VANDERSON BOTELHO DA SILVA

DOSSIÊ: MODELO DE CONFIANÇA PARA SISTEMAS MULTIAGENTES

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutor em Informática.

CURITIBA

Julho/2017

VANDERSON BOTELHO DA SILVA

DOSSIÊ: MODELO DE CONFIANÇA PARA SISTEMAS MULTIAGENTES

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutor em Informática.

Área de Concentração: *Agentes de Software*

Orientador: Prof. Dr. Edson Emílio Scalabrin

CURITIBA

Julho/2017

Silva, Vanderson Botelho da

Dossiê: Modelo de Confiança para Sistemas Multiagentes. Curitiba, 2017. 26-07-2017

Tese de doutorado – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Modelo de Confiança 2. Reputação 3. Sistemas Multiagentes 4. Segurança da Informação. I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática II-t

FOLHA DE APROVAÇÃO

Vanderson Botelho da Silva

DOSSIÊ: MODELO DE CONFIANÇA PARA SISTEMAS MULTIAGENTES

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutor em Informática.

Banca examinadora

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Instituição: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Instituição: _____ Assinatura: _____

Aos meus pais,
minha amada esposa Mirinha,
meus queridos filhos Heitor e Felipe.

Agradecimentos

Primeiramente agradeço a minha família, que sempre me apoiou nos momentos mais difíceis da vida, em especial a minha esposa *Mirinha*, minha grande parceira, ao meu lado me tornou mais forte, aos meus filhos Heitor e Felipe que transformaram minha vida para melhor, desejo que as minhas lutas sirvam de legado para eles. Agradeço aos professores do PPGIa da PUCPR, em especial ao meu orientador Dr. Edson Scalabrin que me fez o convite para entrar nesse programa e foi o grande responsável por eu estar aqui, sempre me acolheu com palavras de ânimo e entendeu minhas limitações. Agradeço a CAPES pelo incentivo financeiro a mim e a tantos outros beneficiados. A educação é uma força transformadora, programas como esse fortalecem a esperança de uma sociedade melhor. Agradeço aos meus colegas de estudo, o Kelvin Kredens com suas dicas e entusiasmo para discutir assuntos do meu tema, ao Jones Granatyr pelo trabalho que fizemos e que resultou no meu artigo mais relevante aceito pela ACM, ao André Pinz que juntos também escrevemos artigos e por várias vezes me ajudou no laboratório e ao Otto Lessing pelas conversas e contribuições ao meu projeto. Chego ao final deste trabalho e percebo que o esforço, horas solitário, somado a motivação e inspiração de amigos, professores e familiares, foram peças chaves dessa vitória. Espero retribuir toda a ajuda e atenção que recebi e compartilhar o fruto dessa jornada com todos que precisarem de mim.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

AC	*Autoridade Certificadora*
ACL	*Agent Communication Language*
ART	*Agent Reputation and Trust*
BDI	*Belief, Desire, Intention*
BOVESPA	*Bolsa de Valores de São Paulo*
CRUD	*Create, Read, Update and Delete*
ESB	*Enterprise Service Bus*
FIPA	*Foundation for Intelligent Physical Agents*
IEEE	*Institute of Electrical and Electronics Engineers*
SIMOC	*Simulador de Modelos de Confiança*
SHA	*Secure Hash Algorithm*
SMA	*Sistema Multiagente*
SOA	*Arquitetura Orientada a Serviço*
SSL	*Secure Sockets Layer*

Resumo

Modelos de confiança para sistemas multiagentes são considerados complexos e desafiadores, à medida que promovem segurança nas relações entre agentes de software, sem restringir a liberdade dos ambientes abertos, nos quais os agentes, *a priori*, desconhecidos, podem entrar e sair livremente das comunidades virtuais. Este trabalho apresenta um novo modelo de confiança que agrega múltiplas fontes de informação para avaliar o comportamento de um agente. Esse modelo incorpora experiências diretas entre os agentes, testemunhos, contexto, informações sociais, preconceito, reputação, e propõe uma nova estrutura de dados, denominada *Dossiê*. Em termos gerais, o *Dossiê* é uma estrutura de dados que permite manter a história dos *feedbacks*/avaliações sobre um dado agente; tais avaliações são mantidas localmente pelo próprio agente avaliado sem modificação ou omissão que o beneficie. Esta abordagem visa reduzir as limitações em termos de fontes de informação baseadas em *experiências diretas* entre agentes e *experiências indiretas* obtidas por meio de testemunhos. As *fontes diretas* possuem baixo desempenho *vis-à-vis* a dificuldade de um agente realizar certo número de interações para produzir um conjunto significativa de experiências. As *fontes indiretas* dependem do desprendimento de testemunhas para compartilhar suas experiências. Por meio de um simulador construído para esse fim, foram realizados experimentos com o modelo proposto *Dossiê* e os resultados foram confrontados com os principais modelos de confiança descritos na literatura. Os experimentos foram configurados para avaliar diferentes cenários de uso. A análise comparativa realizada mostrou que o modelo *Dossiê* tem um comportamento equivalente ou superior aos modelos considerados e que tal modelo pode tornar os agentes mais eficientes na escolha dos seus parceiros.

Palavras-Chave: modelo de confiança, reputação, sistema multiagente, agente de software.

Abstract

Trust models for multi-agent systems are complex and challenging as they promote security in the relationships between software agents without restricting the freedom of open environments in which agents are unknown and can freely enter and exit virtual communities. This work presents the *Dossier*, a new trust model that aggregates multiple sources of information to evaluate the behavior of an agent. This model deals with: direct experiences among agents, testimonies, context, social information, prejudice, reputation, and proposes a new data structure called *Dossier*. In general terms, the dossier consists of assessments about a particular agent which is responsible for maintaining and sharing the assessments without any tampering that might benefit him. This approach aims to reduce the limitations in terms of sources of information based on direct experiences between agents and indirect experiences obtained through testimony. Direct source have poor performance because the difficulty of an agent performing a number of interactions with extraneous agents to produce a meaningful base of experience. Indirect source depend on the detachment of witnesses to share their experiences. By means of a simulator constructed for this purpose, experiments were carried out with the proposed Dossier model and the results were confronted with the main trust models described in the literature. The experiments were configured to simulate different usage scenarios. The comparative analysis, according to each scenario, shows that the *Dossier* model is statistically equivalent or superior to the selected models. The comparative analysis showed that the Dossier model has a behavior equivalent or superior to the considered models and that such a model can make the agents more efficient in choosing their partners.

Keywords: trust model, reputation, multi-agent system, software agent.

Capítulo 1

Introdução

Confiança é um conceito fundamental para a construção e estabilidade das relações humanas. O fato de confiar, seja em alguém ou em algo, permite tomar decisão de como as interações são criadas entre pessoas, sociedades, organizações e máquinas. Novos mercados e comunidades virtuais surgem a todo o instante, tais ambientes são por natureza concebidos para estabelecer e manter interligações entre indivíduos antes desconhecidos. Desse modo, o tratamento da confiança tem se tornado cada vez mais imprescindível e desafiador.

Há recorrentes casos de indivíduos ludibriados ao solicitarem serviços para *sítios eletrônicos* que pouco ou nunca ouviram falar, ou falsos serviços na *Internet* que se passam por legítimos. Adicionalmente, pela percepção reduzida ou ínfima dos riscos que envolvem estes ambientes, os participantes de mercados e comunidades virtuais estão muitas vezes, vulneráveis a situações enganosas. Diante do potencial risco, os *modelos de confiança* contribuem na interação entre indivíduos ou organizações desconhecidas com níveis de segurança maior.

A percepção de confiança leva a interações bem-sucedidas, assim como a mercados de melhor qualidade, enquanto a confiança equivocada ou a desconfiança descabida são prejudiciais para as interações comerciais. Para haver evolução na assimilação da confiança e melhor tomada de decisão, as sociedades e os mercados precisam compreender melhor essa dinâmica em relação aos aspectos tecnológicos, comerciais, comportamentais, legais e culturais (JOSANG, 2010).

A partir deste ponto de vista, a comunidade de *inteligência artificial* motiva-se a propor alternativas que permitam transcender o conceito de confiança e demais representações mentais para modelos computacionais. Alguns exemplos de sistemas que

podem atribuir crenças e outras qualidades mentais: termostatos; sistemas de autoconfiguração; *time-sharing*; e sistemas de inferência lógica (Mccarthy, 1979). Os *sistemas multiagentes*, um campo derivado da *inteligência artificial distribuída*, tem contribuído neste sentido por meio de linhas de estudo nas quais *agentes*, definidos como entidades capazes de interagir socialmente, podem operar modelos mentais. Os *modelos de confiança*, nos *sistemas multiagentes*, têm maior aplicabilidade em ambientes *abertos*, nos quais agentes dependem de mecanismos para avaliar o comportamento de potenciais parceiros. Neste campo, inúmeras propostas de modelos foram sugeridas, sendo necessário estabelecer classificações para diferentes abordagens conforme suas características e de acordo com a natureza do problema. Dentre essas classificações, destacam-se Tyrone Grandison e M. Sloman (2000), Sarvapali Ramchurn *et al*. (2004), Jordi Sabater e Carles Sierra, (2005), Gehao Lu *et al*. (2009) e Isaac Pinyol e Jordi Sabater, (2013) que se propõem a identificar conceitos e características dos modelos de confiança. A pesquisa científica nesta área desenvolveu-se significativamente, configurando-se como elemento essencial aos *sistemas multiagentes*.

Este trabalho propõe a criação de um modelo de confiança que permite, em *ambientes multiagentes abertos*, selecionar bons parceiros, indivíduos que cooperem para atingir objetivos em comum. O modelo proposto baseia-se no paradigma *numérico*, utiliza as fontes de informação *direta* e *indireta* e atua sobre a suposição *nível 2*. Além disso, o modelo propõe uma estrutura de dados denominada de *Dossiê*. Tal modelo permite a cada agente armazenar localmente os *feedbacks*/avaliações recebidas sobre si e compartilhá-las sob demanda. Por se tratar de um modelo de suposição *nível 2*, a abordagem proposta apresenta mecanismos que impedem um dado agente omitir avalições sobre si.

1.1 Motivação e Descrição do Problema

Os problemas voltados à confiança têm sido explorados por diversas pesquisas, tendo por motivação: o crescente número de aplicações em *web* semântica, *grid* computacional, teoria dos jogos, *web services*, redes *peer-to-peer*, sistemas de avaliação e recomendação (Artz e Gil, 2007).

A descentralização de um sistema, proposta na especialidade da *inteligência artificial distribuída*, pressupõe a distribuição do controle e dos dados, sem autoridades

reguladoras ou entidades controladoras. Essa abordagem, apesar de longínqua no campo da *Ciência da Computação*, renova-se a partir de recente tendência comportamental em que pessoas optam por confiar em *estranhos* ao mesmo tempo que questionam a credibilidade de instituições tradicionalmente seguras. Conflitos ligados a ideais ou intolerância religiosa; escândalos de corrupção entre instituições públicas e multinacionais; ou crises globais envolvendo grandes instituições financeiras, são alguns dos inúmeros episódios cotidianos que enfraquecem a imagem das reconhecidas *autoridades confiáveis* (Botsman e Rogers, 2009).

Concomitantemente expande-se o número de mercados sob comunidades virtuais abertas, das quais indivíduos tomam coragem para interagir com desconhecidos. São exemplos: plataformas que conectam motoristas e passageiros a dividirem despesas de viagens de longa distância; sistemas de locação entre hóspedes e anfitriões, criando um mercado concorrente a hotelaria; ou sistemas de pagamento eletrônico, baseados em moedas virtuais, dispensando bancos para intermediar transações. Essas novas formas de interação, mostram como a descentralização do controle e dos dados tem se tornado corriqueira. Tal fato é possível por meio de modelos computacionais que viabilizam a percepção de confiança mesmo entre indivíduos desconhecidos.

Os *sistemas multiagentes abertos* são conhecidos, não apenas pela descentralização, mas por permitirem criar ambientes virtuais livres, onde os indivíduos podem entrar e sair das comunidades de forma facultativa. Os riscos intrínsecos desses ambientes são equivalentes à de outras comunidades abertas. O estudo dos *modelos de confiança* para ambientes virtuais permite propor, simular e experimentar novas possibilidades de interação entre indivíduos desconhecidos.

Como já dito, as fontes de informações necessárias para a percepção de confiança podem ser classificadas em *diretas* e *indiretas*. Ambas as abordagens possuem limitações. Modelos *diretos* apresentam baixo desempenho até que os agentes possam estabelecer uma quantidade razoável de interações para construir percepções de confiança. Os modelos *indiretos*, baseados em testemunhos, convivem com o desafio de motivar os agentes a compartilharem suas experiências, situação nem sempre factível para agentes menos colaborativos ou em ambientes competitivos (Huynh *et al*. 2006). Além disso, o paradigma indireto apresenta outros desafios: *Como encontrar testemunhas*? *Como relacionar-se com elas*? *Como confiar em testemunhas*? (Jurca e Faltings, 2006).

Apesar das diversas iniciativas para o tratamento das fontes de informação, por meio de modelos de confiança, elas permanecem insuficientes para atender a complexidade dos inúmeros tipos de comunidades virtuais. Diante desses problemas novas propostas que ampliem a eficiência dos modelos de confiança são imprescindíveis para a coexistência nas comunidades virtuais abertas.

1.2 Hipóteses

As hipóteses que norteiam essa proposta fundamentam-se: (i) sob um modelo de confiança descentralizado, onde o controle e os dados podem estar distribuídos logicamente e fisicamente, os agentes avaliados podem testemunhar sobre si mesmos; (ii) os testemunhos, fornecidos pelos agentes avaliados, podem ser considerados legítimos à medida que as informações trafegadas são verificadas quanto a integridade e autenticidade por meio de algoritmos de criptografia assimétrica; e (iii) o *dossiê* é uma estrutura de dados capaz de evitar que agentes omitam ou modifiquem qualquer avaliação.

1.3 Objetivos

Grande parte dos modelos de confiança utiliza fontes de informação *diretas* ou *indiretas*, normalmente combinando-as para obter melhores resultados. Este trabalho tem como objeto de estudo a construção de um modelo de confiança capaz de integrar diferentes fontes de informação, e propor uma nova estrutura de dados denominada de *Dossiê*, cuja finalidade é prover aos agentes um conjunto de informações sobre seus parceiros de modo seguro e eficiente.

1.3.1 Objetivo Geral

Este trabalho visa construir e avaliar um modelo de confiança que permita aos integrantes de uma comunidade virtual selecionar bons parceiros para interagir, de modo eficiente em relação aos modelos atuais, mitigando riscos nas relações entre agentes de um sistema aberto.

1.3.2 Objetivos Específicos

Para atingir o objetivo principal foram identificados os seguintes objetivos específicos:

1. Examinar os principais modelos de confiança em face as suas estratégias para obtenção das informações necessárias à percepção da confiança;

2. Propor um modelo de confiança que permita distribuir logicamente e fisicamente o controle e os dados de um sistema multiagente;
3. Construir um sistema de simulação genérico capaz de avaliar sob diversas métricas qualquer modelo de confiança; e
4. Avaliar os modelos de confiança examinados na literatura e o modelo proposto sob diferentes cenários de utilização.

1.4 Organização

O presente trabalho está estruturado em seis capítulos. O Capítulo 2 apresenta uma revisão teórica dos principais conceitos e fundamentos relacionados ao tema do trabalho, tais como: sistemas multiagentes, modelos de confiança e ferramentas de avaliação. Ele configura o estado da arte em relação ao tema estudado, fornece um resumo dos principais trabalhos com ênfase nas soluções propostas para problemas de confiança em ambientes abertos. O Capítulo 3 apresenta a tecnologia de *ledger* distribuído que utiliza algoritmos criptográficos para manter a integridade das informações; esse capítulo auxilia o entendimento do modelo proposto, à medida que utiliza alguns desses conceitos para certificar a segurança da informação. No Capítulo 4 é apresentado o modelo proposto deste trabalho, detalhando seus mecanismos; e descreve também o sistema de avaliação proposto. O Capítulo 5 relata os resultados obtidos nos experimentos realizados com o modelo proposto e outros modelos examinados. Finalmente o Capítulo 6 apresenta as discussões finais do trabalho e sugestões para trabalhos futuros.

Capítulo 2

Modelos de Confiança para Sistemas Multiagentes

Este capítulo apresenta os referenciais teóricos fundamentais relacionados a esta pesquisa. Inicialmente, os sistemas multiagentes são abordados quanto aos conceitos gerais, características e tipos de problemas nos quais se aplicam. Estes conceitos são importantes para o entendimento deste trabalho, à medida que o modelo proposto foi avaliado sob um sistema multiagente. Em seguida são apresentados os principais conceitos sobre os modelos de confiança, paradigmas, características e trabalhos relacionados ao seu *estado da arte*. Além disso, são apresentadas ferramentas para avaliação de modelos de confiança e por fim é apresentado um resumo comparativo dos trabalhos analisados.

2.1 Sistemas Multiagentes

O estudo dos sistemas multiagentes tem evoluído ao longo das últimas três décadas, inicialmente como uma subárea da *inteligência artificial distribuída*, atualmente esses sistemas representam um importante tema de pesquisa em franca expansão, indo além do âmbito acadêmico, se propagando na indústria de software. Portanto, a fundamentação teórica sobre o tema é atemporal e requer atenção.

Em Alan Bond e Les Gasser (1986) são apresentados conceitos chaves sobre os sistemas multiagentes, dentre eles, destaca-se:

> "O estudo dos sistemas multiagentes trata a coordenação do comportamento inteligente entre um conjunto de agentes inteligentes e autônomos, como eles podem coordenar seus conhecimentos,

> objetivos, habilidades e planejar juntos para agir ou resolver problemas".

Os sistemas multiagentes apresentam uma alternativa para a modelagem e construção de soluções direcionadas a problemas de alta complexidade. Nessa perspectiva, o uso de agentes, permite construir um conjunto de técnicas, ferramentas e abstrações que ampliam consideravelmente a forma como atualmente são construídas as soluções de software (Jennings *et al*. 1998). Recorre-se cada vez mais a utilização de agentes em uma ampla variedade de aplicações, desde sistemas com menor criticidade, tais como análise e gestão de redes sociais (Palanca, 2014) até sistemas complexos de missão crítica, como o uso de agentes para controle de tráfego aéreo (Molina, 2014) ou sistemas de monitoração em pacientes de alto risco (Campillo-Sanchez, 2014). Em um primeiro momento, pode parecer que tais sistemas, por serem de naturezas distintas, não tenham nada em comum. No entanto, em ambas situações, a abstração essencial para estes sistemas é o modelo de *agente inteligente*. A flexibilidade de uso dos sistemas multiagentes em múltiplos contextos é um dos principais motivadores que leva a academia e a indústria de software ao investimento de estudos desse paradigma.

2.1.1 Agentes

Seria possível supor uma concordância universal sobre o que é um *agente*, visto ser um termo habitualmente utilizado em diversas situações, seja no coloquial, ou em pesquisas científicas. Contudo não há uma definição universalmente aceita desse termo. A falta de consenso pode ser explicada por uma boa dose de debates ainda em curso e controvérsias nesse assunto. Essencialmente, há um consenso geral de que a *autonomia* é o atributo central para a noção de agente. Entretanto, há dificuldades para validar diversos outros atributos associados a um agente. Para alguns sistemas a habilidade do agente em aprender é essencial, para outros a aprendizagem pode ser irrelevante ou até mesmo indesejada.

Apesar dos diferentes pontos de vista, certas definições são importantes e precisam ser consideradas, do contrário, o significado desse termo poderia ser completamente inócuo.

Em Kurt Konolige (1980) temos:

> "Temos interesse de construir um agente que seja inteligente o suficiente para executar cooperativamente tarefas envolvendo outros agentes".

Em N. Sridharan (1986) temos:

> "Agentes são indivíduos que cooperam para atingir um conjunto comum de objetivos".

Em Michael Georgeff (1988) temos:

> "Uma das coisas que robôs e agentes precisam fazer é organizar suas atividades de modo que possam cooperar entre si e evitar conflitos".

Em Pattie Maes (1995) temos:

> "Agentes são sistemas computacionais que habitam, um ambiente dinâmico e complexo, percebem e agem de forma autônoma nesse ambiente, e ao fazê-lo, percebem um conjunto de metas ou tarefas para as quais foram concebidos".

Em Gerhard Weiss, (1999) temos:

> "Um agente é um sistema computacional situado em algum ambiente, e que é capaz de agir de forma autônoma nesse ambiente, a fim de atingir seus objetivos projetados".

Há percepções que distinguem os agentes de outros elementos, como simples termostatos, sistemas orientados a objetos, ou *daemons*, pois, de modo geral, esses são pouco providos de inteligência. Entretanto, o que significaria ser *inteligente*? Ou o que é *inteligência*? Essas são perguntas difíceis de responder. Para fins desse trabalho um agente inteligente é aquele capaz de agir com certo grau de *autonomia* e *flexível* a fim de atender seus objetivos preestabelecidos. A autonomia preconiza—ao agente—operar sem intervenções diretas, humanas ou de outro tipo, e devem possuir controle sobre suas ações e estados internos (Wooldridge e Jennings, 1995). A flexibilidade prover três características adicionais (Weiss, 1999):

i. Reatividade: um agente deve interagir com seu ambiente, seja físico ou virtual, percebendo modificações e reagindo sobre elas em tempo hábil para as mudanças que ocorrem sobre o ambiente;
ii. Pró-atividade: agentes devem agir além das simples respostas ao seu ambiente, ou seja, eles devem tomar a iniciativa, quando necessário, e apresentar comportamentos dirigidos a objetivos; e
iii. Sociabilidade: agentes devem ser capazes de interagir com outros agentes ou seres humanos, a fim de atingir seus objetivos ou ajudar outros em suas atividades.

Um agente pode ser definido como uma entidade de software que exibe comportamentos autônomos e está situado em algum ambiente sob o qual ele é capaz de perceber estímulos e realizar ações para alcançar seus objetivos. O termo ambiente (Figura 2.1) refere-se a uma representação do sistema estudado, onde os agentes são simulados.

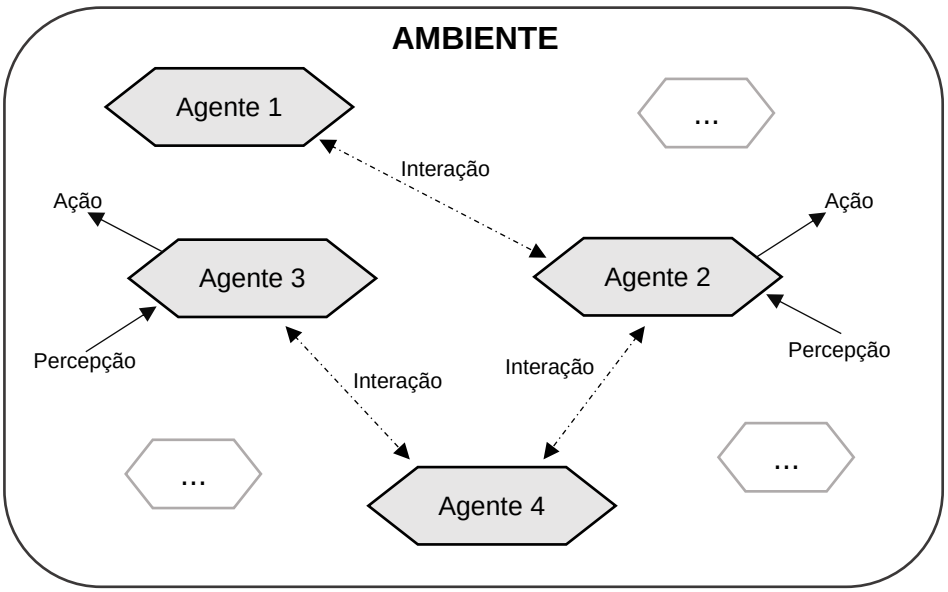

Figura 2.1: Representação lógica de um ambiente para uma sociedade de agentes.

Há outras características que auxiliam o entendimento e diferenciam um sistema multiagente de outros sistemas (Bradshaw, 1997):

i. Inferência: poder agir sob tarefas utilizando conhecimentos anteriores, indo além das informações fornecidas, e possuem alguma representação de si mesmo ou sobre outros;
ii. Continuidade: ser capaz de armazenar e recuperar sua identidade e seus estados durante períodos de tempo distintos;
iii. Mobilidade: poder mover-se intencionalmente de um determinado local para outro; e
iv. Adaptabilidade: ser capaz de melhorar seu desempenho e aprender a partir das suas experiências.

Tradicionalmente, as pesquisas relacionadas a esses sistemas eram apresentadas em conferências de *Inteligência Artificial Distribuída*. Posteriormente o assunto passou a ser dividido em dois campos: RDP (***R**esolução de **P**roblemas **D**istribuídos*) e SMA

(***S**istema **M**ulti**a**gente*). Atualmente, o termo *sistema multiagente* generaliza qualquer sistema que utilize agentes, sejam autônomos ou semiautônomos.

Em RDP, um problema pode ser resolvido por um número de módulos que cooperam e compartilham suas informações sobre certo problema com objetivo de solucioná-lo. Portanto em um sistema puramente baseado em RDP, todas as estratégias de interação entre os módulos são previamente especificadas pelo sistema. Diferente de um SMA, que se entende por uma rede de solucionadores de problemas, com baixo grau de acoplamento entre si, trabalhando de forma conjunta para resolver problemas que vão além das suas capacidades individuais, ou conhecimentos de cada solucionador de problemas. De forma geral, um sistema multiagente apresenta um conjunto de fatores que o diferencia (Jennings *et al*. 1998):

i. Cada agente possui informações ou capacidades incompletas para solucionar um problema, assim cada agente é limitado sobre seu ponto de vista;
ii. Não há um controle central;
iii. Os dados são descentralizados; e
iv. O processamento é assíncrono.

2.1.2 Coordenação

A capacidade de construir soluções robustas e eficientes; a possibilidade de permitir interoperabilidade com sistemas legados; e a habilidade de solucionar problemas cujos dados ou controle são distribuídos, são razões que estimulam a pesquisa dos sistemas multiagentes. No entanto as mesmas características que atraem interesse, também trazem grandes desafios (Bond e Gasser, 1988):

1. Como formular, descrever, decompor e alocar problemas, além de sintetizar resultados entre um grupo de agentes inteligentes?
2. Como tornar agentes capazes a comunicar e interagir? Quais linguagens e protocolos eles devem utilizar? Como e quando comunicar?
3. Como garantir que os agentes atuem de forma coerente em seu processo de tomada de decisão e como lidar com os efeitos externos a suas decisões evitando interações que o coloque em risco?
4. Como habilitar agentes a representar e raciocinar sobre ações, planos, e o conhecimento de outros agentes, a fim agir coordenadamente com eles? Como raciocinar sobre o estado de seus processos?

5. Como reconhecer e conciliar pontos de vista diferentes e objetivos conflitantes entre um conjunto de agentes que tentam coordenar suas ações?
6. Como equilibrar de maneira eficaz a computação e a comunicação local? Como gerir a alocação de recursos limitados?
7. Como evitar ou mitigar um comportamento prejudicial do sistema como um todo, tais como comportamentos caóticos ou oscilatórios?
8. Como conceber tecnologias, plataformas e metodologias de desenvolvimento para sistemas multiagentes?

Ao longo das últimas décadas diversas iniciativas contribuíram com a solução total ou parcial de algumas das questões citadas. É o caso do trabalho de Randall Davis e Reid Smith, (1983), ao propor a resolução de conflitos por meio da negociação, resultando no conhecido protocolo *Contract Net*. Neste os agentes podem assumir dois papeis: o *manager* responsável por monitorar a execução de uma tarefa e o *contractor* responsável por executar as tarefas.

Outros trabalhos empregaram esforços na resolução de problemas por meio de modelos de *cooperação*. Martin Rehak *et al.* (2005), Xiangrong Tong *et al.* (2009) utilizam a *coalizão* como estratégia de cooperação na qual agentes se agrupam para solucionar problemas complexos que sozinhos seriam incapazes de resolvê-los. A formação de times de agentes, por meio de coalizão, é tipicamente realizada em função das habilidades de cada agente, *vis-à-vis*, as necessidades impostas pelas tarefas. As tarefas, sempre que possível, são decompostas em tarefas mais simples e atribuídas para agentes que tenham as habilidades para resolvê-la. Alan Bond e Les Gasser (1986), trata a *competição* como uma forma de interação em situações de conflitos, na maioria das vezes, motivada por escassez de recursos. Nestes casos, a disputa pode ser controlada por um conjunto de regras que regem os agentes concorrentes.

Marcos A. H. Shmeil (1997) define um protocolo de negociação, o qual inclui noções importantes de estratégia e tática a serem aplicadas durante um processo de negociação. A estratégia consiste em gerar uma oferta inicial, incrementar os valores que satisfaçam os critérios, para seus valores de utilidade máxima (satisfaz mais), e para as demais ofertas/contra-ofertas. Quando não for mais possível manter o valor do critério escolhido, decrementa-se de uma unidade, o valor relativo da instância do critério em questão. A tática consiste na troca de informações. É ainda um dos trabalhos mais completos como proposta de um protocolo para desenvolver aplicações

envolvendo agentes de software que negociam a compra/venda de um produto ou serviço.

Não tão distante, a FIPA (***F**oundation for **I**ntelligent **P**hysical **A**gents)*, uma iniciativa mantida pela IEEE, define um conjunto de especificações para a modelagem e construção de sistema multiagente (FIPA, 2002): abstrações arquiteturais de software, linguagens de comunicação, gerenciamento de agentes, protocolos, entre outras. As especificações FIPA, conjuntamente com outras iniciativas, têm facilitado e permitido a construção de sistemas baseados em agentes de forma padronizada e interoperável.

2.1.3 Microserviços

Os sistemas multiagentes são alvo de estudo a mais de três décadas, por mais antigo que pareça, novas abordagens para construção de sistemas utilizam seus princípios de descentralização e autonomia. Um exemplo atual são as *arquiteturas de microserviços*, uma tendência para construção de sistemas de escala global, tais como redes sociais, serviços de *streaming* ou mercados virtuais. A redescoberta da abstração dos agentes aplicada a indústria de software moderna tem oferecido alternativas promissoras para problemas de escalabilidade, disponibilidade e adaptabilidade.

Os *microserviços* são pequenos serviços autônomos que trabalham juntos (Newman, 2015). Uma arquitetura de *microserviços* é composta por elementos de baixo acoplamento com ciclo de vida próprio. O baixo acoplamento permite a atualização e evolução do serviço de forma independente, sem gerar qualquer impacto aos demais serviços da solução. Equipes de desenvolvimento de software podem entender e atualizar seus *microserviços* sem conhecimento das estruturas internas dos *microserviços* pares, visto que eles interagem estritamente por meio de interfaces, assim não compartilham estruturas de dados, esquemas de banco de dados, ou outras representações internas de seus objetos.

Essa nova abordagem vem se tornando cada vez mais popular em grandes organizações como Amazon[1], Google[2] e Netflix[3] que fomentam a divulgação abertamente das suas arquiteturas de *microserviços*. A Amazon, por exemplo, relata que até 2001 seu sistema de vendas não passava de um sistema monolítico, *pesado* e que tinha chegado ao limite da escalabilidade. Deixar de ser uma mera loja de livros para se

[1] https://www.amazon.com
[2] https://www.google.com
[3] https://netflix.com

tornar uma das maiores companhia de varejo on-line, dependeu de várias iniciativas, dentre elas a mudança arquitetural do seu sistema de informação, de modelo um centralizado para uma solução plenamente distribuída sob pequenos serviços coordenados. Esses e outros depoimentos incentivam organizações a repensarem suas soluções na busca de maior escalabilidade, desempenho e disponibilidade dos serviços.

Um aspecto importante dos *microserviços* é propiciar condições para o sistema se adaptar de acordo com as necessidades ou circunstâncias. Evoluções de desempenho são aplicadas pontualmente nas funcionalidades que necessitam. Correções de comportamento, atualização de produtos, mudanças no *código fonte*, entre outros ajustes, ocorrem no serviço alvo sem afetar os demais serviços.

A diversidade de soluções técnicas é uma situação propícia para sistemas projetados sob *microserviços*. Tendo como exemplo, um *sistema de rede social* que: armazenar imagens em banco de dados orientado a gráficos; usar banco de dados relacional para lista de contatos; e armazena as mensagens trocadas entre usuários por meio de base de dados orientado a documentos. No exemplo visto, cada serviço possui necessidades distintas e por consequência soluções também diferentes. A Figura 2.2. ilustra a arquitetura do sistema descrito.

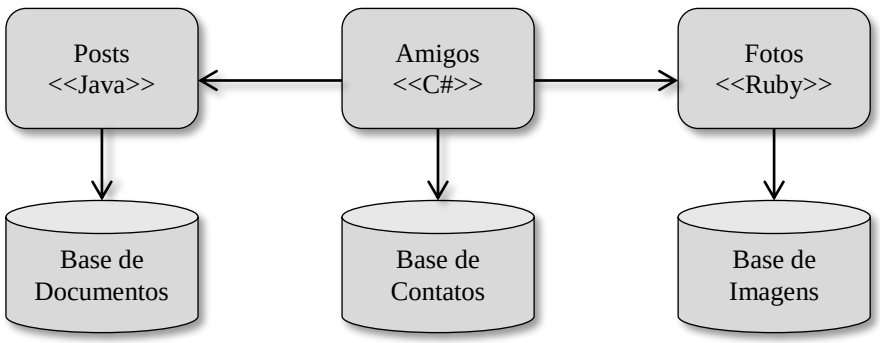

Figura 2.2: Solução de *microserviços* sob tecnologias heterogêneas.

O modelo de negócio da indústria de software depende da constante busca pela inovação. Essa necessidade vai ao encontro da arquitetura de *microserviços* que proporciona a adoção de novas tecnologias com mais velocidade, permitindo que equipes compreendam como novas soluções podem ajudá-los. Uma das maiores barreiras para experimentar e adotar uma nova tecnologia são os riscos associados a mudança. Em grandes sistemas monolíticos, mudanças do tipo: novas linguagens de programação; novos modelos de dados; ou novas interfaces com o usuário, têm impacto

significativo em diversas partes do sistema. Porém quando bem modularizados, há espaço para experimentar alternativas tecnológicas em um determinado módulo e planejar com mais assertividade a mudança dos demais. Uma estratégia para experimentar mudanças é escolher serviços com menor risco e progressivamente ampliar para serviços mais críticos. Muitas organizações consideram a capacidade de absorver mais rapidamente novas tecnologias como a principal vantagem da abordagem de *microserviços*.

A disponibilidade e escalabilidade de um sistema pode determinar o sucesso de um negócio. Porém, o custo para elevar a qualidade de um serviço deve ser avaliado, pois envolve investimentos em infraestrutura e mudanças estruturais, ainda assim, sem garantias de que o esforço será bem-sucedido. Um grande sistema monolítico é escalado por inteiro. Mesmo que a necessidade de desempenho e disponibilidade seja para algumas funcionalidades mais críticas, o sistema é tratado como uma única peça. O dimensionamento de memória, processamento, disco ou replicação de instâncias é feito para todas as funcionalidades do sistema. Essa abordagem resulta em alto custo e provável desperdício de recurso, pois nem todas as funcionalidades do sistema precisam da mesma infraestrutura. A arquitetura de *microserviços* racionaliza custos, pois apenas os serviços que dependem de escalabilidade e alta disponibilidade são redimensionados. Serviços de menor criticidade podem executar suas atividades em ambientes de menor custo. A Figura 2.3. ilustra uma estratégia de escalabilidade em função da necessidade de desempenho de cada serviço.

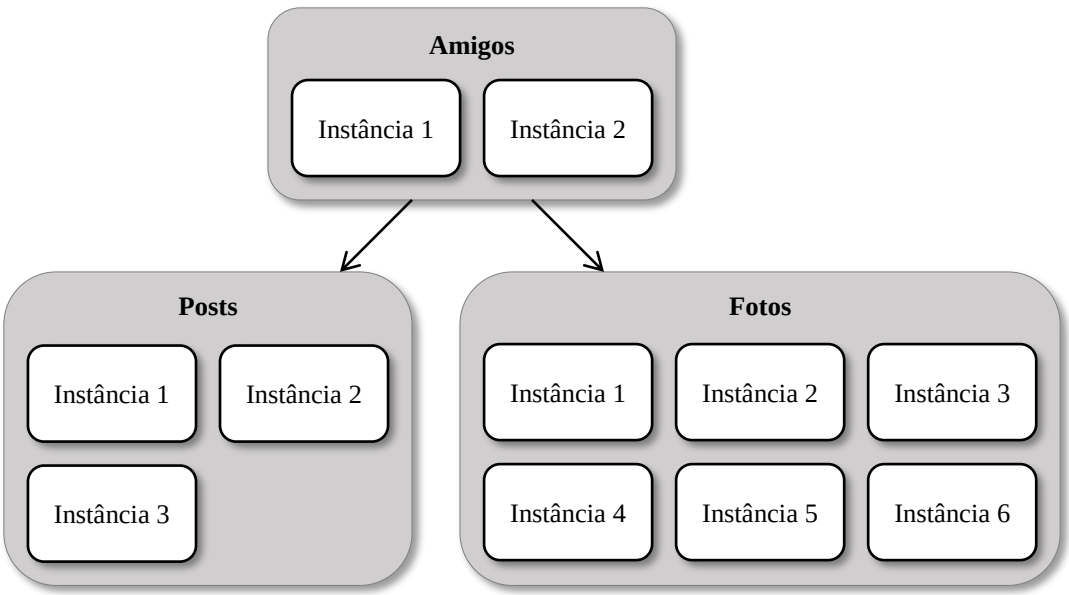

Figura 2.3: Escalabilidade direcionada ao tipo de problema

Alguns princípios podem auxiliar na escolha de uma arquitetura monolítica ou orientada a *microserviços*. É natural que cada organização, a partir das suas experiências, estabeleçam seus próprios princípios, porém Sam Newman (2015) destaca alguns princípios chaves, ilustrados na Figura 2.4.

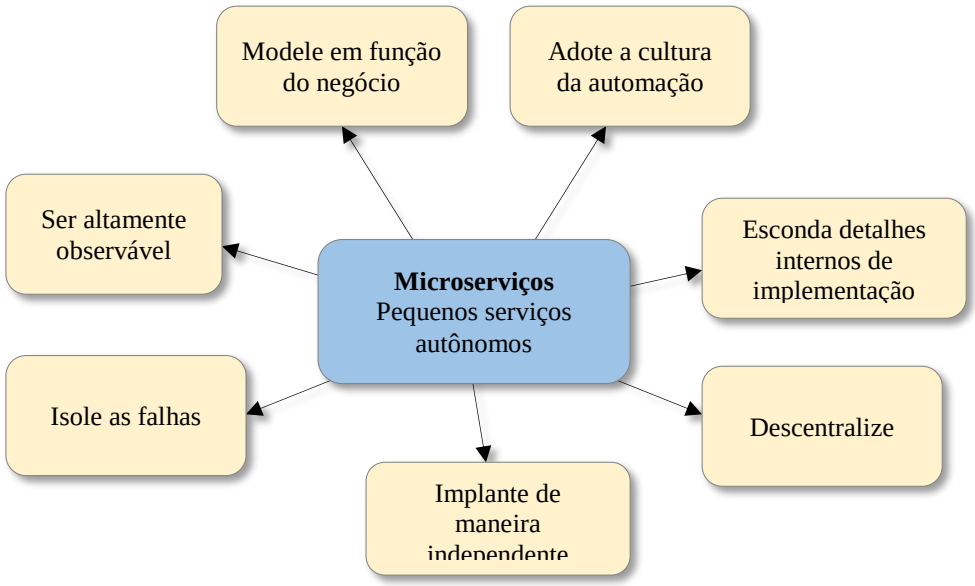

Figura 2.4: Princípios dos *microserviços* (Newman, 2015).

Modele em função do negócio – As interfaces dos serviços, quando modeladas em função das necessidades de negócio, são mais estáveis em relação àquelas modeladas a partir de conceitos técnicos. Deve-se modelar o domínio no qual o sistema operará, além de formar interfaces mais estáveis, também criará melhores condições para refletir mudanças nos processos de negócio.

Adote a cultura da automação – Os *microserviços* adicionam complexidade ao ambiente no qual eles habitam. Atividades de implantação, testes e monitoração de serviços tendem a gerar esforços significativos à medida que os *microserviços* são pulverizados no ambiente. Abraçar uma cultura de automação é a chave para mitigar esse tipo de problema. Automação de testes é essencial, pois garante que os serviços continuarão funcionando, de toda forma, a automação se torna mais complexa quando comparada aos sistemas monolíticos.

Esconda detalhes internos de implementação – Para aumentar a capacidade de evolução dos serviços de forma independente de quaisquer outro é necessário ocultar detalhes de sua construção. A modelagem delimitada por contextos de negócio pode

ajudar, pois essa técnica permite destacar os modelos que devem ser compartilhados daqueles de devem ser escondidos. Os serviços também devem esconder suas bases de dados para evitar o tipo mais comum de acoplamento, feito por bases compartilhadas, comuns em arquiteturas tradicionais orientadas a serviços. Deve-se considerar o uso de REST (***Re****presentational* ***S****tate* ***T****ransfer*) para formalizar as interfaces expostas e usar as chamadas RPC (***R****emote* ***P****rocedure* ***C****alls*) para interfaces internas que devem ser escondidas.

Descentralize – Para ampliar a autonomia dos *microserviços* deve-se constantemente observar as oportunidades para delegar a tomada de decisão e o controle para a equipe de detém o próprio serviço. Este processo é iniciado quando a equipe detentora do serviço tem a responsabilidade de testar e implantar o software sob demanda, sem a necessidade de outras equipes para realizar essas tarefas. Esse princípio também se estende à arquitetura de um *microserviço*. Evitar abordagens como sistemas de ESB (***E****nterprise* ***S****ervice* ***B****us*) ou *orquestração*, pois direciona a centralização de lógica de negócio. Em vez disso, deve-se preferir uma arquitetura baseada em *coreografia*, na qual a decisão da ordem na execução dos serviços é estimulada por eventos gerados pelos próprios serviços. Por coesão, a lógica e os dados devem ser associados dentro das fronteiras do serviço.

Implante de maneira independente – Deve-se buscar implantar um *microserviço* sem interferência ou dependência diretamente de outro. Mesmo que seja necessário fazer alterações significativas na implementação, deve-se procurar manter as funcionalidades antigas para que os consumidores tenham tempo para realizar suas migrações. Este princípio permite otimizar a velocidade de liberação de novos recursos e aumentar a autonomia das equipes que dependem desse serviço, garantindo que eles não precisam modificar frequentemente suas implementações.

Isole as falhas – Uma arquitetura de *microserviços* deve ter mais resiliência quando comparado aos sistemas monolíticos. Os problemas de comunicação devem ser previstos e planejados para cada serviço. Ao construir um módulo, deve-se levar em conta o fato de que suas chamadas externas podem falhar. Sem essa percepção, o sistema inteiro poderá sofrer falhas em cascata se tornando mais frágil que um sistema monolítico. O tratamento das chamadas remotas deve ser feito com o mesmo cuidado que tradicionalmente se faz nas chamadas locais.

Ser altamente observável – Em ambientes amplamente distribuídos não é permito confiar na observação do comportamento de uma única instância do serviço ou

no *status* de um único servidor para garantir o pleno funcionamento do sistema. Em vez disso, deve-se buscar uma visão mais abrangente de tudo que está acontecendo. O monitoramento precisa abranger elementos com maior significado semântico para inferir se o sistema como um todo está se comportando corretamente. Uma técnica de monitoração pode usar simuladores que agem conforme o comportamento de um usuário real. Agregar logs e dados estatísticos, podem facilitar a detecção de comportamentos inesperados.

Após breve discussão sobre os princípios dos *microserviços*, é possível fazer correlação entre essa nova arquitetura da indústria de software com os sistemas multiagentes. Cerca de quatro décadas separam as primeiras literaturas sobre os sistemas multiagentes das referências aos *microserviços*. Apesar dos agentes existirem há muito tempo, as tendências atuais para uso de arquiteturas distribuídas e inteligentes podem ser analisadas sob a perspectiva de um agente. Um *sistema multiagente* pode ser entendido como um ambiente mais sofisticado em relação aos *microserviços*. A Figura 2.5 ilustra um comparativo entre os dois conceitos.

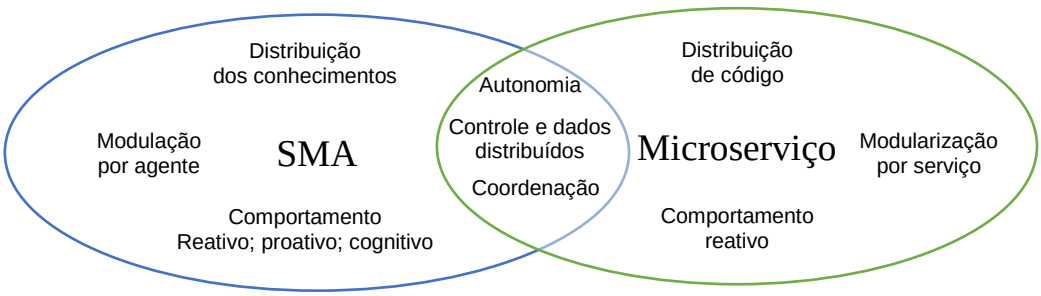

Figura 2.5: Correlação entre *microserviços* e os sistemas *multiagentes*.

Diante de um ambiente com aspectos semelhantes, onde o controle e os dados são distribuídos, a autonomia é o principal elo conceitual entre os sistemas multiagentes e os microserviços, além disso, ambas abordagens necessitam fortemente de mecanismos de cooperação, entre seus pares, para resolução de problemas que por si sós não seriam capazes de resolver.

Uma diferença fundamental entre os *microserviços* e os *sistemas multiagentes* é o comportamento contextual que um agente pode exibir. Um agente é orientado a objetivos e tenta cumpri-los tomando decisões baseadas em diversos fatores, como o conhecimento sob suas próprias habilidades, seu ambiente e outros agentes que estão presentes em determinado momento. Em outras palavras, o agente visa compreender seu contexto e tomar decisões que o leva ao melhor caminho para atingir seus objetivos.

A *arquitetura de microserviços* pode ser vista como uma evolução do SOA (***S**ervice-**O**riented **A**rchitecture*), porém sob uma granularidade mais fina. De modo geral os *microserviços* têm comportamentos predominantemente reativos e são organizados visando distribuir os códigos e as lógicas de negócio; com menor preocupação quanto a distribuição do conhecimento ou da capacidade cognitiva.

2.2 Modelos de Confiança

Um agente inteligente, conforme apresentado nas seções anteriores, possui limitações que reduzem sua capacidade de interação, particularmente em ambientes de grande escala. Além disso, o próprio ambiente pode apresentar restrições de conectividade ou velocidade dos canais de comunicação que interferem na capacidade sensorial dos agentes. Portanto, é tecnicamente improvável aos agentes obterem um estado de informação perfeito acerca do seu ambiente e dos demais indivíduos que habitam nele. Apesar do alto grau de incerteza, característico desses ambientes, os agentes devem continuar suas interações sem perder de vista um nível aceitável de segurança. A partir desse cenário, surge naturalmente a necessidade de *confiar* uns nos outros, no intuito de mitigar impactos negativos decorrentes das interações entre indivíduos completamente ou parcialmente desconhecidos.

O termo *confiança* possui definições distintas de acordo com cada área de estudo, no entanto algumas definições podem ser compreendidas e analisadas de maneira complementar. A confiança pode ilusoriamente ser dispensável na existência de contratos rigorosos com punições severas em caso de descumprimentos. Entretanto, por mais elaborado que seja um acordo, não é possível detalhar todas as eventualidades. Desta forma, a confiança representa as *expectativas* sobre o que os outros farão ou que fizeram ou sobre o que eles transmitirão em suas mensagens em circunstâncias que não estão explicitamente incluídas em seus acordos ou contratos (Dasgupta, 2000). A confiança é antes de tudo um *estado mental*, uma *atitude* (Castelfranchi e Falcone, 1998). Confiança é a *crença* quantificada de honestidade, veracidade, competência e segurança de alguém sobre outra pessoa ou serviço (Sloman, 2004).

Quando a percepção da confiança passa de uma crença pessoal para ter abrangência coletiva, surge o fenômeno da *reputação*. A reputação pode ser entendida como uma coleção de opiniões vindas de outros usuários que produzem uma expectativa de comportamento baseado nas interações anteriores de outros (Abdul-Rahman e Hailes, 2000). Uma cidade “mal falada” pode inibir substancialmente o

turismo e impactar a economia daquela região, inclusive de cidades vizinhas. As crenças transmitidas de pai para filho podem motivar guerras recorrentes entre povos. Críticas inflamáveis em redes sociais podem resultar em grandes manifestações sob um *comportamento de manada*. As opiniões de uma sociedade sobre alguém, outra sociedade ou uma organização são elementos formadores de reputações capazes de influenciar desde grupos virtuais, mercados econômicos, chegando a conflitos étnicos e religiosos (Falkenreck, 2011).

Para sistemas multiagentes de larga escala, a reputação é essencial, na medida em que procurasse evitar interações diretas potencialmente prejudiciais ou desnecessárias. Por meio da reputação, por exemplo, agentes consumidores de serviços podem selecionar melhores provedores, consequentemente, é possível aperfeiçoar de maneira global a comunidade, em razão dos agentes comportarem-se melhor a fim de evitar a perda de futuras negociações por conta de uma má reputação (Dellarocas, 2003). Os *sistemas de confiança* surgem com a finalidade de coletar, distribuir e agregar *avaliações* sobre o comportamento passado dos seus participantes (Resnick, 2000). Outros objetivos podem ser obtidos em um modelo de confiança: promover formas para incentivar a produção de testemunhos sobre outros agentes; promover estruturas de dados para coletar e armazenar as avaliações produzidas; e promover mecanismos para recompensar testemunhos conforme nível de acerto.

2.2.1 Revisões sobre Modelos de Confiança

Esta seção apresenta as principais revisões da literatura sobre modelos de confiança. Por se tratar de tema multidisciplinar com diferentes perspectivas, as revisões selecionadas abordam especificamente modelos de confiança aplicados à *Ciência da Computação*. Cada revisão da literatura possui diferenças quanto à estruturação e classificação dos conceitos. Para a fundamentação teórica deste trabalho, foi utilizado um conjunto de características denominadas de *dimensões*. As dimensões são importantes na classificação dos modelos de confiança, pois cada dimensão ressalta um problema que pode ser tratado em um modelo. A escolha do modelo de confiança depende da análise entre os problemas e as dimensões que cada modelo se propõe a atender. A Tabela 2.1 apresenta de forma resumida, cinco das principais revisões sobre os modelos de confiança e as dimensões consideradas em cada trabalho.

Tabela 2.1: Revisões e dimensões.

1) Grandison e Sloman 2000	2) Ramchurn *et al*. 2004	3) Sabater e Sierra 2005	4. Lu *et al*. 2009	5. Pinyol e Sabater 2013
Acesso a recursos	Nível Individual	Modelo Conceitual	Dimensão	Confiança
Acesso a serviços	- Modelos cognitivos	Fontes de Informação	Semântica	Cognitivo
Certificação Digital	- Modelos de reputação	Visibilidade	Arquitetura	Procedural
Delegação	- Aprendizagem	Granularidade	Modelo	Generalidade
Infraestrutura	Nível Global	Comportamento	Redes de Confiança	
	- Protocolos de Interação	Tipo de Informação	Confiabilidade	
	- Mecanismo de Reputação	Confiabilidade	Risco	
	- Mecanismos de Segurança			

Conforme observado na Tabela 2.1, cada revisão aborda diferentes dimensões para classificar os modelos de confiança. A revisão de Grandison e Sloman (2000) se diferencia pela abordagem da segurança da informação, trata conceitos como autenticação, autorização e criptografia visando tornar a infraestrutura dos sistemas mais confiáveis. Sarvapali Ramchurn *et al*. (2004) discute diferenças entre abordagens distribuídas, chamadas de *nível local*, e modelos centralizados, *nível global*. Sabater e Sierra (2005) apresentam uma rica classificação que abrange múltiplos aspectos. Gehao Lu *et al*. (2009) enfatiza tendências futuras, dentre elas, a maior capacidade de intepretação semântica das informações de confiança e a utilização de padrões e ferramentas para construção de sistemas multiagentes. Finalmente, Pinyol e Sabater (2013) atualizam a revisão de Sabater e Sierra (2005), acrescentaram comparações para os trabalhos propostos até aquele momento.

A partir das revisões analisadas, observou-se divergências na forma como os modelos de confiança são classificados: a revisão de Sabater e Sierra (2005) não trata a dimensão *infraestrutura*, citada por Grandison e Sloman (2000), nem considera o *risco* apresentado por Lu *et al*. (2009); Pinyol e Sabater (2013) não aborda a dimensão de *nível individual* e *global*, porém Ramchurn *et al*. (2004) a considera; algumas dimensões podem se interpretadas como semelhantes, porém são descritas com termos distintos, como no caso da *visibilidade*, descrita por Sabater e Sierra *(2005)* e os *níveis individuais e globais* tratados por Ramchurn *et al.* (2004) em que ambas as revisões tratam a mesma problemática: o tratamento da confiança de modo *centralizado* ou *distribuído*.

Diante do número de divergências entre as revisões, este trabalho propõe a unificação das dimensões mais relevantes sob a ótica das cinco revisões avaliadas e das citações em outros trabalhos examinados neste estudo. A

Tabela 2.2 apresenta as dimensões selecionadas e as respectivas revisões que fazem referência a elas.

Tabela 2.2: Quadro das dimensões selecionadas

Dimensão	Abreviatura	Significado	Revisões Consideradas
Paradigma	N	Numérico	(Sabater e Sierra, 2005) e (Pinyol e Sabater, 2013)
	C	Cognitivo	
Fonte de Informação	ID	Interação Direta	(Sabater e Sierra, 2005); (Ramchurn *et al*. 2004) e (Pinyol e Sabater, 2013)
	OD	Observação Direta	
	TE	Testemunhos	
	SO	Sociológica	
	PR	Preconceito	
	RC	Reputação Certificada	
Visibilidade	L	Local	(Sabater e Sierra, 2005)
	G	Global	
Contexto	U	Único contexto	(Lu *et al*. 2009) e (Sabater e Sierra, 2005)
	M	Múltiplos contextos	
Suposição de Comportamento	0	Ambiente honesto	(Sabater e Sierra, 2005)
	1	Omissão	
	2	Mentira	
Segurança da Informação	ST	Sem Tratamento	(Grandison e Sloman, 2000); (Pinyol e Sierra, 2013)
	AR	Acesso a Recursos	
	PS	Prestação de Serviços	
	EC	Entidades Certificadoras	
	DE	Delegação	
	IN	Infraestrutura	

Nesta seleção há cincos dimensões originárias da revisão de Sabater e Sierra (2005) e utilizadas nas revisões de Pinyol e Sabater (2013) e Gehao Lu *et al*. (2009). Além dessas foi considerada a dimensão, *segurança da informação*, de Grandison e Sloman (2000), pois é um dos temas centrais tratado neste trabalho. A Figura 2.6 apresenta as dimensões selecionadas que compõem a seguinte taxonomia:

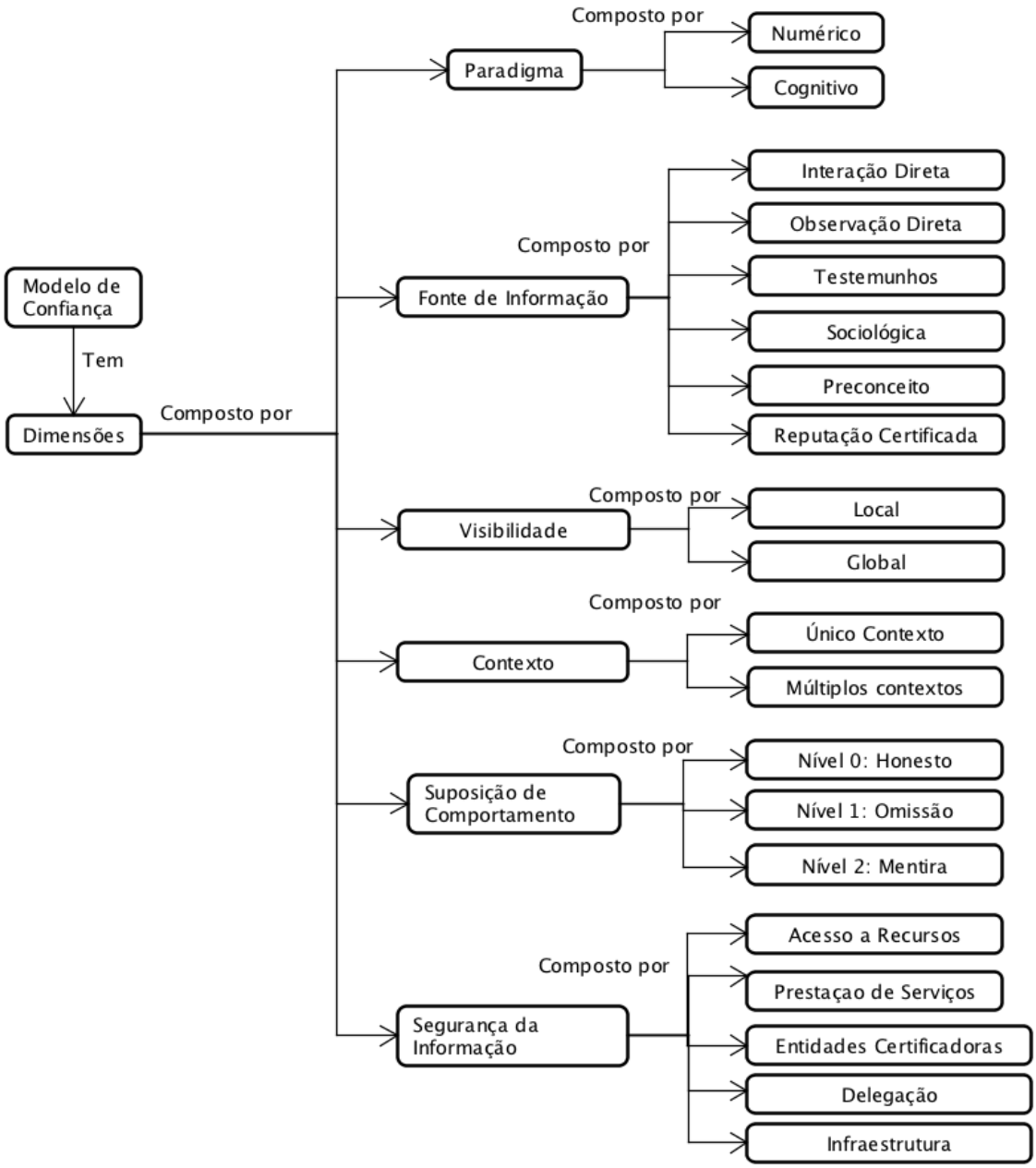

Figura 2.6: Taxonomia para modelos de confiança.

A partir da classificação proposta, as próximas seis seções detalham cada uma das dimensões selecionadas, apresenta a respectiva fundamentação teórica, equaliza divergências de conceitos ou nomenclaturas entre as cinco revisões analisadas e apresenta os principais trabalhos relacionados a cada dimensão.

2.2.2 Paradigma

De acordo com Sabater e Sierra (2005), os modelos de confiança podem ser classificados em função de dois paradigmas: *cognitivo* e *matemático*. Os modelos cognitivos tratam a confiança como uma convicção derivada de um conjunto de crenças.

Ela é um estado mental, logo apenas indivíduos cognitivos podem confiar (Castelfranchi e Falcone, 1998). Modelos de confiança cognitivos geralmente utilizam a arquitetura BDI (*Belief, Desire, Intention*) para modelar o sistema. De modo geral, agentes BDI representam processos internos por meio de *estados mentais* e definem mecanismos de controle para selecionar suas ações (Bratman, 1987).

O paradigma *matemático* interpreta a confiança como uma *probabilidade* de que um indivíduo se comportará de acordo com parâmetros estabelecidos. De modo geral, os modelos deste paradigma utilizam agregações numéricas sobre interações passadas e apresentam a probabilidade de um agente desempenhar corretamente seus comportamentos.

> "A principal vantagem do raciocínio probabilístico sobre o raciocínio lógico é o fato de que os agentes podem tomar decisões racionais mesmo quando não há informação suficiente para se provar que uma ação funcionará." (Charniak, 1991)

Grande parte dos modelos numéricos utilizam métodos estatísticos para calcular a confiança: probabilidades bayesianas (Regan *et al.* 2006); lógica fuzzy (Yu e Singh, 2003); distribuições de probabilidades (Sen e Sajja, 2002); funções Dempster-Shafer (Liu *et al.* 2012); entre outras. Técnicas de aprendizagem de máquina também são aplicadas para auxiliar na tomada de decisão dos agentes, tais como algoritmos de árvores de decisão e aprendizagem por reforço (Tran e Cohen, 2004).

Modelos Cognitivos

Dentre os primeiros modelos cognitivos destaca-se a proposta de Falcone e Castelfranchi (2001), a principal referência deste paradigma. O trabalho contribui na definição do termo confiança e sua importância do ponto de vista de um agente. Além de confiança, o termo *delegação* é enfatizado como uma *ação*. E o resultado de uma decisão que depende diretamente do agente confiar e consequentemente delegar ações para outros. As diferentes formas de relacionamento entre os agentes, dependem da capacidade de delegação, i.e. da confiança. Isso torna a confiança um conceito intimamente ligado aos agentes cognitivos.

O *framework sócio cognitivo CSCEF* (Neville e Pitt, 2004) utiliza uma abordagem de raciocínio composta por elementos de confiança, reputação, recomendação e aprendizagem. Esses elementos são derivados de experiências diretas.

As relações sociais são estabelecidas entre agentes consumidores e fornecedores participantes de um ambiente de mercado eletrônico.

Outro modelo aplicado ao comércio eletrônico propõe a seleção de parceiros por meio do conceito de *familiaridade* entre agentes (Zhang *et al*. 2007). Esta abordagem considera quatro atributos cognitivos: experiências próprias; exposição repetitiva; nível de processamento; e taxa de esquecimento. A Figura 2.7, ilustrada a especialização da familiaridade para os fatores principais e suas respectivas representações no comércio eletrônico.

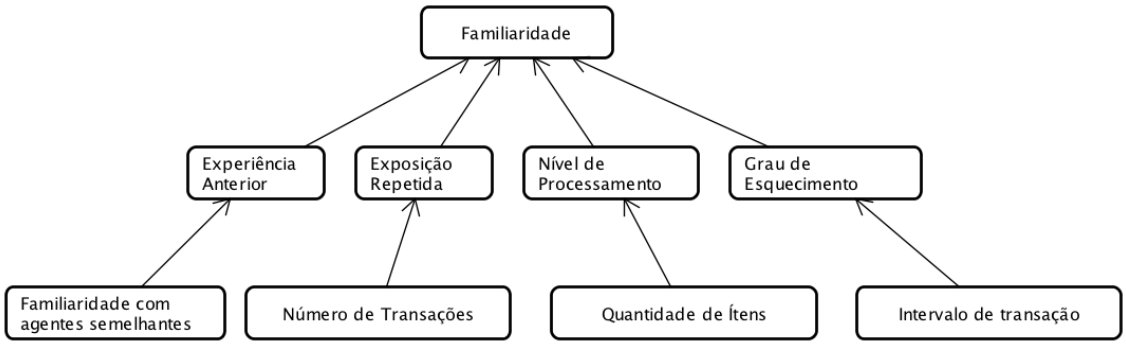

Figura 2.7: Atributos cognitivos para familiaridade (Zhang *et al*. 2007).

Além da familiaridade, há outras abordagens para estabelecer níveis de relacionamento entre agentes. A percepção de confiança também pode ser representada por meio de *conectores arquiteturais* (Singh, 2011). Nesta proposta, agentes BDI constroem, a partir de crenças, suas conexões de afinidade com outros agentes. Além disso, o modelo apresenta formalização semântica para confiança.

O UnCM (**Un**informed **C**ognitive **M**aps) (Piunti, 2012) propõe heurísticas cognitivas que permitem aos agentes obter um grau de confiança a partir de fontes de informações heterogêneas. Uma derivação do UnCM, chamada de LFCM (Venanzi, 2011), agrega mecanismos de aprendizagem que permitem ao modelo adaptar-se utilizando uma série de relações observáveis entre os agentes e suas habilidades para cumprir tarefas em determinadas condições.

No intuito de mostrar a importância dos modelos de confiança baseado na abordagem BDI, Castelfranchi *et al*. (2003) propôs um modelo sócio cognitivo construído por meio de mapas de fuzzy—*Fuzzy Cognitive Maps* (Kosko, 1986).

Modelos Numéricos

A inferência estatística é uma técnica comumente utilizada nos modelos de confiança. Um exemplo é o uso da *probabilidade bayesiana*, útil para representar incertezas. O modelo BLADE (Regan *et al.* 2006), por exemplo, utiliza *redes bayesianas* para aprender o comportamento dos agentes sobre suas avaliações e *redes bayesianas dinâmicas* para detectar mudanças no comportamento dos agentes ao longo do tempo. O projeto *HABIT – **H**ierarchical **A**nd **B**ayesian **I**nferred **T**rust –* (Teacy *et al*. 2012) também utiliza *redes bayesianas* para avaliar o quanto um agente pode confiar em seus pares, sua principal contribuição é a capacidade de lidar com diferentes fontes de informação e contextos. A função de densidade sob a probabilidade condicional bayesiana é utilizada no sistema SARC – ***S**ubjectivity **A**lignment for **R**eputation **C**omputation* – que sugere correlacionar as avaliações dos agentes, por meio da captura de atributos subjetivos de agentes compradores sob um ambiente de comércio eletrônico (Fang *et al*. 2012).

Além da *probabilidade bayesiana*, há alternativas matemáticas para determinar a percepção de confiança. A lógica *difusa* admite valores lógicos intermediários entre o *falso* (0) e o *verdadeiro* (1), por exemplo, um valor *quase falso* (0,02) ou um valor *quase verdadeiro* (0,99). Desse modo, é possível tratar problemas de incerteza avaliando conceitos como: sensação de temperatura (quente, frio, agradável...) ou a confiança (desonesto, confiável, muito confiável, regular...). Por meio da lógica *difusa* é possível atribuir níveis de desempenho dos agentes (Sen e Sajja, 2002). A detecção de agentes fraudulentos, ilustrado na Figura 2.8, é um problema tratável pela lógica *difusa* (Schillo *et al*. 2000).

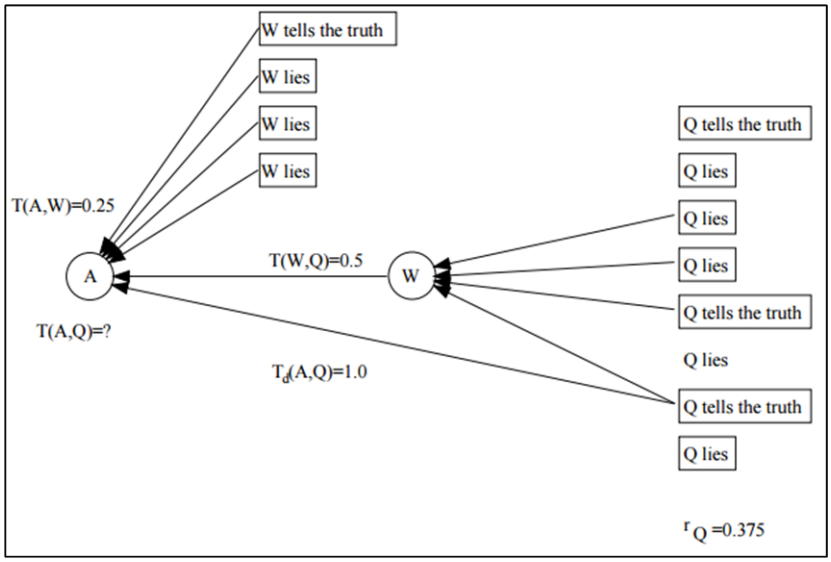

Figura 2.8: Lógica *difusa* para representação de confiança (Schillo *et al*. 2000)

A teoria matemática da evidência *Dempster-Shafer* determina graus de credibilidade combinando evidências de diferentes fontes (Shafer, 1976). Esta teoria pode ser utilizada em modelos de confiança como uma estrutura computacional para combinar funções de crenças dos agentes, sejam *locais* ou *globais* (Yu e Singh, 2003) e (Liu *et al*. 2012).

Técnicas de *aprendizagem de máquina* também são utilizadas para auxiliar na tomada de decisão dos agentes, trabalhos como (Tran e Cohen, 2004) propõem um modelo de confiança utilizando algoritmos de *aprendizagem por reforço*. Além disso, existem também os modelos *híbridos* que utilizam os dois paradigmas, como por exemplo, os trabalhos de (Matt *et al*. 2010) (Bentahar *et al*. 2007) (Koster *et al*. 2012) e (Parsons, 2011). Esses modelos são considerados cognitivos por utilizarem argumentos baseados em crenças e numéricos por fazerem uso de agregações numéricas. Seguindo uma linha semelhante, Griffiths e Luck (2003) utilizam a arquitetura de agentes BDI para a formação de coalizões.

2.2.3 Fonte de Informação

Diversos trabalhos citam duas grandes fontes: as *experiências diretas* e as *experiências indiretas* por meio de testemunhos. Entretanto, outras fontes de informação são citadas como as *sociais* e o *preconceito*. Cada fonte de informação requer capacidades sensoriais específicas dos agentes. A combinação de várias fontes de informação pode, quando integradas, aumentar a confiabilidade dos valores calculados. Entretanto, a

utilização de múltiplas fontes aumenta a complexidade do modelo e exige maior capacidade computacional dos agentes.

1) Experiências Diretas

A *interação direta* é caracterizada, quando um agente A interage com um agente B e, a partir do seu próprio entendimento, A infere um valor de confiança sobre B. Neste mesmo cenário um agente C observa a interação entre A e B, a partir da *observação direta*, C infere um valor de confiança sobre A e B. Nas duas situações, seja interagindo ou observando, os agentes dependem exclusivamente de si para interpretar os fatos e deduzir o comportamento dos seus parceiros.

As interações diretas trazem como vantagem a personalização das informações, à medida que os agentes avaliam seus parceiros a partir das suas próprias impressões. Entretanto, as experiências diretas resultam em sérios riscos, dada a necessidade dos agentes precisarem interagir com frequência e sob um conjunto representativo de outros parceiros para calcular a confiança, além de ser tecnicamente inviável em comunidades de grande escala.

A observação direta é uma fonte de informação com menor risco; à medida que os agentes podem aprender com as experiências positivas e negativas dos outros. Entretanto, para possibilitar a observação, as interações devem ter caráter público e acessível pelos demais agentes (Sabater e Sierra, 2005), tal situação pode ser inviável em certos sistemas.

O modelo de (Marsh, 1994) é um dos primeiros modelos de confiança propostos na literatura e caracteriza-se por utilizar exclusivamente as experiências diretas para calcular a confiabilidade dos agentes. De forma similar, o modelo MDT (Griffiths, 2005) faz uso apenas das interações diretas na seleção de parceiros para delegar tarefas, enquanto o modelo Ghanea-Hercock, (2007) utiliza as experiências diretas na escolha de agentes mais apropriados para a formação de coalizão. São poucos os modelos que empregam apenas a interação direta para avaliar a confiabilidade dos agentes. A grande maioria dos modelos utiliza experiência diretas de forma conjunta com outras fontes para obter maior precisão nas avaliações.

Os modelos (Carter e Ghorbani, 2003), IHRTM (Rettinger *et al*. 2008), (Sierra, Debenham, 2005) e (Klejnowski *et al*. 2010) realizam a observação direta por meio de características perceptíveis como: a quantidade de feedbacks emitidos ou recebidos pelo agente; o percentual de avaliações positivas; histórico de ações anteriores; papéis dos

agentes; e resultado da execução das tarefas. Similarmente, o modelo de (Teacy *et al*. 2008) observa o resultado dos serviços prestados, com isso é capaz de atualizar suas crenças em relação ao provedor do serviço. Os modelos de (Rehak *et al*. 2005) e (Zheng *et al*. 2006) propõem funções de utilidade sob interações passadas para avaliar e selecionar parceiros. O trabalho de (Regan *et al*. 2006) mostra como a observação direta pode ser compartilhada entre agentes em cenários de negociação sob ambiente de *e-commerce*. O trabalho de (Serrano *et al*. 2012) faz a captura e análise das mensagens que trafegam sob um SMA com o intuito de extrair dados para a composição da confiança do agente, utilizando aprendizagem de máquina. Nessa abordagem, a confiança é construída a partir do modelo de conversação dos agentes.

2) Testemunhos

Também chamada de *informação indireta*, o testemunho surge a partir do compartilhamento das experiências dos agentes, seja por meio das interações diretas ou repassando informações recebidas de outros. A *experiência direta* é a fonte mais valiosa para personalização da percepção de confiança, o *testemunho* é a mais abundante e de menor risco, entretanto esta fonte de informação agrega complexidade aos modelos dada a incerteza que circunda sobre as testemunhas. Em cenários de competição, por exemplo, as testemunhas podem omitir suas experiências ou mentir sobre elas visando algum benefício. Em casos como este, os modelos de confiança têm o desafio de promover mecanismos de recompensa que incentivem o compartilhamento e evitem atitudes desonestas. O *Dilema do Prisioneiro* (Kreps, 1982) é clássico problema sobre confiança que prever recompensas pela cooperação mútua dos agentes e punições pela desistência mútua. Outro desafio é a identificação de *conluio* das testemunhas, quando grupos de agentes se reúnem para gerar testemunhos com a finalidade de modificar a reputação de um indivíduo ou de um grupo. O *conluio* se torna mais comum a medida em que as organizações se tornam mais complexas, por exemplo, quando há vários níveis hierárquicos (Tirole, 1986).

Em suma, a confiança é calculada por meio da agregação de opiniões provenientes de agentes que tiveram algum contato com o avaliado (Koster *et al*. 2012). Em (Sen e Sajja, 2002) a confiança é representada por probabilidade a partir dos testemunhos, além disso o trabalho trata a confiabilidade das testemunhas. Há dois tipos de fontes de informação nesta categoria, que são (i) a transmissão simples de avaliações de testemunhos e (ii) as recomendações ou opiniões. No primeiro caso, os relatórios de

testemunhos são obtidos por meio da consulta de diversos agentes, mas os agentes que transmitem as informações não são avaliados. Há diversos modelos que simplificam sua estratégia e desconsideram a confiabilidade das testemunhas (Jsang e Ismail, 2002) e Travos (Teacy *et al*. 2006), entretanto há situações em que a recomendação ou opinião deve ser considerada a partir de agentes confiáveis, i.e. uma rede de relacionamento entre testemunhas (Montaner *et al*. 2002). Essa rede, denominada *Trust Net*, foi proposta por (Yu e Singh, 2003) e visa avaliar a honestidade das testemunhas utilizando a *teoria da evidência* para calcular a possibilidade de uma testemunha enviar relatos falsos. Há outros exemplos com mesma finalidade TRUMMAR (Derbas *et al*. 2004) e (Wang e Zhang, 2005).

Assim como foi dito em relação a interação direta, há poucos modelos que fazem uso exclusivo desta fonte de informação, sendo em geral, combinada com outros tipos de informação. Um exemplo de sistema que pode utilizar tão somente testemunhos são os sistemas de recomendação, no qual agentes devem buscar dados sobre produtos e serviços a partir da informação vinda de outros agentes (Montaner *et al*. 2002) (Bedi *et al*. 2007) (Song *et al*. 2004). Outro modelo baseado puramente em testemunhos é apresentado por (Koster *et al*. 2012), no qual agentes provêm recomendações personalizadas sob uma rede de *amigos* do próprio agente. O modelo de Siyuan Liu *et al*. (2012), trata a veracidade das avaliações enviadas nos testemunhos. Assim como os modelos de (Parsons, 2011), (Bertocco e Ferrari, 2008), iCLUB (Liu *et al*. 2011) e (Liu *et al*. 2013).

3) Social

A *fonte de informação social* representa relações de convivência entre os agentes e suas relações perante a comunidade. As relações podem ter vários propósitos como: dependência, negociação, competição, colaboração entre outras. João Dias e Ana Paiva (2013), por exemplo, descrevem um modelo computacional que trata o conceito de *amizade* para representar o relacionamento entre agentes. Nessa linha, o crescimento das *redes sociais* tem promovido novos trabalhos relacionados as fontes de informações sociais.

Um dos modelos mais representativos em informações sociais é o Regret (Sabater e Sierra, 2012). Tal proposta faz uso de grafos chamados *sociogramas* (Figura 2.9) que indicam a relação entre os agentes. Os *sociogramas* são usados para agrupar os

agentes e obter informações daqueles que são mais representativos, criando a confiança por vizinhança composta pelas ligações entre eles.

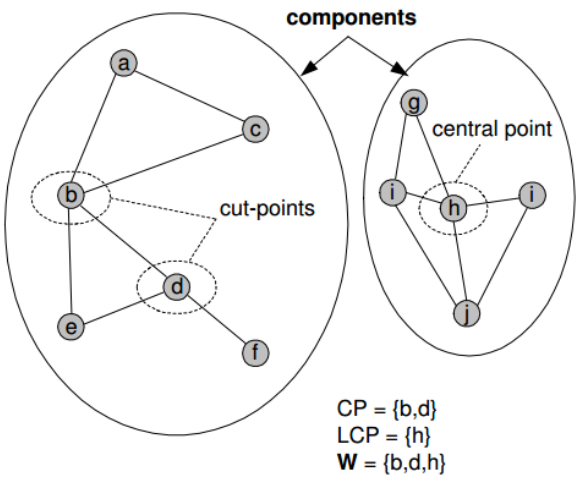

Figura 2.9: Exemplo de *sociograma* do Regret (Sabater e Sierra, 2012)

Os modelos (Sutcliffe e Wang, 2012) e (Liu e Datta, 2012) mostram como o processo de formação da confiança pode ser obtido por meio de estruturas de redes sociais, utilizando medições de amizade, familiaridade, coleguismo e estruturas sociais. Neville e Pitt (2004) propõem um modelo *sócio cognitivo* onde as ações dos agentes, sob o comércio eletrônico, são influenciadas por suas relações sociais. O modelo de (Klejnowski *et al*. 2010) propõe o conceito de *comunidade de confiança* utilizando agentes adaptativos. Sua hipótese considera que agentes pertencentes a mesma comunidade, interagem entre si com menor grau de risco. De forma similar o projeto TRUMMAR (Derbas *et al*. 2004) coleta informações por meio de hierarquias de confiança formadas por estranhos, amigos e vizinhos. O modelo (Li *et al*. 2007) utiliza redes de agentes para localizar testemunhos visando obter referências para seleção de parceiros.

Outra abordagem de coleta em redes sociais é apresentada por Ashri *et al*. (2005), que identifica dinamicamente relacionamentos entre os agentes e posteriormente constrói categorias de relação para inferir a confiança. O modelo *DiffTrust* (Fang *et al*. 2013) considera a *identidade* e o *status* do agente, como fatores relevantes em sua rede social.

4) Preconceito

Apesar da conotação negativa atribuída a palavra *preconceito*, este termo deve ser revisto quando referido às comunidades de agentes. O *preconceito* representa um

conjunto de atributos predefinidos sobre um agente. Cada atributo colabora com a fórmula de cálculo da confiança. Por exemplo, um cargo, um título ou nacionalidade são sinais que identificam o indivíduo como membro de certo grupo (Sabater e Sierra, 2005). O *preconceito* é uma fonte de informação que permite inicializar a confiança quando nenhuma outra informação está disponível. Esse comportamento coincide com a percepção humana. *Estereótipo* também é um conceito relacionado ao *preconceito* e pode ser utilizado com o mesmo propósito (Pinyol e Sabater, 2013).

O modelo de (Burnett *et al*. 2013) aborda a observação de características dos agentes com o objetivo de construir um *estereótipo* do indivíduo. Essas características são dinamicamente construídas a partir da experiência do agente sob diferentes tarefas associadas a técnicas de aprendizagem de máquina. Semelhante a isso, o modelo de (Liu e Datta, 2012) aborda o conceito de "profile" do agente, coletando os dados básicos do indivíduo, tais como idade, gênero e localização. O modelo de (Mokhtari *et al*. 2011) leva em conta os aspectos culturais, de linguagem, nacionalidade e moralidade para integrar o processo de avaliação do agente.

Seguindo outra linha, o modelo proposto por Godo *et al*. (2004), apresenta uma abordagem *institucional*, ele considera que agentes pertencentes a grupos de maior força (instituições legais, governo) possuem maior credibilidade que agentes de grupos mais fracos, tais como pequenas empresas ou indivíduos. O modelo Regret (Sabater e Sierra, 2012) traz o conceito de reputação de sistema, na qual permite definir a confiança inicial para o agente baseado também em estruturas institucionais.

5) Reputação Certificada

A reputação certificada é caracterizada pelo fato do agente avaliado apresentar avaliações sobre si, obtidas a partir das interações com seus parceiros (Huynh *et al*. 2004). Quando um agente A avalia um agente B, B armazena localmente uma referência de A como sua testemunha e armazena a avaliação feita por A. Caso outro agente queira interagir com as testemunhas de B basta fazer uma consulta direta a B. A reputação certificada serve de atalho para conseguir, com poucas interações, um conjunto relevante de testemunhas capazes de mensurar a reputação do agente avaliado.

A reputação certificada pode ser analogamente representada como uma *carta de recomendação*, de um empregador para um ex-funcionário. O empregador registra suas considerações sobre o funcionário que terá posse desta informação por meio de uma

carta impressa. A carta, ou o conjunto de outras cartas, podem ser utilizadas sempre que alguém requisitar informações sobre este indivíduo.

Deve-se assumir que a informação da reputação certificada possivelmente superestima o comportamento esperado de um agente. Assim, embora não se possa garantir o desempenho do agente B em futuras interações, as informações fazem revelar uma perspectiva parcial sobre o comportamento passado do agente B. A principal vantagem da reputação certificada é sua alta disponibilidade (Dong-Huynh *et al*. 2004).

Apesar do compartilhamento da informação ser uma alternativa inovadora, à medida que imputa responsabilidade ao agente avaliado na manipulação das avaliações sobre si, esta abordagem apresenta fragilidades evidentes em ambientes maliciosos, visto o modelo não prever nenhum tipo de tratamento para o comportamento desonesto dos agentes. Em ambientes competitivos, por exemplo, os agentes podem adulterar suas avaliações ou simplesmente omitir avaliações negativas de tal forma que o cálculo da confiança seja manipulado ao seu benefício. Apesar da intenção de prover perceptivas parciais sobre os agentes, a *reputação certificada* pode, na maioria dos casos, resultar em impressões enganosas e prejudiciais para o sistema como um todo.

O modelo FIRE (Dong-Huynha *et al*. 2004) destaca-se por apresentar grande diversidades de fontes de informação. O trabalho propõe a *reputação certificada* como parte do cálculo da confiança dos agentes. O modelo apresentado por (Huynh *et al*. 2006) também utiliza a *reputação certificada*, mas não integra outras fontes de informação. O modelo de confiança proposto em (Botelho *et al*. 2009 e Botelho *et al*. 2011) amplia a *reputação certificada* propondo ao agente avaliado armazenar a referência das suas testemunhas, bem como as avaliações recebidas por elas. Desta forma é possível calcular a reputação com uma única consulta ao agente avaliado. Para evitar fraudes na avaliação, o modelo utiliza recursos de criptografia e assinatura digital.

2.2.4 Visibilidade

Na dimensão visibilidade, a confiança assume duas formas: (i) ser uma propriedade global acessível a todos; ou (ii) ser uma propriedade local, privada e subjetiva que cada agente constrói para si (Pinyol e Sabater, 2013). A visibilidade é uma das dimensões mais abordadas e objeto de várias pesquisas. Apesar de certos estudos utilizarem nomenclaturas distintas (e.g. individual, global, centralizado, descentralizado ou local), os significados dos termos são similares. De acordo com Ramchurn *et al*. (2004), a confiança de *nível individual* parte da crença local de cada agente sobre a honestidade

ou reciprocidade de outrem a partir das interações entre seus parceiros, a confiança no *nível de sistema* é derivada a partir de regras, mecanismos e protocolos regulados globalmente pelo sistema. Para Lu *et al*. (2009) há duas abordagens: centralizada e distribuída. Na primeira abordagem, as avaliações dos agentes são armazenadas, classificadas e recuperadas a partir de um serviço central, enquanto que na segunda abordagem todas estas tarefas são realizadas pelos próprios agentes.

Modelos de confiança globais são largamente empregados em sistemas para *Internet* que atuam sobre milhares ou milhões de usuários. Em ambientes com numerosos usuários, as chances de haver interações repetidas são raras e isto reduz o incentivo dos agentes a cooperarem baseados na expectativa de construir um relacionamento vantajoso (Dellarocas, 2003). Por exemplo, em um sistema de compras *online* que permite múltiplos compradores e vendedores. Em dado momento, um usuário compra uma camisa e no dia seguinte um celular. O número de vendedores que comercializam conjuntamente camisas e celulares é provavelmente pequeno, logo o comprador não se fidelizará com o vendedor de camisas para a sua segunda compra. As experiências pessoais acumuladas dos compradores de camisetas não possuem utilidade para compradores de celulares. A eficiência dos modelos de confiança globais baseia-se no número de opiniões disponíveis para certo indivíduo. O grande número de opiniões mitiga o risco das percepções isoladas e tendenciosas que estatisticamente se tornam irrelevantes.

Em modelos que consideram a confiança como uma propriedade global, o principal desafio é a falta de personalização dos valores. Embora esta abordagem possa ser aceitável em cenários simples, onde é possível atribuir um "modo de pensar" comum para todos os membros da comunidade, não é útil quando os agentes precisam lidar com assuntos mais complexos e subjetivos (Sabater e Sierra, 2005).

Em oposição a proposta centralizada, há modelos que consideram a confiança como uma propriedade subjetiva. Cada agente usa sua experiência pessoal e as experiências compartilhadas por outros agentes, além de outras fontes de informação, como as *informações sociais* e de *preconceito* que contribuem para a confiança sobre cada membro da comunidade. Esses modelos são indicados para indivíduos que interagem com frequência, pois estabelecem fortes laços entre eles.

A visibilidade global centraliza o acesso as informações de confiança por meio de serviços. O exemplo mais emblemático é o eBay (Ebay, 2015), sistema de *e-commerce* que une compradores e vendedores de todas as partes do mundo. Um dos

motivos de seu sucesso foi—para a época—o inovador modelo de reputação que centraliza todos os históricos dos usuários. Para cada avaliação positiva, o comprador ou vendedor recebe 1 ponto, se a avaliação for neutra, 0 pontos e se negativa, -1 ponto. Além disso, o modelo classifica seus usuários por categorias, apresentada pela Tabela 2.3, que identificam facilmente os níveis de confiabilidade. A maioria dos modelos globais utilizam classificadores semelhantes.

Tabela 2.3: Classificação dos usuários por tipo de estrela (Ebay, 2015)

Tipo de Estrela	Quantidade de Avaliações
Amarela	10 a 49
Azul	50 a 99
Turquesa	100 a 499
Roxa	500 a 999
Vermelha	1.000 a 4.999
Verde	5.000 a 9.999
Estrela cadente amarela	10.000 a 24.999
Estrela cadente turquesa	25.000 a 49.999
Estrela cadente roxa	50.000 a 99.000
Estrela cadente vermelha	100.000 a 499.999
Estrela cadente verde	500.000 a 999.999
Estrela cadente prata	1.000.000 ou mais

O modelo proposto por (Jurca e Faltings, 2003) possui uma abordagem híbrida por centralizar a avaliação dos relatórios provindos de testemunhos em uma única entidade, a qual é responsável por avaliar a credibilidade das informações a fim de evitar falsos testemunhos. De forma semelhante, o CORE (Qing-Hua *et al*. 2009) apresenta um mecanismo chamado *SuperAgent* que centraliza as competências e habilidades dos agentes para a formação de coalizões.

Quando trata-se de agentes, de modo geral, há poucos modelos para abordagens centralizadas, algo natural para uma arquitetura que preconiza a descentralização do controle. Ainda assim, há modelos que optam por centralizar como (Bertocco e Ferrari, 2008) que utiliza um gerenciador de contexto responsável por armazenar todas as avaliações de testemunhos. Nesse caso, quando um agente necessita obter ou fornecer uma avaliação, ele deve acessar o gerenciador.

2.2.5 Contexto

A confiança é uma percepção fortemente correlacionada às circunstâncias contextuais. Por exemplo, a reputação profissional de um médico pouco diz sobre suas habilidades culinárias, um policial pode ser um excelente motorista, mas péssimo orador. A confiança depositada sobre os indivíduos geralmente está relacionada às suas características sob determinados contextos. Os modelos de confiança podem considerar um conjunto menor ou maior de contextos.

Griffiths, (2005) denomina estes contextos de *dimensões* e descreve a necessidade de ter modelos que manipulem informações sob múltiplas dimensões. Apesar da evidente importância, a inclusão desta capacidade pode produzir custos em termos de complexidade e gerar efeitos colaterais nem sempre desejáveis. Um modelo *monodimensional* é projetado para associar um único valor de confiança para um indivíduo, sem considerar outros contextos. Ao contrário, um modelo *multidimensional* é composto por um mecanismo que trata simultaneamente diversos contextos de forma que cada um possa receber diferentes valores e pesos distintos. Nem sempre a utilização de várias dimensões é recomendada, principalmente se o modelo enfatiza cenários específicos. Nesse caso, um número reduzido de contextos pode simplificar e ao mesmo tempo ser efetivo na avaliação dos agentes.

O modelo FOCET (Mokhtari *et al*. 2011), por exemplo, utiliza ontologia para representar e reconhecer o contexto. Essa ontologia possui oito dimensões: ambiente, cultura, fatores espaciais e temporais, histórico, subjetividade, perfil do usuário e políticas. Todas as características em conjunto são utilizadas na avaliação do agente. O modelo (You, 2007) utiliza três contextos direcionados ao mercado: tempo decorrido da interação, quantidade e valor monetário negociado. O modelo (Liu e Datta, 2012) utiliza *HMM* (*Hiden Markov Model*) para tratar múltiplos contextos e adiciona informações de preconceito que podem ser acessadas a partir do perfil do usuário. Outros modelos similares, fazem uso de informações de contexto como (Burnett *et al*. 2013), (Wang *et al*. 2002), IHRTM (Rettinger *et al*. 2008), (Bertocco e Ferrari, 2008) e (Rettinger *et al*. 2007). O modelo de (Rettinger *et al*. 2007), introduz o conceito de *transferência da confiança* entre diferentes contextos, a fim de reaproveitar certas informações de um contexto na construção de outro. O modelo de Dondio (Dondio e Barrett, 2007) trata a seleção de evidências de confiança sob o domínio a ser analisado por meio de funções

heurísticas. Um exemplo de seleção de evidências é o projeto *Wikipedia*[4], enciclopédia multilíngue de licença livre, que considera fatores como o número de autores que contribuíram para o mesmo artigo, a gramática escrita de forma correta, tamanho do texto entre outros.

Dentre os modelos *monodimensionais* destaca-se o N. Elgohary *et al*. (2010) que enfatiza ambientes de comércio eletrônico e o modelo de M. R. Mokhtar *et al*. (2007) que simula a confiança em uma organização virtual de construção de produtos. Observa-se o menor número de trabalhos que abordam apenas uma dimensão; isso deve-se ao problema da confiança ser naturalmente dependente de múltiplas características.

2.2.6 Suposição de Comportamento

A dimensão de *suposição de comportamento* auxilia os modelos de confiança a estimar como os agentes se comportarão sob um nível de malícia, ou comportamento desonesto. A partir da suposição, os modelos podem investir mais ou menos em mecanismos que tratem agentes desleais. Esta dimensão é classificada em três níveis de suposição (Pinyol e Sabater, 2013):

- *Nível 0*: o comportamento malicioso não é considerado. Esta abordagem assume que o elevado números de avaliações verdadeiras podem neutralizar, estaticamente, o efeito potencial das avaliações desonestas fornecidas por uma minoria de agentes maliciosos.
- *Nível 1*: o modelo assume que os agentes podem omitir informações, a fim de obter benefício ou evitar alguma penalidade, porém eles nunca mentem.
- *Nível 2*: o modelo supõe que os agentes podem omitir e também mentir, muito comum em comunidades abertas. Nesse caso, há a necessidade de mecanismos que lidem com os dois comportamentos.

O modelo *PRep* (Vogiatzis *et al*. 2010), é de *Nível 1* e tem como característica o tratamento de informações "tendenciosas" ou incompletas. Ao invés de ignorá-las, pois isso poderia acarretar na perda de dados importantes para o cálculo da confiança, o modelo utiliza aprendizagem bayesiana para analisar os testemunhos e compará-los contra as demais interações dos agentes.

[4] Sistema WIKIPEDIA, disponível em: <http://wikipedia.org>.

O modelo LCCM (Tong *et al*. 2009) é considerado *Nível 2*, i.e. inclui agentes que podem mentir. Ele possui recursos para lidar com mentirosos por meio de informações personalizadas do agente. Os resultados com o *SecuredTrust* (Das e Islam, 2012) mostram que, em geral, bons agentes sempre transmitem informações verdadeiras e maus agentes sempre repassam informações falsas. Porém, em um cenário real, bons agentes podem transmitir dados falsos para seus concorrentes e os maliciosos podem prover relatórios verdadeiros ocasionalmente para esconder sua verdadeira natureza. Com base nisso, o modelo apresenta uma função de similaridade para identificar falsos relatórios que leva em conta a credibilidade do agente. O *Travos* (Teacy *et al*. 2006) é um modelo que utiliza estimativas de probabilidade para averiguar se a informação é verdadeira por meio da comparação dos resultados das interações anteriores com a informação atual. Caso não haja similaridade, as chances do relatório ser falso são maiores.

Seguindo uma linha similar, o *DiReCT* (Aboulwafa e Bahgat, 2010) e o modelo de Liu (Liu *et al*. 2012), ao receber uma recomendação, comparam a avaliação recebida contra sua própria opinião. Conforme a similaridade da comparação, o agente decide se dará importância à informação recebida. O modelo iCLUB (Liu *et al*. 2011) faz a coleta de testemunhos e agrupa-os em *clusters* para avaliar a veracidade das informações por meio de cálculos de intersecção entre os *clusters*.

2.2.7 Segurança da Informação

Na dimensão da *segurança da informação* a confiança está relacionada ao consentimento de acesso a recursos. Por exemplo, *o banco* confia no cliente para realizar pagamentos *online* de até R$ 5.000,00. Entretanto, acima disso o cliente precisará se dirigir a sua agência. Emprestar um livro usado para um amigo depende de menos confiança comparado a emprestar um veículo pessoal. Tais exemplos, ilustram como a confiança pode delimitar a forma de concessão a determinados recursos. Grandison e Sloman, (2000) propõe quatro categorias de confiança baseadas na segurança da informação:

1. *Acesso a recursos*: quando o indivíduo confia em um administrador para controlar seus recursos. Esse administrador pode ser um software que controla o ambiente ou um serviço projetado para funcionar com o sistema;

2. *Prestação de serviço*: quando o indivíduo confia em outro para prestar um serviço. Por exemplo, serviços de recomendação de filmes, restaurantes ou hotéis;
3. *Entidades certificadoras*: quando a confiança do indivíduo é baseada na certificação da sua idoneidade por meio de um terceiro, de modo que a confiança é obtida sob um conjunto de certificados apresentados. Por exemplo, quando confia-se em um advogado credenciado pela *OAB*, ou confia-se apenas em sites com certificado digital da *ICP-Brasil*;
4. *Delegação*: quando o indivíduo confia em alguém para tomar decisões em seu nome, sobre recursos ou serviços que o outorgante possui ou controla. Por exemplo, delegar decisões de investimento para um conselheiro financeiro;
5. *Infraestrutura*: quando o indivíduo confia na infraestrutura que está utilizando. Esta relação deve ser capaz de abonar credibilidade nas estações de trabalho de uma organização, suas redes locais e servidores. A partir de uma infraestrutura confiável é possível evoluir a segurança para outros níveis de serviço.

2.3 Ferramentas para Avaliação de Modelos de Confiança

Esta seção apresenta iniciativas que tratam ferramentas, frameworks ou tecnologias para utilização de modelos de confiança. A ferramenta AVALANCHE (Padovan *et al*. 2002) é um simulador voltado ao mercado que simula agentes vendedores e consumidores de produtos e serviços. A arquitetura do sistema, ilustrada pela Figura 2.10, consiste de três entidades: o localizador de agentes (*AvLocationAgent)*, os agentes (*AvTradeAgent*), e o monitorador do sistema (*AvInfoServer).*

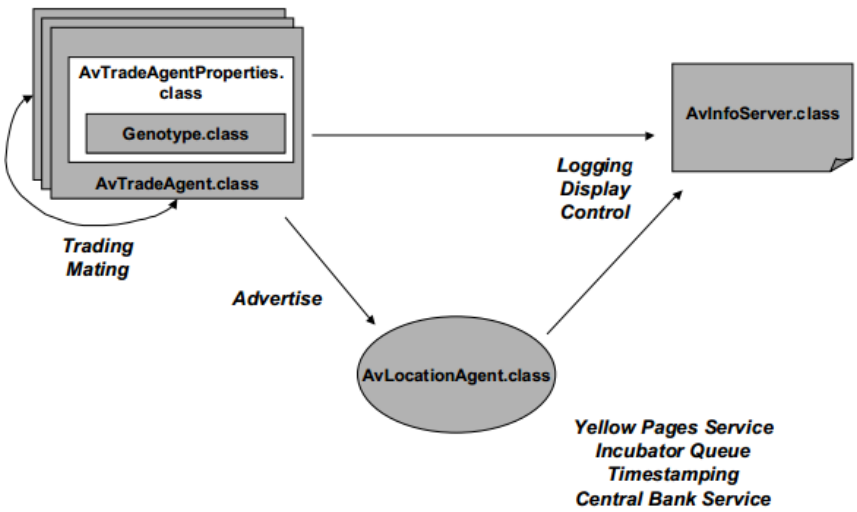

Figura 2.10: Arquitetura do AVALANCHE (Padovan *et al*. 2002).

O AVALANCHE possui alternativas para calcular a confiança dos agentes partindo de três situações: quando o agente vendedor é totalmente desconhecido, o agente consumidor arbitra um valor entre 0 ou 1. Esse valor inicial é a média aritmética de todas as avaliações do sistema. Quando um agente vendedor é desconhecido pelo comprador, mas é conhecido por outros, é possível obter informações sobre o vendedor a partir de testemunhos; e, finalmente, quando o comprador conhece o vendedor, pode-se usar esse conhecimento e agregar à informações externas. O cálculo da confiança é dado pela média ponderada das avaliações, conforme pesos de cada contexto.

O projeto *ART Testbed* (Fullam *et al*. 2006) propõe uma competição em forma de jogo com a finalidade de comparar diferentes estratégias de confiança. De modo geral, o jogo inicia a partir de clientes que solicitam avaliações sobre pinturas de diferentes épocas. Se um agente avaliador não possui experiência sobre determinada obra, ele pode solicitar opiniões de outros avaliadores. As avalições são remuneradas pelos clientes, deste modo, os avaliadores competem entre si, mas precisam colaborar para maximizar seu lucro e vencer a competição. A Figura 2.11 ilustra as interações entre agentes clientes e avaliadores.

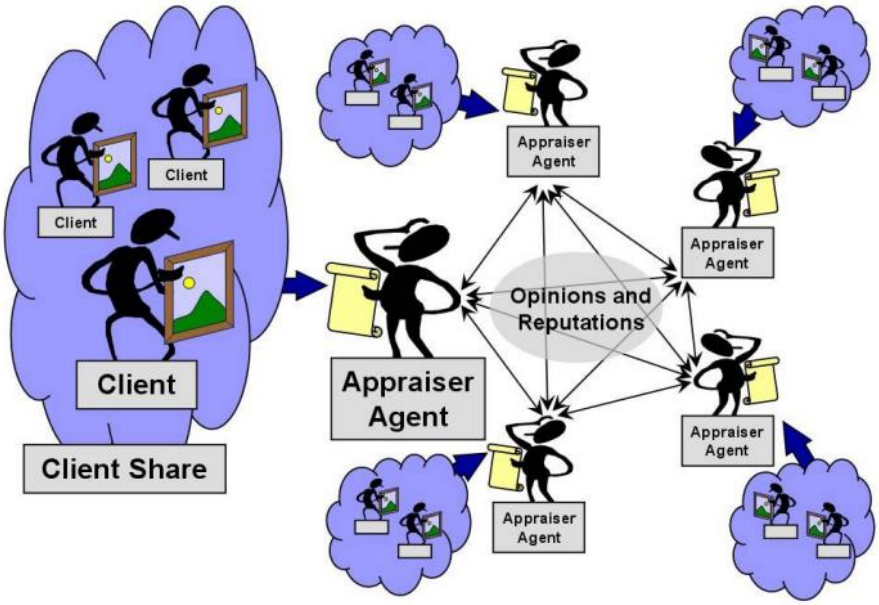

Figura 2.11: ART Testbed – Visão geral do jogo (Fullam *et al*. 2006)

O desempenho do modelo de confiança é aferido a partir do desempeno de cada agente. O melhor agente é aquele que (1) estima o valor das suas pinturas mais precisamente e (2) compra informações de forma prudente. Até 2008 o projeto promovia competições para a comunidade científica com objetivo de selecionar as melhores estratégias de confiança. Trabalhos com o de (Teacy *et al*. 2008) e (MUÑOZ

et al. 2009) utilizaram o ART Testbed para comparar seus experimentos contra outros modelos propostos. O projeto foi descontinuado, sendo impraticável utilizá-lo para novas simulações.

Durante a revisão deste trabalho notou-se a escassez de ferramentas para produção de experimentos me modelos de confiança. Até o momento desta pesquisa, constata-se a inexistência de ferramentas amplamente reconhecidas na comunidade científica para simulação de modelos de confiança. As revisões de Sabater e Sierra, (2005) e por Pinyol e Sabater, (2013) enfatizam a carência e necessidade de ferramentas para avaliar e comparar modelos de confiança. Sabater e Sierra, (2005) afirmam que uma das emergências na área, é o desenvolvimento de ferramentas semelhantes às utilizadas na aprendizagem de máquina. A partir da lacuna identificada, este trabalho propõe na Seção 4.2 um sistema genérico para avaliação de modelos.

2.4 Análise de Trabalhos Selecionados

Para fundamentação teórica desta pesquisa, foram analisados 63 trabalhos, que apresentam propostas para novos modelos de confiança ou melhorias em modelos existentes. A Tabela 2.4 apresenta um resumo comparativo desses trabalhos. As legendas e classificação das dimensões estão disponíveis na Tabela 2.2 comentada anteriormente.

Tabela 2.4: Comparativos dos modelos de confiança e reputação

	Paradigma	Fonte da Informação	Visibilidade	Contexto	Suposição de Comportamento	Segurança da Informação
Marsh, (1994)	N	ID	L	U	0	ST
Abdul-Rahman e Hailes, (2000)	N	ID + TE	L	M	2	ST
Schillo *et al*. (2000)	N	ID+OD+TE	G	M	2	ST
Castelfranchi e Falcone, (2001)	C	ID	L	U	0	DE
Jsang e Ismail, (2002)	N	ID+TE	G	M	0	ST
Montaner *et al*. (2002)	N	TE	L	U	0	ST
Sen e Sajja, (2002)	N	ID+TE	L	U	2	ST

	Paradigma	Fonte da Informação	Visibilidade	Contexto	Suposição de Comportamento	Segurança da Informação
Carter e Ghorbani, (2003)	C	OD	G	UC	0	ST
Castelfranchi *et al*. (2003)	C	ID+TE+OD	L	U	0	DE
Griffiths e LUCK, (2003)	N+C	ID+TE	L	U	0	ST
Jurca e Faltings, (2003)	N	ID+TE	L+G	M	2	EC
Yu e Singh, (2003)	N	ID+TE	L	U	2	ST
Derbas *et al*. (2004)	N	ID+TE+SO	L	U	0	ST
Dong-Huynha *et al*. (2004)	N	ID+TE+ RC+SO	L+G	M	0	ST
Godo *et al*. (2004)	N	ID+TE+PR	L	U	0	ST
Neville e Pitt, (2004)	C	ID +TE	L	M	0	ST
Song *et al*. (2004)	N	TE	L	M	0	ST
Tran e Cohen, (2004)	N	ID+TE	L	U	2	ST
Ashri *et al*. (2005)	N	ID+TE+SO	L	U	0	ST
Griffiths, (2005)	N	ID	L	U	0	ST
Sierra e Debenham, (2005)	N	OD	L	M	0	ST
Wang e Zhang, (2005)	N	ID+TE	L	U	0	ST
Huynh *et al*. (2006)	N	RC	G	U	2	EC
Regan *et al*. (2006)	N	ID+TE	L	U	2	ST
Teacy *et al*. (2006)	N	ID+TE	L	M	2	ST
Zheng *et al*. (2006)	N	ID+TE	G	U	1	ST
Bedi *et al*. (2007)	N	TE	L	U	0	ST
Bentahar *et al*. (2007)	N+C	ID+TE	L	U	0	ST
Dondio e Barrett, (2007)	N	TE	G	U	-	ST
Ghanea-Hercock, (2007)	N	ID	L	M	0	ST
Li *et al*. (2007)	N	ID+TE+SO	L	M	0	ST
M. R. Mokhtar *et al*. (2007)	N	ID+TE	L	U	0	ST
Rettinger *et al*. (2007)	N	OD	L	M	0	ST
You, (2007)	N	ID+TE	L	U	2	ST
Zhang *et al*. (2007)	C	ID+TE	L	U	2	ST
Bertocco e Ferrari, (2008)	N	TE	G	U	0	IN
Rettinger *et al*. (2008)	N	OD	L	U	0	ST

	Paradigma	Fonte da Informação	Visibilidade	Contexto	Suposição de Comportamento	Segurança da Informação
Teacy *et al*. (2008)	N	OD	L	M	0	ST
Qing-Hua *et al*. (2009)	N	TE	L+G	M	0	ST
Tong *et al*. (2009)	N	ID+TE	L	M	2	ST
Aboulwafa e Bahgat, (2010)	N	ID+TE	L	U	2	AR
Elgohary *et al*. (2010)	N	TE	G	U	0	ST
Klejnowski *et al*. (2010)	N	OD+SO	G	M	0	ST
Matt *et al*. (2010)	N+C	ID	L	M	0	ST
Vogiatzis *et al*. (2010)	N	ID+TE	L	M	0	ST
Mokhtari *et al*. (2011)	N	ID+TE+PR	L	U	0	ST
Liu *et al*. (2011)	N	TE	L	M	2	ST
Parsons, (2011)	N+C	TE	L	M	0	ST
Singh, (2011)	C	ID	L	M	0	ST
Venanzi, (2011)	C	OD	L	U	0	ST
Das e Islam, (2012)	N	ID+TE	L	M	2	ST
Fang *et al*. (2012)	N	ID+TE	L	M	2	ST
Koster *et al*. (2012)	N+C	TE	L	U	0	ST
Liu *et al*. (2012)	N	TE	L	M	2	ST
Liu e Datta, (2012)	N	ID+TE+ PR+SO	L	U	0	ST
Piunti, (2012)	C	OD	L	U	0	ST
Serrano *et al*. (2012)	N	ID+TE+OD	L	U	0	ST
Sutcliffe e Wang, (2012)	C	SO	L	M	0	ST
Burnett *et al*. (2013)	N	ID+PR	L	U	0	ST
Fang *et al*. (2013)	N	ID+OD+SO	L	U	0	ST
Liu *et al*. (2013)	N	ID+TE	L	U	0	ST
Ebay, (2015)	N	TE	G	M	0	IN
Wikipedia, (2015)	N	TE	G	M	0	ST

A partir da Tabela 2.4, foi realizado o agrupamento das dimensões referenciadas em cada trabalho analisado. O gráfico da Figura 2.12 apresenta o resultado.

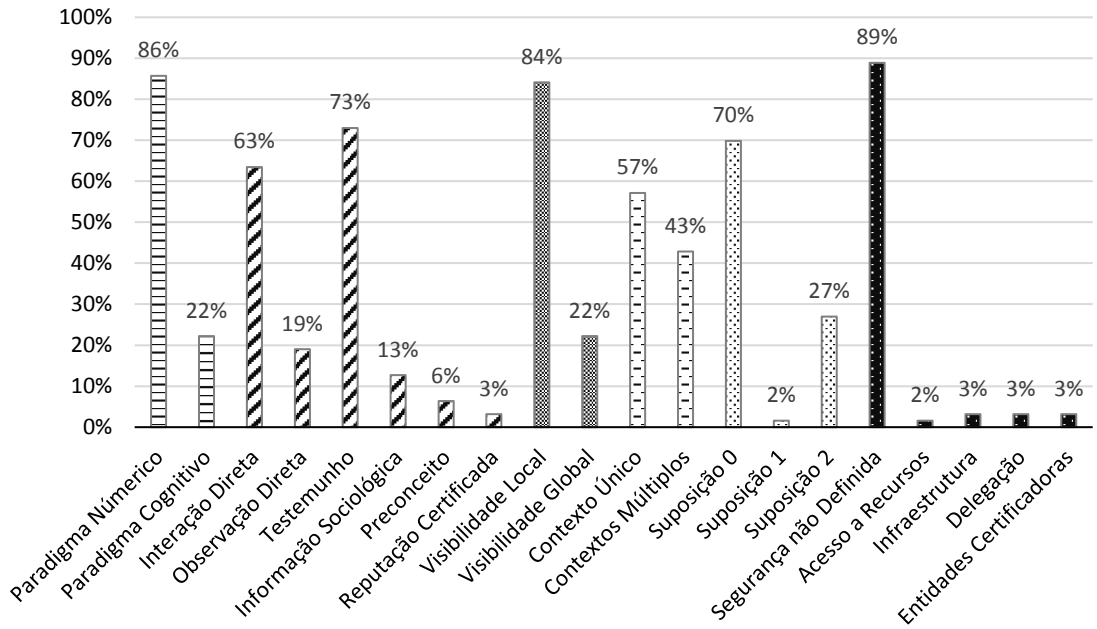

Figura 2.12: Uso das dimensões nos modelos de confiança analisados.

A partir do gráfico apresentado, percebe-se alguns comportamentos: o *paradigma numérico* está presente em 86% dos casos, isso pode ser explicado pela complexidade da construção dos modelos cognitivos; 73% dos trabalhos utilizam *testemunhos*, pois o uso exclusivo da interação direta é ineficiente e arriscado, à medida que expande a comunidade de agentes; e 89% utilizam a *visibilidade local*, i.e. modelos sem controle central, algo também esperado por se tratar da linha de pesquisa de Sistemas Multiagentes.

Entretanto, alguns resultados surpreenderam, 89% dos trabalhos analisados não apresentam mecanismos claros à segurança das informações e 70% dos trabalhos consideram que os agentes não omitem ou não mentem. Esse resultado pode ser explicado pelo fato dos trabalhos serem menos direcionados às comunidades abertas, onde o risco da interação com agentes desconhecidos é elevado, ou por considerarem a segurança da informação como um objeto de estudo para outro campo de pesquisa. Este trabalho considera a segurança da informação como uma dimensão imprescindível nos modelos de confiança, assim como Grandison e Sloman, (2000), pois comunidades abertas, por natureza, são vulneráveis a presença de agentes desleais, onde potenciais ameaças devem ser tratadas pelo próprio modelo de confiança.

2.5 Considerações Finais

Este capítulo apresentou a principal fundamentação teórica deste trabalho, os modelos de confiança para sistemas multiagentes. Foram analisadas cinco das principais revisões sobre o tema que resultou na proposta de uma nova revisão. Esta revisão atualiza as anteriores com ênfase nas dimensões de maior representatividade, sob o ponto de vista deste trabalho. Além disso, descreveu as lacunas identificadas ao longo deste estudo; a falta de ferramentas para avaliação dos modelos de confiança e o escasso número de trabalhos que tratam a *Segurança da Informação*.

Capítulo 3

Tecnologia de Registro Distribuído

Nos últimos anos, sistemas baseados em DLT, do inglês – ***D****istributed* ***L****edger* ***T****echnology*, têm atraído a atenção de muitas organizações do setor financeiro, originalmente por conta do protocolo inovador dos sistemas de pagamento sob *moedas virtuais* e mais recentemente pelos sistemas de *contratos inteligentes*. De modo semelhante aos sistemas multiagentes, discutidos no capítulo anterior, a tecnologia de *legder* distribuído permite a construção de novas soluções descentralizadas, e em particular, as DLT tratam notadamente a problemática de manter a integridade das informações mesmo sob um ambiente sem controles centralizados ou autoridades confiáveis. Este Capítulo analisa as principais características e técnicas das DLT no intuito de fundamentar parte do modelo de confiança proposto. Essa proposta lida com o desafio de estabelecer mecanismos de confiança para comunidades virtuais abertas e distribuídas, cenário típico e propício ao uso da DLT. Portanto, compreender os conceitos desse novo paradigma auxiliará o entendimento do modelo proposto.

3.1 Considerações Iniciais

Quando o assunto trata de segurança da informação é natural referenciar os algoritmos de criptografia, originalmente endereçados unicamente como uma técnica de *confidencialidade,* cujo processo consiste em codificar uma informação de tal forma que apenas o destinatário possa acessá-la. Com o surgimento dos algoritmos de *criptografia assimétrica* (Diffie, 1976), é possível garantir, além da *confidencialidade*, a *autenticidade* da informação, i.e., quem enviou a informação é de fato quem diz ser. Esse tipo de criptografia está atualmente presente em todas as transações feitas por meio

da *Internet*, desde o acesso a um *web site* até a assinatura digital de documentos ou comunicações entre servidores utilizando SSL (***S**ecure **S**ockets **L**ayer*).

Quatro décadas separam as primeiras publicações dos algoritmos assimétricos ao trabalho de Nakamoto (2008) que propõe uma estrutura de dados submetida a um conjunto de algoritmos criptográficos amplamente conhecidos, mas combinados de forma inovadora. A estrutura de dados da DLT é um tipo de *livro razão* de transações de negócios, que foi originalmente projetada para viabilizar transações com *moedas virtuais*, mas o seu uso despertou grande interesse para novas soluções, em particular, graças a sua capacidade de tratar inúmeros ativos – como documentos, registros, bens e arquivos digitais – sem a necessidade de entidades intermediárias. A DLT é uma tecnologia essencialmente aberta, distribuída e de escala global que permite fazer circular não apenas informações, mas tudo que pode ter valor – incluindo dinheiro, contratos, ações, propriedades, votos, etc. Um exemplo de *ledger* que mantém histórico de transações é a *Blockchain*. Para alguns "a *Blockchain* é fundamentalmente um novo paradigma para a organização das atividades, com menos atrito e mais eficiência, e sob escala muito maior do que nos atuais paradigmas." (Swan, 2015). Os usuários podem confiar nas informações mantidas por diferentes nós desconhecidos e espalhados ao redor do planeta em vez de confiar em autoridades intermediárias como bancos ou estruturas governamentais.

Desde o início, a tecnologia de *ledger* distribuído foi concebida para criação das *moedas virtuais*, com vista aos sistemas de pagamento. Diante do potencial da tecnologia, surgiu uma nova camada de desenvolvimento, os *Contratos Inteligentes*, que marcam a segunda geração das DLT. Um *contrato inteligente* permite a transferência de propriedade de modo transparente, sem necessidade de intermediários. Uma maneira de entender como funciona um *contrato inteligente* é compará-lo a uma *máquina de vendas automática*. Normalmente, na locação de um imóvel, necessita-se de uma imobiliária que intermediará o contrato entre locador e locatário. Com os contratos inteligentes, bastar enviar moedas à *máquina de vendas* para iniciar um novo contrato de locação, regras e penalidades em torno desse contrato são definidas de modo semelhante a um contrato tradicional, porém com a vantagem de tornar as averiguações automatizadas no próprio contrato. A Figura 3.1 ilustra o funcionamento dos contratos inteligentes.

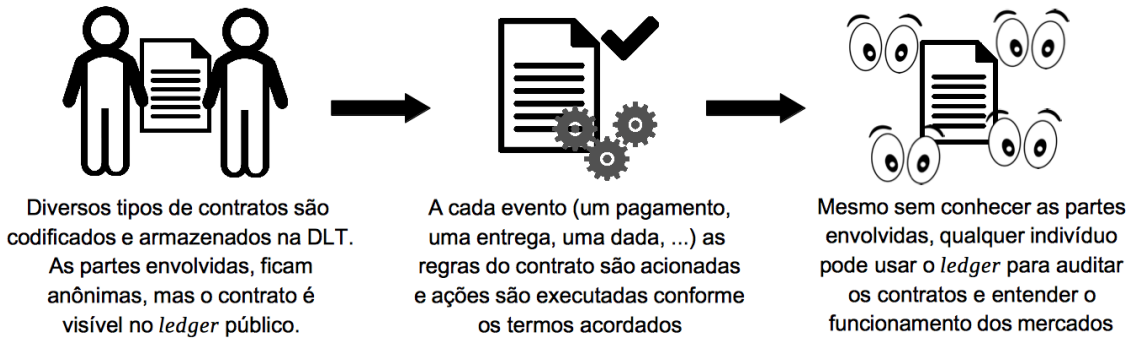

Figura 3.1: Funcionamento dos *contratos inteligentes.*

No exemplo da locação de um imóvel, supõe-se que o pagamento do aluguel seja feito por *moeda virtual*. O locatário terá seu recibo de pagamento registrado no contrato virtual, o locador enviará uma chave virtual para um período específico. Se a chave não for enviada a tempo, o locatário será reembolsado. Se o locatário devolver a chave em data posterior ao acordado ele pagará multa. As regras são registradas no *contrato inteligente* com a premissa *se-então* e são testemunhadas por milhares de pessoas, portanto tanto locador como locatário podem esperar um do outro que haja cumprimento dos seus deveres. Os *contratos inteligentes* podem ser utilizados nas mais diversas situações, desde prêmios de seguro, execução de crédito, sistemas de votação, processos legais, acordos trabalhistas, entre outros.

Dentre os projetos para construção de contratos inteligentes destaca-se o *Ethereum*, "uma plataforma descentralizada para execução de contratos inteligentes, aplicações que funcionam rigorosamente como programado sem possibilidade de tempo de inatividade, censura, fraude ou interferência de terceiros" (Wood, 2014). A plataforma *Ethereum*, diferente da rede *Bitcoin*, ela propõe uma DLT genérica que permite a criação de diferentes sistemas.

3.2 Estrutura de Blocos

A estrutura de dados do *legder* é uma lista cronologicamente ordenada de blocos. Cada bloco representa um conjunto de transações. Os blocos estão interligados de tal forma que cada um faz referência ao bloco anterior da cadeia. Eles podem ser entendidos como uma pilha vertical na qual a *altura* se refere à distância do primeiro bloco até o *topo*, o bloco mais recente da lista. Cada bloco é identificado por um valor *hash*, gerado por uma função de *embaralhamento* criptográfico e armazenado no próprio cabeçalho do bloco. O bloco anterior, também chamado de bloco *pai*, tem seu *hash* armazenado no cabeçalho do bloco *filho*. Por meio dessa estrutura, ao percorrer as ligações de cada

elemento da lista chega-se ao primeiro bloco criado, conhecido como o bloco *Gênese*. A Figura 3.2 ilustra uma lista ordenada dos blocos.

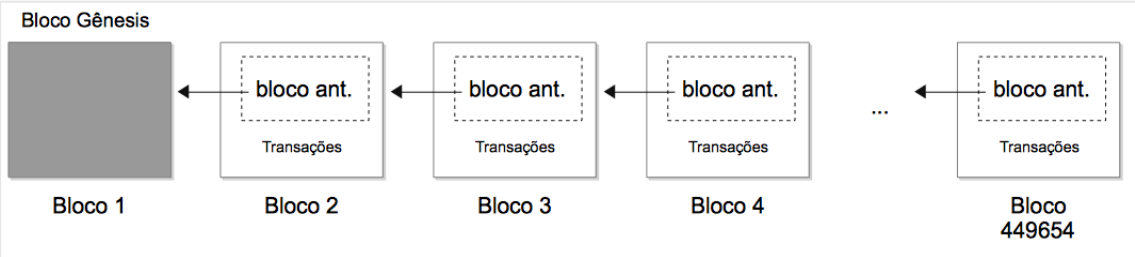

Figura 3.2: *Ledger* representado na forma de uma lista ordenada de blocos.

A estrutura de dados do *Bloco* tem por objetivo resumir um conjunto de transações, sejam pagamentos, transferências de bens ou outro tipo de registro. Em suma, tal bloco possui um cabeçalho, alguns metadados e uma lista de transações. A Tabela 3.1 descreve a sua estrutura em termos de dados.

Tabela 3.1: Estrutura de um bloco.

Tamanho	Campo	Descrição
4 bytes	Tamanho do bloco	O tamanho do bloco, em bytes
80 bytes	Cabeçalho do bloco	Campos que foram o cabeçalho do bloco
1-9 bytes	Contador de transações	Número de transações contidas no bloco
Variável	Transações	As transações registradas no bloco

O cabeçalho do bloco é estruturado sobre três conjuntos de metadados. O primeiro conjunto possui o campo *hash do bloco anterior*, que conecta o bloco corrente ao bloco anterior na cadeia. O segundo conjunto contém os campos *dificuldade alvo*, *carimbo de tempo* e *número arbitrário*, que são usados para resolver o problema de concorrência na geração de novos blocos. O último conjunto de metadados contém a *raiz da árvore de Merkle* que resume de modo eficiente todas as transações do bloco em operação. A Tabela 3.2 descreve a estrutura do cabeçalho de um bloco.

Tabela 3.2: Estrutura do cabeçalho do bloco de transações.

Tamanho	Campo	Descrição
4 bytes	Versão	Número da versão para acompanhar as atualizações do protocolo.
32 bytes	*Hash* do bloco anterior	Referência ao *hash* do bloco anterior na cadeia.
32 bytes	Raiz *Merkle*	*Hash* da raiz da árvore *Merkle* das transações do bloco.
4 bytes	Carimbo de tempo	Tempo aproximado de criação do bloco.
4 bytes	Dificuldade	Algoritmo da prova de trabalho utilizado no bloco.
4 bytes	Número arbitrário (*nonce*)	Contador usado no algoritmo da prova de trabalho.

O bloco é identificado de forma única e inequívoca pelo *hash* criptográfico do seu cabeçalho. Esse *hash* não está incluído na estrutura do bloco, nem é enviado com o bloco. O *hash* do bloco é calculado para cada nó, à medida que o bloco é recebido pela rede. Opcionalmente, o *hash* pode ser armazenado em bases de dados como parte de metadados que podem ajudar na indexação e na recuperação mais rápida dos blocos. Outra forma de identificar um bloco é por sua posição na cadeia de blocos, denominada de *altura do bloco*. Como cada bloco faz referência ao bloco anterior é possível percorrer a cadeia até chegar à altura desejada. No entanto, a altura do bloco não é um identificador exclusivo. Embora um bloco tenha sempre uma única altura, o inverso não é verdadeiro, pois dois ou mais blocos podem ter a mesma altura e temporariamente estarem competindo pela mesma posição na cadeia de blocos. Esta situação é discutida na Seção 3.6. Portanto, o *hash* do bloco sempre identifica de forma exclusiva um bloco da cadeia.

3.3 Árvore de *Merkle*

Os blocos—de uma cadeia de blocos—possuem um campo chamado *Raiz Merkle* que resume todas as transações ligadas contidas em um bloco. Esse resumo é feito pela *árvore de Merkle* (Merkle, 1987), uma estrutura de dados útil para verificar a integridade de grandes volumes de dados. As árvores de *Merkle*, também conhecidas como *árvores de dispersão*, são árvores binárias que contém informações resumidas sobre parte de um todo. A *árvore de Merkle* encerra um algoritmo criptográfico útil para proteger a integridade de qualquer tipo de dado armazenado, manipulado ou transferido sob uma rede de computadores.

A *Árvore de Merkle* é construída de forma recursiva. Esse processo de construção começa pela criação de pares de *hash* até que haja apenas um *hash*, chamado de *raiz de Merkle*. A árvore é construída de baixo para cima, partindo dos nós folhas até o nó raiz. No exemplo a seguir, mostrado na Figura 3.3, há quatro transações: A, B, C e D. A partir das transações são formadas as folhas da árvore, porém as transações não fazem parte da árvore.

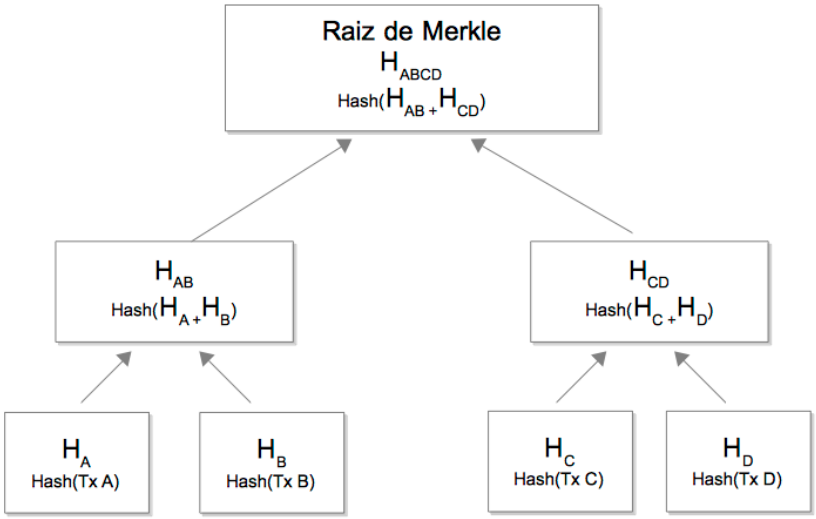

Figura 3.3: Cálculo dos nós para a construção da árvore de *Merkle*.

O valor *hash* de cada transação é armazenado nos nós folhas: H_A, H_B, H_C e H_D. Os pares dos nós folhas são resumidos em um nó superior, dado pelo *hash* da concatenação do par de *hash*. Surgem então os nós H_{AB} e H_{CD}. Por ser uma árvore binária, requer-se sempre um número par de nós, havendo número ímpar de nós o *hash* da última transação é duplicado para criar um par de nós. Este processo continua recursivamente até que haja apenas um nó, conhecido como a *Raiz de Merkle*.

Para verificar se uma transação foi incluída em um bloco composto por N transações são necessários $log_2(N)$ cálculos de *hash*, constituindo o chamado *caminho Merkle*, que leva qualquer nó folha à raiz da árvore. Caso tal caminho não seja atingido, então é provado que a transação em questão não faz parte do bloco. A Figura 3.4 apresenta um exemplo de prova para o nó G, na cor cinza escura. O caminho de *Merkle* é composto por quatro *hashes*, H_H, H_{EF}, H_{ABCD}, $H_{IJKLMNOP}$, que podem ser observados na cor cinza claro. Esses quatro *hashes* resultam o caminho de prova de tal forma que qualquer nó pode provar que H_G está incluído na raiz de *Merkle*. Para verificar o caminho de prova é necessário computar adicionalmente os *hashes* H_{GH}, H_{EFGH}, $H_{ABCDEFGH}$ e $H_{ABCDEFGHIJKLMNOP}$, representado pelos nós com linha pontilhada.

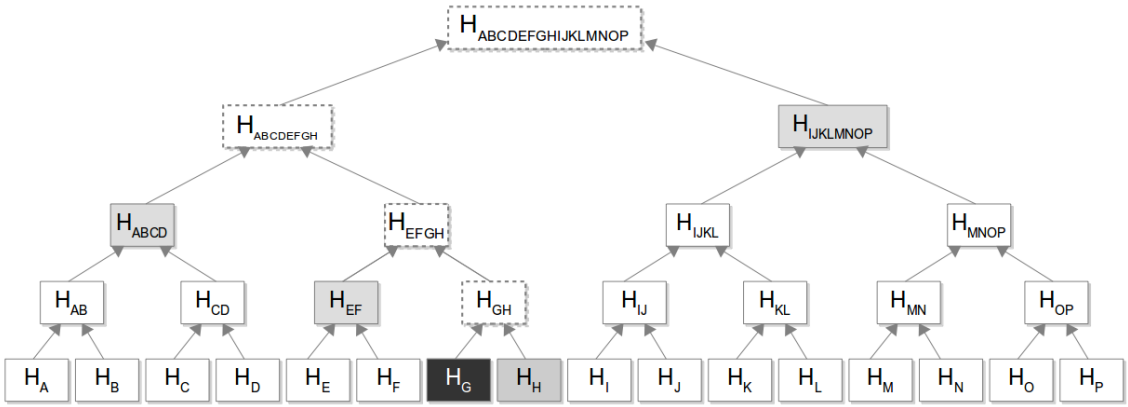

Figura 3.4: Caminho de *Merkle* para provar a inclusão de um nó no bloco.

A partir do *caminho Merkle* é possível verificar a integridade de qualquer nó da árvore. A eficiência da árvore de *Merkle* é reconhecida à medida que o número de transações cresce. Por exemplo, para provar que uma transação está incluída em um bloco com 4.294.967.296 transações seria necessário um caminho de *Merkle* composto por apenas 32 *hashes*. Considerando o tamanho de 32 bytes para um *hash* e 256 bytes para uma transação, tamanho médio de uma transação em sistemas de pagamento com moeda virtual, um caminho de *Merkle* de 1Kb pode provar a inclusão de uma transação em um bloco com 1Tb de dados.

As *Árvores de Merkle* são bastante utilizadas em sistemas de verificação de pagamento (Antonopoulos, 2014). Nesse tipo de problema não é preciso acessar todas as transações e nem o conteúdo completo dos blocos de uma cadeia de blocos. É suficiente armazenar o cabeçalho de cada bloco do *ledger* e validar uma transação a partir do seu caminho de *Merkle*. Esta abordagem resulta na necessidade de armazenamento quase 1.000 vezes menor que se fosse armazenado os blocos por completo.

3.4 Blocos da Rede *Bitcoin*

O *Bitcoin* é uma rede *peer-to-peer* para sistema de pagamento baseado em moeda eletrônica (Nakamoto, 2008). A principal contribuição da rede *bitcoin* à Ciência da Computação foi a DLT, à medida que ela estabelece um mecanismo de prova para todas as transações na rede sem a necessidade de *autoridades confiáveis*, como bases centrais, cartórios ou demais organizações que necessitem de confiança irrefutável. A DLT deixou de ser uma exclusividade do sistema *Bitcoin* e ganhou espaço em diversos outros segmentos da indústria de software. Conhecer os fundamentos desse sistema ajuda a entender a própria tecnologia criada por ele.

Os pagamentos eletrônicos feitos na rede *Bitcoin*, são registrados em blocos publicamente acessíveis na *Internet*. Informações como o volume financeiro das transações ou a localização geográfica do usuário que construiu o bloco são divulgados sem sigilo. Para ilustrar a estrutura de um desses blocos, a Figura 3.5 apresenta o registro de um bloco da rede *Bitcoin* numa ferramenta de exploração de blocos[5].

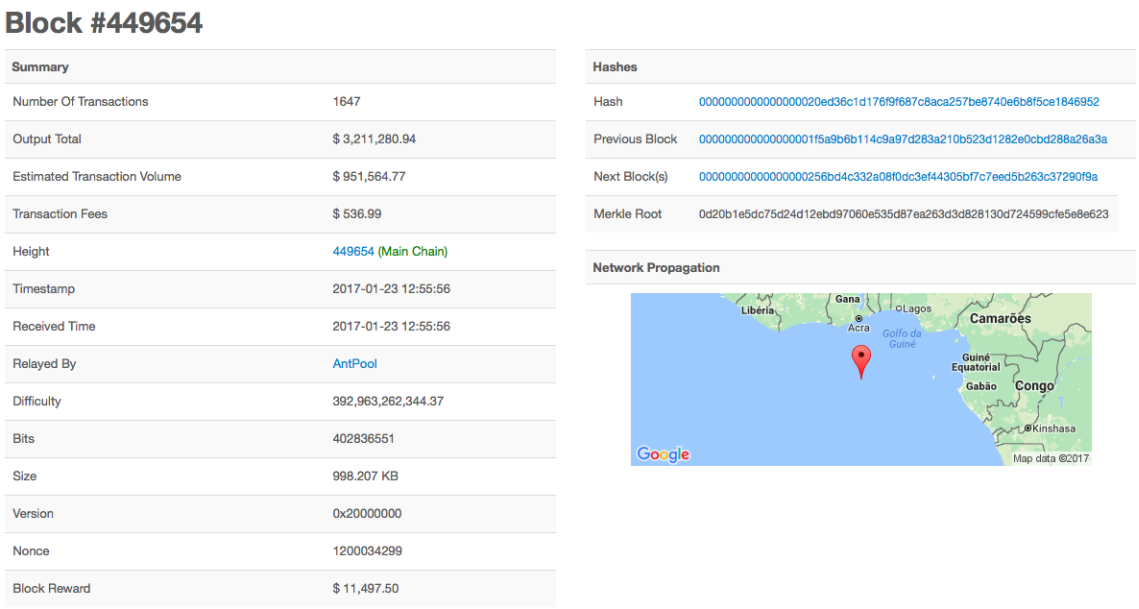

Figura 3.5: Representação de um bloco da rede *Bitcoin*.

Primeiramente é possível observar como o bloco é identificado. Sua altura de 449.654 é apresentada em destaque. A altura é a informação mais simples para um ser humano referenciar um bloco, porém o valor *hash* é a única forma de identificá-lo exclusivamente na cadeia. No exemplo da Figura 3.5, o *hash* do bloco também é apresentado: `00000000000000000020ed36c1d176f9f687c8aca257be8740e6b8f5ce1846952`, e pode ser facilmente localizado na *Internet* por meio do seu endereço eletrônico:

`https://blockchain.info/block/00000000000000000020ed36c1d176f9f687c8aca257be8740e6b8f5ce184695`

O primeiro bloco na cadeia é denominado *gênesis.* Na rede *Bitcoin* o bloco *gênesis* foi criado em janeiro de 2009. Ao percorrer qualquer bloco da rede *Bitcoin* em direção ao seu ancestral de modo sucessivo, resultará no bloco *gênesis*. O identificado bloco *gênesis* é o *hash*: `000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f.`
Esse bloco possui um conteúdo especial, a mensagem que diz: "*The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.*". A mensagem prova que o primeiro bloco foi criado após o dia 03/01/2009, pois faz referência a uma manchete veiculada

[5] Explorador de Blocos Bitcoin – Blockchain.info – http://blockchain.info

pelo jornal britânico *The Times*. A mensagem do primeiro bloco também resgata o apelo para a necessidade de um sistema financeiro independente, o ano de 2008 foi o auge da pior crise financeira desde a *Grande Depressão de 1929*, bancos e demais instituições financeiras precisaram de intervenções dos governos afetados (Crotty, 2009). A rede *Bitcoin* se apresenta como uma alternativa de sistema financeiro independente, na qual seus participantes atuam sem a necessidade de entidades intermediárias.

3.5 Transações e Consenso Descentralizado

Algo exclusivo com relação aos sistemas de *ledgers* distribuídos é seu mecanismo de *consenso descentralizado*. Todo nó da rede valida ou invalida as transações com o mesmo algoritmo de consenso, ou de acordo. As transações validadas por todos os nós são escritas no *ledger*.

A *transação* é uma das informações mais relevantes para um sistema baseado em DLT. Ela representa uma abstração do mundo real: um pagamento, uma transferência de bens, um voto, um contrato ou qualquer outra informação que precise ser mantida de modo imutável e irrevogável em um ambiente descentralizado. Por representar diferentes entidades, a transação possui uma estrutura específica de dado, mas há informações recorrentes que independem do tipo de sistema. Um exemplo, que visa compreender essa estrutura de dados, pode ser observada na transação da rede *Bitcoin*, apresentada pela Tabela 3.3.

Tabela 3.3: Estrutura de uma transação da rede *Bitcoin* (Antonopoulos, 2014).

Tamanho	Campo	Descrição
4 bytes	Versão	Defini qual regra a transação obedece.
1-9 bytes	Contador de entrada	Quantas entradas foram incluídas.
Variável	Entradas	Uma ou mais entradas de transação.
1-9 bytes	Contador de saída	Quantas saídas foram incluídas.
Variável	Saídas	Uma ou mais saídas de transação.
4 bytes	*Locktime*	Um carimbo de tempo ou número do bloco.

O destino previsto para toda transação é ser incluída no *ledger*, se tornando um registro público e irrefutável, semelhante à um cartório virtual. Após inclusão, a transação é reconhecida como legítima e as partes interessadas podem usufruir dessa prerrogativa. A *mineração*, termo usado no sistema *Bitcoin*, é o processo para incluir transações em novos blocos que farão parte da cadeia principal de blocos. A mineração tem como papel essencial evitar que transações sejam fraudadas, por exemplo, utilizar a

mesma moeda mais de uma vez, um problema conhecido como *despesa dupla* (Chaum, 1985). A criação de novos blocos, por meio da *mineração*, aumenta o tamanho da pilha e dificulta o esforço para quebrar a cadeia principal, garantido que as transações passadas não poderão ser removidas. Quão mais antigo for o bloco, mais difícil de quebrá-lo.

Integrantes da rede, chamados de *mineiros*, em busca de recompensas em moedas virtuais, validam novas transações e as registram em um novo bloco que possui todas as transações criadas desde o último bloco. Na rede *Bitcoin*, um bloco é adicionado ao *ledger* em média a cada dez minutos. Os mineiros competem entre si tentando resolver um problema matemático, o primeiro a resolvê-lo tem a oportunidade de incluir seu bloco na cadeia. A solução do problema matemático é incluída no novo bloco como prova de que o mineiro empreendeu significativo esforço computacional, chamado de *prova de trabalho* (Dwork e Naor, 1992). As transações incluídas em um bloco ganham *status* de *confirmada*. Esse *status* permite ao destinatário da transação usufruir do bem a que tem direito, seja uma moeda, propriedade, atestado, título de conhecimento, entre outros.

Até o momento o *ledger* distribuído tem sido apresentado como um local onde as transações, uma vez inseridas, são aceitas como irrefutáveis por todos os integrantes de uma comunidade. Entretanto, seria possível haver consenso sobre quem é *proprietário do que*, na ausência de *autoridades confiáveis*? O sistema financeiro tradicional submete-se a mecanismos centralizados em que há subordinação a um órgão principal que autoriza, normatiza e fiscaliza as demais instituições financeiras. Essa autoridade confiável prove o *serviço de compensação* que verifica e, quando necessário, ajusta ou elimina transações fora das regras estabelecidas. Em um *ledger* distribuído não há hierarquia ou divisão de poderes, a ordem é regida coletivamente por cada nó da rede que, por si só, pode verificar a cadeia de blocos e chegar a conclusão da veracidade das informações.

A Figura 3.6 ilustra os quatro processos fundamentais que resultam no *consenso descentralizado* da DLT (Antonopoulos, 2014). Cada nó da rede é responsável pela execução desses processos de formar independente:

- *Validação de Transações*: a partir de um conjunto de critérios, apresentados na próxima seção, as transações são verificadas de modo independente por todos os nós;

- *Agrupamento de Transações*: agregação de transações para a criação de novos blocos de forma independente com demonstração de esforço computacional por meio de algoritmos de *prova de trabalho*;
- *Validação de Blocos*: verificação independente de novos blocos *por cada nó* e montagem da cadeia;
- *Seleção da Cadeia de Blocos*: seleção independente, *por cada nó*, da cadeia principal, em suma, a seleção considera a maior cadeia, pois representa a maior *prova de trabalho*

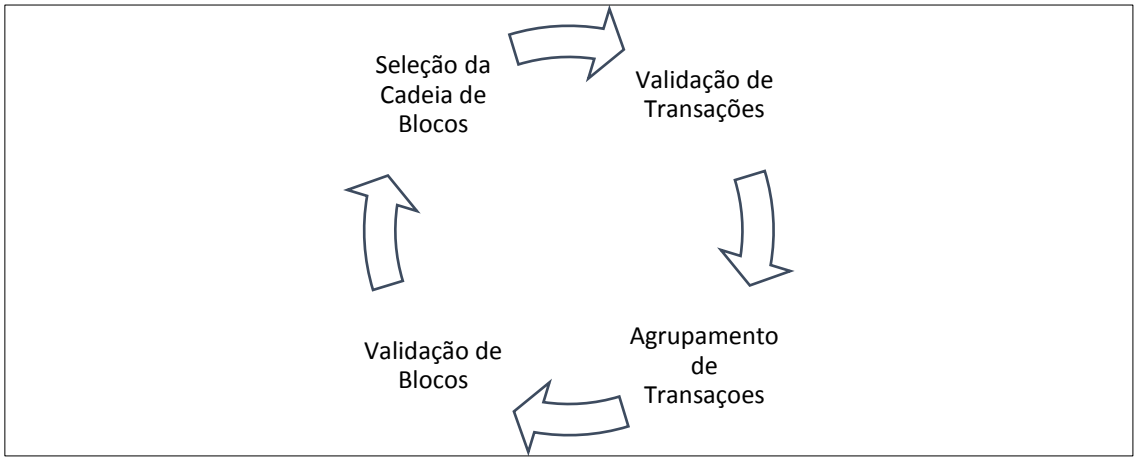

Figura 3.6: Processos para o consenso descentralizado.

Todo nó tem a função primária de verificar e propagar transações válidas, além disso, os nós *mineiros* agrupam as transações validadas para construir novos blocos. Quando um novo bloco é propagado na rede, todos os demais *mineiros* sabem que *perderam* a competição e no mesmo instante iniciam a construção de um novo bloco. Semelhante as transações, os blocos são propagados apenas se forem validados, as operações de validação e propagação são repetidas por todos os nós da rede. O *pool de transações* representa o conjunto de transações temporariamente armazenadas por um nó *mineiro* que ainda não foram confirmadas em um bloco do *ledger*. O *mineiro*, ao receber um bloco válido, deve remover todas as transações em seu *pool* que foram confirmadas naquele bloco. As próximas seções discutem com mais detalhes os quatro processos do consenso descentralizado.

3.5.1 Validação de Transações

As transações são propagadas na rede por difusão. Cada nó ao receber uma transação deve enviá-la para outros nós vizinhos até que toda a rede seja notificada. Entretanto, antes do envio, o nó verifica se a transação atende um conjunto de critérios pré-estabelecidos. Desta forma, apenas transações válidas são propagadas na rede. Transações inválidas são descartas na primeira validação feita pelo nó mais próximo. Os critérios são definidos conforme as necessidades de cada sistema. Segue um exemplo de critérios para validar uma transação da rede *Bitcoin*:

1. A transação deve estar sintaticamente correta.
2. Nenhuma lista de entrada ou saída pode estar vazia.
3. O tamanho da transação em *bytes* deve ser menor que o valor máximo permitido.
4. Cada valor de saída, bem como o total, deve estar no intervalo permitido.
5. Nenhuma entrada pode ter *hash* = zero ou N = -1—isso evita a retransmissão de moedas.
6. O campo *nLockTime* deve ser menor ou igual que o valor máximo permitido.
7. O tamanho da transação em *bytes* deve ser maior ou igual que o valor 100.
8. O total de operações contidas na transação deve ser menor que o limite permitido.
9. Deve existir uma transação correspondente no *pool* local do nó ou em um bloco no ramo principal.
10. Para cada entrada monetária, se houver uma saída referenciada em outra transação no *pool*, a transação deve ser rejeitada.
11. Para cada *entrada* monetária, procurar uma respectiva *saída*, seja no ramo principal ou no *pool* de transações. Se a transação de saída estiver faltando para qualquer entrada, ela é marcada como *órfã*, neste caso transação deve ser adicionada ao *pool*.
12. Para cada entrada monetária, se a transação de saída for uma saída de moeda, ela deve ter pelo menos uma quantidade mínima de confirmações.
13. Para cada entrada monetária, a saída referenciada deve existir e não pode ser imediatamente gasta.

14. Usando as transações de saída referenciadas para obter valores de entrada, deve-se verificar se cada valor de entrada, bem como a soma, está no intervalo permitido de valores.
15. A transação deve ser rejeitada se a soma dos valores de entrada for menor que a soma dos valores de saída.
16. A transação deve ser rejeitada se a recompensa da transação for abaixo do mínimo aceitável.

Os critérios para verificar uma transação, assim como as demais regras do sistema estão definidas no próprio código-fonte do software de referência para programas clientes da rede *Bitcoin*[6]. Por ser um projeto de código aberto, os clientes têm plena visibilidade das instruções que devem seguir para atuar corretamente e fazer parte dessa comunidade virtual.

3.5.2 Agrupamento de Transações

Após validação bem-sucedida da transação, ela é enviada ao *pool de transações*. Cada nó tem o seu *pool*, logo, é uma área local onde as transações permanecem até serem incluídas em um novo bloco. Enquanto isso não acontece, as transações ficam associadas a um bloco candidato. O nó *mineiro* tem as funções de escutar, validar e propagar transações, além de tentar construir um novo bloco, propagar novos blocos descobertos e manter uma cópia local do *ledger*. Quando um novo bloco é descoberto significa o fim da construção do bloco N e início da competição para o bloco $N + 1$. Durante as tentativas de criação do bloco N, o *mineiro* coleta e armazena centenas ou milhares de transações em seu *pool*. No momento em que o bloco N é recebido, o *mineiro* remove do seu *pool* as transações confirmadas nesse bloco. As demais transações não confirmadas no bloco N são mantidas no *pool* para serem incluídas no bloco $N + 1$; esse último pode deixar de ser um bloco candidato se o *mineiro* encontrar uma solução para o algoritmo da *prova de trabalho*.

Durante o agrupamento das transações, os *mineiros* priorizam a inclusão das transações mais antigas e de maior valor de recompensa. A idade da transação, no caso da rede *Bitcoin*, é calculada em função do número de blocos que decorreram desde até a inclusão da transação antecessora. As transações não possuem tempo de expiração, mas enquanto permanecem no *pool* sua existência é frágil. Se a transação é transmitida na

[6] Código-fonte do software *Bitcoin*: https://github.com/bitcoin/bitcoin

rede apenas uma vez e os *mineiros* a perdem, por algum tipo de falha, é como se a transação nunca tivesse existido. Para aumentar a resiliência, alguns nós se especializam em manter e retransmitir transações.

Cada transação tem seu valor de recompensa, uma taxa ofertada ao *mineiro* que conseguir criar o bloco para ela. Por isso, a primeira transação do bloco é chamada de *transação de moeda* que sumariza todas as recompensas das transações incluídas no bloco. A transação de moeda é criada pelo próprio *mineiro*, que se coloca como destinatário do valor. É a partir dessa transação que surgem novas moedas na rede, por esse motivo é dado, ao processo, o nome de *mineração*.

A promoção de um bloco candidato para *confirmado* depende da solução do algoritmo de *prova de trabalho* que torna o bloco verificável pelos demais nós da rede. A função de dispersão criptográfica é o mecanismo mais utilizado ao processo de *mineração* de rede *Bitcoin*. Por meio da função de dispersão é tecnicamente improvável encontrar duas entradas distintas que resultem no mesmo valor *hash*, por corolário também é impossível determinar uma entrada que produza um *hash* específico. De modo geral, a *mineração* é uma *busca por força bruta* de um valor que combinado ao cabeçalho do bloco candidato, resulta em um *hash* com uma determinada característica esperada pela rede. A característica do *hash* é chamada de *dificuldade da prova de trabalho*.

Ao construir um bloco candidato, o *mineiro* calcula o *hash* do cabeçalho e verifica se resultou em um valor menor que a meta estabelecida, também chamada de *alvo*. Se o *hash* for maior, o *mineiro* acrescentará um número ao cabeçalho e calculará o novo valor *hash*. Esse processo é repetido—i.e. a busca do valor é feita pelo incrementado numérico—até que se atinja o *alvo*. Quanto menor o valor do *alvo* mais difícil será a *prova de trabalho*.

3.5.3 Validação de Blocos

O terceiro processo do mecanismo de *consenso descentralizado* é a validação dos blocos, feita de modo independente por cada nó da rede. Após validar um bloco recém propagado, o nó constrói localmente uma cadeia ligando o novo bloco recebido ao último bloco da cadeia principal do *ledger*. Os nós mantêm três conjuntos de blocos: blocos da cadeia principal; blocos que formam ramos secundários; e blocos órfãos, que não apresentam um bloco pai conhecido. O bloco é rejeitado pelo nó caso não atenda os seguintes critérios de validação:

1. A estrutura de dados do bloco deve estar sintaticamente correta.
2. O *hash* do cabeçalho do bloco deve ser menor que a dificuldade estabelecida na *prova de trabalho*.
3. O *timestamp* do bloco deve ser menor que duas horas no futuro.
4. O tamanho do bloco deve estar dentro dos limites aceitáveis.
5. Todas as transações inclusas no bloco devem ser validas.

A validação independente de cada bloco—feita por cada nó da rede—evita possíveis fraudes ou erros cometidos pelos *mineiros*. Sem a validação coletiva, um *mineiro* poderia criar um bloco cuja transação de recompensa o deixaria milionário. Por conta de as transações serem verificadas sob regras comuns, que todos as seguem, a transação fraudulenta não se propagada na rede. A veracidade de uma informação, em um *ledger* distribuído, é apenas possível se for confirmada pela maioria simples dos nós.

3.5.4 Seleção da Cadeia de Blocos

O último processo do mecanismo de consenso é a seleção da cadeia principal de blocos. Logo após a validação de um bloco é iniciada a montagem da cadeia que ligará o bloco legitimado ao último bloco do *ledger*. Deve-se lembrar que a cadeia principal de blocos é aquela que possui a maior dificuldade de *prova de trabalho* acumulada, em geral a que contém o maior número de blocos. Quando duas cadeias possuem o mesmo número de blocos, temporariamente, os nós não poderão determinar a cadeia principal.

Por ser uma estrutura de dados descentralizada é admissível haver, num dado momento, réplicas do *ledger* com diferentes sequências de blocos. A ordem de chegada dos blocos pode ocorrer em momentos distintos, resultando, por alguns instantes, em diferentes visões da cadeia em construção. Essa situação é aceitável provisoriamente, à medida que todos os nós devem convergir para uma única cadeia principal. Para resolver este problema, os nós selecionam sempre a cadeia de blocos que possui a maior *prova de trabalho*.

O exemplo a seguir mostra uma situação de conflito, onde tem-se cadeias com bifurcações. A Figura 3.7 ilustra o momento em que a rede de nós possui uma visão única do *ledger*; o *Bloco AZ*—na cor azul—representa o topo da cadeia.

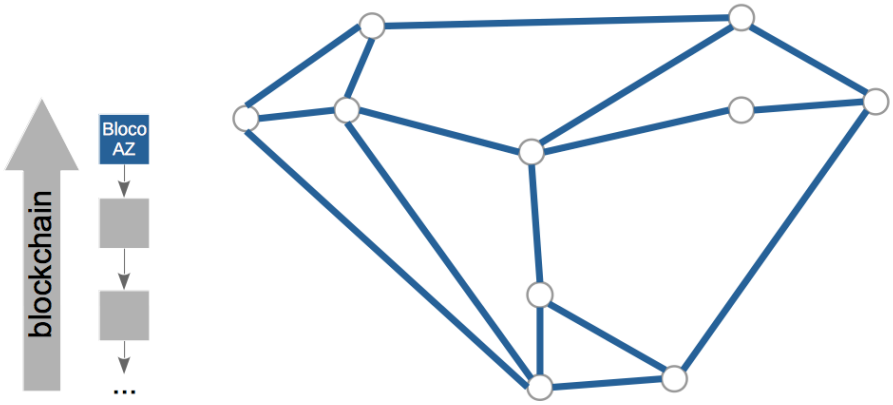

Figura 3.7: Rede antes da bifurcação.

A bifurcação da cadeia acontece quando dois blocos candidatos são promovidos a blocos válidos em datas muito próximas entre eles. Nesta situação, ilustrada pela Figura 3.8, dois nós *mineiros* resolvem a solução da *prova de trabalho* e transmitem seus blocos potencialmente *vencedores* aos seus nós vizinhos. Os nós vizinhos recebem tais blocos e os adicionam em sua cadeia principal. Alguns nós receberão primeiro o *Bloco LA*—na cor laranja—, outros receberão primeiro o *Bloco VD*—na cor verde. Durante certo tempo a rede estará dividida entre duas cadeias válidas, uma com o *Bloco LA* no topo e a outra com o *Bloco VD* no topo.

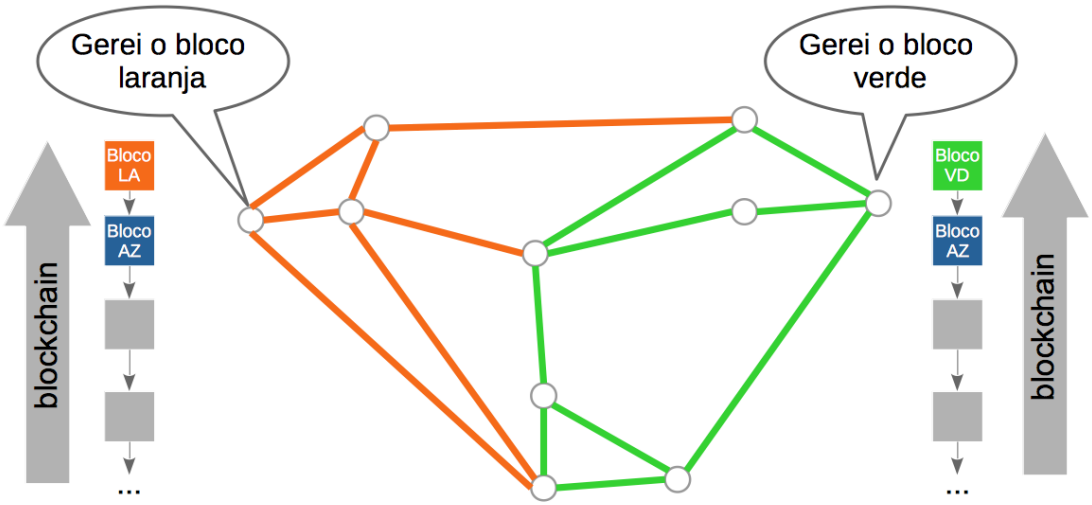

Figura 3.8: Rede com duas representações do *ledger*.

Em um dado instante os blocos *LA* e *VD* são propagados por toda a rede. Com o passar do tempo, os nós terão dois blocos apontando para o bloco *AZ*. Nessa situação, prevalece a ordem de chegada dos nós. O primeiro bloco recebido é considerado parte da cadeia principal, o segundo é armazenado no conjunto de blocos da *cadeia*

secundária. Os nós continuam suas atividades de *mineração* e quando um novo bloco é criado, inicia-se nova propagação na rede. No exemplo, ilustrado pela Figura 3.9, um *Bloco VM*—na cor vermelha—é criado por um nó que possui o *Bloco VD* em sua cadeia principal. Ao propagar o novo bloco, parte da rede terá duas cadeias: uma composta pelo *Bloco LA* e outra composta pelo *Bloco VM* e *Bloco VD*. Ambas cadeias apontam para o mesmo topo, *Bloco AZ*.

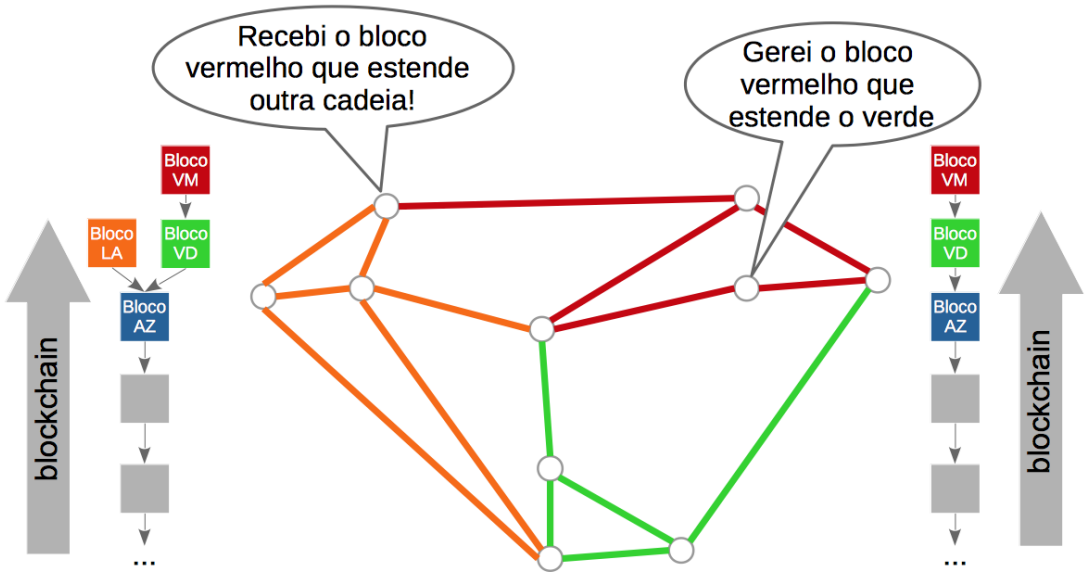

Figura 3.9: Rede com a terceira representação do *ledger*.

Neste caso, a cadeia de menor altura é eliminada e o *Bloco LA* deixa de existir. As transações adicionadas ao *Bloco LA* retornam ao *pool de transações* com suas prioridades incrementadas para serem adicionadas a um novo bloco em construção. O *Bloco VD* é propagado e rapidamente a bifurcação da rede é desfeita. A Figura 3.10 apresenta a nova configuração do *ledger* após a resolução do problema de consenso.

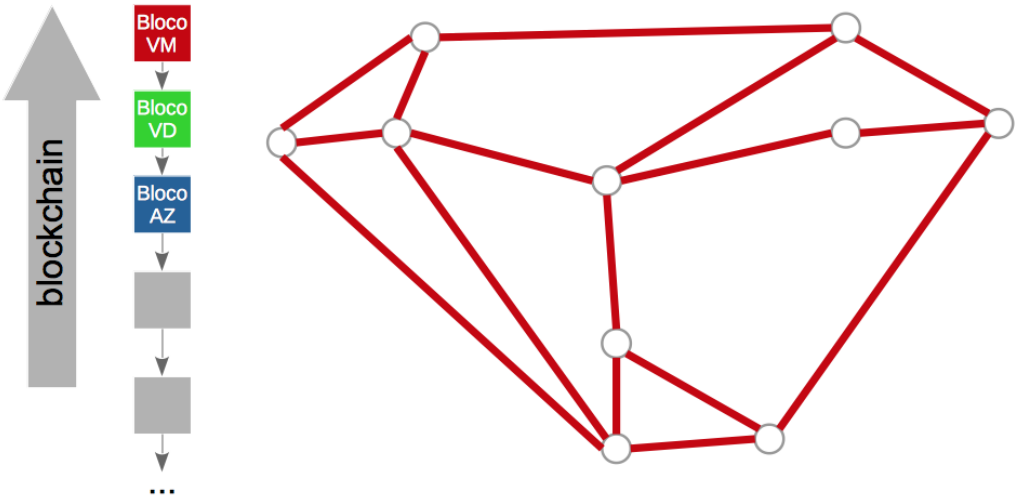

Figura 3.10: Rede solucionada por meio do consenso descentralizado.

A possibilidade de bifurcação da cadeia de blocos com mais níveis é rara de acontecer, mas é possível. Aumentar a dificuldade de *prova de trabalho* é a medida mais efetiva para reduzir as bifurcações, na medida que tal esforço adicional desacelera a produção de blocos e consequentemente a concorrência para inclusão de um novo bloco no *ledger*.

3.6 Considerações Finais

Nesse capítulo foram abordados os principais conceitos da DLT: estruturas de dados para a formação de blocos de transações, funções de dispersão, árvore de *Merkle*, consenso descentralizado, entre outros. Os assuntos tratados neste capítulo servem de fundamentação teórica para o modelo de confiança por *Dossiê*. A proposta visa um modelo distribuído, sem controles ou dados centralizados, que mantem a veracidade das informações entre indivíduos de uma comunidade aberta. O modelo *Dossiê*, detalhado a capítulo seguinte, possui exigências correlacionadas as atendidas pela abordagem da DLT.

Capítulo 4

Dossiê: um Modelo de Confiança Descentralizado

Nos dos capítulos anteriores foram explorados, de um lado, um conjunto de estruturas de dados e algoritmos com vistas à construção de bases de dados descentralizadas e seguras, e de outro lado, os desafios de comunidades virtuais abertas com vistas à construção de representações locais imutáveis sobre si mesmo, onde a confiança de um agente sobre outro depende do compartilhamento de tais informações locais. Com base nesta exploração, nós propomos uma solução para o problema da confiança em ambientes abertos—com controle e registro local de dados—, utilizando conceitos e mecanismos próprios, aliados a outros propostos nos sistemas *DLT*. O modelo proposto é denominado de *Dossiê—modelo de confiança para sistemas multiagentes*. Trata-se de uma abordagem descentralizada, onde os dados e os controles são distribuídos. Esta proposta permite a qualquer indivíduo de uma comunidade virtual aberta obter informações sobre outrem por meio de testemunhos armazenados localmente—no próprio agente em questão—, de modo seguro, i.e., sem que incidam adulterações ou omissões de informações.

O restante deste capítulo é consagrado as definições do modelo *Dossiê* e do sistema construído para a sua avaliação, *vis-à-vis*, outros modelos clássicos da literatura especializada.

4.1 Modelo *Dossiê*

O modelo de confiança Dossiê estabelece a premissa que todas as informações trocadas entre os agentes são *assinadas* e que cada agente pode ser identificado por seu *certificado digital*. Tal modelo pode ser ilustrado por meio do seguinte cenário: um agente provedor p, fornece um serviço a um agente consumidor c. O serviço pode ser entendido como qualquer ação destinada a satisfazer as necessidades do solicitante, seja uma transação comercial, uma

assistência médica ou uma mera pergunta a ser respondida. Na sequência, o agente c avalia o serviço e envia um *feedback* f para o agente p. O agente p armazena f localmente. O conjunto de *feedbacks* recebidos e armazenados por p, é chamado de *Dossiê* e denotado por $D(p)$. Esse último é utilizado como testemunho a respeito de p e pode ser consultado por outro agente que desejar aferir a confiança de p. Assim, para uma determinada interação i, um agente c avalia um agente p por meio da atribuição de um valor v—que expressa um grau de confiança—para um termo t. O termo é similar ao conceito da dimensão *contexto*, i.e., o termo pode ser qualquer característica a ser avaliada como, por exemplo, em transações comerciais: preço, prazo, qualidade, atendimento, dentre qualquer outro contexto necessário para o agente. Finalmente, um *feedback* é representado pela quíntupla $f = (c, p, i, v, t)$.

Até o momento, pode-se dizer que a representação do *Dossiê* endereça dois problemas comuns aos atuais modelos de confiança aplicados às comunidades virtuais abertas:

i) a falta de interesse dos agentes em compartilhar suas experiências e, na medida em que a comunidade cresce,
ii) o aumento do número de mensagens necessárias para localizar boas testemunhas.

Como os *feedbacks* são armazenados no próprio agente avaliado, nenhum outro agente necessita testemunhar sobre ele. A vantagem que decorre deste caso é o fato do agente consumidor c não precisar colocar em prática uma abordagem sofisticada para encontrar boas testemunhas ou avaliar a qualidade dos testemunhos, pois os *feedbacks* já se encontram registrados e disponíveis no *Dossiê* de cada agente provedor p.

Cálculo de Confiança

Além de atribuir—a um dado provedor *p*—o valor v que retrata um opinião geral sobre o agente avaliado, um consumidor c pode qualificar cada *termo* com pesos diferenciados. Na avaliação de c, por exemplo, o *preço* e a *qualidade* podem ser mais relevantes que o *prazo de entrega*. Nesse caso c atribui um peso *w* para o termo t_i do *feedback f, denotado por* $w(t_i)$. Do mesmo modo, c pode diferenciar *feedbacks* por meio de suas relações sociais s_p, por exemplo, *feedbacks* de agentes mais próximos como amigos, familiares ou colegas de trabalho, podem ter peso social $w(s_p)$ superior aos demais. O preconceito sobre p: $w(p_p)$, também pode ser necessário, de modo hipotético, avaliações de agentes estrangeiros podem ser menos relevantes que avaliações nacionais. Os pesos $w(t_i)$, $w(s_p)$ e $w(p_p)$ devem assumir valores no intervalo [0, 1]. Caso seja desconsiderado algum destes parâmetros o

valor padrão é 1. A confiança do agente c em relação ao agente p é denotada por $T(c,p)$. Ela é calculada pelo próprio agente consumidor c a partir dos *feedbacks* contidos no *dossiê D(p)* do agente provedor *p*. A Equação 1 apresenta a cálculo de confiança de c sobre p.

$$T(c,p) = \frac{\sum_{f_i \in D(p)} \alpha(f_i) \cdot w(t_i) \cdot w(s_p) \cdot w(p_p)}{\sum_{f_i \in D(p)} \alpha(f_i)} \quad (1)$$

No modelo de cálculo da confiança por *Dossiê*, o fator tempo também é considerado de maneira que haja decaimento da relevância dos *feedbacks* com o passar do tempo. Tal redução é determinada pelo coeficiente α. Os *feedbacks* mais recentes se tornam mais relevantes sobre *feedbacks* mais antigos. O coeficiente α é representado pela seguinte função exponencial (Eq. 2):

$$\alpha(f_i) = e^{-\frac{\Delta_t \cdot (v_i)}{\gamma}} \quad (2)$$

Onde:

- Δ_t representa o tempo decorrido entre o momento da criação do *feedback* e o momento em que o cálculo de confiança é feito; e
- γ é o fator que determina a velocidade do decremento da função exponencial.

4.1.1 Criptografia Assimétrica

O modelo Dossiê tem por princípio garantir a legitimidade das informações trafegadas, requisito necessário às comunidades abertas dado que agentes desleais, no intuito de obter vantagens indevidas, se passam por outrem ou modificam o seu perfil. Para evitar tal descompasso, propõe-se usar a técnica de *criptografia assimétrica* (Merkle, 1987) para assinar os *feedbacks* veiculados na rede de interconexão da comunidade de agentes.

Tecnicamente, para colocar em prática a assinatura das informações trocadas entre os membros da comunidade, cada agente é identificado por um *Certificado Digital*—documento eletrônico—, que garante a identidade do emissor, a integridade da mensagem e, opcionalmente, a sua confidencialidade. Os passos para identificação de um dado agente por *certificado digital* é ilustrado (cf. Figura 4.1) e descrito da seguinte forma:

1. Um agente gera um par de chaves assimétricas: uma chave é *secreta* (ou privada) e uma chave é *pública*. Embora diferentes, as partes desse par de chaves permanecem matematicamente ligadas. A chave *privada* é mantida em sigilo pelo agente que a

detém e a chave *pública* é distribuída livremente para qualquer indivíduo da comunidade.

2. Um agente solicita seu *certificado digital* para uma AC (*Autoridade Certificadora*) reconhecida como notável pela comunidade virtual. No requerimento, o agente informa sua chave *pública* e um conjunto de dados que o identificará conforme o protocolo de inscrição definido pela AC.
3. Após a verificação da inscrição, a AC emite um *certificado digital assinado* por ela com as informações de inscrição do agente solicitante. O certificado possui as principais informações sobre o agente e sua chave *pública*.

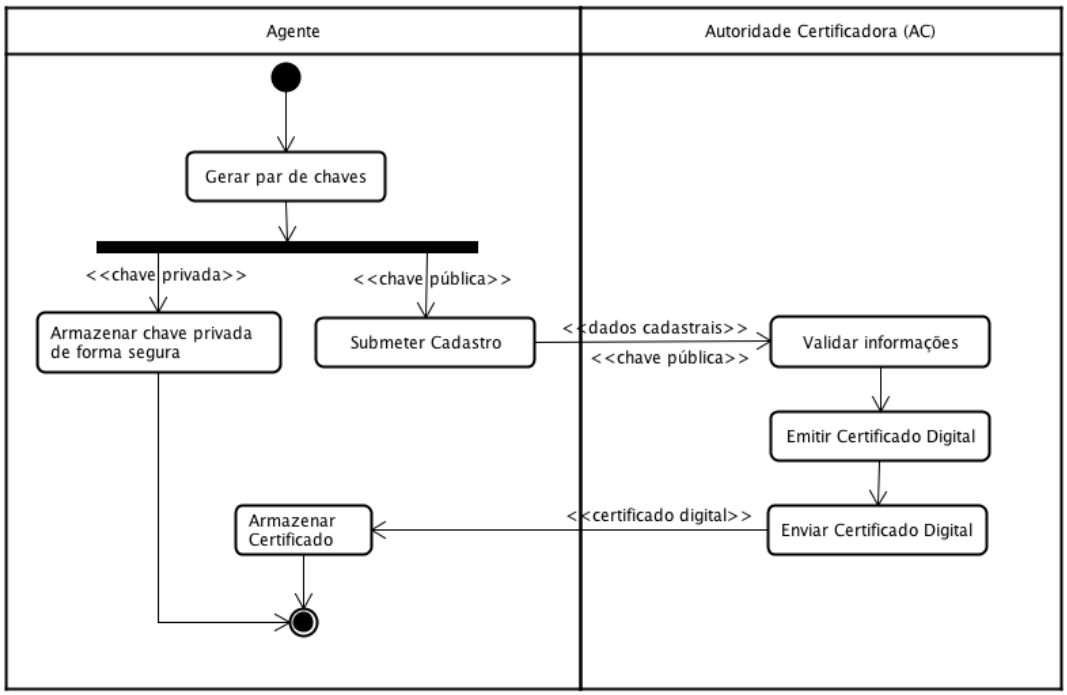

Figura 4.1: Identificação do gente por meio de *certificado digital*.

O *certificado digital* é a identificação do agente na comunidade virtual. E com tal certificado é possível assegurar a autenticidade das suas mensagens, aplicando o método de *Assinatura Digital* (Diffie, 1976). A *Assinatura Digital* é um método de criptografia que permite atestar a autoria de um documento eletrônico, de modo independente e descentralizado. A Figura 4.2 apresenta os passos para o envio e o recebimento de mensagens assinadas.

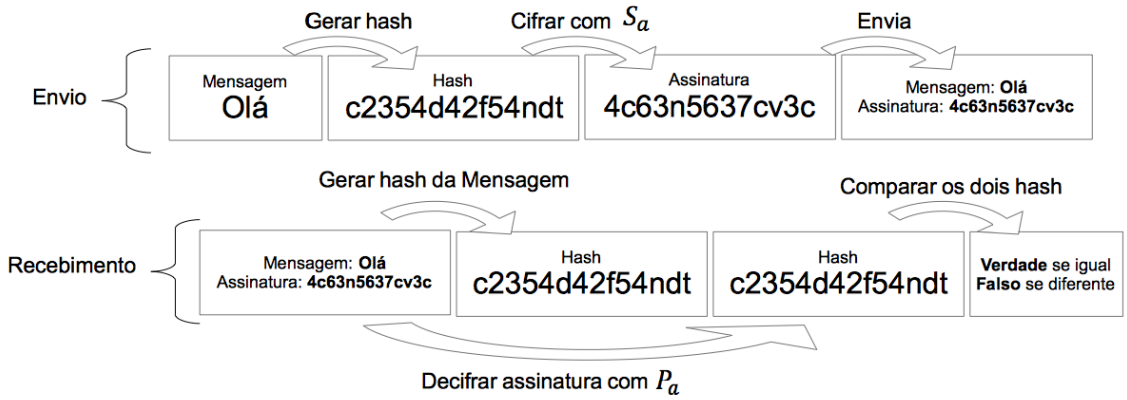

Figura 4.2: Verificação de mensagem por assinatura digital.

Dado um agente a e seu par de chave: secreta S_a e pública P_a. Quando a envia um dado D_a para um destinatário, a calcula a função *hash* de D_a que resulta em $h_{D_a} = hash(D_a)$ e criptografa h_{D_a} com a sua chave secreta $A_{D_a} = encrypt(h_{D_a}, S_a)$, onde A_{D_a} é chamada de assinatura digital do dado D_a. Assim, a ao enviar D_a, precisa enviar seu certificado digital que possui a chave pública S_a. O destinatário da mensagem pode verificar se D_a foi criado por a calculando novamente o *hash* da mensagem recebida h_{D_a} e decifrando A_{D_a} a partir da chave pública P_a, logo se: $decipher(A_{D_a}, P_a) = h_{D_a}$ assume-se que D_a foi criado pelo agente a.

4.1.2 Cadeia de Blocos

O modelo *Dossiê* inclui uma estrutura de dados mantida localmente pelo agente avaliado, por isso é fundamental prover mecanismos que garantam a integridade dos dados a fim de proteger-se de possíveis agentes maliciosos que tentem modificar intencionalmente suas avaliações ou omitir avaliações indesejadas. O mecanismo de assinatura digital, conforme descrito anteriormente, assegura a autoria e imutabilidade dos *feedbacks*, mas ainda resta o potencial problema da comunicação seletiva de avaliações, i.e., a remover seletivamente do *Dossiê* as avaliações acusativas com vista a dissimular a má reputação de fato de um agente mal-intencionado.

Para o desafio supracitado, a proposta deste trabalho é abordá-lo na perspectiva de um *ledger* distribuído para manter a imutabilidade do conjunto de avaliações de cada agente, i.e., prevenir que o registro de um *feedback* seja removido do *Dossiê*. Conforme já discutido, a aplicação do método de vinculação e estruturação de dados em uma *árvore com nós*

assinados—denominada por árvore de Merkle – torna possível resumir um conjunto de dados de modo eficiente. O modelo proposto prever que os *feedbacks* de um *Dossiê* sejam associados a uma árvore de *Merkle* (cf. Figura 4.3).

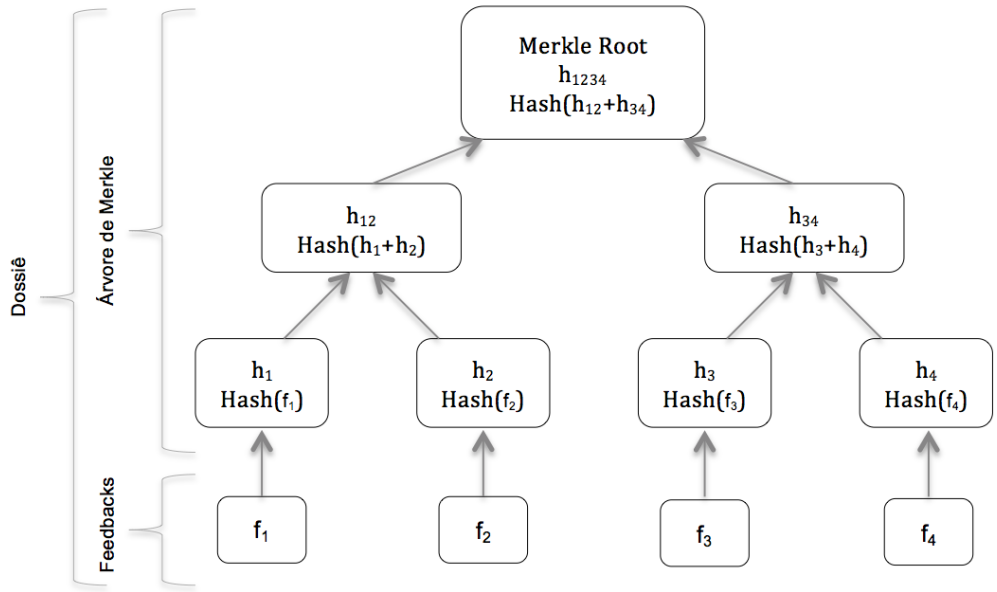

Figura 4.3: Estrutura imutável de um *Dossiê* baseada em uma *árvore de Merkle*.

O *Dossiê* é uma estrutura de dados composta por um conjunto de *feedbacks* e por uma respectiva *árvore de Merkle*. Dessa forma, para provar que um *feedback* foi incluído em um *dossiê* de tamanho *N* será preciso produzir $log_2(N)$ *hashes* para validar o *caminho de autenticação*. Apesar da árvore de *Merkle* permitir de forma eficiente a verificação da integridade dos *feedbacks* de um *dossiê*, ainda há outra questão a ser respondida: Como verificar que um agente provedor, a partir de um *dossiê* anterior, não construiu um novo *dossiê* a fim de descartar eventuais avaliações ruins? No exemplo a seguir, ilustrado pela Figura 4.4, observa-se a remoção de um *feedback* negativo $-f_3$ e reconstrução do dossiê sem ele.

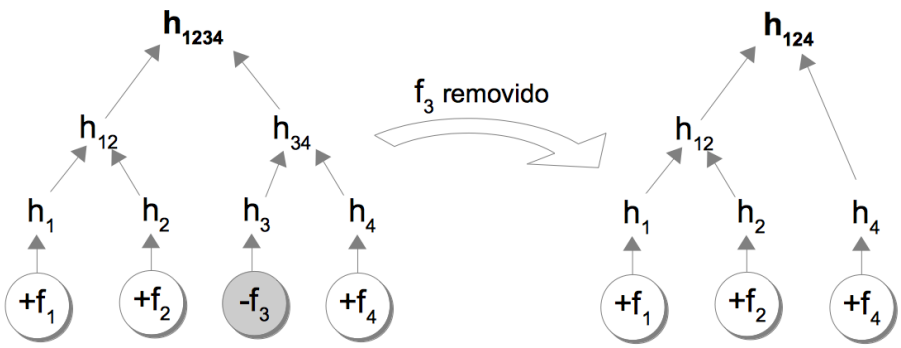

Figura 4.4: Remoção de *feedback* negativo no *dossiê*.

Para evitar a remoção de *feedbacks*, o modelo proposto define a *Transação de Dossiê*, um tipo de transação que é armazenada no *ledger* distribuído. Essa transação resume o *dossiê* em um dado momento no passado. Ela assemelha-se a uma fotografia do *dossiê*, armazenada de modo imutável e irrevogável no *ledger*.

Estrutura da Transação de *Dossiê*

Conforme já dito, a *transação de dossiê* persiste a situação de um *dossiê* em um dado momento. Os *feedbacks* incluídos nessa transação são imutáveis e eventuais remoções deixam rastros facilmente verificáveis. A transação contém os campos apresentados na Tabela 4.1:

Tabela 4.1: Estrutura de uma *transação de dossiê.*

Tamanho	Campo	Descrição
4 bytes	Versão	Especifica quais regras tal transação segue.
32 bytes	Identificador do agente avaliado	*Hash* do certificado digital do agente provedor.
32 bytes	Raiz de Merkle	Valor *hash* da raiz de *Merkle* para os *feedbacks* de um dossiê.
32 bytes	Transação anterior	Um *hash* da transação de dossiê anterior.
32 bytes	Feedback de origem	*Hash* do *feedback* que originou a nova versão do dossiê.
4 bytes	Locktime	Carimbo de tempo.

A *raiz da árvore de Merkle* representa o resumo criptográfico do *dossiê*, o *feedback de origem* armazena o *hash* do *feedback* incluído no *dossiê*. O campo *transação anterior* serve para interligar todas as transações de dossiê relacionadas ao agente avaliado. Dessa forma é possível percorrer o *ledger* no sentido inverso das mudanças do *dossiê* e obter todos os *feedbacks* de origem; qualquer *feedback* omitido é facilmente verificado. O campo *identificador do agente avaliado* possui o *hash* do certificado digital do referido agente avaliado, a partir desse campo pode-se obter as transações de dossiê de qualquer indivíduo. Caso o agente avaliado inicie um novo *dossiê* para *limpar* sua reputação é possível verificar a existência de transações criadas antes do novo *dossiê*, i.e., a fraude é descoberta. A estrutura completa do Modelo *Dossiê* com a estrutura da *blockchain* é ilustrada na Figura 4.5.

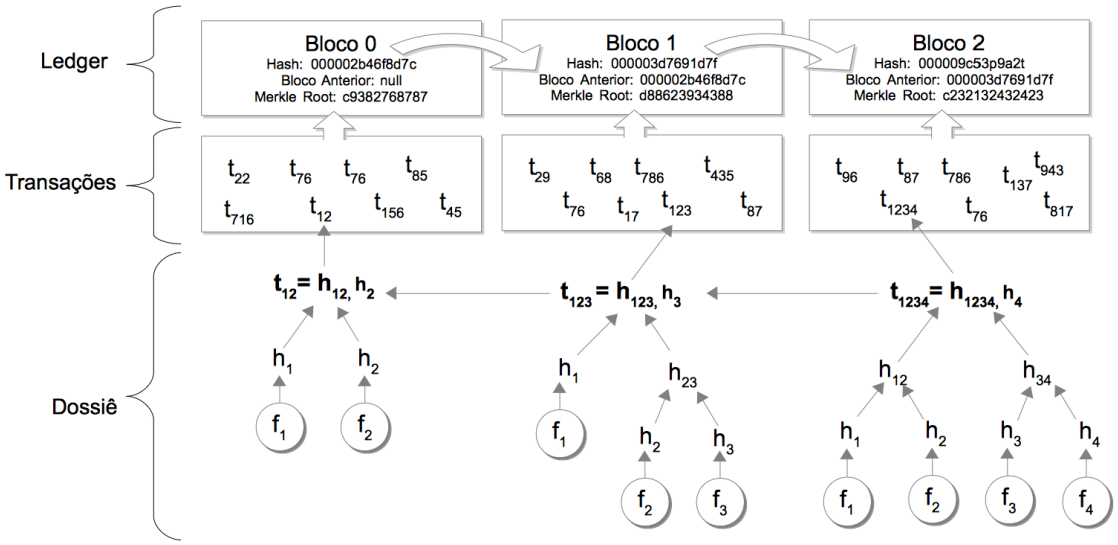

Figura 4.5: Verificação da integração do *dossiê* por meio do *ledger*.

A Figura 4.6 apresenta as atividades realizadas pelos agentes *consumidor* e *provedor* desde a solicitação de serviço, envio do *feedback*, atualização do *dossiê*, até o armazenado da transação do *dossiê* no *ledger*.

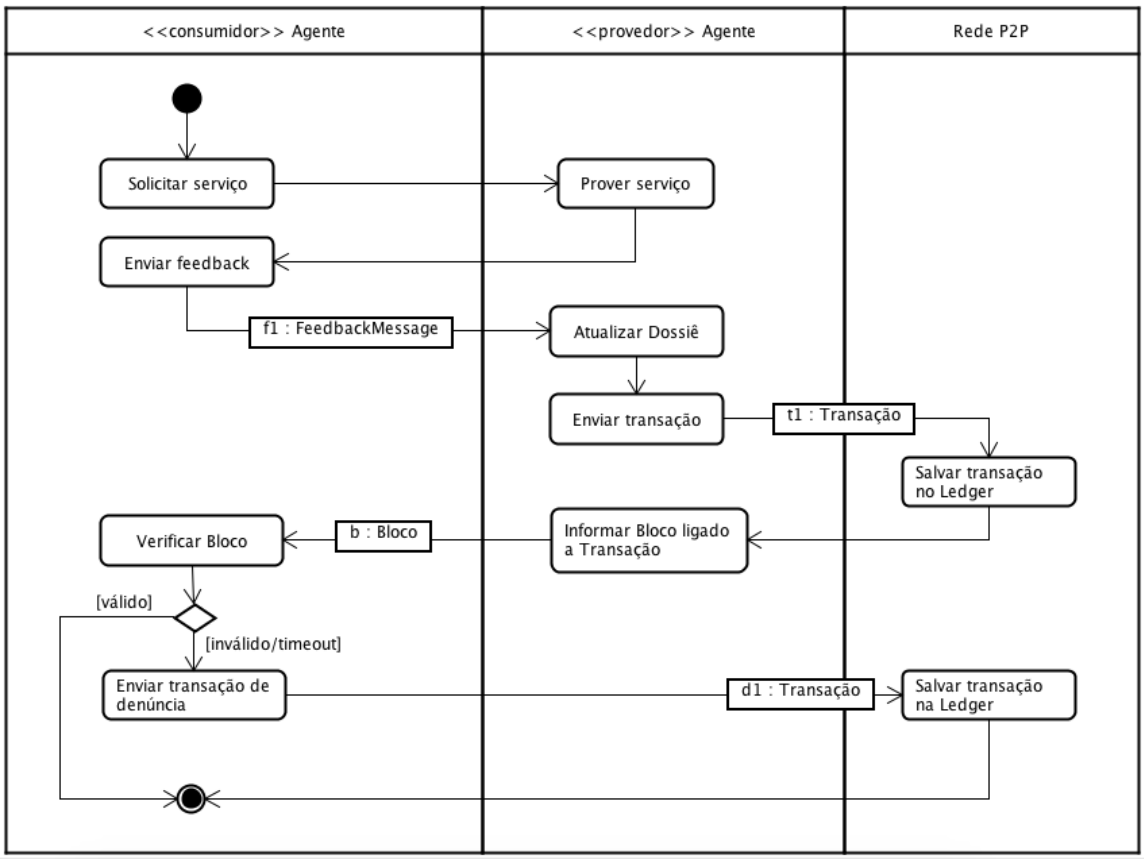

Figura 4.6: Envio do *feedback* e atualização do *dossiê* no *ledger*.

O envio do *feedback* ao agente provedor resulta em uma inclusão no *dossiê* de tal provedor. Cabe ao provedor registrar essa inclusão e devolver ao agente consumidor uma nova *transação de dossiê* registrada no *ledger*. O agente consumidor aguardará do provedor a identificação do bloco que contém a sua transação. Dessa forma, o consumidor terá a confirmação de que seu *feedback* se encontra no *dossiê* do provedor. Caso o consumidor não receba tal confirmação, ele poderá inserir no *ledger* uma *transação de denúncia*. A denúncia consiste na identificação dos agentes consumidor e provedor e o suposto *feedback* não inserido no *dossiê*. Com esses dados, qualquer indivíduo da rede pode verificar se a denúncia é verdadeira, pois o *feedback* é assinado pelos agentes participantes da transação e sua inclusão é verificada ao percorrer a cadeia de blocos.

4.1.3 Semântica das Mensagens no SMA

Como já dito, esse trabalho propõe o modelo de confiança *Dossiê* aplicado a comunidades virtuais abertas, representadas por sistemas multiagentes. Em uma comunidade de agentes, a interação das suas partes é operacionalizada por meio da troca de mensagens. Para haver um vocabulário comum entre as partes foi necessário definir uma representação semântica de cada elemento que compõe as mensagens trocadas. Mensagens que dizem respeito ao tema da confiança, como por exemplo: *Quem conhece <um dado agente>*?; *Envie-me o seu dossiê; e Informo-lhe* o *meu feedback sobre você*. Tais mensagens são modeladas em um espaço *ontológico* que permite mapear um modelo computacional eficiente—poucas mensagens. A Figura 4.7 apresenta os conceitos e os relacionamentos utilizados em uma mensagem para o uso do modelo de confiança *Dossiê*.

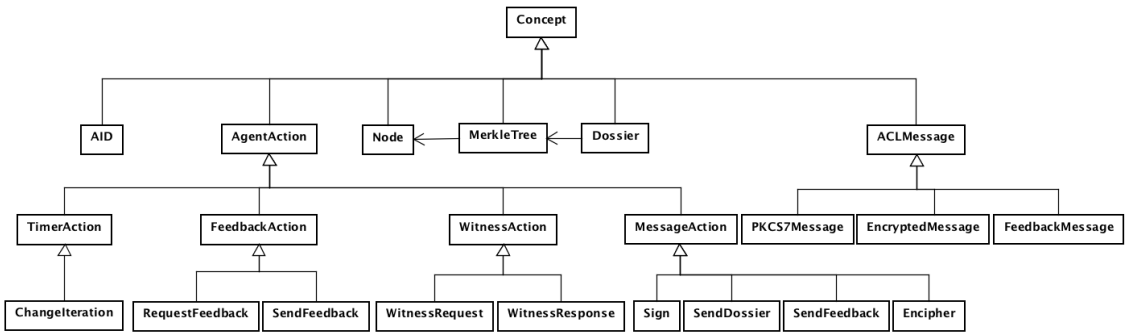

Figura 4.7: Estrutura taxonômica dos conceitos usados na composição das mensagens trocadas entre as partes operando com o modelo *Dossiê*.

A ontologia—estruturada na forma de uma taxonomia—encerra essencialmente três conceitos: Ação (*AgentAction*), Mensagem (*ACLMessage*) e o Dossiê (*Dossier*). Cada ação veiculada por meio de uma mensagem resulta em uma alguma atividade executada pelo remetente ou pelo destinatário de tal mensagem. Segue alguns exemplos de mensagens que poderiam ser colocadas em prática como *AgentAction*: "Envie um *feedback* sobre meu serviço" (*RequestFeedback*); "Segue *feedback* sobre seu atendimento" (*SendFeedback*); "Envie-me testemunhas sobre <aquele funcionário>" (*WitnessRequest*); ou "Segue meu Dossiê" (*SendDossier*). A Tabela 4.2 detalha os conceitos que herdam do conceito *AgentAction*.

Tabela 4.2: Rol de ações básicas para operação com o modelo *Dossiê.*

Conceito	Descrição
RequestFeedback	Solicitar um *feedback*.
SendFeedback	Informar um *feedback*.
WitnessRequest	Solicitar testemunhas sobre um agente.
WitnesseResponse	Informar testemunhas sobre um agente.
ChangeIteration	Informar aos agentes sobre a mudança do tempo ou iteração.
Sign	Informar um conteúdo assinado.
SendDossier	Informar um *dossiê* solicitado.

O segundo grupo de conceitos encerra diferentes mensagens no formato *FIPA ACL* (FIPA, 2002). Tais mensagens permitem aos agentes, independentemente da sua tecnologia interna de construção, comunicar-se por meio de um protocolo padrão reconhecido pela comunidade. As mensagens representadas por *ACLMessage* podem ser: assinadas digitalmente (*PKCS7Message*) de acordo com sintaxe de formação do padrão *Pkcs#7* (KALISKI, 1998); criptografadas (EncryptedMessage); e representar um *feedback* (FeedbackMessage) seguindo a estrutura do modelo *Dossiê*.

As mensagens na especificação FIPA devem incluir os campos apresentados na Tabela 4.3.

Tabela 4.3: Campos de uma mensagem FIPA ACL.

Campo	Categoria do Campo
Performativo	Tipo de ato comunicativo
Remetente	Participante da comunicação
Receptor	Participante da comunicação
Responde a	Participante da comunicação
Conteúdo	Conteúdo da mensagem
Linguagem	Descrição do conteúdo
Codificação	Descrição do conteúdo
Ontologia	Descrição do conteúdo
Protocolo	Controle da conversação
Id da conversa	Controle da conversação
Responda com	Controle da conversação
Em resposta a	Controle da conversação
Resposta por	Controle da conversação

O remetente da mensagem deve utilizar o campo *Conteúdo* para descrever os conceitos disponíveis na ontologia proposta. A mensagem a seguir, ilustra o uso do campo *content* para descrever o envio de um *dossiê* vazio, sem *feedbacks.*

```
(INFORM
 :sender  (agent-identifier :name agent_00001@127.0.0.1:1099/JADE :addresses (sequence
http://192.168.0.8:7778/acc ))
 :receiver  (set ( agent-identifier :name Agent_00002@127.0.0.1:1099/JADE ) )
 :content  "((SendDossier :dossier (Dossier :feedbacks :tree)))"
 :language  fipa-sl  :ontology  OpenJade  :conversation-id  DOSSIER )
```

O campo *content* encerra uma ação do tipo *SendDossier*, tomando como argumento uma instância de *Dossier* que, por sua vez, contém uma lista de *feedbacks* e sua respectiva árvore de *Merkle*. O exemplo a seguir apresenta uma mensagem com dois *feedbacks* de um dado *dossiê.*

```
((INFORM
:sender  ( agent-identifier :name agent_00001@127.0.0.1:1099/JADE)
:receiver  (set ( agent-identifier :name Agent_00002@127.0.0.1:1099/JADE ) )
:content "
  ((SendDossier
     :dossier (Dossier
       :feedbacks (sequence
         (Feedback
           :server (agent-identifier :name agent_00001@127.0.0.1:1099/JADE)
           :round 1
           :value great
           :client (agent-identifier :name agent_00002@127.0.0.1:1099/JADE))
        (Feedback
           :server (agent-identifier :name agent_00001@127.0.0.1:1099/JADE)
           :round 2
           :value great
           :client (agent-identifier :name agent_00002@127.0.0.1:1099/JADE)))
     :tree (MerkleTree
       :node (Node
         :value \"1c93ac8e23e1c54b7930d566f6e9c35\"
         :nodeR (Node :value fdef3e306d29a50aecb4b50c3caed9b)
         :nodeL (Node :value \"1380dc97c4dee35fbee0c12ee6f06333\")))))"
:language  fipa-sl  :ontology  OpenJade  :conversation-id  DOSSIER ))
```

A representação proposta permite construir um conjunto de mensagens que atende o modelo de confiança *Dossiê*. Conforme o tipo de comunidade virtual, outras representações ontológicas podem complementar a semântica da comunicação entre os agentes.

4.2 Sistema de Avaliação

Conforme dito no Capítulo 2, poucos trabalhos sobre modelos confiança dispõem de ferramentas computacionais de avaliação. Dentre as encontradas, a mais referenciada é o ART Testbed (Fullam *et al*. 2006). Entretanto, o projeto foi descontinuado em 2008. Uma alternativa frequente, na comunidade, passa pela construção de simuladores específicos para cada tipo de problema (Piunti, 2012), (Neville e Pitt, 2004) e (Dong-Huynha *et al*. 2004).

Em linha com a comunidade, propõe-se aqui um ambiente de simulação, denominado *SIMOC* (acrônimo de **Si**mulador de **Mo**delos de **C**onfiança). Ele permite avaliar diversos tipos de problemas ligados a confiança. Busca-se também generalizar problemas de confiança ligados a um ambiente multiagente aplicado ao contexto do *mercado de capitais*; tratado aqui de forma *lúdica*. O sistema SIMOC não tem a intenção de simular um *home broker*—sistema que conecta investidores a *pregões eletrônicos* do mercado de capitais—, seu objetivo é

oferecer um ambiente de competição e cooperação, por meio de um jogo, no qual os agentes têm o dilema de concorrer entre si e precisam compartilhar informações para atingir seus objetivos. Encontrar, em tempo hábil, bons parceiros para interagir é o principal desafio do jogo.

4.2.1 Descrição do Jogo no SIMOC

A ferramenta *SIMOC* define um jogo onde os agentes enfrentam o dilema de até onde devem interagir com outros competidores, pois a confiança entre parceiros, mesmo que concorrentes, pode estabelecer uma estratégia vencedora. O jogo apresenta as seguintes regras:

1) Ao iniciar, todos os agentes recebem igualitariamente um valor em pontos, creditados em sua *carteira* C_a;
2) O jogo é configurado com um número finito r de rodadas;
3) Para iniciar cada rodada, sorteia-se aleatoriamente para cada agente participante, um ação da bolsa de valores e um dia de pregão. O agente em questão recebe a série história dos últimos H pregões da ação. Por exemplo: para H = 90, ativo = VALE3, pregão = 24/01/2011, a série histórica será decrescente de 24/01/2011 à 26/10/2010, considerado o conjunto de 90 pregões;
4) Após a instanciação de cada agente participante (cf. passo o 3), as apostas podem ser lançadas. Como é feito? Cada agente participante pode fazer uma *aposta*, que consiste em informar se para a ação—que lhe foi atribuída—deve-se comandar a operação de *venda ou compra,* visando obter um retorno positivo em termos de acertos para P_+ pregões. Por exemplo em P_+= 5 o agente aposta na venda da ação, no pregão 24/01/2011, pois ele supõe que no pregão 29/01/2011 a ação valerá menos.
5) O valor da aposta V_{ap} fica a critério de cada agente que optar por um valor inferior ou igual a sua carteira: $0 < V_{ap} \leq C_a$;

6) Por *default*, uma aposta bem-sucedida consiste em, de um lado, vender quando o valor da ação em P_+ pregões for menor do que um dado valor de resistência (movimento de baixa—*movDown*), de outro lado, comprar quando o valor da ação em P_+ pregões for maior do que um dado valor de suporte (movimento de alta—*movUp*), portanto o resultado da aposta R para uma rodada r tem o seguinte critério:

$$R(r) = \begin{cases} 1, & se\ vendeu\ com\ movDown\ ou\ comprou\ com\ movUp \\ -1, & se\ vendeu\ com\ movUp\ ou\ comprou\ com\ movDown \end{cases}$$

7) Quando uma aposta for bem-sucedida, a carteira do agente investidor é creditada. O número de pontos creditados é dado em função da variação do valor da ação Δ_{ac}—na rodada r para o pregão P_+—multiplicado pelo valor da aposta V_{ap}. Caso a aposta seja malsucedida, a carteira é debitada, usando a mesma estratégia de cálculo para uma operação bem-sucedida. Por exemplo, para a venda de um ativo que subiu 4% após P_+ pregões, em aposta malsucedida no valor de 1.000 pontos, a perda será de 400 pontos.

8) Cada agente pode fazer apenas uma ou nenhuma aposta em cada rodada. A métrica de benefício da rodada $B(r)$ é calculada pela seguinte função:

$$B(r) = \begin{cases} R(r)\,.\,V_{ap}(r)\,.\,|\Delta_{ac}(r)|, & se\ apostou \\ 0, & se\ não\ apostou \end{cases}$$

9) Cada agente pode solicitar ou fornecer recomendações de compra e/ou de venda para outros agentes. As recomendações são recompensadas em pontos; assim um agente pode acumular pontos apostando ou servindo recomendações.

10) Ao término da última rodada, o agente com o maior número de pontos será o vencedor do jogo.

A partir dessas regras, cada agente competidor pode fazer uso de diferentes estratégias de jogo. Por exemplo, quando se detém a competência para avaliar uma certa *ação*, pode-se apostar sem requisitar recomendações de *outrem*. E quando se é perito em diferentes ações, pode-se vender recomendações para ampliar ganhos. Como o sistema atribui aleatoriamente a cada agente uma ação qualquer, pode-se esperar que o agente destinatário de uma dada ação ficará inseguro em alguns momentos e precisará dos outros agentes que conheçam a ação sorteada. Surge-se assim a necessidade de confiar em participantes desconhecidos. Tais agentes desconhecidos podem mentir em suas recomendações para superar seus rivais. Porém, o comportamento malicioso observado pode ser denunciado, e assim, os outros agentes

podem evitar futuras interações. Um agente bem-sucedido deve considerar as seguintes capacidades: analisar ações do mercado de capitais e selecionar bons agentes aliados.

4.2.2 Modelo de Dados

O *SIMOC* possui modelo de dados próprio, estruturado sob quatro entidades: *Configuração* que encerra as variáveis do jogo; *Agente* que mantem a definição de cada agente—-identificação e comportamento; *Ação* que dispõe do conjunto de ações utilizadas nas rodadas do jogo; e *Cotação* que mantem a série histórica das ações. Para identificar unicamente cada registro de dados, todos as entidades possuem um atributo *id*, conforme ilustrado no diagrama UML da Figura 4.8.

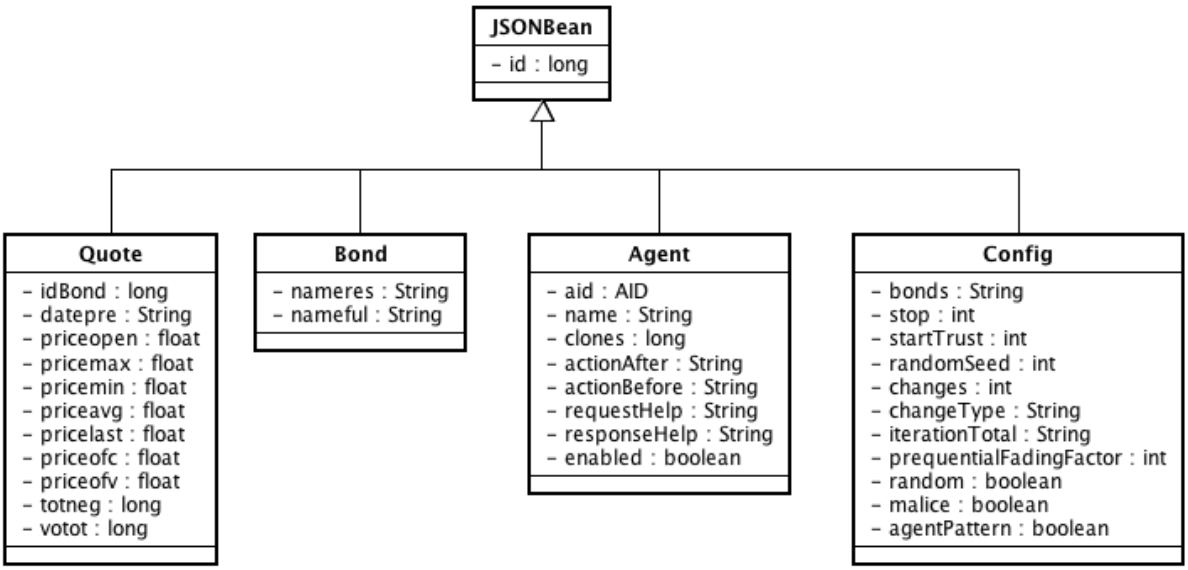

Figura 4.8: Modelo lógico de dados.

Por exemplo, no caso da ação *VALE3*, durante a importação de dados, o registro foi salvo com o identificador *1465352120159*. Assim, para obter todos os dados desta ação deve-se acessar o endereço: http://servidor:9200/simoc/bond/1465352120159. O resultado é apresentado conforme segue:

```
{"_index":"simoc","_type":"bond","_id":"1465352120159","_version":1,"found":true,"_
source":{"nomeres":"VALE3","nomefull":"VALE PNA N1","id":1465352120159}}
```

Para obter as cotações dessa ação, deve-se acessar o recurso *quote* e informar o identificador da ação (e.g., http://servidor:9200/simoc/quote/_search?q=idBond:1465352120159). O resultado da consulta é a lista de pregões da ação. Cada pregão possui as seguintes informações:

identificação da ação, data do pregão, valor de abertura, valor máximo, valor mínimo, valor médio, valor de fechamento e o volume negociado. Segue o exemplo das cotações para a ação VALE3 nos pregões do dia 04/01/2011 e 06/01/2011:


```
{"_index":"simoc", "_type":"quote", "_id":"1465352132136", "_score":8.441107,
"_source":{"idAcao":1465352120159, "datapre":"2011-01-04", "preabe":50.14, "pre-
max":50.83, "premin":50.1, "premed":50.57, "preult":50.83, "preofc":50.83,
"preofv":50.84, "totneg":26400, "quatot":20553600, "votot":103966275300,
"id":1465352132136}},

{"_index":" simoc ", "_type":"quote", "_id":"1465352149097", "_score":8.441107,
"_source":{"idAcao":1465352120159, "datapre":"2011-01-06", "preabe":51.5, "pre-
max":51.89, "premin":51.26, "premed":51.53, "preult":51.32, "preofc":51.3,
"preofv":51.33, "totneg":20362, "quatot":15451000, "votot":79620135100,
"id":1465352149097}}
```


Além de fornecer as cotações histórias, o SIMOC dispõe de visualizações gráficas. A Figura 4.9 exibe o gráfico da série história da VALE3 entre o período de 2011 à 2016.

Figura 4.9: Visão gráfica das séries históricas pelo SIMOC.

4.2.3 Comportamento de um agente SIMOC

Os agentes, no SIMOC, possuem comportamento orientado a mensagens e um conjunto de atividades customizadas. A Figura 4.10 resume os fluxos de atividades entre os agentes e o SIMOC.

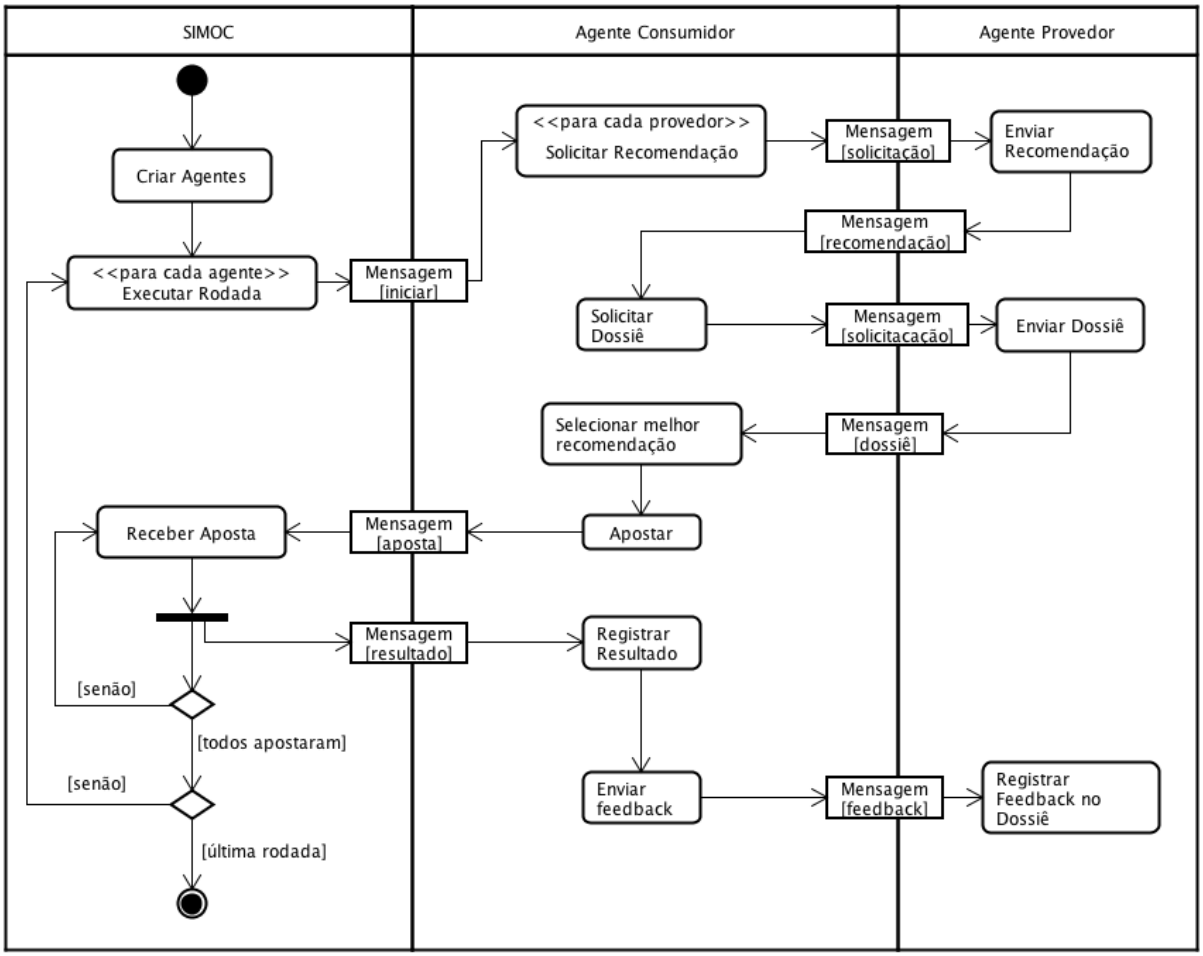

Figura 4.10: Máquina de estado para um agente SIMOC.

A Tabela 4.4 descreve as atividades do sistema SIMOC.

Tabela 4.4: Atividades do SIMOC.

Atividade	Descrição
Criar agentes	Constrói os agentes de acordo com as configurações salvas no SIMOC
Executar rodada	Envia, para cada agente, uma mensagem de início da rodada. Na mensagem consta também a *ação* selecionada e o dia de *pregão* para ser apostado.
Receber aposta	Recebe do agente consumidor uma mensagem com a aposta para a rodada presente. O sistema calcula o resultado da aposta e enviar uma mensagem de retorno ao agente consumidor. Quando todos os agentes enviam suas apostas, o SIMOC encerra a rodada, se estiver na última rodada é estabelecido o fim do jogo.

A Tabela 4.5 descreve as atividades dos agentes consumidores e provedores.

Tabela 4.5: Atividades dos agentes consumidores e provedores.

Campo	Categoria do Campo
Solicitar recomendação	Envia uma mensagem para um conjunto de agentes provedores solicitando recomendação para uma determinada *ação* sob um dia de pregão selecionado.
Enviar Recomendação	Retornar uma mensagem de recomendação.
Solicitar *Dossiê*	Solicita o *Dossiê* do agente provedor.
Enviar *Dossiê*	Envia o *Dossiê*
Selecionar melhor recomendação	Selecionar a melhor recomendação a partir do melhor índice de confiança que foi calculado em função dos feedbacks do *Dossiê*.
Apostar	Envia mensagem de aposta para o SIMOC
Receber Aposta	Recebe a aposta e computa seu resultado, envia mensagem de resultado ao agente consumidor.
Registrar resultado	Registar o resultado para eventual análise de desempenho.
Enviar *feedback*	Envia o *feedback* para o agente provedor que fez a recomendação da aposta.
Registrar feedback no *Dossiê*	Registra o *feedback* recebido pelo agente consumidor no seu *Dossiê*.

Algumas atividades são programadas diretamente nas classes do modelo de confiança, como nos casos de *Calcular Confiança* ou *Enviar Dossiê*. Outros comportamentos são partes diretas de cada agente e podem ser programados na própria interface do sistema. A Figura 2.10 ilustra a interface de configuração de um agente; aqui o agente está na atividade *apostar*.

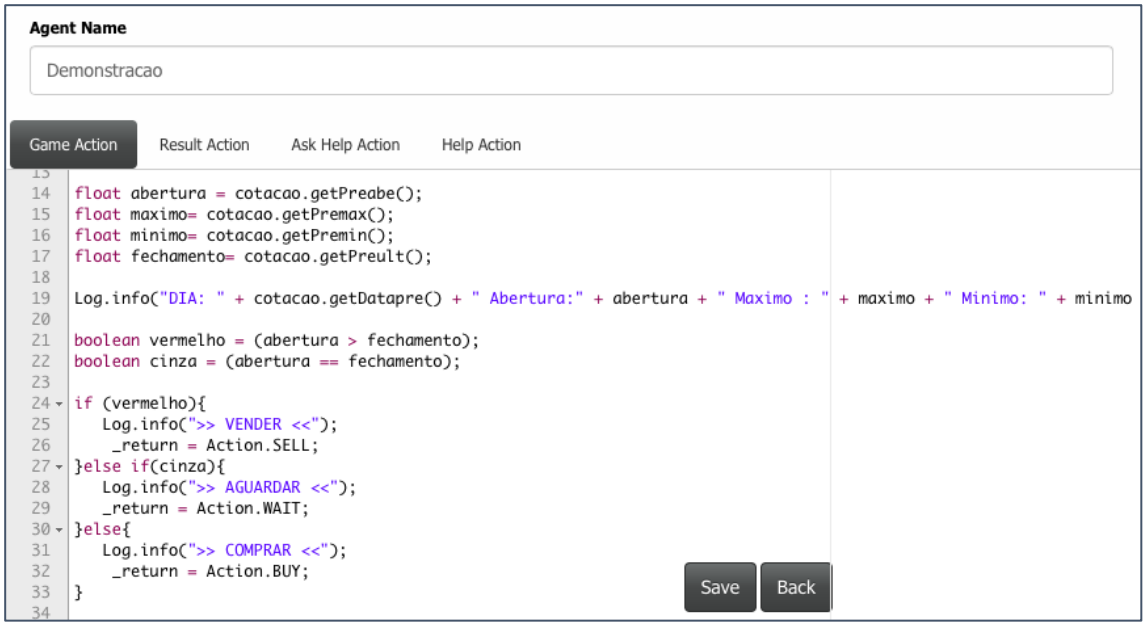

Figura 4.11: Interface de configuração de um agente SIMOC – atividade *apostar*.

No exemplo acima, da linha 14 até linha 17 são obtidos os valores de *abertura*, *fechamento, máximo e mínimo* de um dado pregão. Quando o valor de *abertura* é maior que o *fechamento*—linha 21—há a indicação de queda no valor da ação, situação *vermelha*. Quando os valores de *abertura* e *fechamento* são iguais—linha 22–há a indicação de estabilidade, situação *cinza*. A decisão de apostar ou não está codificada entre a linha 24 a linha 33, onde deve-se: *vender* se a situação é *vermelha*; não agir ou *aguardar* se a situação é *cinza*; e *comprar* caso contrário.

4.2.4 Modelagem do SIMOC

O SIMOC é uma ferramenta *web* construída sob o estilo arquitetural *RestFul API* (GUINARD, 2011), portanto todas as funcionalidades do sistema são acessadas por meio de serviços REST. A ferramenta, ilustrada na Figura 4.12, é projetada sob uma *plataforma responsiva*, i.e. se ajusta automaticamente ao dispositivo usado para acessar o serviço.

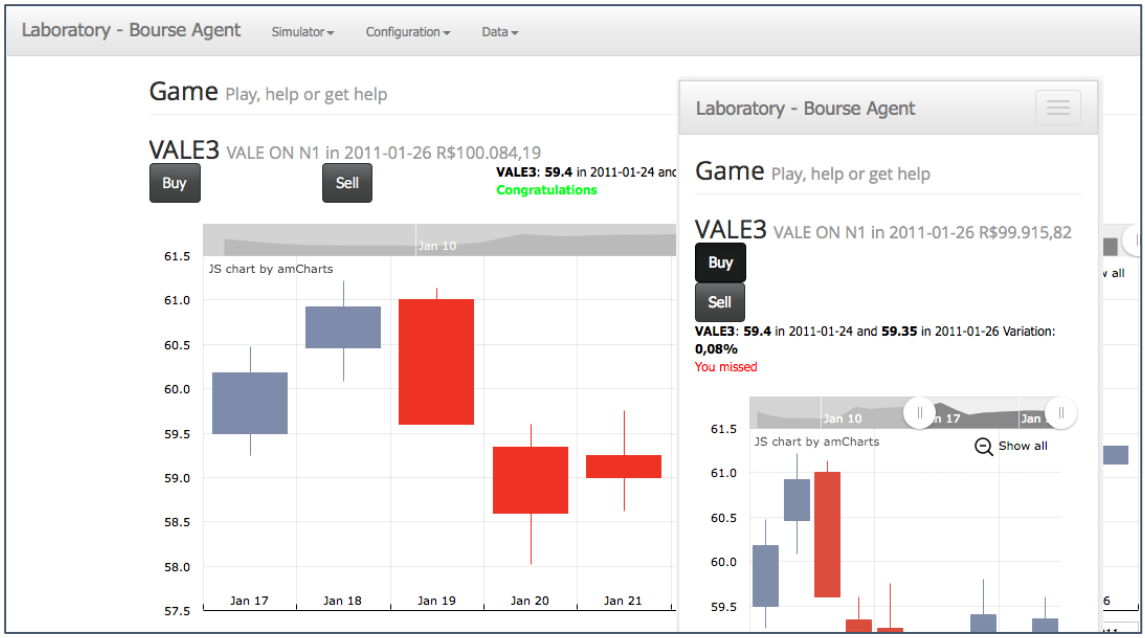

Figura 4.12: Interface responsiva para múltiplas plataformas

O projeto foi estruturado em três camadas (Figura 4.13): *Frontend*, responsável pela apresentação dos resultados e interação com usuários humanos; *Backend*, conjunto de *web services* no estilo *REST*, responsável pelo processamento da aplicação e manipulação dos agentes, interage diretamente com a camada *Frontend*; e *Database*, que prove o armazenamento de dados.

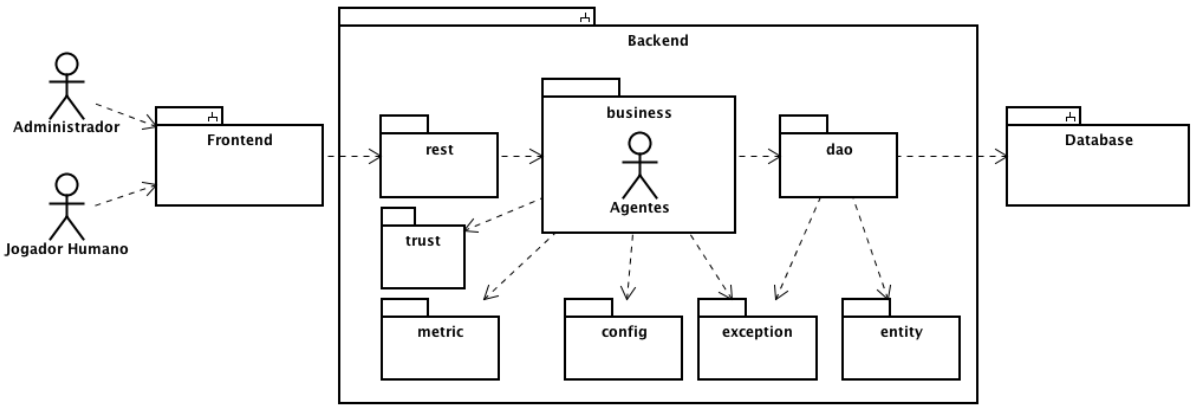

Figura 4.13: Arquitetura em camadas do SIMOC.

A camada *Backend* codifica a lógica do *jogo* e o ambiente para simulação dos agentes. Em detalhes, essa camada trata os serviços *REST*, as métricas dos experimentos, as configurações iniciais, os tratamentos de erros, as entidades, o acesso à base de dados, as regras do *jogo*, o ambiente virtual dos agentes e os eventos que disparam os comportamentos

dos agentes. A Figura 4.14 apresenta o modelo que fornece uma visão geral das principais entidades. A fachada do pacote *rest* é responsável por receber, da camada de apresentação *frontend*, as chamadas aos serviços REST e retornar as respostas vindas do pacote *business*. O pacote *business* é responsável por executar as regras de cada serviço exposto no pacote *rest*, os serviços de *CRUD*—acrónimo de ***C**reate*, ***R**ead*, ***U**pdate* e ***D**elete* na língua inglesa—são responsáveis por manter a persistência das entidades, como: configurações, ações, cotações, agentes dentre outras representadas pelo pacote *entity*. As operações de acesso a dados são realizadas no pacote *dao*.

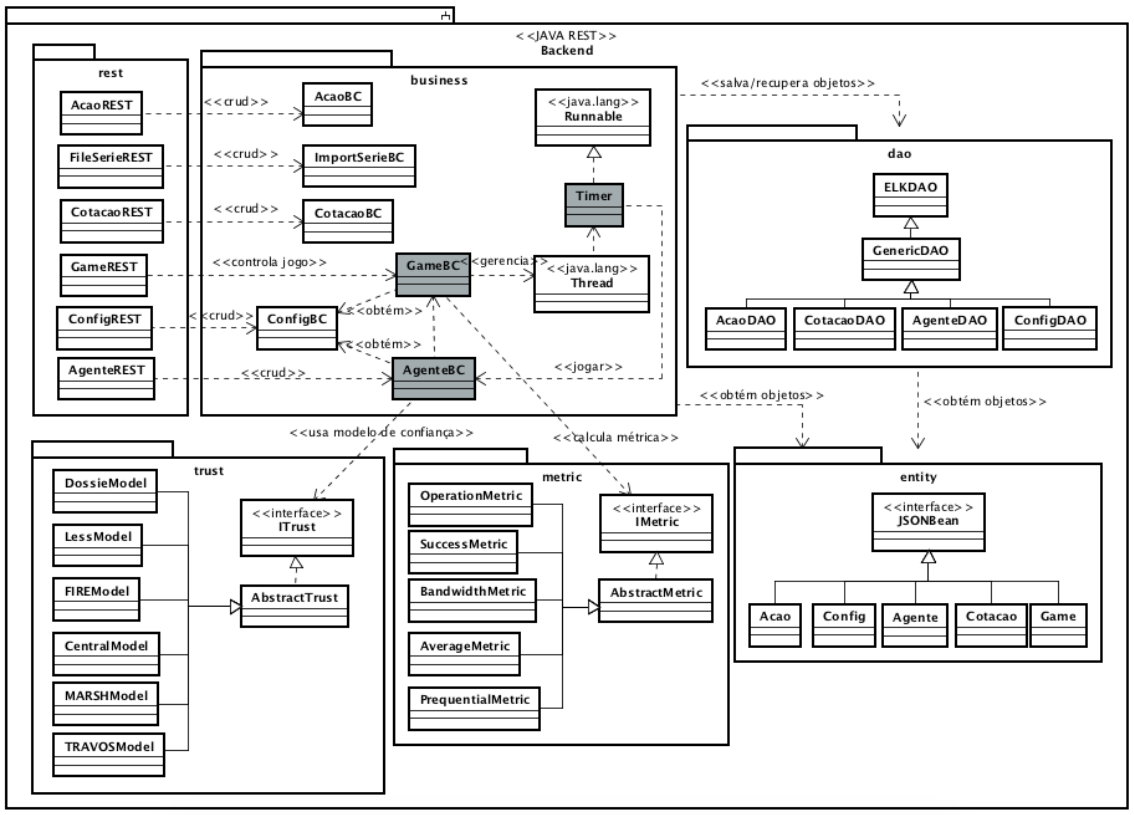

Figura 4.14: Pacotes e classes da camada *backend.*

Dentre as diversas classes da camada *Backend*, há três que se destacam: *GameBC*, *Timer e AgenteBC.* A classe *GameBC* é responsável por iniciar o jogo, criar os agentes e apresentar os resultados a partir de uma métrica selecionada pelo usuário, definida no pacote *metric*. A classe *Timer* instancia um *thread* de controle das rodadas, responsável por iniciar e finalizar os pregões, além de lançar os eventos para as ações dos agentes. A classe *AgenteBC* é responsável por criar os agentes.

4.2.5 Modelos de Confianças e Métricas de Avaliação

Outro pacote importante da camada *Backend* é o *trust*, responsável pelas classes que corresponde os modelos de confiança avaliados. Segue, na Tabela 4.6, essas classes e seus respectivos modelos:

Tabela 4.6: Modelos de confiança implementados.

Classe	Modelo de Confiança
LessModel	Ausência de modelo
TravosModel	Indireto (Teacy *et al*. 2006)
MarshModel	Direto (Marsh,1994)
CentralModel	Centralizado (Ebay, 2015)
FIREModel	Reputação Certificada (Dong-Huynha *et al*. 2004)
DossieModel	Modelo *Dossiê* proposto

O pacote *metric* possui as classes que colocam em prática as métricas de avaliação dos modelos de confiança, que são:

- *OperationMetric*: apresenta o número de operações que cada agente executa para processar o modelo de confiança. Por exemplo, solicitar um dossiê, enviar um *feedback*, selecionar um agente para interagir, são atividades que incrementam o número de operações. Diante do desafio para mensurar a complexidade de cada operação, atribuiu-se aqui peso igual para todas as operações.
- *SuccessMetric*: apresenta percentual taxa de acerto de cada agente em suas apostas. Esse percentual está diretamente relacionado com a capacidade de selecionar bons parceiros para interagir.
- *BandwidthMetric*: apresenta a quantidade de *bytes* trafegados—expresso em *Kb*. Ela é usada para mensurar a eficiência do modelo: quanto menor o tráfego de dados, melhor a eficiência.
- *AverageMetric*: apresenta a média das carteiras de todos os agentes. Ela é uma métrica de eficácia: quanto maior a média das carteiras, maior a eficácia da seleção dos parceiros.
- *PrequentialMetric*: apresenta a taxa de acerto das apostas dos agentes para as últimas *N* rodadas, onde *N* é chamado de *fator de desvanecimento*, pois reduz o impacto dos resultados mais antigos em relação aos mais recentes. Esta métrica ajuda na detecção

de mudanças no desempenho dos modelos, na medida em que a simples média aritmética se torna mais estável com o passar do tempo.

Os resultados são apresentados em tempo real ou ao final da rodada. A Figura 4.15 ilustra a saída gerada durante execução e acompanhamento de cada experimento.

Figura 4.15: Devolução da execução e do acompanhamento de um experimento.

Permite-se configurar diferentes experimentos, que vão desde o número de rodadas até a presença de agentes maliciosos, assim como a mudança de comportamento dos agentes provedores. A Figura 4.16 ilustra uma configuração de experimento registrada a partir da interface de configurações do *SIMOC*.

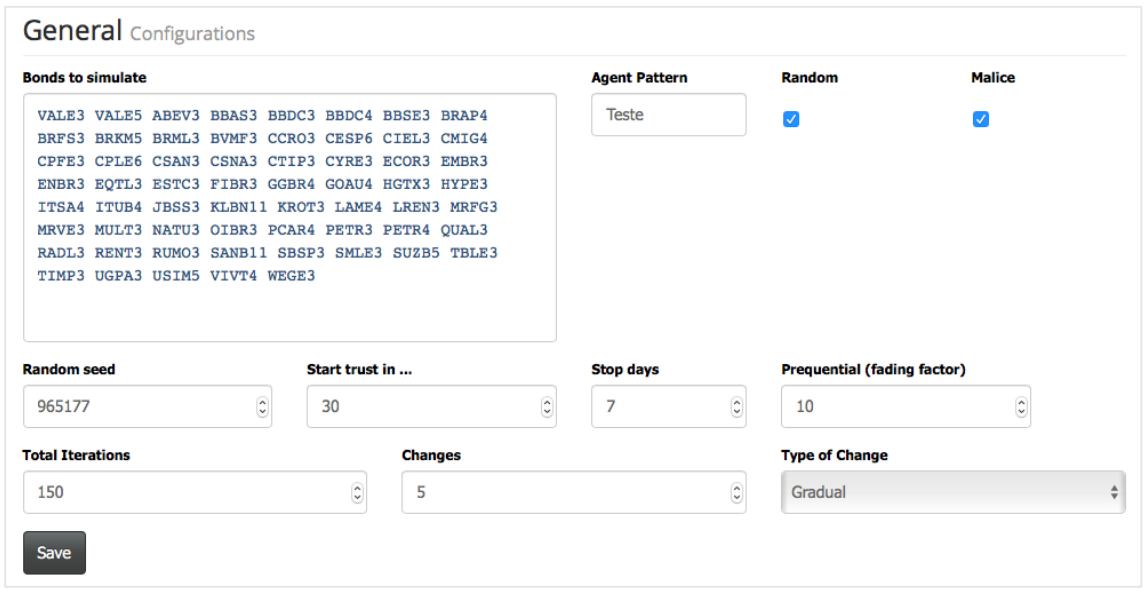

Figura 4.16: Configuração de um experimento.

Conforme apresentado, cada o experimento é definido com a relação das ações do *mercado de capitais* que serão utilizados – *bonds to simulate* –; o prefixo do nome dos agentes simulados – *agente pattern* –; se o sistema selecionará as ações e pregões aleatoriamente – *random* –; se haverá agentes maliciosos que tentarão omitir informações e se necessário mentir – *malice* –; semente para geração dos números aleatórios – *random seed* – permite reproduzir experimentos, mesmo que usem variáveis aleatórias; a partir de qual rodada o modelo de confiança será avaliado – *start trust in* –; número de pregões para avaliar a aposta – *stop days* –; para a métrica *prequential*, informar o fator de desvanecimento – *fading fator* –; o número de rodadas – *total iterations* –; o número de mudanças de desempenho dos agentes – *changes* – essas mudanças fazem, por exemplo, um agente tenha um desempenho reduzido ou ampliado, parâmetro importante para avaliar a adaptabilidade do modelo de confiança; e finalmente definir se a mudança dos agentes será abrupta ou gradual – *type of change*.

4.3 Considerações Finais

Neste capítulo foi apresentado o *Modelo de Confiança Dossiê*. Lembramos que tal proposta está centrada, de um lado, no uso de algoritmos de criptografia para garantir a segurança das informações trafegadas e, de outro lado, na manutenção da prerrogativa de que o armazenamento das avaliações sobre um dado agente deve ser local, i.e., mantidas localmente pelo próprio agente avaliado. Desta forma torna-se possível, de forma direta e segura, obter a reputação de um dado agente com uma única requisição. Neste capítulo também foi apresentada a proposta para avaliação genérica dos modelos de confiança sob diferentes cenários e métricas. O SIMOC implementa a proposta de avaliação de um conjunto de modelos de confiança e pode contribuir com a atual escassez de ferramentas neste campo de pesquisa. No próximo capítulo são apresentados os experimentos realizados e os resultados obtidos.

Capítulo 5

Resultados

Neste capítulo são discutidos os resultados obtidos nos experimentos. Esses últimos foram projetados para a avaliação do modelo de confiança *Dossiê*. Inicialmente será apresentada a metodologia aplicada para a avaliação. E na sequência, serão apresentados os resultados dos experimentos. Por fim, será apresentada a análise dos resultados.

5.1 Metodologia de Avaliação

Este trabalho utiliza bases de dados reais de ações negociadas na *Bovespa*[7], bolsa de valores de São Paulo. As séries históricas usadas compreendes os exercícios de 2011 a 2016. Dentre o conjunto completo de ações da *Bovespa*, 57.350 ações com respectivas 1.538.836 cotações, optou-se por utilizar apenas o subconjunto de ações que compõem o *índice Bovespa*. Esse subconjunto representa as ações mais negociadas no mercado brasileiro, listado no quadro a seguir.

```
VALE3 VALE5 ABEV3 BBAS3 BBDC3 BBDC4 BBSE3 BRAP4 BRFS3 BRKM5 BRML3 BVMF3 CCRO3 CESP6
CIEL3 CMIG4 CPFE3 CPLE6 CSAN3 CSNA3 CTIP3 CYRE3 ECOR3 EMBR3 ENBR3 EQTL3 ESTC3 FIBR3
GGBR4 GOAU4 HGTX3 HYPE3 ITSA4 ITUB4 JBSS3 KLBN11 KROT3 LAME4 LREN3 MRFG3 MRVE3
MULT3 NATU3 OIBR3 PCAR4 PETR3 PETR4 QUAL3 RADL3 RENT3 RUMO3 SANB11 SBSP3 SMLE3
SUZB5 TBLE3 TIMP3 UGPA3 USIM5 VIVT4 WEGE3
```

As métricas consideradas na avaliação dos modelos de confiança são: *taxa de acerto* e *processamento*. Para mensurar a taxa de acerto, optou-se pelo método da avaliação *Prequential* (Gama, Sebastião e Rodrigues, 2009). Essa métrica permite monitorar a

[7] Bovespa: http://www.bmfbovespa.com.br

eficiência dos modelos conforme sua evolução ao longo do tempo dado um fator de desvanecimento.

Os agentes do sistema foram organizados em dois grupos: *provedores* e *consumidores*. Agentes provedores são especializados em fornecer dicas de investimento e agentes consumidores são munidos de modelos de confiança para auxiliar na escolha de bons provedores. Deve-se frisar que, como os agentes *consumidores* utilizam exclusivamente o modelo de confiança para selecionar os provedores, o desempenho dos *mesmos* está diretamente ligado a taxa de acerto do modelo de confiança.

A eficiência dos agentes *provedores* é dada pelas configurações do experimento; isso permite reproduzir e comparar os resultados em momento futuro. Este trabalho **se exime em tratar técnicas de investimento**, apesar do sistema criado poder auxiliar futuros trabalhos nessa linha, os agentes *provedores* são configurados com percentuais de acerto previsíveis, porém desconhecidos pelos agentes consumidores. A Tabela 5.1 apresenta a distribuição dos agentes *consumidores* e *provedores*, bem como a quantidade utilizada de agentes e seus respectivos níveis de eficiência.

Tabela 5.1: Configuração dos agentes consumidores e provedores.

Tipo	Eficiência	Quantidade
Provedor *horrível*	10%	30
Provedor *ruim*	30%	20
Provedor *razoável*	50%	20
Provedor *bom*	70%	20
Provedor *excelente*	80%	10
Consumidores	--	30

Conforme apresentado na Tabela 5.1, o experimento foi configurado com 130: 30 *consumidores* e 100 *provedores*, com distintas faixas de desempenho. Os experimentos foram executados com seis configurações diferentes. Cada configuração encerra um conjunto de parâmetros (cf. Tabela 5.2).

Tabela 5.2: Parâmetros de configuração dos experimentos: cada coluna C define um cenário de execução.

Parâmetro	C1	C2	C3	C4	C5	C6
Semente Aleatória	965177	965177	965177	965177	965177	965177
Total de Rodadas	150	150	150	150	150	150
Rodada Inicial	30	30	30	30	30	30
Avaliar resultados	7	7	7	7	7	7
Desvanecimento	10	10	10	10	10	10
Número de mudanças	0	0	1	1	5	5
Tipo de Mudança	-	-	AB	GR	AB	GR
Malícia	N	S	S	S	S	S
Aleatório	S	S	S	S	S	S

O valor do parâmetro *semente aleatória foi mantido o mesmo para todos os cenários.* Geralmente, tal *semente* é definida por um número primo, serve para repetir os experimentos e obter o mesmo resultado. O *total de rodadas* indica o número de pregões simulados—cada pregão corresponde um dia. A *rodada inicial* indica em qual dia de pregão os modelos de confiança passam a serem considerados pelos agentes *consumidores*, antes disso os agentes selecionam seus parceiros aleatoriamente; esse período de seleção aleatória é utilizado para alimentar os modelos com alguns dados prévios da comunidade. O parâmetro *avaliar resultados* indica o número de pregões para avaliar o resultado de uma aposta. O *desvanecimento* indica o número de rodadas para calcular a métrica *prequential*. O *número de mudanças* define quantas vezes os agentes *provedores* terão sua eficiência modificada. O *tipo de mudança* estabelece, quando houver variação do comportamento dos *provedores*, como ela será: *abrupta* (AB) ou *gradual* (GR). O parâmetro *malícia* indica a presença de agentes *maliciosos*. Finalmente o parâmetro *aleatório* determinada a aleatoriedade da seleção nos ativos e pregões.

Os modelos de confiança considerados nos experimentos foram:

- **Travos**: modelo que utiliza predominantemente testemunhos para a construção da confiança (Teacy *et al*. 2006);
- **Marsh**: modelo baseado fortemente nas interações diretas entre os agentes (Marsh, 1994);
- **Central**: modelo baseado em testemunhos, que são disponibilizados para consulta a partir de uma fonte de informação central (Ebay, 2015);

- **FIRE**: modelo baseado na *reputação certificada* (Dong-Huynha *et al*. 2004);
- **Less**: não há modelo de confiança, a escolha dos seus parceiros é aleatoriamente; neste cenário é possível avaliar o alto grau de risco da comunidade e servir de valor base para comparação com os demais modelos.
- **Dossiê**: modelo de confiança proposto com testemunhos mantidos localmente; i.e., cada provedor mantém uma memória imutável com os *feedbacks* recebidos.

5.2 Avaliação dos Modelos

O primeiro experimento consistiu na avaliação dos modelos com a métrica *taxa de acerto* para a configuração C1, descrita na Tabela 5.2. Essa configuração apresenta apenas agentes honestos, i.e., que não mentem ou omitem as informações, além disso, os agentes *provedores* possuem eficiência constante. O resultado desse experimento é apresentado no gráfico da Figura 5.1:

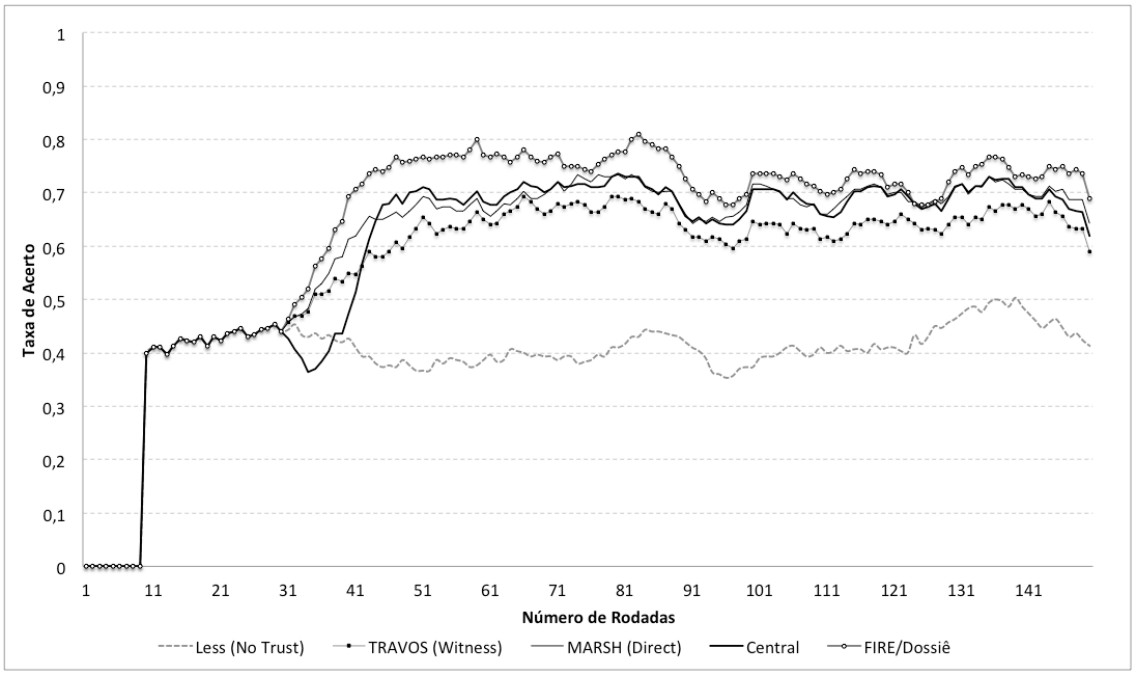

Figura 5.1: Evolução do desempenho para configuração C1 – agentes honestos com desempenho constante.

O modelo *Less (No Trust)* assume a seleção aleatória de agentes *provedores*, por ser a estratégia menos sofisticada, sua eficiência serve para nortear a avaliação dos demais, i.e., quanto mais próximo do *Less* mais ineficiente será o modelo. O modelo *Less* também contribui no cálculo da eficiência dos *provedores*, por ser uma escolha aleatória,

estatisticamente os *provedores* serão requisitados com frequência semelhante. Neste primeiro experimento é possível verificar que a média de acerto dos *provedores* da rodada 10 até a rodada 30 é de aproximadamente 41%.

A análise da Figura 5.1, sem considerar o modelo *Less*, indica que os modelos, ao final do experimento, mostram resultados percentualmente próximos, com diferenças de 6% à 10% entre o pior e o melhor resultado. Os modelos FIRE e *Dossiê* tiveram uma pequena vantagem em relação aos demais modelos. Tal vantagem foi mais acentuada nas primeiras rodadas, mas com o passar das rodadas, os demais modelos chegaram a resultados cada vez mais próximos do FIRE e Dossiê. Este resultado pode ser explicado pela estabilidade do cenário configurado. Como todos os agentes *provedores* possuem o mesmo desempenho durante todas as rodadas, os modelos conseguem, uns mais rápidos, outros mais lentos, identificar os melhores *provedores* e nas últimas rodas a maioria dos agentes *consumidores* conseguem selecionar esses *provedores* o que explica a eficiência aproximada.

Os modelos FIRE e Dossiê apresentaram resultados semelhantes, pois ambos utilizam a mesma abordagem que mantem as avaliações no agente avaliado, porém o modelo FIRE não possui mecanismos contra agentes maliciosos. Visando diferenciar esses dois modelos, o próximo experimento é definido com a configuração C2, cuja diferença à configuração C1 é dada apenas pela presença de agentes maliciosos. Os resultados são apresentados pela Figura 5.2.

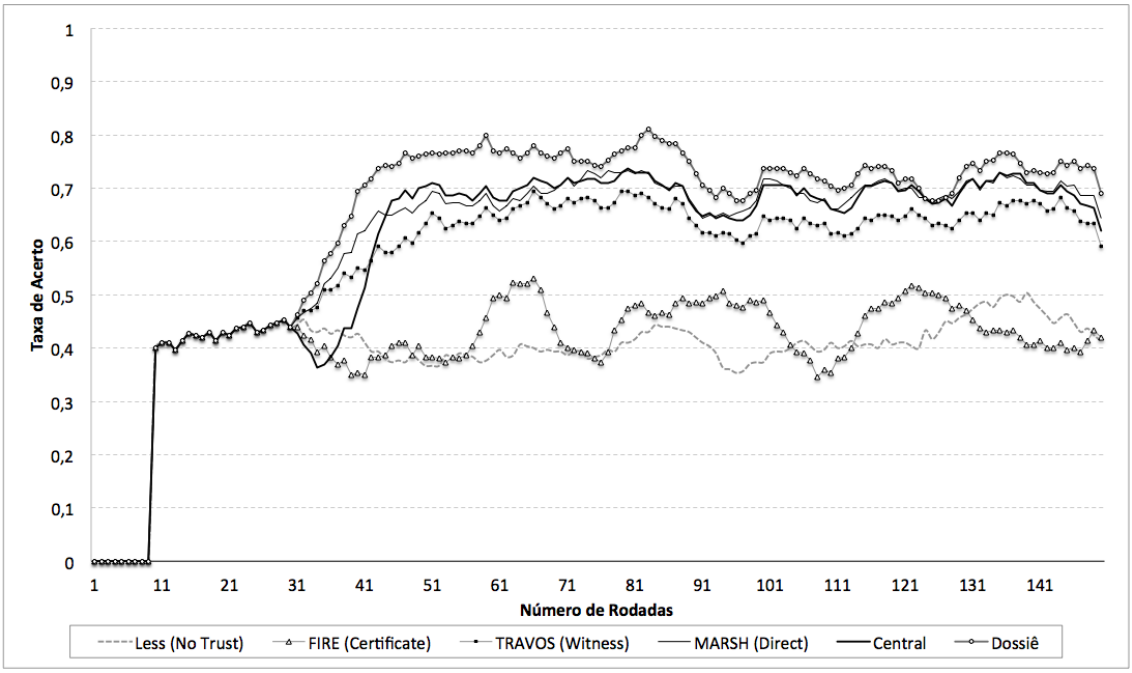

Figura 5.2: Evolução do desempenho para a configuração C2 – gentes maliciosos com desempenho constante.

Os resultados desse experimento mostram uma significante redução de eficiência do modelo FIRE, quando na presença de agente maliciosos, em alguns momentos, sua eficiência foi inferior ao modelo *Less*. Nessa configuração, os agentes *provedores* com baixa eficiência modificam seus *feedbacks* negativos para valores positivos, dessa forma torna-se impossível, a partir das suas avaliações, diferenciar os bons dos maus agentes. Por conta dos algoritmos criptográficos do modelo *Dossiê* os agentes maliciosos não conseguem adulterar seus *feedbacks*, pois são facilmente validados por sua assinatura digital.

No experimento a seguir, com a configuração C3, os agentes maliciosos continuam atuando, semelhante a configuração C2, porém na rodada 75, os agentes *provedores* mudam sua eficiência de forma abrupta, neste caso, provedores *horríveis* se tornam *ótimos*, provedores *bons* se tornam *ruins* e os provedores *razoáveis* mantém sua eficiência. A Figura 5.3 apresenta os resultados dessa configuração. A linha vertical tracejada representa o momento da mudança.

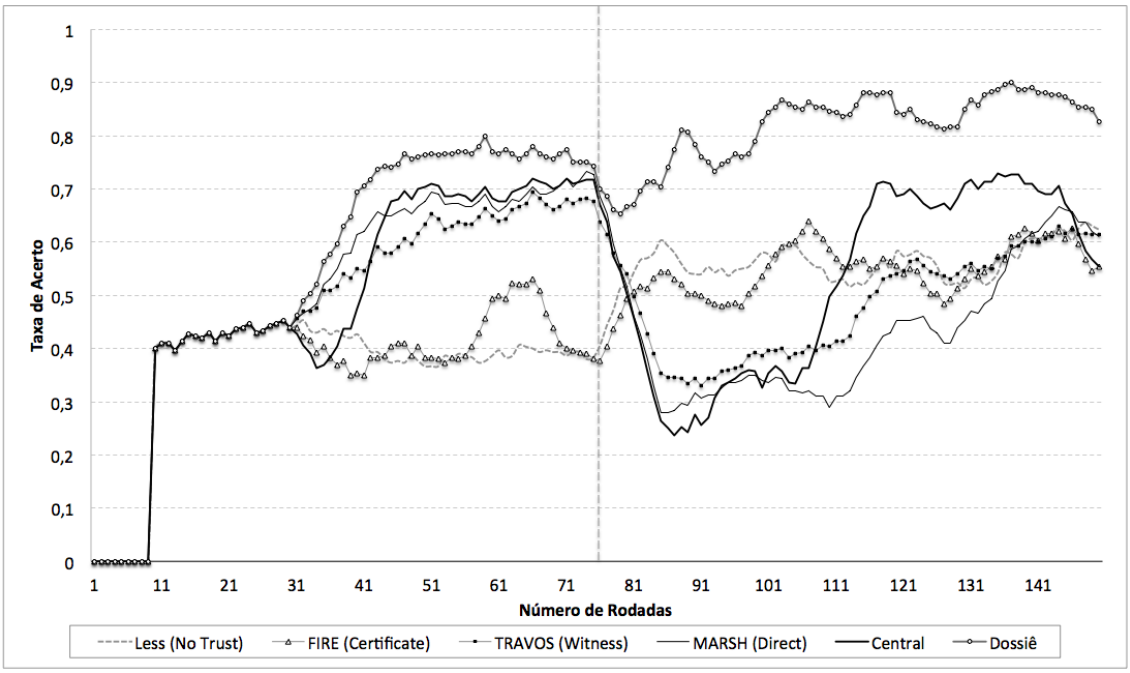

Figura 5.3: Evolução do desempenho para a configuração C3 – agentes maliciosos com uma variação abrupta de desempenho.

Nessa configuração é possível observar a similaridade dos resultados com o experimento anterior até a rodada 75, i.e., enquanto o desempenho dos agentes *provedores*

quase não se altera. A partir da rodada 76 é possível notar a queda brusca no desempenho dos modelos *Marsh*, *Travos* e *Central*. Uma explicação para este fato, no caso do *Marsh*, é a baixa capacidade de adaptação à mudança de comportamento do modelo direto, face a necessidade de constantemente de interação entre os agentes de uma comunidade para detectar uma mudança de comportamento; a abordagem direta requer várias rodadas para perceber a detecção da mudança de desempenho. No caso do *Travos*, a complexidade para encontrar testemunhas que tenham identificado uma mudança de comportamento dos agentes *provedores* também provoca certo atraso na resiliência de tal abordagem indireta. Contudo, a estratégia *Travos* se mostrou ligeiramente superior ao *Marsh* na detecção de mudança de comportamento. No caso do modelo *Central*, por considerar todo o histórico de cada agente *provedor*, mudanças bruscas são menos perceptíveis, necessitando de número mais elevado de rodadas para a detecção de mudança de comportamento. De qualquer forma, o modelo *Central* foi superior ao *Marsh* e *Travos*, pois a centralização dos dados sobre os *provedores*, facilita a detecção de mudança de comportamento. Os modelos FIRE e *Dossiê* foram os menos impactados com a mudança de comportamento dos agentes *provedores*. Esta situação era esperada pois a estrutura do *dossiê* foi projetada para reduzir o peso das avalições mais antigas; desta forma a detecção de mudança é feita de maneira mais tempestiva.

O próximo experimento, dado pela configuração C4, aborda o comportamento dos modelos quando a variação de eficiência dos agentes *provedores* acontece de modo gradual, i.e., a cada rodada agentes *provedores excelentes* e *bons* invertem progressivamente sua eficiência no sentido de menor eficiência e *provedores péssimos* e *ruins* mudam paulatinamente de sua eficiência para melhor. Em ambos casos a inversão só é plenamente atingida na última rodada. Os resultados obtidos podem ser observados na Figura 5.4.

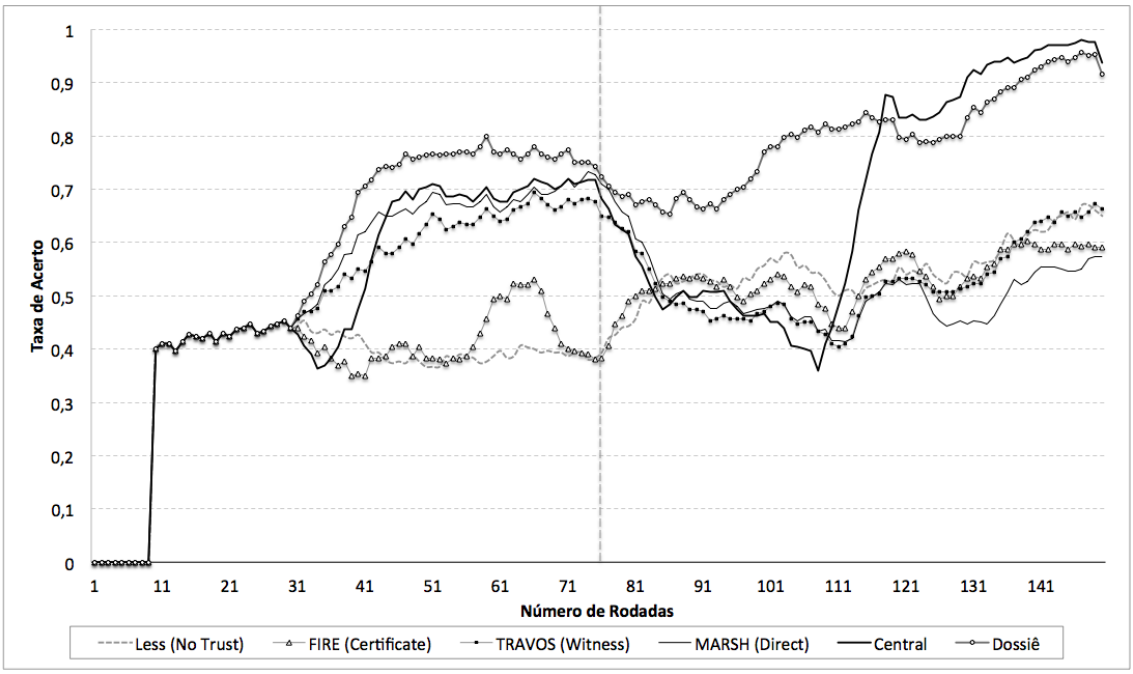

Figura 5.4: Evolução do desempenho para a configuração C4 – agentes maliciosos com uma variação gradual de desempenho.

Semelhante ao experimento anterior, a partir da rodada 76, os modelos *Marsh*, *Travos* e *Central* apresentam queda de eficiência; mas um pouco mais suave. Por exemplo, no modelo *Marsh*—na configuração C3—a redução de eficiência leva a taxa de acerto para 0,3 e na configuração C4 a taxa é mantida acima de 0,4. A mudança gradativa de desempenho dos agentes *provedores* é um cenário mais favorável para detecção de mudanças para estes três modelos. O modelo *Central* se recuperou a partir da rodada 110, obtendo um resultado similar ao *Dossiê*, e até mesmo um pouco superior a partir da rodada 118. O *Dossiê* foi o modelo menos impactado com a mudança ocorrida na rodada 75, seu desempenho caiu certa de 10% e recuperou-se a partir da rodada 90.

O próximo experimento, dado pela configuração C5, avalia a situação que combina mudanças abruptas dos agentes *provedores* com mais frequência. Neste cenário foram configuradas 5 mudanças, uma a cada 25 rodadas; cada mudança está marcada com linha tracejada na vertical da Figura 5.5.

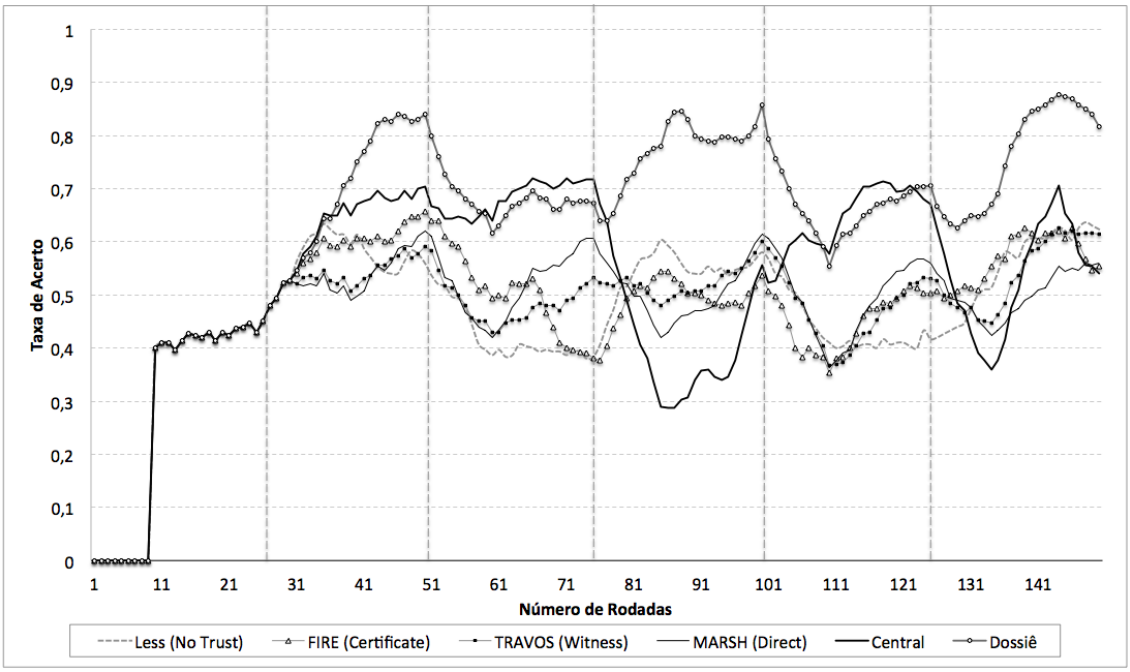

Figura 5.5: Evolução do desempenho para a configuração C5 – provedores maliciosos com cinco variações abruptas de desempenho.

No primeiro momento de mudança, rodada 25, não houve variação dos resultados, pois os modelos ainda estravam na fase de coleta de dados. A partir da rodada 30, os modelos de confiança iniciam a seleção dos parceiros. Nas demais mudanças de comportamento dos *provedores*, observa-se a queda de eficiência com maior intensidade dos modelos *Central*, *Marsh* e *Travos*. Os modelos *FIRE* e *Dossiê* apresentam maior tolerância às mudanças. O *Dossiê*, entretanto, é superior ao *FIRE* por conta dos seus mecanismos contra agentes *provedores* maliciosos. Esses resultados evidenciam que a função de decaimento para os *feedbacks* mais antigos resulta em uma melhor capacidade de adaptabilidade as mudanças de comportamento dos agentes *provedores*.

O último cenário de avaliação, definido pela configuração C6, assemelha-se ao cenário anterior. Aqui, as mudanças de comportamento nos agentes *provedores* acontecem de modo gradual ao longo do tempo. A Figura 5.6 apresenta os resultados para essa configuração.

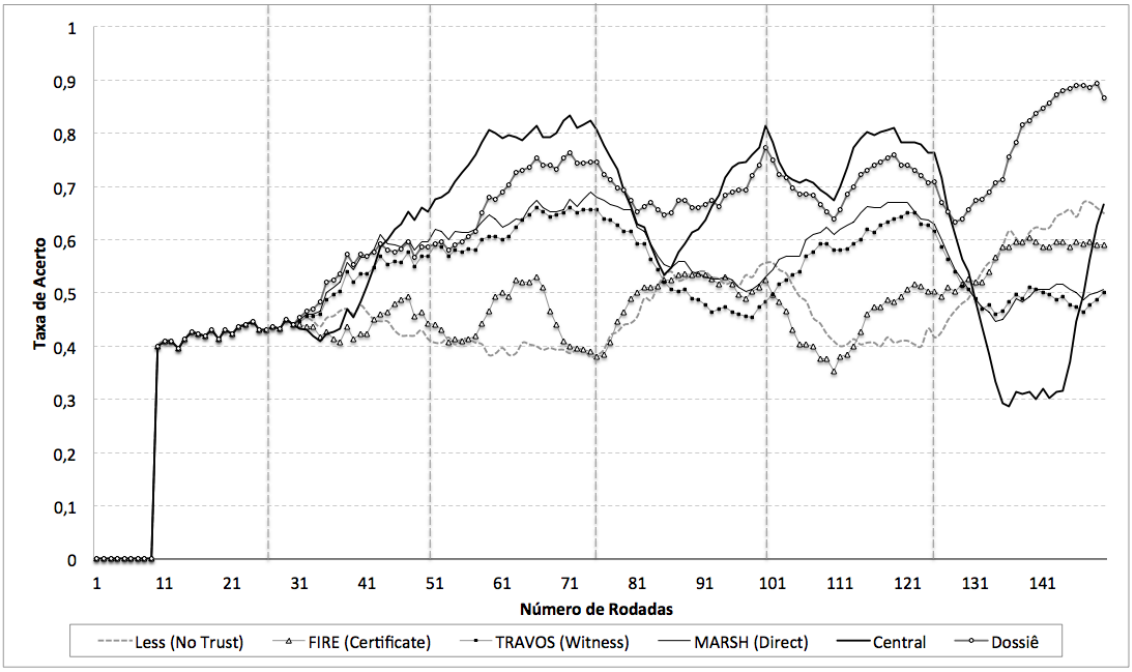

Figura 5.6: Evolução do desempenho para a configuração C6 – *provedores* maliciosos com cinco variações graduais de desempenho.

Semelhantemente ao experimento do cenário C5, os modelos *Central*, *FIRE*, *Marsh* e *Travos* apresentaram reduções mais acentuadas de eficiência no momento das mudanças de comportamento dos agentes *provedores*. O modelo *Dossiê* apresentou-se uma eficiência um pouco inferior ao modelo *Central* até a rodada 130, na sequência, ele foi superior aos demais modelos até a última rodada. Os últimos dois cenários avaliados, C5 e C6, são os mais complexos a serem tratados pelos modelos, à medida que eles mesclam os desafios de lidar com agentes maliciosos e frequentes alterações de comportamento dos agentes *provedores*.

Para fins desse trabalho, a eficiência dos modelos é determinada pela métrica *taxa de acerto*. Porém, o lucro dos agentes, na perspectiva do jogo proposto pelo sistema SIMOC, permite diferenciar os modelos de confiança sob outra grandeza de valor. O *lucro* representa a média de todos os rendimentos obtidos nas apostas dos agentes *consumidores*. Nos experimentos os agentes consumidores iniciaram a *carteira* com R$ 100.000,00. A Figura 5.7 apresenta a métrica de lucro para a configuração C5 e a Figura 5.8 para a configuração C6.

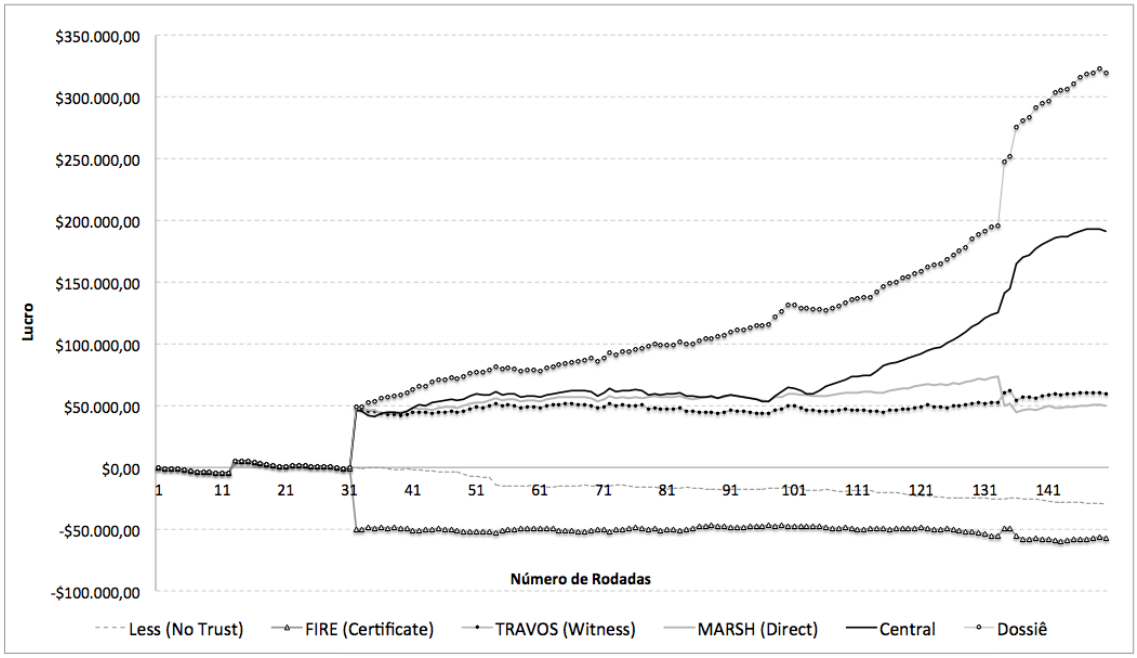

Figura 5.7: Evolução do lucro para a configuração C5 – provedores maliciosos com cinco variações abruptas de desempenho.

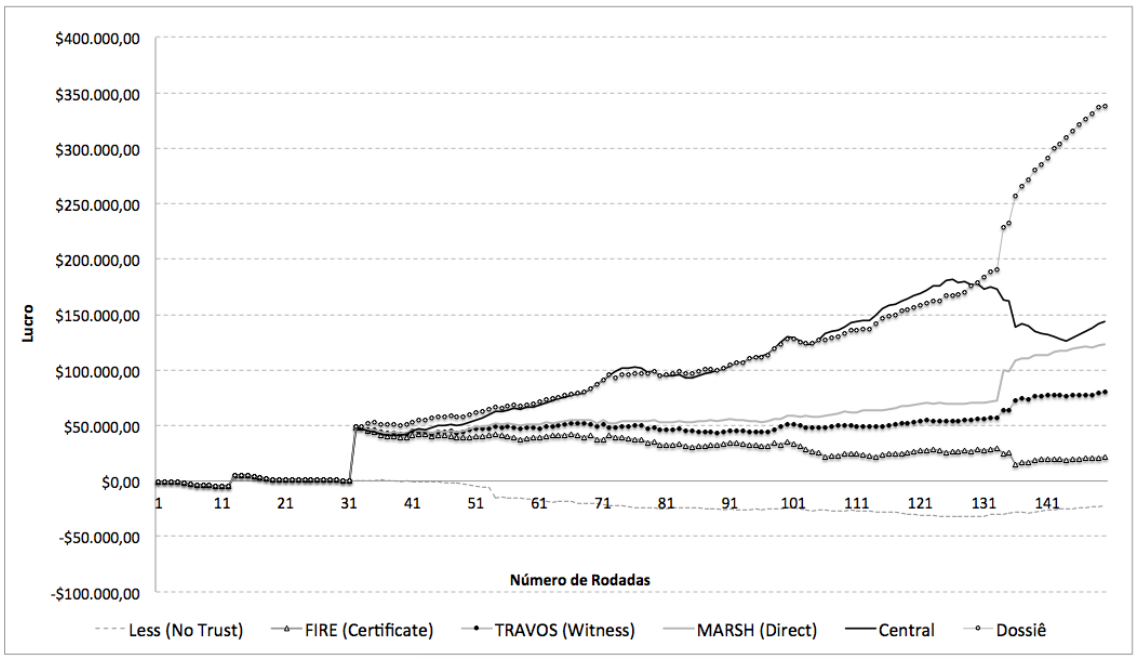

Figura 5.8: Evolução do lucro para a configuração C6 – *provedores* maliciosos com cinco variações graduais de desempenho.

Os resultados apresentados para a métrica *lucro*, destacam a superioridade do modelo *Dossiê* em relação aos demais. Para a configuração C5 os agentes com o modelo *Dossiê lucram* R$ 320.000,00, resultado 168% superior aquele com o modelo *Central*, segundo

colocado, que *lucrou* R$ 190.000,00. Para o cenário C6, os agentes com o modelo *Dossiê* lucram R$ 338.000,00, em quanto que o modelo *Central* apresentou um *lucro* de R$ 143.000,00, menos da metade do primeiro colocado.

Adicionalmente, as Figura 5.9 e Figura 5.10 apresentam comparativos para o número de operações realizadas pelos agentes. Os resultados auxiliam na avaliação do custo computacional de cada modelo.

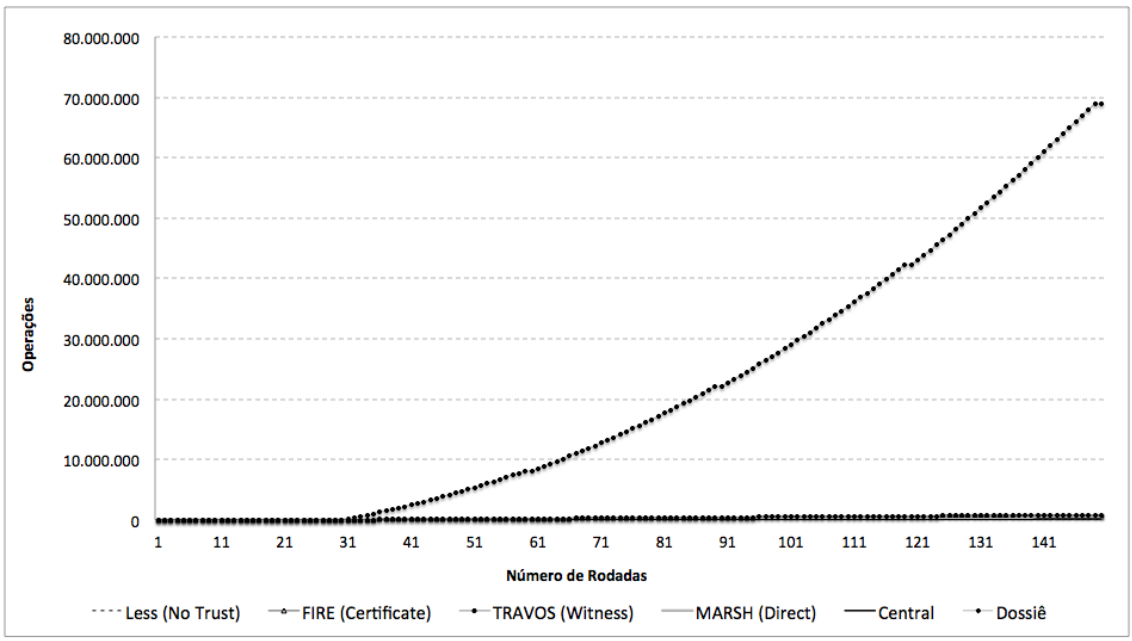

Figura 5.9: Somatório de operações em cada rodada.

Os resultados apresentados pela Figura 5.9 mostram o alto custo computacional para construção de modelos indiretos. As atividades para descoberta de testemunhas e frequentes interações, provocam constante troca de mensagens, elevando o número de operações em ritmo exponencial. Devido ao alto número de operações do modelo *Travos*, os demais modelos ficaram visualmente sobrepostos no gráfico. Para complementar a análise desse experimento, a Figura 5.10 apresenta o mesmo resultado, porém omitindo o modelo *Travos*.

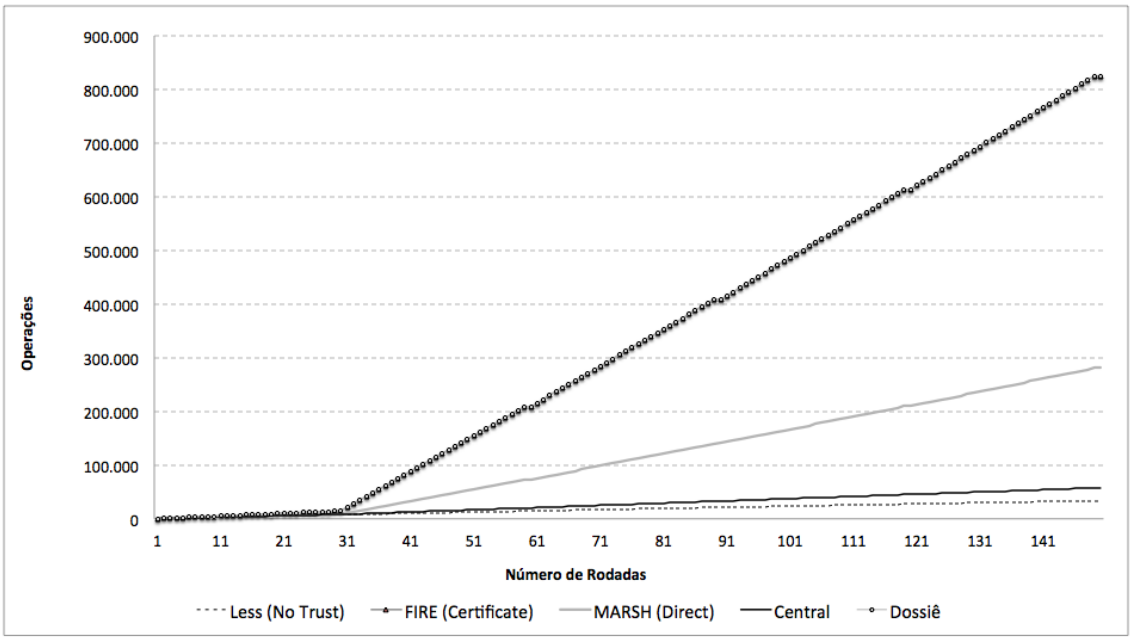

Figura 5.10: Somatório de operações em cada rodada sem o modelo *Travos*

O gráfico apresentado acima, sem o modelo *Travos*, mostra com mais clareza o modelo *Central*, com o menor número de operações, chegando a 58.000 operações para as 150 rodadas do jogo. Em seguida o *Marsh* com 5 vezes mais operações em relação ao modelo *Central* e finalmente os modelos FIRE e *Dossiê*, com o maior consumo, cerca de 14 vezes maior que o modelo *Central*. A diferença de desempenho entre os modelos—em termos de número de operações—pode ser explicada pela natureza dos sistemas descentralizados *versus* monolíticos. No modelo centralizado, o acesso a informação é feito de modo direto, sem necessidade de técnicas mais sofisticadas para armazenar ou compartilhar as avaliações dos agentes *provedores*. O modelo direto de confiança, representado pelo *Marsh*, é baseado em uma arquitetura distribuída que necessita do processamento de cada nó, mesmo que simplificado, esse mecanismo requer maior esforço computacional em relação ao modelo *Central*. O modelo *Dossiê* também se vale de uma arquitetura distribuída. Todavia, esse modelo requer mecanismos mais sofisticados para o tratamento da integridade da informação, que são: criptografia de dados; transmissão de *feedbacks*; manutenção do *Ledger distribuído*; compartilhamento do *dossiê; e o* tratamento de agentes maliciosos. Esse conjunto de mecanismo aumenta a necessidade processamento. Entretanto, sob a ótica de uma arquitetura distribuída, o processamento descentralizado permite alcançar escalabilidade mais efetiva em relação as arquiteturas monolíticas.

5.3 Análise dos Resultados

Esta seção apresenta a análise consolidada dos resultados obtidos com base nos experimentos descritos e colocados em prática anteriormente. Tal análise fundamenta-se no método estatístico não paramétrico de Milton Friedman (FRIEDMAN, 1937), para verificar a hipótese de que os modelos de confiança podem diferenciar vis-à-vis a eficiência dos agentes. Ele aplica-se quando há três ou mais situações experimentais. Neste trabalho, analisou-se os resultados obtidos a partir das execuções de seis experimentos dados pelas configurações: C1 à C6 (cf. Tabela 5.2). Adicionalmente, aplicou-se o teste *pós-hoc* de *Nemenyi* (NEMENYI, 1963) para identificar quais modelos de confiança provocaram diferenças significativas.

Aqui, a métrica *taxa de acerto* foi usada para mensurar a eficiência de cada modelo de confiança. Observou-se que no início de cada experimento os modelos encontravam-se em fase de ajuste, logo, para fins de análise, fez necessário usar a média da *taxa de acerto* a partir da segunda metade dos experimentos, i.e., entre as rodadas 75 e 150; neste intervalo tem-se um comportamento melhor definido. A matriz *F* a seguir, apresenta as médias dos cinco modelos de confiança para os seis experimentos realizados:

	70	65	73	73	69	**[C1]**
	70	65	45	73	69	**[C2]**
***F* =**	45	49	54	82	54	**[C3]**
	51	53	53	80	69	**[C4]**
	50	51	50	73	54	**[C5]**
	57	54	50	73	63	**[C6]**
	MARSH	**TRAVOS**	**FIRE**	**DOSSIÊ**	**CENTRAL**	

Dada matriz *F*, o teste de *Friedman* resultou em um *p-value* de 0,0067. Tal valor é inferior ao nível de significância de 0,05. Portanto, pode-se rejeitar a hipótese *nula* e concluir que pelo menos um dos cinco modelos tem efeito diferenciado ao desempenho dos agentes. Para identificar tal diferença nos modelos aplicou-se o teste de *Nemenyi*; esse último permite a comparação um a um. A matriz *N* a seguir apresenta o resultado desse teste.

	1,000	-	-	-	**TRAVOS**
	1,000	0,999	-	-	**FIRE**
***N* =**	**0,029**	**0,016**	**0,038**	-	**DOSSIÊ**
	0,705	0,588	0,759	**0,470**	**CENTRAL**
	MARSH	**TRAVOS**	**FIRE**	**DOSSIÊ**	

O resultado do teste mostrou que há diferença significativa entre o modelo de confiança Dossiê e os demais modelos com nível de significância inferior a 0,05. A Figura 5.11 apresenta o gráfico de *boxplot*, também conhecido por *diagrama de extremo e quartis*, que ilustra a variação da taxa de acerto dos modelos sob os seis experimentos realizados.

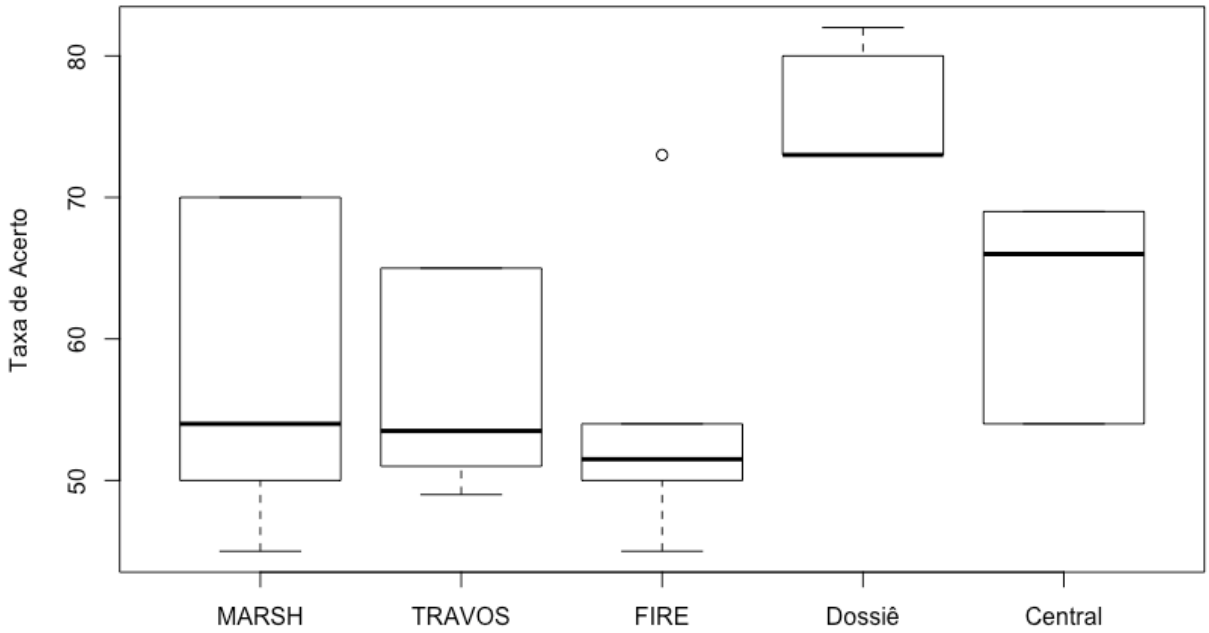

Figura 5.11: Boxplot da taxa de acerto dos modelos de confiança para os seis experimentos.

A partir dos testes de *Friedman* e *Nemenyi* observa-se estatisticamente a superioridade do modelo Dossiê em relação ao demais. Para cada modelo avaliado, pode-se dizer o que segue: (i) O modelo *FIRE* apresentou o pior resultado quando estava na presença de agentes maliciosos (cf. Figura 5.2). Em ambientes seguros, onde não há agentes maliciosos, seu desempenho é semelhante ao modelo *Dossiê* (cf. Figura 5.1). Isso se explica à medida que ambos os modelos operaram com o mesmo conjunto de informações e utilizaram a mesma estratégia para a seleção de parceiros. (ii) Os modelos *Marsh* e *Travos* obtiveram resultados semelhantes. Esses resultados são explicados pelos custos elevados para detectar bons parceiros, uma vez que precisam interagir constantemente para manter suas percepções de confiança atualizadas. (iii) Os modelos *Central* e *Dossiê* apresentam os melhores resultados. Porém, em cenários com maior variabilidade de comportamento, o modelo *Dossiê* foi superior, em termos de *taxa de acerto*. Considerando os desafios naturais para o compartilhamento de informação sob uma arquitetura descentralizada, a abordagem *Dossiê* foi capaz de prover informações para a tomada de decisão de forma semelhante ao modelo

centralizado; mas com qualidade superior quando se trata de informação de impacto, i.e., uma mudança de conceito—abrupta ou gradual—é percebida mais rapidamente.

5.4 Considerações Finais

Este capítulo apresentou a metodologia de avaliação dos modelos de confiança e os experimentos, em especial, o modelo proposto *Dossiê*. Foram realizados experimentos sob bases de dados reais a partir de um *mercado de capital* existente e os resultados foram comparados com seis modelos de confiança distintos. Um dos principais desafios tratados foi a descoberta de bons parceiros para interagir em ambientes virtuais abertos. Em geral, o modelo proposto, baseado numa estrutura de dados distribuída e segura, permitiu vislumbrar uma eficiência satisfatória, sobretudo nas simulações mais complexas que envolvem agentes maliciosos e variabilidade de comportamento. A partir dos resultados obtidos, pode-se concluir que o modelo *Dossiê* é uma alternativa eficiente para problemas de confiança em ambientes virtuais abertos semelhante aos encontrados em sistemas multiagentes.

Capítulo 6

Conclusão

Neste trabalho, foi apresentado o *Dossiê*, um modelo de confiança para *sistemas multiagentes*. Os *feedbacks* trocados entre os agentes de um sistema são tratados na perspectiva de reduzir as atuais limitações dos modelos baseados nas *experiências diretas* e nas *experiências indiretas*. As fontes de informação diretas possuem baixo desempenho vis-à-vis a dificuldade de um agente realizar um número significativo de interações com agentes estranhos e as fontes indiretas dependem do desprendimento de testemunhas para compartilhar suas experiências. O modelo *Dossiê* permite dispor informações sobre qualquer indivíduo de uma comunidade virtual aberta de maneira rápida e segura, rápida por conta do acesso direto aos feedbacks/avaliações a partir do agente examinado/alvo e segura por conta a imutabilidade garantida pela estrutura de dados do *Dossiê*.

Apesar do modelo *Dossiê* ser avaliado sob a perspectiva de um sistema multiagente é importante ressaltar que essa proposta pode ser expandida para outros sistemas com características similares, como a ausência de controles centrais e distribuição dos dados. Pode-se notar, durante realização desta pesquisa, um movimento na direção de paradigmas que enfraquecem a necessidade de instituições confiáveis para que integrantes de comunidades possam interagir. Essa nova abordagem aguça a busca por plataformas tecnológicas que permitam aos próprios indivíduos utilizarem mecanismos seguros para compartilhar informações.

Os experimentos realizados apontam bons resultados ao modelo *Dossiê* em todos os cenários avaliados. Em especial, o modelo apresentou-se eficaz no tratamento de agentes maliciosos, à medida que os *feedbacks*/avaliações se tornam imutáveis, e na capacidade de recuperação de eficiência frente as mudanças de comportamento dos agentes *provedores*,

sejam elas *graduais* ou *abruptas*. A flexibilidade desse modelo é resultante das funções de decaimento dos *feedbacks* em função do tempo. Sob ótica de um ambiente descentralizado, é possível concluir, a partir dos experimentos realizados, que o *Dossiê* é uma modelo capaz de auxiliar integrantes de uma comunidade virtual a selecionar bons parceiros para interagir, mitigando o risco das relações entre agentes em um sistema aberto.

6.1 Contribuições

A principal contribuição está na proposta do modelo de confiança *Dossiê*, que apresentou resultados superiores a outros modelos de referência considerados. Ele foi construído sob uma abordagem distribuída e, de um lado, seu desempenho é equivalente ao modelo centralizado, e de outro lado, tem a vantagem de conciliar naturalmente a complexidade da distribuição do controle e dos dados. Além disso, foi proposta uma metodologia para a avaliação de modelos de confiança, algo pouco explorado na literatura da área. Assim, o sistema de avaliação proposto contribui para reduzir a carência apontada nas principais revisões, em particular, a falta de ferramentas que possam avaliar múltiplos aspectos dos modelos de confiança. Em menor relevância, a revisão teórica e bibliográfica realizada sobre os modelos de confiança e principais trabalhos realizados até o momento podem auxiliar novas pesquisas nesse campo.

6.2 Trabalhos Futuros

Como sugestão de continuidade em trabalhos futuros sugere-se aprofundar os experimentos realizados sob ambientes mais complexos. Por exemplo, lidar com situações de *conluio* entre as testemunhas, o uso de outras fontes de informação como as *sociológicas* e o *preconceito*. Durante os experimentos foram selecionados modelos de confiança que representam de modo geral as duas principais abordagens: *direta* e *indireta*. No entanto outros modelos poderão ser avaliados e comparados a partir dos experimentos desse trabalho. A construção de uma base de avaliações históricas, à medida que novos modelos são avaliados, permitirá a simplificação de novos trabalhos, pois não haverá a necessidade de reconstruir modelos uma vez salvos na base. Apesar dos resultados apresentados serem encorajadores, novos trabalhos podem avaliar o modelo *Dossiê* sob outras perspectivas não tratadas no momento, como a taxa de desempenho considerando outras sementes aleatórias, outros ativos do mercado de capitais, outros tipos de mudança de comportamento, além de avaliar os modelos sob outras métricas voltadas a indústria de software como a utilização de memória, processamento, otimização de

armazenamento, tráfego de rede, entre outras. Apesar do método de avaliação dos modelos está conceitualmente desvinculado ao sistema de avaliação, seria importante evoluir a ferramentas construída para uso em outros contextos como, na área de *e-commerce*, prestação de serviços em hotelaria, transporte, transações financeiras, entre outras. Isso poderia tornar o sistema de avaliação mais atrativo para novas pesquisas que busquem avaliar a confiança sob contextos próximos das suas necessidades.

6.3 Publicações

Parte do trabalho apresentado tem sido alvo de outras pesquisas realizadas pelo grupo de *Agentes de Software* do *Programa de Pós-Graduação em Informática* (PPGIA) que produziu as seguintes publicações:

- GRANATYR, Jones; LESSING, Otto Robert; **SILVA, Vanderson Botelho**; SCALABRIN, Edson Emílio; BARTHÈS, Jean-Paul André; ENEMBECK, Fabrício. *Trust and Reputation Models for Multiagent Systems*. **ACM Computing Surveys (CSUR)**, v. 48, n. 2, p. 27, 2015.

- BORGES, André Pinz; **SILVA, Vanderson Botelho**; DORDAL, Osmar Betazzi; ÁVALIA, Bráulio Coelho e SCALABRIN, Edson Emílio. *Safety in Multi-Agent Systems: Reputation based on Dossier*. In: **28th International FLAIRS Conference**. May 18-20. 2015. Hollywood. Florida. USA.

- **SILVA, Vanderson Botelho;** ENEMBRECK, Fabrício; ÁVILA, Bráulio Coelho; DE AZEVEDO, Hilton José Silva; SCALABRIN, Edson Emílio. *Using asymmetric keys in a certified trust model for multiagent systems*. **Expert systems with applications**, v. 38, n. 2, p. 1233-1240, 2011.

Referências

ABDUL-RAHMAN, Alfarez; HAILES, Stephen. Supporting trust in virtual communities. In: **System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on**. IEEE, 2000. p. 9 pp. vol. 1.

ABOULWAFA, Somaya; BAHGAT, Reem. DiReCT: Dirichlet-based Reputation and Credential Trust management. In: **Informatics and Systems (INFOS), 2010 The 7th International Conference on**. IEEE, 2010. p. 1-8.

ANDREONI, James; MILLER, John H. Rational cooperation in the finitely repeated prisoner's dilemma: Experimental evidence. **The economic journal**, p. 570-585, 1993.

ANTONOPOULOS, Andreas M. Mastering Bitcoin: unlocking digital cryptocurrencies." O'Reilly Media, Inc.", 2014.

ARTZ, Donovan; GIL, Yolanda. A survey of trust in computer science and the semantic web. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 5, n. 2, p. 58-71, 2007.

ASHRI, Ronald et al. Trust evaluation through relationship analysis. In: **Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems**. ACM, 2005. p. 1005-1011.

BENTAHAR, Jamal; MEYER, J.-J. Ch; MOULIN, Bernard. Securing agent-oriented systems: An argumentation and reputation-based approach. In: **Information Technology, 2007. ITNG'07. Fourth International Conference on**. IEEE, 2007. p. 507-515.

BEDI, Punam; KAUR, Harmeet; MARWAHA, Sudeep. Trust Based Recommender System for Semantic Web. In: **IJCAI**. 2007. p. 2677-2682.

BERTOCCO, Cristian; FERRARI, Carlo. Context-dependent reputation management for soft security in multi agent systems. In: **Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on**. IEEE, 2008. p. 77-81.

BOND, Alan H.; GASSER, Les (Ed.). **Readings in distributed artificial intelligence**. Morgan Kaufmann, Publishers: San Mateo, CA, 1988.

BOTSMAN, Rachel; ROGERS, Roo. **What's mine is yours: how collaborative consumption is changing the way we live**. London: Collins, 2011.

BRADSHAW, J. M. An Introduction to software Agents. In: **Bradshaw, J. M. (Ed.). Software Agents. Massachussetts**: MIT Press 1997.

BROMLEY, D. B. **Reputation, Image and Impression Management**. John Wiley & Sons, 1993.

BRATMAN, Michael E.; INTENTION, Plans. Practical Reason. **Harvard University**, 1987.

BOTELHO, Vanderson; ENEMBRECK, Fabrício; ÁVILA, Bráulio C; AZEVEDO, Hilton; SCALABRIN, Edson. Encrypted certified trust in multi-agent system. In: **Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on**. IEEE, 2009. p. 227-232.

BOTELHO, Vanderson; ENEMBRECK, Fabrício; ÁVILA, Bráulio; DE AZEVEDO, Hilton; SCALABRIN, Edson. Using asymmetric keys in a certified trust model for multiagent systems. **Expert systems with applications**, v. 38, n. 2, p. 1233-1240, 2011.

BURNETT, Chris; NORMAN, Timothy J.; SYCARA, Katia. Stereotypical trust and bias in dynamic multiagent systems. **ACM Transactions on Intelligent Systems and Technology (TIST)**, v. 4, n. 2, p. 26, 2013.

CAMPILLO-SANCHEZ, Pablo; GÓMEZ-SANZ, Jorge J. Agent Based Simulation for Creating Ambient Assisted Living Solutions. In: **Advances in Practical Applications of Heterogeneous**

Multi-Agent Systems. The PAAMS Collection. Springer International Publishing, 2014. p. 319-322.

CARTER, Jonathan; GHORBANI, Ali A. Value centric trust in multiagent systems. In: **Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on**. IEEE, 2003. p. 3-9.

CASTELFRANCHI, Cristiano; FALCONE, Rino. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In: **Multi Agent Systems, 1998. Proceedings. International Conference** on. IEEE, 1998. p. 72-79.

CASTELFRANCHI, Cristiano; FALCONE, Rino; PEZZULO, Giovanni. Trust in information sources as a source for trust: a fuzzy approach. In: **Proceedings of the second international joint conference on Autonomous agents and multiagent systems**. ACM, 2003. p. 89-96.

CHARNIAK, Eugene. Bayesian networks without tears. **AI magazine**, v. 12, n. 4, p. 50, 1991.

CHAUM, David. Security without identification: Transaction systems to make big brother obsolete. **Communications of the ACM**, v. 28, n. 10, p. 1030-1044, 1985.

CROTTY, James. Structural causes of the global financial crisis: a critical assessment of the 'new financial architecture'. **Cambridge journal of economics**, v. 33, n. 4, p. 563-580, 2009.

DAS, Anupam; ISLAM, Mohammad Mahfuzul. SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems. **Dependable and Secure Computing, IEEE Transactions on**, v. 9, n. 2, p. 261-274, 2012.

DASGUPTA, Partha. Trust as a commodity. **Trust: Making and breaking cooperative relations**, v. 4, p. 49-72, 2000.

DAVIS, Randall; SMITH, Reid G. Negotiation as a metaphor for distributed problem solving. **Artificial intelligence**, v. 20, n. 1, p. 63-109, 1983.

DELLAROCAS, Chrysanthos. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. **Management science**, v. 49, n. 10, p. 1407-1424, 2003.

DERBAS, Ghada et al. Trummar-a trust model for mobile agent systems based on reputation. In: **Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on**. IEEE, 2004. p. 113-120.

DIAS, João; PAIVA, Ana. I want to be your friend: Establishing relations with emotionally intelligent agents. In: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2013. p. 777-784.

DIFFIE, Whitfield; HELLMAN, Martin. New directions in cryptography. **IEEE transactions on Information Theory**, v. 22, n. 6, p. 644-654, 1976.

DONDIO, Pierpaolo; BARRETT, Stephen. Presumptive selection of trust evidence. In: **Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems**. ACM, 2007. p. 166.

DONG-HUYNHA, T.; JENNINGS, N.; SHADBOLT, N. FIRE: An integrated trust and reputation model for open multi-agent systems. In: **ECAI 2004: 16th European Conference on Artificial Intelligence, August 22-27, 2004, Valencia, Spain: including Prestigious Applicants [sic] of Intelligent Systems (PAIS 2004): proceedings**. 2004. p. 18.

DWORK, Cynthia; NAOR, Moni. Pricing via processing or combatting junk mail. In: **Annual International Cryptology Conference**. Springer Berlin Heidelberg, 1992. p. 139-147.

EBAY. 2015. Disponível em: <http://www.ebay.com>. Acesso em: 15 mar. 2015.

ELGOHARY, Nagwa E.; ELFETOUH, Ahmed A.; BARAKAT, Shereif I. Developing a Reputation Model for Electronic Markets. **International Journal of Electrical & Computer Sciences**, v. 10, n. 6, 2010.

FALCONE, Rino; CASTELFRANCHI, Cristiano. Social trust: A cognitive approach. In: **Trust and deception in virtual societies**. Springer Netherlands, 2001. p. 55-90.

FALKENRECK, Christine; WAGNER, Ralf. The impact of perceived innovativeness on maintaining a buyer–seller relationship in health care markets: A cross-cultural study. Journal of Marketing Management, v. 27, n. 3-4, p. 225-242, 2011.

FANG, H.; ZHANG, J.; ŞENSOY, M.; THALMANN, N. M. SARC: subjectivity alignment for reputation computation. In: **Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3**. International Foundation for Autonomous Agents and Multiagent Systems, 2012. p. 1365-1366

FANG, Hui; ZHANG, Jie; THALMANN, Nadia Magnenat. A trust model stemmed from the diffusion theory for opinion evaluation. In: **Proceedings of the 2013 international conference on autonomous agents and multi-agent systems.** International Foundation for Autonomous Agents and Multiagent Systems, 2013. p. 805-812.

FIPA, A. C. L. Fipa acl message structure specification. **Foundation for Intelligent Physical Agents, http://www. fipa. org/specs/fipa00061/SC00061G. html (30.6. 2004)**, 2002.

FRIEDMAN, Milton. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. **Journal of the american statistical association**, v. 32, n. 200, p. 675-701, 1937.

FULLAM, Karem K.; KLOS, Tomas B.; MILLER, Guillaume; SABATER, Jordi; SCHLOSSER, Andreas; TOPOL, Zvi; BARBER, Suzanne, ROSENSCHEING, VERCOUTER, Laurent e VOSS, Marco. The agent reputation and trust (art) testbed game description (version 2.0). **URL http://megatron. iiia. csic. es/art-testbed/pdf/**SpecSummary. pdf, 2006.

GAMA, João; SEBASTIÃO, Raquel; RODRIGUES, Pedro Pereira. Issues in evaluation of stream learning algorithms. In: **Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining**. ACM, 2009. p. 329-338.

GHANEA-HERCOCK, Robert. Dynamic trust formation in multi-agent system. In**: Tenth international workshop on trust in agent societies at the autonomous agents and multi-agent systems conference (AAMAS 2007), Hawaii**. 2007.

QING-HUA, Zhou; CHONG-JUN, Wang; JUN-YUAN, Xie. Core: A trust model for agent coalition formation. In: **Natural Computation, 2009. ICNC'09. Fifth International Conference on**. IEEE, 2009. p. 541-545.

GODO, L.; RAMCHURN, Sarvapali D.; JENNINGS, N. R., SIERRA, C. Devising a trust model for multi-agent interactions using confidence and reputation. **Applied Artificial Intelligence**, v. 18, n. 9-10, p. 833-852, 2004.

GEORGEFF, Michael. Communication and interaction in multi-agent planning. **Readings in distributed artificial intelligence**, v. 313, p. 125-129, 1988.

GRANDISON, Tyrone; SLOMAN, Morris. A survey of trust in internet applications. Communications Surveys & Tutorials, IEEE, v. 3, n. 4, p. 2-16, 2000.

GRIFFITHS, Nathan; LUCK, Michael. Coalition formation through motivation and trust. In: **Proceedings of the second international joint conference on Autonomous agents and multiagent systems**. ACM, 2003. p. 17-24.

GRIFFITHS, Nathan. Task delegation using experience-based multi-dimensional trust. In: **Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems**. ACM, 2005. p. 489-496.

GUINARD, Dominique et al. From the internet of things to the web of things: Resource-oriented architecture and best practices. In: **Architecting the Internet of things**. Springer Berlin Heidelberg, 2011. p. 97-129.

HUYNH, Trung Dong; JENNINGS, Nicholas R.; SHADBOLT, Nigel R. Certified reputation: how an agent can trust a stranger. In: **Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems**. ACM, 2006. p. 1217-1224.

HUYNH, T. Dong; JENNINGS, Nicholas R.; SHADBOLT, N. Developing an integrated trust and reputation model for open multi-agent systems. 2004.

JENNINGS, Nicholas R.; SYCARA, Katia; WOOLDRIDGE, Michael. A roadmap of agent research and development. Autonomous agents and multi-agent systems, v. 1, n. 1, p. 7-38, 1998.

JOSANG, Audun; BEWSELL, Glenn. Guest Editors' Introduction Trust and Trust Management. Journal of theoretical and applied electronic commerce research, v. 5, n. 2, p. 1-2, 2010.

JSANG, Audun; ISMAIL, Roslan. The beta reputation system. In: **Proceedings of the 15th bled electronic commerce conference**. 2002. p. 41-55.

JURCA, Radu; FALTINGS, Boi. An incentive compatible reputation mechanism. In: **E-Commerce, 2003. CEC 2003. IEEE International Conference on**. IEEE, 2003. p. 285-292.

JURCA, Radu; FALTINGS, Boi. Using CHI-scores to reward honest feedback from repeated interactions. In: **Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems**. ACM, 2006. p. 1233-1240.

KALISKI, Burt. PKCS# 7: Cryptographic message syntax version 1.5. 1998.

KATZ, L. “A New Status Index Derived from Sociometric Analysis”, Psychometrika, 18, pp. 39-43, 1953.

KERCKHOFFS, A. La cryptographie militaire. Journal des sciences militaries. **IX (38)**, v. 5., pp. 5–83, Jan. 1883, pp. 161–191, Feb. 1883.

KLEJNOWSKI, Lukas et al. An architecture for trust-adaptive agents. In: **Self-Adaptive and Self-Organizing Systems Workshop (SASOW), 2010 Fourth IEEE International Conference on**. IEEE, 2010. p. 178-183.

KONOLIGE, Kurt. **A First-Order Formalization of Knowledge and Action for a Multiagent Planning System**. SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, 1980.

KOSKO, Bart. Fuzzy cognitive maps. **International journal of man-machine studies**, v. 24, n. 1, p. 65-75, 1986.

KOSTER, Andrew; SABATER-MIR, Jordi; SCHORLEMMER, Marco. Personalizing communication about trust. In: **Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1**. International Foundation for Autonomous Agents and Multiagent Systems, 2012. p. 517-524.

KREPS, David M. et al. Rational cooperation in the finitely repeated prisoners' dilemma. **Journal of Economic theory**, v. 27, n. 2, p. 245-252, 1982.

LI, Li et al. A Quantifiable Trust Model for Multi-agent System Based on Equal Relations. In: **Computational Intelligence and Security, 2007 International Conference on**. IEEE, 2007. p. 291-295.

LIU, Siyuan et al. iCLUB: An integrated clustering-based approach to improve the robustness of reputation systems. In: **The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3**. International Foundation for Autonomous Agents and Multiagent Systems, 2011. p. 1151-1152.

LIU, Siyuan et al. A dempster-shafer theory based witness trustworthiness model to cope with unfair ratings in e-marketplace. In: Proceedings **of the 14th Annual International Conference on Electronic Commerce**. ACM, 2012. p. 99-106.

LIU, Siyuan et al. A fuzzy logic based reputation model against unfair ratings. In: **Proceedings of the 2013 international conference on autonomous agents and multi-agent systems**. International Foundation for Autonomous Agents and Multiagent Systems, 2013. p. 821-828.

LIU, Xin; DATTA, Anwitaman. Modeling context aware dynamic trust using hidden markov model. In: **Twenty-Sixth AAAI Conference on Artificial Intelligence**. 2012.

LU, Gehao et al. A review on computational trust models for multi-agent systems. *The open information science journal*, v. 2, p. 18-25, 2009.

MARSH, Stephen Paul. Formalizing trust as computational concept. Department of Computing Science and Mathematics University of Stirling. 1994.

MAES, Pattie. Artificial life meets entertainment: lifelike autonomous agents. **Communications of the ACM**, v. 38, n. 11, p. 108-114, 1995.

MATT, Paul-Amaury; MORGE, Maxime; TONI, Francesca. Combining statistics and arguments to compute trust. In: **Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1**. International Foundation for Autonomous Agents and Multiagent Systems, 2010. p. 209-216.

MERKLE, Ralph C. A digital signature based on a conventional encryption function. In: **Conference on the Theory and Application of Cryptographic Techniques**. Springer Berlin Heidelberg, 1987. p. 369-378.

MCCARTHY, John. Ascribing mental qualities to machines. 1979.

MEYER, C. Cryptography-A state of the art review. In: **CompEuro'89.,'VLSI and Computer Peripherals. VLSI and Microelectronic Applications in Intelligent Peripherals and their Interconnection Networks', Proceedings**. IEEE, 1989. p. 4/150-4/154.

MOKHTARI, Ehsan et al. A context-aware reputation-based model of trust for open multi-agent environments. In: **Advances in Artificial Intelligence**. Springer Berlin Heidelberg, 2011. p. 301-312.

MOKHTAR, M. R.; WAJID, Usman; WANG, W. Collaborative trust in multi-agent system. In: **Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. WETICE 2007. 16th IEEE International Workshops on**. IEEE, 2007. p. 30-34.

MOLINA, Martin; CARRASCO, Sergio; MARTIN, Jorge. Agent-Based Modeling and Simulation for the Design of the Future European Air Traffic Management System: The Experience of CASSIOPEIA. **In: Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection**. Springer International Publishing, 2014. p. 22-33.

MONTANER, Miquel; LÓPEZ, Beatriz; DE LA ROSA, JOSEP, Luís. Developing trust in recommender agents. In: Proceedings **of the first international joint conference on Autonomous agents and multiagent systems: part 1**. ACM, 2002. p. 304-305.

MUÑOZ, V.; MURILLO, J.; LÓPEZ, B.; BUSQUETS, D. Strategies for exploiting trust models in competitive multi-agent systems. In: **Multiagent System Technologies**. Springer Berlin Heidelberg, 2009. p. 79-90.

NEMENYI, P. B. **Distribution-Free Multiple Comparisons, Princeton University NJ, 1963**. Tese de Doutorado. Ph. D. thesis.

NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

NEVILLE, Brendan; PITT, Jeremy. A computational framework for social agents in agent mediated e-commerce. In: **Engineering Societies in the Agents World IV**. Springer Berlin Heidelberg, 2004. p. 376-391.

NEWMAN, Sam. **Building Microservices**. " O'Reilly Media, Inc.", 2015.

PADOVAN, Boris; SACKMANN, Stefan; TORSTEN EYMANN, Ingo Pippow. A prototype for an agent-based secure electronic marketplace including reputation-tracking mechanisms. **International Journal of Electronic Commerce**, v. 6, n. 4, p. 93-113, 2002.

PALANCA, Javier et al. receteame.com: A Persuasive Social Recommendation System. In: **Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection**. Springer International Publishing, 2014. p. 367-370.

PARSONS, Simon et al. Argumentation-based reasoning in agents with varying degrees of trust. In: **The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2**. International Foundation for Autonomous Agents and Multiagent Systems, 2011. p. 879-886.

PINYOL, Isaac; SABATER-MIR, Jordi. Computational trust and reputation models for open multi-agent systems: a review. Artificial Intelligence Review, v. 40, n. 1, p. 1-25, 2013.

PIUNTI, Michele et al. Multimodal Trust Formation with Uninformed Cognitive Maps (UnCM). In: **Proceedings of the 11th International Conference on Autonomous Agents and Multia-**

gent Systems-Volume 3. International Foundation for Autonomous Agents and Multiagent Systems, 2012. p. 1241-1242.

POLLOCK, G. B.; DUGATKIN, L. A. "Reciprocity and the Evolution of Reputation" Journal of Theoretical Biology, 159, pp. 25-37, 1992.

RAMCHURN, Sarvapali D.; HUYNH, Dong; JENNINGS, Nicholas R. Trust in multi-agent systems. **The Knowledge Engineering Review**, v. 19, n. 01, p. 1-25, 2004.

REGAN, Kevin; POUPART, Pascal; COHEN, Robin. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In: **Proceedings of the National Conference on Artificial Intelligence**. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006. p. 1206.

REHAK, Martin *et al*. Trust model for open ubiquitous agent systems. In: **Intelligent Agent Technology, IEEE/WIC/ACM International Conference on**. IEEE, 2005. p. 536-542.

RESNICK, Paul et al. Reputation systems. Communications of the ACM, v. 43, n. 12, p. 45-48, 2000.

RETTINGER, Achim; NICKLES, Matthias; TRESP, Volker. Learning initial trust among interacting agents. In: **Cooperative Information Agents XI**. Springer Berlin Heidelberg, 2007. p. 313-327.

RETTINGER, Achim; NICKLES, Matthias; TRESP, Volker. A statistical relational model for trust learning. In: **Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2**. International Foundation for Autonomous Agents and Multiagent Systems, 2008. p. 763-770.

ROSACI, Domenico; SARNÉ, Giuseppe ML; GARRUZZO, Salvatore. Integrating trust measures in multiagent systems. International Journal of Intelligent Systems, v. 27, n. 1, p. 1-15, 2012.

SABATER, Jordi; SIERRA, Carles. Reputation and social network analysis in multi-agent systems. In: **Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1**. ACM, 2002. p. 475-482.

SABATER, Jordi; SIERRA, Carles. Review on computational trust and reputation models. Artificial intelligence review, v. 24, n. 1, p. 33-60, 2005.

SCHILLO, Michael; FUNK, Petra; ROVATSOS, Michael. Using trust for detecting deceitful agents in artificial societies. **Applied Artificial Intelligence**, v. 14, n. 8, p. 825-848, 2000.

SEN, Sandip; SAJJA, Neelima. Robustness of reputation-based trust: Boolean case. In: **Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1. ACM**, 2002. p. 288-293.

SERRANO, Emilio; ROVATSOS, Michael; BOTIA, Juan. A qualitative reputation system for multiagent systems with protocol-based communication. In: **Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1**. International Foundation for Autonomous Agents and Multiagent Systems, 2012. p. 307-314.

SHAFER, Glenn. **A mathematical theory of evidence**. Princeton: Princeton university press, 1976.

SIERRA, Carles; DEBENHAM, John. An information-based model for trust. In: **Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems**. ACM, 2005. p. 497-504.

SINGH, Munindar P. Trust as dependence: a logical approach. In: **The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2**. International Foundation for Autonomous Agents and Multiagent Systems, 2011. p. 863-870.

SLOMAN, M., Trust-management in Internet and pervasive systems, IEEE Intelligent Systems, vol. 19, no. 5, pp. 77-79, Sept/Oct 2004.

SHMEIL, M. A. H., Sistemas Multiagente na Modelação da Estrutura e Relações de Contratação de Organizações, Tese de Doutorado, Faculdade de Engenharia, Universidade do Porto, Portugal, 1999.

SONG, Weihua; PHOHA, Vir V.; XU, Xin. An adaptive recommendation trust model in multiagent system. In: **Intelligent Agent Technology, 2004. (IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on**. IEEE, 2004. p. 462-465.

SRIDHARAN, N. S. 1986 Workshop on AI Distributed. **AI magazine**, v. 8, n. 3, 1987.

SUTCLIFFE, Alistair; WANG, Di. Computational modelling of trust and social relationships. **Journal of Artificial Societies and Social Simulation**, v. 15, n. 1, p. 3, 2012.

SWAN, Melanie. **Blockchain: Blueprint for a new economy**. " O'Reilly Media, Inc.", 2015.

TEACY, W. T. *et al*. Sequential decision making with untrustworthy service providers. In: **Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2**. International Foundation for Autonomous Agents and Multiagent Systems, 2008. p. 755-762.

TEACY, WT Luke *et al*. Travos: Trust and reputation in the context of inaccurate information sources. **Autonomous Agents and Multi-Agent Systems**, v. 12, n. 2, p. 183-198, 2006.

TEACY, W. T. *et al*. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. **Artificial Intelligence**, v. 193, p. 149-185, 2012.

TIROLE, Jean. Hierarchies and bureaucracies: On the role of collusion in organizations. **Journal of Law, Economics, & Organization**, v. 2, n. 2, p. 181-214, 1986.

TONG, Xiangrong; HUANG, Houkuan; ZHANG, Wei. Agent long-term coalition credit. **Expert systems with applications**, v. 36, n. 5, p. 9457-9465, 2009.

TRAN, Thomas; COHEN, Robin. Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. In: **Proceedings of the Third In-**

ternational Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2. IEEE Computer Society, 2004. p. 828-835.

VENANZI, Matteo et al. Facing openness with socio-cognitive trust and categories. In: **Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One**. AAAI Press, 2011. p. 400-405.

VOGIATZIS, George; MACGILLIVRAY, Ian; CHLI, Maria. A probabilistic model for trust and reputation. In: **Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1**. International Foundation for Autonomous Agents and Multiagent Systems, 2010. p. 225-232.

WANG, Ping; ZHANG, Zili. A computation trust model with trust network in multi-agent systems. In: **Active Media Technology, 2005. (AMT 2005). Proceedings of the 2005 International Conference on**. IEEE, 2005. p. 389-392.

WANG, Y.; LI, M.; DILLON, E.; CUI, L. G.; HU, J. J.; LIAO, L. J. A context-aware computational trust model for multi-agent systems. In: **Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on**. IEEE, 2008. p. 1119-1124.

WEISS, Gerhard. **Multiagent systems: a modern approach to distributed artificial intelligence**. MIT press, 1999.

WOOD, Gavin. Ethereum: A secure decentralised generalised transaction ledger. **Ethereum Project Yellow Paper**, v. 151, 2014.

WOOLDRIDGE, Michael; JENNINGS, Nicholas R. Intelligent agents: Theory and practice. **The knowledge engineering review**, v. 10, n. 02, p. 115-152, 1995.

YOU, Liangjun. **An adaptive reputation-based trust model for intelligent agents in e-marketplace**. ProQuest, 2007.

YU, Bin; SINGH, Munindar P. Detecting deception in reputation management. In: **Proceedings of the second international joint conference on Autonomous agents and multiagent systems**. ACM, 2003. p. 73-80.

YU, Han et al. A survey of multi-agent trust management systems. **Access, IEEE**, v. 1, p. 35-50, 2013.

ZHANG, Jie; GHORBANI, Ali A.; COHEN, Robin. A familiarity-based trust model for effective selection of sellers in multiagent e-commerce systems. **International Journal of Information Security**, v. 6, n. 5, p. 333-344, 2007.

ZHENG, Xiaoqing et al. Developing a composite trust model for multi-agent systems. In: **Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems**. ACM, 2006. p. 1257-1259.