

Utilização de Sistemas Multiagentes na Construção de Sistemas de Recomendação

**Christiano O. Ávila, Rosaura Silva, Luiz A. Palazzo, Stanley Loh,
PPGINF - Universidade Católica de Pelotas R. Felix da Cunha, 412, Pelotas, RS –
96010-000**

**chris AT direto2.ucpel.tche.br, rosaura AT sisinfo.inf.br, lpalazzo AT ucpep.tche.br,
loh AT ucpep.tche.br**

***Resumo.** O presente artigo relata a utilização de funcionalidades e características de um sistema multiagente em um sistema de recomendação chamado SisRecAC. Este sistema descobre e recomenda artigos científicos baseados nos próprios documentos armazenados pelos usuários da ferramenta. Os agentes autônomos adicionados ao sistema possibilitarão automatizar a descoberta de novos artigos e proporcionará maior interatividade entre os usuários e o sistema, tendo em vista que os agentes estarão enviando mensagens (sistema de alertas) aos usuários sobre estes novos artigos que estarão sendo descobertos.*

Palavras-chave - sistema de recomendação, agentes inteligentes, biblioteca digital, extração de palavras-chave.

1. INTRODUÇÃO

Para resolver o problema da sobrecarga de informações [2][5] os Sistemas de Recomendação foram criados e hoje são alvo de inúmeras pesquisas. Este tipo de sistema visa apresentar ao usuário somente aquilo que é de seu interesse e aí está grande parte do foco das pesquisas, ou seja, localizar e apresentar informações que são relevantes ao usuário. Neste sentido, foi desenvolvido o sistema SisRecAC que, baseado em documentos textuais enviados pelo usuário, recomenda artigos científicos, capturados na web, similares a estes documentos. O sistema funciona como uma ferramenta de metabusca, que extrai palavras-chave do documento do usuário, envia para o Google Acadêmico, captura os artigos que foram localizados e os apresenta para o usuário em forma de uma lista de recomendação. A ferramenta permite também que o usuário forneça um feedback, avaliando a relevância dos artigos recomendados pelo sistema e os recomendando para outras pessoas.

2. OBJETIVO DO TRABALHO

O objetivo deste trabalho é definir e implementar um sistema de alertas que utilize características de multiagentes inteligentes. Os alertas devem informar aos usuários do SisRecAC sobre alterações ocorridas na lista artigos anteriormente descobertos. O foco deste projeto é utilizar agentes para dar mais dinamicidade ao sistema, avisando aos usuários periodicamente, sobre eventuais novos artigos descobertos a partir de um determinado documento armazenado no sistema. Além de proporcionar maior dinamicidade ao sistema, esses alertas terão o objetivo de fidelizar os usuários à ferramenta, uma vez que serão lembrados periodicamente de sua existência e importância na descoberta de informações de seu interesse.

3. SISTEMAS MULTIAGENTES

Um Sistema Multiagente (SMA) é um tipo de sistema de Inteligência Artificial Distribuída no qual é utilizado um conjunto de componentes de inteligência artificial, chamado agentes. Os agentes são componentes autônomos com objetivos particulares que se inter-relacionam em um ambiente, interagindo, negociando e coordenando esforços para resolução de tarefas.

As características dos agentes estão diretamente relacionadas com o ambiente onde está inserido, o ambiente define se os agentes são homogêneos ou heterogêneos, se os objetivos são comuns a todos os agentes ou não, se os agentes têm caráter competitivo ou colaborativo, etc.

Basicamente um SMA é composto por um ambiente, um conjunto de objetos (sensores, atuadores), um conjunto de agentes, um conjunto de relações que ligam os objetos, um conjunto de operações (tarefas), operadores que representam os resultados das operações e as reações do ambiente a eles e um conjunto de interações possíveis entre os agentes.

Um sistema multiagente (SMA) se caracteriza por ser um sistema onde vários agentes atuam em conjunto na busca da resolução de problemas. Os agentes interagem entre si de forma cooperativa buscando alcançar seus objetivos [1].

Um agente pode ser entendido como uma entidade de software que atua em um ambiente com capacidade de interagir tanto com o ambiente como com outros agentes. Um agente caracteriza-se por ter um comportamento autônomo possibilitando a execução de ações para alcançar os objetivos do sistema. Observando as propriedades e características de um SMA é possível entender porque estes sistemas têm sido utilizados para a resolução de problemas complexos, SMAs possibilitam definir de forma distribuída, autônoma, descentralizada e cooperativa as ações que executadas irão obter a resolução dos problemas.

Segundo [6] os agentes são classificados em quatro tipos:

- Arquiteturas baseadas em lógica.
- Arquiteturas reativas.
- Arquiteturas em camadas.
- Arquiteturas de crença-desejo-intenção (BDI).

O presente trabalho será baseado na Arquitetura de crença-desejo-intenção (BDI). A arquitetura BDI é baseada em estados mentais do raciocínio prático. O estado do agente é representado por três estruturas: suas crenças (*beliefs*), seus desejos (*desires*) e suas intenções (*intentions*).

As crenças de um agente são o conhecimento que o agente possui sobre o ambiente em que ele se encontra. Os desejos de um agente representam o que o agente deseja é o estado motivacional do sistema. As intenções de um agente são as ações que ele irá executar para alcançar os seus objetivos. Além destes componentes, algumas arquiteturas BDI usam o conceito de planos, que seriam o "passo-a-passo" a ser seguido para a realização de uma ação sobre o ambiente.

Na figura 1 é apresentada a arquitetura genérica de um BDI proposta por [6]. A arquitetura além de apresentar os estados mentais de crenças, desejos e intenções e apresenta também funções que integram a arquitetura.

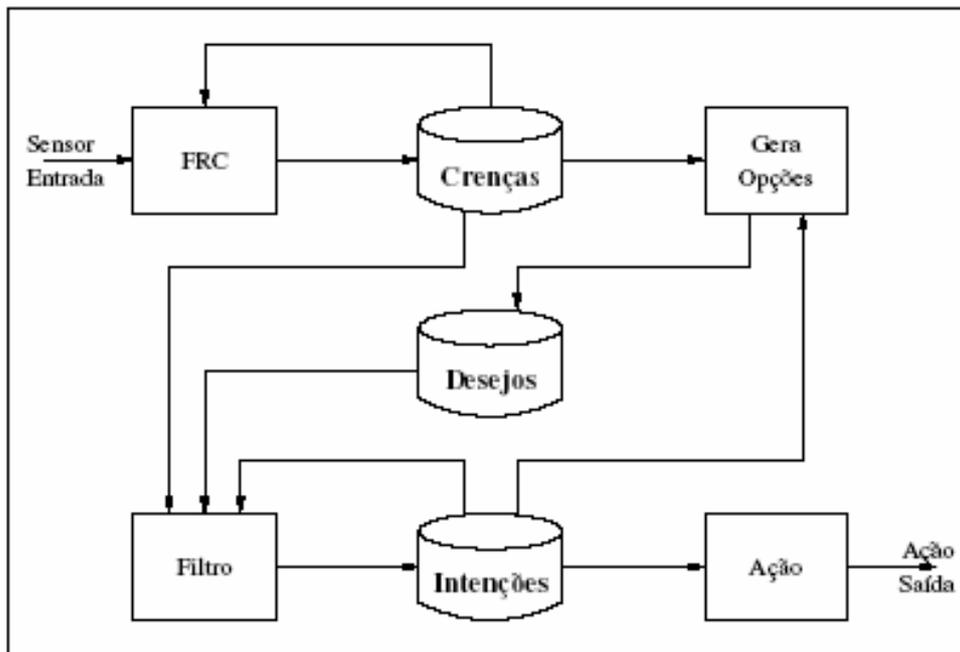


Fig. 1 – Arquitetura BDI genérica

A função FRC recebe as informações do ambiente podendo atualizar as crenças do agente conforme o novo estado do ambiente. A partir do novo estado do ambiente os objetivos do agente podem ser alterados. A função Gera Opções verifica quais são os resultados que devem ser atingidos no estado atual e as intenções com as quais o agente já está comprometido. A função Ação determina qual ação deverá ser realizada. A função Filtro atualiza o conjunto de intenções do agente baseado nas suas crenças e desejos.

4. METODOLOGIA

Uma grande variedade de técnicas e metodologias tem sido proposta para a Engenharia de Software baseada em agentes e ainda não existe um padrão reconhecido que reúna o melhor de cada uma delas.

A maioria das propostas evidencia duas fases para a construção de um SMA, análise e projeto. Dentre essas metodologias podemos citar as metodologias GAIA e MAS-CommonKADS entre outras.

Para o desenvolvimento do sistema multiagente no SisRecAC, foi escolhida a metodologia de modelagem de agentes baseada na plataforma JADE (Java Agent Development Framework). A metodologia baseada em JADE [4] combina uma abordagem top-down e bottom-up de forma que ambas existam na capacidade do sistema de forma a satisfazer as necessidades das aplicações mais complexas. A metodologia proposta tenta formalizar a fase de análise e projeto do ciclo de vida de desenvolvimento de software baseado em agente. A formalização das fases de planejamento e implementação do ciclo de vida do desenvolvimento do software está atualmente fora do escopo da metodologia.

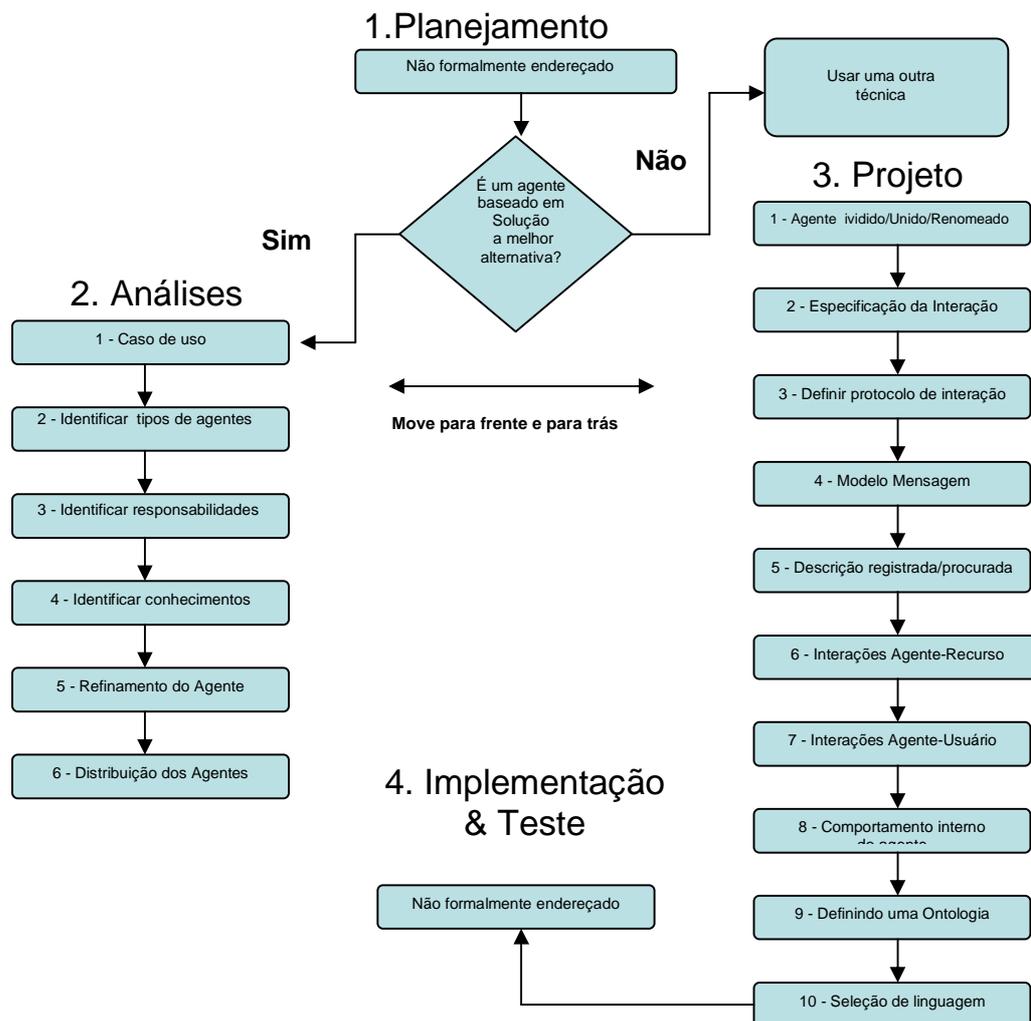


Fig. 2 – Visão geral da metodologia

Fase de análise: esta fase aponta claramente o problema sem preocupação sobre a solução.

Passos:

1 - Caso de Uso: permitem capturar os requisitos funcionais potenciais para um novo sistema. Cada caso de uso apresenta um ou mais cenários que demonstram como o sistema pode interagir com o usuário final ou com outro sistema para alcançar os objetivos específicos. Popularmente utiliza-se a Linguagem de Modelagem Unificada (UML).

2 - Identificação inicial de tipos de agentes: Este passo envolve identificação dos principais tipos de agentes e subsequente formação de um primeiro rascunho do diagrama de agente. As seguintes regras poderiam ser aplicadas neste passo:

- Adiciona um tipo de agente por o usuário/dispositivo.
- Adicionar um tipo de agente por o recurso (que inclui o software legado).

3 - Identificação de responsabilidade: Neste passo, para cada tipo de agente identificado, uma lista inicial é feita das principais responsabilidades de uma maneira intuitiva e informal. O produto como consequência deste processo é a tabela de responsabilidade. As regras seguintes deveriam ser aplicadas neste passo:

- Derivar o conjunto inicial de responsabilidades para o caso de uso identificado no passo 1.

- Considerar os agentes onde estas responsabilidades são transparentes primeiro e atrasar a identificação de responsabilidades para outros agentes para passos posteriores.
- 4 - Identificação dos conhecimentos: Nesta etapa, o foco está em quem necessita interagir com quem e o diagrama do agente é atualizado adicionando as relações apropriadas do conhecimento que conectam os agentes que necessitam ter um ou mais interações.
- 5 - Refinamento do Agente: Neste passo, o conjunto de tipos de agentes inicialmente identificados no passo 2 é refinado aplicando um número de considerações. Estes são relacionados à:
 - Suporte: que suporte de informação os agentes necessitam para executar suas responsabilidades, e como, quem, e onde esta informação é gerada/armazenada.
 - Descoberta: como agentes ligados por uma relação de conhecimento descobrem um ao outro.
 - Gerenciamento e monitoramento: sistema requerido para manter-se a par de agentes existentes, ou começar ou parar os agentes sob demanda.
- 6 - Informação da distribuição do agente: Um outro instrumento que pode ser útil de produzir é o diagrama da distribuição do agente, onde os anfitriões/agentes físicos (*hosts/devices*) dos dispositivos estão sendo desenvolvidos (referenciam os domínios em algumas metodologias) é indicado.

Fase de projeto: esta fase aponta para a solução do problema.

Passos:

- 1 - Agente dividido/unido/renomeado: Esta etapa envolve observar os instrumentos produzidos na fase do projeto e determinar se os tipos do agente produzidos no diagrama do agente devem ser divididos ou unidos. Esta etapa é considerada importante, desde que tem um efeito direto na eficiência e na complexidade do sistema total.
- 2 - Especificação das interações: para cada tipo do agente, todas as responsabilidades que são relacionadas a um outro agente (baseado na tabela da responsabilidade produzida na análise) são examinadas e uma tabela da interação é produzida para cada tipo do agente.
- 3 - Definição ad hoc do protocolo da interação: os protocolos possíveis, existentes na interação devem ser definidos pela FIPA. Entretanto, é frequente o caso que uma interação requerer a definição de um protocolo *ad hoc* (isto ocorre quando nenhum dos protocolos de interação definidos pela FIPA são julgados adequados). Nesses casos, o protocolo da interação deve ser definido por meio de um formalismo apropriado.
- 4 - Moldes da mensagem: todos os papéis do protocolo da interação identificados na etapa precedente são executados como comportamentos do JADE. Nesta etapa, os objetos apropriados de *MessageTemplate* são especificados para ser usados nestes comportamentos de recebimento de mensagens, e estes moldes são adicionados às fileiras da tabela da interação.
- 5 - Descrição registada/procurada (*Yellow Pages*): nesta etapa, nas convenções nomeando e nos serviços registrando/procurando por agentes em um catálogo de páginas amarelas mantidos por um diretório facilitador do JADE são formalizados (quando relevante).
- 6 - Interação Agente-Recurso: é frequente o caso que um ou o mais agentes no sistema devem interagir com os recursos externos tais como bases de dados, arquivos que armazenam a informação, ou software legado. Em alguns casos alguns dispositivos de hardware devem ser controlados ou monitorados, isto acontece sempre através de algum software dedicado que esconde realmente o hardware atrás dele.

7 - Interação Agente-Usuário: Em muitos casos, um agente necessita interagir com um usuário. Há diversas maneiras em que um usuário humano pode interagir com uma parte de software tal como um agente do JADE. Aqui, o foco está na interface gráfica de usuário (GUI), que é o tipo mais usado de interface com o usuário.

8 - Comportamento interno do Agente: O trabalho que um agente tem que fazer é realizado tipicamente dentro dos comportamentos do agente. Nesta etapa, o projetista do sistema deve olhar as responsabilidades do agente (através da tabela da responsabilidade) identificadas na fase de análise e traçá-las aos comportamentos do agente.

9 - Definir uma ontologia: quando os agentes interagem no sistema, trocam informações referentes a entidades, abstratas ou concretas, que existem no ambiente onde os agentes residem. Estas entidades podem ser primitivas, tal como uma string ou um número, ou podem ter estruturas complexas definidas por modelos especificados nos termos de um nome e de um conjunto dos slots cujos valores devem ser de um tipo dado. Estes modelos complexos de entidade são referentes aos conceitos.

10 - Seleção do conteúdo da linguagem: JADE fornece códigos para duas linguagens, a linguagem LS e a linguagem LEAP, (através do pacote de *jade.content*). Além disso, um código pode ser definido por um programador se desejar que os agentes falem uma língua diferente.

A metodologia baseada em JADE utiliza um conjunto de diagramas, esquemas ou representação em documentos em gráficos ou textos para representar um modelo, como descrito a seguir:

Diagrama de uso de caso : lista preliminar de cenários possíveis após entrevista com os usuários do sistema.

Diagrama de Estados: usado para representar os cenários possíveis e os diversos tipos de agentes identificados no modelo.

Diagrama de Agentes O diagrama do agente é um dos artefatos principais produzidos na fase de análise este diagrama inclui quatro tipos de elementos:

- - Tipos de agentes – representados por círculos;
- - Humanos – pessoas que interagem com desenvolvimento do sistema, representado pelo símbolo ator da UML;
- - Recursos – sistemas externos que interagem com o sistema, representado por retângulos;
- - Conhecimentos – representado por setas, define a interação entre os elementos do sistema.

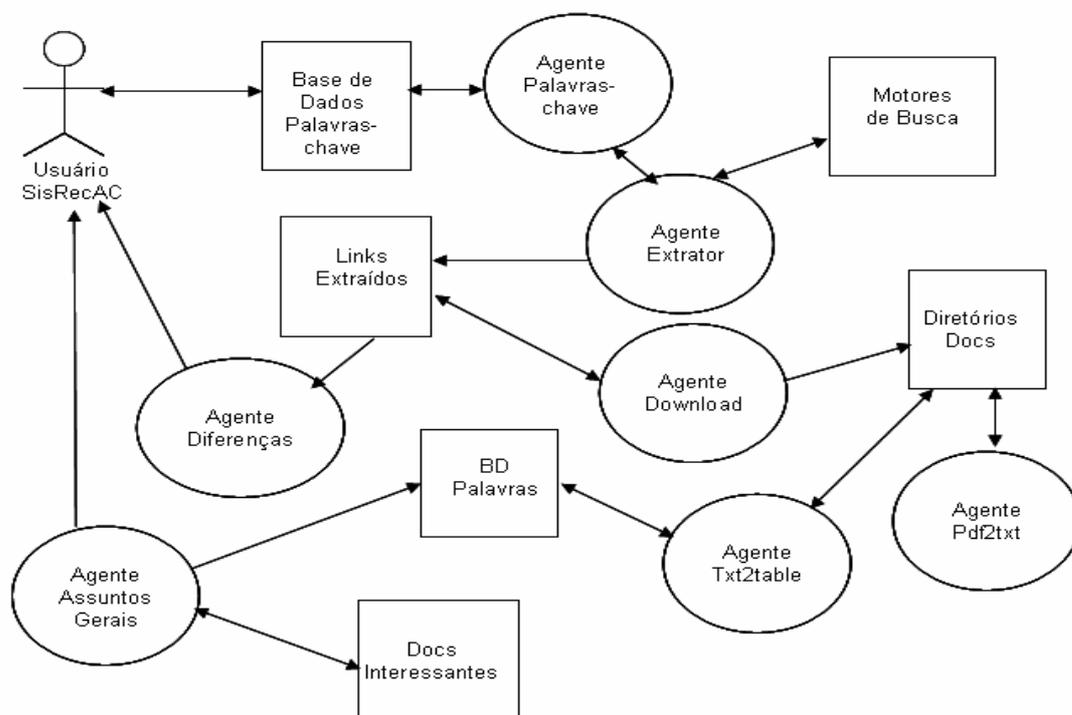


Fig. 3 – Diagrama de Agente para SisRecAC após o passo 5 da fase de análise.

Tabela de Responsabilidades: Lista as responsabilidades de cada um dos agentes identificados no modelo.

Tipo de Agente	Responsabilidades
Agente Palavras-chave	- ler a tabela do banco de dados com as palavras-chave dos documentos armazenados - enviar uma mensagem para o agente Extrator com as palavras-chave de cada documento Observação: Não existe um agente que extraia as palavras-chave dos documentos, pois essa função é executada automaticamente quando o usuário realiza o <i>upload</i> de um documento. No momento do <i>upload</i> , além de extrair as palavras-chave o módulo do SisRecAC também realiza a gravação destas palavras em uma tabela.
Agente Extrator	- recebe mensagem do Agente Palavras-chave contendo as palavras-chave e o identificador do documento - submete as palavras-chave ao motor de busca - recebe a informação de retorno do motor de busca - extrai os links resultantes no formato pdf - armazena os links em uma tabela
Agente Download	- monitora a tabela de links - verifica se já existe o documento no servidor - caso não encontre o documento no servidor, realiza o <i>download</i> do documento para diretório específico
Agente Pdf2txt	- monitora o diretório dos documentos pdf extraídos - verifica se já existe o documento correspondente no formato txt - caso não exista o formato txt do documento o Agente converte o documento para txt possibilitando a extração das palavras-chave
Agente Txt2table	- transfere o documento em formato txt para tabela onde cada palavra ocupará uma tupla da tabela
Agente Diferenças	- analisa a lista de recomendação original, que foi gravada no momento do upload com a nova lista descoberta pelos agentes - encontrando alguma alteração na lista (inclusão ou exclusão de documentos) o agente envia mensagem ao usuário informando a alteração
Agente InteressesGerais	- analisa as palavras-chaves de todos os documentos submetidos pelo usuário - tenta descobrir um padrão de interesse do usuário

5. Descrição do SisRecAC

O objetivo do SisRecAC é ser um repositório de documentos que recomenda artigos científicos baseados nos próprios documentos armazenados. A ferramenta funciona como um sistema de metabusca, que utiliza o Google Acadêmico (<http://scholar.google.com/>) para descobrir os artigos. Pode-se dizer que o principal objetivo do sistema é recomendar artigos científicos que sejam similares a documentos armazenados no sistema ou indicados pelos usuários. O sistema recebe do usuário um documento como referência e extrai automaticamente palavras-chaves, as quais são utilizadas para a busca de documentos similares no Google Acadêmico. Somente documentos em formato PDF são capturados. Estes documentos são então recomendados ao usuário e podem ser avaliados por parte do usuário, que por sua vez pode comentar e recomendar a outras pessoas, criando uma rede de cooperação e disseminação de artigos científicos.

Outro objetivo desta ferramenta é possibilitar uma análise dos melhores métodos de extração de palavras-chaves, onde os diferentes métodos serão avaliados em relação a qualidade da relevância dos documentos recuperados. Para extrair as palavras-chaves do documento utiliza-se o método da frequência simples, ou seja, as 7 (sete) palavras que mais se repetem no documento, retirando-se as *stopwords*, são consideradas as palavras-chaves do documento que são submetidas ao Google Acadêmico. A fig. 3 mostra a tela principal do SisRecAC, com as categorias do usuário a esquerda, dentro das categorias os documentos enviados e a direita os artigos que foram descobertos pelo sistema.

GPSI - SisRecAC

ENVIAR DOCUMENTO
INTELIGÊNCIA ARTIFICIAL
SISTEMAS DE RECOMENDAÇÃO
TESES

S A I R

DESCRIÇÃO

IA (webmedia.pdf)

Recomendações do SisRecAC

Avalie os links abaixo para que possamos aprimorar os algoritmos que geram as recomendações.

UM ESTUDO SOBRE GOOGLE: QUESTÕES PARA UMA LEITURA MICROPOLÍTICA DAS TECNOLOGIAS DA INFORMAÇÃO
... Como resultado disso, as buscas ... Para tal efeito, o Google, além das palavras-chave, sua proximidade ... pública de ações da Google traz informações sobre os ...
© Diaz-Isernath - luc.ufrj.br
Este documento é relevante no contexto do documento de origem?
 Totalmente relevante
 Parcialmente relevante
 Irrelevante
Avaliar

GERADOR INTELIGENTE DE SISTEMAS COM AUTO-APRENDIZAGEM PARA GESTÃO DE INFORMAÇÕES E CONHECIMENTO
... consubstanciaram nesta pesquisa em novos conhecimentos. ... partir de informações oriundas

Fig. 4 – Página inicial do SisRecAC.

Em síntese o SisRecAC, atualmente:

- permite que o usuário armazene documentos
- recomenda artigos similares aos documentos enviados
- permite que o usuário avalie a relevância dos artigos recomendados pelo SisRecAC
- permite que o usuário recomende a outras pessoas os seus documentos e os artigos recomendados pelo SisRecAC que foram considerados relevantes pelo usuário.

O uso de agentes no SisRecAC pode ser considerado como uma estratégia de marketing do sistema, ou seja, uma forma de satisfazer e surpreender o usuário com novidades a respeito de suas área de interesse. Essas novidades a respeito de suas áreas de interesse serão periodicamente enviadas para o email (fig. 5) deste usuário configurando-se em uma estratégia de manter o vínculo entre a ferramenta e seus usuários. A fig. 4 ilustra a estrutura básica do SisRecAC.

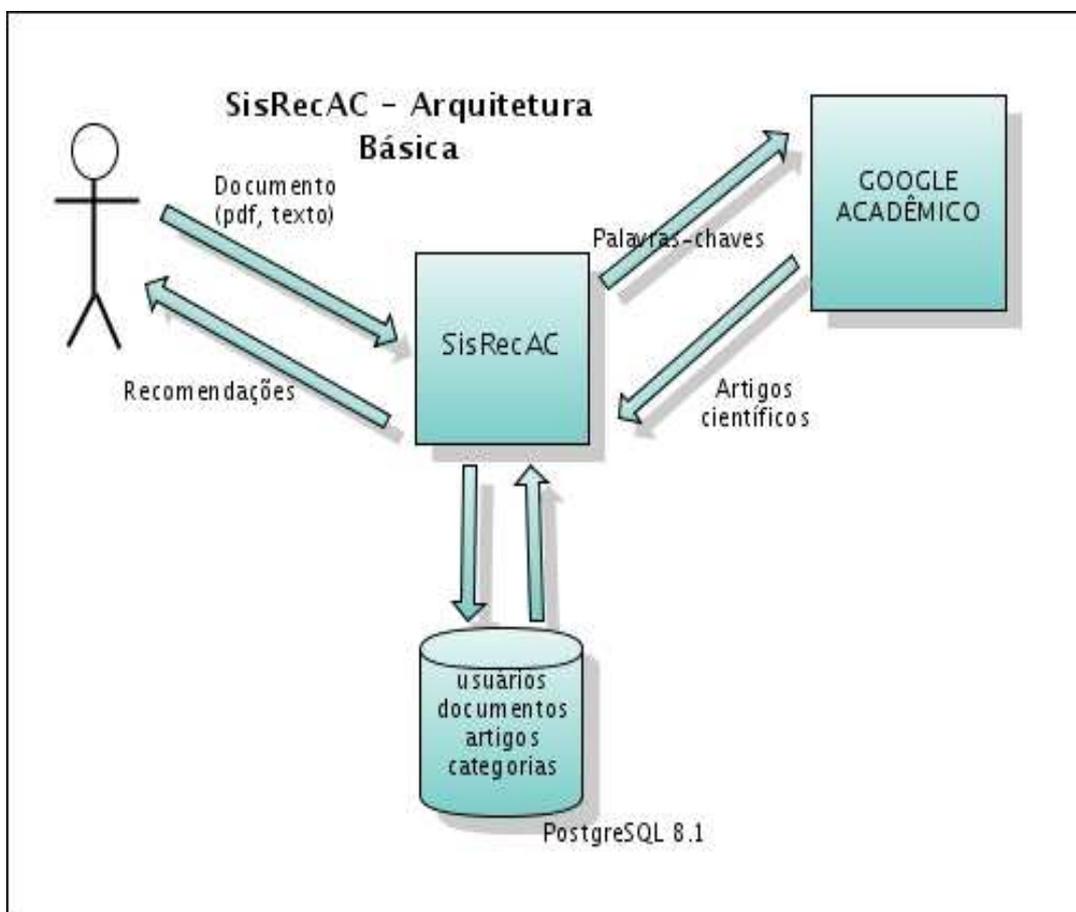


Fig. 5 - Arquitetura básica do SisRecAC



Fig. 6 – Email enviado pelo agente para um usuário

6. AGENTES PROPOSTOS PARA O SisRecAC

Os agentes propostos buscam satisfazer a característica essencial da abordagem SMA que é a filosofia de resolução distribuída de problemas, utiliza a estratégia de dividir para conquistar. A resolução cooperativa distribuída de problemas diz que um problema é dividido em subproblemas e cada um é solucionado separadamente por um agente ou grupo de agentes, cada um destes comunicando ou cooperando entre si quando necessário, com a idéia básica de que a soma dos resultados locais corresponderá à solução do problema geral [3]. Desta forma é possível constatar que aplicar uma abordagem orientada à agentes para a resolução de um problema significa decompô-lo em múltiplos componentes autônomos com objetivos particulares e que se interrelacionam.

Para contemplar esta abordagem é possível perceber que a organização, coordenação, cooperação, comunicação e interação são características fundamentais em um SMA. A seguir serão relacionados os agentes que já foram identificados e seus objetivos principais e um diagrama pode ser visualizado na Fig.6.

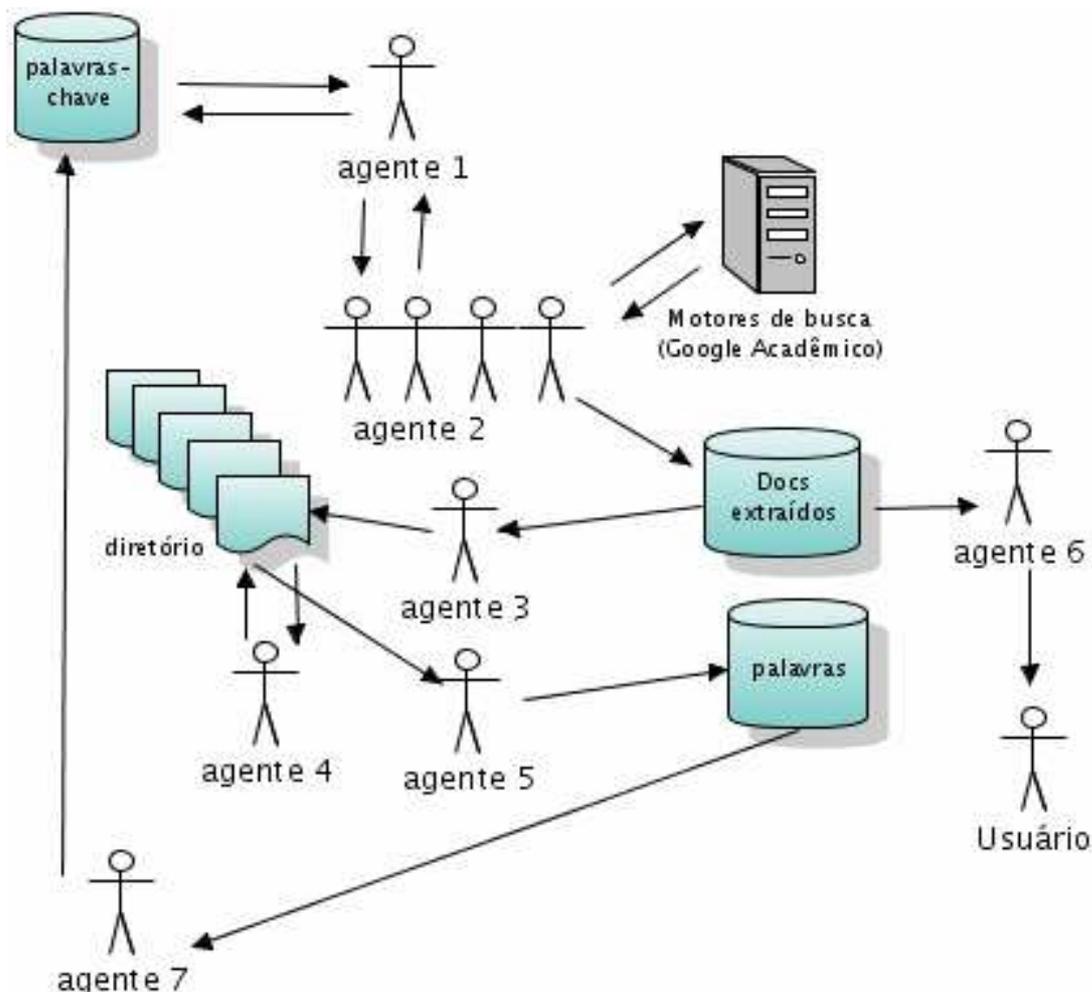


Fig. 7 – diagrama de relacionamento entre os agentes

- **Agente Palavras-chave**

Agente responsável por ler a tabela do banco de dados com as palavras-chave dos documentos armazenados.

De posse das palavras, o agente deve transmitir essa informação para o Agente Extrator, que será responsável por submeter as palavras e descobrir os documentos resultantes. O agente Palavraschave ficará constantemente verificando a tabela com as palavras-chave e enviando para o agente Extrator. A tabela do SisRecAC com as palavras-chave é alimentada no momento em que um documento é enviado (upload) pelo usuário.

- **Agente Extrator**

Esse agente deve receber mensagens do agente Palavras-chave contendo as palavras-chave e o código do documento do qual foram extraídas as palavras. De posse dessas informações, o agente deverá submetê-las a outros motores de busca, extrair os links resultantes e gravar em uma tabela. Esse agente busca apenas links para arquivos pdf tendo em vista que normalmente este tipo de arquivo se constitui em um padrão para publicações científicas. Essa tabela será utilizada por outro agente para comparar a lista de documentos originais e verificar se houve alterações.

- **Agente Download**

A função deste agente é realizar o download dos links resultantes das pesquisas realizadas nos motores de busca. Esse agente fica analisando a tabela de links, que é alimentada pelo agente Extrator, verificando se o arquivo já está no servidor. Se o arquivo não estiver no servidor, o agente deve realizar o *download* para um diretório específico do servidor.

- **Agente Pdf2txt**

Esse agente deve analisar o diretório dos arquivos pdf e verificar se cada arquivo já tem o seu correspondente no formato texto. Os arquivos que não estiverem no formato texto serão convertidos pelo agente. É necessário realizar a conversão para arquivo texto para que possam ser extraídas palavras-chaves desses documentos.

- **Agente Txt2table**

A função deste agente é transferir o arquivo texto (resultante de uma pesquisa) para uma tabela do banco de dados, onde cada palavra ocupará uma tupla da tabela. Estando o arquivo armazenado em uma tabela onde cada palavra é uma tupla será possível realizar posteriormente outras análises léxicas sobre o documento, como por exemplo, a descoberta de expressões.

- **Agente Diferenças**

Esse agente será responsável por analisar a lista de recomendação original, que foi gravada no momento do *upload* do arquivo base, com a nova lista de recomendação que foi descoberta pelos agentes. Se o agente descobrir alguma diferença na lista (entrada ou saída de documentos), ele deve avisar ao usuário sobre essas alterações. Os avisos ou alertas sobre eventuais alterações serão efetivados através do envio de *e-mails* e/ou apresentados no momento em que o usuário realizar o *login* no SisRecAC

- **Agente InteressesGerais**

O **Agente InteressesGerais** é responsável por descobrir um padrão de interesse do usuário, através de uma análise geral das palavras-chave de todos os documentos submetidos pelo usuário. Seria uma lista de recomendação baseada nas palavras-chave do conjunto de todos os *uploads*. A partir do momento que esse agente descobrir essas palavras-chave, deverá gravá-las na base de dados e fazer essas sejam submetidas ao motor de busca e uma nova lista seja encontrada e periodicamente seja avaliado se houve alteração, ou seja, entrará no fluxo normal de trabalho dos demais agentes.

Além de descobrir as palavras-chave resultantes da análise de todos os documentos enviados, o agente InteressesGerais será responsável também por descobrir o conjunto de palavras-chave de todos os documentos resultantes de pesquisas e que foram avaliados como “relevantes”. A intenção da pesquisa com esse agente é avaliar se ele conseguirá encontrar informação relevante para o usuário, não em áreas de interesse específicas do usuário, mas documentos que eventualmente combine mais de uma área de interesse deste usuário.

7. CONCLUSÃO

O desenvolvimento do SisRecAC foi e está sendo realizado com o objetivo de avaliar diversos métodos de extração de palavras-chave no sentido da relevância para o usuário dos documentos recuperados a partir de cada método. Na medida em que o projeto evolui, outras oportunidades são detectadas e novos recursos estão sendo planejados e incorporados ao sistema. A utilização de agentes autônomos confere ao SisRecAC

funcionalidades que o tornam mais efetivo enquanto ferramenta de recuperação, pois o sistema estará a serviço do usuário de forma pró ativa, constantemente descobrindo novos documentos que provavelmente serão de interesse deste usuário. O SisRecAC foi desenvolvido em PHP e utiliza intensivamente o banco de dados PostgreSQL. Para implementar os agentes será utilizada a metodologia baseada na plataforma JADE [4]. A tecnologia de agentes inteligentes deve proporcionar ao SisRecAC mais interatividade com seus usuários e impedir que estes esqueçam da existência da ferramenta, tendo em vista que periodicamente, de forma autônoma, os usuários receberão informações sobre novos artigos similares aos que foram armazenados e provavelmente relevantes para o usuário.

Assim como os demais artigos recomendados pelo SisRecAC, os descobertos pelos agentes terão sua relevância avaliada pelos usuários. Atualmente o sistema encontra-se na fase de levantamento e análise de requisitos e definição da plataforma para desenvolvimento do sistema multiagente.

Futuramente será desenvolvido um agente para descobrir artigos científicos a partir do cruzamento de diferentes categorias. Exemplificando, esse agente será responsável por trazer artigos que tratem de assuntos com “Informática na Educação” ou “Informática na Medicina” quando a palavra chave for “informática”. Além disso, o sistema poderá avaliar outros tipos de arquivos e não apenas arquivos do tipo pdf.

8. REFERÊNCIAS

- [1] GIRARDI, R. Engenharia de Software baseada em Agentes. Anais do IV Congresso Brasileiro de Ciência da Computação (BCOMP 2004), Ed. UNIVALI, pp. 913-937. Itajaí, Santa Catarina, Brasil, 08 a 12 de outubro, 2004.
- [2] GOLDBERG, D. Nichols, D., Oki, B. M., and Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (Dec.1992), 61—70
- [3] JENNINGS, N. R. “Cooperation in Industrial Multi-agent Systems”, World Scientific, 1994.]
- [4] NIKRAZ, Madig; Caire, Giovanni; Bahri,A. Parisa. A Methodology for the Analysis and Design of Multi-Agent Systems using JADE. School of Engineering Science and Parker Center, Murdoch University, Rockingham, Australia. Telecom Italia Lab, Turin, Italy.
- [5] Resnick, P. and H. Varian (1997). “Recommender Systems (introduction to special section).” *Communications of the ACM* 40(3): 56-58.
- [6] WOOLDRIDGE, M. Intelligent Agents, In: Multi-Agent Systems - A Modern Approach to Distributed Artificial Intelligence, G. Weiss (ed.), The MIT Press, 1999.