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

FABIANA LORENZI

Uma abordagem multiagente de recomendação baseada em suposições e confiança para cenários dinâmicos

Tese apresentada como requisito parcial
para a obtenção do grau de
Doutor em Ciência da Computação

Profa. Dra. Ana L. C. Bazzan
Orientador

Profa. Dra. Mara Abel
Co-orientador

Porto Alegre, novembro de 2010

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Lorenzi, Fabiana

Uma abordagem multiagente de recomendação baseada em suposições e confiança para cenários dinâmicos / Fabiana Lorenzi. – Porto Alegre: PPGC da UFRGS, 2010.

84 f.: il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2010. Orientador: Ana L. C. Bazzan; Co-orientador: Mara Abel.

1. Sistemas Multiagente. 2. Sistemas de Recomendação Multiagente. 3. Suposições. 4. Confiança. 5. Sistemas de Manutenção de Verdade. 6. Raciocínio Baseado em Casos. I. Bazzan, Ana L. C.. II. Abel, Mara. III. Título.

AGRADECIMENTOS

> "Há pessoas que nos falam e nem as escutamos, há pessoas que nos ferem e nem cicatrizes deixam, mas há pessoas que simplesmente aparecem em nossas vidas e nos marcam para sempre."
> *Cecília Meireles*

Dedico esta tese a meu marido André, por todo seu carinho e incentivo durante esta jornada, e aos meus pais e meu irmão, por todo apoio, dedicação e carinho de sempre.

Agradeço as minhas orientadoras: Ana Bazzan e Mara Abel. Obrigada por todos sábios conselhos ao longo da minha caminhada, inspiração, incentivo e paciência. São pesquisadoras como vocês que me fazem amar minha área e minha profissão. Obrigada por tudo, especialmente por terem sido orientadoras quando preciso, mas amigas nas horas mais importantes.

Agradeço também ao professor Francesco Ricci, que em 2004 me deu a oportunidade de trabalhar no e-CTRL (e-Commerce Tourism Research Laboratory) em Trento/Itália, no projeto de sistemas de recomendação baseados em casos aplicados ao turismo. Sem sombra de dúvida este estágio foi fundamental para meu amadurecimento como pesquisadora e me trouxe muitos resultados positivos.

Meu muito obrigada também aos meus colegas do grupo MASLAB: Denise, Daniela, Paulo, Fernando, Samuel, Fábio, Michael, Daniel, que nunca cansaram de assistir aos meus seminários e exemplos de turismo. Obrigada a cada um de vocês que de uma forma ou outra contribuiram no meu trabalho.

Um especial agradecimento também aos amigos da Ponto do Turismo, que foram extremamente compreensíveis comigo durante o período do doutorado e gentilmente cederam os dados e os especialistas para auxiliarem na validação da abordagem.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 6

LISTA DE SíMBOLOS 7

LISTA DE FIGURAS 8

LISTA DE TABELAS 9

LISTA DE ALGORITMOS 10

RESUMO 11

ABSTRACT 12

1 INTRODUÇÃO 13

2 REFERENCIAL TEÓRICO 18
2.1 Sistemas de Recomendação 18
2.2 Sistemas de Recomendação Multiagente 21
2.2.1 FAB 22
2.2.2 SmartClients 23
2.2.3 HeraclesII 23
2.2.4 MAPWEB 23
2.2.5 CASIS 24
2.2.6 Marketplace 24
2.3 Sistemas de Manutenção de Verdade 25
2.4 Confiança 31
2.5 Conclusões 32

3 ABORDAGEM MATRES 35
3.1 Tarefas 36
3.2 Modelo do usuário 37
3.3 Modelo do agente 38
3.3.1 Componente de manutenção da verdade: suposições 39
3.3.2 Confiança e especialização 42
3.3.3 Base de casos 45
3.3.4 Componente de manutenção de verdade: manutenção das bases 47
3.4 Resolução de tarefas 50
3.4.1 Busca da informação: localKBSearch 51
3.4.2 Busca da informação: communitySearch 53

3.4.3 Busca da informação: searchWeb 53
3.5 Conclusões 54

4 APLICAÇÃO E VALIDAÇÃO DA ABORDAGEM NO DOMíNIO DO TURISMO 56
4.1 Introdução ao domínio do turismo 56
4.2 Definições no cenário do turismo 57
4.2.1 Modelo do usuário 57
4.2.2 Suposições 58
4.2.3 Especialização e confiança 60
4.2.4 Base de conhecimento 61
4.3 Validação e experimentos 62
4.3.1 Aquisição de conhecimento 63
4.3.2 Avaliações das recomendações 64
4.4 Calibragem do número de agentes 65
4.5 Resultados obtidos 67
4.5.1 Experimento 1: suposições 67
4.5.2 Experimento 2: troca de informações e confiança 69
4.5.3 Experimento 3: comunicação 71
4.6 Avaliação dos resultados 73

5 CONCLUSÕES 75
5.1 Contribuições 75
5.2 Trabalhos Futuros 76

6 PUBLICAÇÕES RELACIONADAS A TESE 78

REFERÊNCIAS 80

LISTA DE ABREVIATURAS E SIGLAS

ATMS	*Assumptions-Based Truth Maintenance System*
CASIS	*Case-based Swarming Intelligence Recommender System*
CMV	Componente de Manutenção de Verdade
DTMS	*Distributed Truth Maintenance System*
JADE	*Java Agent Development Framework*
JTMS	*Justification Based Truth Maintenance System*
RBC	Raciocínio Baseado em Casos
SMA	Sistemas Multiagente
SMV	Sistemas de Manutenção de Verdade
SMVD	Sistemas de Manutenção de Verdade Distribuído
SMVJ	Sistemas de Manutenção de Verdade Baseados em Justificativas
SMVM	Sistemas de Manutenção de Verdade Multiagente
SMVS	Sistemas de Manutenção de Verdade Baseados em Suposições
SR	Sistemas de Recomendação
SRBC	Sistemas de Recomendação Baseado em Casos
SRMA	Sistemas de Recomendação Baseado na Arquitetura Multiagente

LISTA DE SÍMBOLOS

a_n	é o agente;
C	é a comunidade de agentes;
E_{t_n}	é o modelo de avaliação para o tipo de tarefa t_n;
η	é o tempo no qual uma nova avaliação é considerada;
γ	é a informação conflitante detectada nas bases de conhecimento dos agentes;
i_m	é o atributo escolhido pelo usuário;
$\mathcal{K}$	é o conjunto de tipos de tarefas;
KB_a	é a base de conhecimento do agente a;
L	é a lista de tarefas disponíveis para os agentes;
$\mathcal{O}_{i_j}$	é o conjunto de possíveis opções
$P(t_n)$	é o conjunto de predecessores do tipo de tarefa t_n;
q	é a consulta do usuário;
r_m	é o item recomendado pelo agente para o atributo i_m definido;
$\mathcal{S}_{i_m}$	é o conjunto de suposições possível para i_m;
t_n	é o tipo de tarefa;
τ	é o peso do valor $v_x^{(t)}$ na atualização do grau de confiança do agente;
$\mathcal{U}$	é o modelo do usuário;
$v_x^{(t_n)}$	é a avaliação do usuário do tipo de tarefa t_n no tempo x;
φ	é o limiar de similaridade aceitável na aplicação do *Método 2* de geração de suposições.

LISTA DE FIGURAS

Figura 2.1: Ciclo de funcionamento de SRBC. 20
Figura 2.2: Arquitetura de um SMV. 26
Figura 2.3: Exemplo do JTMS. 27
Figura 2.4: Exemplo do ATMS. 28
Figura 2.5: Exemplo de treliça de ambientes. 28
Figura 2.6: Dados compartilhados e particulares. 29

Figura 3.1: Processo da abordagem multiagente de recomendação. 35
Figura 3.2: Preferências do usuário e geração da lista de tarefas. 37
Figura 3.3: Modelo do *AgentWorker*. 39
Figura 3.4: Informações do *AgentWorker* 47
Figura 3.5: Processos do agente para resolução das tarefas. 50
Figura 3.6: Comunicação entre o agente a_i e os agentes de sua confiança. 54

Figura 4.1: Tela principal do sistema 57
Figura 4.2: Exemplo da lista de tarefas L 58
Figura 4.3: Arquivo XML de configuração 60
Figura 4.4: Exemplo de caso - a_4, tarefa de *atração* 61
Figura 4.5: Exemplo de caso - a_1, tarefa de *voo* 62
Figura 4.6: Exemplo de caso - a_3, tarefa de *hotel* 63
Figura 4.7: Valores médios das avaliações de cada tipo de tarefa: voo, hotel e atração, em cada método de recomendação utilizado (ESP, MMP e MSIM). 70
Figura 4.8: Valores médios das avaliações de cada tipo de tarefa: voo, hotel e atração, em cada método de recomendação utilizado (ESP, NTrust e WTrust). 71

LISTA DE TABELAS

Tabela 2.1: Comparação dos sistemas multiagente desenvolvidos para o comércio eletrônico 33
Tabela 2.2: Comparação dos sistemas de manutenção da verdade 33

Tabela 3.1: Exemplo que mostra o decréscimo da importância da avaliação com o passar do tempo ($d = 50$ dias, τ=0.088) 45
Tabela 3.2: Exemplo de um caso na recomendação de filmes - a_1 47

Tabela 4.1: Exemplos de casos adquiridos da agência de viagem 64
Tabela 4.2: Médias obtidas na calibragem do número de agentes 66
Tabela 4.3: Comparação dos resultados obtidos: média, desvio padrão e erro padrão das avaliações obtidas em cada método. 69
Tabela 4.4: Comparação dos resultados obtidos: média, desvio padrão e erro padrão das avaliações obtidas em cada método (ESP, NTrust e WTrust). 72
Tabela 4.5: Média do número de agentes envolvidos na resolução de cada tarefa nos métodos *NTrust* e *WTrust* 73
Tabela 4.6: Média do número de agentes envolvidos na resolução de cada tarefa nos métodos *NTrust* e *WTrust* - após ajuste da atualização dos graus de confiança. 73

LISTA DE ALGORITMOS

1 MMP: Opção mais popular na comunidade de agentes 41
2 MSIM: Casos similares 42
3 MMPU: Opção mais popular no histórico do usuário 43
4 Componente de manutenção das bases de conhecimento dos agentes 49
5 Resolvendo tarefas 51
6 *LocalKBSearch* 53

RESUMO

A falta de informação e de confiança entre os agentes em sistemas de recomendação que lidam com domínios dinâmicos podem ser fatores que contribuem para que os agentes gerem resultados de baixa qualidade. Na falta de informação para gerar recomendações, é necessário que os agentes sejam capazes de assumir ou compartilhar informações, criem laços de confiança entre si e que se adaptem às mudanças do estado do conhecimento para que sejam capazes de resolver os problemas. Esta tese apresenta a abordagem MATRES - uma abordagem multiagente baseada em suposições com mecanismo de confiança aplicada em um sistema de recomendação multiagente. Na abordagem MATRES, os agentes são capazes de lidar com conhecimento distribuído. Cada agente trabalha como especialista e é capaz de compartilhar seu conhecimento com os demais, de acordo com seus índices de confiança. Para a solução de um problema, diferentes tarefas são distribuídas entre os agentes. Algumas tarefas apresentam uma relação de dependência, fazendo com que uma tarefa dependa do resultado de outra. Nesta situação, o agente possui um componente de manutenção da verdade que permite a utilização de suposições para a realização das tarefas de forma assíncrona. Na falta de informação proveniente de outra tarefa, o agente é capaz de manipular suposições, sendo capaz de executar sua tarefa. Além disto, o componente de manutenção da verdade auxilia na manutenção da integridade das bases de conhecimento dos agentes. A abordagem MATRES foi validada em um cenário de recomendação de pacotes turísticos. Casos reais de uma agência de viagem foram utilizados na validação da abordagem e os resultados obtidos corroboram a hipótese de que que a abordagem proposta aumenta a assertividade das recomendações geradas pelos agentes em ambientes distribuídos e dinâmicos.

Palavras-chave: Sistemas Multiagente, Sistemas de Recomendação Multiagente, Suposições, Confiança, Sistemas de Manutenção de Verdade, Raciocínio Baseado em Casos.

A Multiagent Recommender Approach Based in Assumptions and Trust for Dynamic Scenarios

ABSTRACT

The lack of trust and information among agents in dynamic domains may contribute to the generation of poor results in multiagent recommender systems. These domains requires that agents exchange information, establishing bonds of trust among themselves and adapting the modification of the status of the knowledge to be able to solve problems. In systems where the knowledge is distributed among several agents, the exchange of information is essential for improving the performance of the agents and maybe leading to inconsistencies when the information exchanged has different status. This thesis presents the MATRES approach - a multiagent Assumption-Based recommender approach with a trust mechanism. In this approach agents are able to deal with distributed knowledge. Each agent works as an expert and is able to share its knowledge with other agents, according to its trust degree. In order to solve a problem, different tasks are distributed among the agents. Some tasks are interdependent, which means that to solve a task it is necessary to use the result from other one. In this situation, the agent has a truth maintenance component that allows using assumptions to perform tasks in a assynchronous ways and helps the maintenance of the integrity of the knowledge bases of the agents. The MATRES approach was validated in the travel recommendation scenario. The results show that the proposal increases the assertiveness of the recommendations provided by the agents in this dynamic domain.

Keywords: Multiagent Systems, Assumptions, Trust, Truth Maintenance System, Case-Based Reasoning.

1 INTRODUÇÃO

Nos dias atuais, cada vez mais as pessoas têm utilizado a internet em busca de informações, comparando produtos, preços e serviços oferecidos pelas empresas. O aumento da quantidade de informações disponíveis ocasiona uma sobrecarga de dados, tornando-se um problema para o usuário, que busca encontrar informações relevantes, úteis e específicas para resolver o seu problema. Sistemas de recomendação foram criados para lidar com o problema de sobrecarga de informação (RESNICK et al., 1994). Estes sistemas têm sido utilizados para sugerir produtos ou fornecer informações que ajudem o cliente a decidir sobre a compra em diversos domínios, tais como, livros, filmes ou viagens (SCHAFER; KONSTAN; RIEDL, 2001) (SCHMIDT-THIEME; FELFERNIG; FRIEDRICH, 2007).

A recomendação pode ser obtida através da utilização de diferentes técnicas, como, por exemplo, filtragem colaborativa, filtragem baseada em conteúdo ou filtragem baseada em conhecimento. O algoritmo de filtragem colaborativa (HERLOCKER et al., 2004) (SYMEONIDIS et al., 2008) utiliza as preferências do usuário para compará-lo com outros usuários que tenham gostos similares. Por exemplo, o *usuário X* define suas preferências e, neste momento, o sistema busca usuários que tenham os mesmos gostos. Considerando que o sistema tenha encontrado o *usuário Y*, no próximo passo, o sistema verifica os itens já recomendados para o *usuário Y* e os recomenda para o *usuário X*. A grande desvantagem desta técnica é a necessidade das avaliações de outros usuários, o que leva ao problema de *Cold Start*, ou seja, sem avaliações de outros usuários o sistema não tem como comparar com quem o usuário atual é parecido e gerar uma nova recomendação para ele.

Na técnica de filtragem baseada em conteúdo (TORRES et al., 2004), o sistema analisa os itens já comprados pelo usuário e com base nestes itens, recomenda outros. O maior problema encontrado nesta técnica é a exigência de uma lista inicial de compras feitas pelo usuário. Se for a primeira vez que o usuário utiliza o sistema, ou seja, se a lista de compras do usuário estiver vazia, não é possível recomendar próximos itens.

No intuito de minimizar os pontos fracos das técnicas anteriores, surgiu a recomendação baseada em conhecimento (GUNAWARDANA; MEEK, 2009). Esta abordagem é considerada por muitos autores como híbrida (BURKE, 2007; BERKOVSKY; KUFLIK; RICCI, 2008) e as recomendações são geradas a partir de conhecimento sobre o usuário ou sobre os produtos. Um sistema de recomendação baseado em conhecimento tem a capacidade de perceber como um determinado item se encaixa com a necessidade do usuário. Através disto ele consegue raciocinar sobre a relação entre a necessidade do usuário e uma possível recomendação.

Sistemas de recomendação baseados em casos (LORENZI; RICCI, 2005) são exemplos da utilização de conhecimento no processo de recomendação. Raciocínio baseado em casos (RBC) (KOLODNER, 1993) é uma abordagem aplicada na resolução de problemas

com base nas experiências. Novos problemas são resolvidos com base nas experiências armazenadas em problemas resolvidos anteriormente. Uma característica importante deste processo é a habilidade de representar e de processar conhecimento sobre a similaridade entre problemas e também de como adaptar as soluções anteriores para novos problemas. Por este motivo, a abordagem de RBC tem sido explorada em sistemas de recomendação baseados em conhecimento (RICCI; VENTURINI, 2008) (SMYTH, 2007).

Apesar da eficiência destes sistemas em auxiliar o usuário nas recomendações, um sistema de recomendação centralizado não é capaz de tratar domínios dinâmicos e distribuídos, pois o conhecimento pode estar distribuído em diferentes fontes e sofrem alterações com muita frequência. No domínio do turismo, por exemplo, o conhecimento necessário para a recomendação pode estar distribuído em diferentes operadoras, cias aéreas, hotéis, etc. Em vista disto, surgiram os sistemas de recomendação multiagente que são compostos por um conjunto de agentes autônomos que têm um papel fundamental na busca da recomendação. Normalmente, os agentes têm o objetivo de representar os usuários, buscando recomendar os melhores produtos e ofertas (ZHANG et al., 2008). Os agentes têm o objetivo de automatizar as etapas de maior consumo de tempo do processo de compra na internet. Um agente no comércio eletrônico age de maneira flexível em nome de um usuário, tentando atingir objetivos particulares (ZAMBONELLI et al., 2001).

Um sistema de recomendação multiagente deve ser capaz de coletar, filtrar, avaliar e utilizar informações disponíveis na Internet em um processo de recomendação. Um conjunto de agentes é responsável pelo processo de recomendação (LORENZI; SANTOS; BAZZAN, 2005) e cada agente é responsável por uma parte da recomendação. O conjunto destas recomendações parciais torna-se a recomendação final que será apresentada ao usuário.

O primeiro sistema de recomendação multiagente desenvolvido foi o FAB (BALABANOVIC; SHOHAM, 1997). Este sistema recomenda páginas *web* aos usuários através da combinação das técnicas de filtragem colaborativa e filtragem baseada em conteúdo. Dois tipos de agentes foram implementados para que o sistema chegasse a uma recomendação: um agente coletor, responsável pela coleta de documentos e um agente de seleção, responsável pela seleção de páginas a serem recomendadas a um usuário.

Este sistema teve grande impacto na área de sistemas de recomendação multiagente, pois ele propôs a combinação das duas principais técnicas de recomendação (filtragem colaborativa e filtragem baseada em conteúdo) em um ambiente multiagente. Esta união resultou em pontos fortes do sistema, tais como: a) através da utilização das recomendações colaborativas, as experiências de outros usuários são levadas em consideração; b) através da utilização das recomendações baseadas em conteúdo, é possível lidar com itens ainda não vistos por outros usuários; e c) é possível recomendar bons itens a um usuário mesmo que não exista usuários semelhantes a ele. Entretanto, o sistema possui um ponto fraco que é a utilização de um agente coletor. Esta abordagem não seria muito eficiente em um domínio dinâmico, onde as informações sofrem constantes mudanças.

Outro exemplo de sistema de recomendação multiagente foi apresentado em (WEI et al., 2008). O objetivo do sistema é a recomendação de *websites* e para isso o sistema tenta utilizar o método mais adequado de recomendação (filtragem colaborativa ou filtragem baseada em conteúdo) de acordo com a situação apresentada no momento. Como o processo de mostrar uma *URL* na barra de endereços pode ser considerado um problema de alocação de recursos, o sistema aplica técnicas de mercado livre para vender os recursos (os *links*). Cada agente representa um *link* e todos agentes participam de leilões onde se conhecem os ganhadores da recomendação. Os agentes que tiveram seus *links* reco-

mendados são recompensados. Os agentes que se tornam mais recompensados ao longo do tempo têm seus *links* cada vez mais recomendados. Entretanto, esta característica de recomendar sempre os *links* dos agentes mais recompensados pode ser vista como uma desvantagem desta abordagem, pois ela não permite que o sistema surpreenda o usuário com *links* diferentes.

A maioria dos sistemas de recomendação multiagente encontrados na literatura utiliza agentes coletores. Agentes coletores trabalham *offline* na busca por informações e as disponibilizam em uma base central que pode ser acessada por outros agentes ao longo do processo de recomendação. BIG (LESSER et al., 2000) é o exemplo de agente coletor mais importante na área. Este agente auxilia no processo de apoio a decisão, extraindo informações de documentos estruturados e não estruturados e utilizando esta informação para refinar suas atividades de busca e processamento.

Apesar de eficientes, as abordagens multiagente apresentam algumas limitações, como por exemplo:

- em problemas mais complexos, o agente que representa o usuário necessitaria de muito conhecimento para atender as solicitações do usuário;
- em alguns domínios é necessário que os agentes tenham conhecimento especialista para compor uma recomendação;
- um único agente pode não possuir o conhecimento suficiente para gerar uma recomendação;
- agentes coletores podem manter informações desatualizadas, pois eles coletam informações que sofrem atualizações frequentes. Como os agentes coletores não buscam informações conforme a demanda do processo de recomendação, a informação coletada por eles pode estar desatualizada, levando à geração de recomendações incorretas;

Esta tese apresenta uma nova abordagem mutiagente de recomendação baseada em componentes de geração de suposições e confiança para tratar as limitações existentes nas abordagens atuais. Suposições podem ser assumidas pelos agentes no momento da geração da recomendação com o objetivo de fornecer uma resposta satisfatória para o usuário mesmo na falta de informações necessárias para a prover a recomendação. O primeiro sistema baseado em suposições foi apresentado por (DEKLEER, 1986), que previa a utilização de suposições para sustentar as proposições de um determinado agente. Para que uma proposição fosse considerada verdadeira era necessário um conjunto de suposições que validasse esta proposição. O conjunto de suposições era gerado a partir do conhecimento de domínio na forma de regras.

Além da utilização das suposições, os agentes podem colaborar para resolver alguma recomendação e, para que esta colaboração seja eficiente, é importante que exista confiança entre os agentes. Um agente pode confiar em outro à medida que tem boas experiências com este agente. Esta confiança também é utilizada como forma de melhorar o desempenho do sistema de recomendação, evitando um tempo alto de resposta dos agentes.

A abordagem proposta nesta tese tem como objetivo principal melhorar a qualidade das recomendações feitas pelos agentes em domínios dinâmicos que necessitam de conhecimento especialista. Melhorar a qualidade da recomendação dos agentes significa apresentar recomendações que agradem o usuário, apesar das dificuldades que podem

ser encontradas ao longo do processo de recomendação. A abordagem é composta por agentes que possuem características importantes para o auxilio na busca de melhores recomendações. Os agentes trabalham de forma cooperativa, resolvendo suas tarefas de forma assíncrona, especializando-se ao longo de sua existência, utilizando suposições durante o processo de recomendação e adquirindo confiança entre si com o objetivo de trocar informações quando necessário.

A principal hipótese abordada por este trabalho é:

Uma abordagem multiagente de recomendação baseada em suposições e com um método de confiança aplicada em domínios dinâmicos contribui para a geração de recomendações mais assertivas.

Como premissas para o desenvolvimento da abordagem, assume-se que:

- As fontes de informação necessárias para a recomendação estão distribuídas;
- As informações do domínio são dinâmicas e atualizadas frequentemente;
- As recomendações a serem resolvidas são, por natureza, complexas, sendo necessário sua decomposição em subtarefas interdependentes;
- É necessário conhecimento especialista para filtrar a informação relevante para a recomendação;
- Um agente conhece os demais agentes da comunidade;
- Os agentes são cooperativos;
- O comportamento dos agentes não se altera ao longo de sua existência.

Como contribuição principal deste trabalho, espera-se:

- Permitir que, na falta de informações no processo de recomendação, os agentes possam ser capazes de utilizar suposições e compartilhar informações entre eles, garantindo a qualidade do processo.

Este trabalho contribui para avanços relativos ao estado da arte no estudo e desenvolvimento de sistemas de recomendação multiagente, entre os quais pode-se destacar:

- A especialização dos agentes ao longo dos processos de recomendação. Através do componente de confiança os agentes se tornam especialistas em determinados tipos de tarefas. Este comportamento contribui para a melhora da assertividade das recomendações do agente, pois quanto maior o número de recomendações de um determinado tipo ele resolver, maior será a probabilidade de continuar resolvendo este mesmo tipo com bons resultados;
- Diminuição do número de troca de informações entre os agentes durante o processo de recomendação. Os agentes podem colaborar ao longo das recomendações e para isto eles se comunicam e trocam informações. Através do componente de confiança, os agentes podem se comunicar somente com agentes que consideram confiáveis. Isto contribui para o desempenho dos agentes, pois diminui o número de agentes com quem cada agente se comunica para buscar informações;

- Permitir que os agentes mantenham a consistência de suas bases de conhecimento. Quando agentes compartilham informações em ambientes dinâmicos, é comum a existência de inconsistências entre as informações compartilhadas em algum momento. Entretanto, estas inconsistências podem levar os agentes a gerarem recomendações incorretas. O componente de manutenção da verdade contribui para que seja mantida a consistência do conhecimento compartilhado entre os agentes, evitando a geração de recomendações incorretas.

Na abordagem desenvolvida, cada agente possui sua base de conhecimento na qual são armazenadas as recomendações já resolvidas. A cada nova recomendação, os agentes incrementam suas bases de conhecimento e se tornam especialistas em um determinado tipo de tarefa.

Existe um mecanismo de manutenção da verdade distribuído com utilização de suposições que permite que os agentes trabalhem de forma assíncrona no processo de recomendação e realiza a manutenção das bases de conhecimento dos agentes. As trocas de informações entre os agentes são realizadas de acordo com o grau de confiança entre eles.

Todos os mecanismos da abordagem proposta foram validados através de experimentos. Os resultados obtidos nestes experimentos mostraram que a abordagem proposta, baseada em suposições e confiança, pode ser aplicada com sucesso em domínios distribuídos e dinâmicos de recomendação.

Este trabalho está organizado da seguinte forma:

Capítulo 2 Este capítulo apresenta o referencial teórico necessário para compreensão da abordagem desenvolvida e discute os trabalhos relacionados à abordagem proposta;

Capítulo 3 Neste capítulo a abordagem proposta é apresentada e detalhada;

Capítulo 4 Este capítulo apresenta a validação da abordagem proposta no domínio do turismo. Os experimentos realizados neste cenário são apresentados e os resultados obtidos são discutidos;

Capítulo 5 Neste capítulo são apresentadas as conclusões e pontos em abertos que podem levar a trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo aborda os fundamentos principais envolvidos no desenvolvimento da abordagem proposta.

2.1 Sistemas de Recomendação

A gigantesca fonte de informação que a internet se configura atualmente traz alguns problemas relacionados à seleção da informação relevante e sua utilização: a informação encontra-se desorganizada e distribuída em diversos servidores e a quantidade de fontes de dados e serviços aumenta diariamente.

Com o crescimento do comércio eletrônico, houve a necessidade de oferecer serviços personalizados aos usuários e os sistemas de recomendação (SR) (RESNICK et al., 1994) apareceram. Estes sistemas aprendem com as preferências do usuário e automaticamente sugerem produtos que atendam ao seu perfil (ADOMAVICIUS; TUZHILIN, 2005). SRs são utilizados no comércio eletrônico para sugerir produtos ou fornecer informações para ajudar o cliente a decidir sobre a compra (SCHAFER; KONSTAN; RIEDL, 2001).

A abordagem mais popular utilizada em SRs é a filtragem colaborativa (RESNICK et al., 1994) que agrega dados sobre as preferências de clientes para recomendar produtos a novos clientes. SRs baseados na filtragem colaborativa recomendam para um usuário soluções que foram úteis para outros, com o objetivo de auxiliar no processo social de indicar ou receber indicação, seja esta indicação referente a livros, artigos, filmes ou restaurantes (RESNICK et al., 1994).

Um exemplo da aplicação desta abordagem é o sistema implementado em Amazon.com. Na seção de livros, o usuário pode obter recomendação de livros, mas primeiro é solicitado que colabore, avaliando livros que ele tenha lido. Outro exemplo é o MovieLens (MILLER et al., 2003) que também utiliza a filtragem colaborativa para sugerir filmes ao usuário.

Outra abordagem utilizada em recomendação é chamada recomendação baseada em conteúdo (BALABANOVIC; SHOHAM, 1997). Esta abordagem utiliza preferências (passadas e atuais) de um cliente específico para recomendar novos produtos a ele. NewsDude (BILLSUS; PAZZANI, 1999), por exemplo, observa quais são as histórias que o usuário já leu (e também as que ele ainda não leu) e aprende para apresentar novas histórias que ele possa estar interessado.

Na filtragem colaborativa, a recomendação depende da informação prévia de clientes e é necessário um grande número de interações entre usuários e o sistema para que a recomendação seja confiável. Já na abordagem baseada em conteúdo, apenas os dados do usuário atual são utilizados no processo de recomendação. Ambas abordagens, se não forem treinadas com muitos exemplos (avaliações de produtos ou preferências do usuá-

rio), resultam em recomendações pobres. Esta limitação motivou uma terceira abordagem chamada recomendação baseada em conhecimento, que tenta utilizar o conhecimento específico pré-existente do domínio (por exemplo, turismo ou computadores) para construir um modelo de recomendação *online*.

A abordagem baseada em conhecimento é considerada complementar às outras duas abordagens (GUNAWARDANA; MEEK, 2009; BURKE, 2007). O conhecimento sobre clientes e sobre o domínio da aplicação é utilizado para raciocinar sobre quais produtos se enquadram melhor, com as preferências do usuário. A principal vantagem desta abordagem é que ela não depende das avaliações prévias de outros clientes, evitando a dificuldade inicial do sistema.

Conforme apresentado em (LORENZI; RICCI, 2005) nos sistemas de recomendação baseados em conhecimento, o conhecimento pode ser expresso como um modelo detalhado do usuário, um modelo de um processo de seleção ou a descrição de itens que serão sugeridos ao usuário. O conhecimento sobre os usuários e/ou produtos e o domínio da aplicação são utilizados para raciocinar sobre qual produto melhor responde às expectativas do usuário. Estes sistemas sabem como um determinado produto se encaixa na necessidade de um usuário específico (BURKE, 2007) e são capazes de raciocinar sobre a relação entre uma necessidade e uma possível recomendação.

Alguns exemplos de sistemas de recomendação baseados em conhecimento podem ser encontrados na literatura. Em (BURKE, 2000), por exemplo, o autor apresenta o sistema *Entrée*, um sistema de recomendação de restaurantes que provê recomendações encontrando restaurantes parecidos com os que o usuário conhece e gosta, porém em uma outra cidade. A abordagem empregada neste sistema permite que ele responda questões do tipo *"Para qual restaurante parecido com o restaurante X eu poderia ir ¿'*. O sistema permite que o usuário critique as recomendações apresentadas a ele, com o objetivo de modificar a consulta inicial.

Outra abordagem é apresentada em (TOWLE; QUINN, 2000), onde o sistema permite responder a questão *"O que eu posso comprar que eu poderia gostar ou precisar ¿'*. Os autores propõem a utilização de um modelo explícito de cada cliente e modelos explícitos de cada produto do mercado. Dependendo da natureza do mercado e das necessidades para melhorar as vendas, o modelo do cliente pode ser derivado da consulta atual ou dos dados históricos das interações anteriores do cliente.

Raciocínio baseado em casos (RBC) é uma das abordagens utilizadas nos sistemas baseados em conhecimento. Nos sistemas de recomendação baseados em casos (SRBC), o conhecimento é representado através de casos na base de casos. O conhecimento pode ser sobre os produtos, os usuários, as sessões de recomendação, sobre a avaliação do usuário sobre produtos ou até mesmo uma mistura de todos estes exemplos.

No processo de recomendação básico deste tipo de sistema, o usuário está procurando por algum produto para comprar e o sistema solicita suas preferências. Com estas informações o sistema inicia uma busca na base de casos para identificar produtos que possam ser recomendados, ou seja, aqueles produtos que satisfazem as preferências deste usuário. Neste processo pode-se identificar alguns elementos básicos, como a entrada (onde o usuário informa suas preferências), a recuperação de produtos (onde o sistema procura os produtos de acordo com as preferências do usuário) e a saída (onde alguma recomendação é feita ao usuário).

A figura 2.1 apresenta o ciclo de funcionamento de um sistema de recomendação baseado em casos. A partir de um novo problema, o sistema recupera da base os casos mais similares. Estes casos recuperados são recomendados ao usuário. Algumas vezes,

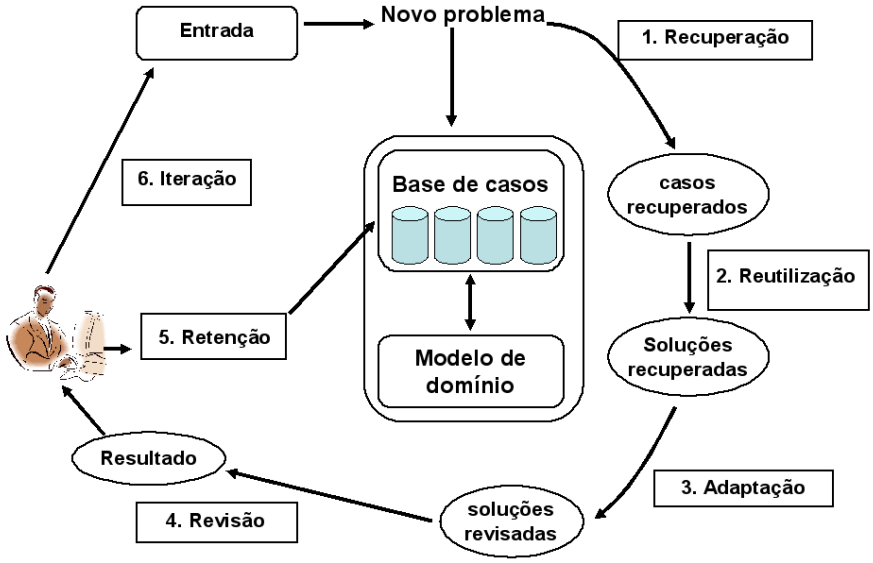

Figura 2.1: Ciclo de funcionamento de SRBC.

as soluções de casos apresentados precisam de alguma modificação e isto é realizado na etapa de adaptação. A solução adaptada passa pelo processo de revisão e após pode ser inserida na base como um novo caso (etapa de retenção). O último passo deste processo é a interação com o usuário, onde este pode criticar o resultado apresentado pelo sistema.

A recuperação de casos é a etapa principal de um sistema de RBC e a maioria dos SRBC implementados podem ser descritos como sofisticadas máquinas de recuperação. Por exemplo, em (MCSHERRY, 2002), o sistema primeiro recupera casos similares da base de casos e em seguida agrupa os casos de acordo com as concessões feitas pelo usuário, apresentando ao usuário apenas um caso representativo de cada grupo.

SRs estão sendo aplicados em diversos domínios (GOY; ARDISSONO; PETRONE, 2007), como na recomendação de livros (amazon.com) ou filmes (netflix.com). Normalmente, estes *sites* da web auxiliam no processo de venda, gerenciando grandes catálogos de produtos. O turismo é outro domínio explorado em recomendações, como o sistema DieToRecs (RICCI et al., 2003), por exemplo. Este sistema sugere serviços de viagens separados ou um plano de viagem completo, e é capaz de personalizar a recomendação com base nas sessões de recomendações previamente armazenadas (vistas como casos). Um dos pontos fortes deste sistema é a representação dos casos. No DieToRecs, um caso representa uma interação entre o usuário e o sistema. Cada caso é construído durante a sessão da recomendação e é composto pelos seguintes itens:

- *Características colaborativas* são atributos que descrevem as características gerais do usuário, seus desejos e objetivos (como por exemplo preferência por um spa ou por praticar esportes). Estes atributos são utilizados para medir a similaridade entre as possíveis opções a serem recomendadas.

- *Características de conteúdo* são atributos utilizados para fazer consultas sobre os catálogos de produtos. De acordo com as características informadas pelo usuário, o sistema busca por produtos a serem recomendados no catálogo de produtos.

- *Carrinho* contêm o conjunto de produtos escolhidos pelo usuário durante a sessão de recomendação.

- *Notas* é uma coleção de notas dadas pelo usuário aos produtos contidos no carrinho.

O usuário interage com o sistema através de consultas, por exemplo, solicitando uma recomendação para um determinado destino. As características colaborativas são utilizadas para gerar um caso e as características de conteúdo são utilizadas para montar a consulta que será aplicada ao catálogo de produtos. A partir desta consulta, o sistema procura por produtos que combinem com as preferências definidas pelo usuário.

Outro exemplo de sistema de recomendação aplicado no turismo é o Sei-Tur (SCHIEL et al., 2007). Este sistema auxilia o usuário a montar uma viagem, considerando os serviços de acomodação, eventos e alimentação. Cada item do plano possui uma ontologia e o Sei-Tur utiliza *web services* durante o processo de criação dos planos. A utilização de *web services* torna-se um ponto fraco do sistema, pois exige que todos fornecedores definam o seu *web service*, onde devem constar todas suas ofertas. No domínio do turismo esta criação de *web service* para cada fornecedor ainda não é uma realidade possível.

A recomendação de pacotes turísticos é uma tarefa complexa que ainda apresenta vários desafios em aberto, tais como, as diversas fontes de conhecimento distribuídas pela internet ou a necessidade de conhecimento específico para compor uma viagem.

Muitas vezes, no processo de recomendação, uma única fonte de informação (*website*) talvez não seja suficiente ou não contenha a informação necessária no processo de recomendação. Os dados disponíveis podem estar fragmentados, superespecializados, muito generalizados, ou até mesmo podem ser irrelevantes para a recomendação solicitada. Assim, o problema de localizar fontes de informações relevantes, acessar, filtrar e integrar informações para resolver um problema tornou-se uma tarefa crítica.

Em sistemas de raciocínio baseado em casos distribuídos, os casos são distribuídos em diferentes agentes e ganha-se na eficiência através da distribuição do trabalho entre os agentes (PLAZA; MCGINTY, 2005). Um trabalho importante nesta área foi apresentado em (NAGENDRA et al., 1996), onde os autores propuseram um modelo formal para recuperação cooperativa de casos através da utilização de subcasos distribuídos em diversos agentes.

Entretanto, cada agente possui conhecimento limitado sobre o domínio e são necessários vários agentes com conhecimento independente para melhorar a eficiência global do sistema (ONTANON; PLAZA, 2005)

Dada a existência de diversas fontes de informações e serviços distribuídos em sistemas e pela internet, que nem sempre estão disponíveis e em algumas vezes oferecem informações incorretas ou desatualizadas e também à característica dinâmica das informações, os sistemas de recomendação baseados na arquitetura multiagente tornaram-se uma abordagem promissora que pode ser aplicada para recuperar, filtrar e utilizar informação relevante para recomendação. Estes sistemas (BALABANOVIC; SHOHAM, 1997; MONTANER; LÓPEZ; ROSA, 2003) são mais eficientes nestas tarefas, quando comparados aos sistemas de recomendação tradicionais não distribuídos.

2.2 Sistemas de Recomendação Multiagente

Agentes autônomos estão sendo utilizados em um grande número de aplicações devido a capacidade de decidir por conta própria sobre seus objetivos e quais tarefas devem

ser realizadas para que estes objetivos sejam atingidos. As aplicações da internet, especialmente as que auxiliam o usuário no processo de decisão, ganharam destaque nos últimos anos, como por exemplo (BUFFETT et al., 2004) (WU et al., 2006) (AL-NAZER; HELMY, 2007).

Em um sistema de recomendação multiagente (SRMA), os agentes são considerados recomendadores não-monotônicos, que se comunicam com o objetivo de trocar informação e fazer inferências com base nestas informações trocadas. Estas características auxiliam no processo de recomendação, porém geram um novo problema: o compartilhamento de informações entre os agentes pode levar a inconsistências em suas bases de conhecimento.

2.2.1 FAB

Um SRMA pioneiro é o FAB (BALABANOVIC; SHOHAM, 1997). Este sistema recomenda aos usuários páginas *web* utilizando as técnicas de filtragem colaborativa e filtragem baseada em conteúdo. O objetivo deste sistema híbrido é explorar as vantagens de ambas técnicas de recomendação utilizadas, minimizando os problemas encontrados em cada uma delas.

O sistema utiliza a técnica de filtragem baseada em conteúdo para criação dos perfis dos usuários. Com os perfis, é possível descobrir os usuários semelhantes, e tendo os usuários semelhantes, o sistema utiliza filtragem colaborativa para recomendar itens ao usuário. Dessa forma, um usuário receberá uma recomendação de um item se este se encaixar no seu perfil ou se esse item se encaixar nos perfis dos usuários similares a ele (correlação usuário-a-usuário).

O processo de recomendação é realizado em duas etapas: a coleta de itens para montar a base de dados e a seleção de itens desta base. O sistema FAB possui 2 tipos de agentes: um agente coletor, responsável pela coleta de documentos e um agente de seleção, responsável pela seleção de páginas a serem recomendadas a um usuário. Cada tipo de agente mantém um perfil (representado por um vetor de palavras), baseado em palavras contidas nas páginas *web* que foram avaliadas. O perfil do agente coletor representa o tópico de interesse atual (que pode ser de interesse para vários usuários) e o perfil do agente de seleção representa o interesse de um único usuário.

Após ter recebido a recomendação, o usuário é solicitado a avaliar o item recomendado, utilizando uma escala *likert* de 1 a 7, onde quanto maior a avaliação mais o usuário gostou do documento que lhe foi recomendado. Essa avaliação é enviada ao sistema para uma atualização dos perfis dos agentes coletor e seleção.

A implementação desta estrutura híbrida no sistema FAB apresenta algumas vantagens, tais como:

- Utilizando recomendações colaborativas, as experiências de outros usuários são levadas em consideração;
- Utilizando recomendações baseadas em conteúdo, é possível lidar com itens ainda não vistos por outros usuários;
- É possível recomendar bons itens a um usuário mesmo que não exista usuários semelhantes a ele.

2.2.2 SmartClients

Um domínio que vem sendo utilizado para validação de aplicações multiagente no comércio eletrônico é o turismo. Em (TORRENS; FALTINGS; PU, 2002), os autores apresentam o sistema *SmartClients*, que utiliza satisfação de restrições para auxiliar o usuário a coletar informação e planejar sua rota aérea. O usuário informa a origem, as cidades que ele deseja visitar e as datas que deseja viajar e o sistema automaticamente constrói uma rede de restrições capaz de explorar as possíveis rotas.

Alguns problemas podem ser apontados nesta abordagem, como por exemplo:

- A partir da definição das restrições e preferências do usuário, o sistema recupera informações do servidor em uma única vez, para evitar vários acessos. Isso limita o espaço de busca;

- Os domínios das variáveis na rede de restrições são fixados pelo sistema. Isto indica que o usuário não pode explorar soluções que estejam fora daquele espaço definido, ou seja, o usuário não tem como criticar o resultado apresentado.

2.2.3 HeraclesII

Outro sistema desenvolvido para planejamento de viagens é o *Heracles II* (AMBITE et al., 2005). Este *framework* de planejamento e coleta de informações tem como objetivos determinar como chegar a um destino (vôos, táxi, ônibus, etc) e escolher uma hospedagem. O sistema modela cada pedaço de informação como uma variável em uma rede de restrições. Esta rede é particionada em diferentes tarefas, que são chamadas de *templates* e estes *templates* são apresentados ao usuário seguindo a hierarquia definida no sistema. Por exemplo, primeiro apresenta-se a rede de como chegar a uma determinada cidade, depois como sair do aeroporto e chegar até um hotel, etc.

O sistema é capaz de fazer uma nova recuperação de informações e modificar a rede durante o processo de planejamento. Isto indica que o usuário pode interagir com o sistema durante o processo de planejamento, alterando os valores das variáveis na rede. Entretanto, o sistema não lida com conhecimento especialista e não considera o conhecimento utilizado em planejamentos anteriores. A cada nova consulta, o processo de planejamento é novamente realizado. Não foi apresentado pelos autores também, se existe algum tipo de avaliação do resultado apresentado pelo sistema.

2.2.4 MAPWEB

Em (CAMACHO et al., 2006), os autores apresentam o sistema MAPWEB, que planeja viagens de acordo com as preferências do usuário. Diferentes agentes foram desenvolvidos: *UserAgent*, responsável pela comunicação do usuário com o sistema. Este agente recebe a solicitação do usuário e é responsável por apresentar o resultado final a ele; *PlannerAgent*, que é responsável pelo planejamento da viagem; e o *Webbot* que busca informação necessária na internet; e o *CoachAgent* que atua como um técnico no grupo de agentes, controlando os agentes, ordenando tarefas aos agentes, gerenciando quem pode auxiliar quem ou até mesmo suspender a comunicação com um determinado agente.

Este sistema apresenta características importantes, como por exemplo, os agentes armazenam os planejamentos já realizados e utilizam este conhecimento para tentar fazer novos planejamentos. Existe também a comunicação entre agentes, com o objetivo de que um agente pode auxiliar o outro na execução da tarefa.

Entretanto, esta comunicação não é um processo natural que acontece entre os agentes. Existe um agente que controla a comunicação possível e indica quem deve auxiliar quem. Apesar do sistema ser distribuído, o controle e gerenciamento das tarefas são centralizados no *CoachAgent*. Além disto, não existe nenhum processo para validar o conhecimento dos agentes. É possível que os agentes utilizem informações desatualizadas durante o processo de planejamento.

2.2.5 CASIS

Outro exemplo de SRMA é o *case-based swarming intelligence recommender system* (CASIS) (LORENZI et al., 2007), um SRBC inspirado em insetos sociais, que utiliza a metáfora da *dança das abelhas* no processo de recomendação de pacotes turísticos baseado em casos. O objetivo do sistema é recomendar pacotes turísticos ao usuário. Os agentes são representados pelas abelhas, que durante o ciclo de recomendação buscam os melhores casos na base de casos de acordo com as preferências do usuário. Cada caso é visto como uma fonte de néctar para os agentes. A partir de uma nova consulta do usuário, as abelhas são distribuídas nas fontes de néctar, com o objetivo de encontrar a melhor fonte.

A vantagem desta aplicação é que as abelhas sempre retornam algum pacote para o usuário, pois elas se adaptam ao ambiente e não dependem de um limiar de similaridade. Normalmente, os sistemas de recomendação baseados em casos usam similaridade pura para recuperar os casos mais parecidos. Os resultados então dependem do limiar de similaridade definido e, algumas vezes, o sistema não encontra casos que tenham o percentual de similaridade acima deste limiar. A desvantagem do sistema CASIS é que a base de casos é centralizada. As abelhas não são capazes de procurar por informação distribuída em outras bases.

2.2.6 Marketplace

A utilização de agentes na área de recomendação pode ser encontrada também na abordagem *marketplace* (CHEN; VAHIDOV; KERSTEN, 2005). Ela pode ser aplicada como método de negociação entre os agentes, com o objetivo de coordenar os métodos de recomendação a serem utilizados a cada ciclo de recomendação. No comércio eletrônico, o primeiro passo de um agente para atingir seu objetivo de compra ou venda de um determinado produto consiste na procura de outras partes interessadas. O *MarketPlace* providencia a forma de concretizar este encontro, entre agentes que pretendem negociar a transação do mesmo produto. Quando entra no mercado, o agente se registra e anuncia o seu objetivo no *MarketPlace*. Isto permite que outros agentes que já entraram no mercado tenham acesso ao seu contato. Esta informação é de acesso público, ou seja, qualquer agente registrado pode questionar o *MarketPlace* sobre os objetivos de outros agentes, ou obter os nomes dos agentes que têm um determinado objetivo. No segundo passo, o agente questiona o *MarketPlace* sobre a existência no mercado de agentes com objetivos complementares ao dele, isto é, objetivos incluindo intenções simétricas sobre o mesmo produto.

Para facilitar a comparação dos produtos, estes são descritos através de pares descritor/valor, que no seu conjunto definem o objeto da negociação. Os valores dos mesmos descritores em objetivos diferentes são comparados para verificar se o produto é o mesmo. Caso exista agentes no mercado com objetivos complementares ao do agente que fez a solicitação ao *MarketPlace*, este responderá indicando os seus nomes. O agente entra em contato com os potenciais oponentes de negociação. Quando os agentes se comunicam,

eles verificam seus objetivos. Este passo é conveniente por duas razões: primeiro, porque a informação disponibilizada pelo *MarketPlace* pode não estar atualizada; e segundo, as funções de comparação dos objetivos dos agentes podem ser mais refinadas do que aquelas aplicadas pelo *MarketPlace*. Por exemplo, ele poderia concluir que a compra e a venda de um automóvel são objetivos complementares, mas um determinado agente poderia ser mais rigoroso na comparação dos dois objetivos, restringindo a marca do automóvel. Este refinamento pode ser feito através da utilização mais minuciosa dos descritores definidos no *MarketPlace*, ou pela inserção, no processo de comparação, de outras informações que por qualquer razão o agente não quis tornar públicas. Finalmente, se os agentes chegarem à conclusão que eles têm intenções simétricas sobre o mesmo produto, iniciarão o processo de negociação.

Em (WEI et al., 2008) a abordagem de *Marketplace* é utilizada como método de negociação em um sistema de recomendação de *websites* multiagente. O objetivo é que seja utilizado o algoritmo de recomendação mais adequado (filtragem colaborativa ou baseada em conteúdo) de acordo com a situação apresentada no momento. Isso é possível pois o processo de mostrar uma URL na barra de endereços pode ser considerado um problema de alocação de recursos. Segundo os autores, uma das melhores formas de alocar recursos é vendê-los usando técnicas de *Marketplace*. Leilões, por exemplo, são muito utilizados no problema de alocação de recursos, distribuindo recursos de acordo com quem paga mais.

A abordagem de *marketplace* foi aplicada da seguinte forma: quando o usuário abre uma janela no navegador, o leilão é acionado. Em cada ativação, o agente leiloeiro pede um número de lances e cada agente *(bidder)* submete um número *X* de lances. Após um tempo determinado o leiloeiro ordena todos os lances e direciona os lances mais altos para a página do navegador, ou seja, o sistema recomenda ao usuário os sites dos agentes que deram os maiores lances. Um detalhe importante neste método deve ser observado: assume-se que o usuário permanece sempre no mesmo contexto e não altera suas preferências durante a utilização do navegador.

O agente leiloeiro (agindo em nome do usuário) vende espaço na barra de endereços. Os *bidders* agem como provedores de informação e agem de forma rápida para terem suas recomendações no navegador do usuário. Uma lista com os melhores lances é mostrada ao usuário, que seleciona as recomendações que lhe interessam (com nota de 0 a 5). Os agentes que tiverem suas recomendações escolhidas pelo usuário são os ganhadores do leilão e são premiados. Ao longo da recomendação, os agentes que derem boas recomendações tornam-se mais ricos e tem suas recomendações mostradas ao usuário com mais frequência.

Percebe-se um ponto negativo neste processo: a falta da característica de *serendipity*. Esta é uma característica necessária nos sistemas de recomendação, onde busca-se surpreender o usuário com recomendações diversas. Como o sistema tem uma tendência de escolher os agentes mais premiados, as recomendações podem se tornar muito repetitivas.

2.3 Sistemas de Manutenção de Verdade

Sistemas de manutenção da verdade (SMV) são algoritmos utilizados para manter a integridade das bases de conhecimento. Estes sistemas são utilizados para detectar contradições nas bases de conhecimento. Eles são responsáveis por excluir as crenças sem justificativas satisfatórias e incluir novas crenças, com justificativas ou que dependam de outras crenças justificadas na base. Crença é uma forma de representação de conheci-

mento temporária suportada pela existência de fundamentos válidos (também conhecidos como justificativas).

O SMV é dividido em dois componentes: o resolvedor de problema (crenças) e o componente SMV que registra as inferências (chamadas de justificativas). O componente SMV é separado do sistema de inferência, conforme apresentado na figura 2.2. O resolvedor de problemas envia as novas informações ao componente de manutenção da verdade para que este valide as justificativas. O componente de manutenção da verdade avalia as informações recebidas e retorna o estado da crença. Em alguns SMV existe ainda um conjunto chamado *nogoods*, que armazena as crenças que não são mais válidas.

O primeiro SMV, sistema de manutenção da verdade baseado em justificativas (SMVJ), no inglês *Justification Based Truth Maintenance System* (JTMS), foi implementado por (DOYLE, 1979). Nesta implementação, cada dado tem um estado associado a ele: acreditado (*in*) ou não-acreditado (*out*).

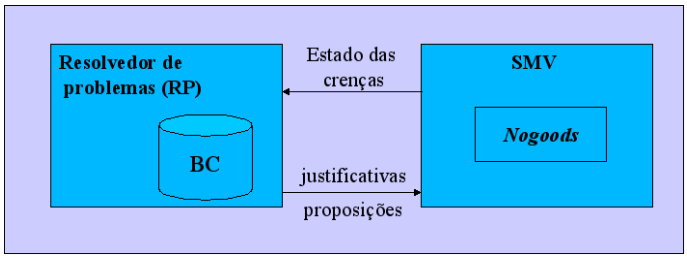

Figura 2.2: Arquitetura de um SMV.

Uma justificativa do JTMS pode ser representada por um par de listas $< listain, listaout >$, onde $listain$ é a lista que contém os dados acreditados e $listaout$ contém os dados não-acreditados. Estas listas são utilizadas para que um nodo justificado seja acreditado e podem representar:

- **Premissa**: quando ambas as listas estão vazias;
- **Dedução**: quando $listain$ não está vazia e $listaout$ está vazia;
- **Hipótese**: quando $listaout$ não estiver vazia.

A justificativa é considerada válida se os dados que estiverem em $listain$ realmente estiverem acreditados e todos que estiverem em $listaout$ forem não-acreditados. O SMV acredita em um dado se ele tiver um argumento para o nodo e para as crenças nos nodos envolvidos neste argumento.

As justificativas criadas podem ser usadas para gerar uma explicação ao usuário. No exemplo apresentado na figura 2.3, as suposições A, C e E são acreditadas e os nodos B, D e F não-acreditados. A, B e C são suposições pois a $listain$ de cada uma delas está vazia e a $listaout$ possui o nodo não-acreditado.

Utilizando as suposições A e C, o resolvedor de problemas pode concluir, por exemplo, que o tempo está bom (dedução G). Ele passa ao SMV a justificativa para o nodo G, transformando este nodo em *in*: $G : tempobom\{(A, C), ()\}$. Com este nodo inferido G e a hipótese E, o resolvedor pode inferir se o tempo está bom para nadar: $H : nadar\{(E, G), ()\}$.

A desvantagem deste algoritmo é a característica conhecida como contexto específico, ou seja, um único ponto do espaço de busca pode ser examinado por vez.

Proposições	**Justificativas**
A : temperatura >= 25	{(), B}
B : temperatura < 25	
C : não está chovendo	{(), D}
D : está chovendo	
E : é dia	{(), F}
F : é noite	

Figura 2.3: Exemplo do JTMS.

Para cada agente que raciocina de forma não monotônica, existem propriedades adicionais usadas para descrever a integridade da sua base de conhecimento. São elas:

- Estabilidade: um estado estável da base de conhecimento é aquele onde cada elemento da base que possui uma justificativa é visto como *acreditado* e cada elemento que não possui uma justificativa válida é visto como *não-acreditado*;
- Fundamentação: a base de conhecimento é bem fundamentada quando nenhum conjunto de crenças é mutuamente dependente;
- Consistência lógica: uma base de conhecimento consistente é aquela considerada estável no momento que a sua consistência for definida e que não possua contradições lógicas.

A evolução do SMV levou ao sistema de manutenção da verdade baseado em suposições, do inglês *Assumptions Truth Maintenance System* (ATMS) apresentado por (DEKLEER, 1986). O ATMS não lida apenas com rótulos do tipo acreditado e não-acreditado, mas baseia-se na manipulação de um conjunto de proposições, onde cada proposição deduzida possui um conjunto de rótulos (suposições a partir do qual foi inferida).

Uma proposição é vista como um nodo, que tem um estado de crença único e justificado: **acreditado**, quando existe um conjunto de suposições consistentes a partir do qual a proposição foi inferida ou **não-acreditado**, quando não existem suposições que levem a acreditar na proposição. Cada nodo pode ser representado da seguinte forma: $< Nodo, Rotulo, Justificativa >$, sendo:

- Nodo: identificação do nodo (proposição);
- Rótulo: conjunto de suposições a partir dos quais o nodo foi inferido. Este conjunto de suposições é chamado de ambiente e podem existir vários ambientes em um ATMS. Um nodo pode pertencer a um ambiente se ele puder ser derivado deste conjunto de suposições. Um ambiente é considerado inconsistente se uma contradição puder ser derivada dele. Os ambientes inconsistentes são chamados de *nogoods*;
- Justificativa: o motivo que leva o nodo a ser acreditado.

A figura 2.4 apresenta o mesmo exemplo visto no JTMS, porém agora aplicado ao ATMS, onde inicialmente existem 4 suposições (A, B, C e D). No ATMS, o conhecimento de domínio necessário para manipular as suposições é dado na forma de regras.

Dado o conhecimento de domínio abaixo para este exemplo:

Proposições	**Rótulo**	**Justificativas**
A : temperatura >= 25	{{A}}	{(A)}
B : temperatura < 25	{{B}}	{(B)}
C : não está chovendo	{{C}}	{(C)}
D : é dia	{{D}}	{(D)}

Figura 2.4: Exemplo do ATMS.

Se a temperatura estiver acima dos 25oC (A) e a temperatura estiver abaixo dos 25oC (B)
então existe uma contradição

Se o tempo estiver bom (E) e for dia (D)
então é hora de nadar (F)

o ATMS é capaz de derivar os seguintes nodos: $E : tempobom\{\{A, C\}\}\{(A, C))\}$ e $F : nadar\{\{A, C, D\}\}\{(E, D))\}$. O nodo F, por exemplo, é derivado do ambiente {A,C,D} e os nodos {E,D} formam a justificativa que torna o nodo F *acreditado*.

A figura 2.5 apresenta a treliça com os possíveis ambientes gerados a partir das 4 suposições iniciais. O ambiente {A,B} é considerado *não-acreditado* (*nogood*), pois os nodos A e B se contradizem. Como consequência, todos ambientes gerados a partir deste, também serão *não-acreditados*.

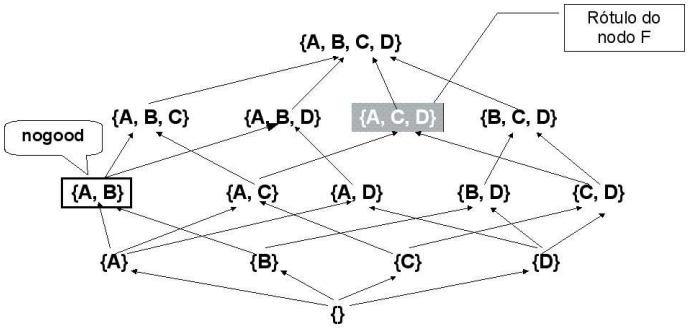

Figura 2.5: Exemplo de treliça de ambientes.

A existência de vários ambientes é uma das vantagens do ATMS, pois significa que diferentes contextos podem ser representados no sistema e garante flexibilidade ao algoritmo, pois é possível lidar com múltiplos estados possíveis de crenças. A desvantagem deste algoritmo pode-se citar o fato dele ter sido projetado para validar a consistência de um único agente.

Em (MASON; JOHNSON, 1989), os autores apresentaram o sistema de manutenção da verdade baseado em suposições distribuído, do inglês *Distributed Assumption Based Truth Maintenance System* (DATMS). Nesta implementação, o objetivo é garantir a consistência das suposições de cada agente, desconsiderando as informações compartilhadas entre eles. O SMV é aplicado em um SMA, onde agentes tomam decisões com base nas suas próprias evidências e/ou informações. Apesar deste algoritmo ter sido aplicado a um SMA, não é permitido que a informação recebida de outro agente altere o estado do dado

do agente. A troca de informações entre os agentes serve apenas para que o agente utilize estas informações para reforçar sua decisão.

Outro algoritmo de manutenção da verdade distribuído é o sistema de manutenção da verdade distribuído (SMVD), no inglês *Distributed Truth Maintenance System* (DTMS) apresentado por (HUHNS; BRIDGELAND, 1991). Nesta implementação, dada uma rede de vários agentes, eles interagem e trocam informações. Cada agente tem dois tipos de dados em sua base de conhecimento: compartilhado e particular. Um dado particular se torna um dado compartilhado quando o agente informa sobre este a outro agente.

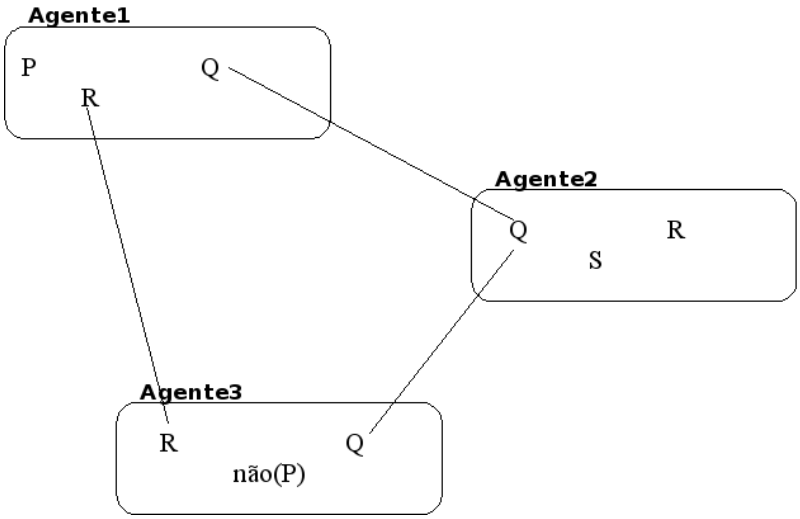

Figura 2.6: Dados compartilhados e particulares.

Nesta abordagem, o rótulo *acreditado* foi dividido em dois rótulos adicionais: *interno* e *externo*. Um dado *interno* é aquele que é acreditado e tem uma justificativa válida. Um dado *externo* é aquele que é acreditado, mas sem precisar de uma justificativa válida. A justificativa de um dado *externo* é "um outro agente compartilhou a informação".

A figura 2.6 apresenta um exemplo do compartilhamento de informações entre os agentes. O dado *Q* por exemplo, está sendo compartilhado entre o *agente1*, *agente2* e o *agente3*. Já o dado *R* está sendo compartilhando entre o *agente1* e o *agente3*, mas é considerado um dado *interno* no *agente2*.

Quatro níveis de estados de consistência da base de conhecimento foram definidos por (HUHNS; BRIDGELAND, 1991). São eles:

- **Inconsistência**: um ou mais agentes são individualmente inconsistentes (um dado sem justificativa válida está acreditado (*in*));
- **Consistência Local**: cada agente é individualmente consistente;
- **Consistência Local e Compartilhada**: cada agente é individualmente consistente e grupos de agentes são mutuamente consistentes sobre o dado que eles compartilham;
- **Consistência Global**: os agentes são individualmente e mutuamente consistentes.

Apesar da consistência global ser o ideal, este nível é extremamente difícil de ser atingido. A preocupação principal na abordagem apresentada por Huhns e Bridgeland foi manter a consistência dos dados compartilhados entre os agentes, pois eles afetam

o processo de resolução de problema de outros agentes. Apesar da consistência local e compartilhada, os agentes podem decidir ter pontos de vista diferentes, ou seja, os agentes podem não concordar com a crença de outro agente.

Quando uma nova justificativa é inserida no SMV, o algoritmo DTMS executa os seguintes passos:

1. Retira os rótulos do dado dos agentes, incluindo a nova justificativa e suas consequências. Se for retirado um rótulo de um dado compartilhado em algum agente, então ele deve ser retirado de todos agentes que compartilham este dado;

2. Rotula cada um dos agentes afetados de acordo com as restrições impostas pelo dado compartilhado (*label-with-constraints*).

A vantagem desta abordagem é que o SMV só é disparado quando existe uma mudança no estado do dado compartilhado. Um sistema de manutenção da verdade multiagente (SMVM) pode ser considerado como uma versão distribuída de um SMV. Em um SMVM existem vários agentes e cada um deles tem seu próprio SMV. Cada agente tem dados que podem estar acreditados ou não-acreditados e podem compartilhar dados com outros agentes. Cada agente deve determinar o rótulo de seus dados de forma consistente e os dados compartilhados devem ter o mesmo rótulo em todos agentes que o compartilham.

Alguns trabalhos encontrados na literatura apresentam implementações de SMV sugerindo algumas melhorias no processo de manutenção da verdade. Em (ALTHEBYAN; HEXMOOR, 2005), por exemplo, os autores criaram um parâmetro para verificar a solidez das proposições derivadas na base de conhecimento dos agentes, auxiliando na decisão de quais proposições devem ser retiradas da base de conhecimento, quando contradições aparecem. A utilização deste parâmetro auxilia o SMV a retirar as proposições corretas, tentando melhorar a eficiência do sistema.

Em (BOJDUJ; WEBER; TAYLOR, 2006) os autores propuseram um agente que raciocina de forma autônoma sobre os eventos do sistema, tentando garantir a integridade da base de conhecimento, independente dos agentes externos. Nesta implementação, os agentes do sistema acessam uma única base de conhecimento e existe um agente de manutenção de verdade que é executado independentemente do sistema. A desvantagem desta abordagem é que, enquanto o agente está verificando a base de conhecimento, buscando por inconsistências que devem ser retiradas, os outros agentes podem utilizar informações incorretas.

No trabalho apresentado por (BOGAERTS; LEAKE, 2004), diferentes estratégias foram investigadas para a recuperação de casos considerando a falta de informações. Quando casos tem descrições parciais do problema é necessário saber quando e como recuperar os casos podem ser recuperados através desta descrição parcial. Os autores apresentaram uma técnica para geração de suposições quando faltam informações sobre a descrição de caso. Entretanto, a técnica apresentada foi utilizada em um contexto centralizado e apenas uma base de casos foi considerada.

O componente de manutenção da verdade proposto nesta tese foi inspirado nos modelos de (HUHNS; BRIDGELAND, 1991) e (DEKLEER, 1986), porém difere destes modelos nos seguintes aspectos:

- A geração das suposições que poderão ser assumidas pelo agente no processo de recomendação é realizada de acordo com a demanda de cada agente. Ao invés de

representar o conhecimento de domínio através de regras, 3 diferentes métodos foram desenvolvidos para a geração das suposições possíveis em cada situação do agente. Estes métodos permitem a geração das suposições de acordo com a necessidade dos agentes durante o processo de recomendação;

- Na atualização do estado do conhecimento dos agentes, os níveis de confiança são considerados com o objetivo de que prevaleça o estado do conhecimento do agente especialista. Além disto, a manutenção da verdade das bases de conhecimento dos agentes não é realizada no momento em que um conflito é detectado, mas sim de uma forma sistemática.

2.4 Confiança

Nos sistemas multiagente, a abordagem de confiança (*trust*) é definida como a expectativa ou chance em que um agente pode confiar em qualquer outro agente (RAMCHURN; HUYNH; JENNINGS, 2004). Como apresentado por (O'DONOVAN; SMYTH, 2005), diversas pesquisas têm sido realizadas em relação ao problema de confiança, o que tem resultado em diversos pontos de vista de como medir ou como utilizar a confiança.

Em sistemas multiagente aplicados ao comércio eletrônico, a abordagem de confiança é utilizada pelos agentes no processo de decisão. Os agentes, muitas vezes, precisam decidir por um determinado item que eles ainda não conhecem com base apenas na recomendação de outros agentes. Por exemplo, na compra de um determinado item, agentes podem solicitar recomendações sobre este item aos seus vizinhos (SABATER; SIERRA, 2001).

No trabalho apresentado em (NASSIRI, 2008), os autores definiram um modelo de confiança onde o objetivo é aumentar o grau de confiança de usuários nos sistemas de comércio eletrônico, tais como, transações bancárias ou compra de produtos. Dois tipos de confiança foram apresentados pelos autores: 1) Confiança na tecnologia: que representa a confiança nos meios de transmissões utilizados no comércio eletrônico; e 2) Confiança social: que representa a confiança na pessoa ou na empresa que está oferecendo o serviço.

Segundo (RAMCHURN; HUYNH; JENNINGS, 2004), dois diferentes tipos de abordagens de confiança podem ser aplicados em sistemas multiagente:

1. O agente precisa ter a habilidade de raciocinar sobre a reciprocidade e honestidade dos outros agentes para que ele confie neles. Esta habilidade é modelada através de um modelo de confiança;

2. O agente precisa ser capaz de calcular o grau de confiança nos demais agentes. Quanto mais alto o grau de confiança em um determinado agente, maior a probabilidade dele interagir com este agente.

As redes de confiança contribuem para o sucesso dos sistemas de recomendação, pois permitem aos usuários estabelecer melhores opiniões sobre itens através do julgamentos de fontes confiáveis que já tenham avaliado estes itens (VICTOR et al., 2009).

Em (MASSA; BHATTACHARJEE, 2004), por exemplo, os autores apresentaram um modelo de confiança que é construído a partir de dados de confiança informados pelos usuários no *site* epinions.com, que permite que diversos usuários revisem diferentes itens, como por exemplo, livros, carros ou até músicas). Os dados de confiança podem ser extraídos e utilizados como parte do processo de recomendação, amenizando os problemas de dados esparsos ou de *cold start* (quando um vasto número de avaliações de usuários

similares é necessário para a construção de recomendações confiáveis para os usuários) apresentados nos algoritmos de filtragem colaborativa. Entretanto, os níveis de confiança entre os usuários não são definidos durante a utilização do sistema, o que compromete o modelo se aplicado a domínios dinâmicos.

Outro importante modelo de confiança aplicado a sistemas de recomendação foi apresentado em (MONTANER; LóPEZ; ROSA, 2002). Neste modelo, o agente utiliza a opinião de outros agentes como conselho no momento de gerar sua recomendação. Quando falta informação sobre um determinado item, o agente busca pela opinião sobre este item nos agentes de sua confiança. Agentes são considerados entidades que podem ser mais ou menos confiáveis e os valores de confiança podem ser calculados nos pares de agentes, com base na comunicação entre estes agentes. Este modelo foi um dos primeiros aplicados a sistemas de recomendação multiagente e ele permite a atualização dos níveis de confiança durante a execução do sistema. Entretanto, este modelo não trata da confiança do agente nele mesmo e não diferencia o tempo das avaliações recebidas pelos agentes.

Um desafio ainda enfrentado pelos modelos de confiança é a dependência de contexto (RAMCHURN; HUYNH; JENNINGS, 2004). A maioria dos métodos de confiança não considera que as interações entre os agentes acontecem em um determinado ambiente. Por exemplo, se um agente resolveu uma tarefa com pouca qualidade devido a mudanças ocorridas no ambiente, ele não deveria ser considerado como não confiável. Ao invés disto, deveria existir a possibilidade de considerar as variáveis do ambiente durante a atualização da confiança entre os agentes.

Considerando os pontos fracos dos modelos de confiança apresentados e os desafios ainda em aberto na área, esta tese propõe um método de confiança que difere-se dos modelos apresentados das seguintes formas:

1. Os níveis de confiança são atualizados de acordo com os resultados das interações entre os agentes durante os ciclos de recomendação;

2. A atualização dos níveis de confiança consideram o tempo de validade das avaliações recebidas pelos agentes;

3. Os níveis de confiança são utilizados para saber com quem os agentes devem trocar informação.

O modelo de confiança desenvolvido nesta tese foi baseado no modelo de Montaner (MONTANER; LóPEZ; ROSA, 2002). Entretanto ele apresenta uma equação de atualização diferente que possui dois objetivos principais:

1. Permitir a especialização dos agentes através do nível de confiança em si mesmo que o agente possui;

2. Através da utilização de pesos na atualização, o modelo de confiança pode ser generalizado para qualquer domínio.

2.5 Conclusões

Conforme apresentado neste capítulo diversos SMAs vêm sendo utilizado em processos de recomendação. A tabela 2.1 apresenta uma comparação de alguns sistemas de recomendação baseados na arquitetura multiagente apresentados. Percebe-se que todos sistemas possuem a característica de coleta de informação, ou seja, todos os sistemas

apresentados utilizam agentes coletores no processo de recuperação de informações. Esta característica torna-se um ponto negativo das abordagens apresentadas, pois existe a possibilidade dos agentes trabalharem com informações desatualizadas.

Tabela 2.1: Comparação dos sistemas multiagente desenvolvidos para o comércio eletrônico

Sistema	Coleta de inf.	Crítica	Aprendizado	Especialização	Confiança
FAB	Sim	Não	Não	Não	Não
MAPWEB	Sim	Não	Sim	Não	Não
Heracles II	Sim	Sim	Não	Não	Não
SmartClientes	Sim	Não	Não	Não	Não

Outra característica importante é a *crítica*, ou seja, a interação com o usuário. Somente o sistema *Heracles II* permite que o usuário critique a informação apresentada. Nos demais sistemas, não existe a possibilidade do usuário opinar sobre a informação apresentada, na tentativa de refinar a consulta.

Sobre a característica de aprendizado, somente o sistema MAPWEB permite que os agentes aprendam ao longo do processo de resolução de problemas. Os demais não utilizam experiências anteriores para solucionar novos problemas.

Quanto à especialização dos agentes e à utilização de confiança entre os agentes, nenhum dos sistemas apresentados permite que os agentes se especializem nas tarefas realizadas ou que utilizem grau de confiança entre eles para compartilhamento de informações.

Neste capítulo foi apresentada também uma comparação entre as diferentes implementações de sistema de manutenção da verdade. A tabela 2.2 apresenta um resumo desta comparação.

Tabela 2.2: Comparação dos sistemas de manutenção da verdade

Sistema	Número agentes	Estados possíveis	Suposições
JTMS	1	*in*/*out*	Não
ATMS	1	*in*/*out*	Sim
DATMS	2 ou mais	*in*/*out*	Sim
DTMS	2 ou mais	interno/externo/*out*	Não

Quanto ao número de agentes, o JTMS e o ATMS são considerados monoagente, ou seja, utilizam apenas 1 agente. O DATMS é considerado multiagente, pois utiliza dois ou mais agentes, que interagem entre si no sistema. Apesar de existir interação, os agentes não influenciam os demais na tomada de decisão. Por último, o DTMS é considerado multiagente, tendo dois ou mais agentes que interagem no sistema.

Quanto aos estados possíveis, o DTMS trabalha com os estados interno, externo e *out*. Isto porque ele é o único que permite que os agentes troquem informações. Os demais sistemas utilizam apenas os estados *in* e *out*.

Outra característica comparada foi a utilização de suposições. Somente o ATMS e o DATMS permitem a utilização de suposições durante o processo de manutenção da verdade. Entretanto, as suposições são geradas inicialmente pelo sistema de manutenção

da verdade na forma de uma treliça e estas suposições devem ser validadas no momento de sua utilização.

Após análise das abordagens de recomendação baseadas na arquitetura multiagente existentes, percebe-se algumas limitações em relação a assertividade da recomendação gerada pelos agentes:

- O número de mensagens trocadas pelos agentes aumenta quando os agentes não sabem quem pode auxiliar na tarefa;
- Os agentes não sabem lidar com a dependência entre as recomendações parciais resolvidas em um processo de recomendação;
- Os agentes não aprendem ao longo de sua existência;
- As suposições geradas pelo sistema de manutenção da verdade são validadas ao longo do processo de utilização destas suposições, o que torna o processo lento, pois são necessárias diversas validações;
- Os agentes não são capazes de se especializar em algum tipo de recomendação.

Com base nestes itens identificados, esta tese apresenta uma nova abordagem de recomendação baseada na arquitetura multiagente que permite que os agentes utilizem suposições quando se deparam com falta de informação para resolver uma determinada tarefa ou compartilhem informações de acordo com o nível de confiança entre eles, e se especializem ao longo de sua existência, armazenando as tarefas resolvidas em suas bases de conhecimento.

3 ABORDAGEM MATRES

Este capítulo apresenta a abordagem MATRES (**M**ulti**A**gen**T** **RE**commender**S**) e está organizado da seguinte forma: nas primeiras seções a abordagem é introduzida e contextualizada e a partir da seção 3.3 são apresentados os principais componentes do agente, como eles foram modelados e seu funcionamento.

O objetivo da abordagem proposta é permitir que os agentes gerem recomendações de qualidade mesmo em situações críticas como a falta de informação necessária para gerar a recomendação. Para isto, a recomendação gerada para uma solicitação do cliente é baseada na colaboração de múltiplos agentes que são capazes de assumir informações, se especializar e confiar ou não nos demais agentes ao longo do processo de recomendação.

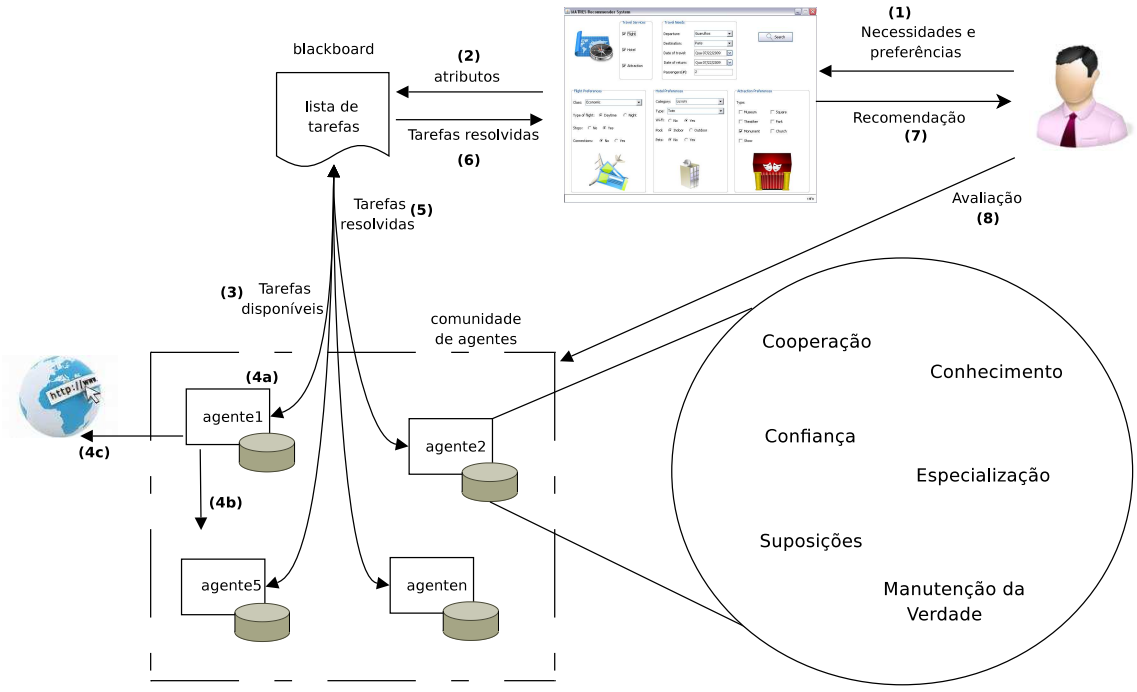

Figura 3.1: Processo da abordagem multiagente de recomendação.

A abordagem multiagente de recomendação proposta neste trabalho é composta por um conjunto de agentes de busca (chamados de *AgentWorkers*) com um objetivo global (a recomendação a ser gerada) e objetivos individuais (as subtarefas que cada agente precisa resolver para que a recomendação final seja gerada). A comunidade de agentes é definida por *C*, onde *C* é um conjunto de *AgentWorkers*, tal que $C = \{a_1, a_2, ..., a_i\}$.

A figura 3.1 apresenta um ciclo de recomendação e os passos definidos na abordagem. O ciclo inicia quando o usuário solicita uma consulta, informando suas necessidades e

preferências (**passo 1**). É considerado um usuário a pessoa que acessa o sistema em busca de uma recomendação. Das necessidades e preferências do usuário são extraídos atributos (**passo 2**), que representam a divisão da consulta em partes menores (recomendações parciais chamadas de tarefas). Estas tarefas são armazenadas em uma lista (**passo 3**) e serão posteriormente realizadas pelos agentes.

Cada agente escolhe uma tarefa para realizar e é responsável por procurar a informação necessária para realizar esta tarefa. O processo de busca pode ser realizado de três formas: (**passo 4a**), quando os agentes procuram pela informação em suas próprias bases de conhecimento, (**passo 4b**), quando os agentes procuram pela informação na comunidade, ou seja, os agentes cooperam e se comunicam com o objetivo de compartilhar informação, ou (**passo 4c**), quando os agentes buscam pela informação na internet.

Após resolvida a tarefa, os agentes alteram o estado da tarefa (**passo 5**) e o resultado da tarefa é enviado para a interface principal (**passo 6**), que é responsável por apresentar a recomendação final ao usuário (**passo 7**). O último passo necessário para complementar o ciclo de recomendação é a avaliação (**passo 8**), onde o usuário avalia a recomendação recebida.

As próximas seções apresentam a modelagem dos componentes da abordagem MATRES e a figura 3.1 será detalhada em cada subseção.

3.1 Tarefas

Em sistemas de recomendação multiagente, uma recomendação pode ser dividida em várias partes para que os agentes possam cooperar e resolver a recomendação. Cada parte da recomendação é considerada uma tarefa e o conjunto de tipos de tarefas é dado por $\mathcal{K} = \{t_1, ...t_n\}$.

Considerando um exemplo de recomendação de filmes, o conjunto de tipos de tarefa poderia ser dado por $\mathcal{K} = \{t_1, t_2\}$, onde t_1 representa o tipo de tarefa *estilo do filme* e t_2 representa o tipo de tarefa *elenco*. Quando o usuário solicita a recomendação e informa suas preferências, o sistema decompõe o problema nestas duas tarefas: *estilo do filme* e *elenco* para que diferentes agentes possam resolvê-las. Em alguns domínios, além das preferências, o usuário pode definir também suas necessidades. As necessidades representam itens que são indispensáveis para o usuário na recomendação.

As tarefas podem ser interdependentes, ou seja, um tipo de tarefa t_n pode ter uma tarefa predecessora. Neste caso, a tarefa necessita de informação proveniente de outra tarefa e, ela só poderá ser realizada após suas tarefas predecessoras terem sido realizadas. O conjunto de tarefas predecessoras é dado por $P(t_n)$ e indica que tarefas devem ser realizadas antes da tarefa t_n.

No exemplo da recomendação de filme, o tipo de tarefa *elenco* depende do resultado da tarefa *estilo de filme*, ou seja, $P(t_2) = \{t_1\}$. Isto indica que, para que o agente resolva a tarefa de *elenco* é preciso ter informações provenientes da tarefa *estilo do filme*.

A partir das preferências informadas pelo usuário é gerada a lista L de tarefas, que representa as preferências do usuário no atual ciclo de recomendação. L é representada através do modelo *blackboard* (ENGELMORE; MORGAN, 1988), ou seja, uma estrutura compartilhada onde cada agente é capaz de visualizar as tarefas disponíveis e escolher uma tarefa para resolver, contribuindo para a recomendação final que será apresentada.

A figura 3.2 apresenta o processo inicial onde o usuário define suas necessidades e preferências e, a partir destas, a lista de tarefas é gerada e disponibilizada aos agentes da comunidade. A resolução da tarefa pelo agente é dada através da busca da informação

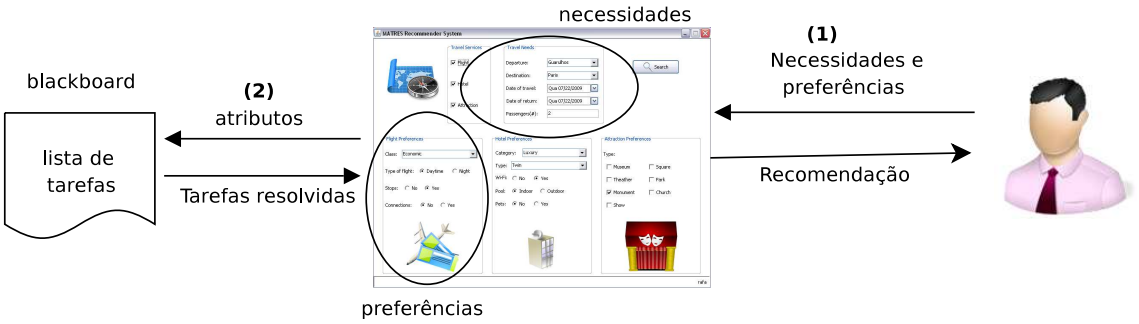

Figura 3.2: Preferências do usuário e geração da lista de tarefas.

necessária para atender a recomendação parcial indicada na tarefa. Após resolvidas as tarefas referente à consulta do usuário, o sistema apresenta a contribuição de cada agente na forma da recomendação final.

3.2 Modelo do usuário

Em diferentes domínios no comércio eletrônico existe a necessidade do cliente fazer uma escolha considerando um conjunto de produtos ou serviços. Entretanto, nem sempre o cliente tem conhecimento suficiente ou até mesmo tempo para tomar uma decisão. Por esta razão, os sistemas de recomendação se tornaram uma ferramenta importante na personalização dos produtos oferecidos ao cliente.

Esta tarefa de gerar recomendações personalizadas aos clientes requer uma modelagem das suas características, preferências e necessidades. Este conjunto de informação é conhecido na literatura como o *modelo do usuário* (KOBSA, 2001). O *modelo do usuário* normalmente contém informações pessoais do usuário (cliente), tais como, nome, endereço, idade, ou informações sobre utilizações prévias do sistema, como, por exemplo, produtos já comprados pelo usuário.

A geração do modelo do usuário pode ser realizada de duas maneiras:

- *Implícita*: quando o sistema aplica alguns mecanismos de raciocínio na tentativa de inferir informações através da observação do comportamento do usuário. A forma *implícita* é mais transparente ao usuário, pois o sistema observa o comportamento do usuário para gerar seu modelo, como por exemplo, quanto tempo o usuário gastou visitando um *website* ou que tipo de produto ele visualizou. O ponto fraco desta técnica é o fato de que o usuário estar analisando um produto ou visitando um determinado *website* não, necessariamente, representa suas preferências. O usuário pode estar comprando um presente para alguém, por exemplo;

- *Explícita*: quando o sistema faz algumas perguntas ao usuário. A forma *explícita* é mais fácil de ser implementada, pois normalmente os sistemas de recomendação geram formulários que devem ser preenchidos pelo usuário. Entretanto, esta é uma prática não atraente para o usuário, pois ele não gosta de preencher muitos campos e na maioria das vezes mente sobre seus dados pessoais.

Na abordagem MATRES optou-se por uma técnica explícita, porém sem a necessidade do usuário preencher um formulário extenso com questões. O usuário deve somente preencher suas necessidades e preferências em relação à recomendação que ele está buscando.

As necessidades representam os itens que o usuário não abre mão na recomendação recebida. Já as preferências são itens onde o usuário é flexível. Por exemplo, se o usuário busca um filme de comédia e não gosta de filmes de terror, esta é considerada uma necessidade e ele não vai aceitar uma recomendação de um filme de terror ao invés de uma comédia. Entretanto, se o usuário solicitou uma recomendação de uma comédia com o ator "Steven Martin" e o sistema recomendar um filme com outro ator, o usuário pode aceitar.

O modelo do usuário é representado por um conjunto de atributos $\mathcal{U} = \{i_1, \ldots, i_m\}$ onde cada atributo i_m pode assumir um valor de um conjunto de valores possíveis dado por $\mathcal{O}_{i_m} = \{o_{i_1}, ..., o_{i_m}\}$. Cada i_m está associado a um tipo de tarefa que pertence ao conjunto de tarefas $\mathcal{K}$.

Na recomendação de filmes, por exemplo, o conjunto de atributos poderia ser dado por $\mathcal{U} = \{i_1, i_2, i_3, i_4, i_5\}$, onde i_1 representa o atributo *diretor*, i_2 representa o atributo *ator*, i_3 representa o atributo *atriz*, i_4 representa o atributo *gênero* e i_5 representa o atributo *tipo do filme*. Cada atributo possui um conjunto de possíveis opções: $\mathcal{O}_{i_1}$ = {steven spielberg, christopher nolan}, $\mathcal{O}_{i_2}$ ={christian bale, robert de niro}, $\mathcal{O}_{i_3}$ ={nicole kidman, glen close}, $\mathcal{O}_{i_4}$ ={comédia, drama, suspense, terror} e $\mathcal{O}_{i_5}$ ={longa, curta, documentário}. O usuário pode escolher uma opção em cada atributo no momento da consulta.

3.3 Modelo do agente

O *AgentWorker* é o componente essencial nesta abordagem e é responsável por resolver parte da recomendação solicitada pelo usuário. Cada agente possui as seguintes habilidades:

- **Cooperação**: os agentes cooperam no processo de recomendação. Cada agente executa sua tarefa com eficiência e solicita informação com outros agentes quando necessário. Assim, o agente busca atingir os objetivos (locais e globais) com qualidade;

- **Comunicação**: os agentes podem se comunicar durante o processo de recomendação, com o objetivo de trocar informações quando necessário. Se um agente precisa de uma informação para compor uma recomendação e esta informação não existe em sua base de conhecimento, ele pode iniciar uma comunicação com outros agentes para obtê-la.

Além das habilidades de cooperação e comunicação, como mostra a figura 3.3, cada *AgentWorker* é composto pelos seguintes componentes:

- **Componente de Manutenção de Verdade (CMV)**: o agente possui um componente de manutenção de verdade que possui duas funções essenciais:

  - **Suposições**: quando o agente não possui informação necessária para gerar uma recomendação, ele é capaz de assumir informação com o objetivo de resolver sua tarefa de recomendação;
  - **Manutenção das bases**: auxilia no gerenciamento de inconsistências nas bases de conhecimento dos agentes que compartilham informações;

- **Confiança**: o agente possui um grau de confiança, que permite que ele confie ou não em outros agentes; e um índice de confiabilidade, que permite que o agente se torne um especialista em um tipo de tarefa ao longo do tempo;

- **Conhecimento**: o agente possui sua própria base de conhecimento. Esta base é incrementada à medida que o agente resolve novas tarefas de recomendações.

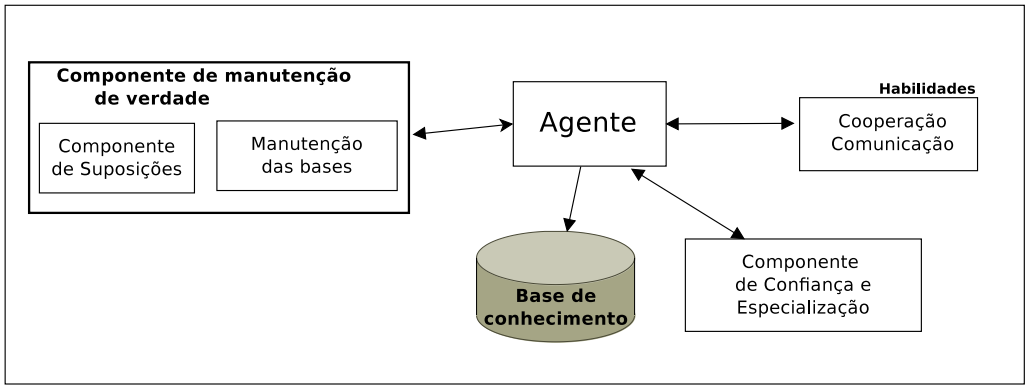

Figura 3.3: Modelo do *AgentWorker*.

As habilidades e os componentes do *AgentWorker* serão detalhados e apresentados nas próximas subseções.

3.3.1 Componente de manutenção da verdade: suposições

Na abordagem MATRES os agentes possuem um componente que permite a geração e manipulação de suposições. Esta habilidade de gerar e manipular suposições é fundamental no processo de recomendação multiagente baseado em conhecimento devido a três diferentes situações que podem acontecer:

- Falta de informação na interdependência existente entre as tarefas que compõem a recomendação: os agentes realizam suas tarefas de forma assíncrona, porém em algumas tarefas precisam do resultado da tarefa que está sendo executada por outro agente. Por exemplo, se o agente a_i está executando a tarefa t_2 dependente do resultado da tarefa t_1, que está sendo executada pelo agente a_j, ele precisa esperar que a_j finalize a tarefa. Muitos problemas podem acontecer durante este processo, como por exemplo, o agente demorar muito para resolver a tarefa predecessora ou até mesmo estar *offline* e não poder finalizar a tarefa. Estes problemas podem levar a um tempo de resposta muito alto, o que é considerado um ponto negativo para um sistema de recomendação multiagente. Na tentativa de agilizar o processo de recomendação, ao invés de esperar pela informação, o agente pode assumir o valor da informação ainda não recebida de outro agente;

- Preferências implícitas do usuário: no momento de uma recomendação, o especialista normalmente considera as preferências do cliente. Este tipo de suposição é interessante, pois o especialista considera o histórico do cliente para gerá-las. Na abordagem MATRES os agentes são capazes de gerar e manipular este tipo de suposição. Estas suposições poderão ser utilizadas para geração de recomendações solicitadas por clientes ainda não conhecidos pelos agentes;

- Não preenchimento de alguma preferência: é possível que o usuário não preencha as preferências necessárias para geração da recomendação. Neste caso, os agentes podem assumir uma suposição em substituição a preferência não informada.

Na falta de informação proveniente de outra tarefa ou na falta de preferências não informadas pelo usuário, o agente pode gerar um conjunto de suposições dado por $\mathcal{S}_{n_{i_m}} = \{s_{1_{i_m}}, ..., s_{n_{i_m}}\}$ onde cada $s_{n_{i_m}}$ representa uma suposição diferente que o agente pode assumir durante o processo de recomendação referente ao atributo i_m.

As preferências implicítas do usuário não são relacionadas aos atributos i_m. Elas são definidas de acordo com o domínio e no momento da implementação. Por exemplo, pode-se definir que uma suposição de preferência implícita é o tipo de filme que uma pessoa prefere, de acordo com o histórico de filmes que ela já assistiu, mesmo que *tipo de filme* não seja um atributo definido.

A abordagem MATRES possui um diferencial em relação às abordagens baseadas em suposições existentes (MASON; JOHNSON, 1989; DEKLEER, 1986): as suposições são geradas conforme a demanda dos agentes. Ao invés de gerar um conjunto inicial de possíveis suposições e validá-lo após a utilização, as suposições são geradas à medida que o agente necessita das informações.

Além das diferentes situações que levam os agentes a assumirem informações, os agentes precisam lidar com diferentes tipos de usuários. Em um sistema de recomendação multiagente existem dois tipos de usuários: *novos*, que estão acessando o sistema de recomendação pela primeira vez, e os usuários *conhecidos*, que já utilizaram o sistema e possuem um histórico de recomendações já solicitadas.

Assumir informações relacionadas a um usuário *conhecido* é mais fácil do que assumir informações de um usuário *novo*, pois as informações podem ser assumidas com base no seu histórico. O usuário *novo* não possui um histórico de antigas recomendações. Torna-se mais difícil para os agentes saberem que informações sobre ele poderão ser assumidas.

Com base nos tipos de usuários, esta abordagem propõe a utilização de três diferentes métodos para gerar suposições no processo de recomendação:

- **MMP** - **M**étodo opção **M**ais **P**opular: o agente assume o_{i_m} para o atributo i_m de acordo com a opção mais popular presente no conjunto de recomendações realizadas pelos agentes para todos usuários. Este método é aplicado para novos usuários, ou seja, para usuários ainda não conhecidos no sistema.

- **MSIM** - **M**étodo casos **SIM**ilares: o agente busca pelo caso mais similar nas bases de conhecimento de todos agentes e utiliza a opção o_{i_m} presente neste caso. Neste método, é necessário definir um limiar de similaridade aceitável para definir o "quanto" um caso é similar a outro. Este limiar deve ser definido de acordo com o domínio. Este método pode ser aplicado na geração de suposições para usuários novos e conhecidos.

- **MMPU** - **M**étodo opção **M**ais **P**opular para o **U**suário: para usuários conhecidos, o agente assume a opção o_{i_m} para o atributo i_m de acordo com a opção mais popular deste atributo presente em todos os casos deste usuário. O agente faz uma busca pelo valor que mais aparece naquele atributo nas recomendações anteriores do usuário.

O algoritmo 1 apresenta a função que executa a geração das suposições através da opção mais popular na comunidade de agentes (*MMP*). O algoritmo inicia percorrendo as bases de casos de todos agentes da comunidade em busca da opção mais preferida pelos usuários. Todos casos do tipo de tarefa t_n são analisados nas bases de todos agentes (linha 12). O algoritmo soma todas ocorrências de todas opções (linha 10), com o objetivo de retornar a opção com o maior número de ocorrências para o tipo de tarefa t_n.

Algoritmo 1: MMP: Opção mais popular na comunidade de agentes

```
1  C é a comunidade de agentes;
2  t_n é o tipo de tarefa;
3  o_{i_m} é a opção;
4  KB(_a) é a base de conhecimento do agente a;
5  Função MMP (t_n,C);
6  início
7      para cada a ∈ C faça
8          repita
9              para cada caso(t_n) ∈ KB(_a) faça
10                 Soma a ocorrência de cada o_{i_m};
11             fim
12         até toda KB(_a) tenha sido verificada ;
13     fim
14     Verifica qual o_{i_m} teve maior ocorrência;
15 fim
16 Retorna o_{i_m} com a maior ocorrência;
```

O algoritmo 2 apresenta a implementação do *MSIM* para geração de suposições. Neste método, todos os casos das bases de todos agentes são comparados com a consulta do usuário q (linha 11) e a opção o_{i_m} do caso mais similar é retornada.

A equação 3.1 apresenta o cálculo realizado pelo agente na comparação dos casos com a consulta do usuário, que resulta o valor da similaridade entre o caso e a consulta no intervalo de [0..1], sendo que 1 representa que o caso e a consulta são exatamente iguais, ou seja, 100% de igualdade entre eles.

$$similaridade(q, caso(t_n) = \frac{\sum_{f=1}^{k} sim(q_f, caso(t_n)_f)}{k} \tag{3.1}$$

Na equação 3.1 q representa a consulta do usuário, $caso(t_n)$ é o caso que está sendo analisado, k representa o número de atributos que serão comparados, f representa os atributos da consulta e do caso, e $sim(q_f, caso(t_n)_f)$ representa a função de similaridade local aplicada, sendo $sim(q_f, caso(t_n)_f)$ = 1, se $q_f = caso(t_n)_f)$ e $sim(q_f, caso(t_n)_f)$ = 0, se $q_f \neq caso(t_n)_f)$. Neste método é necessária a definição de um limiar de similaridade (φ), que representa o valor aceitável de similaridade entre o caso e a consulta.

A implementação do *MMPU* para geração de suposições é apresentada no algoritmo 3. Este método é muito parecido com o *MMP* onde as bases de casos de todos agentes da comunidade são verificadas. Porém são considerados somente os casos resolvidos para o usuário atual que solicitou a recomendação. Isto significa que é necessário que os agentes tenham um histórico deste usuário.

A utilização de suposições no processo de recomendação ajuda os agentes a completarem suas tarefas em um tempo aceitável pelo usuário, sem prejudicar o desempenho do sistema e melhorando a qualidade da recomendação apresentada. Esta contribuição será validada no capítulo 4, seção 4.5.

Algoritmo 2: MSIM: Casos similares

```
1  C é a comunidade de agentes;
2  t_n é o tipo de tarefa;
3  o_{i_m} é a opção;
4  q é a consulta do usuário;
5  KB(_a) é a base de conhecimento do agente a;
6  Function MSIM (t_n,C, q);
7  início
8      para cada a ∈ C faça
9          repita
10             para cada caso(t_n) ∈ KB(_a) faça
11                 Calcula a similaridade entre o caso e a consulta do usuário q -
                   através da equação 3.1;
12             fim
13         até toda KB(_a) tenha sido verificada ;
14     fim
15 fim
16 Retorna o_{i_m} do caso mais similar;
```

3.3.2 Confiança e especialização

Um dos diferenciais da abordagem MATRES é o componente de *confiança* que os agentes possuem. Este componente permite que o agente defina o quanto confia em outro agente, além de permitir a especialização do agente em um determinado tipo de tarefa.

Cada agente possui valores de confiança que são utilizados de duas formas:

- Confiança em si mesmo (chamado de *índice de confiabilidade*): cada agente a_i possui um *índice de confiabilidade* para cada tipo de tarefa t_n. O índice de confiabilidade representa o quanto o agente a_i é especialista no tipo de tarefa t_n e ele auxilia o agente na escolha da tarefa a ser realizada. O agente escolhe a tarefa do tipo t_n que possui o maior índice de confiabilidade;

- Confiança nos demais agentes (chamado de *grau de confiança*): cada agente a_i possui um *grau de confiança* nos demais agentes. O grau de confiança auxilia o agente quando ele precisa se comunicar com outros agentes da comunidade para solicitar informações.

O objetivo destes valores de confiança é tornar o processo de recomendação mais eficiente, evitando comunicação desnecessária entre os agentes e especializando os agentes em tipos de tarefas. Cada agente tem a capacidade de aprender ao longo de sua existência e incrementar sua base de conhecimento, tornando-se um especialista em um determinado tipo de tarefa.

Quando os agentes são criados eles recebem valores de confiança neutros (0,5), ou seja, os agentes inicialmente não possuem opinião sobre os demais agentes e não são especialistas em nenhum tipo de tarefa. Entretanto, esta inicialização pode variar de acordo com o domínio. Dependendo do domínio em que a abordagem está sendo aplicada, pode-se definir que os agentes não devem confiar em ninguém no momento que são criados ou que os agentes inicialmente confiam em todos da comunidade.

Algoritmo 3: MMPU: Opção mais popular no histórico do usuário

```
1  C é a comunidade de agentes;
2  t_n é o tipo de tarefa;
3  o_{i_m} é a opção;
4  KB(_a) é a base de conhecimento do agente a;
5  u é usuário que solicitou a recomendação;
6  Função MMPU (t_n,C);
7  início
8      para cada a ∈ C faça
9          repita
10             para cada caso(t_n) ∈ KB(_a) para usuário u faça
11                 Soma a ocorrência de cada o_{i_m};
12             fim
13         até toda KB(_a) tenha sido verificada ;
14     fim
15 fim
16 Retorna o_{i_m} com a maior ocorrência;
```

Na abordagem proposta, a confiança entre os agentes é baseada em contexto. Os valores de confiança de cada agente dependem do tipo de tarefa. O agente a_1 pode confiar no a_2 em relação a t_1, mas pode não confiar em a_2 em relação a t_2.

Normalmente, a informação sobre a confiança entre os agentes é representada por uma matriz $n \times n$. Entretanto, esta forma de representação não é adequada para domínios dinâmicos e distribuídos, pois a matriz seria centralizada. Na abordagem MATRES, cada agente possui um arquivo local com os graus de confiança nos demais agentes da comunidade, ou seja, os graus de confiança não são vistos por todos agentes.

A cada tarefa resolvida o agente atualiza seus índices de confiabilidade e os graus de confiança (caso tenha utilizado informação compartilhada com outro agente). Esta atualização é realizada de acordo com a avaliação do usuário para as recomendações geradas pelos agentes. Após receber as recomendações, o usuário é convidado a avaliá-las.

3.3.2.1 Avaliação da recomendação apresentada

No final de cada ciclo de recomendação, o usuário é convidado a avaliar as recomendações recebidas pelo sistema. A opinião do usuário sobre a recomendação recebida é importante porque ela é utilizada para atualização dos graus de confiança e dos índices de confiabilidade e, consequentemente auxilia na validação da assertividade das recomendações geradas pelos agentes.

Neste processo de avaliação foi utilizada a escala de atitude *Thurstone* para verificar as opiniões do usuário sobre as recomendações recebidas. Uma escala de atitude é um instrumento criado para medir o grau de intensidade das atitudes e das opiniões de uma pessoa a respeito de um determinado fenômeno (MOWEN JOHN C. E MINOR, 2003).

Como uma recomendação é composta de vários atributos, optou-se pela utilização da escala *Thurstone*, onde o usuário pode concordar ou discordar do valor recomendado em cada item. O usuário avalia os atributos individualmente, definindo uma nota (*gostei* ou *não gostei*) para cada atributo.

O modelo de avaliação é representado pelo vetor $\overrightarrow{E_{t_n}} = \langle e(i_1, r_1), ..., e(i_m, r_m) \rangle$ que descreve a avaliação do usuário em relação a cada item recomendado (r_m), considerando a preferência informada pelo usuário na consulta (i_m). A nota *gostei* significa que o usuário concorda com o valor recomendado e é representada por $e(i_m, r_m) = 1$. A nota *não gostei* significa que o usuário não concorda com o valor recomendado e é representada por $e(i_m, r_m) = 0$.

A média dos valores que o usuário concorda representa a medida da concordância. A equação 3.2 apresenta como a avaliação final $v_{(t_n)}$ é calculada para cada tipo de tarefa t_n, onde m é o número de atributos para cada tipo de tarefa t_n e $e(i_m, r_m)$ é a avaliação para o atributo i_m.

$$v_{(t_n)} = \frac{\sum_{j=1}^{m} e(i_m, r_m)}{m} \tag{3.2}$$

A soma de todas notas *gostei* atribuídas para cada tipo de tarefa t_n representa $v_{(t_n)}$. Assim, para cada tipo de tarefa, a nota de avaliação final pode ser um valor no intervalo $[0..1]$.

3.3.2.2 Atualização da confiança

Quando um agente a_i recebe a avaliação do usuário referente a um tipo de tarefa t_n existem duas situações que podem acontecer:

1. a_i se comunicou com outro agente (a_j) e solicitou informações para resolver a tarefa t_n;
2. a_i realizou a tarefa sem solicitar informações com outros agentes.

$$T_{a_i,a_j}^{t_n} = \frac{\sum_{x=1}^{z} (v_x^{(t_n)} \times e^{-\tau\theta})}{\sum_{x=1}^{z} e^{-\tau\theta}} \tag{3.3}$$

No primeiro caso, quando o agente solicitou informação com outro agente, além do índice de confiabilidade, ele atualiza também o seu grau de confiança em relação ao agente com quem se comunicou para obter informação. Quando a_i se comunica com a_j para solicitar informações necessárias para completar sua tarefa, ele atualiza o grau de confiança em a_j de acordo com avaliação do usuário recebida. A equação 3.3 é utilizada para atualizar o grau de confiança do agente a_i no agente a_j. Nesta equação, tem-se:

- z é o número de avaliações recebidas em recomendações do tipo de tarefa t_n realizadas pelo agente a_i utilizando informação recebida do agente a_j;
- $v_x^{(t_n)}$ é um valor no intervalo $[0, 1]$ que representa a avaliação do usuário para o tipo de tarefa t_n;
- θ representa o número de dias passados entre o dia que a avaliação foi realizada e a data atual. As avaliações mais recentes têm maior influência do que avaliações mais antigas;
- τ é uma constante que define o peso da avaliação $v_x^{(t_n)}$ na atualização do grau de confiança dos agentes. A partir de um tempo de duração da avaliação (d) definido de acordo com o tipo de tarefa é necessário definir o peso que a avaliação tem na atualização dos graus de confiança. Ou seja, este peso representa um desgaste na

avaliação com o passar do tempo. Por exemplo, considerando o exemplo apresentado na tabela 3.1, para d = 50 (avaliação tem prazo de 50 dias), para a avaliação no dia que ela foi efetuada ($\theta = 0$) ter-se-ia $e^{(-0{,}088 \cdot 0)} = 1$. Considerando a avaliação passados 10 dias ($\theta = 10$) ter-se-ia $e^{(-\tau 10)} \approx 0,4148$. Ou seja, a equação tende a zero quando se aproxima do prazo d definido: $e^{(-\tau 50)} \approx 0$. Percebe-se que, à medida que θ aumenta, a avaliação tem um impacto menor no cálculo do grau de confiança do agente. Quanto $\theta = d$, a avaliação tende a 0%, ou seja, não tem influência na atualização do grau de confiança de a_i em a_j.

Após a atualização, $T^{t_n}_{a_i,a_j}$ retorna um valor entre 0 e 1, onde 0 representa o grau mínimo de confiança de a_i em a_j e 1 representa o grau máximo de confiança de a_i em a_j.

Tabela 3.1: Exemplo que mostra o decréscimo da importância da avaliação com o passar do tempo ($d = 50$ dias, τ=0.088)

θ	$e^{(\tau * \theta)}$	%
0	1,0000	100%
10	0,4148	41%
20	0,1720	17%
30	0,0714	7%
40	0,0296	3%
50	0,0123	1%

No segundo caso, a nova avaliação pode aumentar ou decrementar o índice de confiabilidade de a_i em realizar o tipo de tarefa t_n. Se o usuário gostou da recomendação recebida, então o índice de confiabilidade é incrementado, o que significa que o agente está se tornando um especialista no tipo de tarefa t_n. Por outro lado, se o usuário não aprovou a recomendação, o índice de confiabilidade é decrementado.

Como visto na seção 3.1, a lista de tarefas L é criada a partir das preferências do usuário. Quando L está disponível, o agente escolhe a tarefa que pertence ao tipo t_n o qual ele possui o maior índice de confiabilidade. Por exemplo, se o índice de confiabilidade de t_1=0,4, t_2=0,6 e o t_3=0,2, o agente escolherá o tipo de tarefa t_2, pois isto representa que ele resolveu mais tarefas deste tipo e por este motivo possui mais conhecimento em sua base para resolver este tipo de tarefa.

A atualização do índice de confiabilidade é realizada da mesma forma definida para o grau de confiança. Entretanto, ela envolve apenas o agente a_i, conforme apresentado na equação 3.4, não envolvendo outros agentes.

$$T^{t_n}_{a_i,a_i} = \frac{\sum_{x=1}^{z} (v_x^{(t_n)} \times e^{-\tau\theta})}{\sum_{x=1}^{z} e^{-\tau\theta}} \tag{3.4}$$

3.3.3 Base de casos

A base de conhecimento é um elemento importante em um sistema de recomendação multiagente baseado em conhecimento. Cada agente possui sua base de conhecimento onde são armazenadas recomendações já resolvidas por ele.

Na abordagem MATRES, as recomendações resolvidas pelos agentes são armazenadas na forma de casos. Cada caso é composto pela descrição do problema (o conjunto de necessidades e preferências do usuário), pela descrição da solução (a recomendação parcial gerada pelo agente) e pela avaliação do usuário para a recomendação realizada pelo agente.

Cada tarefa realizada pelo agente gera um novo caso que é armazenado em um arquivo XML. Cada caso é composto por:

- **A identificação do usuário**: a identificação do usuário será utilizada pelo sistema para identificá-lo nas próximas vezes que ele solicitar recomendações. Esta identificação é composta pelo nome e senha do usuário;
- **A consulta**: a consulta é o conjunto de necessidades e preferências definidas pelo usuário. Este conjunto representa a descrição do problema.
  - **Necessidades do usuário**: os agentes utilizam as principais necessidades do usuário como filtros durante o processo de busca da informação em suas bases. Estas necessidades representam aqueles itens que o usuário não abre mão em sua recomendação.
  - **Preferências do usuário**: o conjunto de preferências representa os desejos adicionais do usuário. As preferências diferem-se das necessidades, pois o usuário pode abrir mão de alguma preferência quando não for possível a recomendação de itens de acordo com todas as preferências definidas por ele.
- **A recomendação**: os itens (serviços) que foram recomendados pelo agente ao usuário. Estes itens representam parte da solução do problema, ou seja, parte da recomendação para a consulta definida pelo usuário. A recomendação é particionada de acordo com as tarefas definidas no modelo do usuário. Para cada item armazenado, cada agente armazena também:
  - *Fonte*: a fonte de informação do item recomendado. A fonte pode ser:
    - o próprio agente que realizou a tarefa: quando o agente encontra informação em sua base de conhecimento;
    - outro agente: caso a informação tenha sido solicitada para outro agente;
    - uma suposição assumida pelo agente.
  - *Estado*: o estado da informação pode ser *verdadeiro* ou *falso*. O estado indica se a informação está atualizada.
  - *Avaliação*: a avaliação do usuário para cada item recomendado pelo agente.

A tabela 3.2 apresenta um exemplo de uma consulta do usuário no cenário de recomendação de filmes e a tarefa *elenco* realizada pelo agente a_1. A recomendação desta tarefa foi armazenada na base de conhecimento de a_1, juntamente com a consulta do usuário e com as avaliações do usuário, conforme mostra a tabela 3.2. Analisando o exemplo, percebe-se que o agente solicitou informação sobre o atributo *atriz* com o agente a_3.

O conjunto de arquivos XML, com as diversas tarefas realizadas, compõe a base de conhecimento do agente. Não existe um limite de arquivo para cada agente, mas como trata-se do armazenamento de casos, será definido posteriormente algum tipo de mecanismo para controlar o tamanho das bases de conhecimento dos agentes. Quanto maior for a base, maior será a cobertura do domínio e maior também a probabilidade de encontrar a solução. Entretando, perde-se desempenho, pois o processo de recuperação de casos similares se torna mais lento.

Além da base de conhecimento, cada *Agentworker* possui um arquivo com informações necessárias para o processo de recomendação, tais como: sua identificação e a tarefa que ele está executando, conforme pode ser visto na figura 3.4.

Tabela 3.2: Exemplo de um caso na recomendação de filmes - a_1

< **case** > <user> <username> Fulano de Tal </username> <password> 1234 </password> < /user > < **query** > < **elenco** > <diretor> stanley kubrick </diretor> <ator> robert de niro </ator> <atriz> nicole kidman </atriz> </**elenco**> < **estilo** > <genero> drama </genero> <tipo> longa </tipo> </**estilo**> </**query**> </**case**>	<**recommendation**> <**elencoR**> <diretor> stanley kubrick </diretor> <avaliação> true</avaliação> <fonte> 1</fonte> <status> true</status> <atriz> nicole kidman </atriz> <avaliação> true</avaliação> <fonte> 3</fonte> <status> true</status> <ator> tom cruise </ator> <avaliação> true</avaliação> <fonte> 1</fonte> <status> true</status> </**elencoR**> </**recommendation**>

Figura 3.4: Informações do *AgentWorker*

```
<agent ID="3">
<ctask>9</ctask>
</agent>
```

3.3.4 Componente de manutenção de verdade: manutenção das bases

O componente de manutenção das bases faz parte do componente de manutenção de verdade (CMV) desenvolvido nesta abordagem. Ele tem como objetivo auxiliar a manter a consistência do conhecimento compartilhado entre os agentes.

A manutenção de verdade das bases de conhecimento dos agentes deve ser realizada quando conflitos (inconsistências) são detectadas entre os agentes. Segundo (MALHEIRO; VARGA; OLIVEIRA, 1998), os conflitos entre os agentes são muito comuns e frequentes em um sistema multiagente e as causas podem variar, como por exemplo:

- Em contextos dinâmicos alguns agentes podem ter informação mais recente e completa que outros e as diferenças entre o conhecimento dos diversos agentes faz com que apareçam os conflitos;
- A limitação de recursos, onde os agentes não conseguem utilizar os recursos simultaneamente;
- Agentes possuem conhecimento e capacidades diferentes, podendo gerar diferentes respostas para as mesmas questões.

Em sistemas de recomendação multiagente que lidam com cenários dinâmicos, as inconsistências podem surgir após os agentes compartilharem informações. Por exemplo,

a_1 e a_2 compartilharam uma informação sobre o ano de lançamento de um filme. Entretanto, uma atualização sobre o ano de lançamento do filme pode ser enviada aos agentes e somente um deles atualizar a informação em sua base. Por exemplo, a_1 recebeu a informação e atualizou sua base, mas a_2 não estava disponível e, por sua vez, não atualizou a informação em sua base. Isto significa que os dois agentes têm estados diferentes para a mesma informação, ou seja, existe uma inconsistência em suas bases de conhecimento.

Para gerenciar estas possíveis inconsistências, os agentes utilizam o componente de manutenção das bases. Conforme visto na seção 2.3, existem quatro níveis de consistência: inconsistência, consistência local, consistência local e compartilhada e consistência global. Uma base de conhecimento é consistente se ela foi considerada estável no momento em que sua consistência foi definida e se ela não possui contradições lógicas.

Na abordagem MATRES, o componente de manutenção das bases busca os níveis de *consistência local*, para garantir que cada agente seja individualmente consistente e *consistência local e compartilhada*, tentando garantir que cada agente seja consistente individualmente e grupos de agentes sejam mutuamente consistentes sobre a informação que eles estejam compartilhando. Considerando estes dois níveis de consistência, o componente pode ser disparado de duas formas diferentes.

Para garantir a consistência local de cada agente, o componente de manutenção das bases verifica a nova informação que está sendo incluída na base de conhecimento do agente. Este processo é rápido e de fácil execução, pois é apenas uma verificação de que o agente realmente ainda não possui a informação que está sendo armazenada. Se a informação já existe na base de conhecimento do agente, ele apenas atualiza a informação em sua base. O componente de manutenção das bases não permite que a mesma informação seja armazenada na base com estados diferentes, evitando assim a inconsistência da base do agente.

No nível de consistência local e compartilhada, os dados *compartilhados* entre os agentes podem ter dois estados: *verdadeiro* em todos agentes que compartilham este dado ou *falso* em todos esses agentes. Esta propriedade é muito importante para manter a integridade nas bases de conhecimento, porque os dados compartilhados afetam o processo de resolução de problema de outros agentes.

Entretanto, como vários agentes podem estar compartilhando uma informação, atingir a consistência local e compartilhada se torna mais difícil em domínios dinâmicos. Dada a forma assíncrona que os agentes realizam suas tarefas, quando atualizações são enviadas aos agentes, nem todos são capazes de atualizar suas informações. Alguns agentes podem estar ocupados resolvendo tarefas ou podem não estar conectados no momento da atualização, o que leva a não atualização do estado da informação.

Por este motivo, o componente de manutenção das bases desta abordagem é disparado no momento que o sistema de recomendação é aberto pelo usuário. Cada agente executa a manutenção de verdade das informações compartilhadas armazenadas em sua base.

O algortimo 4 apresenta o algoritmo de manutenção das bases desenvolvido nesta abordagem com o objetivo de atingir a *consistência local e compartilhada* das bases dos agentes. Dada a comunidade de agentes, o primeiro passo do algoritmo (linha 5) é detectar todos agentes que compartilham informações (γ). A partir dai, o algoritmo verifica o estado da informação compatilhada (γ) pelos agente, alterando seu estado conforme o índice de confiabilidade do agente mais especialista (linha 6). Para evitar que todos agentes verifiquem todas informações, o algoritmo registra na lista de visitados a primeira vez que a informação compartilhada entre a_i e a_j é verificada (linha 11).

Uma das deficiências do componente de manutenção da verdade apresentado por

Algoritmo 4: Componente de manutenção das bases de conhecimento dos agentes

```
1  C é a comunidade de agentes;
2  Procedure manutenção das bases de conhecimento (C);
3  início
4      para cada a ∈ C faça
5          Verifica todos agentes que possuem informações compartilhadas (γ) com a_i;
6          para cada γ ∈ a_i faça
7              para cada a_j que compartilha γ com a_i faça
8                  se (índice de confiabilidade de a_i > índice de confiabilidade de a_j)
                   então
9                      Altera o estado da informação γ em a_j conforme o estado de γ
                       em a_i;
10                 fim
11                 Inclui a_j na lista de visitados para informação γ;
12             fim
13         fim
14     fim
15 fim
```

(HUHNS; BRIDGELAND, 1991) é o fato do componente permitir que um agente com menos informação altere o estado de um dado que pertence a um agente com mais informação sobre aquele dado. Se, por exemplo, dois agentes tiverem a mesma justificativa para a crença em um dado compartilhado e um deles aprende um fato que invalida esta justificativa, a justificativa válida do outro agente será o suficiente para que ambos continuem acreditando neste dado.

A abordagem MATRES propõe a utilização da *especialização* e da confiança dos agentes (detalhada na seção 3.3.2) para lidar com este problema, contribuindo na melhoria do componente de manutenção de verdade. Através do índice de confiabilidade e do grau de confiança entre os agentes, o agente mais especialista naquele tipo de tarefa (e que tiver a confiança do outro agente) ganhará a disputa e a informação terá a justificativa que ele acredita.

Como pode ser visto no algoritmo 4, a informação γ somente será atualizada se o agente a_i tiver mais experiência do que o agente que originalmente compartilhou a informação com os demais agentes, ou seja, se o seu índice de confiabilidade naquele tipo de tarefa for maior do que o índice do outro agente.

Este processo é de extrema importância para os agentes, pois evita informação desatualizada nas bases dos agentes. Em ambientes dinâmicos, a alteração constante de informações é normal e por este motivo é necessário um mecanismo que previna inconsistências nas bases dos agentes.

É importante notar o diferencial do algoritmo de manutenção de verdade proposto através da utilização dos índices de confiabilidade, o qual permite que a informação somente seja atualizada se o agente for especialista no tipo de tarefa. Isto evita que agentes inexperientes alterem o estado de informações de outros agentes.

3.4 Resolução de tarefas

Conforme visto na seção 3.1, novas tarefas são disponibilizadas aos agentes a cada ciclo de recomendação. Os agentes então escolhem as tarefas a serem resolvidas de acordo com seus índices de confiabilidade. Para resolver a tarefa escolhida, ou seja, para gerar a recomendação solicitada na tarefa, os agentes precisam buscar a informação necessária para a recomendação.

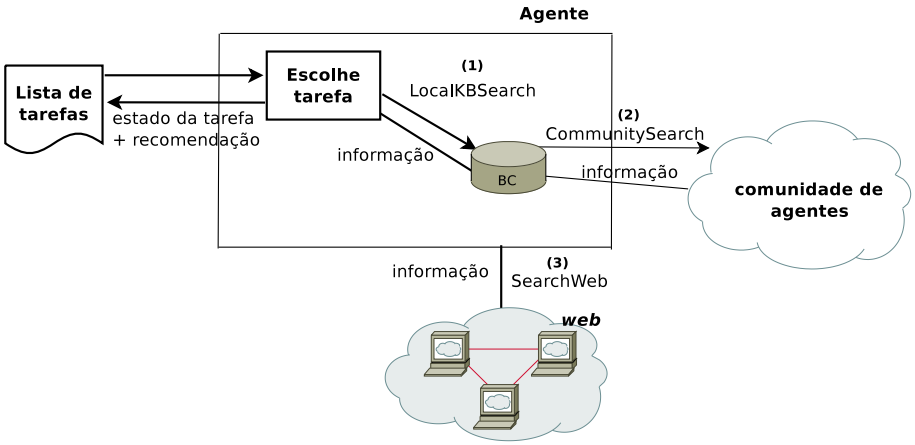

Figura 3.5: Processos do agente para resolução das tarefas.

Conforme apresentado na figura 3.5, os agentes possuem 3 diferentes formas de busca da informação para resolução de suas tarefas:

- *LocalKBSearch*: o agente busca pela informação em sua própria base de conhecimento;
- *CommunitySearch*: o agente busca pela informação nos demais agentes da comunidade, pois ele não possui a informação em sua base;
- *SearchWeb*: o agente busca pela informação na *web*.

Estas três formas de busca de informação são descritas no algoritmo 5. Dada a lista de tarefas L, o agente escolhe uma tarefa para realizar (linha 7). Conforme apresentado na seção 3.3.2, o agente escolhe a tarefa de acordo com os índices de confiabilidade dos tipos de tarefa.

Após escolher a tarefa, o agente altera o estado desta tarefa (linha 8), evitando que outro agente tente resolver a mesma tarefa e inicia o processo de busca pela informação necessária para resolvê-la.

A primeira etapa deste processo é o procedimento $LocalKBSearch$ (linha 9), onde o agente busca pela informação em sua própria base de conhecimento.

Se o agente não encontrar a informação em sua base, ele então inicia o segundo processo de busca, através do procedimento $CommunitySearch$ (linha 12). Neste procedimento, o agente se comunica com agentes de sua confiança com o objetivo de compartilhar a informação necessária.

Se estas duas buscas falharem, o agente executa um processo de busca na internet (linha 14). Neste processo ele busca pela informação, apresenta a recomendação e armazena o caso resolvido em sua base.

Algoritmo 5: Resolvendo tarefas

```
1  C é a comunidade de agentes;
2  L é o conjunto de tarefas a serem realizadas;
3  ι é a tarefa a ser resolvida;
4  I é a informação necessária na tarefa;
5  Function Resolve_tarefa (a_i,L,C);
6  início
7      Escolhe a tarefa ι do tipo de tarefa t_n com o maior índice de confiabilidade;
8      Altera o estado da tarefa ι para que outro agente não selecione a mesma tarefa;
9      I ← LocalKBSearch(ι);
10     se I = ∅ então
11         para a_j ∈ C com o maior grau de confiança faça
12             I ← CommunitySearch(a_j, ι);
13             se I = ∅ então
14                 I ← SearchWeb(ι);
15             fim
16         fim
17     fim
18 fim
19 Retorna(I);
```

3.4.1 Busca da informação: localKBSearch

O agente inicia o processo de resolução de tarefa, buscando pela informação em sua própria base de conhecimento. Como o conhecimento está armazenado na base na forma de casos, o agente procura em sua base por recomendações anteriores que sejam parecidas com a tarefa escolhida. Para isto, o agente aplica o algoritmo de recuperação de casos baseado na similaridade entre os atributos da tarefa e dos casos armazenados em sua base.

O agente compara a informação da tarefa (que é parte da consulta do usuário) com todos os casos armazenados em sua base, ou seja, compara todas as recomendações já feitas por ele com esta nova solicitação do usuário. Esta comparação é realizada conforme apresentado na equação 3.5.

$$similaridade(q, c) = \frac{\sum_{f=1}^{k} sim(q_f, c_f)}{k} \tag{3.5}$$

onde:

- q é a consulta do usuário;
- c é o caso que está sendo comparado;
- f é o atributo;
- k é o número de atributos;
- $sim(q_f, c_f)$ é a similaridade local entre o atributo do caso e o atributo da consulta. Como todos atributos envolvidos no caso são simbólicos, o resultado desta similaridade local será 1 (quando $q_f = c_f$) ou 0 (caso contrário).

É importante salientar que a comparação entre a tarefa e os casos da base do agente apresentada na equação 3.5 pode ser modificada de acordo com o domínio e com os tipos de atributos definidos. Em alguns domínios, por exemplo, pode haver a necessidade da definição de pesos para atributos, refletindo a importância dos atributos na comparação. Nos domínios estudados percebeu-se que a comparação dos atributos, sem a utilização de pesos, é eficiente e por este motivo este tipo de equação foi escolhido.

Considerando um exemplo de consulta do usuário com as seguintes preferências: *elenco* ('steven spielberg', 'robert de niro', 'nicole kidman') and *estilo* (comedia, longa), e que a_1 tenha escolhido a tarefa *elenco* para resolver. Imaginando que a base de conhecimento de a_1 tenha dois casos:
Caso 1: *elenco* ('steven spielberg', 'christian bale', 'nicole kidman')
Caso 2: *elenco* ('christopher noan', 'robert de niro', 'glen close')

Comparando a consulta do usuário com o *caso 1*:
Sim(q, caso1) = (('steven spielberg', 'steven spielberg') + ('robert de niro', 'christian bale') + ('nicole kidman', 'nicole kidman')
Sim(q, caso1) = (1 + 0 + 1)
Sim(q, caso1) = 2/3 = 0,66

e comparando a consulta com o *caso 2* o resultado seria:
Sim(q, caso2) = (('steven spielberg', 'christopher noan') + ('robert de niro', 'robert de niro') + ('nicole kidman', 'glen close')
Sim(q, caso2) = (0 + 1 + 0)
Sim(q, case2) = 1/3 = 0,33

O caso com a maior similaridade é considerado o mais parecido com a consulta do usuário. No exemplo acima, o *caso 1* é o mais parecido conforme as preferências do usuário, e ele será escolhido pelo agente para ser recomendado.

Para encontrar o caso mais parecido é importante definir um limiar de similaridade. Por exemplo, considerando valores de 0 a 5, o especialista no domínio pode definir que um resultado até 2,5 é considerado similar e pode ser utilizado na recomendação. O limiar define o quanto o caso é parecido com a consulta. Este limiar deve ser definido pelo especialista na etapa de validação do sistema e pode variar de acordo com o domínio.

Se o agente não encontrar a informação em sua base de conhecimento, ou seja, se todos os resultados das comparações forem menor do que o limiar definido, o agente passa para o segundo nível de busca, chamado *CommunitySearch*, onde inicia a comunicação com agentes da comunidade.

Algoritmo 6: *LocalKBSearch*

```
1  ι é a tarefa a ser resolvida;
2  c representa um caso da base;
3  limiar representa o limiar de similaridade previamente definido pelo especialista;
4  KB_a é a base de conhecimento do agente a;
5  Function LocalKBSearch (ι);
6  início
7      menorsimilaridade ← 0;
8      para cada c ∈ KB_a faça
9          Calcula a similaridade entre o caso c e ι;
10         se menorsimilaridade > similaridade calculada então
11             menor ← similaridade calculada;
12             info ← informação do caso c;
13         fim
14     fim
15     se menor < limiar então
16         info ← c;
17     fim
18 fim
19 Retorna(info);
```

3.4.2 Busca da informação: communitySearch

Quando o agente não encontra a informação necessária para resolver sua tarefa na sua base de conhecimento, ele inicia o processo de busca da informação na comunidade de agentes.

Para evitar um número alto de troca de mensagens entre os agentes da comunidade, na abordagem MATRES a comunicação entre os agentes depende do grau de confiança entre eles.

Conforme apresentado na figura 3.6, o agente a_i inicia a comunicação com o agente em que ele mais confia (passo 1). Caso ele receba a informação que está procurando (passo 2), ele encerra a comunicação e retorna para o processo de recomendação (passo 3). Caso contrário, o agente continua o processo de comunicação, enviando uma mensagem de solicitação ao segundo agente que ele confia (passo 4) e assim sucessivamente até encontrar a informação ou após ter se comunicado com todos agentes confiáveis disponíveis na comunidade.

Este protocolo de comunicação torna o processo mais eficiente, pois o agente somente se comunica com agentes de sua confiança, evitando trocas de mensagens desnecessárias. É possível definir um limite de agentes confiáveis com quem o agente pode se comunicar. Este limite depende do domínio da aplicação de recomendação.

3.4.3 Busca da informação: searchWeb

Quando a busca local e a busca na comunidade falharem, o agente dispara o processo de busca na *web*. Neste processo, o agente faz uma busca direta pela informação necessária para compor a recomendação.

Uma ferramenta de busca na *web* foi desenvolvida com o objetivo de permitir que o agente procure na *web* pela informação necessária para completar a recomendação.

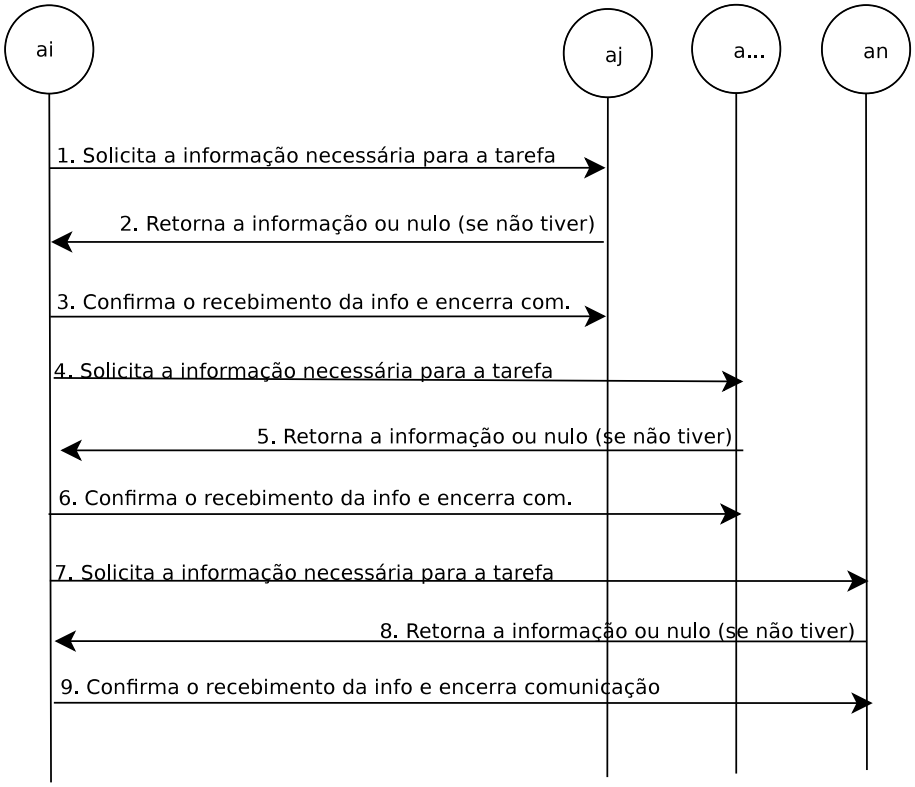

Figura 3.6: Comunicação entre o agente a_i e os agentes de sua confiança.

Com base nas preferências do usuário e de acordo com o tipo de tarefa que está sendo realizada, o agente busca pela informação nos *websites* disponíveis na Internet.

Esta ferramenta permite que os agentes procurem pelas informações de acordo com a necessidade e a informação pesquisada está sempre atualizada. Porém, é importante salientar que esta ferramenta foi desenvolvida somente para validar a abordagem proposta e não foi escopo de estudo nesta tese.

3.5 Conclusões

Este capítulo apresentou a abordagem MATRES, bem como o detalhamento de todos seus componentes. Os diferentes componentes da abordagem MATRES fazem o seu diferencial, contribuindo para o estado da arte de sistemas de recomendação multiagente, e permitem que os agentes gerem recomendações de qualidade mesmo em situações críticas como, por exemplo, na falta de conhecimento durante o processo de recomendação.

O componente de suposições é responsável pela geração de suposições que os agentes podem precisar durante o processo de recomendação. Estas suposições são imprescindíveis quando os agentes lidam com falta de informação em tarefas inderdependentes, ou seja, quando um agente está realizando uma tarefa que precisa do resultado proveniente de outro agente.

O componente de confiança permite que os agentes se especializem ao longo de sua existência e que eles mantenham um grau de confiança nos demais agentes. A utilização dos graus de confiança são importantes pois evita comunicações desnecessárias entre os agentes e evita a comunicação com agentes maliciosos que podem prejudicar a qualidade das recomendações geradas.

Como os agentes possuem suas próprias bases de conhecimento, a abordagem conta

também com o componente de manutenção das bases que auxilia no controle da consistência das bases de conhecimento. A consistência das bases de conhecimento é imprescindível para evitar a geração de recomendações errôneas durante o processo.

O próximo capítulo discute a validação da abordagem MATRES que foi aplicada no domínio do turismo e os resultados obtidos nos experimentos são apresentados.

4 APLICAÇÃO E VALIDAÇÃO DA ABORDAGEM NO DOMÍNIO DO TURISMO

Este capítulo apresenta a aplicação e validação da abordagem MATRES no domínio do turismo, os experimentos realizados na validação aplicada na recomendação de pacotes turísticos e os resultados obtidos.

A próxima seção apresenta uma introdução do domínio do turismo. A seção 4.2 apresenta a modelagem da abordagem MATRES no cenário do turismo, a seção 4.3 apresenta os experimentos realizados no cenário do turismo para validação da abordagem MATRES, a seção 4.5 apresenta e discute os resultados obtidos e finalmente a seção 4.6 apresenta a conclusão deste capítulo.

4.1 Introdução ao domínio do turismo

O turismo foi escolhido como cenário teste da abordagem MATRES devido a sua natureza dinâmica e à complexidade da composição da recomendação de um pacote turístico. Neste tipo de recomendação diversos componentes são necessários, como por exemplo, meio de transporte, acomodação, alimentação, passeios e atrações a serem visitadas. É necessário conhecimento específico para reunir estes componentes em uma única recomendação (RICCI, 2002), considerando as dependências entre os componentes.

O mercado do turismo é distribuído e diversos provedores de serviços ou intermediários gerenciam e armazenam em seus bancos de dados as informações sobre serviços prestados e também sobre seus clientes (WERTHNER; RICCI, 2004).

Para recomendar um pacote turístico, um intermediário (o agente de viagem) deve construir um modelo representando todos elementos (informações) necessários para gerar esta recomendação. Este modelo pode estar implicitamente definido em sua mente ou explicitamente documentado como um plano na agência de viagem. Estes elementos podem incluir recursos (informação, produtos ou serviços), clientes e suas solicitações, fatores que influenciam na recomendação (como por exemplo a temporada), estratégias imediatas para encontrar a melhor opção para o cliente e assim por diante.

Entretanto, planejar uma viagem ou montar um pacote turístico não é uma tarefa desempenhada por um único intermediário e a colaboração entre os agentes de viagem é necessária para integrar as experiências individuais em um plano coerente que satisfaça as necessidades do cliente. Esta é mais uma razão da escolha do domínio do turismo como cenário teste para a validação da abordagem MATRES.

4.2 Definições no cenário do turismo

Todos os componentes modelados e apresentados no capítulo 3 foram implementados no domínio do turismo, em um cenário de recomendação de pacotes turísticos. A recomendação de um pacote turístico é baseada na colaboração de múltiplos agentes que compartilham informações armazenadas em suas bases de conhecimento. Uma solicitação de recomendação do usuário é decomposta em subtarefas que são realizadas por diferentes agentes.

4.2.1 Modelo do usuário

Conforme já apresentado na seção 3.2, é necessário modelar as preferências do usuário. Para que os agentes sejam capazes de recomendar as melhores opções ao usuário, é preciso que este defina suas necessidades e preferências.

Na recomendação de pacotes turísticos, as necessidades representam as informações fundamentais para que a recomendação seja gerada, como por exemplo, a cidade origem do passageiro, o destino da viagem, a data de saída, data de retorno e o número de passageiros.

Após informadas as necessidades, o usuário pode definir suas preferências, como por exemplo, voo diurno ao invés de voo noturno ou primeira categoria de hotel ao invés de categoria turística. É importante salientar que um usuário neste cenário exemplo pode ser um agente de viagem ou um cliente da agência de viagem.

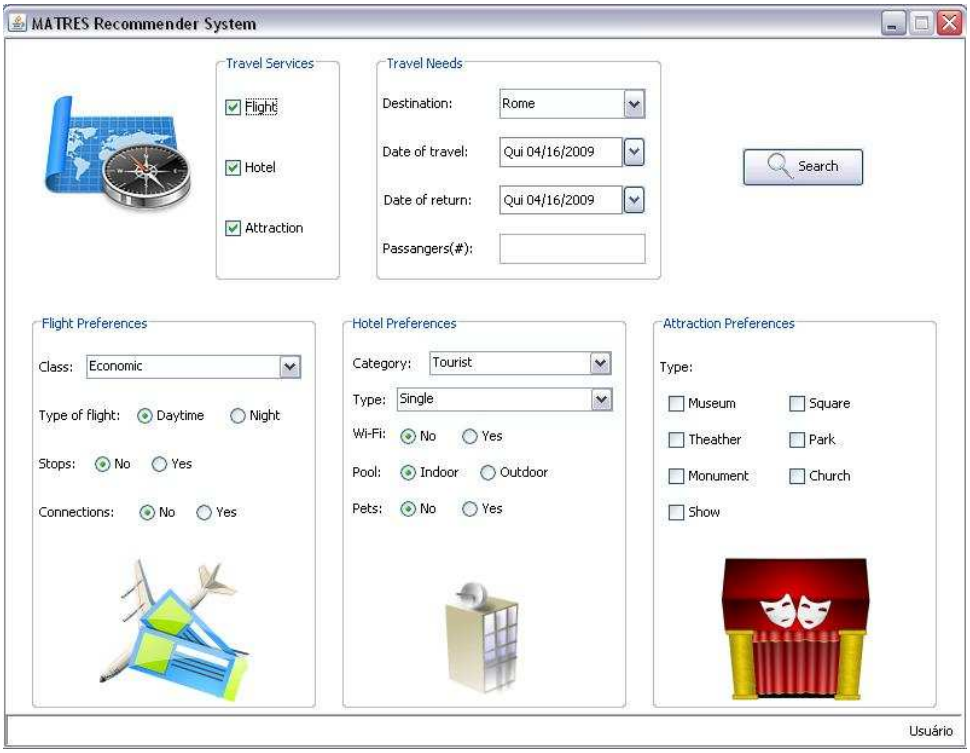

Figura 4.1: Tela principal do sistema

No cenário do turismo foram definidos três tipos possíveis de tarefas: $\mathcal{K} = \{t_1, t_2, t_3\}$, que correspondem aos serviços de viagem onde t_1 representa os voos, t_2 representa as acomodações e t_3 representa as atrações. A partir do conjunto $\mathcal{K}$ de tipos de tarefas, o modelo das preferências do usuário foi definido da seguinte forma:

$U = \{classe\_de\_voo, tipo\_do\_voo, escalas, conexoes, categoria\_hotel$ $wifi, tipo\_do\_apto, piscina, animais, tipo\_de\_atracao\}$. Cada atributo possui seu conjunto $\mathcal{O}$ de opções disponíveis, onde:

$\mathcal{O}_{i_1}$ = {econômica, business, first},
$\mathcal{O}_{i_2}$ = {diurno, noturno},
$\mathcal{O}_{i_3}$ = {sim, nao},
$\mathcal{O}_{i_4}$ = {sim, nao},
$\mathcal{O}_{i_5}$ = {turística, luxo, primeira},
$\mathcal{O}_{i_6}$ = {single, duplo, triplo},
$\mathcal{O}_{i_7}$ = {sim, nao},
$\mathcal{O}_{i_8}$ = {interno, externo},
$\mathcal{O}_{i_9}$ = {sim, nao}
$\mathcal{O}_{i_{10}}$ = {museu, teatro, monumento, show, praça, parque, igreja}

O domínio do turismo apresenta alguns tipos de tarefas interdependentes. Por exemplo, t_1 (tarefa de voo) deve ser finalizada antes de t_2 (tarefa de acomodação) e t_3 (tarefa de atração), pois ambas necessitam de informações provenientes de t_1, ou seja, $Pred(t_1) = \emptyset$, $Pred(t_2) = Pred(t_3) = \{t_1\}$.

A figura 4.1 apresenta a tela de interação com o usuário definida para a aplicação exemplo, onde o usuário define suas necessidades em relação à viagem: destino, datas de saída e retorno e número de passageiros; e suas preferências sobre os 3 diferentes tipos de tarefas: voos, hotéis e atrações.

A partir destas preferências indicadas pelo usuário a lista de tarefas L é criada. A figura 4.2 apresenta um exemplo de L gerada após a consulta de um usuário. Ela contém as tarefas geradas, os agentes que escolheram as tarefas e o estado atual de cada tarefa.

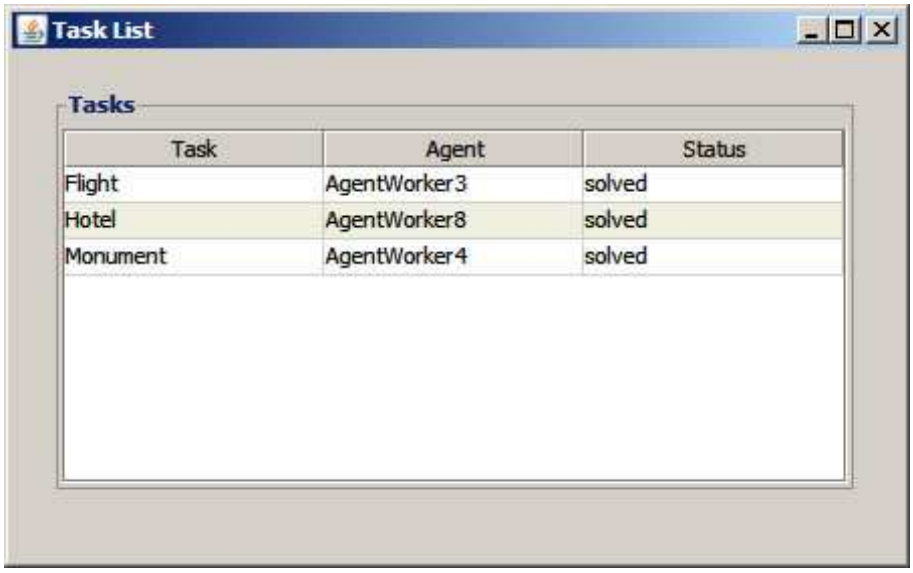

Figura 4.2: Exemplo da lista de tarefas L

4.2.2 Suposições

Conforme visto na seção 3.3.1 os agentes da abordagem MATRES possuem um componente de manutenção de verdade que permite a manipulação de suposições no decorrer do processo de recomendação. Estas suposições são necessárias quando os agentes não possuem informações suficientes para a geração da recomendação. Esta falta de infor-

mação pode ser ocasionada pela demora ou não recebimento do resultado da outra tarefa, pelo não preenchimento de algumas preferências do usuário ou pela falta de conhecimento das preferências implicítas em relação ao usuário.

No cenário do turismo, o conjunto inicial de suposições utilizado na validação da abordagem MATRES foi definido como: $\mathcal{S} = \{s_1, s_2, s_3\}$, onde 3 suposições diferentes podem ser assumidas pelos agentes:

- s_1 - *data de chegada* do voo;
- s_2 - *horário de chegada* do voo;
- s_3 - *data de saída*.

Estas suposições foram necessárias para compor a acomodação do passageiro e as possíveis atrações que ele pode visitar na cidade destino. Todas estas informações são provenientes da tarefa de voo. Isto indica que, os agentes resolvendo as tarefas de *hotel* e *atração* precisam das informações geradas na tarefa de *voo*. Por exemplo, o agente que está realizando a tarefa *atração* na cidade de Roma, precisa saber o horário de chegada do passageiro na cidade, para saber se deve ou não recomendar uma atração neste dia e qual atração melhor se encaixa considerando o horário de chegada.

Além destas suposições relacionadas às tarefas interdependentes, foram permitidas suposições relacionadas às preferências implicítas do usuários utilizadas pelos especialistas no momento da geração da recomendação. Este tipo de suposições foram aplicadas na realização das tarefas de voos e de hotéis. São elas:

- Voos: os agentes utilizaram a suposição que o cliente prefere voos com a cia aérea que oferece programa de milhagem;
- Hoteis: os agentes utilizaram a suposição de que o cliente prefere um hotel com melhor localização ao invés de considerar somente o hotel com o menor preço.

Além da qualidade das recomendações geradas pelos agentes, o tempo de resposta do sistema de recomendação é um item importante e que deve ser considerado. A utilização de suposições é utilizada com o objetivo de evitar um tempo alto de resposta dos agentes. Entretanto, é necessário definir um tempo de espera (ϱ), representado em segundos, em que os agentes podem esperar antes de utilizar as suposições. Eles aguardam pela informação de outro agente pelo tempo ϱ. Se não obtiverem resposta, então iniciam o processo de geração de suposições e assumem esta suposição durante o processo de recomendação. Nos experimentos realizados ϱ= 15.

As suposições são geradas pelos agentes durante o processo de recomendação de acordo com a necessidade de cada agente. Para a validação da abordagem foram utilizados os métodos aplicados para geração de suposições referente aos novos clientes: *MMP* (A opção mais popular na comunidade de agentes) e *MSIM* (casos similares), definidos na seção 3.3.1.

O limiar da similaridade utilizado no *MSIM* foi configurado com 0.5, o que significa que a similaridade aceitável entre os casos e a consulta do usuário deve estar no intervalo [0.5 a 1].

4.2.3 Especialização e confiança

Em uma agência de viagem real é muito comum o agente de viagem se especializar em um tipo de serviço (hotel, vôos, intercâmbios, conferências, etc), com o objetivo de melhor atender às solicitações do cliente, gerando assim uma melhor recomendação. A especialização aumenta o índice de confiabilidade do agente e, consequentemente, melhora a qualidade do serviço prestado por ele.

A abordagem MATRES prevê a especialização dos agentes. Conforme visto na seção 3.3.2, através dos graus de confiança os agentes se especializam em tipos de tarefas, de acordo com seu desempenho na realização das tarefas.

No cenário do turismo validado, foram necessárias as seguintes definições fundamentais em relação aos níveis de confiança:

- O índice de confiabilidade inicial do agente para cada t_n - $(T^t_{a_i,a_i})$ foi definido como 0.5. Isto indica que os agentes não são considerados especialistas no momento de sua criação;
- O grau de confiança inicial do agente nos demais agentes - $(T^t_{a_i,a_j})$ foi definido como 0.5. Isto indica que os agentes não tem opinião sobre os demais agentes na primeira tarefa realizada por eles;
- d foi configurado como 90 dias e consequentemente τ foi configurado como 0.058 para todos tipos de tarefas t_n.

Figura 4.3: Arquivo XML de configuração

```
<object.ConfigurationXML>
  <trustActive>falso</trustActive>
  <weightFlight>0.058</weightFlight>
  <weightHotel>0.058</weightHotel>
  <weightAttraction>0.058</weightAttraction>
  <percTrustedAgents>60</percTrustedAgents>
  <percNewAgents>30</percNewAgents>
  <percOldAgents>10</percOldAgents>
</object.ConfigurationXML>
```

A figura 4.3 apresenta o arquivo XML de configuração de τ para cada tipo de tarefa t_n, onde *weightFlight* representa o peso das avaliações da tarefa de voo, *weightHotel* representa o peso das avaliações da tarefa de hotel e *weightAttraction* representa o peso das avaliações da tarefa de atrações. *TrustActive* é um atributo definido para ser utilizado na etapa de validação, sendo possível ativar ou não a utilização de confiança, para que seja possível a comparação de resultados obtidos pelos agentes com e sem a utilização do componente de confiança.

Outra configuração importante apresentada no arquivo é o percentual utilizado na escolha dos agentes no processo de comunicação. O objetivo da utilização de confiança entre os agentes é evitar a comunicação com agentes que não são confiáveis. Entretanto, é preciso evitar que o agente seja muito rígido e que se comunique apenas com agentes de confiança, pois existe a possibilidade de somente agentes considerados "não confiáveis" estarem disponíveis no momento em que o agente precisa estabelecer uma comunicação

para resolver sua tarefa. Mais crítico até é o fato de que agentes podem considerar outro agente não confiável e depois nunca mais retomarem a confiança neste agente.

Para evitar este tipo de comportamento, é possível configurar-se três tipos de probabilidades que são consideradas pelo agente a_i na escolha do agente para compartilhar informação:

- **Agentes Confiáveis** (*percTrustedAgents*): percentual referente à escolha somente de agentes confiáveis;
- **Agentes Novos** (percNewAgents): percentual referente aos agentes considerados novos, ou seja, agentes que a_i ainda não conhece;
- **Agentes Antigos** (percOldAgents): percentual referente aos agentes que a_i já confiou no passado, mas não confia no momento em que está estabelecendo comunicação com outros agentes.

Os valores apresentados na figura 4.3 foram obtidos após diversos testes. Estes percentuais apresentaram os melhores resultados e por este motivo foram utilizados nos demais experimentos realizados.

4.2.4 Base de conhecimento

A partir da execução da tarefa, o agente busca pela informação e retorna uma recomendação. Este processo de execução da tarefa é armazenado na base de conhecimento dos agentes.

Figura 4.4: Exemplo de caso - a_4, tarefa de *atração*

```
<attraction>Monument</attraction>
<attractionR>
  <source>4</source>
  <status>sim</status>
  <type>monumento</type>
  <name>Coliseo</name>
  <destination>Roma</destination>
  <diasaber>1-7</diasaber>
  <horario>9-17</horario>
</attractionR>
```

No domínio do turismo, a base de conhecimento dos agentes foi modelada de acordo com as preferências definidas em cada tipo de tarefa t_n. Cada agente armazena a identificação do usuário, a consulta definida pelo usuário, a recomendação realizada pelo agente, ou seja, a parte da recomendação gerada pelo agente (voo, hotel ou atração) e a avaliação do usuário para a parte da recomendação gerada pelo agente.

As figuras 4.4, 4.6 e 4.5 apresentam um exemplo de consulta do usuário realizada e as recomendações de voo, hotel e atração apresentadas pelos agentes a_1, a_3 e a_4 para a recomendação solicitada. Entretanto, é importante notar que, estes exemplos não apresentam as avaliações do usuário, somente as recomendações realizadas pelos agentes.

Figura 4.5: Exemplo de caso - a_1, tarefa de *voo*

```
<flight>
  <flightClass>Economic</flightClass>
  <type>Daytime</type>
  <stops>false</stops>
  <connections>false</connections>
</flight>
<flightR>
  <source>1</source>
  <status>sim</status>
  <destination>Roma</destination>
  <home>Guarulhos</home>
  <symbol>Alitalia</symbol>
  <numFlightI>675</numFlightI>
  <typeFlightI>daytime</typeFlightI>
  <departureTimeI>2009-07-22 15:25:45.515 BRT</departureTimeI>
  <arrivalTimeI>2009-07-22 07:25:45.515 BRT</arrivalTimeI>
  <stopsI>false</stopsI>
  <connectionsI>false</connectionsI>
  <numFlightV>674</numFlightV>
  <typeFlightV>daytime</typeFlightV>
  <departureTimeV>2009-07-29 07:20:40.300 BRT</departureTimeV>
  <arrivalTimeV>2009-07-29 17:20:35.315 BRT</arrivalTimeV>
  <stopsV>false</stopsV>
  <connectionsV>false</connectionsV>
  <fClass>Economic</fClass>
  <availability>true</availability>
</flightR>
```

A figura 4.4 apresenta o caso armazenado pelo agente a_4, que escolheu a tarefa de atração na cidade de Roma e retornou o *Coliseo*. Na figura 4.5 pode-se ver o caso armazenado na base do agente a_1 gerado a partir da solicitação feita pelo usuário: voos para Roma, em classe econômica, voo diurno, sem escalas e sem conexões. O agente retornou voos no trecho Guarulhos/Roma/Guarulhos, sem escalas e conexões e disponíveis em classe econômica.

Por fim, a figura 4.6 apresenta a recomendação de hotel gerada pelo agente a_3 para a mesma solicitação do usuário, considerando as preferências de apto duplo, com internet sem fio, piscina interna e hotel que não aceite animais.

4.3 Validação e experimentos

Conforme apresentado por (KONSTAN; RIEDL, 1999), existem duas formas de validar um sistema de recomendação:

- ***Online***: validação que deve ser conduzida enquanto o sistema de recomendação está sendo utilizado por usuários. Os dados podem ser coletados através de observações feitas, questionários ou entrevistas feitas aos usuários. Os experimentos *online* podem ser utilizados também para refinar os parâmetros do sistema e deter-

Figura 4.6: Exemplo de caso - a_3, tarefa de *hotel*

```
<hotel>
  <category>Luxury</category>
  <type>Twin</type>
  <wifi>true</wifi>
  <pool>Indoor</pool>
  <pet>false</pet>
</hotel>
<hotelR>
  <source>3</source>
  <status>sim</status>
  <name>Hotel Reppublica</name>
  <destination>Roma</destination>
  <category>Turistic</category>
  <type>Twin</type>
  <wifi>true</wifi>
  <pool>false</pool>
  <pets>false</pets>
</hotelR>
```

minar o efeito da interação no desempenho e na satisfação do usuário. Um ponto fraco deste tipo de validação é a necessidade de usuários dispostos a participar do experimento (através da utilização do sistema) a avaliar os resultados apresentados pelo sistema. Entretanto, como ponto forte, esta técnica permite avaliar o desempenho do sistema em termos de satisfação do usuário.

- ***Offline***: validação que é conduzida através da utilização de dados que foram coletados previamente durante a utilização do sistema. Neste tipo de validação, os usuários não precisam estar presentes ou disponíveis ao longo da validação. A ténica *offline* não depende da presença de usuários. Os experimentos *offline* podem ser conduzidos diversas vezes, sem a preocupação com o usuário. Entretanto, não é possível coletar novos dados do usuário e nem solicitar algum esclarecimento sobre uma avaliação feita por ele.

A abordagem MATRES foi validada através da forma *offline*, onde foram utilizados dados provenientes de uma agência de viagem. A realização da etapa de aquisição de conhecimento com um especialista na área de turismo foi necessária para obtenção dos casos a serem utilizados na validação.

4.3.1 Aquisição de conhecimento

Na etapa de aquisição de conhecimento, o especialista na área de turismo auxiliou no processo de validação da abordagem fornecendo os casos. Os agentes de viagem envolvidos nesta etapa possuem mais de 15 anos de experiência na área e trabalham em uma agência de viagem de médio porte que existe desde 1989. Esta agência possui dois setores importantes: *coorporativo* (que gerencia viagens de empresas) e *turismo* (atendimento de clientes que buscam recomendações de pacotes turísticos). A validação da abordagem concentrou-se no item *turismo*.

Tabela 4.1: Exemplos de casos adquiridos da agência de viagem

Casos	Destino	Paxs	Classe	Tipo	Categegoria	Apto	Atração
1	Lisboa	2	econômica	diurno	turística	duplo	monumento
2	Paris	2	econômica	noturno	turística	duplo	museu
3	Punta Cana	2	econômica	noturno	luxo	duplo	show
4	Milão	1	econômica	noturno	turística	single	monumento
5	Lisboa	2	econômica	noturno	turística	duplo	monumento
6	Londres	2	econômica	noturno	turística	duplo	museu
7	Cancun	2	econômica	noturno	luxo	duplo	show
8	Miami	2	econômica	noturno	turística	duplo	show
9	Amsterdam	1	econômica	noturno	turística	single	museu
10	Madri	1	econômica	noturno	turística	single	monumento
11	New York	3	econômica	diurno	turística	triplo	show
12	Roma	2	econômica	noturno	luxo	duplo	monumento
13	Porto	1	econômica	diurno	turística	single	museu
14	Valencia	2	econômica	noturno	turística	duplo	museu
15	Dublin	3	econômica	noturno	turística	triplo	show
16	Edinburgo	2	econômica	diurno	turística	duplo	monumento
17	Barcelona	3	econômica	noturno	turística	triplo	museu
18	Orlando	3	econômica	noturno	turística	triplo	parque
19	Rio de Janeiro	2	econômica	noturno	turística	duplo	monumento
20	Madri	2	econômica	noturno	turística	duplo	museu

A agência forneceu 380 casos (pacotes recomendados no período de janeiro a outubro de 2007) que foram utilizados para inicializar as bases de conhecimento dos agentes. Estes casos continham a consulta do cliente, a recomendação apresentada a ele e a avaliação do cliente.

Cada caso era composto de três tipos de serviços (voos, acomodação e atração) e representavam 1140 tarefas. Os casos foram extraídos do sistema de *backoffice* da agência e armazenados em uma planilha. A partir desta planilha, os dados foram convertidos para o formato XML definido.

A tabela 4.1 apresenta exemplos de casos que foram fornecidos pelo especialista. Na tarefa de atração, apenas uma escolha de atração foi inserida no caso, para facilitar a validação deste tipo de tarefa.

Foram obtidos 1259 casos para serem utilizados como novas consultas de usuários nos experimentos. Estes casos representam pacotes turísticos (compostos por passagem aérea, hospedagem e atração) comprados pelos clientes no período de Novembro/2007 a Março de 2010. Para validar a abordagem apresentada nesta tese, estes casos foram utilizados como novas consultas a serem realizadas pelos agentes.

A vantagem da utilização de casos reais como novas consultas é a possibilidade de avaliar os resultados apresentados pelos agentes em problemas reais e utilizar as soluções dos casos como avaliações corretas para as recomendações.

4.3.2 Avaliações das recomendações

Conforme apresentado na seção 3.3.2.2, os graus de confiança e índices de confiabilidade dos agentes são atualizados considerando as avaliações do usuário das reco-

mendações apresentadas. Por este motivo, as recomendações geradas pelos agentes na abordagem apresentada foram validadas de acordo com o número de atributos avaliados como "gostei"de cada tipo de tarefa.

Entretanto, como casos da agência foram utilizados como novas consultas, não foi possível que os próprios clientes avaliassem os resultados apresentados pelos agentes. Ao invés disto, a avaliação foi realizada da seguinte forma:

1. Para cada caso, os atributos recomendados pelos agentes foram comparados com os resultados originais recomendados pelo agente de viagem (especialista);

   (a) Valores iguais foram considerados como avaliação *positiva*. Conforme definido na seção 3.3.2.1, os valores recomendados são avaliados com notas “gostei” (positiva) ou “não gostei” (negativa);

   (b) Os valores recomendados pelos agentes diferentes dos valores recomendados pelo especialista precisaram ser analisados por um segundo especialista. A escala de *thurstone* foi novamente utilizada, onde o especialista atribuiu uma nota positiva ou negativa para o atributo gerado pelo agente. Esta nota é 1 (se nota = “gostei“) ou 0 (se nota = ”não gostei“).

2. Avaliação máxima em cada tipo de tarefa representa a aprovação de todos atributos recomendados: (4, 5 e 1).

4.4 Calibragem do número de agentes

Considerando-se que os sistemas de recomendação são utilizados na Internet, decidiu-se realizar uma calibragem do número de agentes necessários na comunidade com base nos testes de carga e de estresse apresentados por (MOLINARI, 2003). Os testes de carga são responsáveis por testar o desempenho do sistema levando-se em conta uma carga de usuários simultâneos reais e os testes de estresse são responsáveis por testar o desempenho do sistema considerando o número máximo de usuários simultâneos que pode suportar.

A calibragem foi realizada com o objetivo de verificar o número de agentes necessários na comunidade para garantir um tempo de resposta adequado ao sistema de recomendação e a qualidade das recomendações geradas pelos agentes (representando o *teste de carga*). Além disto, utilizou-se um grande número de usuários solicitando recomendações, o que representou o *teste de estresse*.

Para isto, a calibragem foi realizada em diferentes testes, com quatro diferentes números de agentes disponíveis para realização de tarefas:

- Teste 1: 3 agentes;
- Teste 2: 5 agentes;
- Teste 3: 10 agentes;
- Teste 4: 15 agentes.

Os 380 casos adquiridos na agência de viagem (conforme apresentado na seção 4.3.1) foram distribuídos aleatoreamente entre as bases de conhecimento dos agentes. A troca de informações e os componentes de suposições e de confiança dos agentes foram habilitados, permitindo que os agentes compartilhassem informações quando necessário,

utilizassem seus graus de confiança para compartilhar informações e utilizassem suposições ao longo dos processos de recomendações. Este foi considerado o pior cenário para os agentes, pois eles trocam informações, utilizam suposições e atualizam seus graus de confiança, demandando muito processamento. O tempo de espera para utilização de suposições foi configurado em 15s ($\varrho = 15$).

Em seguida, os 1259 casos adquiridos da agência de viagem, que representam 3777 tarefas, foram executados como novas consultas em cada teste. Os testes foram realizados em um computador Intel Core I5-430M, com 4GB RAM.

A tabela 4.2 apresenta os resultados obtidos pelos agentes nos quatro testes realizados. A primeira linha apresenta as médias obtidas no teste com 3 agentes. Percebe-se que a média de tempo de processamento da tarefa de voo ficou em 85 segundos (o que representa 1 min e 47 segundos). A média de tempo da tarefa hotel foi mais alta, pois esta é uma tarefa dependente da tarefa de voo. Por fim, a tarefa de atração teve o tempo menor, pois foi configurado que os agentes buscariam apenas uma tarefa do tipo escolhido pelo usuário (conforme visto na seção 4.3.2).

A segunda linha apresenta as médias após 5 agentes executarem as tarefas. Analisando os resultados, percebe-se que a média de tempo de processamento diminuiu, ou seja, o fato de 5 agentes resolverem tarefas (ao invés de apenas 3) melhora o desempenho de todos. Entretanto, as médias das avaliações das recomendações geradas são muito parecidas com os resultados obtidos no primeiro teste. Este resultado deve-se ao fato de que, apesar do sistema ter 2 agentes a mais, as bases de conhecimento dos agentes não foram incrementadas. Ou seja, os mesmos casos utilizados no *teste 1* foram utilizados no *teste 2*.

A terceira linha da tabela apresenta as médias obtidas no **teste 3** onde 10 agentes foram utilizados. Neste teste pode-se ver que o tempo de processamento diminuiu e as médias da qualidade das recomendações geradas melhoraram. Apesar das bases de conhecimento continuarem do mesmo tamanho, o sistema de recomendação possui mais agentes, o que possibilita mais troca de informações entre eles. Neste teste, foi possível acompanhar a utilização do componente de confiança entre os agentes, pois aconteceram mais interações entre eles.

A última linha da tabela apresenta as médias obtidas através do **teste 4** onde 15 agentes foram utilizados. Os resultados mostram que, apesar do sistema possuir mais agentes, o tempo de processamento não melhorou de forma significativa. Comparando os resultados dos *teste 3* e *teste 4* pode-se ver que a média do tempo de processamento dos 10 agentes foi 63 segundos, contra 71 segundos dos 15 agentes. Analisando os *logs* dos agentes, percebeu-se que neste teste os agentes precisaram compartilhar mais informações e suposições entre eles, aumentando assim o tempo de processamento. Quanto à qualidade das recomendações, nas tarefas de voo e hotel, os agentes no *teste 4* foram superados pelos agentes do *teste 3*. Na tarefa de atração entretanto, os agentes do *teste 4* tiveram melhor resultado.

Tabela 4.2: Médias obtidas na calibragem do número de agentes

Testes	**Nro agentes**	**Tempo de processamento (s)**				**Avaliações positivas**		
		Aéreo	**Hotel**	**Atração**	**Soma**	**Aéreo**	**Hotel**	**Atração**
Teste 1	3	85	91	39	215	3,0	4,1	0,7
Teste 2	5	70	87	30	187	3,1	4,2	0,8
Teste 3	10	42	63	24	129	3,9	4,6	0,8
Teste 4	15	40	71	25	136	3,7	4,4	0,7

A partir dos resultados apresentados na tabela 4.2, optou-se pela utilização de 10 agentes no sistema de recomendação utilizado nos experimentos, pois este teste apresentou a melhor relação entre os resultados de tempo e qualidade dos agentes.

4.5 Resultados obtidos

Diferentes experimentos foram realizados para validar os diferentes componentes propostos na abordagem MATRES. Entretanto, todos experimentos realizados tiveram como objetivo principal a verificação da qualidade das recomendações apresentadas pelos agentes.

Em sistemas de recomendação multiagente, os usuários esperam sempre receber recomendações coerentes com suas preferências. Assume-se que os usuários não aprovam o sistema quando ele apresenta recomendações que ignoram totalmente suas preferências.

Outro ponto verificado nos experimentos foi o desempenho do sistema, pois o tempo de resposta é um item importante em sistemas de recomendação.

A seção 4.5.1 apresenta o experimento realizado para validar a utilização do componente de manutenção de verdade na geração e manipulação das suposições. As avaliações das recomendações geradas pelo especialista foram comparadas com as recomendações geradas pelos agentes, que utilizaram os métodos de geração e manipulação de suposições.

Na seção 4.5.2 é apresentado o experimento realizado para validar a capacidade que os agentes possuem de compartilhar informações entre eles e o componente de confiança utilizado neste processo. Neste experimento, os agentes executaram as tarefas utilizando o componente de confiança e depois, não utilizando seu componente de confiança. As avaliações das recomendações geradas por eles foram comparadas com as avaliações das recomendações geradas pelo especialista.

Por fim, a seção 4.5.3 apresenta o experimento onde foi verificado o número médio de agentes envolvidos no processo de busca de informação para realização de uma tarefa. Neste experimento os agentes realizaram suas tarefas de duas formas: com e sem componente de confiança e os resultados foram comparados com os resultados do especialista.

Todos experimentos que serão apresentados foram validados também através de testes estatísticos, onde buscou-se confirmar que os valores obtidos são estatisticamente significantes e realmente apresentam resultados melhores do que os resultados do especialista.

4.5.1 Experimento 1: suposições

Este primeiro experimento teve o objetivo de verificar a assertividade das recomendações geradas pelos agentes através da utilização de suposições. Para gerar as suposições durante os ciclos de recomendação, os agentes utilizaram dois métodos diferentes definidos na seção 3.3.1: *MMP* e *MSIM*.

A variante *MMP* corresponde ao método que utiliza a opção mais popular na comunidade de agentes. Os agentes geram suposições ao longo do processo de recomendação, através da opção mais popular nas bases de conhecimento de todos agentes. *MSIM* corresponde ao método que utiliza casos similares, onde os agentes geram suposições através do caso mais parecido encontrado nas bases de conhecimento dos agentes.

Estes dois métodos foram escolhidos para validação das recomendações geradas para novos usuários. Segundo o especialista, assumir suposições para clientes novos é mais difícil do que assumir suposições para clientes conhecidos, pois os clientes conhecidos possuem um histórico na agência que serve de base para o agente de viagem analisar e

assumir informações.

O teste estatístico *Kruskal-Wallis* foi utilizado para analisar a variância dos resultados obtidos pelos dois métodos aplicados e os resultados originais do especialista, considerando cada tipo de tarefa (voo, hotel e atração). Optou-se por este teste pois através dele foi possível comparar os resultados (considerados dados categóricos) dos três métodos utilizados (*MMP*, *MSIM* e os resultados obtidos do especialista (*ESP*)). O teste possui os seguintes parâmetros:

- Hipótese nula (H_0): é uma hipótese presumida como verdadeira até que provas estatísticas sob a forma de testes de hipóteses indiquem o contrário. Neste primeiro experimento, H_0 foi definida como "A qualidade das recomendações em os três métodos são iguais". Isto indica que o teste será executado para rejeitar esta hipótese, ou seja, para provar que a qualidade das avaliações dos resultados obtidos através dos métodos propostos é melhor do que a qualidade obtida através do especialista;

- Nível de significância (α): chamado de nível de significância do teste e indica a probabilidade de rejeitar a hiptótese nula quando esta é verdadeira. Optou-se pela definição de $\alpha = 0,05$, por tratar-se do nível de significância mais utilizado na literatura;

- p (conhecido como *p-value*): corresponde ao menor nível de significância que pode ser assumido para rejeitar a hipótese nula. Há significância estatística quando *p-value* é menor que o nível de significância (α) adotado, ou seja, a hipótese nula é rejeitada quando $p \leq \alpha$. O valor de p é obtido após a execução do teste.

Após a execução das tarefas pelos agentes em ambos os métodos foram obtidos os resultados apresentados na tabela 4.3. A quarta coluna refere-se a média dos valores positivos, seguida pela coluna que apresenta o desvio padrão e da coluna que apresenta o erro padrão obtido em cada método.

É importante lembrar que, conforme apresentado na seção 4.3.2, os valores máximos em cada tipo de serviço seriam 4, 5 e 1. Percebe-se que a melhor média no tipo de tarefa *voo* foi obtida pelos agentes quando utilizaram o *MSIM* - (**3,47**). A média obtida pelos agentes através do *MMP* ficou abaixo da média dos resultados do especialista.

Na tarefa *hotel*, o *MSIM* obteve o melhor resultado (**4,70**), quase chegando ao valor máximo de avaliação (**5,00**). Aqui, percebe-se que a utilização de suposições (*MMP* ou *MSIM*) contribuiu para geração de boas recomendações, dado que suas médias (**4,04** e **4,70**) foram superiores à média obtida pelo especialista.

Na tarefa *atração* o *MSIM* também se mostrou superior, tendo uma média de **0,90**, contra **0,70** do especialista e **0,77** do *MMP*.

Em relação a tarefa voo, observa-se que o *ranking* médio das avaliações dos três métodos é significativamente diferente (*p-value*<0). Isto indica que as diferenças encontradas nos resultados dos três métodos são estatisticamente significante e são grandes o suficiente para não serem atribuídas ao acaso. O *MSIM* é significativamente superior ao nível de 5% do *MMP* e do *ESP*. Conclui-se que a diferença entre o *MSIM* e os demais é bastante significativa pois *p-value* resultou em um valor muito abaixo do nível de significância definido ($\alpha = 0,05$).

Para a tarefa *hotel*, obteve-se *p-value*=0,0036, ou seja, o *ranking* médio das avaliações dos três métodos também é significativamente diferente. Os três métodos são diferentes a 5% de nível de significância, porém o *MSIM* é superior aos outros dois métodos, indicando que teve os melhores resultados nas recomendações.

Tabela 4.3: Comparação dos resultados obtidos: média, desvio padrão e erro padrão das avaliações obtidas em cada método.

Tipo de Tarefa	Experimento	Tarefas	Média	Desvio padrão	Erro padrão
Voo	ESP	1259	3,380	0,621	0,018
	MMP	1259	3,310	0,659	0,019
	MSIM	1259	3,470	0,566	0,016
	Total	**3777**	**3,380**	**0,619**	**0,010**
Hotel	ESP	1259	3,810	0,592	0,017
	MMP	1259	4,040	0,636	0,018
	MSIM	1259	4,700	0,581	0,016
	Total	**3777**	**4,180**	**0,712**	**0,012**
Atração	ESP	1259	0,700	0,460	0,013
	MMP	1259	0,770	0,419	0,012
	MSIM	1259	0,900	0,300	0,008
	Total	**3777**	**0,790**	**0,413**	**0,007**

Por fim, na tarefa *atração* o *ranking* médio das avaliações dos três métodos também resultou significativamente diferente (*p-value* 0.0006). Isto quer dizer que há diferença entre os três métodos. Entretanto, *MMP* e *ESP* não diferem significativamente a 5%. A resposta média do *MSIM* foi superior às respostas médias do *MMP* e dos resultados do especialista.

O gráfico 4.7 apresenta a comparação das médias das avaliações nos três tipos de tarefa, para cada método utilizado na geração das recomendações e nos resultados obtidos pelo especialista. Percebe-se que o *MSIM* é o que resulta em melhores recomendações em todos tipos de tarefa. Porém, os melhores resultados foram obtidos nas tarefas de hotel e atração. Isto porque na tarefa de voo só são utilizadas suposições simples, ou seja, aquelas relacionadas ao não preenchimento de algumas preferências do usuário.

Pode-se concluir destes resultados obtidos que a utilização de suposições contribui para a qualidade das recomendações geradas pelos agentes. Conclui-se também que o *MSIM*, que utiliza casos similares na geração das suposições, é mais eficiente do que o *MMP*, ou seja, a utilização de informações de clientes parecidos com o cliente que está sendo analisado é mais eficiente do que a utilização da opção mais popular entre todos clientes de uma agência. Pode-se dizer que a utilização de casos similares personaliza mais as recomendações geradas pelos agentes e, por este motivo, são mais aceitas pelos usuários.

4.5.2 Experimento 2: troca de informações e confiança

Este experimento teve o objetivo de verificar a assertividade das recomendações geradas pelos agentes quando estes utilizam o componente de confiança para escolher com quem se comunicar. Entende-se por comunicação o ato de solicitar a outro agente informação necessária para executar uma tarefa (conforme apresentado na seção 3.4.2).

Para isto, algumas configurações foram necessárias:

- O componente de suposições foi desativado nos agentes, ou seja, os agentes compartilharam informações entre eles ao invés de utilizar suposições;
- A forma de busca de informação *LocalKBSearch* foi desativada nos agentes para forçar que os agentes compartilhassem informações entre eles, ao invés de buscar

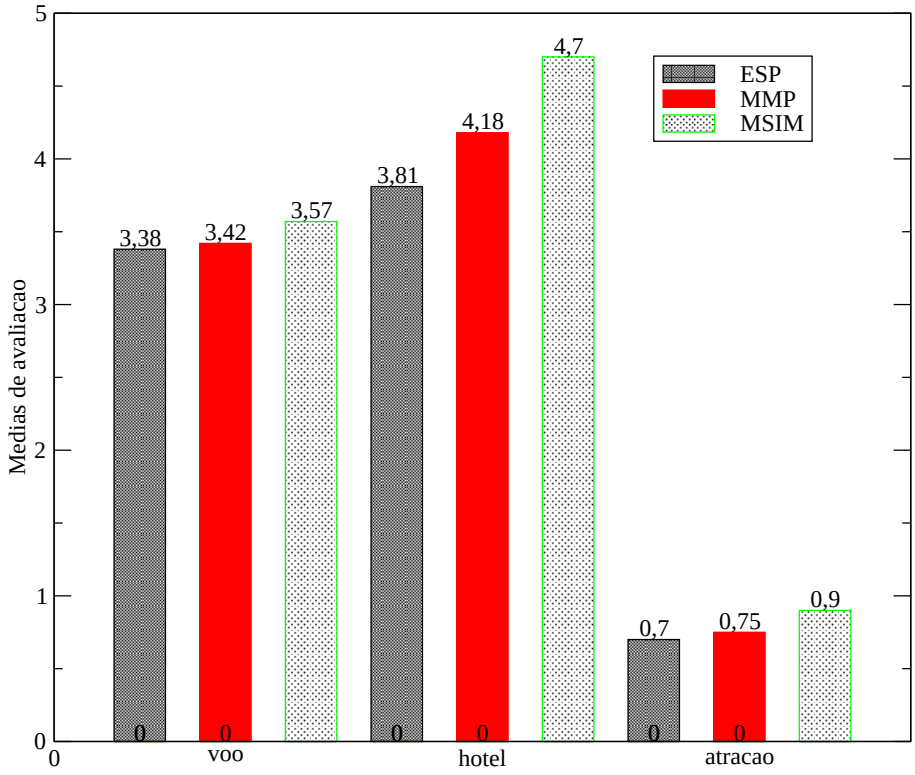

Figura 4.7: Valores médios das avaliações de cada tipo de tarefa: voo, hotel e atração, em cada método de recomendação utilizado (ESP, MMP e MSIM).

pelas informações em suas bases de conhecimento.

As 1259 consultas foram realizadas pelos agentes utilizando dois métodos diferentes:

1. *WTrust*: os agentes consideraram seus graus de confiança durante os processos de recomendação. Isto indica que os agentes escolheram com quem se comunicar para obter informação necessária para a recomendação;

2. *NTrust*: os agentes não consideraram seus graus de confiança durante as recomendações. A comunicação com outros agentes foi realizada de forma aleatória.

A tabela 4.4 apresenta os resultados obtidos em cada tipo de tarefa após execução das 1259 tarefas com ambos os métodos, bem como o resultado da recomendação original realizada pelo especialista. Na tarefa *voo*, os agentes tiveram médias superiores à média das avaliações do especialista.

Na tarefa *hotel* percebe-se que a média das avaliações dos agentes sem a utilização do componente de confiança *NTrust* (**4,18**) foi pouco superior à média das avaliações do especialista (**3,81**). Entretanto, a média das avaliações dos agentes com a utilização do componente de confiança *WTrust* (**4,70**) foi superior às médias das avaliações do especialista (**3,81**) e dos agentes sem a utilização do componente de confiança *NTrust* (**4,18**).

Na tarefa *atração*, os agentes com a utilização do componente de confiança *WTrust* também tiveram as médias das avaliações superiores.

O teste estatístico de *Kruskal-Wallis* também foi utilizado para a comparação dos resultados. A hipótese nula (H_0) foi definida como “A qualidade das recomendações nos três métodos são iguais”.

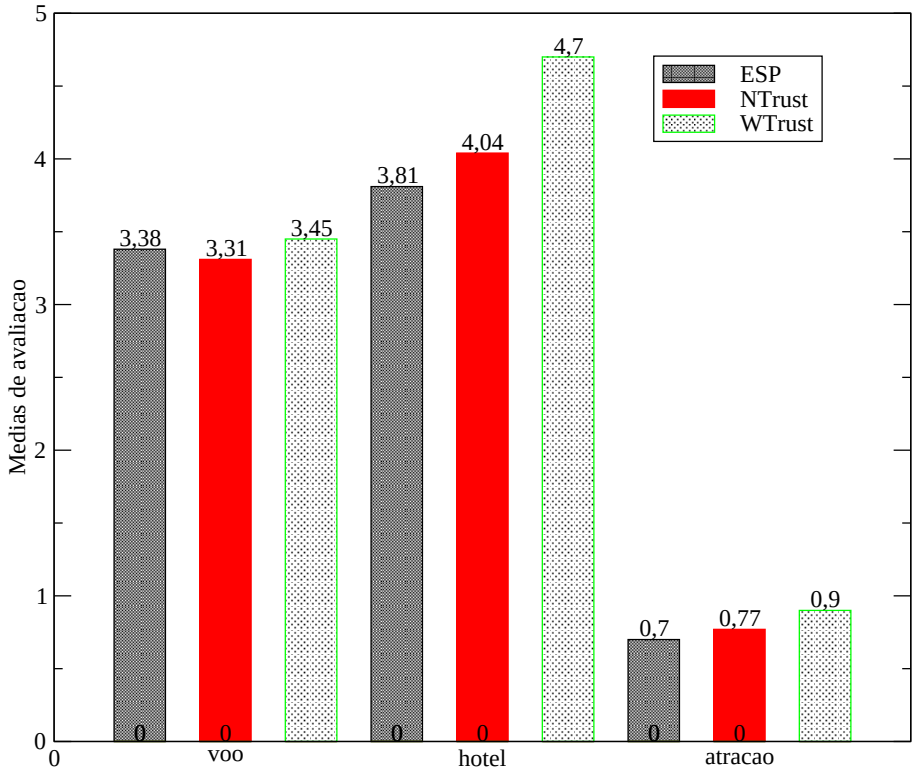

Figura 4.8: Valores médios das avaliações de cada tipo de tarefa: voo, hotel e atração, em cada método de recomendação utilizado (ESP, NTrust e WTrust).

Na tarefa de *voo*, o teste estatístico mostrou que os valores médios dos três experimentos são significativamente diferentes (*p-value*=0). Isto quer dizer que existe diferença entre os resultados do especialista, NTrust1 e WTrust. O gráfico 4.8 mostra que no cenário *NTrust* os agentes tiveram resultados inferiores aos resultados do especialista. Entretanto, esta diferença não é estatisticamente significante. O resultado de *WTrust* é estatisticamente significativo e melhor do que os resultados obtidos pelos agentes no cenário de *NTrust*.

O *ranking* médio das avaliações nos três métodos é significativamente diferente para a tarefa *hotel* (*p-value*<0). Isto significa que existe diferença entre os três métodos: especialista, *NTrust* e *WTrust*. Todos os métodos são diferentes a 5% de nível de significância e o método *WTrust* é superior ao método *NTrust* e aos resultados obtidos do especialista.

Analisando o gráfico 4.8, conclui-se que a utilização do mecanismo de confiança contribui para a geração de recomendações melhores, pois a média de avaliações do método *WTrust* (4,70) foi superior a média dos agentes sem o mecanismo de confiança (4,04) e a média do próprio especialista (3,81).

Por fim, na tarefa de *atração* o teste estatístico *Kruskal-Wallis* mostrou que os valores médios também são significativamente diferentes (*p-value*=0) e que o método *WTrust* é superior.

4.5.3 Experimento 3: comunicação

Este experimento teve o objetivo de verificar o número médio de agentes envolvidos em um processo de compartilhamento de informação. Conforme já mencionado, o tempo de resposta de um sistema de recomendação é uma característica importante que deve ser considerada no desenvolvimento destes sistemas. O sistema pode dar respostas com

Tabela 4.4: Comparação dos resultados obtidos: média, desvio padrão e erro padrão das avaliações obtidas em cada método (ESP, NTrust e WTrust).

Tipo de Tarefa	Cenário	Tarefas	Média	Desvio padrão	Erro padrão
Voo	ESP	1259	3,380	0,621	0,018
	NTrust	1259	3,420	0,618	0,017
	WTrust	1259	3,570	0,617	0,017
	Total	**3777**	**3,460**	**0,624**	**0,010**
Hotel	ESP	1259	3,810	0,592	0,017
	NTrust	1259	4,180	0,601	0,017
	WTrust	1259	4,700	0,528	0,017
	Total	**3777**	**4,230**	**0,681**	**0,015**
Atração	ESP	1259	0,700	0,460	0,013
	NTrust	1259	0,750	0,436	0,012
	WTrust	1259	0,900	0,299	0,008
	Total	**3777**	**0,780**	**0,413**	**0,007**

qualidade, entretanto se tiver um tempo de resposta alto, não terá validade para o usuário.

Considerando este fato, este terceiro experimento teve o intuito de validar a influência do componente de confiança na redução do número de troca de mensagens entre os agentes.

Para tal, novamente foram utilizados dois métodos: *NTrust* e *WTrust*. Na falta de informações para gerar as recomendações, os agentes compartilharam informações e não utilizaram o componente de suposições. A cada tarefa resolvida foram registrados o número de agentes envolvidos na recomendação.

No método *NTrust*, os agentes não consideraram seus graus de confiança no momento de solicitar informação aos demais agentes. A solicitação foi realizada através de *broadcast* a todos agentes disponíveis.

Já no método *WTrust* os agentes somente se comunicaram com agentes confiáveis. Os agentes enviaram solicitação para os agentes confiáveis disponíveis no momento. Conforme apresentado na seção 3.4.2, não foi definido número máximo de agentes confiáveis, os agentes tentam buscar a informação dos agentes considerados confiáveis disponíveis no momento.

Este experimento considerou, além da qualidade das recomendações, o número de agentes envolvidos em cada execução de tarefa. Por exemplo, o agente a_1 precisou solicitar informação para outros agentes durante a execução de sua tarefa. Ele se comunicou com 3 agentes até conseguir a informação que estava precisando. Este valor (3 agentes) foi considerado para avaliar a eficiência do método *WTrust*, comparando-o com o método *NTrust*.

A tabela 4.5 apresenta a média de agentes envolvidos na execução de cada tarefa. Percebe-se que no método *WTrust* a média de agentes envolvidos na recomendação é **1,90** contra **5,69** do método *NTrust*. Isto significa que no método *WTrust* foi necessário acessar o segundo agente mais confiável para resolver uma recomendação.

Esperava-se que o primeiro agente confiável tivesse a informação para a recomendação, porém os experimentos mostraram que na maioria das vezes foi necessário acessar o segundo agente mais confiável para encontrar a informação necessária para a recomendação. Analisando o *log* dos agentes. percebeu-se que o agente considerado o mais confiável tinha data de atualização do grau de confiança inferior ao segundo agente mais confiável.

Para evitar este tipo de problema, foi necessário um pequeno ajuste no momento da escolha dos agentes confiáveis. Primeiro, o agente a_i verifica os agentes em que mais confia, ordenando este conjunto pelo grau de confiança. Deste conjunto, o agente a_i escolhe o agente a_j que tem a data de interação entre eles mais recente.

Tabela 4.5: Média do número de agentes envolvidos na resolução de cada tarefa nos métodos *NTrust* e *WTrust*

Cenário	Tarefas	Média do nro de agentes
NTrust	1259	5,69
WTrust	1259	1,90

As tarefas foram executadas novamente e a tabela 4.6 apresenta os resultados finais obtidos. A média do número de agentes no método *NTrust* continuou em 5 agentes (5,52). A média do número de agentes no método *WTrust* todavia melhorou, diminuindo para 1,14.

Apesar de eficiente, este resultado (1,14) apontou para uma possível alteração necessária na atualização do grau de confiança. Quando o agente não tem informação para compartilhar, ele deveria ter seu grau de confiança diminuído, pois não foi eficiente no auxílio ao agente. Este item será detalhado nos trabalhos futuros na seção 5.2.

Tabela 4.6: Média do número de agentes envolvidos na resolução de cada tarefa nos métodos *NTrust* e *WTrust* - após ajuste da atualização dos graus de confiança.

Cenário	Tarefas	Média do nro de agentes
NTrust	1259	5,52
WTrust	1259	1,14

4.6 Avaliação dos resultados

O presente capítulo discutiu a aplicação e validação da abordagem no domínio do turismo. Domínios com estas características são o escopo de aplicação da abordagem MATRES.

Os experimentos apresentados neste capítulo mostraram que a abordagem MATRES é eficaz e contribui para a melhora da qualidade das recomendações geradas pelos agentes. A análise detalhada dos componentes propostos na abordagem MATRES mostrou que os agentes geram maior número de recomendações positivas quando assumem suposições, compartilham informações entre agentes de confiança ao longo do processo de recomendação e quando possuem bases de conhecimento consistentes.

Uma base de casos reais de uma agência de viagem foi utilizada na validação e os seguintes resultados foram obtidos:

- Quanto à utilização das suposições, conforme apresentado no experimento 4.5.1, o *MSIM* de geração de suposições, que busca casos similares, mostrou maior eficácia do que o *MMP* que utiliza a opção mais popular na comunidade de agentes, visto que ele teve uma média alta de avaliações;

- Quanto à troca de informações entre os agentes, o experimento 4.5.2 mostrou que a utilização dos graus de confiança resultou em um maior número de recomendações

aprovadas pelos usuários apresentadas pelos agentes. Percebeu-se que, à medida que os agentes resolvem tarefas ou compartilham informações e são bem avaliados nas duas situações, eles são capazes de melhorar as recomendações apresentadas. As avaliações possuem um período de validade, o que torna o processo mais real, pois as recomendações perdem seu efeito ao longo do tempo;

- Pode-se concluir do experimento 4.5.2 também que o fato dos agentes se especializarem melhora os resultados das recomendações apresentadas por eles ao longo do tempo;

- Quanto à comunicação entre os agentes, o experimento 4.5.3 mostrou que o componente de confiança torna o processo de comunicação mais eficiente, pois ele reduz o número de troca de mensagens entre os agentes. Os agentes trocam informações com menos agentes quando consideram seus graus de confiança nos demais agentes para escolher com quem se comunicar;

- Quanto à escalabilidade, o experimento inicial realizado na seção 4.3.1 mostrou que a abordagem MATRES permite a expansão do número de agentes e o aumento da quantidade de casos sem degeneração rápida do tempo de resposta dos agentes.

5 CONCLUSÕES

Esta tese teve como objetivo principal provar a hipótese de que "uma abordagem multiagente de recomendação baseada em suposições e com um método de confiança aplicada em domínios dinâmicos contribui para a geração de recomendações mais assertivas". Para isto, este trabalho apresentou a abordagem MATRES, uma abordagem baseada em suposições e confiança dos agentes para ser aplicada no processo de recomendação multiagente em domínios onde é necessário conhecimento distribuído e específico.

Em diversos domínios, a recomendação não é uma tarefa trivial, pois é necessário decompor a solicitação do usuário em diversas partes e, para solucionar cada parte, é necessário conhecimento especialista. Os sistemas de recomendação multiagente são aplicados nestes tipos de domínios. Entretanto, fontes de informações podem estar distribuídas, bem como o conhecimento especialista necessário para gerar uma recomendação.

Conforme visto no capítulo 2, as abordagens de recomendação baseadas na arquitetura multiagente, apesar de terem diversos agentes trabalhando na geração de uma recomendação, não lidam com alguns problemas encontrados, como por exemplo: a) os agentes não utilizam a experiência adquirida ao longo de sua existência em novas recomendações; b) os agentes não sabem lidar com a dependência entre as recomendações parciais resolvidas no processo de geração de uma recomendação; c) um único agente pode não possuir o conhecimento suficiente para gerar uma recomendação; entre outros.

A abordagem MATRES explora a habilidade de decompor um problema de recomendação complexo em partes menores, permitindo que o conhecimento específico de cada agente seja aplicado na resolução de cada uma destas partes.

Como contribuição principal, a abordagem MATRES permite que os agentes sejam capazes de gerar e manipular suposições e compartilhar informações entre os agentes de confiança durante o processo de recomendação, com o objetivo de gerar recomendações de qualidade, ou seja, recomendações que sejam aprovadas pelos usuários.

5.1 Contribuições

As contribuições deste trabalho para o estado-da-arte de sistemas de recomendação baseados na arquitetura multiagente são revistas e detalhadas a seguir:

- A abordagem proposta (apresentada no capítulo 3) permite que os agentes, na falta de informações no processo de recomendação, possam ser capazes de assumir suposições. Esta suposições são de extrema importância para que o processo de recomendação não seja prejudicado quando os agentes enfrentam problemas de falta de informação durante o processo de recomendação. Este tipo de problema pode comprometer a qualidade da recomendação gerada por eles ou até mesmo aumentar

o tempo de espera do usuário, caso os agentes precisem esperar informação de outros. Conforme visto na seção 2.3, as abordagens baseadas em suposições existentes necessitam que o conjunto de suposições seja validado sempre que uma suposição precisa ser utilizada. Esta característica aumenta o tempo de resposta dos agentes. Para evitar este problema, a abordagem MATRES gera as suposições de acordo com a necessidade dos agentes. A geração das suposições conforme a necessidade do agente é um diferencial apresentado na abordagem MATRES, pois evita que seja gerada uma rede de possíveis suposições a ser validada posteriormente.

- Os agentes possuem a habilidade de compartilhar informações entre os agentes considerados confiáveis. Através do mecanismo de confiança, os agentes escolhem as tarefas a serem executadas e também escolhem os agentes com quem compartilham informações ao longo do processo de recomendação, garantindo a qualidade dos resultados obtidos no processo. A utilização deste mecanismo de confiança melhora a comunicação entre os agentes, pois as trocas de mensagens desnecessárias são eliminadas;

- Os agentes são capazes de se especializar ao longo dos processos de recomendação. O agente pode escolher a tarefa para resolver de acordo com seus índices de confiabilidade, dando preferência para as tarefas que já resolveu e obteve boas avaliações. Esta especialização contribui para melhora das recomendações geradas pelo agente;

- O componente de manutenção da verdade desenvolvido nesta abordagem é utilizado também na manutenção das bases de conhecimento dos agentes com o objetivo de manter a consistência de suas bases de conhecimento. O objetivo desta manutenção é manter a consistência das informações compartilhadas entre os agentes. As abordagens apresentadas na seção 2.3 apresentam a limitação de permitir que um agente com menos conhecimento altere o estado da informação de um agente com mais conhecimento. No componente de manutenção de bases proposto na abordagem MATRES, os agentes consideram seus índices de confiabilidade na atualização de suas bases.

Conforme apresentado no capítulo 2 do referencial teórico, não existe abordagem similar na literatura. Assim, estas contribuições tornam a abordagem aqui proposta inovadora, tratando limitações que existiam na área, especialmente pela combinação de diferentes conceitos, e abrindo possibilidades para novos estudos e combinações.

5.2 Trabalhos Futuros

Este trabalho deixa questões em aberto que apontam para trabalhos futuros, como por exemplo:

- Classificação por tipos de suposições: na validação no domíno do turismo algumas suposições foram utilizadas como exemplo. As suposições tinham o mesmo impacto na geração das recomendações dos agentes. Percebeu-se ao longo dos experimentos que as suposições poderiam ser classificadas de acordo com o grau de importância na geração da recomendação. Por exemplo, para gerar uma recomendação, a data de chegada do passageiro na cidade pode ter um impacto menos significativo do que a escolha da localização do hotel. Seria interessante como trabalho futuro definir graus de importância das suposições utilizadas e considerar

estes graus no momento da utilização das suposições na geração das recomendações, verificando assim se existe melhora na qualidade das recomendações geradas a partir destas suposições;

- Ampliar o mecanismo de confiança: o mecanismo de confiança desenvolvido nesta abordagem mostrou-se eficiente na redução da troca de mensagens entre os agentes e na especialização dos agentes. Entretanto, como trabalho futuro sugere-se a ampliação deste mecanismo. Por exemplo, pode-se utilizar o modelo matemático proposto por (BONABEAU; THERAULAZ; DORIGO, 1999), onde um indivíduo torna-se mais sensível ao estímulo associado a uma tarefa em que está engajado e menos sensível aos estímulos associados às outras tarefas. Cada indivíduo no modelo tem um limiar de resposta para cada tarefa e este limiar é atualizado (aumentando ou diminuindo) de acordo com dois coeficientes diferentes (de aprendizado e de esquecimento).

- Alterações na atualização do grau de confiança: quando o agente confiável não tem informação para compartilhar, o agente que solicitou a informação se comunica com o segundo agente confiável e assim sucessivamente até encontrar a informação ou até não ter mais agentes confiáveis para se comunicar. Na abordagem MATRES, o agente não atualiza seu grau de confiança no agente confiável caso este não tenha informação para compartilhar. Entretanto, é necessário investigar se a atualização do grau de confiança no momento da resposta do agente confiável traria resultados diferentes dos que foram obtidos nos experimentos realizados.

6 PUBLICAÇÕES RELACIONADAS A TESE

LORENZI, F.; BAZZAN, A. L. C.; ABEL, M.; RICCI, F. Improving recommendations through an assumption-based multiagent approach: an application in the tourism domain. (Em avaliação, submetido em Março/2010 para a revista Expert Systems With Applications)

LORENZI, F.; BALDO, G.; PEREIRA, R.; BAZZAN, A. L. C.; ABEL, M.; RICCI, F. A Trust Model for Multiagent Recommendations. (Aceito no Journal of Emerging Technologies in Web Intelligence - JETWI, special issue on Web Personalization, Reputation and Recommender Systems)

LORENZI, F.; RICCI, F.; ABEL, M; BAZZAN, A. L. C. Assumption-Based Reasoning for Multiagent Case-Based Recommender Systems. In: FLAIRS 2010 - CBR Track. Daytona Beach, 2010. p.342-347.

LORENZI, F.; BAZZAN, A. L. C.; ABEL, M. Multiagent Truth Maintenance Applied to a Tourism Recommender System. In: Nalin Sharda (Org.). Tourism Informatics: Visual Travel Recommender Systems, Social Communities and User Interface Design. Hershey: Information Science Reference (IGI Global), p. 54-72, 2009.

LORENZI, F.; CORREA, F. ; BAZZAN, A. L. ; ABEL, M. ; RICCI, F. A Multiagent Recommender System with Task-Based Agent Specialization. In: International Workshop on Agent Mediated Electronic Commerce (AMEC), 2008, Estoril. The tenth International Workshop on Agent Mediated Electronic Commerce, 2008.

LORENZI, F. ; SANTOS, F. ; FERREIRA JR., P. ; BAZZAN, A. L. Optimizing Preferences within Groups: A Case Study on Travel Recommendation. In: Brazilian Symposium on Artificial Intelligence, 2008, Salvador. Advanced in Artificial Intelligence - SBIA 2008, 2008. p. 103-112.

LORENZI, F. A Multiagent Knowledge-Based Recommender Approach with Truth Maintenance In: Doctotal Symposium - ACM Conference on Recommender Systems, 2007, Minneapolis. Proceedings of the 2007 ACM Conference on Recommender Systems. New York: ACM, 2007. p.195-198.

LORENZI, F.; BAZZAN, A. L. C.; ABEL, M. MANUTENÇÃO DE VERDADE EM SISTEMAS DE RECOMENDAÇÃO BASEADOS NA ARQUITETURA MULTIAGENTE. Revista Logos. p.18-32.

LORENZI, F.; BAZZAN, A. L. C.; ABEL, M. Truth Maintenance Task Negotiation in Multiagent Recommender System for Tourism. In: AAAI WORKSHOP ON INTELLIGENT TECHNIQUES FOR WEB PERSONALIZATION AND RECOMMENDER SYSTEMS IN E-COMMERCE (AAAI 2007), 2007. Proceedings. . . [S.l.: s.n.], 2007. p.122-125.

LORENZI, F.; BAZZAN, A. L. C.; ABEL, M. Negotiation for task allocation among agents in case-base recommender systems: a swarm-intelligence approach. In: STUDENT ABSTRACT - ON TWENTY-SECOND AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI 2007), 2007. Proceedings. . . [S.l.: s.n.], 2007. p.1886-1887.

LORENZI, F.; BAZZAN, A. L. C.; ABEL, M. An Architecture for a Multiagent Recommender System in Travel Recommendation Scenarios. In: WORKSHOP ON RECOMMENDER SYSTEMS (ECAI 2006), 3., 2006. Proceedings. . . ECAI, 2006. p.88-91.

REFERÊNCIAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the Next Generation of Recommender Systems: a survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, Piscataway, NJ, USA, v.17, n.6, p.734–749, 2005.

AL-NAZER, A.; HELMY, T. A Web Searching Guide: internet search engines and autonomous interface agents collaboration. **wi-iatw**, Los Alamitos, CA, USA, v.0, p.424–428, 2007.

ALTHEBYAN, Q.; HEXMOOR, H. A New Parameter for Maintaining Consistency in an Agent's Knowledge Base Using Truth Maintenance Systems. In: WRAC. **Anais...** [S.l.: s.n.], 2005. p.53–64.

AMBITE, J. L. et al. Conditional Constraint Networks for Interleaved Planning and Information Gathering. **IEEE Intelligent Systems**, Piscataway, NJ, USA, v.20, n.2, p.25–33, 2005.

BALABANOVIC, M.; SHOHAM, Y. Fab: content-based, collaborative recommendation. **Communications of the Association for Computing Machinery**, Orlando, Florida, v.40, n.3, p.66–72, July 1997.

BERKOVSKY, S.; KUFLIK, T.; RICCI, F. Mediation of user models for enhanced personalization in recommender systems. **User Modeling and User-Adapted Interaction**, Hingham, MA, USA, v.18, n.3, p.245–286, 2008.

BILLSUS, D.; PAZZANI, M. A Hybrid User Model for News Story Classification. In: SEVENTH INTERNATIONAL CONFERENCE ON USER MODELING, UM '99, Banff, Canada. **Proceedings...** [S.l.: s.n.], 1999.

BOGAERTS, S.; LEAKE, D. Facilitating CBR for Incompletely-Described Cases: distance metrics for partial problem descriptions. In: EUROPEAN CONFERENCE ON CASE BASED REASONING, ADVANCES IN CASE-BASED REASONING, 7., Madrid. **Proceedings...** [S.l.: s.n.], 2004. p.62–76. (Lecture Notes in Artificial Intelligence).

BOJDUJ, B.; WEBER, B.; TAYLOR, D. A Framework for Truth Maintenance in Multi-Agent Systems. In: MCCASKILL, J. S.; PACKARD, N. H.; RASMUSSEN, S. (Ed.). **Operations Research Proceedings**. [S.l.]: Springer Berlin Heidelberg, 2006. v.VII, p.403–408.

BONABEAU, E.; THERAULAZ, G.; DORIGO, M. **Swarm Intelligence**: from natural to artificial systems. New York, USA: Oxford University Press, 1999. 307p.

BUFFETT, S. et al. Negotiating Exchanges of P3P-Labeled Information for Compensation. **Computational Intelligence**, [S.l.], v.20, n.4, p.663–677, 2004.

BURKE, R. Knowledge-based Recommender Systems. In: DAILY, J. E.; KENT, A.; LANCOUR, H. (Ed.). **Encyclopedia of Library and Information Science**. [S.l.]: Marcel Dekker, 2000. v.69.

BURKE, R. Hybrid Recommender Systems. **The Adaptive Web: Methods and Strategies of Web Personalization**, Berlin-Heidelberg, v.4321, p.377–408, 2007.

CAMACHO, D. et al. Multi-agent plan based information gathering. **Applied Intelligence**, Hingham, MA, USA, v.25, n.1, p.59–71, 2006.

CHEN, E.; VAHIDOV, R. M.; KERSTEN, G. E. Agent-supported negotiations in the e-marketplace. **International Journal of Electronic Business**, Hingham, MA, USA, v.3, n.1, p.28–49, 2005.

DEKLEER, J. An Assumption-based tms, Extending the ATMS, and Problem Solving with the ATMS. **Artificial Intelligence**, [S.l.], v.28, n.2, p.127–224, 1986.

DOYLE, J. A Truth Maintenance System. **Artificial Intelligence**, [S.l.], v.12, n.3, p.231–272, 1979.

ENGELMORE, R.; MORGAN, A. **Blackboard Systems**. [S.l.]: Addison-Wesley, 1988.

GOY, A.; ARDISSONO, L.; PETRONE, G. Personalization in E-Commerce Applications. In: THE ADAPTIVE WEB. **Anais...** Springer Berlin / Heidelberg, 2007. p.485–520.

GUNAWARDANA, A.; MEEK, C. A unified approach to building hybrid recommender systems. In: ACM CONFERENCE ON RECOMMENDER SYSTEMS - RECSYS'09, New York, NY, USA. **Proceedings...** ACM, 2009. p.117–124.

HERLOCKER, J. L. et al. Evaluating Collaborative Filtering Recommender Systems. **ACM Transactions on Information Systems**, [S.l.], v.22, n.1, p.5–53, 2004.

HUHNS, M. N.; BRIDGELAND, D. M. Multiagent Truth Maintenance. **IEEE Transactions on Systems, Man, and Cybernetics**, [S.l.], v.21, n.6, p.1437–1445, November/December 1991.

KOBSA, A. Generic User Modeling Systems. **User Modeling and User-Adapted Interaction**, Hingham, MA, USA, v.11, n.1-2, p.49–63, 2001.

KOLODNER, J. **Case-Based Reasoning**. San Mateo, CA: Morgan Kaufmann, 1993. 668p.

KONSTAN, J. A.; RIEDL, J. Research resources for recommender systems. In: ACM SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS: WORKSHOP INTERACTING WITH RECOMMENDER SYSTEMS. **Anais...** [S.l.: s.n.], 1999.

LESSER, V. R. et al. BIG: an agent for resource-bounded information gathering agent and decision making. **Artificial Intelligence Journal, Special Issue on Internet Information Agents**, [S.l.], v.118, n.1-2, p.197–244, May 2000.

LORENZI, F. et al. Task allocation in case-based recommender systems: a swarm intelligence approach. In: LIN, H. (Ed.). **Architectural Design of Multi-Agent Systems**. [S.l.]: Information Science Reference, 2007. p.268–279.

LORENZI, F.; RICCI, F. Case-based recommender systems: a unifying view. In: INTELLIGENT TECHNIQUES FOR WEB PERSONALIZATION. **Anais...** Springer Verlag, 2005. p.89–113.

LORENZI, F.; SANTOS, D. S.; BAZZAN, A. L. C. Negotiation for task allocation among agents in case-base recommender systems: a swarm-intelligence approach. In: WORKSHOP MULTI-AGENT INFORMATION RETRIEVAL AND RECOMMENDER SYSTEMS - NINETEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI 2005). **Proceedings...** [S.l.: s.n.], 2005. p.23–27.

MALHEIRO, B.; VARGA, L. Z.; OLIVEIRA, E. **Beliefs and Conflicts in a Real World Multiagent System**. Tenerife, Espanha, 1998. 322–329p.

MASON, C. L.; JOHNSON, R. R. DATMS: a framework for distributed assumption based reasoning. **Distributed Artificial Intelligence**, [S.l.], v.2, p.293–317, 1989.

MASSA, P.; BHATTACHARJEE, B. Using trust in recommender systems: an experimental analysis. In: IN PROCEEDINGS OF ITRUST2004 INTERNATIONAL CONFERENCE. **Anais...** [S.l.: s.n.], 2004. p.221–235.

MCSHERRY, D. Diversity-conscious Retrieval. In: EUROPEAN CONFERENCE ON CASE BASED REASONING, ADVANCES IN CASE-BASED REASONING, 6., Aberdeen, Scotland. **Proceedings...** Springer-Verlag, 2002. p.219–233. (Lecture Notes in Artificial Intelligence).

MILLER, B. et al. MovieLens Unplugged: experiences with an occasionally connected recommender system. In: ACM 2003 INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES (IUI'03), Chapel Hill, North Carolina. **Proceedings...** ACM Press, 2003.

MOLINARI, L. **Testes de Software**: produzindo sistemas melhores e confiÃ¡veis. Sao Paulo: Erica Ltda, 2003.

MONTANER, M.; LóPEZ, B.; ROSA, J. L. de la. Developing trust in recommender agents. In: AAMAS '02: PROCEEDINGS OF THE FIRST INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, New York, NY, USA. **Anais...** ACM, 2002. p.304–305.

MONTANER, M.; LÓPEZ, B.; ROSA, J. L. de la. A Taxonomy of Recommender Agents on the Internet. **Artificial Intelligence Review**, [S.l.], v.19, n.4, p.285–330, 2003.

MOWEN JOHN C. E MINOR, M. S. **Comportamento do consumidor**. Sao Paulo: Prentice-Hall, 2003.

NAGENDRA, P. et al. Retrieval and Reasoning in Distributed Case Bases. **Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries**, [S.l.], v.7, n.1, p.74–87, January 1996.

NASSIRI, N. Increasing trust through the use of 3d e-commerce environment. In: SAC '08: PROCEEDINGS OF THE 2008 ACM SYMPOSIUM ON APPLIED COMPUTING, New York, NY, USA. **Anais...** ACM, 2008. p.1463–1466.

O'DONOVAN, J.; SMYTH, B. Trust in recommender systems. In: IUI '05: PROCEEDINGS OF THE 10TH INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES, New York, NY, USA. **Anais...** ACM, 2005. p.167–174.

ONTANON, S.; PLAZA, E. Recycling Data for Multi-Agent Learning. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING ICML-2005, 22., Madrid. **Proceedings...** [S.l.: s.n.], 2005. p.633–640. (ACM Press).

PLAZA, E.; MCGINTY, L. Distributed case-based reasoning. **The Knowledge Engineering Review**, [S.l.], v.20, n.3, p.261–265, 2005.

RAMCHURN, S. D.; HUYNH, D.; JENNINGS, N. R. Trust in multi-agent systems. **Knowl. Eng. Rev.**, New York, NY, USA, v.19, n.1, p.1–25, 2004.

RESNICK, P. et al. GroupLens: an open architecture for collaborative filtering of netnews. In: ACM CONFERENCE ON COMPUTER-SUPPORTED COOPERATIVE WORK. **Proceedings...** [S.l.: s.n.], 1994. p.175–186.

RICCI, F. Travel recommender Systems. **IEEE Intelligent Systems**, [S.l.], v.17, n.6, p.55–57, 2002.

RICCI, F. et al. Product Recommendation with Interactive Query Management and Twofold Similarity. In: INTERNATIONAL CONFERENCE ON CASE-BASED REASONING, 5., Norway. **Anais...** [S.l.: s.n.], 2003. p.479–493.

RICCI, F.; VENTURINI, A. eCTRL Solutions: trip@dvice technology. In: EGGER, R.; BUHALIS, D. (Ed.). **eTourism case studies**: management and marketing issues in etourism (etourism case studies). [S.l.]: Butterworth-Heinemann, 2008. p.–.

SABATER, J.; SIERRA, C. REGRET: reputation in gregarious societies. In: AGENTS '01: PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, New York, NY, USA. **Anais...** ACM, 2001. p.194–195.

SCHAFER, J.; KONSTAN, J.; RIEDL, J. E-Commerce Recommendation Applications. **Data Mining and Knowledge Discovery**, [S.l.], v.5, n.1/2, p.115–153, 2001.

SCHIEL, U. et al. SEI-Tur: a system based on composed web-service discovery to support the creation of trip plans. In: ICDS'07: PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON THE DIGITAL SOCIETY, Washington, DC, USA. **Anais...** IEEE Computer Society, 2007. p.28.

SCHMIDT-THIEME, L.; FELFERNIG, A.; FRIEDRICH, G. Introduction: recommender systems. In: IEEE INTELLIGENT SYSTEMS. **Anais...** IEEE, 2007. p.18–21.

SMYTH, B. Case-Based Recommendation. In: THE ADAPTIVE WEB. **Anais...** Springer Berlin / Heidelberg, 2007. p.342–376.

SYMEONIDIS, P. et al. Collaborative recommender systems: combining effectiveness and efficiency. **Expert Syst. Appl.**, Tarrytown, NY, USA, v.34, n.4, p.2995–3013, 2008.

TORRENS, M.; FALTINGS, B.; PU, P. SmartClient: constraint satisfaction as a paradigm for scaleable intelligent information systems. **CONSTRAINTS: an International Journal**, [S.l.], v.7, n.1, p.49–69, 2002.

TORRES, R. et al. Enhancing Digital Libraries with TechLens. In: JOINT CONFERENCE ON DIGITAL LIBRARIES, Tucson, Arizona. **Anais...** [S.l.: s.n.], 2004. p.228–237.

TOWLE, B.; QUINN, C. **Knowledge Based Recommender Systems Using Explicit User Models**. Menlo Park: CA: AAAI Press., 2000. 74–77p.

VICTOR, P. et al. Gradual trust and distrust in recommender systems. **Fuzzy Sets Syst.**, Amsterdam, The Netherlands, The Netherlands, v.160, n.10, p.1367–1382, 2009.

WEI, Y. Z. et al. User evaluation of a market-based recommender system. **Autonomous Agents and Multi-Agent Systems**, [S.l.], v.17, n.2, p.251–269, 2008.

WERTHNER, H.; RICCI, F. E-commerce and tourism. **Commun. ACM**, New York, NY, USA, v.47, n.12, p.101–105, 2004.

WU, S. et al. Personal assistant agents for collaborative design environments. **Comput. Ind.**, Amsterdam, The Netherlands, The Netherlands, v.57, n.8, p.732–739, 2006.

ZAMBONELLI, F. et al. Agent-Oriented Software Engineering for Internet Applications. In: COORDINATION OF INTERNET AGENTS. **Anais...** Springer-Verlag, 2001. p.326–346.

ZHANG, D. et al. A multi agent recommender system that utilises consumer reviews in its recommendations. **International Journal of Intelligent Information and Database Systems**, [S.l.], v.2, n.1, p.69–81, 2008.