

3 Implementação

O presente capítulo discute os principais detalhes da implementação do servidor e cliente MMS. Em algumas subseções são apresentados detalhes de implementação, problemas, resoluções de problemas, questões para discussões posteriores e trabalhos em aberto. Desta forma, são apresentadas considerações sobre: a camada de dados do MMS, o serviço de localização utilizado, os modos de funcionamento do MMS, discussões a respeito das formas de notificação de usuários, o algoritmo de *matching*, aspectos do processamento pós *matching*, arquitetura do serviço e o cliente do MMS.

3.1. A Camada de Dados do MMS

Como visto no capítulo anterior, o MMS trabalha com perfis de usuários descritos por meio de ontologias, que, por sua vez, foram descritas por meio da linguagem OWL. Para realizar o armazenamento dos perfis e a manipulação de dados descritos em OWL, o MMS utilizou um *framework* para desenvolvimento de aplicações de Web Semântica (Berners-Lee, et al. 2001) desenvolvido pela Hewlett-Packard (HP) chamado *Jena* (Carroll, et al. 2004).

O *Jena* possui APIs Java para manipulação e consultas em ontologias descritas em RDF, DAML+OIL e OWL. Além disso, o *Jena* disponibiliza consultas por meio de uma linguagem denominada RDQL (RDQL, 2004) e mecanismos de armazenamento de dados semânticos em banco de dados relacionais, tais como: MySQL, PostgreSQL e Oracle. Outra característica interessante do *Jena* é a presença de mecanismos de inferência baseados em regras que podem ser utilizadas para derivar novas conclusões sobre os perfis dos usuários. Uma regra é uma expressão onde existe um conjunto de premissas e um conjunto de conclusões; caso as premissas sejam válidas, elas derivam as conclusões. No *Jena*, cada premissa ou conclusão é formada por uma tripla constituída de um sujeito, um predicado e um objeto (similar ao RDF).

Como o *Jena* possui APIs para manipulação de ontologias e possibilita a construção de mecanismos de inferências, ele foi utilizado na construção da camada de dados do MMS. Esta camada é responsável por:

- Manipular os dados sobre perfis, disponibilizando operações básicas como: incluir, excluir e atualizar perfis na base de conhecimento do MMS;
- Prover o armazenamento da base de conhecimento do MMS;
- Prover raciocínio sobre dados de perfis por meio de mecanismos de inferência (*reasoner*).

Quando o servidor MMS é iniciado pela primeira vez, ou seja, seu banco de dados se encontra vazio, este realiza um *download* das definições e instâncias pré-definidas da ontologia que estão disponibilizadas, por exemplo, em uma URL. Estas definições são, então, armazenadas em um banco de dados relacional e, nas inicializações subseqüentes, esta operação de *download* não é realizada. No momento de cada inicialização do serviço, a máquina de inferência do MMS tenta inferir novos fatos sobre os perfis por meio de suas regras de inferências (escritas pelo administrador). Esta máquina de inferência volta a ser acionada quando um novo perfil é adicionado, atualizado ou excluído no MMS.

Cada vez que o MMS é requisitado para realizar uma operação de *matching*, ele transforma os dados armazenados como triplas RDF para objetos Java, que serão posteriormente utilizados pelo algoritmo de *matching*. Durante a implementação do serviço percebe-se que esta operação de recuperação de dados aumenta significativamente a latência do *matching*, sendo necessária a criação de um sistema simples de *cache*, implementado por meio de uma tabela *hash* que armazena objetos em memória. Assim, a cada vez que um novo objeto da classe *Pessoa* é lido, ele é colocado na *cache* do serviço. Se a *cache* estiver cheia e for necessário armazenar mais um objeto em memória, alguns objetos são removidos da *cache* de forma aleatória. O número máximo de objetos é definido pelo administrador do serviço.

O sistema de *cache* do MMS não adota nenhuma política para remover objetos que não estão sendo utilizados. Portanto, eventualmente, o MMS pode remover objetos que logo deverão ser recolocados em memória. Nas implementações futuras, poderão ser utilizadas políticas sofisticadas de gerenciamento de *cache* como, por exemplo, o *JBoss Cache*

(JBossCache,2005). Contudo, mesmo utilizando a tabela *hash* e a remoção aleatória de objetos da *cache*, foi visível a melhora de performance do serviço.

No servidor MMS, inferências podem ser utilizadas em duas situações: para ajudar o administrador a indicar relacionamentos entre as regiões lógicas e instâncias de interesse, e para derivar novos fatos sobre os interesses dos usuários com o objetivo de obter melhores resultados de *matching*.

No MMS é necessário criar instâncias de regiões lógicas sobre as quais o serviço esteja atuando. Para o MMS não basta que o serviço de localização informe que o usuário está em uma região *A*, o *namespace* desta localização também deve existir, por exemplo, <http://www.inf.puc-rio.br/ontologia.owl#A>.

O administrador também é responsável pela criação de instâncias de interesse. Cabe a ele definir quais são os interesses que estão disponíveis para que a escolha seja feita pelos usuários.

O administrador pode criar hierarquias para as instâncias de localizações e para instâncias de interesses. Na interface *Web* para administração, ele pode informar, por exemplo, que uma área *A* é subárea de *B* e que *B* é subárea de *C*. Por transitividade, o serviço cria automaticamente o relacionamento onde *A* é uma subárea de *C*. Não existe a necessidade do usuário informar todos os relacionamentos possíveis entre as regiões lógicas, o que na prática implica uma simplificação das interações do administrador com o serviço. Entretanto, para que isto ocorra, uma regra de inferência deve ser escrita no conjunto de regras do MMS. Cabe, então, ao administrador querer ou não utilizar este recurso. O Quadro 1 ilustra uma regra de transitividade, composta por duas premissas e uma conclusão. Os seus símbolos significam:

- `<>#` - o *namespace* da ontologia.
- `?v` - variável.
- `->` - dedução lógica.
- `,` - conjunção lógica

```
[rule: (?a <>#isPartOf ?b), (?b <>#isPartOf ?c) -> (?a <>#isPartOf ?c)]
```

Quadro 1 - Exemplo de regra de transitividade escrita para o MMS

As regras de inferência podem ser também utilizadas para descobrir novos fatos sobre os interesses dos usuários, ou seja, enriquecer a base de

conhecimento. Por exemplo, dado que *lógica* é uma sub-área da *matemática*, se um usuário tem interesse por *lógica* é possível inferir que ele também possui interesse por *matemática*. No entanto, o contrário não é possível ser inferido. Regras de inferência como esta podem ser escritas no MMS.

A título de exemplo, o Quadro 2 apresenta uma regra que serve para inferir novos fatos sobre os perfis dos usuários com um valor de interesse padrão igual a cinco. O valor cinco foi estabelecido para esta regra, pois representa um valor médio segundo a escala padrão definida na ontologia do MMS.

```
[rule2: (?a <>#hasInterestLevel ?b), (?b <>#hasInterest ?c), (?c
<>#isSubInterestOf ?d), (makeTemp(?T)) -> (?a
<>#hasInterestLevel ?T), (?T rdf:type <>#InterestLevel), (?T
<>#hasInterest ?d), (?T <>#hasLocation <>#allLocations) (?T
<>#value 5)]
```

Quadro 2 - Exemplo de regra de inferência

Basicamente, a expressão `(?c <># isSubInterestOf ?d)` que relaciona duas instâncias de interesse, verifica o caso de existirem interesses que sejam relacionados, como por exemplo, *lógica* e *matemática*. Em caso positivo, uma instância de *InterestLevel* é então criada por meio da expressão `(makeTemp(?T))`. A instância criada é associada ao usuário indicado pela variável `?a`, com o interesse apontado por `d?`, com o nível de interesse igual a cinco e relevância em todas localizações (*allLocations*).

Acredita-se que regras semelhantes às do Quadro 2 poderiam ser utilizadas para se construir perguntas aos usuários que, posteriormente, informariam se possuem interesse no tópico inferido e, em caso afirmativo, qual é o grau deste interesse e em quais localizações o interesse é relevante. Este esquema de perguntas não foi implementado na versão atual do MMS, todavia, pode ser concebido nas novas versões do serviço.

Um thesaurus como o Wordnet (WORDNET, 2005) também poderia ser útil para se descobrir (inferir) interesses de usuários. O WordNet é um léxico da língua inglesa onde substantivo, verbos, adjetivos e advérbios são organizados dentro de conjuntos de sinônimos. Usando um thesaurus seria possível identificar as relações de parentesco entre os termos o que permitiria descobrir novos interesses. Desta forma, os perfis dos usuários poderiam conter mais

informações de interesses o que possivelmente aumentaria as chances de encontros entre usuários. Além das inferências, acredita-se que um thesaurus seria importante para que o MMS se torne um sistema onde não fosse necessária uma configuração prévia (feita por um administrador) de instâncias de interesses.

A próxima seção apresentará o serviço de localização utilizado pelo MMS. Uma compreensão do modo de funcionamento deste serviço é necessária para o entendimento posterior do funcionamento do MMS.

3.2. O Serviço de Localização Utilizado pelo MMS

Para obter as informações sobre localizações de usuários, o MMS utiliza serviços providos por uma infra-estrutura de *middleware* chamada MoCA (*Mobile Collaboration Architecture*) (Sacramento, et al 2004). Esta infra-estrutura consiste de um conjunto de serviços básicos que apóiam a implementação de aplicações de colaboração móvel cientes de contexto (*Context-Aware*), para redes locais sem fio, infra-estruturadas, e que seguem o padrão IEEE 802.11. Na MoCA, a informação de contexto é obtida e processada através dos seguintes componentes: Monitor, Serviço de Informação de Contexto e Serviço de Inferência de Localização.

O monitor, um *daemon* que executa em cada dispositivo móvel e que é responsável por (i) coletar a informação de estado de execução e conectividade do dispositivo como, por exemplo, o *Access Point* 802.11 corrente, a qualidade da conectividade sem fio, o nível de energia etc, e (ii) enviar estes dados para um Serviço de Informação de Contexto (*Context Information Service* - CIS) que executa em um nó da rede fixa.

Além de servir como um repositório de dados de contexto recebidos de cada dispositivo móvel, o CIS também recebe pedidos de notificações sobre eventos de mudança de contexto de outros serviços MoCA ou de aplicações.

Um dos serviços interessados em receber notificações do CIS é o Serviço de Inferência de Localização (*Location Inference Service* - LIS) (Rubinsztein, et al 2004), que é responsável por inferir a localização aproximada do dispositivo em uma região de cobertura de uma *wireless* LAN. Para inferir a localização geográfica, este serviço compara o sinal de rádio frequência (RF) de todos os pontos de acesso do padrão 802.11 percebidos por um dispositivo móvel, com padrão de sinal de RF previamente medido em pontos de referências naquela

região. Para a inferência de posição é utilizado um algoritmo de vizinhos mais próximos, similar ao descrito em (Bahl, 2000).

No LIS também é possível definir regiões retangulares e de tamanho arbitrário, atribuir um nome para estas localizações simbólicas e construir uma topologia hierárquica a partir destas regiões, como, por exemplo: “*Alto-PUC* é composto de *Prédio_FPLF*, *Espaço Aberto* e *Restaurante*.” No MMS, o administrador do serviço pode (e deve) descrever a topologia das regiões simbólicas. Esta topologia é armazenada juntamente com as outras instâncias e definições da ontologia.

O LIS disponibiliza uma API que permite a uma aplicação (ou outro serviço, como o de *matchmaking*) consultar e receber notificações sobre quais dispositivos entram ou saem em uma determinada região, ou para qual região um determinado dispositivo se move. Nas consultas e nas notificações, o dispositivo móvel é identificado pelo seu endereço MAC (*MACAddress*).

A próxima seção apresentará em detalhes os modos de funcionamento do MMS enfatizando as interações que o serviço realiza com o LIS durante um processo de *matching*.

3.3. Os Modos de Funcionamento do MMS

O MMS possui dois modos de funcionamento, o síncrono e o assíncrono. A seguir, serão apresentados em detalhes os dois modos de funcionamento do serviço.

3.3.1. Funcionamento Síncrono

A implementação síncrona é utilizada em situações nas quais um usuário realiza consultas no MMS, para verificar se existem outros usuários com perfis de interesse similar ao dele e em uma mesma região simbólica. Além disso, operações de gerenciamento de perfis como: incluir, excluir, atualizar perfil e, operações que informam a região corrente do usuário para a aplicação cliente são realizadas neste modo.

Para que todas as operações de *matching* possam ser realizadas, inicialmente devem existir perfis cadastrados na base de conhecimento do

servidor MMS. No momento da criação de um perfil, a aplicação cliente⁶ captura o endereço MAC do dispositivo móvel do usuário e acrescenta esta informação ao perfil. Esta operação é necessária na atual implementação do MMS, pois o LIS provê apenas informações de localização referentes a dispositivos identificados pelos seus respectivos endereços MAC. Este requisito obriga o MMS a disponibilizar um método, para recuperar informações sobre o perfil de um usuário, a partir de um endereço MAC e, um método para descobrir o endereço MAC para um dado perfil. Se, por um acaso, o usuário do MMS trocar de dispositivo móvel, ele deverá também atualizar o seu perfil, para indicar o endereço MAC do seu novo dispositivo. Esta atualização pode ser feita de forma automática ou manual, dependerá de como a aplicação cliente estiver implementada. A versão atual da aplicação cliente do MMS implementa atualização manual. A Figura 6 ilustra o processamento das consultas no modo síncrono.

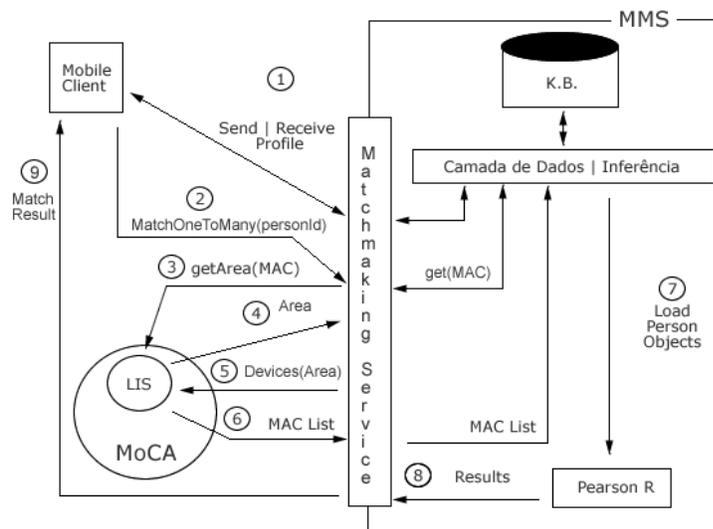


Figura 6 – Integração entre o MMS e LIS no modo síncrono

Uma vez criados os perfis no serviço (1), os usuários poderão consultar o MMS para descobrir se existem pessoas co-localizadas e com perfis similares ao dele. Para realizar esta operação, um usuário deve efetuar a operação de consulta síncrona em seu programa cliente. Uma vez acionada esta operação, o identificador do usuário é enviado ao servidor MMS (2). Este identificador é formado pelo espaço de nome (*namespace*) da ontologia, seguido do endereço

⁶ A componente do MMS que executa no dispositivo móvel.

de e-mail do usuário. Por exemplo, <http://www.inf.puc-rio.br/ontologia.owl#meumail@exemplo.com>.

Quando o servidor MMS recebe a requisição, ele inicialmente recupera o endereço MAC do dispositivo do usuário que está associado ao identificador de perfil recebido. De posse do endereço MAC, o servidor MMS solicita ao LIS que informe a região simbólica em que o dispositivo com este endereço MAC se encontra (3, 4). Identificada a região simbólica, o servidor MMS pede ao LIS para informar os demais dispositivos que se encontram na mesma região (5). O LIS retornará uma lista de endereços MAC que representam todos os dispositivos na área (6)⁷. Para cada endereço MAC contido na lista, o servidor MMS então recupera as informações de perfil dos usuários correspondentes (7). Em seguida, ele compara, por meio do algoritmo de *matching*, o perfil do usuário que solicitou o serviço com perfis de todos os usuários presentes na mesma região simbólica (8). Depois de realizadas todas as comparações entre perfis, o MMS realiza processamentos pós-*matching*. Estes processamentos consistem de uma série de filtros que têm o objetivo de organizar a informação, antes que ela seja enviada para aplicação cliente, que apenas tem a responsabilidade de os apresentar aos usuários (9). Os detalhes do algoritmo de *matching* e dos filtros do processamento pós-*matching* serão abordados nas próximas subseções deste capítulo.

Como apresentado, o modo síncrono permite que usuários realizem consultas para saber se existem outros usuários co-localizados e com perfis similares. Porém, existem aplicações que demandam uma intenção assíncrona com o MMS, ou seja, onde os usuários podem ser notificados, caso eles entrem em alguma região simbólica que contenha outros usuários com perfis similares.

3.3.2. Funcionamento Assíncrono

No modo assíncrono, usuários do MMS podem se inscrever no serviço a fim de receber notificações assíncronas sobre a presença de usuários em uma determinada área lógica (2). A Figura 7 ilustra o processamento de uma operação assíncrona no MMS.

⁷ O MAC do dispositivo do usuário que requisitou o serviço também estará nesta lista. Cabe ao MMS desconsiderar esta informação.

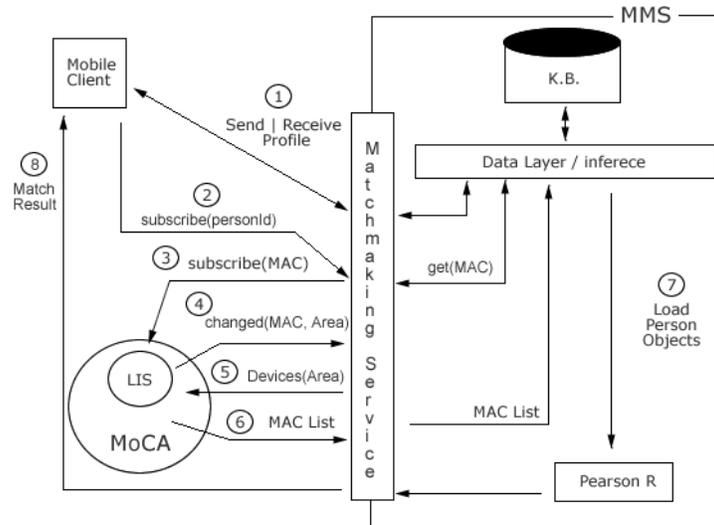


Figura 7 – Integração do MMS e LIS no modo assíncrono

Neste modo, o usuário acessa no cliente a operação de inscrição no servidor MMS que, por sua vez, recebe uma requisição contendo o identificador deste usuário. O servidor MMS, então, recupera o endereço MAC do dispositivo do usuário, correspondente ao identificador recebido, e utiliza esse endereço para realizar um registro no LIS (3). Após esse processo, o LIS passa a monitorar o endereço MAC que lhe foi dado. O monitoramento que o LIS realiza consiste em verificar a movimentação dos usuários dentro de regiões simbólicas. Quando o usuário se movimenta e modifica a sua região simbólica, o LIS notifica o servidor MMS deste evento, enviando o endereço MAC do dispositivo do usuário que se movimentou e o nome da nova área simbólica (4). Através do endereço MAC do dispositivo do usuário, o servidor MMS recupera as informações do perfil do usuário. Com o nome da região simbólica, o servidor MMS consulta o LIS para descobrir todos os endereços de dispositivos que estão nesta região (5), recebendo uma lista de endereços MAC que representam dispositivos presentes na região simbólica (6). Desta forma, o MMS recupera as informações de perfil de cada endereço MAC contido da lista (7). Usando o seu algoritmo de *matching*, o servidor MMS analisa os perfis dos usuários co-localizados com os perfis do usuário que solicitou o serviço. Se for detectada alguma similaridade de perfil com, pelo menos, um usuário da região simbólica, o MMS realiza o processamento pós-*matching*, para organizar a apresentação das informações. Em seguida, o servidor MMS envia os resultados para o programa cliente que exibirá as informações (8). Na hipótese do algoritmo de

matching não encontrar similaridade de perfil com nenhum usuário da região simbólica, o servidor MMS não realiza nenhuma operação, ou seja, o usuário não recebe nenhuma informação e continua sendo monitorado. Um usuário só deixa de ser monitorado, quando este executar uma operação para cancelamento de sua inscrição no modo assíncrono do MMS.

Como visto, quando um usuário se movimenta o LIS notifica o MMS sobre a ocorrência deste evento. Internamente, este evento é tratado no MMS como uma requisição de mudança de região simbólica de um usuário. Entretanto, existe um segundo tipo de requisição que o MMS processa; requisições geradas por usuários que permaneçam parados em uma região simbólica. Por exemplo, imagine um usuário *A* assistindo uma conferência; em um determinado momento, entra na sala um usuário *B* que tenha interesses similares a *A*. Devido à entrada de *B* na sala de conferência, é gerada uma requisição de movimentação, já o usuário *A* acaba também gerando uma requisição devido ao seu longo tempo de permanência na sala de conferência. Desta forma, provavelmente em momentos distintos, *A* é notificado da presença de *B* e vice-versa. Resumidamente, a detecção de usuários no modo assíncrono trabalha da seguinte forma⁸:

- Quando um usuário se movimenta, gera uma requisição de movimentação. Esta requisição (identificada pelo endereço MAC do dispositivo) é colocada na fila de requisições de movimentação.
- No momento do processamento de uma requisição, o MMS salva o horário e o nome da região simbólica na qual o usuário entrou, gerando assim um *timestamp*.
- Continuamente, o MMS verifica todos os *timestamps* para identificar usuários que estejam parados em uma região simbólica por um determinado (configurável) intervalo de tempo.
- Uma vez identificado um usuário parado, o MMS gera uma requisição. Esta requisição é colocada em uma fila de requisições de usuários parados.
- As requisições de movimentação têm prioridade de processamento sobre as requisições provenientes de usuários parados. Assim, se um usuário que contenha uma requisição na fila de requisições de usuários parados

⁸ Alguns detalhes de implementação foram suprimidos para não prejudicar a compreensão geral do algoritmo.

se movimentar, apenas a requisição de movimentação é mantida. Da mesma forma, caso um usuário possua uma requisição na fila de movimentação, não é gerada uma requisição para a fila de usuários parados.

Como visto, as requisições de movimentação têm prioridade de processamento, desta forma, é necessário que exista uma baixa quantidade de requisições na fila de movimentação para que as requisições de usuários parados sejam processadas. De certo modo, esta política adotada sugere que a primeira versão do MMS atenda a cenários com baixa movimentação. Possivelmente, para que o MMS seja utilizado em cenários de maior movimentação, esta política de prioridade no processamento das requisições deva ser alterada.

3.4. Notificações

Como pode ser observado o MMS não garante simetria das notificações tanto no modo assíncrono quanto no modo síncrono, ou seja, se *A* encontra *B*, não necessariamente *B* encontra *A*.

Acredita-se que notificações simétricas poderiam gerar uma sobrecarga de notificações⁹ aos usuários. Por exemplo, imagine quatro usuários {*A*, *B*, *C* e *D*} com similaridades entre seus perfis e que estes entrem um a um em uma região simbólica qualquer. O usuário *A*, primeiro a entrar na região, não recebe notificação. Quando *B* entrar, *A* e *B* são notificados. No momento da entrada de *C*; *A*, *B* e *C* são notificados. Quando *D* entrar, *A*, *B*, *C* e *D* recebem notificações. Desta forma, fica claro que, dependendo da situação, as notificações simétricas podem causar excessos de mensagens recebidas pelos usuários.

Entretanto, notificações assimétricas também possuem desvantagens, pois neste caso, uma interação depende da tomada de iniciativa de apenas um usuário. Esta situação não ocorre com notificações simétricas onde sempre dois usuários são informados simultaneamente (ou com intervalo de tempo desprezível), o que permite que qualquer uma das partes tome a iniciativa de interagir.

Como citado anteriormente, a primeira versão do MMS está implementada com notificações assimétricas. No entanto, devido à monitoração periódica da

região simbólica em que um usuário se encontra, o efeito causado por esta assimetria é diminuído. Por exemplo, imagine dois usuários {A e B} que tenham similaridades entre seus perfis; primeiramente o usuário A entra em uma região vazia, não recebendo nenhuma notificação. Em seguida, B entra na região, recebendo uma notificação que A está presente. Depois de certo tempo dos usuários parados na região simbólica, A recebe a informação que B está presente. Assim, apesar da essência das notificações do serviço se manter assimétrica, o MMS tenta chegar em um “meio termo”, notificando, periodicamente, usuários que se mantenham parados em uma região. Desta forma, o objetivo destas notificações periódicas é evitar que usuários recebam notificações constantes e garantir que, em determinado intervalo de tempo, os usuários parados em uma região se encontrem mutuamente.

3.5. Processamento de Pós-Matching

Como foi visto anteriormente, depois de realizado o cálculo de correlação, através do algoritmo de *matching* entre perfis, o servidor MMS executa um pós-*matching* que consiste no acréscimo de novas informações ao resultado final (como, por exemplo, o e-mail do usuário), na remoção de resultados de correlação negativos e na estruturação dos resultados para sua apresentação pelo programa cliente. O formato de um resultado *matching* solicitado pelo usuário *u* entre o seu perfil e o de um outro usuário *j* consiste da seguinte tupla:

Matching (*u*, *j*) = <E, M, F, D, I, L>, onde:

E = Um *e-mail* de contato do usuário *j*;

M = Valor retornado da função de *matching* entre (*u*, *j*) apresentado como um valor percentual;

F = URL apontando para a página HTML do usuário *j*;

D = Idioma nativo do usuário *j*;

I = Nome do interesse de maior similaridade entre *u* e *j*;

L = Região simbólica que o usuário *j* se encontra.

Desta forma, para apresentar os resultados para a aplicação cliente de forma amigável, o MMS realiza uma série de processamentos de pós-*matching*.

⁹ Principalmente se uma notificação vier acompanhada de um ruído sonoro.

Este processamento é determinado por uma seqüência de filtros, sendo alguns permanentemente ativos, enquanto outros podem ser selecionados pelo usuário antes de cada requisição ao MMS. Exemplos de filtros permanentes são:

- Filtro para remover resultados negativos de *matching*, ou seja, pessoas sem similaridades de perfis;
- Filtro para ordenar os resultados de *matching*, do mais similar ao menos similar;
- Filtro para transformar o valor do resultado de *matching* (saída do algoritmo) para um valor percentual;

Exemplo de filtros que podem ser ativados pelos usuários:

- Filtro para mostrar os resultados acima de um determinado percentual, por exemplo, mostrar somente usuários com similaridade de interesse acima de 50%.

Em resumo, o processamento pós-*matching* é realizado por uma seqüência de filtros onde alguns deles são permanentes e outros são ativados pelos usuários. Portanto, estes filtros realizam tarefas importantes dentro do serviço, pois disponibilizam um resultado de *matching* de uma forma mais amigável ao usuário. Sem eles, o MMS forneceria apenas o resultado direto do algoritmo de *matching*.

3.6. O Matching no MMS

Para realizar a comparação entre perfis, o MMS utiliza um algoritmo baseado em correlação linear chamado *Pearson R* (Shardnand, 1994). Considera-se a correlação como uma medida da magnitude e direção das relações entre duas ou mais variáveis. O coeficiente de correlação *Pearson R* varia de -1.00 até +1.00, onde o valor -1.00 representa a maior correlação negativa, o valor +1.00 representa a maior correlação positiva. O coeficiente de correlação *Pearson R* é calculado segundo a expressão apresentada na Figura 8, onde U_i e J_i representam valores associados a interesses para o conjunto de interesses $i \in C$ e, \bar{U} e \bar{J} representa a média dos interesses de U e J :

$$r_{UJ} = \frac{\sum_i (U_i - \bar{U})(J_i - \bar{J})}{\sqrt{\sum_i (U_i - \bar{U})^2 \sum_i (J_i - \bar{J})^2}}$$

Figura 8 – A expressão Pearson R

Por exemplo, sejam três usuários {1, 2 e 3} com níveis de interesse em dez assuntos (A-J), conforme mostrado na Tabela 2. Os valores dos níveis foram atribuídos de acordo com uma escala onde 1 representa pouco interesse, 5 representa médio interesse e 10 representa grande interesse.

	A	B	C	D	E	F	G	H	I	J	Médias
1	10	9	9	8	3	6	5	2	2	2	5.6
2	9	9	9	9	6	7	4	3	4	1	6.1
3	1	3	4	2	4	3	1	9	10	9	4.6

Tabela 2 - Interesses de usuários

Usando a expressão da Figura 8 o coeficiente de correlação *Pearson R* (1,2) entre os interesses dos usuários 1 e 2 Obteve-se uma correlação positiva de aproximadamente 91%, ou seja, os seus perfis possuem um alto grau de similaridade. Já a correlação *Pearson R* (1,3) entre os perfis dos usuários 1 e 3 indica um baixo grau de similaridade entre seus interesses, apontando um valor de aproximadamente -77%. Os resultados obtidos estão de acordo com a nossa interpretação de que o perfil de 1 e 2 são similares, enquanto os perfis de 1 e 3 não são similares.

Como visto anteriormente, alguns interesses são relevantes somente em algumas regiões. Assim, a implementação do *Pearson R* acompanha este raciocínio da seguinte forma: Suponha que um usuário 1 tenha marcado seus interesses como sendo válidos em todas as localizações (*allLocations*). Suponha também, que um usuário 2 tenha indicado que os interesses (B, C, D e G) como relevantes em uma localização X e, os demais, válidos para todas as localizações. A Tabela 3 ilustra esta disposição de valores, em cinza, os interesses somente válidos na localização X.

	A	B	C	D	E	F	G	H	I	J
1	10	9	9	8	3	6	5	2	2	2
2	9	9	9	9	6	7	4	3	4	1

Tabela 3 – Distribuição de valores de interesses levando-se em conta a localização

O cálculo da correlação entre os usuários 1 e 2 na localização X leva em conta todos os valores apresentados na tabela acima; o resultado do $Pearson(1,2)$ na região X é de +- 91%. Entretanto, se os usuários 1 e 2 se encontraram na região Y , os interesses (B,C,D,G) são desconsiderados do cálculo, desta forma, os resultado do $Pearson(1,2)$ em Y é de +- 87%. A Tabela 4 ilustra este processo.

	A	E	F	H	I	J
1	10	3	6	2	2	2
2	9	6	7	3	4	1

Tabela 4 – Valores considerados no cálculo do $Pearson R$ entre 1 e 2 na região Y

Além do $Pearson R$ existem também outros algoritmos para avaliar a correlação entre perfis de usuários, tais como: *Constrained Pearson R* (Shardnand, 1994), *Mean Squared Difference* (MSD) (Feynman, 1993) e o *Artist-Artist* (Sims, 1993). A escolha do $Pearson R$ se deu por sua simplicidade de implementação e por prover um resultado de *matching* satisfatório como discutido em (Shardnand, 1994).

Cabe salientar que no cálculo do $Pearson R$ no MMS são utilizados apenas os conceitos relacionados a interesses de usuários, ou seja, para efeito de cálculo de similaridade é considerada apenas uma dimensão sobre os perfis dos usuários.

Devido ao fato que uma consulta no MMS procura por pessoas com perfis similares em apenas uma região simbólica, atualmente não é necessário que os usuários tenham conhecimento sobre a topologia das regiões. Se as consultas do MMS levassem em conta mais de uma região simbólica (navegação), os usuários poderiam quer consciência da topologia das regiões com o objetivo de ampliar o escopo de procura. Por exemplo, um usuário dentro de uma sala de conferência poderia também realizar buscas nas salas próximas. Desta forma, para que isto fosse possível, o algoritmo de *matching* deveria ser modificado

para suportar navegações por entre as regiões simbólicas e o resultado de *matching* deveria também ser modificado.

Entretanto, atualmente o algoritmo de *matching* não navega pela topologia das regiões simbólicas, ou seja, é verificada apenas a similaridade entre os perfis de usuários co-localizados em uma região simbólica atômica. Uma implementação de um algoritmo que descubra usuários com perfis similares, mas não co-localizados, não foi realizada devido ao seguinte problema: como conceitualmente o MMS permite que, dependendo da localização, dois usuários possam ter graus de similaridades diferentes (devido aos interesses dependentes de localização), haveria a necessidade de se definir uma forma de calcular a proximidade entre perfis de usuários não co-localizados.

Uma saída para este problema seria considerar dois usuários como co-localizados na região do usuário que requisitou o serviço e calcular o *matching* normalmente. Desta forma, acredita-se não haver necessidade de se utilizar (não seria) o componente de distância diretamente no cálculo do *matching*. Considera-se ser suficiente informar a localização de cada usuário no resultado do *matching* para que os participantes tomem suas decisões sobre quem contatar. No entanto, devido à limitação de tempo para a implementação do MMS, optou-se por não tratar o *matching* de usuários localizados em regiões simbólicas distintas.

Neste ponto, vale a pena discutir também os problemas decorrentes de um mapeamento incorreto ou incompleto de áreas físicas do mundo real para as regiões simbólicas do LIS, pois neste caso os usuários não seriam capazes de relacionar interesses específicos a cada localização do mundo real. Por exemplo, a Figura 9 mostra uma área de cobertura de uma rede 802.11 dentro de uma universidade. A área coberta pelo sinal abrange parte do *Prédio A*, parte do *Prédio B* e o espaço entre os dois prédios. Neste caso, o administrador do sistema não deve indicar que a área coberta esteja relacionada nem com o *Prédio A* e nem com o *Prédio B*.

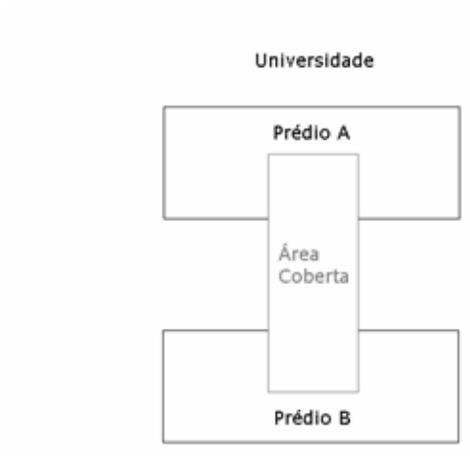


Figura 9 – Exemplo de Região simbólica x localização geográfica

Por exemplo, considere-se que o *Prédio A* é um centro de pesquisas e que o *Prédio B* é um espaço de convivência. Neste caso, os usuários dentro da região simbólica da figura acima podem encontrar tanto pessoas com interesses (localizados) por pesquisa quanto por diversão. Ou seja, quanto mais fiel for a relação entre uma região simbólica no MMS e o seu mapeamento para uma área geográfica no mundo real, maiores serão as possibilidades para os usuários especificarem (restringirem) seus interesses de forma precisa, e, conseqüentemente, encontrar pessoas certas no local correto..

Também pode ser observado que durante o processamento de um *matching* não é disponibilizado nenhum resultado preliminar/parcial, ou seja, somente após realizar o cálculo completo é que o resultado do *matching* é enviado para os usuários que solicitaram o serviço.

Em cenários com muitos usuários (centenas a milhares), uma funcionalidade que libere os resultados de *matching*, à medida que estes são computados, poderia ser necessária para garantir tempos de resposta curtos. Por exemplo, o MMS poderia calcular o *matching* de 1:N (para grandes valores de N) e, no momento em que é finalizado o processamento do *matching* dos cem primeiros usuários, o serviço enviaria os resultados preliminares já obtidos. No entanto, como o MMS foi concebido sem levar em conta questões de escalabilidade, ou seja, para uso em cenários com um número pequeno de usuários, não se tratou de implementar este tipo de funcionalidade.

Contudo, o *matching* realizado pelo MMS indica apenas uma probabilidade de similaridade entre perfis que, por sua vez, levam em conta a totalidade de itens de interesse. Desta forma, se uma aplicação necessitar de um *matching* específico, ou seja, que pares de pessoas devam ser selecionados com relação a um único e específico interesse, então os resultados obtidos pelo o MMS podem não ser os mais adequados. O que o MMS na realidade faz é sugerir quais pessoas possuem perfis de interesses parecidos em uma localização e fornecer informações adequadas para que estas pessoas possam se comunicar.

3.7. Arquitetura do Serviço

Basicamente, o MMS é um serviço que pode ser acessado por meio de um serviço *Web* (Webservices, 2005), na sua forma síncrona, e pelo serviço de eventos da MoCA, em sua forma assíncrona. O MMS também possui uma interface de administração *Web* que facilita para o administrador configurar o serviço. Assim, serão apresentados aspectos de arquitetura e tecnologias utilizadas no MMS sob o ponto de vista de suas três formas de acesso. A título de ilustração, a Figura 10 apresenta uma visão geral da arquitetura do serviço.

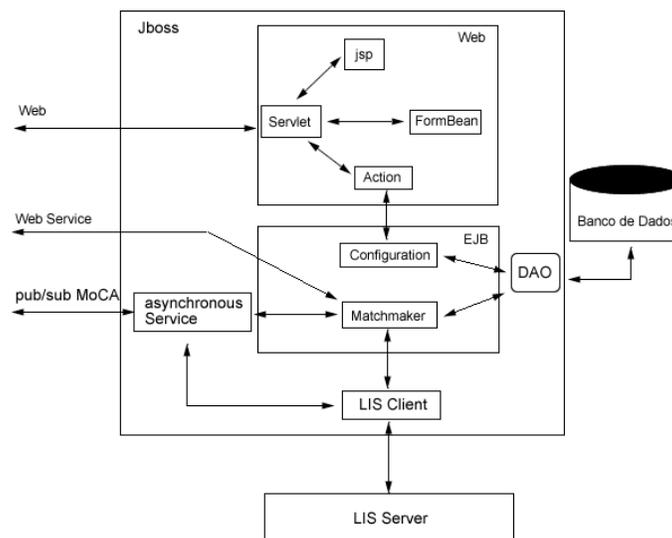


Figura 10 – Arquitetura do MMS

Como pode ser visto, o MMS é um serviço com uma arquitetura cliente/servidor. Entre os trabalhos relacionados, apenas o ActiveMatching segue este

modelo. Os demais trabalhos utilizam arquitetura P2P como foi discutido na seção de trabalhos relacionados no capítulo 1.

3.7.1. Acesso Web

Quando o administrador do serviço realiza alguma configuração no sistema, ele acessa o MMS via um navegador *Web*. As requisições *Web* são tratadas por um *servlet* executando em um contêiner chamado TomCat (Tomcat, 2005) executado sobre o servidor de aplicações J2EE JBoss (JbossAs, 2005). O *servlet* foi implementado utilizando um *framework* que segue o padrão MVC (*Módel View Control*) chamado Struts (Struts, 2005). Este *servlet* valida dados recebidos pelos formulários *Web* em classes chamadas de *FormBeans* e se comunica com componentes de configuração EJB (J2EE, 2005) através de classes que representam as ações que o usuário está executando.

Estes componentes EJB são executados em um container EJB que o servidor de aplicação JBoss oferece. Os componentes realizam todo o trabalho de configuração do MMS como configurações do banco de dados, LIS, inferências etc. Por sua vez, os componentes EJB acessam a camada de dados do serviço por meio de objetos chamados DAO (*Data Access Object*)¹⁰ (Alur, et al. 2001). O DAO é utilizado para separar o código de acesso a dados do MMS. No DAO se encontra o código do *Jena* que realiza persistência e manipulações sobre os perfis dos usuários.

Quando uma operação é realizada, os dados resultantes são enviados pelo componente EJB para o *servlet*, que, por sua vez, exibe o resultado por meio de páginas JSP (Java Server Pages). A Figura 11 ilustra a interface de administração do serviço.

¹⁰ Referente ao padrão DAO

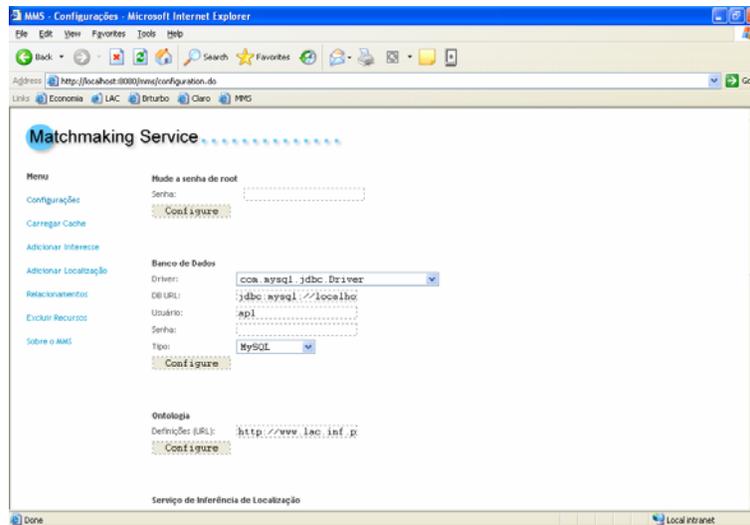


Figura 11 – Interface de administração do MMS

Entre as funcionalidades que, atualmente, o administrador pode realizar, destacam-se:

- Mudar senha de root do serviço;
- Configurar banco de dados (por exemplo, MySQL, ORACLE, PostgreSQL);
- Indicar a URL da ontologia;
- Configurar o serviço de localização (por exemplo, IP e porta que o serviço esta sendo executado);
- Configurar a *cache* do serviço (por exemplo, número máximo de objetos em *cache*);
- Configurar regras de inferência do MMS;
- Adicionar e excluir interesses e localizações;
- Indicar o relacionamento de interesses com sub-interesses;
- Indicar subdivisões em uma localização (por exemplo, *RDC* é sub-divisão de *PUC*).

Como pode ser visto, a interface de administração do MMS é usada para a configuração do serviço. Cabe ao administrador do sistema definir o conjunto de possíveis interesses (o vocabulário), que será posteriormente utilizado pelos usuários para indicar interesses específicos em cada localização. Por exemplo, em um cenário de conferência sobre ciência da computação os interesses indicados pelo administrador poderiam ser as principais áreas de conhecimento

da computação, com por exemplo: redes, engenharia de software, banco de dados, etc. Assim, um usuário que queira utilizar o MMS em uma conferência deve carregar em seu dispositivo móvel (ou melhor, em seu cliente MMS) o perfil referente à aplicação corrente. Devido à existência de um administrador e a sua limitação quanto ao número de interesses que ele consegue cadastrar, o MMS foi direcionado como um serviço que cobre poucos domínios de aplicações. Se o MMS cobrisse muitos domínios, possivelmente existiriam muitos interesses cadastrados no serviço; com poucos usuários e muitos interesses disponíveis para escolha, possivelmente haveria uma disparidade maior na indicação de interesses, o que poderia ter como consequência um número menor de encontros entre usuários.

À medida que cresce o número de usuários (devido à popularização das redes de *wireless*), o número de aplicações (e conseqüentemente interesses) de cada serviço MMS poderia ser aumentado. Entretanto, estes interesses não poderiam ser mais criados de forma estática, visto que o administrador do sistema possui uma visão limitada do mundo, ou seja, dificilmente ele imaginaria todas as possibilidades de interesses para cada cenários.

Uma forma de criar instâncias de interesses de forma dinâmica, seria com a utilização de vocabulários como o *WordNet*. A responsabilidade de manutenção dos interesses seria repassada para o *WordNet*. Desta forma, pouco a pouco o MMS poderia ser transformado em um serviço de apoio para vários cenários o que aumentaria a possibilidade de tornar o MMS um serviço global (para uma grande gama de possíveis casos de uso).

3.7.2. Acesso por Meio de Serviço Web

Como mencionado anteriormente, o serviço síncrono do MMS foi implementado por meio de um serviço Web. Existe um componente chamado *Matchmaker* que é exposto para as aplicações externas como um serviço *Web* através de uma tecnologia disponível pelo servidor de aplicação JBoss, chamada de JBoss.NET. Desta forma, o MMS recebe requisições das aplicações clientes SOAP (*Simple Object Access Protocol*) (Soap, 2003) implementadas sobre HTTP (*Hypertext Transfer Protocol*). Os dados são repassados para o componente de *Matchmaker*, que trata estas requisições acessando a camada de dados, o serviço de localização e computando a similaridade entre os perfis. Deve-se notar que este componente (*Matchmaker*)

trabalha como um *facade* (Gamma, et al. 1995) do sistema e provê uma interface única de alto nível, para um conjunto de interfaces dos subsistemas do MMS. Quando a operação estiver concluída, uma resposta é enviada pelo componente para o cliente por meio de uma mensagem SOAP.

As operações disponíveis para acesso por meio de serviços Web são:

- Realizar *matching* do tipo 1:1 ou 1:N;
- Criar, atualizar e excluir um perfil no servidor;
- Solicitar ao MMS informações sobre a localização de um dispositivo móvel.

3.7.3. Acesso por Meio do Serviço de Eventos MoCA

A operação de inscrição para notificações no MMS é realizada com o Serviço de Eventos da MoCA. Quando um cliente acessa o MMS com o serviço de eventos, a mensagem é enviada para um servidor de eventos no MMS, que, por sua vez, a repassa ao LIS. Quando o LIS detecta um evento relevante, uma mensagem é enviada ao MMS que a trata e executa operações do componente de *Matchmaker*. Quando o resultado estiver pronto, será enviado ao cliente que realizou a inscrição de interesse em notificação.

O serviço de evento da MoCA facilita a criação de aplicações assíncronas em Java. Diferentemente do serviço *Web*, que trabalha em cima de protocolos como HTTP, SMTP (*Simple Mail Transfer Protocol*) etc, o serviço de eventos da MoCA usa TCP (*Transmission Control Protocol*) ou UDP (*User Datagram Protocol*), o que, a princípio, torna a troca de mensagens mais rápida. Porém, o Serviço de Eventos da MoCA possui uma desvantagem crucial: ele é incapaz de serializar objetos que não sejam Java, tornando-o dependente da linguagem de programação.

Na próxima seção será apresentado o editor de perfis que foi construído como programa cliente do MMS.

3.8. Cliente MMS

O cliente MMS foi desenvolvido para *Pocket PC*, usando a linguagem de programação C# (CSharp) e executando sobre *dotNET Compact Framework* (DotNet, 2005).

Para este cliente foram desenvolvidas as seguintes funcionalidades básicas que, atualmente, são capazes de:

- Criar perfis;
- Enviar perfis para que sejam armazenados no servidor MMS;
- Excluir e atualizar perfis no MMS;
- Realizar *matching* do tipo 1 pra 1;
- Realizar *matching* do tipo de 1 pra N;

A operação de criação de perfis utiliza uma base de dados interna implementada em memória. Esta base de dados armazena, temporariamente, as informações indicadas pelos usuários. Se, por um acaso, o usuário perder a conectividade com a rede sem fio no momento que ele estiver criando seu perfil, ele poderá salvar os dados no dispositivo por meio de um arquivo XML. Quando o usuário recuperar a conectividade, ele poderá carregar os dados do arquivo para a base de dados e continuar a editar o seu perfil. Dependendo da aplicação, o arquivo em XML também facilita que diferentes conjuntos de instâncias de interesses sejam configurados na aplicação cliente e, posteriormente, apresentados ao usuário.

Um exemplo interessante da utilização de informação de contexto se dá no momento em que o usuário indica seus interesses. Como visto no capítulo anterior, um interesse para um usuário pode ter um nível X em um local M e um nível Y em um local N . A idéia é que os interesses sejam indicados à medida que o usuário se movimenta pelas regiões simbólicas. Assim, a informação de contexto de localização corrente do usuário é apresentada no momento em que ele está escolhendo os interesses relevantes para esta localização. Quando o perfil é enviado ao MMS, os dados armazenados na base em memória são lidos e convertidos para uma *string* válida no formato OWL.

A operação de *matching* de 1:1 é utilizada apenas para que um usuário verifique qual a similaridade entre o seu perfil e o perfil de interesses de um usuário, previamente conhecido, em uma determinada localização, ou seja, esta operação tem pouca utilidade para se descobrir novos parceiros.

Já a operação de *matching* de 1:N ajuda na descoberta de outros usuários com perfis similares, em uma certa localização. Esta operação realiza o *matching* entre o usuário que requisitou o serviço e os usuários que estejam co-localizados, considerando apenas os níveis de interesse de cada usuário para a região em questão.

Uma funcionalidade interessante, mas não implementada pelo cliente MMS, é a de comunicação entre os usuários. Seria interessante que o cliente MMS apresentasse, também, características de programas de mensagens instantâneas como, por exemplo, AIM (AOL, 2005), ICQ (ICQ, 2005), MSN Messenger (Messenger, 2005) etc, e seguisse protocolos de comunicação padronizados como o Jabber (Jabber, 2005). Atualmente, para usuários se comunicarem, depois de realizar uma consulta de *matching* no servidor MMS e obter uma resposta, precisam se comunicar com os outros usuários por meio de outras ferramentas, por exemplo, via e-mail.

O cliente MMS foi construído sem um projeto de interface gráfica. Este trabalho considera importante um projeto como este, pois, durante o desenvolvimento, notou-se uma visível dificuldade em se implementar uma boa interface devido às limitações do tamanho de tela do dispositivo e à falta de componentes de interface que são normalmente disponíveis para aplicações desktop. A

Figura 12 ilustra a interface gráfica atual do dispositivo que mostra a tela com os resultados de um *matching* 1:N, supostamente solicitado pelo usuário.

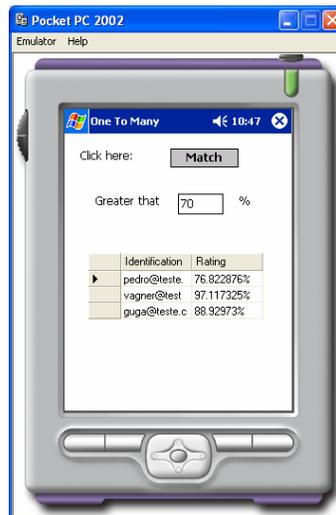


Figura 12 – Interface do cliente MMS

Em princípio, deveria ser possível (e desejável) que usuários também acessem o cliente para se inscrever no servidor MMS com o objetivo de obter notificações assíncronas. Por enquanto, porém, não foi possível implantar tal funcionalidade, pois o serviço de eventos da MoCA, atualmente, trabalha apenas com serialização de objetos em Java. A escolha de C# na construção do editor não teve nenhum motivo em especial, todavia, ela trouxe a comprovação de que

serviços *Web*, de fato, facilitam a integração entre programas escritos em linguagens diferentes.