

# Ambiente Multiagentes para Recuperação de Informação na Rede Municipal de Informática de Belo Horizonte

## Marcus Vinícius Pinto<sup>1</sup>

*Analista de Sistemas da Empresa de Informática e Informação do Município de Belo Horizonte – PRODABEL*

*Gerente do Setor de Aplicações em Educação e Assistência Social –SESS-PB. Doutorando em Engenharia da Computação pelo Programa de Pós-Graduação em Engenharia Elétrica da UFMG*

*Áreas de Interesse: inteligência computacional, orientação a objeto, redes neurais, ontologias, sistemas de informação e recuperação de informação em ambiente Web*

## José Luis Braga<sup>2</sup>

*Professor Titular do Departamento de Informática da Universidade Federal de Viçosa*

*Pós-Doutor em Tecnologias da Informação pela University of Florida*

*Áreas de Interesse: ontologias, sistemas de informação cooperativos, engenharia de software, inteligência computacional*

## PALAVRAS-CHAVE

Inteligência computacional – Agentes de software – Bases de dados herdadas

### RESUMO

O variado acervo de temas tratados pela Prefeitura Municipal de Belo Horizonte (PBH) compõe as bases de dados herdadas temáticas e distribuídas nas máquinas servidoras em operação na Rede Municipal de Informática (RMI). Este artigo apresenta uma aplicação desenvolvida utilizando a Unified Modeling Language (UML) e discute a aplicação da inteligência computacional na forma de agentes de software como solução para elevar o potencial de informação dos dados disponíveis na RMI e como ferramenta para recuperação não estruturada pelos usuários da rede e dos serviços da PBH.

<sup>1</sup> E-mail: mpinto@pbh.gov.br

<sup>2</sup> E-mail: zeluis@mail.ufv.br

## 1. INTRODUÇÃO

A expansão da informática e das telecomunicações tornou possível a discussão do acesso à informação como um direito da sociedade face à evolução da ciência da informação e a um contexto social de redemocratização e de crise do Estado. A análise desse direito como um fator de destaque na construção da cidadania é consequência da evolução de princípios constitucionais, como o direito à liberdade de expressão e às fontes alternativas de informação.

O direito à informação, segundo Bemfica (1997), está diretamente relacionado às “garantias fundamentais do cidadão” constantes da Constituição brasileira de 1988, resguardando aspectos civis da privacidade e da liberdade de opinião, ameaçados pelo poder invisível do segredo governamental.

Independentemente das circunstâncias específicas que tenham resultado na garantia constitucional do direito à informação no Brasil, segundo a autora, este direito ainda encontra-se impregnado de um potencial de ampliação da cidadania, por se referir a um campo cada vez mais central para a vida social: a informação.

O controle social sobre a gestão municipal, a descentralização do Estado ao nível local, a participação popular e a agilidade de comunicação intra e interinstitucional estão fortemente relacionados com a evolução das tecnologias de informação e de comunicação que, por sua vez, promoveram mudanças qualitativas no cotidiano da vida social e individual, nas relações e no modo de produzir e de pensar.

As dificuldades relativas ao processo informacional do setor governamental e os impactos de uma cidadania estratificada reduzem as chances de provimento efetivo do direito à informação e limitam sua contribuição para o controle social do Estado no Brasil e para a ampliação da participação popular no governo. No caso da Prefeitura Municipal de Belo Horizonte – PBH – a implantação da Rede Municipal de Informática – RMI – em 1996 provocou um crescimento expressivo no volume e distribuição de informação no âmbito da administração municipal e teve sua relação com o mundo da informática alterada de forma definitiva.

A RMI descentralizou o processamento e armazenamento de informações antes centralizados e monitorados por processos executados nos *mainframes* da Prodabel, Empresa de Informática e Informação do município de Belo Horizonte, e permitiu a circulação, diariamente, de milhares de bytes de informação sem interrupções. Essas *informações* estão distribuídas em servidores instalados nos diversos órgãos da estrutura administrativa da PBH.

No entanto, o atual sistema de recuperação, que pode ser denominado recuperação estruturada, baseado em janelas temáticas disponibilizadas em sistemas transacionais, faz com que as informações, embora disponíveis nas diversas bases de dados, sejam subutilizadas, limitadas por navegações pré-definidas.

Nesse tipo de estrutura não é possível ao usuário acessar dados sem telas pré-formatadas em que, a cada nova necessidade de informação, é preciso que as equipes responsáveis pelos aplicativos desenvolvam novas telas. Por questões de segurança, as telas, que por sua estrutura já limitam as agregações, não estão disponíveis aos cidadãos. Assim, as informações, além de estarem restritas a mecanismos rígidos de recuperação, mantêm-se fora do alcance do público em geral.

Entre os campos de aplicação da Inteligência Computacional, está a construção de agentes de software que, no caso específico desta proposta, referem-se ao processo de busca e filtragem de informações, referenciado como *recuperação de informações* (Erickson 1996; Hutchins 1986; Shneiderman 1983, 1987; Russel e Norvig 1995 e DiSESSA 1986) e possibilita ao usuário comum ampliar as possibilidades de acesso às bases de dados da RMI.

A proposta de um ambiente multi-agentes busca potencializar o uso dos dados de cada uma das bases de dados com o objetivo de atender aos seguintes requisitos:

- Ser um sistema de busca e recuperação de informação nas bases de dados e sistemas legados que não utiliza software proprietário;
- Conseguir recuperar informações relevantes no contexto do usuário;
- Apresentar uma arquitetura que tenha os detalhes de implantação transparentes ao usuário.
- Preservar a segurança dos dados em suas máquinas servidoras originais;
- Dotar o usuário de ferramentas para recuperação de informação independentemente da arquitetura e local de armazenamento;
- Capacitar o usuário a desempenhar os vários papéis permitidos sem restrições, ou seja, um funcionário da PBH pode ser um usuário via Internet ou ser o responsável por uma base de dados segundo sua preferência no momento do acesso;
- Tornar os dados da PBH acessíveis a qualquer usuário da Internet.

A Figura 1, a seguir, ilustra o processo. C1 e C2 representam duas consultas diferentes. D1...Di representam os domínios de conhecimento - áreas temáticas no conjunto de bases de dados da RMI. O agente de software será a ferramenta cuja função será filtrar dados para dessa forma produzir informações no escopo da consulta.

O agente de software, após interpretar as perguntas, deverá ser capaz de selecionar os domínios de conhecimento onde poderá obter subsídios para produzir informações no escopo da consulta. O agente tratará também de descobrir informações relevantes nas interseções entre os domínios de conhecimento que sejam capazes de aumentar a precisão do resultado.

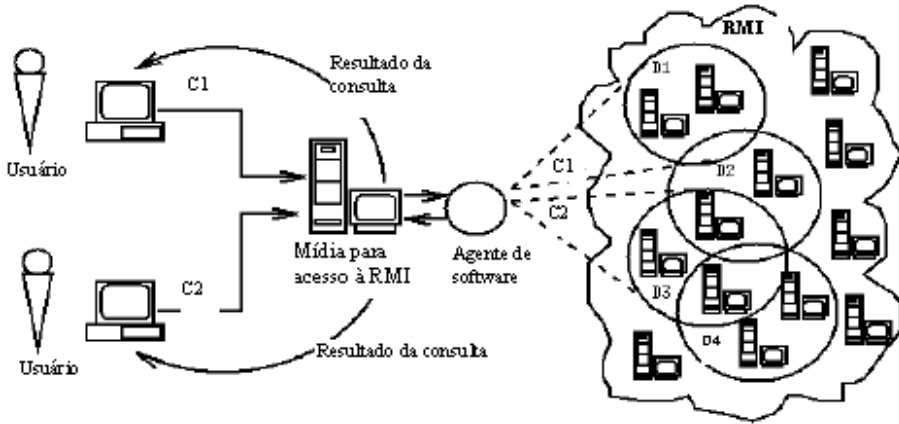


Figura 1 – Representação esquemática da atuação do agente de software proposto

## 2. CONCEITOS SOBRE AGENTES DE SOFTWARE

Uma metáfora que relaciona a figura do agente como um representante, um intermediário dotado de habilidades especiais que atua por outra pessoa, conforme Minsky & Riecken (1994), é um dos conceitos mais intuitivos para o termo agente. Aqui a principal idéia relacionada à palavra *agente* é a de um tipo de dispositivo (artefato computacional) que executa alguma tarefa, sem a necessidade da explicitação do seu funcionamento interno.

Os agentes são projetados com base em alguma especialidade particular, incluindo em sua arquitetura o conhecimento do domínio em que irá agir e as ações necessárias para conduzir esta especialidade (Russel e Norvig, 1995). Isso permite que a existência dos agentes não dependa de qualquer limite previamente existente, ou de outros agentes com quem deva interagir.

Algumas características tornam os agentes aptos a compartilhar um mesmo ambiente, a agir em um mundo multiagente, com certo conjunto de capacidades e habilidades dentre as quais destacam-se as seguintes:

- Possuem a capacidade de comunicar-se entre si com a mesma linguagem;
- Possuem conhecimentos e habilidades específicas que os tornam aptos a executar tarefas próprias a cada um;
- São capazes de se perceberem e saber da existência um do outro;
- São capazes de colaborar entre si para atingir um objetivo comum.

Na conceituação proposta por Demazeau e Müller (1991), um agente é uma entidade inteligente que possui objetivos e busca atingi-los de forma racional e intencional, conforme o estado atual de seu conhecimento. O comportamento autô-

nomo é inerente à sua própria existência e independe da existência de outros agentes. Ele tem a capacidade de criar objetivos próprios, baseados em seu interesse motivacional, e agir de forma flexível e adaptativa para alcançá-los eficazmente.

Uma outra proposta foca o agente como uma entidade de software persistente, dedicado a um propósito específico (Smith et al., 1994). Segundo Gilbert et al. (1996), agentes inteligentes são entidades de software que realizam algum conjunto de operações em benefício do usuário ou de outro software, utilizando certo grau de independência ou autonomia e algum conhecimento ou representação dos objetivos ou preferências do usuário. Para os autores, agentes inteligentes podem ainda ser definidos em função de três dimensões: agência (autonomia), inteligência e mobilidade.

Para Franklin & Graesser (1995), agente inteligente é um sistema limitado por um ambiente, que nele percebe e atua continuamente em busca de sua própria agenda, a fim de aplicar o que percebeu em um momento futuro. Sob o ponto de vista de Coen *et al.* (1995), agentes de software são programas que, através de diálogo, negociam e coordenam a transferência de informações.

Tratando da analogia com processos humanos, Shoham (1993) define os agentes como entidades às quais pode ser atribuído um estado mental. Conforme D'Amico *et al.* (1995), o estado mental de um agente pode ser entendido como o conjunto de aspectos psicológicos, crenças, escolhas, capacidades e compromissos.

Pelas conceituações apresentadas é possível perceber que um agente é, genericamente, uma unidade computacional independente, com algum grau de inteligência, representado pelos estados mentais internos, que possui a capacidade de executar ações conjuntas com outras unidades, numa intenção mútua de atingir um objetivo global em determinado momento.

Para tipificar os agentes pode-se compará-los ao comportamento humano, ficando-os em dois tipos básicos: reativos e cognitivos.

Os agentes reativos não armazenam memórias, registros de ações anteriores ou previsão de ações futuras. Seu comportamento segue processos da categoria estímulo-resposta, atuando de acordo com a percepção das mudanças no ambiente.

Em uma sociedade composta por agentes reativos, o comportamento *inteligente* emerge do comportamento de uma coleção de entidades simples; como consequência, uma sociedade reativa é composta por um grande número de membros. Não existe uma representação do ambiente ou dos agentes nem uma comunicação de alto nível.

A modelagem desses agentes lista ações em partições do mundo, em função apenas das limitações sensoriais dos agentes. Genesereth (1997) define ainda um outro tipo de agente reativo, um agente “histerético”<sup>3</sup>, que é um agente especializado capaz de identificar e memorizar respostas típicas a determinados estímu-

---

<sup>3</sup> Termo derivado de “Histerese”, um fenômeno que ocorre quando a resposta de um sistema a uma solicitação externa se altera conforme varie esta solicitação (Correa Filho, 1994).

los. Esse dinamismo em relação às suas ações representa uma espécie de mapeamento de suas percepções do ambiente.

O agente cognitivo, na análise de Correa Filho (1994) e Sichman et al. (1992), é uma entidade deliberativa que possui uma representação clara do ambiente e dos outros agentes no ambiente, podendo tratar ações passadas e planejar ações futuras. A ação inteligente ocorre quando o agente sabe seu objetivo e sabe também quais ações são necessárias para atingi-lo. De posse disso, ele *delibera* sobre aquela ação e a executa.

Um agente cognitivo precisa possuir qualidades que evidenciem seu comportamento inteligente. Essas qualidades são representações de estados mentais que facultam a interação com outros agentes. Entretanto, para que ele possa interagir com outros agentes, deve ter algum conhecimento dos demais agentes, do ambiente e das possibilidades de mudanças, através de ações que é capaz de executar. Para se predispor à interação, sempre que for necessário, ele deve igualmente conhecer o que os outros agentes sabem sobre o mundo e sobre ele mesmo; assim, a idéia é a de processamento simbólico. Esta abordagem necessita de representações simbólicas internas e externas a fim de predispor o agente à interação e cooperação com outros agentes, representar os estados mentais e viabilizar capacidades como comunicação, raciocínio e aprendizagem.

A *inteligência* para Auer (1995) pode ser considerada como um conjunto de recursos, atributos e características que habilitam o agente a decidir que ações executar e que o tornam apto a negociar efetivamente com ambigüidades. Durante o processo de determinação da ação mais adequada à situação, o agente defronta-se com ambigüidades nos mais diversos níveis. A inteligência pode ainda ser entendida como a medida da capacidade de raciocínio e aprendizagem apresentada pelo agente. A capacidade de desenvolver raciocínio é uma das principais características da inteligência que diferem agentes inteligentes de agentes robóticos (Gilbert et al., 1996).

O raciocínio é tratado por Roseler e Hawkins (1994) como uma característica altamente dependente do grau com que os agentes conduzem seus traços antropomórficos, ou seja, o grau com que demonstram emoções, crenças e intenções semelhantes às humanas. Belgrave (1995) ressalta a habilidade do agente de inferir e extrapolar de maneira racional e reprodutiva, através da propriedade de raciocínio, embasado em conhecimento e experiências anteriores. O raciocínio desenvolve-se baseado em três cenários: regras, conhecimento e evolução artificial pela geração de novos agentes. A propriedade da inteligência pode ser verificada na aprendizagem, comunicação, mobilidade e outras (Auer, 1995).

Segundo Wooldridge & Jennings (1994) a *reatividade* é a propriedade que capacita os agentes a perceber seus ambientes e a responder às mudanças neles ocorridas. O ambiente de percepção do agente pode ser um recorte do mundo real, uma

coleção de outros agentes, um usuário através de uma interface gráfica, a Internet ou uma combinação dos itens anteriores.

A *habilidade social*, tratada por Paraíso e Kaestner (1996), é a característica que habilita os agentes a interagir com outros agentes ou pessoas, no momento adequado, para concluir suas tarefas ou ajudar outros agentes. Para o autor é imperativo que os agentes possuam um mecanismo que identifique o momento em que as interações sociais são convenientes, devendo também ter um meio de comunicar suas necessidades para que possam exercitar a habilidade social. Quando os agentes não apresentam uma linguagem comum, devem ser capazes de mapear e transmitir seus problemas através de um formato de representação compreendido pelos demais agentes.

A *cooperatividade* é a capacidade dos agentes inteligentes de trabalhar juntos para concluírem tarefas mutuamente benéficas e complexas. Para tanto, os agentes devem possuir “espírito de colaboração” a fim de criar e obter êxito nos sistemas orientados a agentes (Gilbert et al., 1996).

Em Sistemas Multi-Agentes, SMA, os agentes são independentes e compartilham o mesmo ambiente competindo pelos mesmos recursos limitados (como tempo e espaço), para alcançar a eficiência e cooperam entre si, obtendo benefícios ou apenas ajudando outros agentes e, conseqüentemente, evitam conflitos (Paraíso e Kaestner, 1996). Foner (1994) estende a cooperatividade dos agentes aos usuários ao considerar que usuário e agente estão essencialmente em cooperação mútua durante um diálogo para a construção de um acordo entre eles.

Belgrave (1995) conecta as propriedades de *aprendizado* e *comportamento adaptativo* e as conceitua como a habilidade apresentada pelo agente de acumular conhecimento baseado em experiências anteriores e, como resultado, modificar seu comportamento em resposta a novas situações.

A aprendizagem, conforme Auer (1995), é a capacidade que um agente deve possuir para executar uma tarefa com maior eficiência do que em execuções anteriores. Caso a capacidade de aprendizagem seja desconsiderada, o agente reagirá sempre da mesma forma para um mesmo ambiente e uma mesma situação. Segundo Lemon et al. (1994), um agente que fornece informações sobre o ambiente, domínio de conhecimento ou a maneira como realizar uma determinada tarefa *online*, está habilitado a aprender através de instruções que podem ser: unidirecionais, quando o usuário fornece conhecimento ao agente através de séries de instruções seqüenciais ou interativas, quando o usuário fornece informações ao agente à medida que as tarefas são executadas.

*Pró-atividade* é a propriedade apresentada pelo agente de exibir um comportamento direcionado a objetivos, de não agir simplesmente em resposta ao ambiente mas sim, de acordo com um propósito (Wooldridge & Jennings, 1994). Como conseqüência, deve exibir um comportamento oportunístico, focado na realização de seus objetivos (Wooldridge & Jennings, 1996).

A *persistência*, para Belgrave (1995), é o comportamento apresentado pelo agente ao manter um estado interno íntegro através do tempo, sem alterá-lo ao acaso. *Personalização* é a capacidade do agente de personalizar tarefas e ajudar pessoas a desenvolvê-las da melhor maneira possível, conforme afirma Foner (1994). Idealmente, deveria haver componentes de memória e treinamento no agente, uma vez que é necessário o aprendizado da tarefa para posterior orientação da maneira de desenvolvê-la. A característica de personalização atribuída ao agente, ainda segundo Foner (1994), possibilita que usuários diferentes executem uma mesma tarefa de forma única e eficaz.

A *representatividade* é a propriedade de representar o usuário através de ações (Auer, 1995). Nesse sentido, a representação é o que Wooldridge & Jennings (1994) definem como “robusta noção sobre agência”, devendo ser vista como extensão dos atributos, desejos, crenças e propósitos da entidade representada, incorporados ao agente. Entretanto para que a representação seja significativa, os propósitos da entidade que está sendo representada devem ser extensivos ao agente.

Para a aceitação do agente pelo usuário é necessário, segundo a Wooldridge & Jennings (1994) e Gilbert et al.(1996), tratar a propriedade da confiabilidade, ou seja, os usuários precisam ter certeza de que os agentes serão fidedignos nas ações e informações, e irão agir em seu benefício. Para evitar conflitos, Foner (1994) acrescenta que a utilização de um agente precisa ser antecedida de uma avaliação entre o risco do agente desenvolver a tarefa erroneamente e a confiabilidade de que ela será executada corretamente, conforme especificação do usuário. Esta avaliação deve considerar dois aspectos: (1) a concepção interna do usuário, em função do nível de confiança do usuário no agente, no que diz respeito a “o quê” o agente irá fazer e (2) o domínio de interesse, em função do ônus que um equívoco pode provocar ao usuário.

Considerando as propriedades apresentadas, os agentes mais inteligentes são os que aprendem e adaptam-se. Em sistemas tais como os do tipo assistente pessoal, o agente pode identificar novos relacionamentos, conexões ou concepções, independentemente do usuário, podendo explorar tais relacionamentos, para antecipação e satisfação das necessidades do usuário. Além disso pode-se considerar o seguinte conjunto de características:

- Agente deve ser um software robusto, pois em função de ser autônomo e presumidamente estar desenvolvendo algo de certa importância, deve estar apto a responder a alterações inesperadas em seu ambiente computacional;
- Quanto à mobilidade no contexto da análise da agência, os agentes podem apresentar as características: estático, roteiros móveis e móveis com estado;
- Agentes em geral são variáveis no tempo - agentes possuem memória e alteram o que fazem com o passar do tempo;

- Agentes engajam-se em diálogos pois não se envia comandos para agentes mas sim, mantém-se comunicação com eles;
- Agentes são autônomos e inteligentes considerando que eles respondem a estímulos complexos com sofisticado e apropriado comportamento.

Os agentes estão, em geral, distribuídos em uma rede e, em função da distribuição, o comportamento do agente pode ter efeito local ou global. Assim, a abstração pode tornar-se muito complexa se o agente for responsável por muitos eventos não-locais. Belgrave (1995) define agência por meio de atributos fundamentais, os quais devem estar presentes no agente, e atributos auxiliares considerados opcionais. Para o autor, os atributos fundamentais são persistência, autonomia, reatividade, comunicabilidade, pró-atividade, mobilidade, raciocínio, aprendizado, comportamento adaptativo e habilidade para planejar.

Ao serem considerados em conjunto, estes atributos definem o agente de software como um novo paradigma, diferente de sistemas orientados a objetos, inteligência computacional e computação distribuída (Wooldridge & Jennings, 1996). As principais características verificadas na literatura podem ser resumidas conforme a Tabela 1.

A análise da síntese apresentada na Tabela 1 destaca que a propriedade de autonomia é fundamental em um agente. Também é possível constatar que as propriedades de comunicabilidade e reatividade são consideradas essenciais para uma grande maioria dos pesquisadores. Além disso verificou-se que a inteligência pode ser considerada como um componente da autonomia pois a decisão de qual ação adotar depende desta propriedade, quer em maior ou menor grau.

Tabela 1- Atributos dos agentes

|                              | Inteligência | Reatividade | Hab. Social | Cooperatividade | Aprendizagem | Pró-atividade | Persistência | Representatividade | Confiabilidade | Comunicabilidade | Mobilidade | Autonomia |
|------------------------------|--------------|-------------|-------------|-----------------|--------------|---------------|--------------|--------------------|----------------|------------------|------------|-----------|
| <b>Nissen</b>                | F            |             |             | F               |              |               |              |                    | F              | F                |            | F         |
| <b>Coen</b>                  | F            | F           |             |                 | A            |               |              |                    |                | F                | F          | F         |
| <b>Wooldridge e Jennings</b> |              | F           | F           |                 |              | F             |              |                    | A              |                  |            | F         |
| <b>Belgrave</b>              | A            | F           |             |                 | A            | A             | F            |                    |                | F                | A          | F         |
| <b>Gilbert et al.</b>        | F            |             |             | A               |              |               |              |                    | A              |                  | F          | F         |
| <b>Foner</b>                 |              |             |             | A               |              |               |              |                    | A              | A                |            | F         |
| <b>Auer</b>                  | A            | F           |             |                 | A            |               |              | F                  |                | A                | A          | F         |

Legenda: F = Propriedade fundamental A = Propriedade auxiliar

A *inteligência* é considerada uma propriedade inerente ao processo de determinação de escolhas entre alternativas, possibilitando a autonomia. Entretanto, se considerada isoladamente, não é tida como atributo fundamental da agência. Auer (1995) conclui que, para a definição de agência, autonomia e representatividade são atributos fundamentais, sendo a representatividade considerada a propriedade mais importante para determiná-la.

Assim, verifica-se, através dessa revisão bibliográfica, que as propriedades desejáveis em um agente são responsáveis pelo seu grau de complexidade e robustez. No presente trabalho, face ao ambiente em que o agente proposto irá operar - a RMI - e ao perfil de seus usuários, serão consideradas imprescindíveis ao agente proposto as propriedades: reatividade, autonomia e comunicabilidade e chega-se ao conceito adotado neste trabalho para um agente de software:

“Um agente é um software que utiliza a comunicação para negociar e coordenar a transferência de informações, agindo autonomamente sobre si próprio ou sobre o ambiente, buscando realizar os objetivos e tarefas para os quais foi projetado”.

### **3. ARQUITETURA DE AGENTES**

A arquitetura de um agente trata a especificação de como os processos internos se organizam para que sua interação com o mundo externo seja viável. Para que um agente demonstre um comportamento realmente inteligente, seus estados “mentais” devem fazer parte do núcleo da arquitetura e também da modelagem do comportamento externo do agente (Correa Filho 1994, Genesereth & Nilsson 1988). Segundo os autores, uma arquitetura global é composta por quatro componentes básicos: crença, desejo, intenção e expectativa.

A infra-estrutura que torna possível aos agentes atuarem em um ambiente são os recursos de hardware e software necessários para sua execução, os quais podem variar em função de suas características. A modelagem interna de um agente identifica o nível de conhecimento necessário à execução de suas metas e evidencia a sua competência em relação à sociedade.

Concordando com Luck & d’Inverno (1995) quando trataram um agente como instanciações de objetos, Demazeau & Müller (1991) acrescentam que um agente pode ser dividido em duas partes: uma parte “estática”, que representa o conhecimento e a estrutura do agente através da sua arquitetura, e uma parte “dinâmica” relacionada aos modelos de processamento deste conhecimento para alcançar seus objetivos. Para Demazeau & Müller (1991), qualquer agente deve possuir uma representação a respeito do mundo ou do problema a resolver. Esta representação constitui o conhecimento adquirido através da percepção do ambiente ou da comu-

nicação com outros agentes, de onde se pode abstrair o conjunto de objetivos da sociedade.

Um agente deve possuir capacidades de raciocínio que o levem a derivar as soluções possíveis para seus objetivos conforme seu conhecimento. Se várias soluções aparecem, ele deve possuir capacidade de decisão para optar por uma delas. Estas considerações estão sintetizadas no modelo do agente genérico proposto por Demazeau (1993), esquematizado na Figura 2.

Para que um agente tenha importância em uma sociedade, o seu conhecimento deve refletir uma representação explícita e abstrata do ambiente e dos objetivos; ele deve ter habilidade em descrever sua competência e seus desejos, bem como conhecer a competência dos outros agentes. Além disso, deve incorporar os planos a serem executados e as alternativas disponíveis para uma tomada de decisão. Os métodos de processamento incluem as propriedades de comunicação, detecção de incoerências, integração, raciocínio e decisão, raciocínio sobre o comportamento e o conhecimento dos outros agentes e mecanismos de decisão.

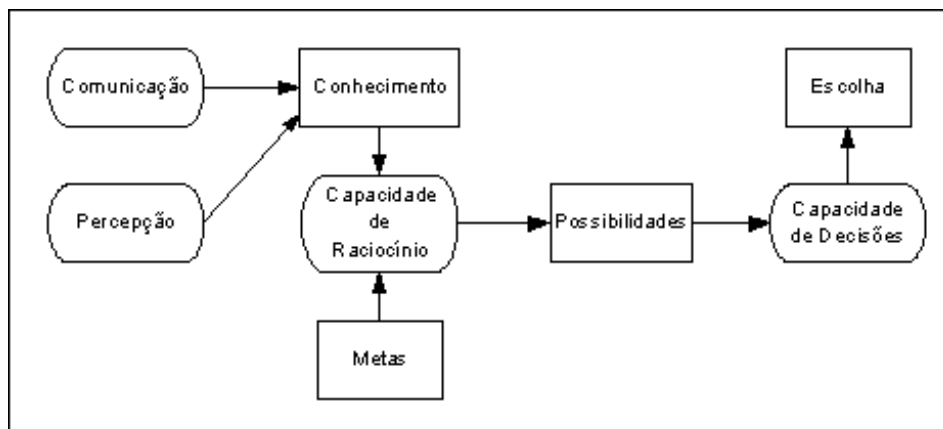


Figura 2 - Modelo de um agente genérico

Duas abordagens podem ser encontradas na literatura (Bond & Gasser 1988, Hayes-Roth 1995, Demazeau & Müller 1991, Sichman et al. 1992) que analisa a concepção de organizações, a abordagem estática, no caso de um sistema aberto, e a abordagem dinâmica, no caso de um sistema fechado. Na abordagem estática os relacionamentos entre os agentes e a estrutura de comunicação são conhecidos e definidos completamente *a priori*, durante a fase de concepção do sistema. Na abordagem dinâmica os relacionamentos e a comunicação entre os agentes são estabelecidos dinamicamente durante a fase de resolução, a exemplo dos sistemas baseados na teoria dos jogos.

É usual estruturar os agentes em uma sociedade implementada por uma rede hierárquica homogênea ou heterogênea, que determina diretamente as ligações de comunicação entre os agentes (Demazeau & Müller 1991). Para que o conjunto de agentes co-opere dentro de uma sociedade, em função de um objetivo específico, a

estrutura desta sociedade precisa garantir recursos e capacidades para facilitar a interação entre os agentes.

Também com relação às sociedades, pode-se conceber duas partes: uma estática, que se refere à representação do conhecimento, e outra dinâmica, que define como utilizar este conhecimento (Boissier & Demazeau 1991, Demazeau & Müller 1991, Stefanini & Talisman 1993). A representação do conhecimento define normas sobre o que é permitido dentro da sociedade, e o processamento do conhecimento trata como e quando o conhecimento representado será usado.

As interações básicas que permitem a troca de conhecimento, objetivos, planos ou escolhas entre agentes podem ocorrer por comunicação direta, quando os agentes conhecem um ao outro, ou por comunicação indireta, no caso em que eles desconhecem uns aos outros. Neste caso os agentes colocam e obtêm informação em alguma estrutura compartilhada.

Um exemplo de comunicação indireta é a arquitetura *blackboard*, ou arquitetura quadro-negro, onde os agentes conhecem e compartilham uma estrutura comum de dados em que as informações são escritas e lidas de acordo com o desenvolvimento da tarefa. Nessa concepção não há comunicação direta entre os agentes, a não ser por meio desta estrutura. Assim, o conhecimento é público e acessível a qualquer momento por qualquer agente. A estrutura pode conter primitivas simbólicas (hipóteses), fatos, regras, organizados por vários níveis de abstração. Se, por um lado, esta abordagem facilita o controle e a resolução do problema simultaneamente, devido ao sistema de controle, por outro lado surge a desvantagem da centralização da informação e do controle.

Conforme Oliveira (1996), ao se conceber uma sociedade de agentes, a especificação das interações que nela ocorrerão são de extrema importância, uma vez que os processos globais desenvolvem-se no âmbito dessas interações. A especificação do processo de comunicação torna-se também de igual importância, pois nele se basearão as interações. Ainda segundo Oliveira, a comunicação nas sociedades de agentes podem ocorrer de duas formas: indireta, predispondo uma arquitetura de quadro-negro, ou direta, indicando uma arquitetura baseada em troca de mensagens.

As sociedades de agentes podem ser consideradas como sistemas abertos ou fechados. Enquanto sistema fechado, a sociedade pressupõe que o projeto dos agentes e das interações seja estabelecido na fase de concepção e que a resolução do problema não necessite de outros agentes. Na consideração como sistema aberto, a sociedade designa a possibilidade de migrações de agentes de outros sistemas a fim de colaborarem na solução de um problema que pode, também, ser dinâmico. Nessa última abordagem, em que os agentes entram e saem livremente de uma sociedade dinâmica, normalmente os agentes tendem a agrupar-se em coligações a fim de suprirem mutuamente possíveis carências individuais de capacidades e de conhecimento.

O principal objetivo do agrupamento é maximizar seu benefício, minimizando seu custo, ou seja, a formação dinâmica de coligações em sistemas multiagentes advém da teoria dos jogos, onde o problema consiste em compartilhar benefícios em grupo. Na maioria dos casos, a medida de benefício é obtida em distância ou unidades de tempo nas abordagens de jogos onde ações podem, pelo menos, ser avaliadas. Entretanto, tal avaliação não é trivial em abordagens como estratégias de tomadas de decisão e alocação dinâmica de tarefas, entre outras. A abordagem proposta em Louçã & Pinson. (1996), por exemplo, procura basear a análise do agrupamento nos objetivos do agente e visa reduzir o esforço de comunicação entre os agentes nas tarefas cooperativas.

Os sistemas de quadro-negro (Figura 3) fornecem uma estrutura de dados central, denominado quadro-negro (*blackboard*), a qual é dividida em regiões ou níveis. Conforme Oliveira (1996), as arquiteturas de quadro-negro precedem aos sistemas multi-agentes. Nelas não há comunicação direta entre os agentes (Oliveira, 1996). Nessas arquiteturas todas as interações utilizam o quadro-negro. Os agentes lêem e escrevem em um ou mais níveis sob a supervisão de um mecanismo global de escalonamento.

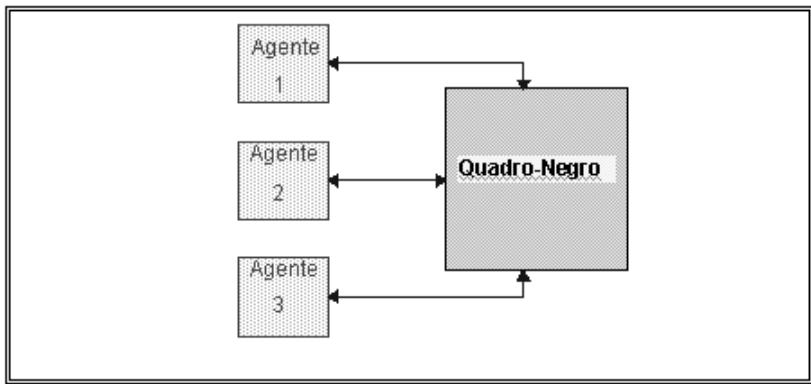


Figura 3 - Arquitetura de quadro-negro

Para Oliveira (1996), algumas questões básicas precisam ser tratadas em um sistema de quadro-negro: a granularidade das ações dos agentes, a indexação para recuperação de conhecimento dos níveis, a granularidade das representações e a sincronização e resolução de conflitos entre agentes.

Na arquitetura baseada em troca de mensagens, conforme Oliveira (1996), os agentes comunicam-se através da troca de mensagens e, nesse caso, torna-se necessário que os nomes dos agentes sejam conhecidos. O planejamento e priorização das interações é feita com base em protocolos que definem as etapas da interação entre os agentes para cada tipo de tarefa possível na sociedade. Os protocolos e os formalismos para representação de mensagens são muito variados, tais como protocolos de apresentação, troca de conhecimentos e aprendizagem cooperativa.

As Arquiteturas Reativas, conforme Oliveira (1996), são naturais aos agentes que possuem estrutura interna simples e interação de forma limitada. Usualmente não possuem representação dos estados mentais e o desempenho da sociedade é resultado do número de agentes e da rapidez nas interações.

A Arquitetura de Suposição, desenvolvida por Brooks, é uma hierarquia de comportamentos realização-tarefas, onde cada comportamento compete com outros para exercer controle sobre um robô (Wooldridge & Jennings, 1994). Essa abordagem apresenta camadas inferiores que representam tipos de componentes mais primitivos e possuem precedência sobre as camadas mais altas da hierarquia.

Vários pesquisadores afirmam que tanto as arquiteturas deliberativas quanto as reativas são inadequadas para agentes, sugerindo a arquitetura híbrida, composta simultaneamente das características deliberativas e reativas (Wooldridge & Jennings, 1994).

Uma das arquiteturas de agentes mais conhecidas, a PRS - *Procedural Reasoning System*, é baseada no conjunto Intenção-Desejo-Crença (Wooldridge & Jennings, 1994). A arquitetura PRS apresenta as características:

- Crenças, que são fatos sobre o mundo externo e sobre os estados internos do agente, estruturados segundo a lógica de primeira ordem;
- Contém uma biblioteca de planos;
- Os desejos são representados como comportamentos do sistema, em vez de representações estáticas de metas.

Um plano pode ser ativado a partir de metas, dados ou de forma reativa, respondendo de imediato a mudanças no ambiente. As intenções são o conjunto de planos ativos em um sistema.

A arquitetura *Touringmachine* é composta por subsistemas de percepção e ação, que agem diretamente no ambiente do agente, e têm três camadas de controle, embutidas em uma estrutura de controle a qual negocia entre estas camadas. As camadas, segundo Wooldridge & Jennings (1994), são concorrentes, independentes e produtoras de atividade, sendo constituídas por uma camada reativa e uma camada de planejamento.

A camada reativa gera cursos de ações potenciais em resposta a eventos que ocorrem muito rapidamente para que os outros níveis possam negociar. É implementada como um conjunto de regras situação-ação. A camada de planejamento constrói planos e seleciona ações a executar para atingir as metas do agente. Esta camada, por sua vez, apresenta dois componentes: (1) o planejador, que integra a geração e execução dos planos e utiliza uma biblioteca de planos, parcialmente elaborados com um mapa topológico do mundo, para construir novos planos e (2) o mecanismo de Foco de Atenção que limita a quantidade de informação recebida pelo planejador, para melhorar sua eficiência.

A partir do estudo das arquiteturas de agentes e aplicações, pode-se constatar que a tecnologia de agentes beneficia o usuário, facilitando a execução de tarefas, seja através do uso de agentes assistentes, seja pela maximização dos recursos computacionais, no que diz respeito à mobilidade de softwares e execução de tarefas de forma corporativa.

## **4. ESPECIFICAÇÃO DE UM AGENTE PARA RECUPERAÇÃO DE INFORMAÇÕES NA RMI**

Esta seção apresenta a proposta do modelo de uma arquitetura multiagentes, um sistema para a recuperação de informações na RMI baseado na orientação a objeto. O sistema recebeu a denominação *Jnanam* que, em sânscrito, significa conhecimento.

A denominação pretende orientar o uso do sistema para que os dados que venham a ser recuperados, como resultado de seu processamento, ultrapassem sua utilização como informação e sejam formadores do conhecimento daqueles que venham a utilizá-lo. Para a modelagem do sistema, baseada nos conceitos da orientação a objetos, utilizou-se a linguagem UML – *Unified Modeling Language* – em que a arquitetura lógica do sistema está especificada nos diagramas de caso de uso, classe e seqüência, e a arquitetura física está especificada nos diagramas de componente e distribuição

### ***4.1 Arquitetura Lógica***

Os estudos indicam que os atores externos ao sistema são o funcionário da PBH, de qualquer nível ou local de trabalho; o analista da Prodabel responsável pela implementação e manutenção de bases de dados e qualquer usuário que esteja acessando os dados disponíveis nos servidores da RMI via Internet, residente ou não em Belo Horizonte.

A análise da interação entre os atores externos e o sistema aponta para uma arquitetura em que agentes especialistas interagem com o usuário e solicitam a execução de tarefas a uma agente responsável pelo acesso às bases de dados da RMI. Esta opção potencializa a segurança na RMI pois os agentes que interagem com os usuários não acessam diretamente os depósitos de dados. Na arquitetura adotada foram instanciados os seguintes objetos:

- Uma interface de acesso ao sistema para que os usuários possam navegar pelo sistema denominada Portal;
- Os agentes, que se apresentam como atores internos ao sistema desempenhando três papéis: (1) Agente-Servidor, responsável pelo acesso físico às bases de dados da RMI; (2) Agente-Cliente, designado a tratar as consultas do usuário, repassar a consulta ao Agente-Servidor

e retornar o resultado da consulta ao usuário; e (3) Agente-Applicativo, proposto para executar tarefas específicas, como atualizar dados dos usuários;

- Uma base de conhecimento composta por informações próprias do agente e bases de dados da RMI. Essa base de conhecimento implementa um dicionário sobre os dados distribuídos entre os servidores da RMI e os dados pessoais dos usuários;
- Uma estrutura de comunicação entre os agentes: o Quadro-Negro - QN. A adoção da estrutura QN permite que a arquitetura seja independente da estrutura dos agentes que interagem com o usuário.

O acervo de informações a ser incorporado à base de conhecimento dos agentes é composto de três categorias, de acordo com a forma de disponibilização na RMI: (1) bases de dados, (2) textos não estruturados tais como artigos, decretos e relatórios e (3) funções dos sistemas legados. Dentre as funções atribuídas aos agentes por Maes (1996) estão presentes na arquitetura proposta: (1) a monitoração de eventos e processos, verificada nas atribuições associadas ao Agente-Servidor, e (2) o desempenho de tarefas em favor do usuário, verificado nas atribuições das três classes de agentes atuantes no sistema.

A ativação dos agentes acontece continuamente no caso do Agente-Servidor, residente em um servidor da RMI, e sob demanda, no caso dos agentes Cliente e Aplicativo que acontece quando um usuário acessa o Portal do sistema. A partir do acesso ao Portal, os agentes passam a monitorar os processos através de mensagens e solicitações de execução entre os agentes disponíveis no sistema. O Agente-Servidor é um agente estático com a função de controlar os acessos às bases de dados da RMI. A arquitetura deste agente caracteriza-se pela presença das propriedades de inteligência, autonomia, representatividade, flexibilidade, habilidade social, reatividade e outras. Os processos são solicitados diretamente ao Agente-Servidor pelo Agente-Cliente ou pelo Agente-Applicativo.

O Agente-Applicativo é o agente acionado sempre que o processamento da consulta do usuário envolva acesso às bases de dados da RMI e o Agente-Cliente quando o processamento envolve apenas acesso às bases de dados no servidor que hospeda o Agente-Servidor. Os Agentes Aplicativos executam também as tarefas específicas de manutenção e monitoramento do ambiente.

A ativação do Agente-Cliente e/ou do Agente-Applicativo implica a transferência desses agentes para a memória da máquina do cliente, caracterizando-os como agentes de Roteiro Móvel. Da mesma forma que o Agente-Servidor, eles são proativos, visto que disponibilizam processos para execução sem exigir a intervenção do usuário para o qual atuam. Entre as funções desses agentes estão a de criticar e selecionar as informações recebidas do usuário para enviá-las ao Agente-Servidor e agrupar e enviar as informações necessárias ao desempenho de outros processos. O Agente-Cliente e o Agente-Applicativo apresentam as propriedades de

inteligência, autonomia, representatividade, flexibilidade, habilidade social, reatividade e outras.

Os cenários de utilização do sistema, de acordo com os atores que os iniciam, estão organizados em 9 casos de uso listados na Tabela 2.

Tabela 2 - Casos de uso de *Jnanam*

| Nº | Caso de uso                               | Quem inicia   | Descrição geral do caso de uso  |
|----|---|---|---|
| 1  | Cadastramento de usuário                  | Analista_Prodabel<br>Funcionário_PBH<br>Usuário_WWW | O usuário aciona o portal do sistema para solicitar seu cadastramento e informa seu perfil.   |
| 2  | Implementar dicionário                    | Analista_Prodabel                                   | O analista aciona o portal do sistema para solicitar o cadastramento das informações do dicionário de dados. O portal aciona o agente aplicativo que gerencia a tarefa e a comunicação com o agente servidor.                                       |
| 3  | Manutenção em dicionário                  | Analista_Prodabel                                   | O analista aciona o portal do sistema para solicitar o processamento da manutenção. O portal aciona o agente aplicativo que gerencia a tarefa e a comunicação com o agente servidor.  |
| 4  | Recuperação de informação na RMI          | Analista_Prodabel<br>Funcionário_PBH<br>Usuário_WWW | O usuário aciona o portal do sistema para solicitar o acesso às bases de dados. O portal aciona o agente cliente que se comunica com o agente servidor através da estrutura de quadro-negro.  |
| 5  | Nova execução de consulta                 | Analista_Prodabel<br>Funcionário_PBH<br>Usuário_WWW | O usuário aciona o portal do sistema e solicita nova execução de uma consulta armazenada em sua base pessoal. O portal aciona o agente aplicativo que gerencia a tarefa e a comunicação com o agente servidor através da estrutura de quadro-negro. |
| 6  | Atualiza base de dados pessoal do usuário | Analista_Prodabel<br>Funcionário_PBH<br>Usuário_WWW | O usuário aciona o portal do sistema e solicita a atualização de sua base de dados pessoal. O portal aciona o agente aplicativo que gerencia a atualização da base de conhecimento e a comunicação com o agente servidor.                           |
| 7  | Exclusão de usuário                       | Analista_Prodabel<br>Funcionário_PBH<br>Usuário_WWW | O usuário aciona o portal do sistema e solicita a exclusão de seus dados. O portal aciona o agente aplicativo que gerencia a atualização da base de conhecimento e a comunicação com o agente servidor.   |
| 8  | Consulta base de conhecimento             | Agente-Cliente/Agente-Applicativo                   | O Agente-Cliente e/ou o Agente-Applicativo interagem com o agente servidor para ter acesso às informações da base de conhecimento.  |
| 9  | Atualiza base de conhecimento             | Agente-Servidor                                     | O agente recebe parâmetros para acesso e atualização da base de conhecimento e é responsável pela execução da tarefa.   |

O módulo gerenciador do Quadro-Negro tem como meta possibilitar ao Agente-Servidor a verificação constante das mensagens contidas na estrutura, permitindo o processo paralelo de requisições. A estrutura de Quadro-Negro armazena as mensagens emitidas pelos agentes, são recuperadas em ordem de inserção pelo Agente-Servidor e aquelas executadas são excluídas em lotes de acordo com a necessidade de espaço livre no Quadro-Negro.

O protocolo TCP/IP permite a comunicação entre os agentes Cliente e Aplicativo com o Agente-Servidor, bem como a transferência dos agentes móveis para a máquina do Cliente.

#### 4.2 As Classes de *Jnanam*

Os diagramas de classe, para Alhir (1998), representam a visão estrutural do sistema. Eles descrevem a sua estrutura estática, ou seja, como ele está estruturado em lugar de como ele se comporta. A análise dos casos de uso e da interação entre os objetos inicialmente propostos tornou possível modelar uma estrutura de classes composta por quatro classes e quatro pacotes que Eriksson & Penker (1998) denominam subsistemas ou camadas do caso estudado.

As classes e pacotes identificados no modelo proposto estão organizados de acordo com as funções que desempenham na Tabela 3:

Tabela 3 - Classes e pacotes de *Jnanam*

| Nome                         | Tipo   | Figura |
|------------------------------|--------|--------|
| Sistema                      | Classe | 4      |
| Agente                       | Pacote | 5      |
| Usuário                      | Pacote | 6      |
| Conhecimento                 | Pacote | 7      |
| Nível de restrição de acesso | Classe | 8      |
| Dicionário                   | Pacote | 9      |

A classe Sistema (Figura 4) agrega os atributos e operações de identificação e acesso ao sistema e apresenta uma interface – IUsuário. A interface IUsuário implementa o Portal do sistema.

As classes referentes aos agentes estão estruturadas no pacote Agente (Figura 5). Nesse pacote estão agrupadas as classes Agente e suas especializações: Agente-Cliente, Agente-Aplicativo e Agente-Servidor; Quadro-Negro e a interface IAgente.

Os usuários estão modelados no pacote Usuário (Figura 6). As classes integrantes da base de conhecimento estão organizadas no pacote Conhecimento (Figura 7) em que estão agrupadas as classes Base-RMI, Base-conhecimento, Base-Usuário, Dic-RMI, Regra-Recuperação e Cópia-Segurança.

As informações de segurança relativas ao nível de autorização de cada usuário e ao nível de restrição de cada elemento da base de conhecimento estão estruturadas na classe Nível\_restrição\_acesso (Figura 8).

O pacote dicionário (Figura 9) agrega as informações sobre a estrutura dos arquivos da base de dados da RMI, textos e funções. A Tabela 4 lista as descrições das classes.

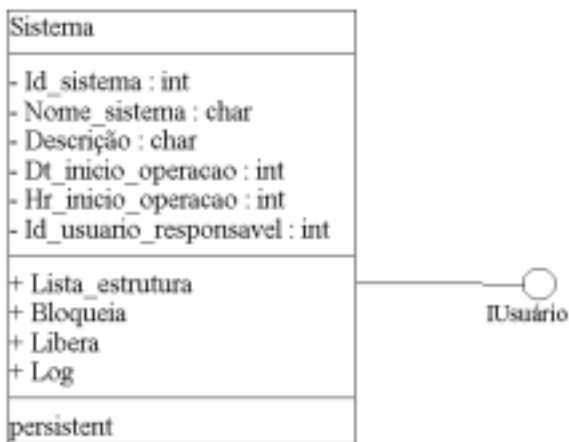


Figura 4 – Classe Sistema

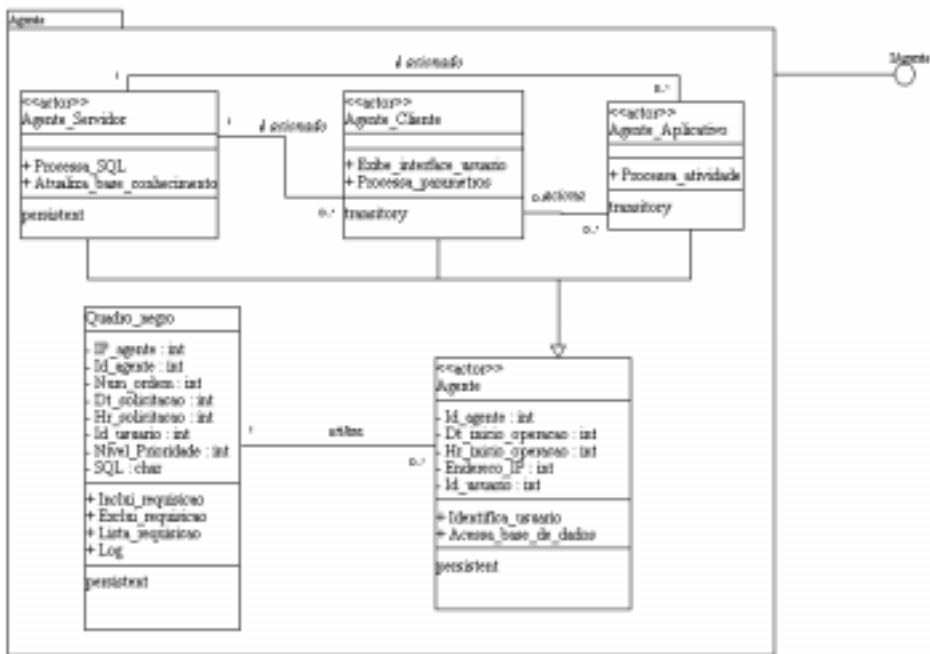


Figura 5 – Pacote Agente

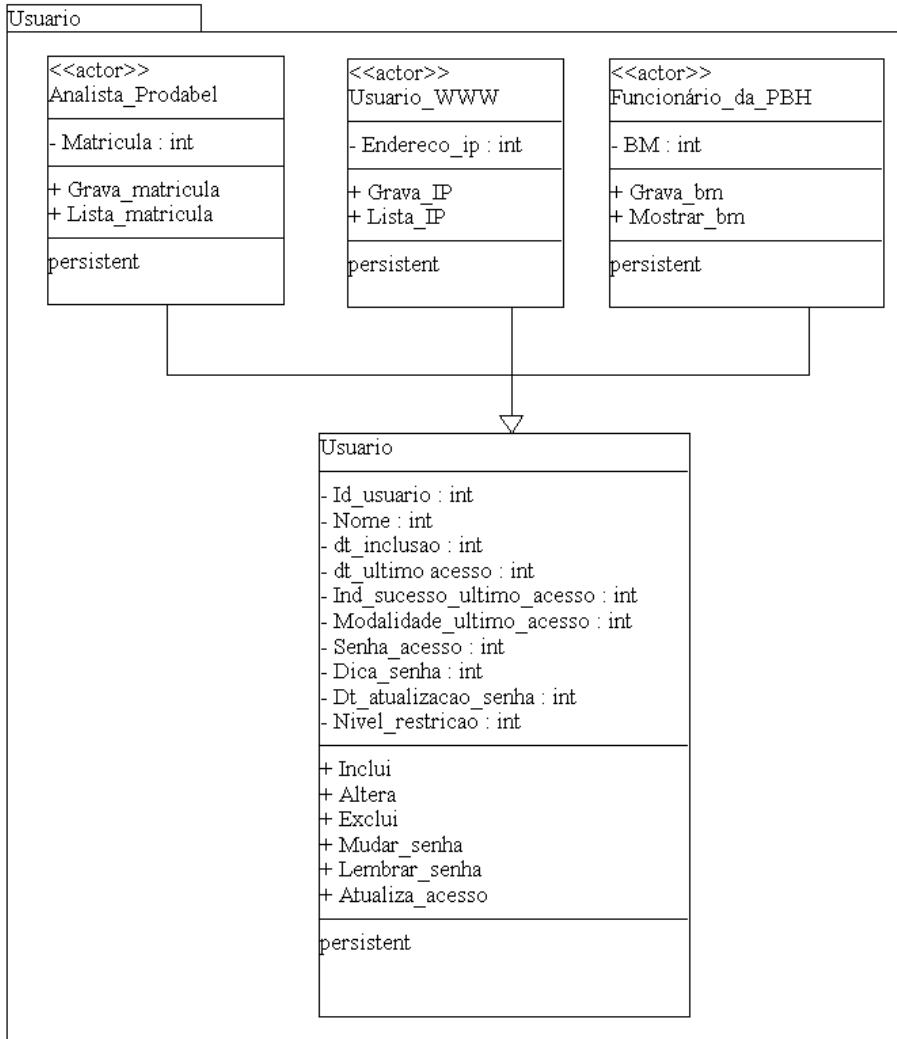


Figura 6 – Pacote Usuário

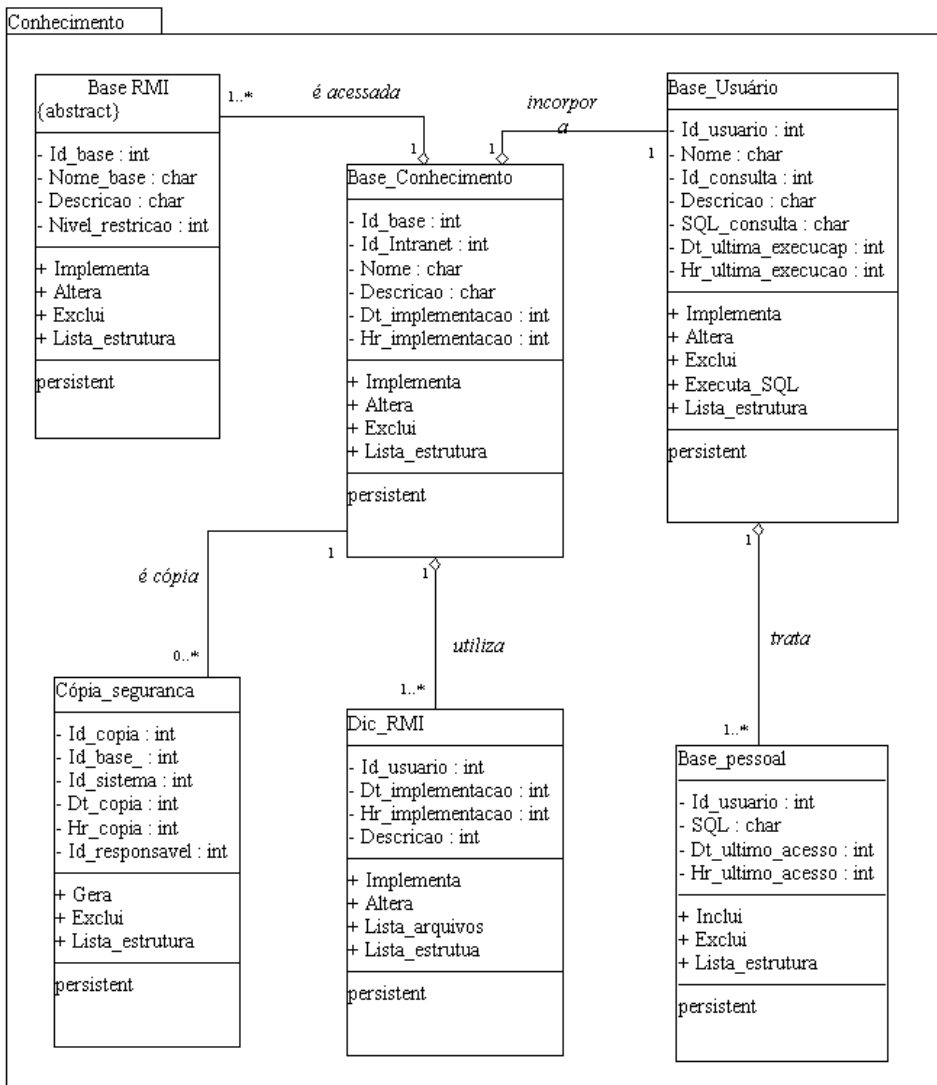


Figura 7 – Pacote Conhecimento

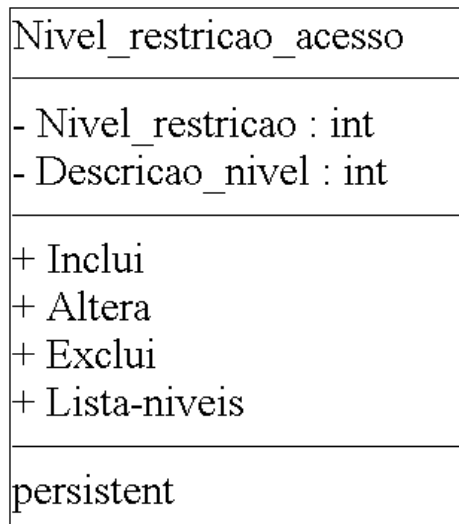


Figura 8 – Classe Nível\_restricção\_acesso

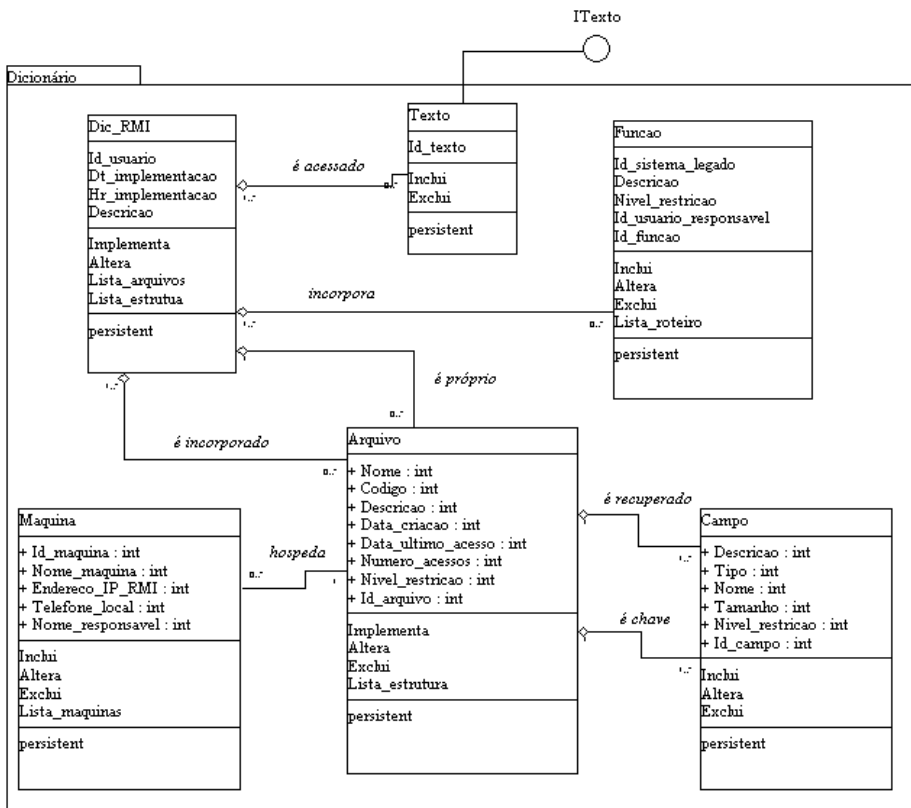


Figura 9 – Pacote Dicionário

**Tabela 4 - Descrição das classes de *Jnamam***

| Classe                 | Descrição  |
|------------------------|--|
| Sistema                | Classe que organiza os atributos e operações do sistema. Apresenta uma interface - IUsuário - para acesso às funcionalidades que implementa o Portal.  |
| Usuário                | Todo usuário de informática que tenha algum interesse nas informações armazenadas nas bases de dados da RMI.   |
| Analista-Prodabel      | Especialização da classe Usuário. É um funcionário da Prodabel responsável pela manutenção das informações em bases de dados da PBH. O papel pode ser desempenhado por outro usuário que tenha o conhecimento e nível de autorização para a função.                            |
| Funcionário-da-PBH     | Especialização da classe Usuário. É um Funcionário da PBH.   |
| Usuário-WWW            | Especialização da classe Usuário. É o usuário que acessa as bases de dados da RMI via Internet. Esse usuário pode não ter qualquer vínculo com a PBH ou ser um analista da Prodabel ou um funcionário da PBH.  |
| Agente                 | Software baseado em inteligência computacional construído para agenciar a recuperação de informações para o Usuário.   |
| Agente-Cliente         | Especialização da classe Agente. É o agente que gerencia as consultas que necessitam recuperar informações armazenadas nas bases de dados da RMI.  |
| Agente-Aplicativo      | Especialização da classe Agente. É o agente que executa tarefas que não necessitam recuperar informações armazenadas nas bases de dados da RMI. Um exemplo dessas informações é a base de dados pessoal do usuário que está armazenada no servidor onde Jnamam está hospedado. |
| Agente-Servidor        | Especialização da classe Agente. É o agente que analisa os pedidos de recuperação das informações armazenadas nas bases de dados da RMI e as executa. Toda informação armazenada na RMI só é acessada por ele no modelo de Jnamam.   |
| Quadro-Negro           | Classe que tem a função de armazenar as mensagens emitidas pelos agentes para posterior recuperação em ordem de inserção pelo Agente-Servidor.   |
| Nível_restrição_acesso | Informações sobre a restrição de acesso de uma base de dados e nível de autorização de acesso dos usuários.  |
| Base-conhecimento      | Agrega as informações que compõem a base de conhecimento de Jnamam. É composta pela base de dados da RMI, base de dados pessoal dos usuários, dicionário dos dados armazenados.  |
| Base-RMI               | Elemento abstrato que referencia os conteúdos das bases de dados da RMI.   |
| Base-Usuário           | Base de dados pessoais do usuário que, entre seus atributos, está o SQL de consultas executadas.   |
| Regra-recuperação      | Parâmetros necessários para a recuperação de uma informação.   |
| Cópia-segurança        | Informações sobre as cópias de segurança da base de conhecimento.  |
| Dic_RMI                | Classe que reúne as informações sobre os dados armazenados na RMI.   |
| TI=exto                | Informação sobre as estruturas textuais indexadas no ambiente.   |
| Função                 | Elemento que agrega informações sobre as funções disponíveis nos sistemas legados no ambiente da RMI.  |
| Arquivo                | Informação sobre as estruturas de dados da RMI.  |
| Campo                  | Elemento que compõe a estrutura de dados.  |
| Máquina                | Informações sobre o computador em que um arquivo está acessível.   |
| Consulta               | Informações sobre como recuperar informações das bases de dados utilizando as regras de recuperação para responder ao pedido do usuário  |

### 4.3 Arquitetura Física

A arquitetura física tem por função modelar os detalhes contidos no sistema em termos de hardware e software. Segundo Eriksson & Penker (1998), ela é definida pelos diagramas de implementação: componentes e distribuição.

No caso de *Jnanam* os componentes estão especificados de acordo com as funcionalidades propostas. Assim, serão implementados os seguintes componentes: Portal, Agente-Servidor, Agente-Cliente, Agente-Aplicativo, Quadro-Negro e Utilitários.

O Portal tem a função de interface com o usuário apresentando as funções do sistema, além de tornar disponíveis as funções de ajuda. Os agentes implementam o código para operacionalização das classes e funcionalidades relativas ao acesso às informações. O Quadro-Negro é o componente que faz a interface entre os agentes. Por fim, os utilitários são subsistemas responsáveis por funções específicas tais como a cópia de segurança.

Por existirem como classes, os componentes serão instanciados de acordo com a iniciativa e navegação do usuário. No caso dos agentes, a modelagem das especializações, em lugar do componente genérico, se deve a uma opção do projeto permitir dar maior visibilidade à solução proposta com a plataforma multiagentes. A Figura 10 mostra o diagrama de componentes.

O Diagrama de distribuição mostra o *layout* físico dos vários componentes (nodos) do hardware que compõe um sistema, bem como a distribuição dos programas executáveis neste hardware (Alhir, 1988).

O diagrama (Figura 11) descreve a natureza da comunicação das ligações entre os vários nodos e, nesse caso, o servidor e os PCs comunicam-se via TCP/IP, enquanto a natureza das conexões entre outros nodos não é especificada e vai depender da configuração local na plataforma do usuário.

O nome destes elementos, segundo sugestão de Eriksson & Penker (1998), deve facilitar a ligação entre o diagrama de distribuição e o diagrama de componentes. Cada objeto nomeado no diagrama de distribuição executa um programa com o mesmo nome como é descrito dentro do diagrama de componente.

Os recursos de hardware, como impressoras e unidades de disquete, pressupõem unidades de qualquer modelo ou fabricante, de acordo com níveis de sofisticação definidos pelo administrador do sistema, no caso do ambiente em que o Agente-Servidor está hospedado, e de acordo com os objetivos do usuário, no caso da plataforma utilizada para consulta.

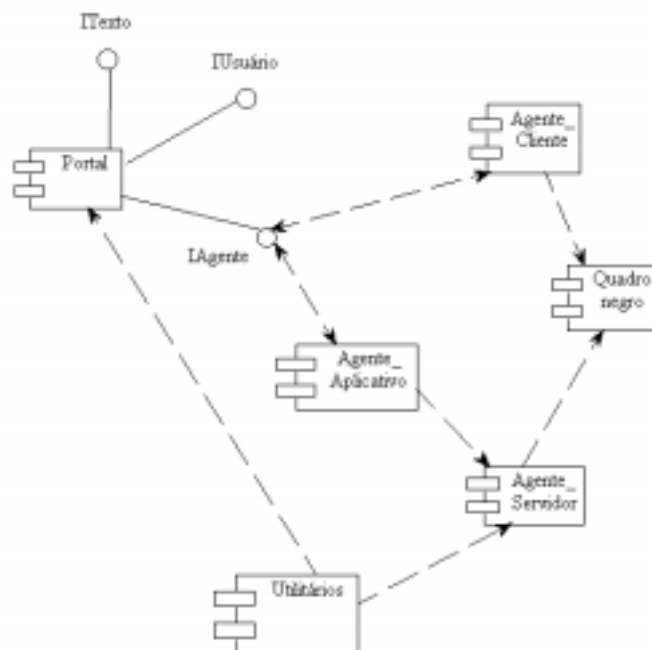


Figura 10 – Diagrama de componentes

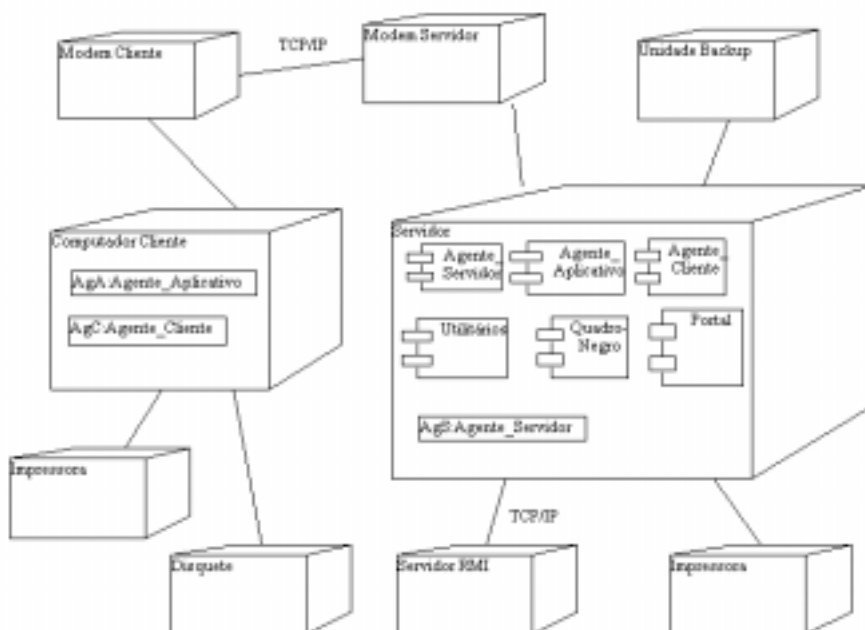


Figura 11 – Diagrama de distribuição

## 5. O AGENTE *JNANAM* NA RMI

Esta seção é destinada a demonstrar a aplicabilidade do Sistema Multiagentes e da arquitetura básica para compor o Sistema de Recuperação de Informações, bem como a viabilidade do acesso a bases de dados distribuídas, através do desenvolvimento de um protótipo do modelo proposto na seção 4. Na programação, utilizou-se a plataforma Delphi, por ser o padrão adotado na PRODABEL, e para gerenciamento do banco de dados, o Microsoft Access. O desenvolvimento tomou, como referência, os casos de uso “Implementa dicionário” e “Recuperação de informação na RMI” por apresentarem as funções essenciais do modelo proposto.

O sistema apresenta, como interface principal para o usuário, a tela t00101 que implementa o Portal e traz a imagem do personagem criado para personificar a idéia do agente que executa tarefas de acordo com os comandos e preferências do usuário (Figura 12). Estão implementadas três formas de consulta no Portal: consulta simples, consulta estruturada com auxílio do gerador de SQL e execução de consulta armazenada na base de dados pessoal.

A consulta simples é informada na caixa de texto identificada pelo rótulo “Formule uma pergunta”, a consulta estruturada utiliza a tela t00301 acionada pelo botão “Assistente” e a execução de consulta armazenada na base de dados pessoal pelo botão “Sua base”.

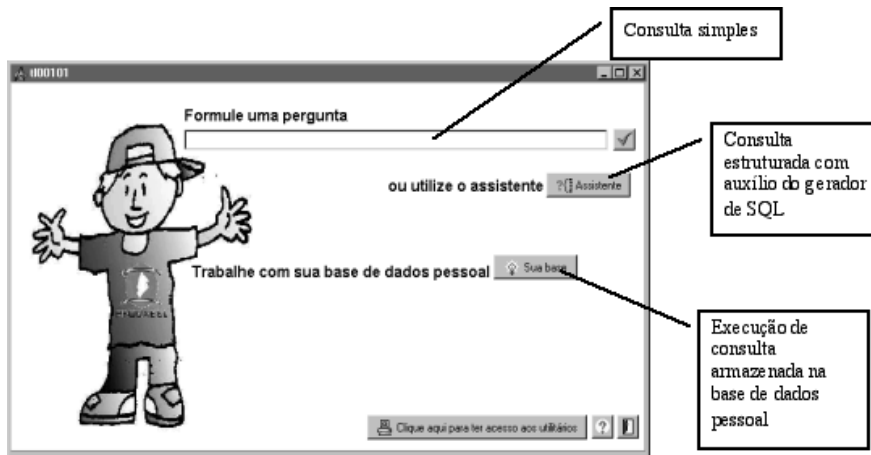


Figura 12 – O Portal e as formas de processamento da Recuperação de Informações

Para todas as modalidades será gerado um pedido para o Agente-Servidor utilizando-se a estrutura do Quadro-Negro implementada pelo arquivo correspondente.

No contexto do modelo proposto, uma consulta simples é aquela possível de ser expressa em uma frase curta. Para processamento da consulta, a interface inclui um registro no Quadro-Negro com o texto da consulta. O Agente-Servidor, ao acessar

o registro no Quadro-Negro, tentará encontrar uma regra dentre as integrantes de sua base de conhecimento utilizando o campo descrição. As regras estão implementadas no arquivo Regra-de-recuperação.

Para processamento de uma consulta estruturada, o usuário clica no botão “Assistente” do Portal que transfere o controle para a tela tl00301. Nessa tela, o usuário constrói a pergunta utilizando as caixas de seleção disponíveis. A pergunta gerada é exibida na porção inferior da tela para que seja possível analisar a pergunta completa. A Figura 13 mostra a tela tl00301 para a consulta dos dados pessoais do aluno com nome José.

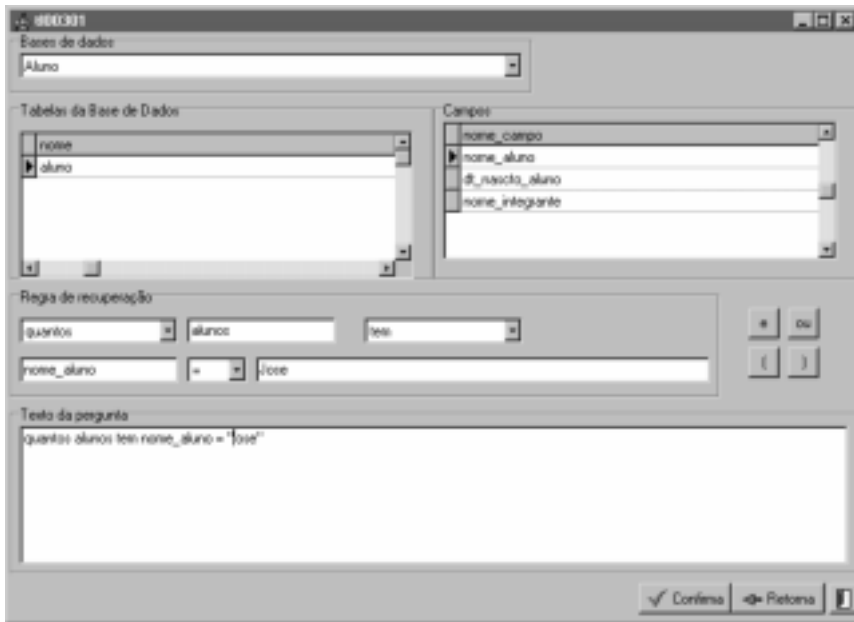


Figura 13 – Tela tl00301 com informações para uma consulta

Para enviar a pergunta para o Agente-Servidor, o usuário clica no botão “Confirma”. A interface traduz a pergunta em SQL e inclui um registro no Quadro-Negro com a pergunta. O processamento do SQL e da exibição do resultado está descrito na seção 4.3.

O usuário, ao receber o resultado de uma consulta, tem a opção de armazená-la em sua base de dados pessoal. Esta ação torna possível que a consulta seja executada novamente quando o usuário considerar conveniente. Para executar uma consulta armazenada em sua base de dados pessoal, o usuário clica no botão “Sua base” do Portal (Figura 12).

### ***3.1 Processamento do Agente-Servidor***

O Agente-Servidor, conforme análise na seção 2, é um agente estático com a função de controlar os acessos às bases de dados da RMI. Na classificação de

Wooldridge & Jennings (1994), seção 2, o Agente-Servidor implementado no protótipo é um agente proativo, uma vez que monitora e disponibiliza os processos de *Jnanam* para todos os usuários, sem a intervenção de um administrador.

O Agente-Servidor recebe o texto do SQL via parâmetro do componente SQL conforme exemplificado na Figura 14.

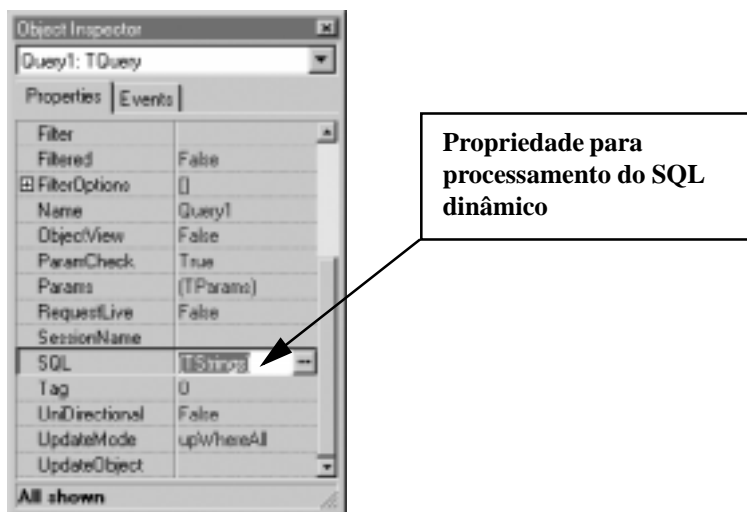


Figura 14 – Object Inspector do componente Query1 do Agente-Servidor

A Figura 15 mostra a interface do Agente-Servidor para a consulta que totaliza o número de alunos matriculados na rede municipal de ensino.

O Agente-Servidor, no protótipo implementado, executa o SQL identificando os locais de armazenamento físico dos arquivos de acordo com a estrutura informada na base de conhecimento.

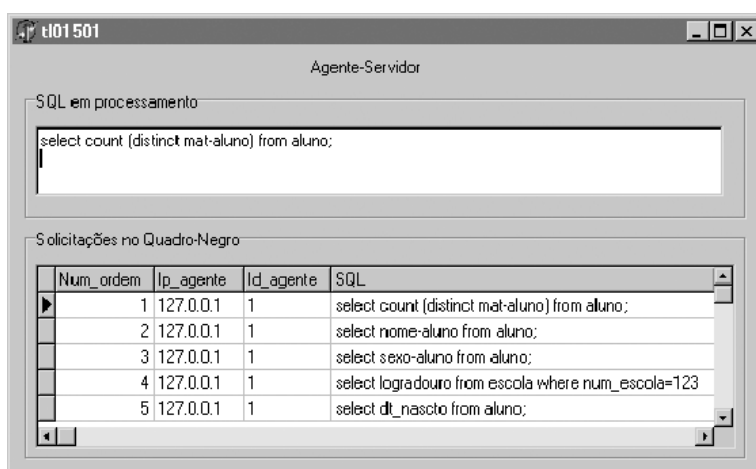


Figura 15 – Interface do Agente-Servidor com informações sobre as consultas em processamento

Para acesso às máquinas da rede, o Agente-Servidor utiliza a tecnologia ADO – *ActiveX Data Objects* – da Microsoft, que dispensa a instalação do BDE, software para interfaceamento entre a aplicação e os gerenciadores de banco de dados, nas máquinas onde os arquivos a serem acessados estão fisicamente instalados. O resultado da consulta é gravado no diretório Agente-Servidor (Figura 16) da máquina onde o agente está residente.

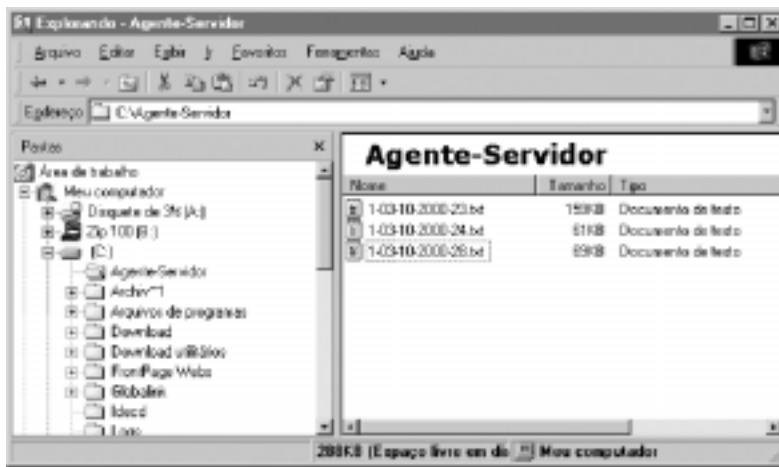


Figura 16 – Diretório Agente-Servidor com arquivos-resultado de consultas

O nome do arquivo gerado segue a seguinte lei de formação:

| Número de bytes | Significado                             |
|-----------------|---|
| 2               | Id do agente solicitante                |
| 8               | Data de solicitação                     |
| 2               | Seqüencial da solicitação para o agente |

A interface responsável pela solicitação, executando a função do Agente-Aplicativo, permanece monitorando esse diretório até identificar que sua consulta foi processada. A exibição do resultado utiliza a tela tl01601. O acesso ao arquivo, com o resultado da consulta, faz uso da tecnologia ADO.

A figura 17 exemplifica o resultado da consulta sobre quantos alunos estão matriculados na rede municipal de ensino. Nesta tela é possível solicitar a impressão em uma impressora local ou o armazenamento do SQL processado na base pessoal do usuário.



Figura 17 – Interface para exibição do resultado da consulta

## 6. CONSIDERAÇÕES FINAIS

No estudo realizado sobre a tecnologia de agentes, verificou-se que esta abordagem encontra-se ainda em estágio de desenvolvimento, apresentando divergências entre os pesquisadores, inclusive sobre a definição da unidade fundamental do paradigma: o agente.

Outros trabalhos analisados, como o de Flecha (1997), sobre mediadores, apoiaram o desenvolvimento de agentes de software para a democratização do acesso a bases de dados temáticas e distribuídas, como as do setor público.

As arquiteturas de agentes e aplicações analisadas indicam que a tecnologia de agentes beneficia o usuário, facilitando a execução de tarefas através do uso de agentes assistentes, bem como possibilita a otimização dos recursos computacionais, no que diz respeito à mobilidade e à execução de tarefas de forma corporativa.

Quanto aos agentes, é possível defini-los como software que utiliza a comunicação para negociar e coordenar a transferência de dados, agindo autonomamente sobre si próprios ou sobre o ambiente, buscando realizar os objetivos e tarefas para os quais foram projetados. Entre as propriedades essenciais para caracterizar a agência, verificou-se que a propriedade de autonomia é fundamental em um agente. Da mesma forma, constatou-se que as propriedades de comunicabilidade e reatividade também são consideradas como essenciais para a grande maioria dos pesquisadores.

A aplicação da metodologia UML permitiu verificar diversas vantagens em relação às outras metodologias não orientadas a objeto. Entre elas, observa-se que o esforço de modelagem é deslocado para as etapas anteriores de análise e projeto, produzindo um projeto limpo e de fácil implementação, pois tornou-se apenas uma tarefa de tradução, mais próxima da tarefa de programação.

Além disso, a ênfase na estruturação dos dados - e não na de funções - permite que as estruturas de dados sejam mais estáveis produzindo, assim, uma base mais robusta. O processo é bastante interativo, não sequencial, adaptando-se bem à prototipação e a um processo evolutivo de desenvolvimento. Os diagramas de análise (diagrama de classe, diagrama de seqüência e diagrama de caso de uso) aproximam a modelagem da realidade do sistema. Os diagramas de implementação (diagrama de componentes e diagrama de distribuição) oferecem suporte e clareza para a codificação do sistema.

No cenário tecnológico em que a PBH se encontra, a implementação de mecanismos como o proposto possibilita a ampliação do acesso às informações governamentais no âmbito do município de Belo Horizonte, em virtude de, pelo menos, dois fatores: a ampliação do escopo de consultas não estruturadas e a facilitação da interação com a base informacional.

A utilização de agentes de software, para viabilizar a recuperação de informações não estruturada, conforme aqui demonstrado, apresenta resultados satisfatórios, visto que os processos podem ser executados de forma interativa com o compartilhamento de recursos, independentemente da plataforma de hardware disponível no cliente.

## **KEYWORDS**

*Computational intelligence – Software agents – Legacy database*

## **ABSTRACT**

*The complex collection of themes treated by the Municipal City hall of Belo Horizonte (PBH) composes the thematic databases distributed in the servers in operation in the Municipal Net (RMI). This article presents an application developed using Unified Modeling Language (UML) and discusses the application of computacional intelligence applied to software agents as a tool or increasing the information potential of the available data in RMI and for not structured recovery by the users of the net and of the services of PBH.*

## **REFERÊNCIAS BIBLIOGRÁFICAS**

- ALHIR, S. A. *UML in a Nutshell*. USA: O'Reilly & Associates, 1998.
- AUER, K. Agents. 1995. [on line] <http://www.biplane.com.au/~kauer/project/>
- BELGRAVE, Marc. *The Unified Agent Architecture: A White Paper*. 1995. [on line] [http://www.ee.mcgill.ca/~belmarc/uaa\\_paper.html](http://www.ee.mcgill.ca/~belmarc/uaa_paper.html)
- BEMFICA, Juliana C. *Política, informação e democracia; O programa de democratização*

- de informações da Prefeitura de Belo Horizonte*. Belo Horizonte: Fundação João Pinheiro. (Dissertação, Mestrado em Administração Pública - Área de Concentração: Sistemas de Informação e Gestão), 1997.
- BOISSIER, O.; DEMAZEAU, Y. *A distributed artificial intelligence view on general purpose vision systems*. In: E. Werner and Y. Demazeau, editors, *Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World*. Kaiserslautern: Elsevier Science Publishers, 1991, p. 311-330
- BOND, A. H.; GASSER, L. *Readings in Distributed Artificial intelligence*. San Mateo: Morgan Kaufmann Publishers, 1988.
- COEN, M.; Kautz, H.; SELMAN, B.; KETCHPEL, S.; RAMMING, C. An experiment in the design of software agents. In *Proceedings of the 1994 AAAI Conference*. Seattle, WA, 1995, p. 438-443.
- CORREA FILHO, M. *A Arquitetura de Diálogos entre Agentes Cognitivos Distribuídos*. Unpublished Phd Thesis. UFRPJ/COPPE. RJ. 1994.
- D'AMICO, C.B et al. *Inteligência Artificial: uma abordagem de Agentes*. Porto Alegre, CPGCC- UFRGS, 1995.
- DEMAZEAU, Yves; MÜLLER, Jean P. Decentralized artificial intelligence. In: *Decentralized Artificial Intelligence - 1*. Org.: DEMAZEAU, Yves, MÜLLER Jean-Pierre. North-Holland, Elsevier Science Publishers, 1991.
- DEMAZEAU, Yves. Distributed Artificial Intelligence & Multi-Agent Systems. In: *X Simpósio Brasileiro de Inteligência Artificial*. Anais. Porto Alegre: II/UFRGS, 1993.
- DiSESSA, A. A. Notes on the Future of Programming: Breaking the Utility Barrier. In: *User-Centered Systems Design*, eds.D.A. Norman and S.W.Draper. Hillsdale, N.J.: Lawrence Erlbaum, 1986.
- ERICKSON, T. H. Design Agents as People Mattered. In: *Software Agents*. Ed. J.M. Bradshaw. Menlo Park, California: AAAI press, 1996.
- ERICKSON, T. H.; PENKER, M. *UML ToolKit*. NY: John Wiley & Sons, 1998.
- FLECHA, Mário L. O. *Processamento Semântico na Mediação de Consultas a Bases de Dados Públicas*. Belo Horizonte: Fundação João Pinheiro. (Dissertação, Mestrado em Administração Pública - Área de Concentração: Ciência da Computação & Administração Pública). 1997.
- FONER, Lenny. *What's an Agent, Anyway? A Social Case Study*. 1994. [on line] <http://foner.www.media.mit.edu/people/foner/Julia/Julia.html>
- FRANKLIN, Stan, GRAESSER, *Is it an Agent, or just a Program?: A Taxonomy*. 1995. [on line] <ftp://venera.isi.edu/sims/sss95/balabanovic.ps.Z>
- GENESERETH, M.R.; NILSSON, N. *Logical Foundations of Artificial Intelligence*. Palo Alto, CA: Morgan Kaufmann Publishers Inc, 1988.
- GENESERETH, M.R. An Agent-Based Framework for Interoperability. In *Software Agents*. Ed. J.M. Bradshaw. Menlo Park, California: AAAI press., 1997.
- GILBERT, D. et al. *IBM Intelligent Agent Strategy*. IBM Corporation, 1996.
- HAYES-ROTH, Barbara. *An Architecture for Adaptive Intelligent Systems*. Artificial

- Intelligence. Stanford University, Knowledge Systems Laboratory, 1995, p. 329-365.
- HUTCHINS, E. Direct Manipulation Interface. In: *User-Centered System Design*, eds. D.A. Norman and S.W. Draper, 87-124. Hillsdale, N.J.: Lawrence Erlbaum. 1986.
- LEMON, Bill; PYNADATH, David; TAYLOR, Glenn; WRAY, Bob. *Cognitive architectures*. 1994. [on line] <http://krusty.eecs.umich.edu/cogarch4/index.html>
- LOUÇÃ, H. C.; PINSON, S. Forming coalitions in task oriented multi-agents systems. In: *Proceedings of First Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems*. Xalapa, ME, 1996, p. 109-125.
- LUCK, Michael; D'INVERNO, Mark. A Formal Framework for Agency and Autonomy In: *ICMAS-95*, 1995.
- MAES, P. *Agents that Reduce Work and Information Overload*, Mit Media Laboratory, Cambridge, 1996. [on line] <http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html>
- MINSKY, M.; RIECKEN, D. A Conversation with Marvin Minsky about Agents. *Communications of the ACM*. 37(7): 23-29, 1994.
- NISSEN, Mark. *Intelligent Agents: A Technology and Business Application Analysis*, 1995. [on line] [http://www.mines.u\\_nancy.fr/~gueniffe/CoursEMN/I31/heilmann/heilmann.html](http://www.mines.u_nancy.fr/~gueniffe/CoursEMN/I31/heilmann/heilmann.html)
- OLIVEIRA, F. Inteligência Artificial Distribuída. In: *IV Escola Regional de Informática*. SBC, SC, 1996.
- PARAÍSO, Emerson Cabrera; KAESTNER, Celso Antônio Alves. Masc: *A Multi-Agent Approach for Control Applications*. DAINF - Federal Center of Education in Technology (CEFET/PR), 1996.
- ROSELER, Marina; HAWKINS, Donald T. Gent. *Agents: software servants for an electronic information world (and more!)*. OnLine. July, 1994.
- RUSSEL, S. e NORVIG, P. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall, inc. 932 p., 1995.
- SHNEIDERMAN, B. *Software psychology: human factors in computer and information systems*. Cambridge: Winthrop Publishers Inc, 1983.
- SHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer interaction*. Readin, Mass.: Addison-Wesley, 1987.
- SHOHAM, Y. *Agent-oriented programming*. *Artificial Intelligence*, 60(1): p.51-92, March. 1993.
- SICHMAN, J.; DEMAZEAU, Y.; BOISSIER, O. When can knowledge-based systems be called agents? In: *IX Simpósio Brasileiro de Inteligência Artificial*. SBC, Rio de Janeiro, 1992.
- SMITH, D.C.; IRBY, C.; KIMBALL, R.; VERPLANK, W; HARLSLEM, E. *Design the STAR Interface*. *Byte* 4, 1994, p. 242-282.
- STEFANINI, M. H.; TALISMAN: *Une architecture multi-agent pour l'analyse du français écrit*. Unpublished PhD thesis, Université Pierre Mendés-France. Grenoble, 1993.

WOOLDRIDGE, Michael; JENNINGS, Nicholas R. *Intelligent Agents: Theory and Practice*, 1994. [on line] <http://citeseer.nj.nec.com/context/18876/13736>

WOOLDRIDGE, Michael; JENNINGS, Nicholas R. *Software Agents*. IEE Review. January, 1996, p. 17-20.