

Navigator: Uma Ferramenta para Colaboração Oportunística

Carlos Eduardo R. de Mello, Diogo Krejci de Souza, João A. de Avellar Menezes,
Leandro Carreira Marques, Marcos da Silva Ferreira,
Adriana S. Vivacqua, Jano M. de Souza

COPPE/UFRJ, Programa de Engenharia de Sistemas e Computação,
Universidade Federal do Rio de Janeiro,
Cidade Universitária, Rio de Janeiro, Brazil

diogokrejci@yahoo.com.br,
{carlosmello, jmenezes, leandrom,
marcos, avivacqua, janoo}@cos.ufrj.br

Resumo. Este relatório descreve os conceitos subjacentes e a implementação de uma ferramenta de apoio ao trabalho cooperativo chamada Navigator. A ferramenta utiliza conceitos de percepção (*awareness*), colaboração oportunística e similaridade de perfis para melhorar a sinergia entre indivíduos com interesses semelhantes. Uma análise de sistemas semelhantes mostrou algumas deficiências como não apoiar interações espontâneas, apenas pré-definidas. Na maioria dos casos, isto ocorre por que as recomendações não se baseiam nos interesses dos usuários. Apresentamos um framework baseado em agentes para incentivar e apoiar a colaboração não planejada. Através do casamento de perfis de usuários, Navigator encontra oportunidades de colaboração que teriam passado despercebidas.

Abstract. This paper describes the underlying concepts and implementation of a tool to support collaborative work called Navigator. Navigator applies concepts such as awareness, opportunistic collaboration and profile similarity to improve the synergy among people with similar interests. An analysis of related systems showed deficiencies such as not supporting spontaneous interactions, only pre-defined ones. In most cases, this happens because recommendations

are not based on users' interests. We present an agent-based framework to encourage and support unplanned cooperation. By matching users' profiles, Navigator reveals new opportunities for collaboration that went unnoticed before.

1 Introdução

A inserção do computador nos ambientes de trabalho, o crescimento da Internet e o aumento da velocidade das redes de computadores contribuíram para que as pessoas de um mesmo grupo de trabalho pudessem realizar suas atividades em espaços geográficos distintos. O uso de ferramentas para dar suporte a atividades de colaboração nas organizações contribui para que os usuários sejam inseridos cada vez mais em ambientes de trabalho virtuais colaborativos. Portanto, estes ambientes devem prover elementos de percepção de forma a permitir a execução de tarefas colaborativas, principalmente onde a comunicação face a face não ocorre.

1.1 Descrição do problema

A percepção dentro de um ambiente envolve vários aspectos cognitivos relativos à habilidade humana. Enquanto a interação entre pessoas em uma conversa face a face ocorre de forma natural, visto que sentidos como visão e audição estão disponíveis em sua plenitude, nos ambientes virtuais o mesmo não ocorre. Quando há uma tentativa de fornecer suporte à percepção em ambientes virtuais [15], estes tendem a diminuir a comunicação informal entre as pessoas. Segundo [8], as formas de comunicação informal são fundamentais para a viabilização da colaboração, em especial para a colaboração oportunística – colaboração onde os participantes utilizam suas percepções dos outros para identificar oportunidades de colaboração entre eles.

A colaboração oportunística pode ser vista de duas formas diferentes. Freqüentemente pessoas estão trabalhando em projetos similares ou estudando uma mesma literatura, sem sequer se darem conta disto. Aqui existe uma oportunidade de colaboração entre essas pessoas, mas que dificilmente é notada. Para isso é necessário que elas possuam uma percepção das atividades das outras pessoas. Outra forma de colaboração oportunística se dá quando uma pessoa necessita encontrar algum especialista para obter ajuda em determinada atividade. Encontrar um especialista

geralmente é difícil e pode levar muito tempo, pois é necessário ter uma percepção do background de conhecimento de outras pessoas.

Outra questão a ser levada em consideração é a Internet. Embora esta possua um grande potencial para facilitar colaboração entre seus milhões de usuários. Este potencial freqüentemente não é utilizado pois não há uma forma adequada de prover a percepção necessária para colaborações oportunísticas.

Portanto, o objetivo deste estudo é tentar, através do desenvolvimento de uma ferramenta, propor uma solução para dar suporte às atividades de colaboração oportunística. A ferramenta proposta detecta quais usuários estão realizando atividades semelhantes, através de uma análise do conteúdo dos documentos e do tempo de utilização dos programas pelos usuários.

1.2 *Organização do trabalho*

Este trabalho encontra-se organizado em cinco seções. A segunda seção apresentará uma breve revisão dos conceitos envolvidos neste trabalho, e também uma análise de algumas ferramentas e trabalhos relacionados aos mesmos. A terceira seção apresentará a solução proposta e o protótipo da ferramenta implementada. A quarta seção apresentará um plano de experimento para o protótipo desenvolvido. Finalizando, a quinta seção apresentará uma breve discussão sobre os resultados obtidos e os possíveis trabalhos futuros.

2 Revisão de literatura

Nesta seção são apresentadas as principais teorias envolvidas no desenvolvimento da ferramenta proposta para dar suporte à colaboração oportunística. Também serão apresentadas soluções semelhantes ou complementares propostas por outros autores encontrados na literatura.

2.1 Principais teorias envolvidas

2.1.1 Percepção (Awareness)

Para que atividades colaborativas suportadas por computador (CSCW) sejam realizadas de maneira efetiva é necessário buscar meios para suportar adequadamente o trabalho em equipe. O conceito de percepção (*awareness*) envolve saber quem está a sua volta, o que está acontecendo e quem está falando com quem [7].

A percepção constitui o ponto vital para o trabalho colaborativo, sem o qual este não pode ser bem coordenado e perde em qualidade e eficiência [26]. Portanto, faz-se necessário que ferramentas de apoio ao trabalho colaborativo tenham meios de suporte à percepção.

A implementação do suporte à percepção nas ferramentas de apoio a atividades de colaboração pode ser feita de maneira ativa ou passiva. Na ativa, os usuários declaram na ferramenta o que estão realizando. Enquanto na passiva, a ferramenta coleta os dados dos usuários sem a intervenção destes [7].

2.1.2 Interações informais e a colaboração oportunística

Atividades colaborativas são baseadas nas interações entre as pessoas. Segundo Kraut [9], existem quatro categorias de interações: planejadas, intencionais, oportunísticas e espontâneas. As interações planejadas são as reuniões formais marcadas. As interações intencionais são quando uma pessoa procura explicitamente outra. As interações oportunísticas são antecipadas por uma pessoa, mas ocorrem apenas quando as pessoas se encontram ao acaso. As interações espontâneas não são antecipadas por ninguém e ocorrem oportunisticamente.

As interações informais, isto é, interações não-planejadas (intencionais, oportunísticas e espontâneas), ocorrem no dia a dia de um grupo que compartilha o mesmo espaço físico de trabalho. Estas interações são mais freqüentes que as interações planejadas, contribuindo para que as pessoas que se encontram no mesmo espaço tenham uma maior percepção do que está ocorrendo e do que as outras pessoas estão fazendo [8][9]. Há uma noção maior do contexto no qual a pessoa está inserida,

aumentando as chances de ocorrerem novas oportunidades de colaboração entre as pessoas [1].

No entanto, em um grupo de trabalho que se encontra distribuído essas interações informais diminuem. Isto acaba diminuindo também a possibilidade de novas colaborações, devido à perda da percepção imposta pelos novos meios de comunicação que proporcionam as interações entre as pessoas [14].

Portanto, é fundamental que ferramentas de suporte à colaboração em grupo tentem suprir ao máximo a carência da percepção decorrente das interações informais. Isto ocorre através do fornecimento de informações sobre o que os outros estão realizando e identificando possíveis oportunidades de colaboração.

Infelizmente, a maioria das ferramentas desenvolvida para dar suporte ao trabalho colaborativo foca apenas nas interações formais e planejadas. Poucas ferramentas focam nas categorias de interações informais e oportunísticas [14].

2.1.3 Similaridade de interesses

Assume-se que, se dois usuários possuem documentos similares entre si, então eles possuem interesses em comum. Existem, na literatura, diversas formas para se encontrar esta similaridade entre os documentos, dentre as quais se destaca o método de encontrar o co-seno entre vetores. Neste método, a similaridade dos documentos é calculada em relação aos vetores de termos e suas respectivas freqüências. Para tanto, utiliza-se a fórmula [16]:

$$\text{Similaridade}(X, Y) = \frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i} \sqrt{\sum_{j=1}^t y_j}},$$

onde t representa o número total termos nos vetores X e Y , e x_i e y_i representam as freqüências do termo i nos vetores X e Y , respectivamente. O valor da similaridade varia entre 0 e 1. Quanto mais próximo de 0, mais distintos são os usuários, enquanto que mais próximo de 1, mais interesses em comum eles possuem.

2.1.4 Sistemas de recomendação

Muitas vezes precisamos tomar decisões onde não possuímos informações suficientes sobre determinado assunto. Diante de alternativas que não são familiares, procuramos recorrer a recomendações de outras pessoas, como amigos, familiares, jornais e revistas. Isto é, utilizamos a experiência de outras pessoas para nos ajudar a tomar decisões.

Os sistemas de recomendação têm por objetivo auxiliar e aumentar esse processo social através de algoritmos de filtragem de informação. Um típico sistema de recomendação recebe e agrupa os dados das recomendações das pessoas e direciona os resultados para as pessoas apropriadas [3]. De posse dos resultados fornecidos pelo sistema, as pessoas poderão tomar suas decisões baseadas nas recomendações de outras.

Os sistemas de recomendação podem ser categorizados utilizando cinco dimensões. Primeira, o conteúdo avaliado pode ser um simples bit (recomendado ou não) ou uma anotação textual não-estruturada. Segunda, a entrada dos dados das recomendações pode ser feita explicitamente no sistema ou de maneira passiva, através de coletores de informação. Terceira, as recomendações podem ser anônimas ou nomeadas (com o nome do remetente). Quarta, a maneira como agregar as informações. Esta talvez seja a área mais rica para se explorar. E, finalmente, a quinta é como fazer a filtragem dessas avaliações de acordo com o receptor da informação [3].

2.2 Sistemas correlatos

Diversas ferramentas foram propostas e desenvolvidas visando resolver o problema da percepção em um ambiente de trabalho.

Algumas abordagens se baseiam na criação da percepção do espaço físico, através de transmissão de imagens e vídeos, permitindo que usuários separados pela distância física possam ver uns aos outros a qualquer momento. Através desta abordagem temos uma reprodução da percepção obtida naturalmente em um ambiente

físico comum, pois as pessoas podem, a qualquer momento, ver o que os outros estão fazendo através dessas imagens.

Como exemplos de sistemas que usam a abordagem do compartilhamento de imagens de um ambiente físico para a percepção podemos citar os projetos Polyscope e Portholes [7]. O projeto Polyscope, desenvolvido e utilizado na Xerox EuroPARC, visava a criação de um espaço de mídia para prover a percepção do ambiente. O sistema capturava imagens regularmente atualizadas de diversos locais e as apresentava na tela de uma estação de trabalho. Essas imagens mostravam atividades em áreas públicas e escritórios. A partir da experiência bem sucedida deste projeto surgiu o Portholes. O Portholes tinha por objetivo estender o espaço de mídia através de uma grande distância física. O projeto consistiu em criar um espaço de mídia ligando os dois grandes complexos de pesquisa da Xerox PARC e da Xerox EuroPARC.

Uma outra abordagem é focada em um objeto do trabalho colaborativo, como um documento sendo escrito por várias pessoas colaborativamente. Alguns exemplos desta abordagem são o Quilt, o PREP e o GROVE [6]. Essas ferramentas consistem em editores de texto colaborativos com algumas funcionalidades extras para facilitar a comunicação do grupo. Basicamente essas ferramentas fornecem a percepção através do próprio documento que está sendo desenvolvido.

Não podemos deixar de citar também ferramentas de mensagem instantânea, como MSN Messenger, ICQ e outros. Nessas ferramentas o usuário possui uma lista de contatos onde são exibidas informações a respeito do status de cada um deles, como também é possível escrever frases personalizadas. Tudo isso contribui para que sejam criadas e disponibilizadas informações que possam ajudar na criação da percepção por parte do usuário. Com o tempo outras funcionalidades foram sendo adicionadas a este tipo de aplicação, como mensagens de voz e transmissões de vídeo. Apesar disso, essas ferramentas ainda possuem suas limitações, como problemas de resolução de vídeo, atrasos de áudio etc.

A maioria das abordagens existentes para ferramentas de voz e vídeo possui diversas limitações. É necessário que as pessoas estejam *online* em determinado

“lugar” para poderem ver outras pessoas. Geralmente falta a percepção necessária para ajudar as pessoas a interagir e a maioria das ferramentas não oferece meios eficientes para a transição entre a visão e a interação. Talvez o fator mais importante seja que a maioria das ferramentas desenvolvidas suporta a percepção apenas em um grupo de trabalho já formado. Assim sendo, elas não oferecem a oportunidade de descoberta de novos parceiros para estes iniciem uma colaboração.

A ferramenta Piazza [27] permite que uma pessoa, enquanto estiver executando suas tarefas, possa visualizar outras pessoas que estão executando tarefas similares. É usado o conceito de “proximidade” entre as pessoas para identificar a similaridade entre elas. A proximidade depende de três dimensões, que são as pessoas estarem manipulando um mesmo dado, ao mesmo tempo, utilizando uma mesma aplicação. Logicamente que essas regras não são totalmente rígidas. Assim sendo, pessoas que estiverem manipulando um mesmo dado, mas utilizando aplicações diferentes também poderiam ser consideradas como próximas. A ferramenta Piazza é composta por outras cinco ferramentas: Encounter, Gallery, People Browser, Glance, Project Rooms. Esta é uma das poucas ferramentas que tem uma grande preocupação com a questão da percepção como forma efetiva de colaboração.

A ferramenta chamada Community Bar [1] mostra uma barra com as fotos e nomes das pessoas que fazem parte dos seus contatos e disponibiliza um perfil com informações sobre estas pessoas. Através desta barra, o usuário consegue ter uma percepção do que cada um de seus contatos está fazendo e, caso seja interessante a ele, poder iniciar uma interação similar às encontradas em salas de bate-papo.

Em [4], é feito um estudo sobre diferentes abordagens da utilização de agentes para facilitar a cooperação. É destacada a importância do retorno (*feedback*) dado ao agente pelo usuário em seu aprendizado. Também é possível buscar assuntos que possam interessar a determinado usuário. Isto é feito analisando os perfis que combinam com o do usuário e encontrando assuntos comuns que interessem a eles, que são, então, apresentados ao usuário.

Em [2], vemos o desenvolvimento de um sistema multiagentes que tem como objetivo integrar pessoas com interesses similares em um ambiente virtual. Para

isso, é feita uma análise de textos (e-mail, artigos, arquivos do usuário) e, a partir desta análise, é feito um casamento (*matchmaking*) entre o perfil encontrado e o perfil de outros usuários presentes no sistema, com objetivo de encontrar usuários com perfis similares.

Duas abordagens distintas podem ser utilizadas em ferramentas multi-agentes e de casamento (*matchmaking*) de interesses. Em algumas ferramentas os usuários são clientes de um servidor central, que se encarrega de fazer o match entre eles. Alguns exemplos desta abordagem são o Webhoun/Webdoggie [11] e HOMR/Ringo/Firefly [12].

Outras ferramentas utilizam uma abordagem distribuída, como, por exemplo, em [13], onde Kautz, Milewski e Selman descrevem um protótipo de uma ferramenta para localização de expertise em uma grande companhia. Nesta ferramenta, cada usuário especifica que outros usuários possam vir a ser contatos apropriados, e então os agentes fazem um processo de encadeamento de referências para fazer recomendações de contatos. Nesta abordagem é necessário que o usuário forneça explicitamente informações.

3 Proposta de solução

Este trabalho propõe o uso de uma aplicação multi-agentes desenvolvida sobre uma arquitetura peer-to-peer (P2P) para possibilitar e dar suporte à colaboração oportunística. Os agentes serão os responsáveis por coletar as informações disponibilizadas pelo usuário em sua máquina e com elas elaborar um perfil que será disseminado na rede P2P.

Nesta seção será apresentado o protótipo da ferramenta desenvolvido como parte deste trabalho. Serão descritos os objetivos da ferramenta, a tecnologia utilizada no desenvolvimento, bem como a arquitetura concebida para o protótipo, que foi planejada visando facilitar futuras extensões. Primeiramente serão analisados os conceitos gerais da solução proposta.

3.1 Os conceitos básicos da ferramenta

Antes de ser apresentada a descrição mais detalhada da ferramenta são definidos os conceitos nos quais o Navigator foi concebido.

3.1.1 Perfil do usuário

Para comparar dois usuários criou-se um perfil para o usuário baseado na lista de termos (tabela 1) e na lista das tarefas (tabela 2). A lista de termos é composta pelos termos encontrados nos documentos do usuário, seus respectivos documentos e freqüências. A lista de tarefas é composta pelos aplicativos, documentos, total de horas que os documentos ficaram abertos e a data e hora dos seus últimos acessos.

Tabela 1 – Lista de termos

Lista de termos		
Termos	Documentos	Freqüências
Navigator	Navigator.doc	4
collaborative	Navigator.doc	9
vector	TextMining.pdf	5
steamming	TextMining.pdf	2
ranking	TextMining.pdf	1

O perfil possui um identificador único, o nome do usuário, uma lista geral de termos acumulada, uma lista geral de termos recentes e a lista de tarefas.

Tabela 2 – Lista de tarefas

Lista de tarefas			
Aplicativos	Documentos	Tempo acumulado	Última execução
Microsoft Word	Navigator.doc	15 horas	11/09/2006 as 8:00
Adobe Acrobat	TextMining.pdf	2 horas	01/07/2006 as 17:21

Para construir a lista geral de termos recentes multiplicamos as suas freqüências por um valor em função do tempo de uso do documento aos quais o termo pertence. Assim os termos dos documentos que foram mais utilizados serão privilegiados em relação aos menos utilizados.

A construção da lista geral de termos acumulada é similar à da lista geral de termos recentes. Porém, ao invés de privilegiarmos os termos com maior tempo acumulado, estaremos privilegiando os termos lidos ou escritos mais recentemente.

3.1.2 A similaridade

Partindo-se do pressuposto que os documentos que o usuário possui são representativos do seu background de conhecimento e de seus interesses. Assim sendo, uma similaridade entre documentos irá implicar em uma similaridade de interesses entre usuários.

A similaridade entre os usuários é calculada através da lista de geral dos termos dos documentos e suas respectivas freqüências. Podemos ter listas gerais de termos de dois tipos: recente ou acumulada.

O cálculo da similaridade é feito utilizando a técnica dos co-senos com as listas gerais de termos acumulada e recente de dois usuários. Onde obteremos como resposta um valor entre 0 e 1. Quanto mais próximo de 1, mais similares são os perfis dos usuários e quanto mais próximo 0, mais distintos.

Uma grande vantagem de utilizar esse modelo que relaciona termos com o tempo que o documento foi utilizado é que podemos tentar privilegiar os termos que mais são lidos pelos usuários.

3.2 *Descrição da ferramenta*

3.2.1 Objetivos

A ferramenta irá auxiliar o usuário a encontrar novas pessoas com quem seja possível obter alguma colaboração. Para isso, ela irá processar perfis de outros usuários obtidos na rede e fazer um match entre esses perfis e o perfil do próprio usuário. Baseado no resultado deste processo, a ferramenta exibe os outros usuários encontrados na rede em uma lista, exibindo também o grau de similaridade identificado entre os perfis.

Por exemplo, se duas pessoas estão trabalhando (lendo, escrevendo) em assuntos similares, elas são indicadas uma a outra. Essa indicação decorre de o seu grau de similaridade recente (que mede a similaridade de seus trabalhos atuais) ser alto (próximo a 1).

A ferramenta também cobre o caso de indicar especialistas às pessoas trabalhando em determinado assunto. Esta indicação ocorre devido à similaridade acumulada entre os usuários ser alta. Ou seja, embora o especialista não esteja trabalhando atualmente no assunto, essa lista leva em consideração todo o histórico do usuário.

A ferramenta exibe, como podemos ver na Figura 1, os usuários para os quais alguma similaridade foi encontrada, indicando o grau dessa similaridade. Para a similaridade recente são exibidas estrelas no ícone do usuário de acordo com o nível de similaridade. Usuários que possuírem maior similaridade irão possuir mais estrelas. A similaridade acumulada é representada pela cor do ícone. As cores utilizadas são amarelo, laranja e vermelho, sendo esta a ordem crescente do grau de similaridade. Nas tabelas 3 e 4 podemos ver o grau de similaridade e suas respectivas estrelas e cores.



Figura 1 - Visão de Contatos

O usuário pode entrar em contato com outros usuários, utilizando uma funcionalidade de Chat, dando início assim a uma possível colaboração, caso seja interesse de ambas as partes.

Tabela 3 – Similaridade recente

Número de estrelas	Similaridade
	0% - 25%
	25% - 50%
	50% - 75%
	75% - 100%

Tabela 4 – Similaridade acumulada

Cor da estrela	Similaridade
	0% - 33%
	33% - 66%
	66% - 100%

A aplicação disponibiliza ainda uma visão de radar (Figura 2), onde o usuário é exibido no centro e os outros usuários da rede podem ser visualizados ao redor, sendo que a distância será dada de acordo com a similaridade identificada. Ou seja, usuários com maior similaridade serão exibidos mais próximos ao centro. O radar também pode apresentar a visualização dos usuários por áreas de interesses, onde cada setor do radar representa uma área. Essas áreas de interesses podem ser obtidas através do processo de clusterização dos documentos indexados. No protótipo implementado, no entanto, a clusterização não é feita, sendo uma possibilidade de trabalho futuro.

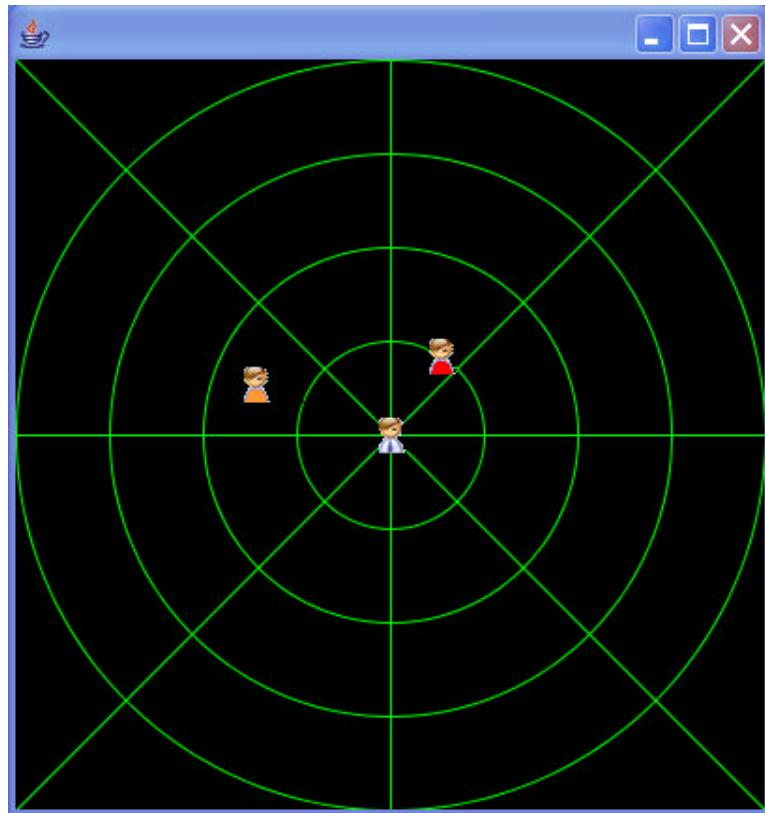


Figura 2 - Visão Radar

Na figura 3 temos uma visão geral da interface da ferramenta. Podemos ver a lista de contatos onde para cada usuário é exibida uma lista de documentos, além das informações de similaridade já discutidas. Estes documentos exibidos são aqueles que estão sendo utilizados por aquele usuário naquele exato momento. Isto contribui para aumentar a percepção que se tem sobre as atividades de cada usuário da lista. Na mesma figura, podemos ver também a interface de Chat, que será o canal de comunicação utilizado para que usuários possam de fato dar início a um processo de colaboração. Nesta primeira versão, a funcionalidade é extremamente básica, mas com o tempo podem ser inseridas novas funcionalidades que permitam uma maior qualidade para a interação entre os usuários. No entanto, essas funcionalidades já são extensamente estudadas na literatura de trabalho colaborativo e fogem ao escopo

deste trabalho, que é destinado especificamente a uma etapa anterior a atividade colaborativa em si.

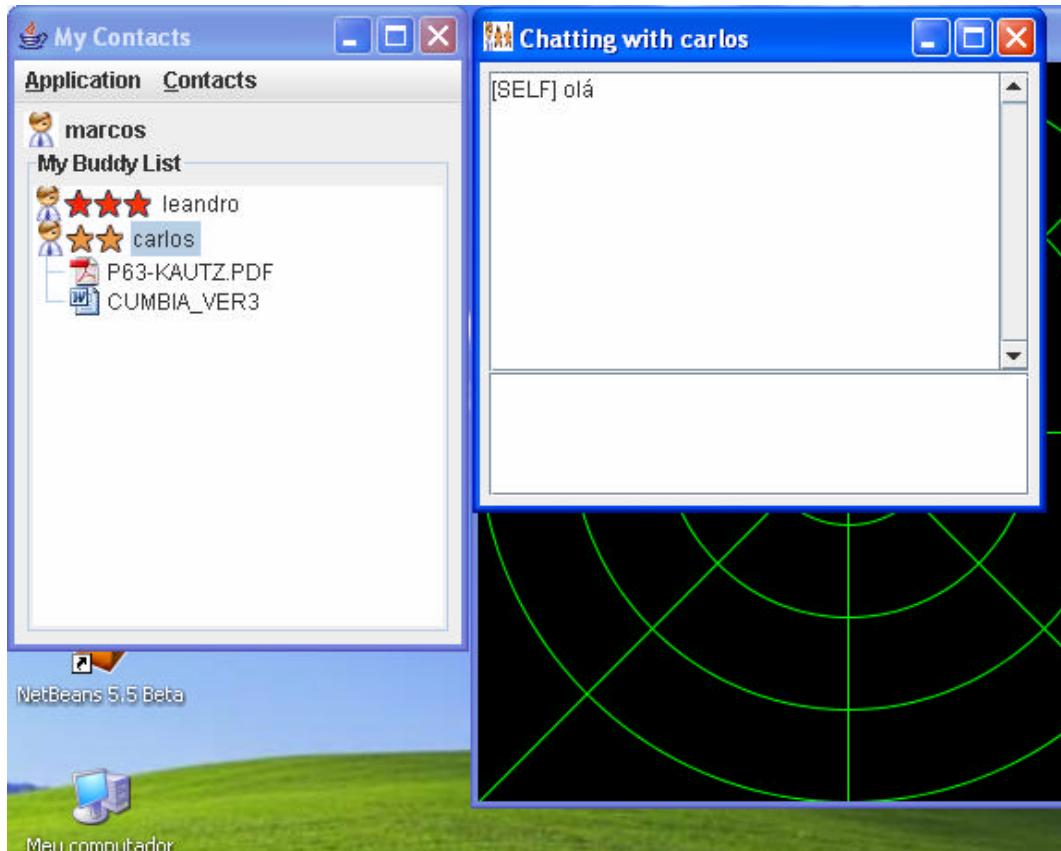


Figura 3 - Visão geral da ferramenta

3.2.2 Tecnologia utilizada

A implementação foi escrita em Java, utilizando diversas APIs específicas, como a biblioteca de indexação de busca Lucene. Foram utilizadas bibliotecas para decodificar tipos de arquivos minerados. Para PDFTM foi utilizada a biblioteca PDFBox [24]. Para formatos de arquivos do Microsoft OfficeTM foi utilizada a biblioteca Jacob [25].

Para a implementação do protótipo da ferramenta foi escolhida uma arquitetura peer-to-peer baseada em agentes. A ferramenta foi desenvolvida sobre a plataforma COPPEER. A seguir é dada uma breve descrição dessas tecnologias e as motivações pelas quais estas foram escolhidas neste trabalho.

3.2.2.1 Agentes

Segundo [18], agentes são artefatos de software que realizam um conjunto de tarefas para um usuário ou programa com algum nível de autonomia para tomar certas decisões baseadas em algum conhecimento sobre os objetivos que se pretende alcançar, que podem ser definidos de forma explícita ou implícita. De acordo com [23], os agentes de software podem interagir com outros agentes gerando um sistema multi-agente e, como visto em Jennings e Wooldridge apud [23], com este tipo de sistema pode-se reduzir o custo e a complexidade associados ao desenvolvimento de sistemas distribuídos.

Agentes podem ser separados em dois grupos, de acordo com sua mobilidade: Agentes Móveis ou Estacionários. Segundo [17], um agente que não fica restrito ao sistema onde começou sua execução é um agente móvel. Ele tem a habilidade de se transportar de um sistema para outro através de uma rede. Essa habilidade é especialmente útil quando um agente se move para um sistema que contém um objeto com o qual ele precisa interagir e, então, usufruir dos benefícios de estar no mesmo host ou rede desse objeto.

O uso de agentes traz diversas vantagens. Eles podem trabalhar de forma assíncrona aguardando outros agentes ou recursos se tornarem disponíveis para prosseguirem com a sua execução. Agentes interagem através de mensagens escritas em uma área compartilhada, o que faz com que eles possuam um acoplamento fraco, permitindo extensão e modificação dos mesmos com um custo reduzido. Além disso, eles podem ser robustos e encapsular protocolos, simplificando o trabalho de usuários ou outros agentes que invocam serviços.

3.2.2.2 Sistemas *Peer-to-peer*

O termo *peer-to-peer* (P2P), segundo [19], se refere à classe de sistemas e aplicações que utilizam recursos distribuídos para realizar algum processamento de forma distribuída. Sistemas P2P têm ganhado notoriedade nos últimos anos com a popularização dos computadores domésticos.

Peers podem ser entendidos como unidades autônomas, isto é, unidades que não são controladas por uma entidade central, seja ela um usuário ou um componente de um sistema. Apesar disso, *peers* dependem de outros *peers* para realizar suas funções, tais como obter informações, realizar computações ou encaminhar mensagens, de modo que o sistema inteiro se beneficie.

As contribuições que a adoção de uma arquitetura P2P pode trazer para este trabalho incluem: ausência de um ponto central de controle, tornando o sistema muito mais escalável e evitando que o sistema seja vulnerável em um único ponto de falha; simplificação de estruturas de controle e comunicações, pois com a possibilidade dos computadores se conectarem e se comunicarem diretamente, cada *peer* é responsável por sua própria manutenção de topologia; as mensagens não precisam ser roteadas através de uma entidade central.

Em [22] foi proposto um método para avaliar o quanto apropriado é o uso de uma arquitetura P2P para um determinado sistema. Analisando as características deste trabalho com o método de Roussopoulos, vemos que fatores como: poucos recursos orçamentários, grande relevância dos recursos locais para os vizinhos e tolerância a falha no compartilhamento desses recursos são bons candidatos a sistemas P2P; enquanto que a baixa confiança dos *peers* nos dados recebidos dos seus vizinhos é um fator negativo para a adoção dessa tecnologia. Porém, considerados esses fatores concluído que é favorável o uso de uma arquitetura P2P nesse sistema.

3.2.2.3 COPPEER

O COPPEER é um projeto em desenvolvimento por alunos da linha de Banco de Dados do Programa de Engenharia de Sistemas e Computação da Universidade Federal do Rio de Janeiro, criado por Mutaleci. Detalhes do framework

COPPEER pode ser encontrado em [20][21]. Apresentamos neste trabalho apenas alguns conceitos básicos.

Em sua versão atual, o COPPEER é um framework para o desenvolvimento de aplicações *peer-to-peer*, provendo a infra-estrutura que os desenvolvedores de tais aplicações necessitam. Sua implementação é baseada nos uso de agentes. Sua arquitetura é composta de múltiplas camadas. O COPPEER provê uma rede dinâmica de agentes que atuam em paralelo, onde cada um deles pode agir e reagir a ações dos demais agentes. O comportamento do sistema é resultado da colaboração dos agentes.

Quando uma instância do CoppeerCAS é iniciada em um *peer*, é criada uma agência. A agência é a principal entidade do CoppeerCAS e serve para gerenciar ambientes, células e agentes para as aplicações.

Cada agência pode conter múltiplas células. Uma célula é um espaço compartilhado de memória, baseado na especificação JavaSpaces [10], nela dados podem ser escritos e lidos e ela pode se conectar a outras células do mesmo ambiente.

Um ambiente é um conjunto de células interconectadas, cada uma pertencente a uma agência diferente. Uma mesma agência pode conter várias células e, assim, participar de diferentes ambientes. Desta maneira, uma célula é identificada sem ambigüidade por sua agência e o ambiente a que pertence.

A célula oferece a seus clientes uma interface que possibilita a eles: ler e escrever dados (na forma de entradas); solicitar que a célula se conecte a outra; pedir a ela que notifique o cliente quando determinados tipos de dados forem escritos nela. Qualquer objeto de uma aplicação, bem como qualquer agente, pode ser cliente de uma célula.

Uma entrada é um objeto que contém dados e pode ser escrito em (ou lido de) uma célula. Além disso, uma entrada pode se propagar para células vizinhas àquela em que foi escrita. É provida uma interface para que os desenvolvedores das aplicações definam o comportamento de propagação das entradas de suas aplicações.

Agentes funcionam como agentes de software tradicionais. Neste caso, eles estão associados a um determinado ambiente, podendo se mover entre as células do mesmo (e portanto entre os *peers* da rede) e executar as computações necessárias.

Desenvolvedores de aplicações para o COPPEER podem determinar comportamentos para os agentes que precisarem utilizar, através da interface oferecida pelo COPPEER para esta finalidade. Assim, podem estabelecer regras para quando os agentes devem se mover e o que devem fazer no *peer* em que se encontram.

A implementação do CoppeerCAS utiliza a linguagem Java. Os agentes móveis e conexões foram feitos sobre a plataforma Aglets, e sua arquitetura é estruturada sobre um microkernel Spring.

3.3 Agentes Implementados

A aplicação possui basicamente cinco tipos de agentes implementados: mineradores, coletor de tarefas, construtor de perfil, *matchmaker* e agentes auxiliares.

Os agentes mineradores são responsáveis por buscar no disco do usuário os documentos que irão ser utilizados para a construção do perfil de usuário. Os agentes mineradores são especializados por tipos de documentos, visto que cada tipo de arquivo deve ser lido de forma diferente. A ferramenta apresenta agentes especializados em minerar arquivos do Microsoft Word™ (*DocIndexerAgent*), PDF™ (*PdfIndexerAgent*), e TXT (*TxtIndexerAgent*). O agente *DocIndexerAgent* utiliza a biblioteca Jacob para fazer a extração de texto e o agente *PdfIndexerAgent*, a biblioteca PDFBox.

Cada um desses agentes busca os arquivos com os formatos correspondentes e os indexa. Na indexação é utilizada o Lucene, uma biblioteca especializada em indexação e busca de termos em arquivos. Após cada agente indexar todo o conteúdo dos arquivos, serão extraídas as informações necessárias para a construção do perfil, estas informações serão enviadas através da célula COPPEER, para que um outro agente (*JoinerAgent*) possa utilizá-las. As informações enviadas consistem

basicamente de um vetor de termos com a sua correspondente contagem de freqüência, para cada documento lido (Figura 4).

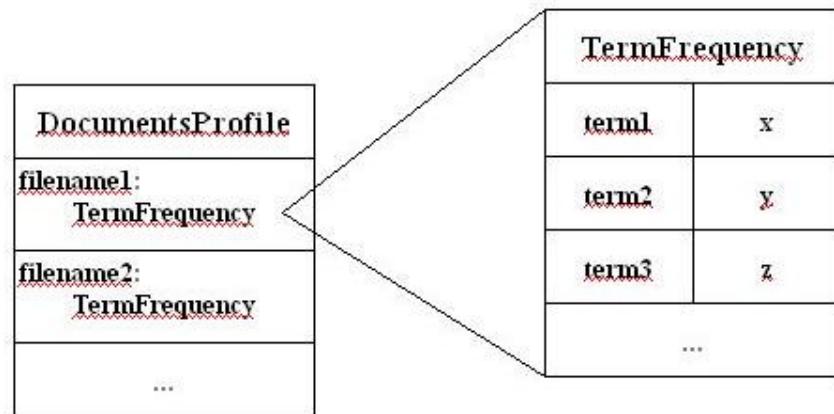


Figura 4 - Perfil de documentos minerados

Existe um agente intermediário (*JoinerAgent*) que funciona como uma ligação entre todos os mineradores e o construtor do perfil. Como existem diversos agentes mineradores gerando diferentes dados, é necessário, antes de construir o perfil, agregar e consolidar todas essas informações. Este agente irá, então, coletar todas as informações geradas pelos agentes mineradores e criar um perfil único, que então poderá ser consumido diretamente pelo construtor do perfil.

O agente coletor de tarefas (*TaskAnalystAgent*) possui o papel de monitorar os processos do sistema, capturando informações sobre a execução de certos aplicativos, como, por exemplo, Acrobat ReaderTM e Microsoft WordTM. Essas informações serão usadas para prover a percepção instantânea, ou seja, permite que o usuário possa visualizar documentos que estão sendo manipulados por outros usuários. Essas informações também são utilizadas para fazer um cruzamento com as informações provenientes do processo de mineração, com o objetivo de montar um perfil ponderado com relação aos documentos e seus respectivos tempos de utilização. O agente se encarrega de manter as informações de tempo coletadas de cada tarefa em meio persistente para que elas não sejam perdidas e, assim, a atribuição de pesos seja

coerente com uma construção de perfil do usuário ao longo do tempo, não sendo restrita a um determinado momento em que a ferramenta é utilizada.

O agente construtor de perfis (*ProfileBuilderAgent*) agrupa as informações coletadas pelos outros agentes e, com base nessas informações, construir o perfil do usuário. Este agente fará o cruzamento do perfil de documentos do usuário com as informações de tarefas fornecida pelo agente *TaskAnalystAgent* e irá fazer os cálculos para obter as listas de termos ponderados pelo tempo. Inicialmente o perfil foi concebido como uma estrutura bem simplificada, contendo informações pontuais como um identificador, um nome de usuário e as informações de tarefas e documentos minerados. As informações de tarefas irão conter a lista de documentos que estão sendo manipulados pelo usuário naquele momento. As informações de documentos que irão constar no perfil final será uma lista de termos de todos os documentos minerados, com suas respectivas freqüências agora ponderadas de acordo a utilização de cada documento. Na verdade, irão existir duas listas de termos no perfil, uma ponderada pelo tempo total de uso e outra ponderada por uso recente.

Esse perfil é então escrito na célula do COPPEER e se propaga uma vez (visto que utilizamos uma topologia em estrela, onde cada usuário está conectado a todos os demais, para a rede P2P) para todos os usuários da rede.

O agente *MatchMakerAgent* se encarrega de comparar o perfil do usuário com os perfis dos usuários dos outros peers. Existem neste ponto dois tipos de comparações (recente e acumulada): cada uma feita utilizando uma lista de termos do perfil recebido. A partir desta comparação, ele calcula os dois tipos de semelhança entre os usuários: a semelhança recente e a semelhança acumulada.

O agente que coordena a interface gráfica da ferramenta (*InterfaceAgent*) produz a visualização das oportunidades de colaboração. Para diferenciar os dois tipos possíveis de colaboração, definimos que a quantidade de estrelas que aparece ao lado do nome representa o grau de similaridade entre o trabalho atual dos usuários (similaridade recente), enquanto a cor das estrelas representa a similaridade acumulada.

3.4 Arquitetura

Na Figura 5 é apresentada uma visão geral da arquitetura utilizada no protótipo da ferramenta, estruturada com os agentes discutidos na seção anterior e a forma como eles irão interagir.

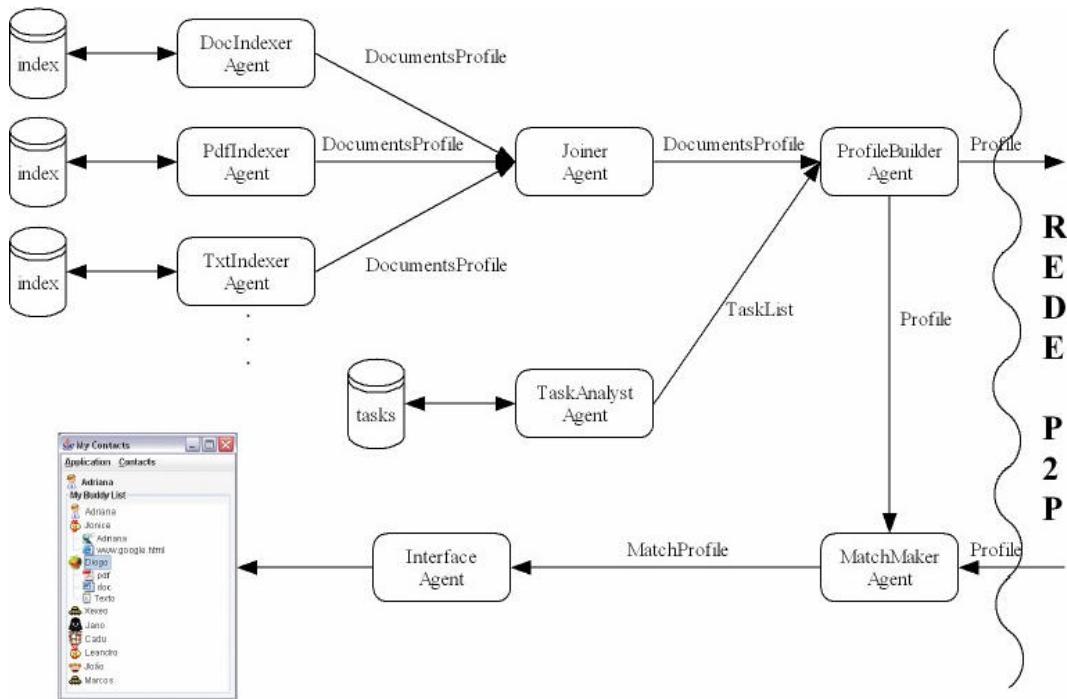


Figura 5 – Arquitetura Navigator

Pode-se notar no canto superior esquerdo os agentes mineradores. Estes agentes acessam os índices construídos para os documentos, que são separados por tipos de arquivos e a partir daí geram o perfis de documentos, que serão coletados pelo JoinerAgent. Lembrando que é possível estender a aplicação para manipular outros tipos de arquivos, bastando para isso implementar outros agentes especializados (essa possível extensão é indicada com três pontos abaixo dos agentes mineradores).

As setas nomeadas na figura representam a comunicação entre os agentes e os tipos de dados que são trocados. Todas as comunicações de dados são feitas através de escrita/leitura das entradas na célula COPPEER.

Devido a necessidade de se manter atualizado o perfil do usuário, os agentes mineradores e o agente *TaskAnalystAgent* foram projetados para executar em determinado intervalo de tempo. O intervalo padrão da aplicação é de 30 em 30 segundos, podendo ser customizado. Cada escrita na célula por parte desses agentes irá disparar novamente todo o processo de construção do perfil do usuário.

O fluxo mostra que, com exceção dos agentes mineradores, os agentes dependem da execução de um anterior para realizar suas tarefas. O agente *JoinerAgent* depende de perfis de documentos que serão recebidos dos mineradores. Para isso ele utiliza um *listener* para receber notificações de escrita na célula. Da mesma forma, o agente *ProfileBuilderAgent* necessita das informações produzidas pelo *JoinerAgent* e também utiliza um *listener*.

O agente *MatchMakerAgent* consome dois tipos de dados produzidos: o perfil gerado para o seu usuário e o perfil de um outro usuário qualquer, recebido através da rede P2P. O agente *InterfaceAgent* irá consumir os dados produzidos pelo *MatchMakerAgent* e atualizar a visão apropriadamente, fechando assim o fluxo.

4 Plano de experimento

Esta seção descreve sucintamente um plano de testes para o projeto Navigator, que permita validar a expressividade dos perfis de usuário construídos pelo sistema e do *matching* realizado.

4.1 Objetivos:

1-Verificar a validade dos perfis:

- a) Perfis recentes representam focos correntes de interesse (do usuário e de outros)
- b) Perfis acumulados representam áreas de especialidade (do usuário e de outros)

2-Verificar a validade do *matching*:

- c) Possíveis colaboradores
- d) *Experts*

3-Verificar a utilidade das informações fornecidas (sobre outros usuários e suas atividades).

Participantes: alunos (de mesmo nível de escolaridade) organizados em pequenos grupos (4-5)

Tempo: Aproximadamente 1 hora por rodada. Alternativamente, pode ser proposta uma duração maior (por exemplo, uma semana, em turnos intermitentes), caso em que a atividade seria realizada assincronamente pelos membros do grupo, para testar o valor dos históricos e aumentar a semelhança com uma situação real.

Tarefa: construção de uma enciclopédia *online* (ex. Wikipedia). Verbetes devem envolver alguma pesquisa sobre o assunto e devem ser distribuídos entre os alunos de modo que haja alguma sobreposição entre verbetes. Cada aluno receberá um conjunto de verbetes (5-10, dependendo do tempo da tarefa) que têm relação com outros que não necessariamente são seus, e guardam alguma similaridade com outros. Ao inserir links entre verbetes, os alunos devem estabelecer a natureza do link (ou seja, as relações entre conceitos são qualificadas), possivelmente inserindo um parágrafo explanatório no verbete de outros.

Recursos disponíveis: internet (IE), alguns arquivos para serem utilizados como fonte de pesquisa talvez fornecer informações conflitantes entre dois verbetes.

Ferramenta: Navigator com *chat* e análise de páginas *web*.

4.2 Hipóteses:

- A utilização recente de recursos representa (está relacionada) os focos de interesse atuais do usuário;

- A utilização de recursos (de forma geral, acumulada ao longo do tempo) indica as áreas de atuação e *expertise* do usuário. O tempo de atuação em uma área de domínio influencia no nível de conhecimento;
- *Matching* (básico): pessoas trabalhando sobre conceitos (ou páginas) similares utilizarão recursos similares;
- Colaboradores: pessoas trabalhando sobre o mesmo assunto (ou editando o mesmo recurso) podem dividir o trabalho ou trocar informações para benefício mútuo;
- *Experts*: ao trabalhar, uma pessoa pode beneficiar-se de outra que detenha conhecimentos de longo prazo (se uma pessoa utiliza conceitos utilizados por outra, pode ser direcionada para esta para efeito de clarificação ou discussão). Espera-se que o expert referenciado tenha conhecimento relacionado ao tema. (para efeitos de experimento, podem ser configurados múltiplos grupos, onde o grupo 2 referencia o trabalho do 1º, etc.).

4.3 Metododologia (para cada questão):

1-Qual a validade dos perfis?

- Perfis recentes representam focos correntes de interesse do usuário: a partir dos perfis recentes, utilizar um questionário para verificar a correlação entre estes e os interesses imediatos do usuário (por ex.: dadas as áreas de atuação, o usuário se dá uma nota para cada área identificada no perfil).
- Perfis acumulados representam áreas de especialidade do usuário: a partir dos perfis acumulados, utilizar um questionário para verificar a correlação entre os perfis e os domínios de conhecimento do usuário (vide exemplo anterior).

2-Qual a precisão do *matching*?

- Possíveis colaboradores: questionário para verificar se os dois usuários apontados estavam realmente realizando atividades complementares e haveria espaço para colaboração, junção de esforços.
- *Experts*: questionário para verificar se os usuários detinham conhecimento relevante (também pode ser usados os questionários de auto-avaliação anteriores) e se haveria possibilidade de auxílio ou discussão sobre o tema.

3-Verificar a utilidade das informações fornecidas (sobre outros usuários e suas atividades). Dois tipos de medidas devem ser feitas :

4.4 *Medidas*:

4.4.1 *Quantitativas*:

- Quantas vezes o *chat* foi iniciado e com quem? (similaridade recente ou acumulada?)
- Com que freqüência as pessoas colaboraram (dividiram esforços)?
- Com que freqüência as pessoas referenciaram outras (perguntaram ao *expert*)?

4.4.2 *Qualitativas*:

- Questionário: Quão úteis foram as recomendações? Os *experts* sabiam o que foram perguntados? Os colaboradores serviram? O interesse indicado pela atividade recente estava correto? O *expertise* indicado pela atividade acumulada estava correto? A informação auxiliou em algo (nas coisas que supostamente percepção traz – ex. auto-coordenação, contextualização, entendimento do trabalho, etc.)

5 Discussão

5.1 Pontos positivos

Uma das grandes vantagens apresentadas na ferramenta Navigator é o modo como a filtragem da informação é feita para a posterior criação do perfil do usuário. Esta ferramenta faz uma busca nos documentos do usuário com a finalidade de buscar os termos chaves existentes, suas respectivas freqüências e o tempo de utilização de cada documento pelo usuário. Este tempo de utilização é um fator fundamental na criação dos perfis, pois contribui para que os termos encontrados nos documentos mais utilizados sejam considerados mais relevantes que os encontrados em documentos com menor utilização.

Por ser uma ferramenta baseada em agentes, ela possui um baixo acoplamento, pois podemos adicionar novos agentes sem impactos significativos na aplicação. Inclusive, uma possibilidade futura de extensão da ferramenta é a implementação de agentes mineradores adicionais, especializados em outros formatos de arquivos, como PPT, HTML e outros.

Como a ferramenta Navigator se caracteriza por ser um sistema multi-agentes, uma de suas principais características é a recomendação pró-ativa. Ou seja, os usuários que possuem perfis similares são recomendados. Esta recomendação é dita pró-ativa porque a ferramenta funciona sem precisar da interação direta do usuário na hora de buscar usuários similares.

A interação do usuário é apenas pela interface, onde estão os nomes dos usuários similares e também os graus de similaridades expressos. Através dos ícones apresentados nela, os usuários possuem uma percepção do conhecimento dos usuários em sua lista e em que eles podem ajudar em seus estudos (problemas, tarefas).

A ferramenta Navigator possui uma grande escalabilidade, dando suporte à constante entrada e saída de usuários em sua rede. Por se tratar de uma rede *peer-to-peer*, cada um dos elementos gerencia sua conexão com os demais *peers*, não existindo uma entidade central controladora.

5.2 Pontos negativos

Infelizmente, não foi possível fazer testes mais detalhados do protótipo apresentado. Alguns detalhes, como os pesos dados aos documentos que ficaram mais tempo em execução pelo usuário, carecem de maior estudo para que o perfil gerado torne-se mais real quanto possível.

5.3 Trabalhos Futuros

Atualmente, só há agentes coletores para documentos PDF, DOC e TXT. Uma possível extensão será a criação de novos agentes, para com isso tornar a ferramenta apta a analisar a maior quantidade de tipos de documentos possíveis. Com isso, aumentaremos a confiança nos resultados fornecidos pela ferramenta, visto que teremos mais dados para criar um perfil mais real dos usuários.

Lidar com o conceito de privacidade em uma ferramenta de CSCW é uma das dificuldades mais complexas. Sempre há dados que não podem ser disponibilizados, pessoas com as quais não se quer relacionar etc.

Futuramente, a ferramenta poderá ser adaptada para aparelhos móveis, como celulares, PDA's, PALM's, etc. Um cenário de utilização desta tecnologia seria em uma conferência. Se os participantes possuírem uma ferramenta deste estilo, pode-se, durante a conferência, encontrar possíveis contatos que serão importantes ao longo de uma pesquisa ou estudo.

Outra extensão é adicionar a funcionalidade de clusterização de documentos minerados. Ao fazer a clusterização dos documentos será possível não apenas comparar os termos dos documentos, mas, também, identificar áreas de conhecimento dos usuários. Sendo assim, torna-se possível a indicação de áreas, usuários, documentos de áreas relacionadas, pois ao identificar que os usuários que se interessam pela área 'A' também se interessam pela área 'B', ao encontrar um usuário com interesse na área 'A', podemos indicar a 'B' como um possível interesse futuro ao usuário.

6 Referências

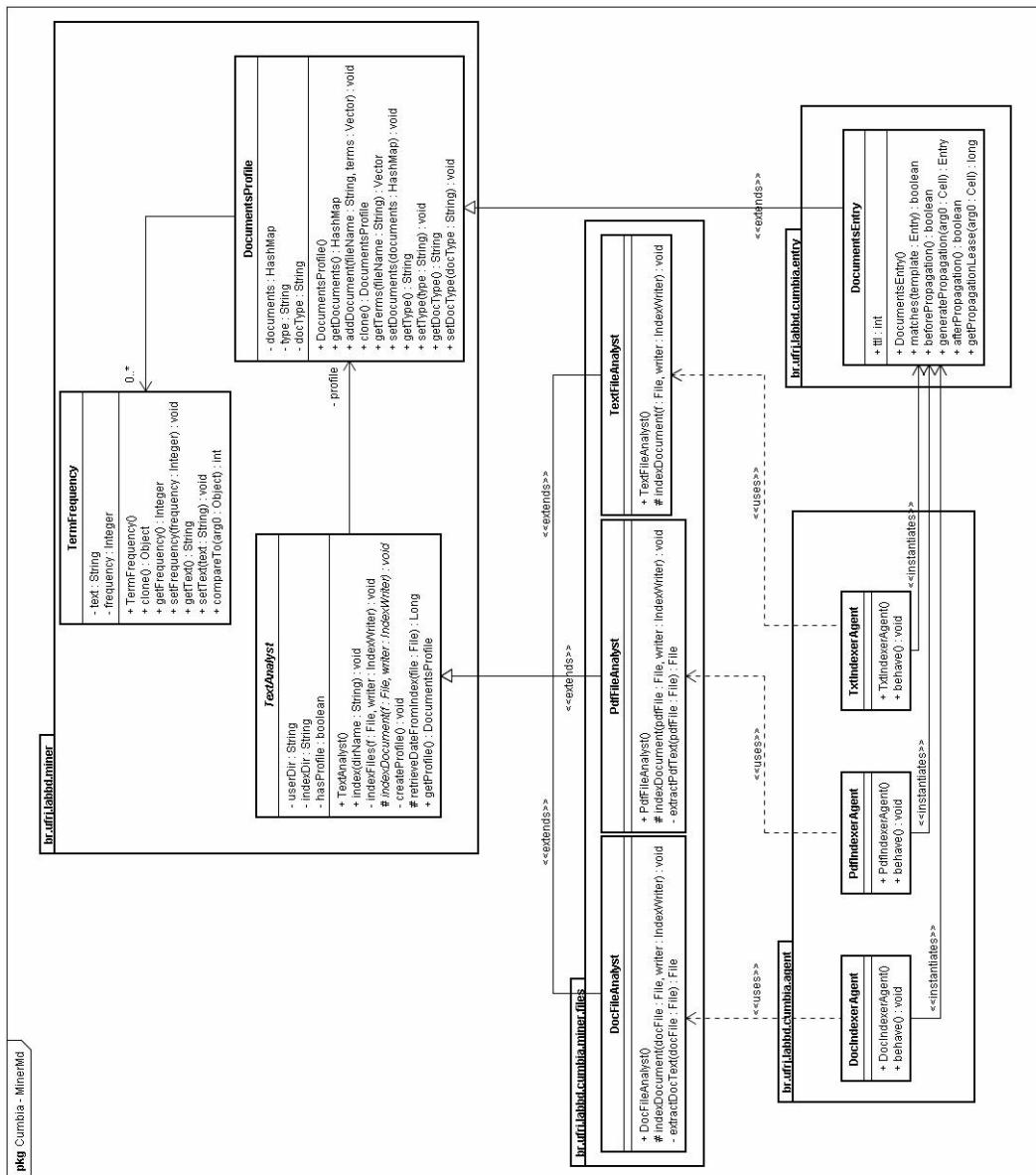
- [1] McEwan, G. and Greenberg, S. (2005) Community Bar Places for Collaboration. In Luigina Ciolfi, Geraldine Fitzpatrick and Liam Bannon (Eds) Workshop Proceedings Settings for Collaboration: The Role of Place, held in conjunction with ECSCW'2005, Paris, Sept 18.
- [2] Foner, L. Yenta: a multi-agent, referral-based matchmaking system, Proceedings of the first international conference on Autonomous agents, p.301-307, , Marina del Rey, California, United States, February 05-08, 1997.
- [3] Resnick, P., Varian, R. H., Recommender Systems, Communications of THE ACM, Vol. 40. No. 3, March 1997.
- [4] P. Maes, Agents That Reduce Work and Information Overload, Comm. ACM, Vol. 37, No. 7, p. 31-40, July 1994.
- [5] Goldberg, D., D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. CACM, 35(12):61--70, Dec 1992.
- [6] Paul Dourish, Victoria Bellotti. Awareness and coordination in shared workspaces. Proceedings of the 1992 ACM conference on Computer-supported cooperative work, p.107-114, Toronto, Ontario, Canada , November 01-04, 1992.
- [7] Paul Dourish , Sara Bly. "Portholes: supporting awareness in a distributed work group". Proceedings of the SIGCHI conference on Human factors in computing systems, p.541-547 Monterey, California, United States May 03-07, 1992.,
- [8] Kraut R.E., Egido C. & Galegher J. Patterns of contact and communication in scientific meamh collaboration. In J. Galegher& R. Kraut (Eds.) Intellectual teamwork, HWe, NJ Erlbauw 149-171, (1990a).
- [9] Kraut, R.E., Fish, R.S., Root, R.W. & Chalfonte, B.L.. Informal communication in organizations: Form, function, and technology, in S. Oskamp & S. Spacapan (E&), People's Reactions to Technology,Newbury Park Sage Publications, 145-199, (1990b).
- [10] JavaSpaces. <http://java.sun.com/developer/products/jini/>
- [11] Shardanand, Upendra, and Maes Pattie, Social Information Filtering: Algorithms for Automating 'Word of Mouth,' Proceedings of the CHI '95 Conference, 1995.

- [12] Lashkari, Yezdi, Metral, Max, and Maes Pattie, Collaborative Interface Agents, Proceedings of the Twelfth National Conference on Artificial Intelligence, MIT Press, Cambridge, MA, 1994.
- [13] Kautz, Henry, Milewski, Al, and Selman Bart, Agent Amplified Communication, AAAI '95 Spring Symposium Workshop Notes on Information Gathering in Distributed, Heterogeneous Environments, Stanford, CA.
- [14] Gutwin, C., Greenberg, S., Blum, R., and Dyck J. Supporting Informal Collaboration in Shared Workspace Groupware. HCI Technical Report 2005-01, The Interactions Lab, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, March.
- [15] Assis, R. L., Facilitando a percepção em ambiente virtuais de aprendizado através da tecnologia groupware, dissertação de mestrado, Departamento de Informática, PUC-Rio, Abril 2000..
- [16] Lee, D.L.; Huei Chuang; Seamons, K., Document ranking and the vector-space model, Software, IEEE , vol.14, no.2 pp.67-75, Mar/Apr 1997.
- [17] Aridor, Y.; Lange, D. B. Agent Design Patterns: Elements of Agent Application Design . Proceedings of the Second International Conference on Autonomous Agents. St. Paul, USA. Maio, 1998.
- [18] Gilbert, D.; Aparicio, M.; Atkinson, B.; Brady, S.; Ciccarino, J.; Grosof, B.; O'Connor, P.; Osisek, D.; Pritko, S.; Spagna, R.; Wilson, L. ; IBM Intelligent Agent Strategy, White Paper, 1995.
- [19] Milojicic, D.S.; Kalogeraki, V; Lukose,R.; Nagaraja, K.; Pruyne, J.; Richard, B.; Rollins, S.; Xu, Z. Peer-to-Peer Computing. in HPL-2002-57 - HP Laboratories Palo Alto, 2002.
- [20] Miranda, M.; Xexéo, G. B.; Souza, J. M. Building Tools for Emergent Design with COPPEER. In: 10th International Conference o Computer Supported Cooperative Work in Design (CSCWD) Nanjing, 2006.
- [21] Miranda, M.; Xexeo, G. B. A complex adaptive system approach for agent-based peer-to-peer collaborative applications. In: Workshop de Teses e Dissertações de Bando de Dados (WTDBD'05) Sociedade Brasileira de Banco de Dados. Uberlandia, 2005.

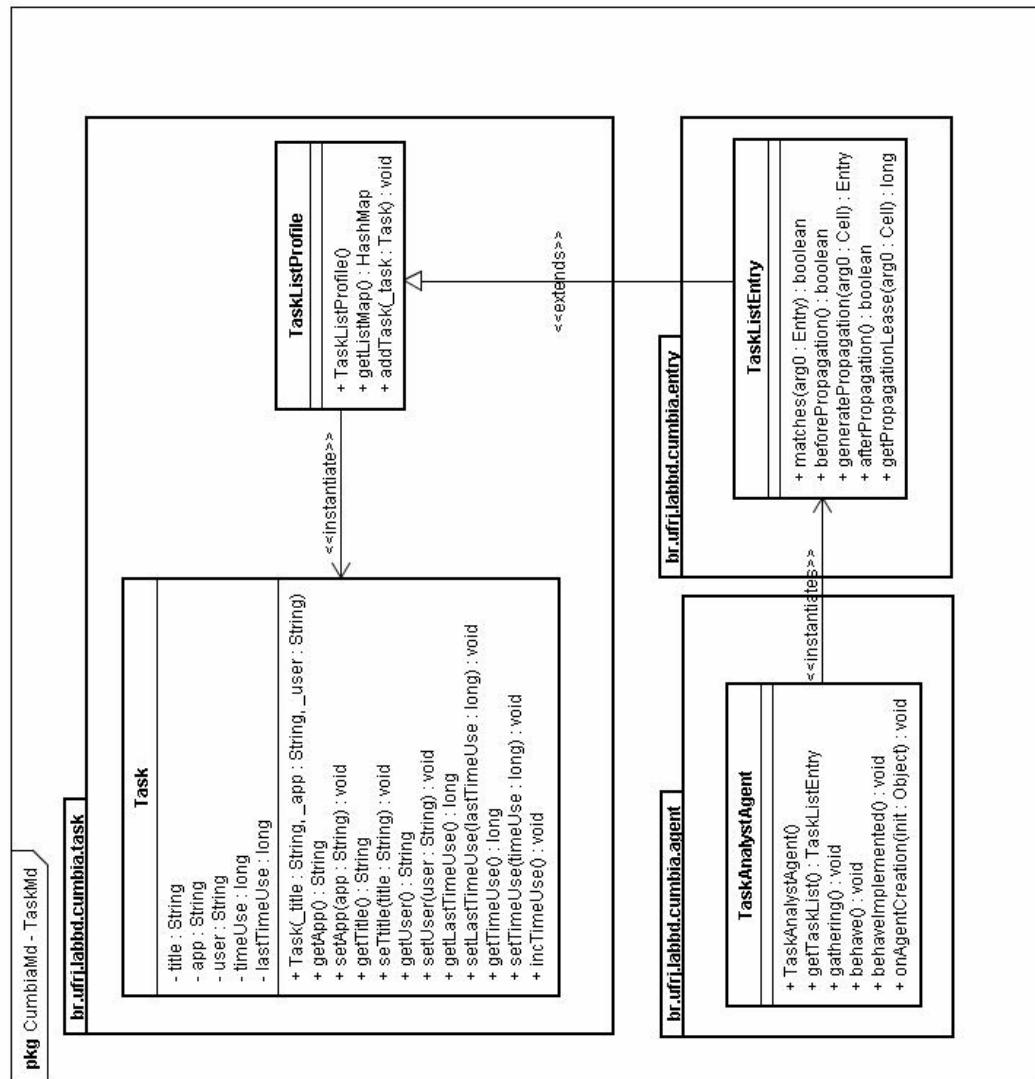
- [22] Roussopoulos, M.; Baker, M.; Rosenthal,D.S.H.; Giuli,T.J.; Maniatis,P.; Mogul, J.C. 2 P2P or Not 2 P2P? In: Third International Workshop, IPTPS 2004, La Jolla, CA, USA, February, 2004.
- [23] Silva, M. M. Arquitetura para Reuso de Ações em Sistemas Multi-Agentes, Programa de Mestrado em Sistema e Computação do IME/RJ, 2005.
- [24] PDFBOX - <http://www.pdfbox.org/>
- [25] JACOB - <http://danadler.com/jacob/>
- [26] Pinheiro, M.K. ; Lima, J.V. ; Borges, M.R.S. Awareness em Sistemas de Groupware. In: Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software, 2001, San Jose. IDEAS 2001, 2001. v. 1. p. 323-335.
- [27] Isaacs, E. A., Tang, J. C. e Morris, T. Piazza: a desktop environment supporting impromptu and planned interactions, 1996

Apêndice A: Diagramas de Classe

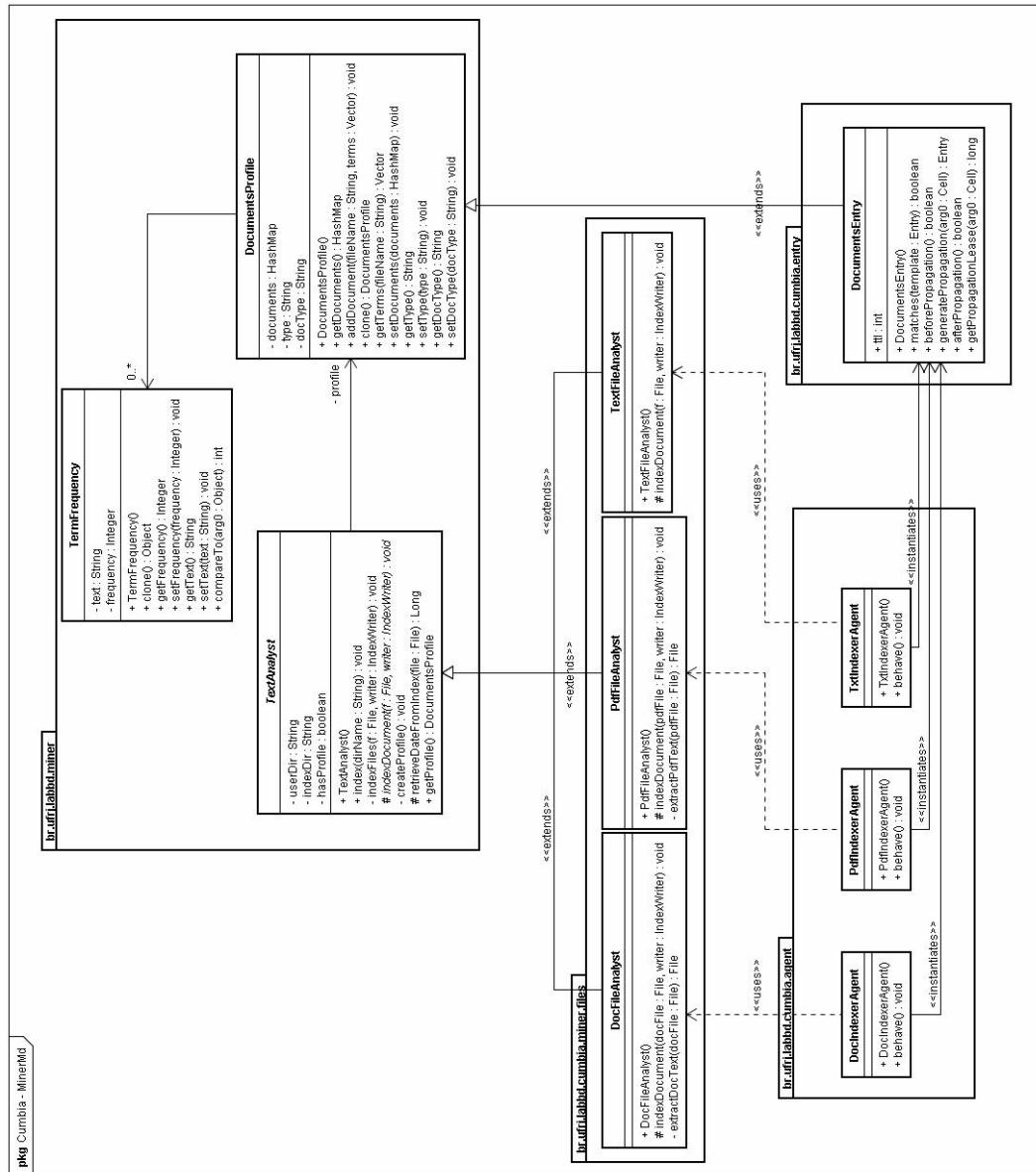
A.1 Minerador



A.2 Coletor de Tarefas



A.3 Match



A.4 Agentes

