



APLICANDO CONHECIMENTO MAPEADO POR MEIO DE DIAGRAMA DE ESTADOS EM AGENTES INTELIGENTES PARA REFINAMENTO DE BANCO DE DADOS GEORREFERENCIADO

Sandrerley Ramos Pires – sandrerley@hotmail.com

Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação

Endereço

CEP – Goiânia – Goiás

Sirlon Thiago Diniz Lacerda – thiago.dfc@gmail.com

Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação

Endereço

CEP – Goiânia – Goiás

Dulcineia G. F. Pires – dulcineiapires@hotmail.com

Instituto Federal de Goiás

Endereço

CEP – Anápolis – Goiás

Sirlon Diniz de Carvalho – sirlondiniz@gmail.com

Instituto Federal de Goiás

Endereço

CEP – Goiânia – Goiás

Resumo: *Este artigo descreve o desenvolvimento de um ambiente multiagente e um conjunto de interfaces homem/máquina. O objetivo é permitir a utilização desse ambiente por usuários menos experientes exigindo, portanto, menos esforço cognitivo na criação de agentes. As ações dos agentes são focadas, inicialmente, no refinamento do conteúdo de um banco de dados a partir de outro. O ambiente possui como estrutura agentes que extraem e atualizam informações, bem como agentes de inferência com capacidade de inteligência híbrida. A interface que mascara o modelo de regras SE...ENTÃO inseridas no ambiente, caracteriza-se como maior contribuição do ambiente proposto. É apresentado, ainda, um exemplo de solução para o refinamento de uma base de dados georreferenciada a fim de demonstrar as facilidades criadas para uso do ambiente.*

Palavras-chave: *Agentes Inteligentes, Georreferenciamento, Estruturação do Pensamento.*

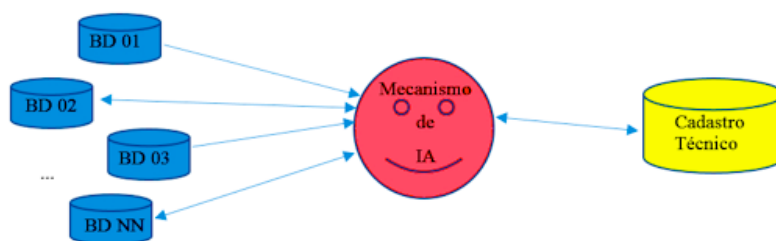
1. INTRODUÇÃO

Uma característica importante que o uso da inteligência artificial (IA) (RUSSEL, 2010) traz aos sistemas computacionais é a flexibilidade. Ela torna os sistemas capazes de assimilar modelos de comportamentos e, a partir desses modelos, resolver problemas variados. Este trabalho trata do uso de um ambiente computacional que se utiliza dos conhecimentos de IA e disponibiliza um ambiente multiagente (WOOLDRIDGE, 2009) com o objetivo de resolver problemas do cadastro técnico georreferenciado da rede de distribuição da Celg D (Celg Distribuição S.A.)

A motivação para o desenvolvimento do projeto foi saber que diversos sistemas da Celg D possuem informações passíveis de serem relacionadas de modo a gerar um novo conhecimento georreferenciado e, dessa forma, possibilitar o refinamento do cadastro técnico da empresa. Entretanto, os sistemas são muitos e cada nova possibilidade de relacionamento para a geração de informações georreferenciadas envolve sempre análises complexas, implementação de consultas SQL (NAVATHE, 2011), além de outras condições que necessitam ser tratadas para constituir a solução.

Esse cenário de soluções complexas e diferenciadas umas das outras remetem à necessidade do uso da IA para tornar o processo de construção de solução mais flexível. Observou-se que todas as soluções imaginadas têm uma estrutura comum que é a sequência de busca de informações em diversos banco de dados, seguido da análise dessas informações até a obtenção de uma conclusão, sendo que esta conclusão irá gerar uma atualização em um outro banco de dados com o objetivo de torna-lo mais exato. A Figura 1 abaixo mostra o fluxo genérico de soluções que refinam o cadastro técnico a partir da busca de conhecimentos nos outros sistemas da organização.

Figura 1 - Visão da arquitetura multiagente usada neste trabalho



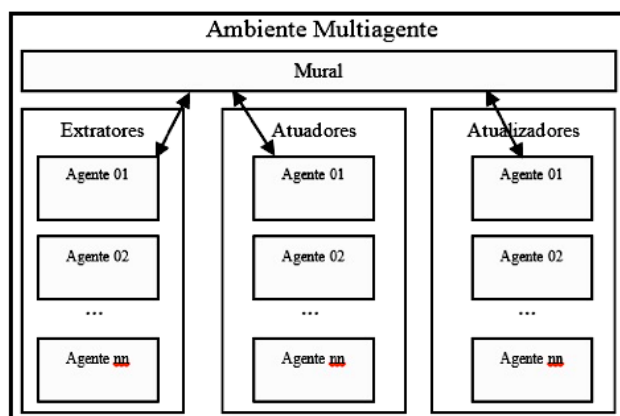
Um outro problema tratado neste trabalho é o da dificuldade do usuário final construir soluções em um ambiente genérico que oferece recursos de IA. Mesmo a criação de regras em um sistema especialista pode ser uma tarefa árdua para o usuário, dependendo da sua experiência com o assunto. Adiciona-se a essa dificuldade a elaboração de complexos SQLs para extração de informação e também para a atualização de informações no cadastro técnico. Assim, este trabalho propõe a criação de um ambiente multiagente baseada na abordagem simbolista (SCHALKOFF, 2009) e com recursos adicionais para o uso de outros modelos de raciocínios, dando-lhe a capacidade de resolver problemas através de mecanismos híbridos de inteligência artificial (RUSSEL, 2011). O objetivo é abstrair do usuário final parte da complexidade inerente à criação de soluções nos moldes descritos e ilustrado na Figura 1, levando a definição de uma interface que exija menos esforço cognitivo.

A interface aqui proposta se estrutura a partir de análise em problemas dentro da Celg D, onde soluções multiagente possam ser úteis para o refinamento de informações no cadastro

técnico da empresa. Notou-se que três atividades principais ocorrem em todas as soluções estudadas, a extração de informação, a análise e produção de novas informações a partir das informações extraídas e o uso desta nova informação para refinamento do banco de dados.

Assim, a aplicação proposta em um conjunto de agentes inteligentes (D'INVERNO, 2004) divididos em três camadas distintas, que deverá suprir os três papéis principais de toda solução multiagente analisada. É necessário também que esses agentes inteligentes possam comunicar-se dentro do ambiente. Desta forma, diante das necessidades do ambiente enumeradas, o trabalho propõe o ambiente ilustrado na Figura 2, cujos elementos principais são: (1) o Mural que é a área comum para a comunicação entre agentes; (2) Agentes Extratores que são entidades capazes de buscar informações em bancos de dados, expondo tais informações no mural; (3) Agentes Atuadores são as estruturas que realizarão as atividades de raciocínio; e (4) Agentes Atualizadores são as entidades que executam atualizações em bancos de dados.

Figura 2 - Estrutura do ambiente multiagente proposto



A arquitetura multiagente criada é similar à suportada pelo JADE (BELLIFEMINE, 2007), recurso utilizado nas aplicações com agentes inteligentes. Camadas de softwares foram criadas para facilitar a atuação do usuário, visto que uma das metas é permitir que usuários possam criar configurações multiagente para a solução de problemas específicos.

Este artigo detalha, na seção 3, a proposta de interface homem-máquina criada para permitir que o usuário possa construir soluções multiagente sem muito esforço cognitivo e também, na seção 4, os resultados obtidos com a construção de uma solução específica. Antes porém, na seção 2, é feito um breve relato sobre abordagens correlatas e a uma análise das dificuldades de sistemas baseados em regras. E, finalmente, na seção 5, as conclusões.

2. BASE TEÓRICA

2.1. Agentes Inteligentes

Dentro da Inteligência Artificial Distribuída, TORSUN (1995) classifica-a em Distributed Problem Solving (DPS), Multi-Agent Systems (MAS) e Parallel Artificial Intelligence (PAI). Os MAS ou Sistemas Multiagente, caracterizam-se pelo estudo e implementação de agentes que interagem de forma autônoma e que trabalham juntos para resolver um determinado problema. Os sistemas multiagente são constituídos por agentes

cognitivos, os quais são inteligentes e conhecem uma representação parcial de seu ambiente e dos outros agentes, podendo comunicar-se entre si, negociar uma informação ou um serviço e planejar ações. Os agentes são dotados de conhecimentos, competências, intenções e crenças, o que lhes permite coordenar suas ações visando à resolução de um problema.

Segundo JENNINGS (1998), algumas razões para o crescimento do interesse em pesquisas com sistemas multiagente são: (1) A capacidade de fornecer robustez e eficiência; (2) A capacidade de permitir interoperabilidade entre sistemas legados; e (3) A capacidade de resolver problemas cujo dado, especialidade ou controle é distribuído.

2.2. O Uso de Regras em Sistemas Especialistas

A expressão textual (Script) de uma regra da base de conhecimento de um agente inteligente pode ser feita da forma mostrada na Figura 3. O usuário enuncia uma condição através da estrutura “SE” (FILHO, 2013), que caso esteja satisfeita, isto é, o resultado da condição é VERDADEIRO, provocará a execução de uma ou mais ações enumeradas após a estrutura “ENTÃO”.

Essa sintaxe garante algumas características interessantes, como: (1) expressa de forma objetiva o conhecimento específico do agente; (2) A representação facilita a inclusão e a exclusão de regras da base de conhecimento, tornando possível o aprendizado; (3) Viabiliza a execução paralela de regras existentes na base de conhecimento; (4) É possível construir grandes bases de conhecimento. Por outro lado, esse modelo de expressão de conhecimento pode se tornar complicado para usuários menos experientes quando: (1) A base de conhecimento se torna muito extensa. (2) As condições para a execução das regras se tornam muito específicas, necessitando da criação de condições compostas.

Figura 3 - Regras estruturando uma base de conhecimento

```
SE <Condição Lógica 1>  
ENTÃO <Ação 1.1>  
      <Ação 1.2>  
SE <Condição Lógica 2>  
ENTÃO <Ação 2>  
SE <Condição Lógica 3>  
ENTÃO <Ação 3.1>  
      <Ação 3.2>
```

Enfim, a linguagem de script pode ser uma poderosa ferramenta para desenvolvedores experientes, mas pode se tornar um fator complicador para usuários menos experientes que queiram desenvolver soluções mais complexas.

2.3. O Uso de Gráficos para Representação de Estados

David Harel (HAREL, 1988) em seu trabalho On Visual Formalisms propõe um modelo de expressão de grafos chamado de Higraphs, os quais através de uma sintaxe formal e boa expressão visual, permite a representação da expressão de diversos outros diagramas como os modelos de entidade e relacionamento (E-R) e os diagramas de transição de estados, além de várias outras aplicações.

A sintaxe proposta é capaz de substituir tais modelos com a vantagem de manter todo o conteúdo semântico, além de ser visualmente mais fácil de ser analisado, desde que se entenda com clareza as regras sintáticas de estruturação do diagrama. Entre as aplicações propostas por Harel está a aplicação na representação do conhecimento, aproximando-se do que está sendo proposto neste artigo.

Outras abordagens utilizando-se de formalismo visual são encontradas para aplicação em engenharia de software como em (JACOB, 1983), onde a expressão gráfica modela interfaces complexas e até mesmo aplicado para a construção de configuração para agentes inteligentes como no trabalho de (SUBBURAJ e URBAN, 2011).

O modelo de estado proposto neste trabalho não se preocupa em encadear os estados porque entende que o comportamento do agente não é determinístico, dependendo do seu processo de inferência para decidir qual será o próximo estado. Além disso, considerando agentes com capacidade de aprendizado autônomo, as características determinísticas seriam ainda mais descaracterizadas.

3. O MODELO PROPOSTO

Visando dar ao usuário as condições necessárias para a criação de soluções no ambiente multiagente e tomando o cuidado para não retirar do usuário mais experiente as condições para a construção de soluções com maior complexidade, estruturou-se o ambiente multiagente apresentado neste trabalho. São detalhados nesta seção os quatro elementos principais do ambiente, bem como os recursos utilizados para viabilizar a sua aplicação por parte do usuário.

3.1. O Mural e seus Símbolos

O primeiro elemento do ambiente multiagente é o Mural. Ele é uma área comum no ambiente que pode ser acessado pelos diversos agentes existentes no ambiente multiagente, propiciando as condições necessárias para que se estabeleça a comunicação entre os diversos agentes do ambiente. Não há outra forma de troca de informação no ambiente que não seja via exposição e retirada de símbolos no Mural. O símbolo é a estrutura básica que é exposta no Mural, sendo que além do nome que o identifica, o símbolo pode conter também uma estrutura de dados ligada a ele.

Uma premissa adotada para a definição do ambiente multiagente é a de que um agente só se ativará no ambiente se, e somente se, for colocado no Mural um símbolo que o ative. Ao se cadastrar um agente no ambiente, deve-se relacionar o conjunto de símbolos que o ativará, sendo que é possível que um único símbolo ative diversos agentes e diversos símbolos ativem um único agente. Sempre que um símbolo for exposto no Mural, o controlador do ambiente se encarregará de ativar os agentes ligados ao símbolo em questão.

3.2. O Agente Extrator

O agente Extrator tem a responsabilidade de extrair informações de um ou mais bancos de dados da organização, exigindo de quem estiver construindo a solução, o conhecimento de quais são as informações necessárias para estruturar a solução, além de conhecer a estrutura de informação e o banco de dados onde esta informação se encontra.

Visando facilitar este processo, este trabalho disponibiliza duas formas distintas de configuração de um agente Extrator:

1. Query by Example: o uso da grade QBE (RAMAKRISHNAN e GEHRKE, 2002) trata-se de um recurso já difundido em várias ferramentas de acesso a banco de dados. O ambiente multiagente, por meio dos seus administradores, conhece a relação de bancos de dados e tabelas que podem ser acessados por ele.
2. SQL: para usuários mais experientes é possível a configuração do agente Extrator de informação por meio da construção de um comando SQL para a extração de dados.

Independentemente do tipo de configuração a ser realizada pelo usuário, o resultado será uma tabela obtida pela execução do SQL digitado ou construído a partir da grade QBE. Este resultado poderá ser gravado em uma tabela de saída, ou postado no Mural tupla a tupla ou, finalmente, postado no Mural uma lista com toda a tabela extraída. No momento da configuração do agente Extrator, o usuário indicará o nome do símbolo de saída que será exposto no Mural com o resultado da query.

3.3. O Agente Atualizador

O agente Atualizador tem a responsabilidade de efetivar o resultado de uma solução. Agentes Atualizadores e Extratores atuam em conjunto até chegarem a uma conclusão, chamando o agente Atualizador para atualizar o banco de dados destino ou, opcionalmente, gerar um relatório com a conclusão obtida pela solução. Observe que o usuário que estiver construindo a solução deve conhecer a estrutura de informação e o banco de dados onde será feita a atualização com os resultados obtidos na solução. Visando facilitar este processo de configuração de um agente atuador, este trabalho disponibiliza duas formas distintas de configuração de um agente Atualizador:

1. Atualização em banco de dados: trata-se de um processo parecido com o da grade QBE para atualização, onde o usuário escolhe a tabela que será atualizada, em seguida a relação de campos desta tabela e os dados que serão gravados nos campos escolhidos e, finalmente, as informações de definição de quais tuplas serão atualizadas (semelhante à cláusula WHERE da declaração UPDATE do SQL).
2. Relatório: opcionalmente, o usuário poderá ao invés de atualizar um banco de dados, gerar um relatório para que as conclusões da solução multiagente possam atuar como um sistema de apoio à decisão, seguindo-se uma atualização do BD de forma manual.

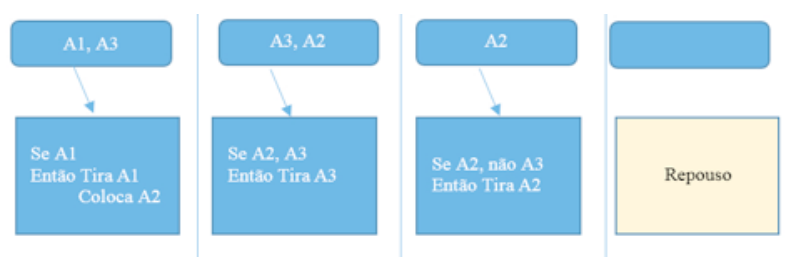
3.4. O Agente Atuador

Um dos aspectos mais difíceis no momento de compor uma solução multiagente é a criação de uma base de conhecimento. Para se criar essa base é necessário entender qual será o correto encadeamento de estados durante o processo de raciocínio até se atingir uma conclusão. Os estados são definidos pela configuração dos símbolos de ativação para o agente inteligente em questão.

Define-se estado de ativação, neste contexto, como sendo a configuração de símbolos no Mural que irão provocar o agente a executar alguma ação. Chama-se de ciclo de inferência o período em que a máquina de inferência do agente está atuando. Assim, o comportamento do agente inteligente se caracteriza pelas transições de estados que ele faz em busca do repouso (nenhum estado de ativação ocorrendo), sempre que os outros agentes provocam o aparecimento de um estado de ativação. A Figura 4 ilustra a ocorrência de um ciclo de inferência composto por três passos, onde em cada passo as ações de alguma regra do agente realizam uma modificação no Mural até o momento em que os símbolos do Mural não ativam nenhuma regra do agente, levando-o ao repouso e ao fim do ciclo de inferência.

Entender quais são os estados e o sequenciamento dos mesmos para formar os ciclos de inferência é o caminho inicial para estruturar a base de conhecimento. Os estados se estruturam a partir dos símbolos e, na arquitetura proposta, esses símbolos se encontram no ambiente comum (o Mural) que é observável por todos os agentes. Pensar para o agente é transitar entre estados, inicialmente mais simples e, à medida que ele avança na busca de uma solução, os estados se tornam mais complexos.

Figura 4 - Ciclo de inferência composto por três passos

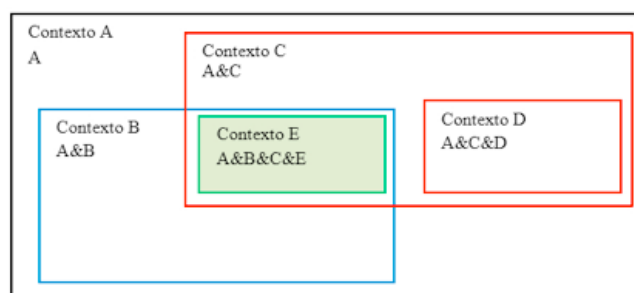


A estruturação de contextos

Toda regra se inicia em uma condição que geralmente é uma condição simples. À medida em que o processo de inferência caminha, as condições se tornam cada vez mais complexas. Durante um ciclo de inferência parte-se de um contexto mais geral o qual se aprofunda em complexidade, mas quase sempre mantendo as generalidades dos contextos anteriores, isto é os contextos mais superficiais. A Figura 5 ilustra essa condição, onde uma região de contexto inicial pode ser a mais geral e simples, a região “A”.

Observe que as condições compostas e o “afunilamento” em um processo de inferência podem ser facilmente expressos em termos da expressão gráfica de regiões que por sua vez contém sub-regiões. O raciocínio iniciou-se em “A”, algumas ações geraram “B” no contexto, agora as regras do contexto A&B serão ativadas e assim por diante.

Figura 5 - Mostra a representação gráfica de contextos



O quadro do “Contexto E”, em destaque na figura, representa uma condição composta (A&B&C&E) de forma bastante natural. Anexar a essa região um conjunto de ações que seriam executadas caso o fluxo de raciocínio chegasse a ela, pode ser algo bem mais simples que a construção, em termos de scripts, de longas condições compostas. Através da visualização gráfica é possível perceber contextos que com o encadeamento de regras não seriam tão simples de serem percebidos como válidos e úteis para certas condições.

A estruturação de contextos permite a redução significativa da complexidade de criação das regras, pois permite uma visão panorâmica de um conjunto finito de estados em que o agente pode estar. Em grande parte dos processos de inferência, esse contexto visualmente coberto pode ser suficiente para a estruturação de todo o conjunto de regras.

Ressalta-se que o nome do contexto utilizados na Figura 5, “A”, “B” e outros, na verdade são os nomes dos símbolos, e que por alguma ação são disponibilizados no ambiente multiagente. Desta forma, o quadro de contextos descritos acima, o qual se encontra mapeado dentro da base de conhecimento do agente, torna-se ativo a partir dos símbolos presentes no Mural do ambiente

Regras a partir de contextos

A implementação de uma regra é facilitada a partir do diagrama de contextos descritos anteriormente. Mas, novamente, visando a construção da interface gráfica, continua a busca de um processo gráfico e intuitivo que consiga inserir um conjunto de ações ligadas a um determinado contexto para que se complete o processo de criação da base de conhecimento. Desta forma, propõe-se que criar a base de conhecimento pelos seguintes passos:

1. Identificar os símbolos envolvidos na estruturação de uma determinada solução;
2. Estabelecer os contextos, de forma gráfica, conforme a Figura 4. Esses contextos podem envolver regiões que se ativam com um determinado símbolo e outras regiões que se ativam com a combinação de dois ou mais símbolos;
3. Estabelecer o conjunto de ações que o agente deve executar em cada contexto. Essas ações poderão levar o agente de um contexto para outro. Alternativamente, as ações podem ativar outro agente, que por sua vez criará símbolos que resultarão na estruturação de um contexto mais complexo para o primeiro agente e assim por diante.

As ações podem envolver operações diversas, mas sempre sensibilizando o ambiente multiagente de alguma forma, onde essa sensibilização pode provocar a ativação de novos contextos, mantendo o processo de paralelismo necessário para a máquina de inferência.

A complexidade do script com regras no formato SE..ENTÃO, quando se trata de usuário final, é um fator impeditivo para o entendimento do conteúdo do script. Se a intenção é que o próprio usuário final consiga construir a base de conhecimento, pode-se perceber que as dificuldades serão maiores ainda que simplesmente entender o que está se passando.

4. RESULTADOS EM UMA APLICAÇÃO EXEMPLO

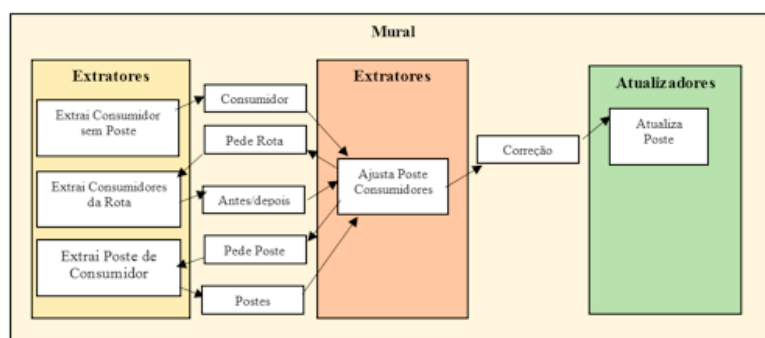
A ideia estruturada e detalhadas nas seções anteriores viabiliza a construção de uma aplicação capaz de criar a base de dados para agentes inteligentes. Neste artigo são apresentados os resultados da implementação de uma aplicação criada para a Celg D.

O problema tratado aqui é a respeito de consumidores que não possuem a localização geográfica de seu poste registrada no sistema. Entretanto, sabe-se que no sistema comercial, através de uma tabela de rota de leitura de consumidores, existe a possibilidade deste consumidor estar ligado no mesmo poste que os consumidores anterior e posterior a ele, na rota de leitura. A solução criada neste exemplo irá fazer as seguintes tarefas:

1. Relacionar todos os consumidores sem posicionamento geográfico do poste. Expor no Mural cada um desses consumidores, um a um.
2. Para cada consumidor exposto determinar o consumidor anterior e posterior a partir da tabela de rota do sistema comercial;

3. Encontrar os dados do poste a que os consumidores anterior e posterior estão ligados;
 4. Avaliar se os três consumidores estão no mesmo poste, sendo que caso estejam, deve-se atualizar o posicionamento do consumidor avaliado.
- Estima-se que esta solução possa reduzir o problema do cadastro técnico em 10% das ocorrências de falhas de posicionamento de consumidor.

Figura 6 - Ilustração da solução exemplo no ambiente multiagente



A Figura 6 mostra a solução projetada, com os agentes que a comporão no ambiente multiagente. Propõe-se criar cinco agentes, sendo três extratores de informação, um para atuar no processo de inteligência e um agente atualizador. Observe que o agente atuador “Ajasta Poste Consumidores” é o agente responsável pela coordenação do processo, sendo ele que solicita a obtenção dos dados da tabela de rota de leitura e depois da obtenção de dados do poste, ativando os agentes extratores correspondentes por meio da escrita dos símbolos correspondentes no Mural.

4.1. A criação dos agentes

Após definir a relação de agentes que comporão a solução, as responsabilidades de cada um deles e os símbolos que sensibilizarão os agentes projetados, passa-se ao processo de criação desses agentes no ambiente multiagente.

Agentes Extratores

Foram criados no ambiente três agentes extratores sendo eles: (1) “Extrai consumidor sem Poste” que é o agente responsável por buscar no banco de dados do cadastro técnico a relação de consumidores que não possuem informações de poste em seu cadastro e expô-los no Mural, um a um; (2) “Extrai Consumidores da Rota” é o agente responsável por extrair do sistema comercial quais são os consumidores vizinhos do consumidor que está em análise; (3) “Extrai Poste de Consumidor” agente responsável por recuperar o número do poste a que um determinado consumidor está ligado.

Agente Atualizador

A solução apresentada neste trabalho propõe a criação de apenas um único agente atualizador. A responsabilidade desse agente será a vinculação de um consumidor a um determinado poste. O processo de criação do agente atualizador é semelhante ao agente extrator, mas neste caso a grade QBE irá estruturar um comando UPDATE para realizar a atualização de dados. Os dados necessários para efetivar a atualização dos dados no cadastro

técnicos estão postados no símbolo que o agente atuador, coordenador do processo, expôs para chamar o agente atualizador.

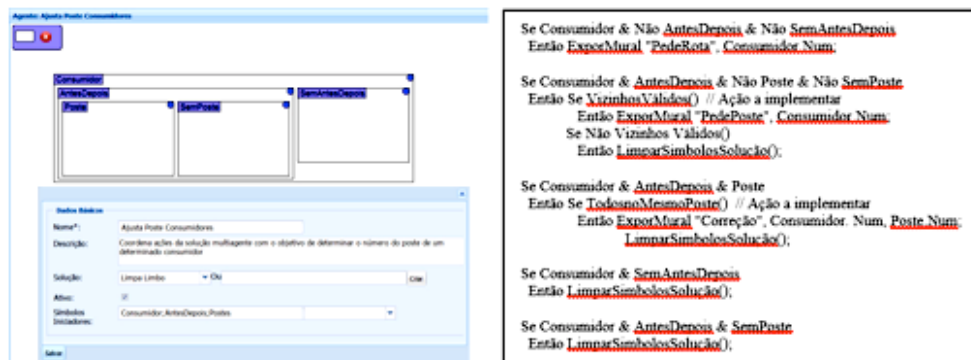
Agente Atuador

Normalmente, uma solução poderá se estruturada com apenas um agente Atuador, embora, a critério do desenvolvedor, possa ser utilizado mais que um agente. Neste exemplo, um único agente foi criado cujo nome é “Ajusta Poste Consumidor”, mas no diagrama de estados importantes, toda a complexidade inerente à solução pôde ser exposta em um único diagrama de forma relativamente simples.

Conforme é demonstrado na Figura 7 (a), para a criação do diagrama de estado no agente Atuador basta selecionar e arrastar a figura do novo estado (canto superior esquerdo da tela) para a área do diagrama, encadeando os estados conforme os ciclos de inferência projetados para o agente lidar. Após a inserção do estado, será apresentada a tela de contextualização do estado inserido. A Figura 7 (b) mostra uma porção de código que será gerado, automaticamente pelo ambiente, para a formação da base de conhecimento do agente Atuador a partir das informações contidas no diagrama de estados encadeados.

Para a contextualização da ativação do estado do agente é necessária a definição de quais serão os símbolos que iniciarão a atividade do agente. Esses símbolos serão produzidos pelos agentes do ambiente no formato já descrito anteriormente neste trabalho. Cada estado importante criado no diagrama de estado terá como nome um símbolo ativador do agente Atuador, sendo que no momento em que tais símbolos estejam presentes no Mural, as ações relativas ao estado em questão serão ativadas.

Figura 7 - (a) Tela de configuração do agente atuador. (b) porção do código gerado a partir do diagrama de estados.



O usuário deve então definir as ações que o agente deverá executar quando um, ou vários, dos estados definidos para o agente forem ativados. Para a definição das ações que o agente deve tomar, o usuário deve clicar no círculo no canto superior dos estados, conforme está demonstrado na Figura 7 (a). É possível, então, descrever a relação de ações que serão executadas pelo agente, caso o estado se ative. Os códigos são escritos em JAVA (DEITEL e DEITEL, 2011) e devem estar sintaticamente corretos para que possam ser executados. O ambiente efetua a validação dos códigos antes de salvar os dados do agente.

Experimentos iniciais realizados com usuários, sem conhecimento prévio de construções de regras lógicas em um sistema especialista, demonstraram que eles foram



capazes de entender com facilidade e também construir as estruturas de estados encadeados, os quais, quando posteriormente transformados em estruturas lógicas SE...ENTÃO, já não eram tão naturais de serem entendidas pelos próprios construtores da estrutura.

Após a configuração da solução, a mesma foi aplicada a uma base de dados que possui cerca de 17.000 consumidores sem a informação de poste, sendo que foram observadas as correções de 1.553 casos no universo analisado, ficando próximo da expectativa de 10% de solução para o problema tratado neste exemplo. Ressalta-se que à medida que novas possibilidades de correção de problemas forem idealizadas, o ambiente multiagente poderá suprir os recursos necessários para a construção de novas soluções, obtendo a médio prazo altos índices de correção para o cadastro técnico da Celg D.

5. CONCLUSÕES

Este trabalho mostrou uma proposta de estruturação de um ambiente multiagente capaz de solucionar problemas complexos, mas, no entanto, exigindo um mínimo de esforço cognitivo do usuário para a estruturação das soluções multiagente. Adicionalmente, apresentou os resultados da implementação de uma de interface gráfica, baseada na abordagem proposta, para a solução de um problema de refinamento do cadastro técnico da Celg D.

Embora seja complexo dizer: “através do entendimento e mapeamento prévios dos estados importantes de um contexto, os quais provocam a ativação de um ou mais agentes inteligentes, e os leva a se comportar de acordo com o estado em que se encontra”, este trabalho demonstrou, experimentalmente, que esse modelo de retratação de conhecimento é mais simples de ser operado por usuários pouco experientes que o modelo de conhecimento expresso com regras lógicas, essas utilizadas nos sistemas especialistas tradicionais.

A elucidação teórica da proposta, a apresentação de um conjunto de telas que compõem a interface gráfica adequada e, finalmente, a apresentação de um exemplo prático, demonstraram que os objetivos deste trabalho foram plenamente alcançados.

6. REFERÊNCIAS / CITAÇÕES

BELLIFEMINE, Fabio; CAIRE, Giovanni; GREENWOOD, Dominic. Developing Multi-Agent Systems with JADE, 1º ed., Wiley, 2007.

DEITEL, Paul; DEITAL, Harvey. Java How To Program, 9ª ed., DEITEL, 2011.

D'INVERNO, Mark; LUCK, Michael. Understanding Agent Systems, 2ª ed., Springer, 2004.

DIAS, M.; CARVALHO, J. A Visualização da Informação e a sua contribuição para a Ciência da Informação, Pesquisa Brasileira em Ciência da Informação e Biblioteconomia, vol.3, nº 8, 2007.

FILHO, Plinio Barbieri; JÚNIOR, Annibal Hetem. Fundamentos de Informática - Lógica Para Computação, 1ª ed., LTC, 2013.



HAREL, D. On Visual Formalisms, Communication of the ACM, vol.31, nº 5, p. 514-530, 1988.

JACOB, R. J. K. Using formal specifications in the design of a human-computer interface, Communications of the ACM, Vol.26, nº 4, p.259-264, 1983.

JENNINGS, N. R.; SYCARA, K.; WOOLDRIDGE, M. A Roadmap of Agent Research and Development, Kluwer Academic Publishers, 1998.

NAVATHE, Shamkant B.; ELMASRI, Ramez. Sistemas de Banco de Dados, 6ª ed., Pearson Education, São Paulo, 2011.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Database Management Systems, 3ª Edition, McGraw-Hill, 2002.

RUSSEL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach, 3ª ed., Pearson Education Limited, United States of America, 2010.

SCHALKOFF, R. J. Intelligent Systems: Principles, Paradigms and Pragmatics, 1ª ed., Jones & Bartlett Publishers, 2009.

SUBBURAJ, V. H.; URBAN, J. Intelligent agent software development using AUML and the Descartes specification language, Proceedings of 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, p. 297-305, 2011.

TORSUN, I.S. Foundations of Intelligent Knowledge-based systems, London: Academic Press, 1995.

WOOLDRIDGE, Michael. An Introduction to Multiagent Systems, 2ª ed., Wiley, 2009.

APPLYING MAPPED KNOWLEDGE THROUGH STATES DIAGRAMS IN INTELLIGENT AGENTS FOR REFINEMENT OF GEOREFERENCED DATABASE

Abstract: *This article describes the development of a multi-agent environment and a set of man / machine interfaces. The goal is to allow the use of this environment for less experienced users and therefore requires less cognitive effort in creating agents. The actions of agents are focused initially on refining the contents of a database from another. The environment has in its structure agents that extract and update information as well as agents with the capacity of hybrid inference intelligence. The interface that masks the model of IF ... THEN rules embedded in the environment is characterized as a major contribution of the environment. It is also presented an example of a solution to the refinement of a georeferenced database to demonstrate the easies created for use of the environment.*

Key-words: *Georeferencing, Intelligent Agents, Thought Structuring*