valor do mesmo atributo do experimento ART/*Misto*. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART/L.I.A.R.}^{DIbRp} > Q_{ART/Misto}^{DIbRp}$$

Para validar a Hipótese F, realiza-se o teste:

$$H0: Q_{ART/L.I.A.R.}^{DIbRp} \leq Q_{ART/Misto}^{DIbRp}$$

$$H1: Q_{ART/L.I.A.R.}^{DIbRp} > Q_{ART/Misto}^{DIbRp}$$

Hipótese G O valor do atributo *IbRp* do modelo de reputação L.I.A.R. do experimento ART/L.I.A.R. é maior que o valor do mesmo atributo do experimento ART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART/L.I.A.R.}^{IbRp} > Q_{ART/Misto}^{IbRp}$$

Para validar a Hipótese G, realiza-se o teste:

$$H0: Q_{ART/L.I.A.R.}^{IbRp} \leq Q_{ART/Misto}^{IbRp}$$

$$H1: Q_{ART/L.I.A.R.}^{IbRp} > Q_{ART/Misto}^{IbRp}$$

Hipótese H O valor do atributo *RpRcbRp* do modelo de reputação L.I.A.R. do experimento ART/L.I.A.R. é maior que o valor do mesmo atributo do experimento ART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART/L.I.A.R.}^{RpRcbRp} > Q_{ART/Misto}^{RpRcbRp}$$

Para validar a Hipótese H, realiza-se o teste:

$$H0: Q_{ART/L.I.A.R.}^{RpRcbRp} \leq Q_{ART/Misto}^{RpRcbRp}$$

$$H1: Q_{ART/L.I.A.R.}^{RpRcbRp} > Q_{ART/Misto}^{RpRcbRp}$$

Hipótese I O valor do atributo *Imagem* do modelo de reputação Repage do experimento ART/Repage é maior que o valor do mesmo atributo do experimento ART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART/Repage}^{Imagem} > Q_{ART/Misto}^{Imagem}$$

Para validar a Hipótese I, realiza-se o teste:

$$H0: Q_{ART/Repage}^{Imagem} \leq Q_{ART/Misto}^{Imagem}$$

$$H1: Q_{ART/Repage}^{Imagem} > Q_{ART/Misto}^{Imagem}$$

Hipótese J O valor do atributo *Reputação* do modelo de reputação Repage do experimento ART/Repage é maior que o valor do mesmo atributo do experimento ART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{ART/Repage}^{Reputacao} > Q_{ART/Misto}^{Reputacao}$$

Para validar a Hipótese J, realiza-se o teste:

$$H0: Q_{ART/Repage}^{Reputacao} \leq Q_{ART/Misto}^{Reputacao}$$

$$H1: Q_{ART/Repage}^{Reputacao} > Q_{ART/Misto}^{Reputacao}$$

A relação entre pares de experimentos e hipóteses é mostrada na Tabela 7.13.

Tabela 7.13: Hipóteses para verificar o efeito da heterogeneidade na plataforma ART

Pares	Experimentos	Hipóteses
(1, 3.1)	ART/L.I.A.R. ART/Misto-D/L.I.A.R	F, G, H
(1, 3.2)	ART/L.I.A.R ART/Misto-D/Repage	F, G, H
(2, 3.1)	ART/Repage ART/Misto-D/L.I.A.R.	I, J
(2, 3.2)	ART/Repage ART/Misto-D/Repage	I, J

Realizando os cálculos descritos na seção 7.5.1 para cada uma das combinações de experimentos, obtêm-se os valores de p apresentados na Tabela 7.14.

Tabela 7.14: Valores de p para verificar o efeito da heterogeneidade na plataforma ART

Pares	Hipótese				
	F	G	H	I	J
(1, 3.1)	0,64	0,54	0,00	n/a	n/a
(1, 3.2)	0,55	0,43	0,00	n/a	n/a
(2, 3.1)	n/a	n/a	n/a	0,76	0,31
(2, 3.2)	n/a	n/a	n/a	0,82	0,42

Com base em um teste ao nível de significância (α) de 0,05, $H0$ é rejeitado se o valor de p for inferior a 0,05.

Analisando a Tabela 7.14, verifica-se que os valores p da hipótese H são inferiores ao limite de 0,05. Porém, o mesmo não ocorre para os valores p das hipóteses F, G, I e J, que não possuem nenhum valor inferior a este limite.

Do ponto de vista dos modelos de reputação, as hipóteses F e G estão associadas aos atributos *Imagem* e *Reputação* do modelo de reputação Repage, enquanto

as hipóteses H, I e J estão associadas aos atributos $Dlbrp$, $Iibrp$ e $RpRcbRp$ do modelo de reputação L.I.A.R. Como já discutido na seção 7.5.2, os atributos $Dlbrp$ e $Iibrp$ não são influenciados pela comunicação dos agentes, já que são obtidos por experiência direta. Assim, pode-se concluir que a heterogeneidade de modelos de reputação tem um impacto estatisticamente significativo nos agentes que usam o modelo de reputação L.I.A.R.

7.5.3.2 Efeito da heterogeneidade na plataforma FOReART

Já nesta seção, é apresentada a análise do impacto da heterogeneidade dos pares de experimentos da plataforma FOReART. Os pares de experimentos que se enquadram nesta classificação são (4, 6.1), (4, 6.2), (5, 6.1) e (5, 6.2). Para a análise destes pares de experimentos são formuladas as seguintes hipóteses:

Hipótese K O valor do atributo $Dlbrp$ do modelo de reputação L.I.A.R. do experimento FOReART/L.I.A.R. é maior que o valor do mesmo atributo do experimento FOReART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{FOReART/L.I.A.R.}^{Dlbrp} > Q_{FOReART/Misto}^{Dlbrp}$$

Para validar a Hipótese K, realiza-se o teste:

$$H0: Q_{FOReART/L.I.A.R.}^{Dlbrp} \leq Q_{FOReART/Misto}^{Dlbrp}$$

$$H1: Q_{FOReART/L.I.A.R.}^{Dlbrp} > Q_{FOReART/Misto}^{Dlbrp}$$

Hipótese L O valor do atributo $Iibrp$ do modelo de reputação L.I.A.R. do experimento FOReART/L.I.A.R. é maior que o valor do mesmo atributo do experimento FOReART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{FOReART/L.I.A.R.}^{Iibrp} > Q_{FOReART/Misto}^{Iibrp}$$

Para validar a Hipótese L, realiza-se o teste:

$$H0: Q_{FOReART/L.I.A.R.}^{Iibrp} \leq Q_{FOReART/Misto}^{Iibrp}$$

$$H1: Q_{FOReART/L.I.A.R.}^{Iibrp} > Q_{FOReART/Misto}^{Iibrp}$$

Hipótese M O valor do atributo $RpRcbRp$ do modelo de reputação L.I.A.R. do experimento FOReART/L.I.A.R. é maior que o valor do mesmo atributo do experimento FOReART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{FOReART/L.I.A.R.}^{RpRcbRp} > Q_{FOReART/Misto}^{RpRcbRp}$$

Para validar a Hipótese H, realiza-se o teste:

$$H0: Q_{FOReART/L.I.A.R.}^{RpRcbRp} \leq Q_{FOReART/Misto}^{RpRcbRp}$$

$$H1: Q_{FOReART/L.I.A.R.}^{RpRcbRp} > Q_{FOReART/Misto}^{RpRcbRp}$$

Hipótese N O valor do atributo *Imagem* do modelo de reputação Repage do experimento FOReART/Repage é maior que o valor do mesmo atributo do experimento FOReART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{FOReART/Repage}^{Imagem} > Q_{FOReART/Misto}^{Imagem}$$

Para validar a Hipótese N, realiza-se o teste:

$$H0: Q_{FOReART/Repage}^{Imagem} \leq Q_{FOReART/Misto}^{Imagem}$$

$$H1: Q_{FOReART/Repage}^{Imagem} > Q_{FOReART/Misto}^{Imagem}$$

Hipótese O O valor do atributo *Reputação* do modelo de reputação Repage do experimento FOReART/Repage é maior que o valor do mesmo atributo do experimento FOReART/Misto. Esta hipótese, do ponto de vista do atributo do modelo de reputação, é expressa matematicamente como segue:

$$Q_{FOReART/Repage}^{Reputacao} > Q_{FOReART/Misto}^{Reputacao}$$

Para validar a Hipótese O, realiza-se o teste:

$$H0: Q_{FOReART/Repage}^{Reputacao} \leq Q_{FOReART/Misto}^{Reputacao}$$

$$H1: Q_{FOReART/Repage}^{Reputacao} > Q_{FOReART/Misto}^{Reputacao}$$

A relação entre pares de experimentos e hipóteses é mostrada na Tabela 7.15.

Tabela 7.15: Hipóteses para verificar o efeito da heterogeneidade na plataforma FOReART

Pares	Experimentos	Hipóteses
(4, 6.1)	FOReART/L.I.A.R. FOReART/Misto L.I.A.R	K, L, M
(4, 6.2)	FOReART/L.I.A.R FOReART/Misto Repage	K, L, M
(5, 6.1)	FOReART/Repage FOReART/Misto L.I.A.R.	N, O
(5, 6.2)	FOReART/Repage FOReART/Misto Repage	N, O

Realizando os cálculos descritos na seção 7.5.1 para cada uma das combinações de experimentos, obtêm-se os valores de p apresentados na Tabela 7.16.

Tabela 7.16: Valores de p para verificar o efeito da heterogeneidade na plataforma FOReART

Pares	Hipótese				
	K	L	M	N	O
(4, 6.1)	0,84	0,60	0,00	n/a	n/a
(4, 6.2)	0,60	0,29	0,00	n/a	n/a
(5, 6.1)	n/a	n/a	n/a	1,00	1,00
(5, 6.2)	n/a	n/a	n/a	1,00	1,00

Com base em um teste ao nível de significância (α) de 0,05, H_0 é rejeitado se o valor de p for inferior a 0,05. Pelos valores apresentados na Tabela 7.16, não há como rejeitar H_0 para as hipóteses K, L, N e O. No entanto, é possível rejeitar H_0 para a hipótese M, a qual está relacionada com o atributo $RpRcbRp$. Conforme discutido nas análises anteriores (seções 7.5.2 e 7.5.3.1), este atributo é o único do modelo de reputação L.I.A.R. influenciado pela comunicação sobre reputação. Portanto, reforça-se a conclusão de que a heterogeneidade de modelos de reputação tem um impacto estatisticamente significativo nos agentes que usam o modelo de reputação L.I.A.R.

Portanto, pode-se presumir que existam características intrínsecas, ou de implementação, do modelo de reputação L.I.A.R. que possibilitem o usufruto da maior expressividade na comunicação. Uma análise mais aprofundada deste aspecto está fora do escopo deste trabalho.

7.6 Considerações Finais

Os diversos resultados obtidos permitem concluir que existe uma melhora na identificação do agente *Desonesto* em todos os pares de experimentos quando se considera uma maior expressividade de comunicação. Entretanto, somente foi possível constatar a melhora de identificação do agente desonesto com modelos heterogêneos de reputação no caso de agentes com modelo de reputação L.I.A.R.

8 Conclusões

Este trabalho propôs a arquitetura SOARI com a finalidade de possibilitar que agentes com modelos de reputação heterogêneos pudessem interoperar. Para tal, estendeu-se a arquitetura genérica de agente para interação de reputação proposta por Vercouter et al. (2007). Esta extensão está embasada na abordagem SOA e divide o *Módulo de Mapeamento de Reputação* (MMR) da proposta original em dois módulos: o *Ontology Mapping Service* (SMO) e o módulo TRADUTOR.

O SMO é um serviço que reside externamente aos agentes e desempenha as funções de *mapeamento* e *tradução* de conceitos da *Ontologia do Modelo de Reputação* para a *Ontologia Comum no Domínio da Reputação* e vice-versa. Esta característica possibilitou a redução no consumo de recursos pelos agentes e a reutilização por vários agentes dos mapeamentos existentes.

O módulo TRADUTOR desempenha diversas funções dentre as quais a principal é o tratamento de *Problemas de Tradução*. Este módulo deve ser integrado ao *Módulo de Interação* e ao *Módulo de Modelo de Reputação* dos agentes. No entanto, devido à sua flexibilidade e configurabilidade, esta adaptação e integração se tornam uma atividade simples. Esta simplicidade é uma das vantagens desta abordagem, já que possibilita aos agentes legados usufruírem, sem muitas alterações, dos benefícios da arquitetura proposta.

Um cenário simples de comércio eletrônico foi usado na demonstração do funcionamento da arquitetura SOARI. Neste cenário, a seqüência de ativações dos componentes da arquitetura, bem como dos módulos dos agentes, foi apresentada.

Por fim, diversos experimentos foram realizados nas plataformas ART e FO-ReART, que possibilitaram responder as questões apresentadas na seção 1.2:

1. Existe uma melhora de exatidão na identificação dos agentes desonestos por agentes que utilizam uma maior expressividade em suas interações sobre reputação?

Analisando os resultados gerados pelo *Teste da Soma das Ordens* de Wilcoxon, verificou-se uma melhora de exatidão na identificação da reputação

do agente desonesto em todos os experimentos realizados para ambos os modelos de reputação Repage e L.I.A.R.

2. Pode-se afirmar que sociedades compostas por agentes com modelos de reputação distintos têm uma melhora de exatidão na identificação dos agentes desonestos em relação as sociedades homogêneas?

Analisando os resultados gerados pelo pelo *Teste da Soma das Ordens* de Wilcoxon, verificou-se uma melhora de precisão na identificação da reputação quando o agente utiliza o modelo de reputação L.I.A.R. Entretanto, tal conclusão não pôde ser inferida quando esta mesma análise foi realizada para os agentes que utilizam o modelo de reputação Repage. Isto provavelmente sugere uma relação entre características intrínsecas, ou de implementação, dos modelos de reputação.

Os resultados parciais obtidos no decorrer do trabalho foram publicados em (NARDIN et al., 2008a), (NARDIN et al., 2008b) e (NARDIN et al., 2008c).

Além do desenvolvimento da arquitetura SOARI, o resultado e análise dos experimentos pode ser considerado uma contribuição deste trabalho. Como não foi identificado na literatura analisada nenhuma arquitetura e implementação para tratar o problema de interoperabilidade de reputação, não foi possível comparar o desempenho da arquitetura SOARI a nenhuma outra arquitetura semelhante.

Uma das possibilidades de trabalhos futuros é melhorar os mecanismos de tratamento de *Problemas de Tradução*, uma vez que algumas das estratégias geram problemas adicionais. Uma idéia é a criação de heurísticas que tentam refinar o processo de mapeamento com métodos específicos para seu domínio de aplicação, assim criando dois níveis para a realização da identificação da correspondência entre conceitos.

Uma outra possibilidade é avaliar a aplicação da abordagem de interoperabilidade semântica com o uso de SOA em aspectos distintos em SMA, por exemplo, em modelos organizacionais (COUTINHO et al., 2008).

A arquitetura SOARI pode ser ainda estendida para possibilitar o uso de métodos semi-automáticos de alinhamento, servindo até mesmo como uma plataforma de teste para tais métodos.

Uma das premissas adotadas neste trabalho é a confiabilidade no SMO, assim considerando-o sempre disponível e acessível, além de sempre responder corretamente às requisições de tradução de conceitos. Se desconsiderada esta premissa, algumas das possíveis extensões a esta arquitetura são: (1) a replicação do SMO

para tratamento de falhas e (2) considerar a reputação do serviço de mapeamento.

Além de questões relacionadas à arquitetura, diversos outros experimentos podem ser realizados, inclusive com o uso de uma variedade maior de modelos de reputação usando as mesmas plataformas de simulação. Desta forma, é possível expandir a análise relacionada ao impacto da heterogeneidade de modelos de reputação na exatidão da identificação da reputação de agentes. Além disto, pode-se realizar uma análise para identificar o relacionamento entre as características dos modelos de reputação e os impactos positivos trazidos com o uso da arquitetura SOARI.

Referências

- AVEGLIANO, P. *Simulação da formação de parcerias entre agentes baseadas no conceito de reputação*. Dissertação (Mestrado) — Escola Politécnica - Universidade de São Paulo, 2008. 85 p.
- AXELROD, R. *The evolution of cooperation*. New York: Basic Books, 1984. 256 p.
- BECKETT, D.; BROEKSTRA, J. *SPARQL Query Results XML Format*. [S.l.], 2008. Disponível em: <<http://www.w3.org/TR/rdf-sparql-XMLres/>>.
- BOCKTING, S. *A Semantic Translation Service using Ontologies*. 2005.
- BOOTH, D. et al. (Ed.). *Web Services Architecture*. [S.l.], 2004. Disponível em: <<http://www.w3.org/TR/ws-arch/>>.
- BOSLAUGH, S.; WATTERS, D. P. A. *Statistics in a nutshell*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2008. ISBN 9780596510497.
- BRANDÃO, A. A. F. et al. Extending the art testbed to deal with heterogeneous agent reputation models. In: CASTELFRANCHI, C. et al. (Ed.). *Proceedings of the 10th International Workshop on Trust in Agent Societies*. Honolulu, USA: [s.n.], 2007.
- BURANARACH, M. *The Foundation for Semantic Interoperability on the World Wide Web*. Tese (Doutorado) — University of Pittsburgh, 2001.
- CARTER, J.; MOLINA, J. M.; MURO, J. D. Trust management through fuzzy reputation. *Journal in Cooperative Information Systems*, v. 12, n. 1, p. 135–155, 2003.
- CASARE, S. *Uma Ontologia Funcional de Reputação*. Dissertação (Mestrado) — Escola Politécnica - Universidade de São Paulo, 2005. 169 p.
- CASARE, S.; SICHTMAN, J. S. Towards a functional ontology of reputation. In: *AAMAS '05: Proceedings of the 4th Joint Conference on Autonomous Agents and Multiagent Systems*. Utrecht, The Netherlands: ACM, 2005. p. 505–511.
- CASARE, S.; SICHTMAN, J. S. Using a functional ontology of reputation to interoperate different agent reputation models. *Journal of the Brazilian Computer Society*, v. 11, n. 2, p. 79–94, 2005.
- CASTELFRANCHI, C. Engineering social order. In: *ESAW '00: Proceedings of the First International Workshop on Engineering Societies in the Agent World*. London, UK: Springer, 2000. v. 1972, p. 1–18.

- CHANG, E.; HUSSAIN, F. K.; DILLON, T. S. Reputation ontology for reputation systems. In: MEERSMAN, R.; TARI, Z.; HERRERO, P. (Ed.). *SWWS '06: Proceedings of the International Workshop on Web Semantics*. Montpellier, France: Springer, 2006. (Lecture Notes in Computer Science, v. 4278), p. 1724–1733. ISBN 3-540-48273-3.
- CHRISTENSEN, E. et al. *Web Services Description Language (WSDL) 1.1*. [S.l.], 2001. Disponível em: <<http://www.w3.org/TR/wsdl>>.
- CONTE, R.; PAOLUCCI, M. *Reputation in Artificial Societies. Social Beliefs for Social Order*. Boston: Kluwer, 2002. Hardcover. ISBN 9781402071867.
- CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies: where is their meeting point? *Data Knowledge Engineering*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, v. 46, n. 1, p. 41–64, 2003.
- COUTINHO, L. et al. Model-driven integration of organizational models. In: *AOSE '08: Proceedings of the 9th International Workshop on Agent Oriented Software Engineering*. Lisbon, Portugal: [s.n.], 2008.
- DAMERON, O. et al. Accessing and manipulating ontologies using web services. In: *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*. Hiroshima, Japan: [s.n.], 2004.
- DOAN, A. et al. Learning to match ontologies on the semantic web. In: *The VLDB Journal*. Secaucus, USA: Springer-Verlag New York, Inc., 2003. v. 12, n. 4, p. 303–319.
- DOU, D.; MCDERMOTT, D.; QI, P. Ontology translation on the semantic web. In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Heidelberg, Germany: Springer, 2003. (Lecture Notes in Computer Science, v. 2888), p. 952–969.
- ERL, T. *SOA: Principles of Service Design*. Upper Saddle River, USA: Prentice Hall PTR, 2007. ISBN 0132344823.
- FARQUHAR, A.; FIKES, R. E.; RICE, J. The ontolingua server: a tool for collaborative ontology construction. In: . Stanford, USA: Elsevier Science Inc., 1996.
- FERNÁNDEZ-LOPES, M.; GÓMEZ-PÉREZ, A.; JURINO, N. Methontology: From ontological art towards ontological engineering. In: *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*. Stanford, USA: AAAI, 1997. p. 33–40.
- FULLAM, K. K. et al. The agent reputation and trust (art) testbed architecture. In: *Proceedings of the 8th International Workshop on Trust in Agent Societies*. Utrecht, The Netherlands: [s.n.], 2005.
- FULLAM, K. K. et al. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In: *AAMAS '05: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*. New York, USA: ACM, 2005. p. 512–518.

- GENESERETH, M. R.; FIKES, R. E. *Knowledge interchange format, version 3.0 reference manual*. Stanford, USA, 1992.
- GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, Academic Press Ltd., London, UK, v. 5, n. 2, p. 199–220, 1993.
- GUARINO, N. Formal ontology and information systems. In: *FOIS '98: Proceedings of the First International Conference on Formal Ontologies in Information Systems*. Amsterdam, The Netherlands: IOS Press, 1998. p. 3–15.
- HASS, H.; BROWN, A. *Web Services Glossary*. [S.l.], 2004. Disponível em: <<http://www.w3.org/TR/ws-gloss/>>.
- HORRIDGE, M. et al. *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and COODE Tools Edition 1.0*. The University Of Manchester, 2004.
- HORROCKS, I. Fact and ifact. In: *DL '09: Proceedings of the International Workshop on Description Logics*. Linköping, Sweden: CEUR-WS.org, 1999. (CEUR Workshop Proceedings, v. 22), p. 133–135.
- HORROCKS, I. Reviewing the design of daml+oil: An ontology language for the semantic web. In: *AAAI '02: Proceedings of the 18th National Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2002. p. 792–797.
- IEEE. *IEEE standard computer dictionary : a compilation of IEEE standard computer glossaries*. New York, USA: IEEE Computer Society Press, 1991. Hardcover. ISBN 1559370793.
- IEEE. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. [S.l.], 2000. i–23 p.
- ISO. *ISO 8879:1986 - Information Processing, Text and Office Systems, Standard Generalized Markup Language (SGML)*. Geneva, Switzerland: International Organization for Standardization, 1986. xi + 155 p. Disponível em: <<http://www.iso.ch/cate/d16387.html>>.
- KALFOGLOU, Y.; SCHORLEMMER, M. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, Cambridge University Press, New York, USA, v. 18, n. 1, p. 1–31, 2003.
- KLYNE, G.; CARROLL, J. J. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. [S.l.], 2004. Disponível em: <<http://www.w3.org/TR/rdf-concepts/>>.
- MADHAVAN, J.; BERNSTEIN, P. A.; RAHM, E. Generic schema matching with cupid. In: *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*. San Francisco, USA: Morgan Kaufmann, 2001. p. 49–58.
- MCGOVERN, J. et al. *Java Web Service Architecture*. San Francisco, USA: Morgan Kaufmann, 2003.

- MCGUINNESS, D. L. et al. The chimaera ontology environment. In: *Proceedings of the 7th National Conference on Artificial Intelligence and 20th Conference on Innovative Applications of Artificial Intelligence*. Austin, USA: AAAI Press/The MIT Press, 2000. p. 1123–1124.
- MCGUINNESS, D. L.; HARMELEN, F. van. *OWL Web Ontology Language Overview*. [S.l.], 2004. Disponível em: <<http://www.w3.org/TR/owl-features/>>.
- MCILRAITH, S.; MARTIN, D. Bringing semantics to web services. *Intelligent Systems, IEEE*, v. 18, n. 1, p. 90–93, 2003.
- MITRA, N.; LAFON, Y. *SOAP Version 1.2 Part 0: Primer (Second Edition)*. [S.l.], 2007. Disponível em: <<http://www.w3.org/TR/soap12-part0/>>.
- MULLER, G.; VERCOUTER, L. *L.I.A.R. Achieving Social Control in Open and Decentralised Multi-Agent Systems*. Saint-Étienne, France, 2008.
- NARDIN, L. G. et al. An ontology mapping service to support agent reputation models interoperability. In: *Proceedings of the 11th International Workshop on Trust in Agent Societies*. Estoril, Portugal: [s.n.], 2008. p. 140–144.
- NARDIN, L. G. et al. A service-oriented architecture to support agent reputation models interoperability. In: *Proceedings of the 3rd Workshop on Ontologies and Their Applications*. Salvador, Brasil: [s.n.], 2008.
- NARDIN, L. G. et al. *SOARI: A Service Oriented Architecture to Support Agent Reputation Models Interoperability*. Heidelberg, Germany: Springer, 2008. 292–307 p. (Lecture Notes in Computer Science, v. 5396).
- NOY, N. F.; MUSEN, M. A. Prompt: Algorithm and tool for automated ontology merging and alignment. In: *Proceedings of the 7th National Conference on Artificial Intelligence and 20th Conference on Innovative Applications of Artificial Intelligence*. [S.l.]: AAAI Press/The MIT Press, 2000. p. 450–455.
- OSTROM, E. A behavioral approach to the rational choice theory of collective action: Presidential address, american political science association, 1997. *The American Political Science Review*, v. 92, n. 1, p. 1–22, 1998. Disponível em: <<http://www.jstor.org/stable/2585925>>.
- PINYOL, I.; SABATER-MIR, J.; CUNÍ, G. How to talk about reputation using a common ontology: From definition to implementation. In: *Proceedings of the 9th International Workshop on Trust in Agent Societies*. Honolulu, USA: [s.n.], 2007. p. 90–101.
- PRUD'HOMMEAUX, E.; SEABORNE, A. *SPARQL Query Language for RDF (Working Draft)*. [S.l.], 2007. Disponível em: <<http://www.w3.org/TR/2007-WD-rdf-sparql-query-20070326/>>.
- RACER SYSTEMS GMBH AND CO. KG. *RACERPro User's Guide Version 1.9.2*. Hamburg, Germany, 2007.
- RESNICK, P. et al. Reputation systems. *Commun. ACM*, ACM Press, New York, USA, v. 43, n. 12, p. 45–48, 2000.

- SABATER-MIR, J.; PAOLUCCI, M. On representation and aggregation of social evaluations in computational trust and reputation models. *International Journal of Approximate Reasoning*, Elsevier Science Inc., New York, USA, v. 46, n. 3, p. 458–483, 2007.
- SABATER-MIR, J.; PAOLUCCI, M.; CONTE, R. Repute: Reputation and image among limited autonomous partners. *Journal of Artificial Societies and Social Simulation*, v. 9, n. 2, 2006. Disponível em: <<http://jasss.soc.surrey.ac.uk/9/2/3%20-.html>>.
- SABATER-MIR, J.; SIERRA, C. Social regret, a reputation model based on social relations. *SIGecom Exch.*, ACM Press, New York, USA, v. 3, n. 1, p. 44–56, 2002.
- SILVA, D. L. da; SOUZA, R. R.; ALMEIDA, M. B. Ontologias e vocabulários controlados: comparação de metodologias para construção. *Ciência da Informação*, Brasília, Brasil, v. 37, n. 3, p. 60–75, 2008.
- SIRIN, E. et al. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, v. 5, n. 2, p. 51–53, 2007.
- USCHOLD, M. et al. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, v. 11, p. 93–136, 1996.
- VALENTE, A. *Legal Knowledge Engineering - A Modeling Approach*. Amsterdam: IOS Press, 1995.
- VERCOUTER, L. et al. An experience on reputation models interoperability based on a functional ontology. In: *IJCAI '07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*. Hyderabad, India: [s.n.], 2007. p. 617–622.
- VISSER, U. et al. Enabling technologies for interoperability. In: VISSER, U.; PUNDT, H. (Ed.). *Proceedings of the Workshop on the 14th International Symposium of Computer Science for Environmental Protection*. Bonn: TZI, 2000. p. 35–46.
- WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. Chichester, England: John Wiley & Sons, 2002. Paperback. ISBN 047149691X.
- YU, B.; SINGH, M. P. An evidential model of distributed reputation management. In: *AAMAS '02: Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*. New York, USA: ACM Press, 2002. p. 294–301.
- ZACHARIA, G.; MAES, P. Trust management through reputation mechanisms. *Journal of Applied Artificial Intelligence*, Taylor & Francis, v. 14, n. 9, p. 881–907, 2000.

Apêndice A – Resultados dos Experimentos

Neste apêndice são apresentados os resultados obtidos dos experimentos realizados neste trabalho. Os valores apresentados correspondem ao valor médio dos atributos de reputação atribuídos pelos agentes *Honestos* ao agente *Desonesto*, obtido a cada ciclo de simulação.

Tabela A.1: Resultados do Experimento 1: ART/L.I.A.R.

Ciclo	DibRp	IibRp	RpRcbRp
1	1,0000	1,0000	1,0000
2	0,9900	0,8100	1,0000
3	0,9900	0,8100	1,0000
4	0,9900	0,8100	1,0000
5	0,9900	0,8100	1,0000
6	0,9900	0,8100	1,0000
7	0,9900	0,8100	1,0000
8	0,9900	0,8100	1,0000
9	0,9800	0,7797	1,0000
10	0,9700	0,5965	1,0000
11	0,9700	0,5965	0,9975
12	0,9700	0,5965	0,9953
13	0,9700	0,5965	0,9933
14	0,9700	0,6007	0,9922
15	0,9500	0,6007	0,9906
16	0,9500	0,6007	0,9902
17	0,9500	0,6007	0,9883
18	0,9500	0,6007	0,9880
19	0,9500	0,6007	0,9866
20	0,9500	0,6037	0,9840

21	0,9400	0,5661	0,9817
22	0,9400	0,5661	0,9815
23	0,9400	0,5683	0,9796
24	0,9400	0,5683	0,9787
25	0,9400	0,5683	0,9781
26	0,9400	0,5683	0,9762
27	0,9400	0,5683	0,9762
28	0,9300	0,5395	0,9759
29	0,9300	0,5395	0,9754
30	0,9300	0,5395	0,9749
31	0,9300	0,5395	0,9745
32	0,9300	0,6959	0,9741
33	0,9300	0,6959	0,9739
34	0,9300	0,6959	0,9735
35	0,9200	0,7770	0,9735
36	0,9200	0,7840	0,9732
37	0,9200	0,7840	0,9729
38	0,9200	0,7854	0,9728
39	0,9200	0,7864	0,9727
40	0,9200	0,8017	0,9724
41	0,9200	0,8365	0,9723
42	0,9200	0,8365	0,9721
43	0,9200	0,8415	0,9721
44	0,9200	0,8457	0,9721
45	0,9200	0,8457	0,9719
46	0,9200	0,8489	0,9717
47	0,9200	0,8489	0,9716
48	0,9100	0,8292	0,9715
49	0,9085	0,8247	0,9715
50	0,9085	0,8328	0,9713
51	0,9085	0,8490	0,9712
52	0,9085	0,8608	0,9711
53	0,9085	0,8646	0,9710
54	0,9085	0,8646	0,9710
55	0,9085	0,8682	0,9708
56	0,9085	0,8682	0,9708

57	0,9085	0,8687	0,9707
58	0,9085	0,8687	0,9707
59	0,9085	0,8687	0,9706
60	0,9085	0,8713	0,9706
61	0,9085	0,8701	0,9706
62	0,8985	0,8385	0,9705
63	0,8985	0,8390	0,9698
64	0,8985	0,8413	0,9687
65	0,8885	0,8286	0,9673
66	0,8785	0,8220	0,9661
67	0,8785	0,8454	0,9653
68	0,8785	0,8454	0,9650
69	0,8785	0,8474	0,9640
70	0,8785	0,8474	0,9636
71	0,8785	0,8507	0,9629
72	0,8785	0,8507	0,9626
73	0,8785	0,8521	0,9620
74	0,8785	0,8643	0,9613
75	0,8779	0,8641	0,9611
76	0,8679	0,8371	0,9606
77	0,8679	0,8427	0,9596
78	0,8679	0,8447	0,9592
79	0,8679	0,8447	0,9584
80	0,8679	0,8447	0,9575
81	0,8679	0,8510	0,9562
82	0,8643	0,8474	0,9553
83	0,8643	0,8474	0,9547
84	0,8643	0,8502	0,9542
85	0,8408	0,8158	0,9537
86	0,8283	0,7988	0,9525
87	0,8283	0,7988	0,9512
88	0,8283	0,7988	0,9500
89	0,8183	0,7952	0,9496
90	0,8152	0,7882	0,9492
91	0,8152	0,7882	0,9481
92	0,8152	0,7882	0,9471

93	0,8152	0,7882	0,9464
94	0,8152	0,7882	0,9455
95	0,8152	0,7882	0,9452
96	0,8152	0,7882	0,9447
97	0,8052	0,7815	0,9441
98	0,7898	0,7644	0,9431
99	0,7898	0,7644	0,9417
100	0,7898	0,7644	0,9406

Tabela A.2: Resultados do Experimento 2: ART/Repage

Ciclo	Imagem	Reputação
1	1,0000	1,0000
2	0,8354	1,0000
3	0,7809	1,0000
4	0,7875	0,9999
5	0,7699	0,9999
6	0,7658	0,9999
7	0,7638	0,9999
8	0,7664	0,9999
9	0,7197	0,9999
10	0,7013	0,9987
11	0,6889	0,9933
12	0,7028	0,9842
13	0,7367	0,9701
14	0,7496	0,9684
15	0,7344	0,9645
16	0,7090	0,9599
17	0,7215	0,9571
18	0,7066	0,9527
19	0,7353	0,9498
20	0,7112	0,9464
21	0,7083	0,9356
22	0,6858	0,9258
23	0,6736	0,9194

24	0,6923	0,9076
25	0,6889	0,9038
26	0,6778	0,8956
27	0,6746	0,8882
28	0,6748	0,8852
29	0,6851	0,8824
30	0,6855	0,8805
31	0,6992	0,8761
32	0,7012	0,8750
33	0,7117	0,8780
34	0,7185	0,8789
35	0,7327	0,8785
36	0,7226	0,8774
37	0,6975	0,8760
38	0,7076	0,8746
39	0,7370	0,8731
40	0,7371	0,8776
41	0,7069	0,8835
42	0,6985	0,8852
43	0,7182	0,8887
44	0,7085	0,8899
45	0,7519	0,8873
46	0,7481	0,8919
47	0,7007	0,8976
48	0,7204	0,8966
49	0,7067	0,8976
50	0,7080	0,8965
51	0,7199	0,8951
52	0,7625	0,8919
53	0,7466	0,8973
54	0,7231	0,9046
55	0,7063	0,9105
56	0,7211	0,9140
57	0,7194	0,9206
58	0,7018	0,9275
59	0,6999	0,9202

60	0,6914	0,9141
61	0,7048	0,9111
62	0,7144	0,9105
63	0,7313	0,9039
64	0,7111	0,9018
65	0,6938	0,9006
66	0,7132	0,8956
67	0,6995	0,8955
68	0,7079	0,8928
69	0,7123	0,8908
70	0,7001	0,8850
71	0,7132	0,8776
72	0,7242	0,8761
73	0,7402	0,8736
74	0,7113	0,8751
75	0,7350	0,8784
76	0,7264	0,8807
77	0,7198	0,8801
78	0,7063	0,8823
79	0,7397	0,8786
80	0,7278	0,8808
81	0,7340	0,8826
82	0,7109	0,8847
83	0,7174	0,8835
84	0,7156	0,8884
85	0,7401	0,8914
86	0,7262	0,8916
87	0,7410	0,8971
88	0,7666	0,8985
89	0,7717	0,9020
90	0,7616	0,9016
91	0,7475	0,8992
92	0,7323	0,8979
93	0,7273	0,8988
94	0,7203	0,8997
95	0,7575	0,9006

96	0,7221	0,9007
97	0,7156	0,9020
98	0,7345	0,9042
99	0,7241	0,9096
100	0,7214	0,9121

Tabela A.3: Resultados do Experimento 3.1: ART/Misto-D/L.I.A.R.

Ciclo	DIbRp	IbRp	RpRcbRp	Imagem	Reputação
1	1,0000	1,0000	1,0000	1,0000	1,0000
2	1,0000	0,4481	1,0000	0,8218	1,0000
3	1,0000	0,4481	0,9826	0,7996	0,9995
4	1,0000	0,4061	0,9650	0,8057	0,9997
5	1,0000	0,5091	0,9520	0,7584	0,9997
6	1,0000	0,6233	0,9288	0,7469	0,9997
7	1,0000	0,6650	0,9217	0,7419	0,9997
8	1,0000	0,6907	0,9148	0,7663	0,9980
9	1,0000	0,7194	0,9096	0,7474	0,9980
10	1,0000	0,7534	0,9082	0,7446	0,9980
11	1,0000	0,7534	0,9036	0,7503	0,9979
12	1,0000	0,7398	0,9012	0,7191	0,9999
13	0,9928	0,7407	0,8986	0,7216	0,9962
14	0,9728	0,7274	0,8954	0,7224	0,9906
15	0,9728	0,7274	0,8900	0,7603	0,9860
16	0,9728	0,7171	0,8855	0,7486	0,9785
17	0,9728	0,7213	0,8825	0,7476	0,9603
18	0,9728	0,7213	0,8777	0,7519	0,9622
19	0,9728	0,7210	0,8755	0,7325	0,9533
20	0,9728	0,7233	0,8729	0,7285	0,9469
21	0,9728	0,7379	0,8721	0,7528	0,9469
22	0,9728	0,7507	0,8699	0,7180	0,9469
23	0,9728	0,7446	0,8653	0,7193	0,9450
24	0,9728	0,6914	0,8617	0,7096	0,9434
25	0,9528	0,6733	0,8545	0,7237	0,9412
26	0,9528	0,6754	0,8496	0,7130	0,9305

27	0,9528	0,6801	0,8442	0,7385	0,9253
28	0,9528	0,7121	0,8435	0,7843	0,9175
29	0,9528	0,7276	0,8424	0,7545	0,9186
30	0,9528	0,7337	0,8405	0,7245	0,9198
31	0,9528	0,7414	0,8394	0,7161	0,9201
32	0,9328	0,7361	0,8374	0,6734	0,9167
33	0,9328	0,7361	0,8315	0,6092	0,8921
34	0,9328	0,7405	0,8212	0,6169	0,8710
35	0,9328	0,7405	0,8175	0,6136	0,8500
36	0,9128	0,7397	0,8054	0,5980	0,8423
37	0,9128	0,7397	0,8005	0,6383	0,8277
38	0,9128	0,7427	0,7944	0,6404	0,8216
39	0,9128	0,7427	0,7880	0,6545	0,8070
40	0,9128	0,7427	0,7847	0,6482	0,8043
41	0,9128	0,7427	0,7802	0,6550	0,7997
42	0,9128	0,7293	0,7732	0,6643	0,8017
43	0,9128	0,7385	0,7682	0,7283	0,7927
44	0,9128	0,7340	0,7660	0,7010	0,7982
45	0,9128	0,7187	0,7652	0,7143	0,7987
46	0,9128	0,7187	0,7629	0,7031	0,8023
47	0,9128	0,7187	0,7609	0,7251	0,8083
48	0,9128	0,7239	0,7545	0,6973	0,8198
49	0,9128	0,7266	0,7528	0,6733	0,8179
50	0,8928	0,7239	0,7519	0,6855	0,8182
51	0,8903	0,7207	0,7467	0,6879	0,8143
52	0,8903	0,7241	0,7429	0,7171	0,8178
53	0,8903	0,7258	0,7419	0,6654	0,8130
54	0,8903	0,7258	0,7383	0,6723	0,8130
55	0,8903	0,7258	0,7364	0,6900	0,8147
56	0,8903	0,7258	0,7355	0,7116	0,8118
57	0,8903	0,7275	0,7336	0,7460	0,8116
58	0,8703	0,7250	0,7310	0,7599	0,8170
59	0,8703	0,7250	0,7296	0,7390	0,8169
60	0,8703	0,7302	0,7293	0,7460	0,8161
61	0,8703	0,7324	0,7290	0,6721	0,8161
62	0,8703	0,7324	0,7267	0,7178	0,8193

63	0,8703	0,7324	0,7252	0,7290	0,8213
64	0,8703	0,7333	0,7230	0,7265	0,8243
65	0,8639	0,7308	0,7226	0,7285	0,8289
66	0,8639	0,7372	0,7204	0,7439	0,8288
67	0,8639	0,7349	0,7197	0,7817	0,8325
68	0,8639	0,7360	0,7185	0,7851	0,8366
69	0,8639	0,7381	0,7182	0,7909	0,8407
70	0,8639	0,7390	0,7172	0,7303	0,8426
71	0,8639	0,7390	0,7153	0,7547	0,8401
72	0,8639	0,7523	0,7152	0,7350	0,8381
73	0,8639	0,7576	0,7146	0,7477	0,8313
74	0,8639	0,7576	0,7146	0,7261	0,8269
75	0,8639	0,7583	0,7131	0,7556	0,8307
76	0,8639	0,7583	0,7126	0,7280	0,8324
77	0,8439	0,7501	0,7118	0,7051	0,8314
78	0,8439	0,7501	0,7103	0,6814	0,8293
79	0,8439	0,7501	0,7082	0,6911	0,8267
80	0,8297	0,7319	0,7067	0,7479	0,8236
81	0,8297	0,7425	0,7049	0,7223	0,8169
82	0,8297	0,7435	0,7038	0,7179	0,8074
83	0,8297	0,7440	0,7028	0,7354	0,8034
84	0,8297	0,7434	0,7015	0,7824	0,7998
85	0,8297	0,7451	0,7004	0,7554	0,8112
86	0,8297	0,7451	0,6990	0,7853	0,8077
87	0,8097	0,7419	0,6981	0,7056	0,8095
88	0,8097	0,7427	0,6973	0,7467	0,8093
89	0,8097	0,7439	0,6962	0,7672	0,8068
90	0,8097	0,7429	0,6945	0,7322	0,8069
91	0,8097	0,7476	0,6941	0,7651	0,8077
92	0,8097	0,7486	0,6934	0,7565	0,8029
93	0,8097	0,7493	0,6932	0,7760	0,8044
94	0,8097	0,7557	0,6924	0,7235	0,8045
95	0,8097	0,7557	0,6913	0,7673	0,8040
96	0,8097	0,7563	0,6905	0,7837	0,7984
97	0,8097	0,7557	0,6901	0,8015	0,8006
98	0,8097	0,7549	0,6896	0,7621	0,8037

99	0,8097	0,7549	0,6884	0,7914	0,8038
100	0,8097	0,7549	0,6880	0,7548	0,8038

Tabela A.4: Resultados do Experimento 3.2: ART/Misto-D/Repage

Ciclo	DIbRp	IbRp	RpRcbRp	Imagem	Reputação
1	1,0000	1,0000	1,0000	1,0000	1,0000
2	1,0000	0,4153	1,0000	0,8489	1,0000
3	0,9800	0,4406	0,9845	0,8406	1,0000
4	0,9800	0,4406	0,9761	0,8507	0,9998
5	0,9800	0,4406	0,9645	0,8071	0,9998
6	0,9800	0,6197	0,9544	0,7573	0,9994
7	0,9800	0,6076	0,9409	0,7102	0,9974
8	0,9800	0,6076	0,9305	0,7170	0,9956
9	0,9800	0,6076	0,9235	0,6575	0,9901
10	0,9800	0,6312	0,9103	0,7175	0,9884
11	0,9600	0,6400	0,9017	0,7157	0,9924
12	0,9600	0,7256	0,8962	0,7207	0,9828
13	0,9600	0,7256	0,8861	0,7044	0,9798
14	0,9600	0,7120	0,8797	0,7218	0,9768
15	0,9600	0,7195	0,8794	0,7270	0,9749
16	0,9600	0,7390	0,8761	0,7649	0,9714
17	0,9400	0,7263	0,8723	0,7779	0,9770
18	0,9400	0,7463	0,8684	0,7697	0,9788
19	0,9400	0,7583	0,8585	0,7458	0,9750
20	0,9400	0,7598	0,8545	0,7624	0,9700
21	0,9400	0,7598	0,8495	0,7422	0,9715
22	0,9400	0,7601	0,8479	0,7816	0,9726
23	0,9400	0,7601	0,8458	0,7354	0,9745
24	0,9400	0,7601	0,8427	0,7519	0,9713
25	0,9400	0,7488	0,8419	0,7399	0,9669
26	0,9400	0,7518	0,8404	0,7228	0,9666
27	0,9354	0,7454	0,8394	0,7250	0,9647
28	0,9354	0,7454	0,8371	0,6914	0,9570
29	0,9354	0,7494	0,8320	0,7477	0,9545

30	0,9354	0,7494	0,8305	0,7343	0,9545
31	0,9354	0,7281	0,8241	0,7258	0,9548
32	0,9354	0,7281	0,8223	0,6946	0,9566
33	0,9354	0,7212	0,8201	0,7424	0,9582
34	0,9354	0,7093	0,8188	0,7135	0,9557
35	0,9354	0,7484	0,8175	0,6525	0,9550
36	0,8954	0,7333	0,8152	0,6169	0,9505
37	0,8954	0,7374	0,8127	0,6871	0,9345
38	0,8754	0,7216	0,8101	0,6935	0,9388
39	0,8754	0,7410	0,8093	0,7653	0,9388
40	0,8754	0,7420	0,8077	0,7487	0,9370
41	0,8754	0,7506	0,8070	0,7787	0,9446
42	0,8754	0,7506	0,8067	0,8086	0,9446
43	0,8754	0,7577	0,8064	0,7920	0,9607
44	0,8754	0,7577	0,8063	0,7684	0,9646
45	0,8754	0,7577	0,8060	0,7308	0,9642
46	0,8706	0,7510	0,8053	0,7405	0,9642
47	0,8706	0,7510	0,8047	0,6984	0,9569
48	0,8706	0,7510	0,8038	0,7364	0,9567
49	0,8706	0,7640	0,8023	0,7457	0,9623
50	0,8706	0,7600	0,8022	0,7413	0,9642
51	0,8706	0,7600	0,8016	0,7443	0,9623
52	0,8706	0,7606	0,8011	0,7489	0,9661
53	0,8706	0,7606	0,8005	0,7760	0,9684
54	0,8706	0,7634	0,8005	0,7478	0,9631
55	0,8706	0,7634	0,7998	0,7430	0,9629
56	0,8706	0,7634	0,7990	0,7845	0,9586
57	0,8706	0,7600	0,7992	0,7702	0,9609
58	0,8706	0,7649	0,7990	0,7438	0,9647
59	0,8506	0,7431	0,7991	0,7429	0,9647
60	0,8506	0,7439	0,7989	0,7533	0,9556
61	0,8506	0,7595	0,7981	0,7694	0,9539
62	0,8506	0,7599	0,7978	0,7645	0,9640
63	0,8506	0,7635	0,7975	0,7879	0,9672
64	0,8506	0,7649	0,7976	0,7647	0,9672
65	0,8506	0,7653	0,7972	0,7222	0,9691

66	0,8506	0,7653	0,7964	0,7349	0,9708
67	0,8506	0,7653	0,7963	0,7316	0,9637
68	0,8506	0,7653	0,7960	0,7109	0,9637
69	0,8306	0,7627	0,7952	0,6870	0,9662
70	0,8306	0,7645	0,7949	0,7302	0,9609
71	0,8106	0,7484	0,7945	0,7209	0,9572
72	0,8106	0,7484	0,7942	0,7639	0,9612
73	0,7867	0,7405	0,7931	0,7522	0,9483
74	0,7867	0,7405	0,7929	0,7538	0,9390
75	0,7867	0,7333	0,7928	0,7701	0,9309
76	0,7867	0,7294	0,7914	0,6953	0,9288
77	0,7867	0,7297	0,7907	0,6614	0,9217
78	0,7834	0,7264	0,7885	0,6774	0,9195
79	0,7634	0,7278	0,7848	0,6703	0,9194
80	0,7634	0,7278	0,7839	0,6891	0,9126
81	0,7634	0,7278	0,7832	0,6678	0,9128
82	0,7634	0,7287	0,7812	0,6427	0,9182
83	0,7634	0,7287	0,7795	0,6669	0,9219
84	0,7634	0,7287	0,7783	0,7691	0,9068
85	0,7634	0,7287	0,7781	0,7529	0,9132
86	0,7593	0,7247	0,7780	0,7895	0,9062
87	0,7393	0,7201	0,7766	0,7362	0,9072
88	0,7393	0,7201	0,7759	0,7598	0,9064
89	0,7393	0,7128	0,7760	0,7487	0,9018
90	0,7393	0,7245	0,7754	0,7525	0,8971
91	0,7393	0,7245	0,7752	0,7566	0,8913
92	0,7393	0,7250	0,7734	0,7244	0,8940
93	0,7393	0,7273	0,7718	0,7311	0,8994
94	0,7393	0,7253	0,7705	0,7264	0,8992
95	0,7393	0,7256	0,7684	0,7581	0,9072
96	0,7393	0,7289	0,7673	0,7307	0,9032
97	0,7393	0,7236	0,7665	0,7319	0,8982
98	0,7393	0,7236	0,7657	0,7383	0,9020
99	0,7358	0,7181	0,7643	0,7006	0,9000
100	0,7358	0,7191	0,7638	0,7511	0,9002

Tabela A.5: Resultados do Experimento 4: FOReART/L.I.A.R.

Ciclo	DlbRp	IlbRp	RpRcbRp
1	1,0000	1,0000	1,0000
2	0,9900	0,9587	0,9600
3	0,9800	0,7037	0,9340
4	0,9700	0,5137	0,9070
5	0,9600	0,3237	0,8910
6	0,9600	0,4566	0,8790
7	0,9500	0,2866	0,8600
8	0,9500	0,3066	0,8470
9	0,9400	0,2966	0,8410
10	0,9300	0,2255	0,8350
11	0,9200	0,2388	0,8280
12	0,9200	0,2388	0,8230
13	0,9200	0,2117	0,8160
14	0,9200	0,2117	0,8140
15	0,9200	0,2338	0,8110
16	0,9200	0,2338	0,8080
17	0,9200	0,3555	0,8080
18	0,9200	0,3853	0,8080
19	0,9200	0,3853	0,8060
20	0,9200	0,3974	0,8040
21	0,9200	0,4269	0,8020
22	0,9200	0,4269	0,8020
23	0,9200	0,4269	0,8010
24	0,9100	0,4361	0,8000
25	0,9100	0,4361	0,8000
26	0,9100	0,4679	0,8000
27	0,9100	0,4746	0,8000
28	0,9000	0,6336	0,8000
29	0,9000	0,6336	0,8000
30	0,9000	0,6336	0,8000
31	0,9000	0,6402	0,8000
32	0,9000	0,6556	0,8000
33	0,9000	0,6556	0,8000

34	0,9000	0,6556	0,8000
35	0,8967	0,6275	0,8000
36	0,8867	0,6606	0,8000
37	0,8867	0,6606	0,8000
38	0,8843	0,6535	0,8000
39	0,8843	0,6535	0,8000
40	0,8843	0,6535	0,8000
41	0,8843	0,6535	0,8000
42	0,8743	0,6619	0,8000
43	0,8743	0,6619	0,8000
44	0,8743	0,6619	0,8000
45	0,8743	0,6619	0,8000
46	0,8643	0,6474	0,8000
47	0,8643	0,6474	0,8000
48	0,8643	0,6562	0,8000
49	0,8643	0,6619	0,8000
50	0,8606	0,6536	0,8000
51	0,8606	0,6536	0,8000
52	0,8620	0,6621	0,8000
53	0,8620	0,6621	0,8000
54	0,8520	0,6789	0,8000
55	0,8520	0,6789	0,8000
56	0,8420	0,6782	0,8000
57	0,8220	0,6559	0,8000
58	0,8220	0,6559	0,8000
59	0,8120	0,6492	0,8000
60	0,8120	0,6674	0,8000
61	0,8120	0,6718	0,8000
62	0,8120	0,6725	0,8000
63	0,8120	0,6745	0,8000
64	0,8020	0,6671	0,8000
65	0,8020	0,7010	0,8000
66	0,8020	0,7010	0,8000
67	0,7920	0,6900	0,8000
68	0,7920	0,6911	0,8000
69	0,7920	0,7000	0,8000

70	0,7920	0,7029	0,8000
71	0,7920	0,7029	0,8000
72	0,7820	0,6932	0,8000
73	0,7820	0,6932	0,8000
74	0,7820	0,6955	0,8000
75	0,7820	0,6955	0,8000
76	0,7820	0,6955	0,8000
77	0,7820	0,6955	0,8000
78	0,7820	0,6955	0,8000
79	0,7783	0,6897	0,8000
80	0,7783	0,6921	0,8000
81	0,7783	0,6950	0,8000
82	0,7683	0,7131	0,8000
83	0,7683	0,7131	0,8000
84	0,7583	0,7122	0,8000
85	0,7549	0,7059	0,8000
86	0,7449	0,7050	0,8000
87	0,7449	0,7058	0,8000
88	0,7449	0,7077	0,8000
89	0,7420	0,6971	0,8000
90	0,7320	0,6986	0,8000
91	0,7220	0,6948	0,8000
92	0,7120	0,6930	0,8000
93	0,7096	0,6913	0,8000
94	0,6996	0,6895	0,8000
95	0,6996	0,6911	0,8000
96	0,6881	0,6757	0,8000
97	0,6781	0,6684	0,8000
98	0,6781	0,6690	0,8000
99	0,6781	0,6690	0,8000
100	0,6781	0,6690	0,8000

Tabela A.6: Resultados do Experimento 5: FOrEART/Repage

Ciclo	Imagem	Reputação
1	1,0000	1,0000
2	0,8309	0,9500
3	0,8220	0,9062
4	0,7830	0,8474
5	0,7437	0,7949
6	0,6793	0,7349
7	0,6573	0,6962
8	0,6281	0,6487
9	0,6293	0,6137
10	0,6000	0,5675
11	0,5690	0,5275
12	0,5640	0,5063
13	0,5344	0,4788
14	0,5413	0,4613
15	0,5453	0,4413
16	0,5176	0,4126
17	0,4865	0,3901
18	0,5210	0,3751
19	0,5180	0,3601
20	0,5569	0,3376
21	0,5488	0,3089
22	0,5244	0,2803
23	0,5010	0,2478
24	0,5267	0,2303
25	0,5139	0,2129
26	0,5570	0,1979
27	0,5178	0,1755
28	0,4541	0,1680
29	0,4485	0,1492
30	0,4345	0,1280
31	0,4127	0,1218
32	0,4118	0,1167
33	0,4051	0,1080
34	0,3920	0,1004

35	0,3793	0,0891
36	0,4336	0,0766
37	0,4374	0,0716
38	0,4190	0,0679
39	0,4003	0,0541
40	0,3446	0,0467
41	0,3631	0,0342
42	0,4196	0,0267
43	0,4047	0,0242
44	0,4060	0,0167
45	0,4142	0,0155
46	0,4063	0,0130
47	0,3771	0,0130
48	0,3335	0,0130
49	0,3214	0,0105
50	0,3247	0,0105
51	0,3318	0,0067
52	0,2958	0,0067
53	0,3844	0,0042
54	0,3846	0,0042
55	0,3701	0,0042
56	0,3262	0,0017
57	0,3219	0,0017
58	0,3554	0,0004
59	0,3663	0,0004
60	0,3831	0,0004
61	0,3637	0,0005
62	0,3316	0,0005
63	0,3295	0,0005
64	0,3184	0,0005
65	0,3329	0,0005
66	0,3127	0,0005
67	0,2965	0,0005
68	0,2993	0,0005
69	0,3230	0,0004
70	0,3263	0,0004

71	0,3521	0,0004
72	0,3201	0,0004
73	0,3679	0,0004
74	0,3675	0,0004
75	0,3768	0,0004
76	0,4148	0,0004
77	0,3934	0,0004
78	0,3895	0,0005
79	0,4341	0,0005
80	0,3810	0,0005
81	0,3779	0,0005
82	0,3942	0,0005
83	0,3909	0,0005
84	0,3558	0,0005
85	0,4223	0,0005
86	0,4074	0,0005
87	0,4126	0,0005
88	0,3936	0,0005
89	0,3914	0,0005
90	0,4032	0,0005
91	0,4369	0,0005
92	0,4080	0,0005
93	0,3782	0,0005
94	0,3651	0,0005
95	0,3263	0,0005
96	0,3687	0,0005
97	0,3140	0,0005
98	0,2931	0,0004
99	0,3626	0,0004
100	0,3359	0,0004

Tabela A.7: Resultados do Experimento 6.1: FOReART/Misto-D/L.I.A.R.

Ciclo	DIbRp	IbRp	RpRcbRp	Imagem	Reputação
1	1,0000	1,0000	1,0000	1,0000	1,0000

2	1,0000	1,0000	0,9600	0,8711	0,9500
3	1,0000	0,9271	0,9192	0,8064	0,9175
4	1,0000	0,9478	0,8680	0,7992	0,8850
5	1,0000	0,9449	0,8458	0,7627	0,8575
6	0,9600	0,7316	0,8168	0,7410	0,8150
7	0,9600	0,7189	0,7870	0,7343	0,7925
8	0,9600	0,7189	0,7624	0,7224	0,7625
9	0,9600	0,7189	0,7495	0,7319	0,7475
10	0,9600	0,7189	0,7353	0,7300	0,7275
11	0,9600	0,7107	0,7117	0,7202	0,7000
12	0,9600	0,6696	0,6988	0,7544	0,6700
13	0,9600	0,6696	0,6950	0,7262	0,6301
14	0,9400	0,6617	0,6761	0,7117	0,6151
15	0,9400	0,6690	0,6692	0,7070	0,6076
16	0,9400	0,6690	0,6594	0,6721	0,6001
17	0,9400	0,6690	0,6493	0,6384	0,5826
18	0,9400	0,6690	0,6323	0,6818	0,5776
19	0,9400	0,6690	0,6182	0,6939	0,5651
20	0,9400	0,6756	0,6081	0,6783	0,5551
21	0,9400	0,6756	0,5874	0,6455	0,5551
22	0,9400	0,6756	0,5776	0,6409	0,5476
23	0,9400	0,6756	0,5720	0,6696	0,5451
24	0,9400	0,6858	0,5559	0,6976	0,5451
25	0,9400	0,6858	0,5429	0,6833	0,5326
26	0,9400	0,6871	0,5362	0,6568	0,5276
27	0,9400	0,6871	0,5310	0,6509	0,5251
28	0,9400	0,6871	0,5274	0,6627	0,5226
29	0,9332	0,6705	0,5233	0,5984	0,5176
30	0,9332	0,6705	0,5158	0,6528	0,5151
31	0,9332	0,6791	0,5109	0,6308	0,5151
32	0,9332	0,6791	0,5075	0,6360	0,5101
33	0,9332	0,6834	0,5072	0,6340	0,5076
34	0,9332	0,6834	0,5007	0,6366	0,5076
35	0,9332	0,6834	0,4968	0,6556	0,5076
36	0,9332	0,6834	0,4895	0,6232	0,5051
37	0,9332	0,6842	0,4849	0,6767	0,5051

38	0,9332	0,6842	0,4832	0,6833	0,5051
39	0,9332	0,6842	0,4818	0,6726	0,5051
40	0,9332	0,6842	0,4787	0,6683	0,5026
41	0,9332	0,6842	0,4769	0,6553	0,5026
42	0,9332	0,6842	0,4745	0,6469	0,5026
43	0,9332	0,6842	0,4719	0,6469	0,5026
44	0,9332	0,6849	0,4722	0,6287	0,5076
45	0,9332	0,6849	0,4665	0,6228	0,5051
46	0,9132	0,6713	0,4630	0,6693	0,5201
47	0,9132	0,6731	0,4632	0,6605	0,5401
48	0,9132	0,6731	0,4622	0,6173	0,5401
49	0,9132	0,6731	0,4623	0,6420	0,5401
50	0,9132	0,6731	0,4627	0,6577	0,5401
51	0,9132	0,6740	0,4636	0,6791	0,5401
52	0,9132	0,6740	0,4653	0,6994	0,5401
53	0,9132	0,6727	0,4697	0,7058	0,5401
54	0,9132	0,6764	0,4737	0,6843	0,5451
55	0,9132	0,6764	0,4722	0,6821	0,5251
56	0,8932	0,6696	0,4734	0,6215	0,5251
57	0,8932	0,6696	0,4707	0,6213	0,5151
58	0,8932	0,6696	0,4663	0,6383	0,5151
59	0,8932	0,6696	0,4672	0,5780	0,5151
60	0,8932	0,6696	0,4621	0,6236	0,5151
61	0,8932	0,6672	0,4601	0,6594	0,5151
62	0,8932	0,6695	0,4611	0,6977	0,5051
63	0,8932	0,6695	0,4607	0,6503	0,5051
64	0,8732	0,6655	0,4616	0,6273	0,5051
65	0,8732	0,6655	0,4584	0,6379	0,5051
66	0,8732	0,6655	0,4591	0,6606	0,5051
67	0,8532	0,6455	0,4563	0,6498	0,5051
68	0,8532	0,6455	0,4536	0,6342	0,5051
69	0,8532	0,6455	0,4486	0,6315	0,5051
70	0,8532	0,6455	0,4502	0,6788	0,5051
71	0,8532	0,6455	0,4507	0,6605	0,5001
72	0,8532	0,6455	0,4510	0,6575	0,5001
73	0,8532	0,6399	0,4502	0,6602	0,5001

74	0,8532	0,6399	0,4484	0,6275	0,5001
75	0,8532	0,6399	0,4485	0,6764	0,5001
76	0,8532	0,6399	0,4476	0,6426	0,5001
77	0,8532	0,6404	0,4499	0,6824	0,5001
78	0,8532	0,6404	0,4488	0,6262	0,5001
79	0,8332	0,6361	0,4466	0,6175	0,5001
80	0,8332	0,6392	0,4457	0,6435	0,5001
81	0,8332	0,6396	0,4458	0,6493	0,5001
82	0,8332	0,6396	0,4468	0,6622	0,5001
83	0,8332	0,6396	0,4477	0,6393	0,5001
84	0,8332	0,6396	0,4472	0,7095	0,5001
85	0,8332	0,6396	0,4501	0,6631	0,5001
86	0,8332	0,6401	0,4502	0,6252	0,5001
87	0,8332	0,6401	0,4487	0,6462	0,5001
88	0,8332	0,7361	0,4482	0,6535	0,5001
89	0,8332	0,7361	0,4448	0,6222	0,5001
90	0,8332	0,7365	0,4443	0,5907	0,5001
91	0,8332	0,7369	0,4434	0,6219	0,5001
92	0,8332	0,7378	0,4415	0,6309	0,5001
93	0,8332	0,7378	0,4409	0,6486	0,5001
94	0,8332	0,7378	0,4409	0,6265	0,5001
95	0,8332	0,7378	0,4429	0,6385	0,5001
96	0,8332	0,7260	0,4389	0,6534	0,5001
97	0,8332	0,7260	0,4405	0,6517	0,5001
98	0,8332	0,7260	0,4442	0,6416	0,5001
99	0,8132	0,7040	0,4445	0,6604	0,5001
100	0,8132	0,7040	0,4447	0,6621	0,5001

Tabela A.8: Resultados do Experimento 6.2: FOrEART/Misto-D/Repape

Ciclo	DIbRp	IbRp	RpRcbRp	Imagem	Reputação
1	1,0000	1,0000	1,0000	1,0000	1,0000
2	1,0000	1,0000	0,9600	0,8500	0,9500
3	1,0000	1,0000	0,9191	0,7812	0,9225
4	1,0000	1,0000	0,8874	0,6867	0,8925

5	0,9800	0,8200	0,8592	0,7109	0,8450
6	0,9600	0,6400	0,8268	0,7193	0,8150
7	0,9600	0,6400	0,7941	0,7373	0,7825
8	0,9600	0,6400	0,7739	0,7267	0,7550
9	0,9600	0,6400	0,7556	0,7466	0,7175
10	0,9200	0,5346	0,7234	0,7007	0,6875
11	0,9200	0,5209	0,6986	0,7254	0,6675
12	0,9000	0,5176	0,6899	0,7275	0,6575
13	0,9000	0,5176	0,6749	0,6935	0,6475
14	0,9000	0,4515	0,6556	0,7147	0,6400
15	0,9000	0,4515	0,6387	0,7221	0,6326
16	0,9000	0,4580	0,6325	0,7346	0,6276
17	0,9000	0,4029	0,6277	0,7022	0,6201
18	0,9000	0,3968	0,6160	0,7081	0,6151
19	0,9000	0,3926	0,6125	0,6846	0,6051
20	0,9000	0,3836	0,6088	0,6827	0,5876
21	0,8800	0,3808	0,5978	0,6873	0,5801
22	0,8800	0,3808	0,5828	0,6808	0,5726
23	0,8800	0,3808	0,5749	0,6792	0,5701
24	0,8800	0,3808	0,5712	0,6862	0,5676
25	0,8800	0,3808	0,5494	0,6627	0,5651
26	0,8800	0,3808	0,5451	0,6683	0,5626
27	0,8800	0,3808	0,5427	0,6629	0,5551
28	0,8800	0,3856	0,5384	0,6957	0,5551
29	0,8600	0,3808	0,5303	0,6896	0,5526
30	0,8600	0,3808	0,5270	0,6999	0,5501
31	0,8600	0,3808	0,5243	0,6894	0,5476
32	0,8600	0,3808	0,5205	0,6984	0,5476
33	0,8600	0,3808	0,5144	0,6704	0,5351
34	0,8600	0,3808	0,5083	0,6843	0,5226
35	0,8600	0,3808	0,5045	0,6705	0,5201
36	0,8600	0,3808	0,5000	0,6459	0,5201
37	0,8600	0,3980	0,4957	0,6658	0,5201
38	0,8600	0,3847	0,4954	0,6879	0,5126
39	0,8600	0,3847	0,4940	0,6815	0,5076
40	0,8600	0,3729	0,4905	0,6752	0,5026

41	0,8600	0,5026	0,4875	0,6789	0,5026
42	0,8600	0,5026	0,4839	0,6506	0,5026
43	0,8600	0,5079	0,4790	0,6319	0,5001
44	0,8600	0,5079	0,4761	0,6653	0,5001
45	0,8600	0,5079	0,4732	0,7019	0,5001
46	0,8600	0,5079	0,4716	0,6787	0,5001
47	0,8600	0,5032	0,4693	0,6313	0,5001
48	0,8600	0,5032	0,4630	0,5957	0,5001
49	0,8600	0,5032	0,4583	0,6127	0,5001
50	0,8600	0,5032	0,4562	0,6744	0,5001
51	0,8600	0,5032	0,4572	0,6682	0,5001
52	0,8600	0,5032	0,4587	0,6574	0,5001
53	0,8600	0,5032	0,4587	0,6862	0,5001
54	0,8600	0,6170	0,4586	0,6835	0,5001
55	0,8600	0,6170	0,4615	0,6842	0,5001
56	0,8600	0,6170	0,4638	0,6688	0,5001
57	0,8400	0,6135	0,4642	0,7055	0,5001
58	0,8200	0,6005	0,4645	0,6525	0,5001
59	0,8200	0,6005	0,4635	0,6672	0,5001
60	0,8200	0,6322	0,4642	0,6878	0,5001
61	0,8200	0,6447	0,4661	0,6629	0,5001
62	0,8200	0,6563	0,4649	0,6558	0,5001
63	0,8000	0,6451	0,4643	0,6539	0,5001
64	0,8000	0,6451	0,4614	0,6670	0,5001
65	0,8000	0,6374	0,4594	0,6680	0,5001
66	0,8000	0,6131	0,4603	0,6568	0,5001
67	0,8000	0,6131	0,4598	0,6659	0,5001
68	0,8000	0,6048	0,4601	0,6837	0,5001
69	0,7972	0,6022	0,4605	0,6521	0,5001
70	0,7972	0,6147	0,4620	0,6805	0,5001
71	0,7972	0,6221	0,4617	0,6776	0,5001
72	0,7972	0,6221	0,4619	0,6926	0,5001
73	0,7772	0,6046	0,4628	0,6680	0,5001
74	0,7772	0,6046	0,4625	0,6778	0,5001
75	0,7772	0,6073	0,4630	0,6756	0,5001
76	0,7772	0,6136	0,4601	0,6935	0,5001

77	0,7772	0,6274	0,4609	0,6722	0,4951
78	0,7772	0,6345	0,4602	0,6690	0,4951
79	0,7697	0,6210	0,4579	0,6623	0,4926
80	0,7697	0,6194	0,4573	0,6550	0,4876
81	0,7646	0,6085	0,4552	0,6490	0,4851
82	0,7646	0,6182	0,4533	0,6759	0,4826
83	0,7446	0,6107	0,4540	0,6717	0,4801
84	0,7446	0,6141	0,4545	0,6596	0,4801
85	0,7446	0,6141	0,4535	0,6511	0,4801
86	0,7446	0,6108	0,4526	0,6358	0,4801
87	0,7446	0,6130	0,4524	0,6261	0,4776
88	0,7446	0,6130	0,4520	0,6229	0,4776
89	0,7446	0,6163	0,4507	0,6700	0,4776
90	0,7446	0,6301	0,4498	0,6530	0,4751
91	0,7446	0,6323	0,4499	0,6351	0,4751
92	0,7446	0,6381	0,4492	0,6151	0,4751
93	0,7446	0,6381	0,4476	0,6251	0,4751
94	0,7446	0,6381	0,4474	0,6387	0,4751
95	0,7446	0,6376	0,4457	0,6268	0,4751
96	0,7446	0,6347	0,4458	0,6240	0,4751
97	0,7446	0,6427	0,4448	0,6353	0,4751
98	0,7388	0,6221	0,4434	0,6326	0,4751
99	0,7188	0,5886	0,4445	0,6447	0,4751
100	0,7188	0,5886	0,4440	0,6664	0,4751

Apêndice B – Testes Estatísticos

Neste apêndice é apresentada uma seqüência de testes estatísticos para a seleção do melhor teste para a análise dos resultados dos experimentos deste trabalho.

Primeiramente, uma análise utilizando o *Teste T* de Student (BOSLAUGH; WATTERS, 2008) foi realizada sobre os resultados dos experimentos. Esta análise foi realizada sobre os pares de experimentos conforme Tabelas 7.11, 7.13 e 7.15.

Os resultados obtidos com o uso deste teste são apresentados nas Tabela B.1 e B.2, respectivamente para a análise referente a expressividade da comunicação e heterogeneidade de modelos de reputação.

Tabela B.1: Valores de p para verificar o efeito da expressividade de comunicação

Pares	Hipótese				
	A	B	C	D	E
(1, 4)	0,14	0,21	0,00	n/a	n/a
(2, 5)	n/a	n/a	n/a	0,00	0,00
(3.1, 6.1)	0,52	0,35	0,01	0,03	0,01
(3.1, 6.2)	0,19	0,11	0,01	0,02	0,00
(3.2, 6.1)	0,75	0,46	0,00	0,02	0,00
(3.2, 6.2)	0,45	0,18	0,00	0,01	0,00

Finalizada a geração de p usando o Teste T de Student, identificou-se que tal teste possui duas premissas que não haviam sido verificadas, quais sejam, (i) os dados têm ter distribuição normal e (ii) a variância dos conjuntos de dados analisados tem que ser homogênea.

Assim, para verificar a possibilidade de aplicação do Teste T de Student realizou-se os testes *Shapiro-Wilk* e *Teste F* para verificação, respectivamente, da distribuição normal e homogeneidade da variância dos resultados dos experimentos.

O resultado da aplicação dos Testes Shapiro-Wilk e F são apresentados nas Tabelas B.3 e B.4.

Tabela B.2: Valores de p para verificar o efeito da heterogeneidade

Pares	Hipótese				
	F	G	H	I	J
(1, 3.1)	0,60	0,46	0,01	n/a	n/a
(1, 3.2)	0,32	0,34	0,02	n/a	n/a
(2, 3.1)	n/a	n/a	n/a	0,75	0,15
(2, 3.2)	n/a	n/a	n/a	0,75	0,44
(4, 6.1)	0,91	0,59	0,00	n/a	n/a
(4, 6.2)	0,63	0,30	0,00	n/a	n/a
(5, 6.1)	n/a	n/a	n/a	1,00	1,00
(5, 6.2)	n/a	n/a	n/a	1,00	1,00

Tabela B.3: Valores de p para Teste Shapiro-Wilk

Experimento	DIbRp	IIbRp	RpRcbRp	Imagem	Reputação
1	0,73	0,57	0,00	n/a	n/a
2	n/a	n/a	n/a	0,55	0,00
3.1	0,04	0,08	0,04	0,01	0,00
3.2	0,01	0,01	0,00	0,04	0,00
4	0,52	0,15	0,00	n/a	n/a
5	n/a	n/a	n/a	0,13	0,37
6.1	0,08	0,05	0,59	0,49	0,00
6.2	0,05	0,40	0,75	0,21	0,00

Para a análise do Teste Shapiro-Wilk, formulou-se duas hipóteses:

H_0 : Distribuição Normal

H_1 : Distribuição Não Normal

Considerando um nível de significância (α) de 0,05, rejeita-se H_0 para valores de p menor ou igual a 0,05.

Analisando os valores de p da Tabela B.3, verifica-se que diversos valores são inferiores a 0,05, assim, não sendo possível sustentar a premissa de distribuição normal para todos os experimentos.

Uma análise similar é realizada para os resultados obtidos com o Teste F. Para tanto, formulou-se duas hipóteses:

H_0 : Variância Homogênea

H_1 : Variância Não Homogênea

Tabela B.4: Valores de p para Teste F

Pares	D I b R p	I I b R p	R p R c b R p	Imagem	Reputação
(1, 4)	0,92	0,92	0,00	n/a	n/a
(2, 5)	n/a	n/a	n/a	0,90	0,00
(3.1, 6.1)	0,21	0,88	0,00	0,29	0,00
(3.1, 6.2)	0,78	0,87	0,00	0,10	0,00
(3.2, 6.1)	0,02	0,71	0,00	0,63	0,00
(3.2, 6.2)	0,26	0,69	0,00	0,34	0,00
(1, 3.1)	0,71	0,69	0,97	n/a	n/a
(1, 3.2)	0,97	0,87	0,90	n/a	n/a
(2, 3.1)	n/a	n/a	n/a	0,44	0,99
(2, 3.2)	n/a	n/a	n/a	0,15	0,94
(4, 6.1)	0,05	0,60	1,00	n/a	n/a
(4, 6.2)	0,45	0,58	1,00	n/a	n/a
(5, 6.1)	n/a	n/a	n/a	0,03	0,00
(5, 6.2)	n/a	n/a	n/a	0,01	1,00

Considerando um nível de significância (α) de 0,05, rejeita-se H_0 para valores de p menor ou igual a 0,05.

Analisando os valores de p da Tabela B.4, verifica-se que diversos valores são inferiores a 0,05, assim não foi possível sustentar a premissa de variância homogênea para os diversos pares de combinação de experimentos.

Mediante a análise dos resultados dos testes Shapiro-Wilk e F concluiu-se que a aplicação do Teste T de Student não era adequada para a análise dos resultados dos experimentos.

Por fim, optou-se pelo uso do *Teste da Soma das Ordens* de Wilcoxon neste trabalho.