

UNIVERSIDADE FEDERAL DE GOIÁS  
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE  
COMPUTAÇÃO

Sirlon Thiago Diniz Lacerda

**PROPOSTA DE ARQUITETURA DE UM AMBIENTE MULTIAGENTE  
COM APLICAÇÃO DE FORMALISMO VISUAL PARA  
REPRESENTAÇÃO DE CONHECIMENTO**

GOIÂNIA

2015

UNIVERSIDADE FEDERAL DE GOIÁS

Sirlon Thiago Diniz Lacerda

**PROPOSTA DE ARQUITETURA DE UM AMBIENTE MULTIAGENTE  
COM APLICAÇÃO DE FORMALISMO VISUAL PARA  
REPRESENTAÇÃO DE CONHECIMENTO**

Dissertação apresentada ao Programa de Pós-Graduação da Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica e de Computação.

Área de concentração: Engenharia de Computação

Orientador: Gélson da Cruz Júnior

GOIÂNIA

2015

Ficha catalográfica elaborada automaticamente  
com os dados fornecidos pelo(a) autor(a), sob orientação do Sibi/UFG.

Lacerda, Sirlon Thiago Diniz

Proposta de Arquitetura de um Ambiente Multiagente com Aplicação  
de Formalismo Visual para Representação de Conhecimento  
[manuscrito] / Sirlon Thiago Diniz Lacerda. - 2015.

CXIII, 113 f.: il.

Orientador: Prof. Dr. Gélson da Cruz Júnior; co-orientador Dr.  
Sandrerley Ramos Pires.

Dissertação (Mestrado) - Universidade Federal de Goiás, Escola de  
Engenharia Elétrica (EEEEC) , Programa de Pós-Graduação em  
Engenharia Elétrica e de Computação, Goiânia, 2015.

Bibliografia. Anexos.

Inclui fotografias, gráfico, tabelas, lista de figuras, lista de tabelas.

1. Ambiente multiagente. 2. Estruturação de pensamento. 3.  
Formalismo visual. 4. Georreferenciamento. I. Júnior, Gélson da  
Cruz, orient. II. Pires, Sandrerley Ramos, co-orient. III. Título.

*À minha família e amigos,  
pelo apoio incondicional.  
Em memória de Kênia Gonçalves Pires,  
a primeira a acreditar nessa ideia e também responsável por esse trabalho,  
com quem tenho a humildade de dividir esse momento.*

*Este trabalho teve o suporte financeiro da Celg Distribuição que foi executado em parceria com a Fundação de Apoio à Pesquisa (FUNAPE) e a Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás por meio do Projeto de Pesquisa e Desenvolvimento intitulado "Desenvolvimento de Ambiente Multiagente e de cultura organizacional para a construção de agentes inteligentes. Um estudo de caso no refinamento da qualidade do cadastro georreferenciado da Celg D".*

## **AGRADECIMENTOS**

Ao meu orientador, Gélson da Cruz Júnior, pela orientação e motivação.

Ao Prof. Sandrerley Ramos Pires, pela paciência e conhecimento compartilhados ao longo deste projeto.

Ao meu pai, Sirlon Diniz de Carvalho, pela lucidez e companheirismo em todas as horas.

## RESUMO

Cada vez mais tem-se utilizado o paradigma multiagente na resolução de problemas complexos. Sua flexibilidade permite a utilização em diversos setores, entre eles o setor elétrico. Dentro de uma distribuidora de energia elétrica, uma das consequências da expansão da malha de distribuição é a grande quantidade de atualizações nos dados técnicos georreferenciados. Esse processo contínuo de atualização faz com que a qualidade das informações seja prejudicada, de forma que haja necessidade de interferência humana na correção de dados imprecisos ou incorretos. Como solução, este trabalho tem como objetivo propor um ambiente multiagente com uma interface gráfica que possa ser utilizada por usuários menos experientes exigindo, portanto, menor 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. São apresentadas, ainda, três soluções para o refinamento de uma base de dados georreferenciada a fim de demonstrar as facilidades criadas para uso do ambiente.

## **ABSTRACT**

The multi-agent paradigm has been increasingly used to solve complex problems. Its flexibility allows for use in various industries, including the energy sector. The result of the expansion of the electricity distribution network is a lot of updates in the georeferenced technical data of the electricity distributors. This continuous process of updating makes the quality of information is impaired, so that there is need for human intervention to correct inaccurate or incorrect data. As a solution, this paper proposes a multi-agent environment with a graphical interface that can be used by less experienced users and therefore requires less cognitive effort in creating agents. The agents' actions are focused initially on refining contents of a database from another. The environment has in its structure agents that extract and update information as well as inference agents with hybrid intelligence capacity. The interface that represents the IF..THEN rules is the largest contribution of the proposed system. Finally, three solutions are presented for the refinement of a geo-referenced database to demonstrate the facilities created for environmental use.

## LISTA DE FIGURAS

Figura 2.1 - Estrutura do neurônio humano. Adaptado de Levitan e Kaczmarek (2001) .....	8
Figura 2.2 - Estrutura proposta por McCulloch e Pitts. Fonte: Haykin (2009) .....	9
Figura 2.3 - Elementos básicos de um sistema especialista .....	12
Figura 2.4 - Regras estruturando uma base de conhecimento .....	15
Figura 2.5 - Sistemas especialistas e baseados em conhecimento. Adaptado de Waterman (1986).....	17
Figura 2.6 - Statechart do Citizen Quartz Multi-Alarm III. Fonte: Harel (1987).....	23
Figura 2.7 - Diagrama de Estados para o comando "Login". Fonte: Jacob (1983).....	24
Figura 3.1 - Visão da arquitetura multiagente usada neste trabalho.....	25
Figura 3.2 - Estrutura do ambiente multiagente proposto .....	26
Figura 3.3 - Ciclo de inferência composto por três passos.....	29
Figura 3.4 - Representação gráfica dos contextos .....	30
Figura 3.5 - Protótipo da tela de criação do agente atuador .....	32
Figura 3.6 - Porção do código gerado a partir do diagrama de estados da Figura 3.4 .....	33
Figura 3.7 - Estrutura do banco de dados exemplo .....	35
Figura 3.8 - Representação do agente extrator exemplo .....	35
Figura 3.9 - Representação do agente atuador exemplo.....	36
Figura 3.10 - Representação do agente atualizador exemplo .....	36
Figura 3.11 - Arquitetura MANNFIS. Adaptado de Wang, Abdulla e Salcic (2007).....	38
Figura 3.12 - Arquitetura híbrida para data mining. Adaptado de Fowler e Hammell (2011).39	
Figura 3.13 - Arquitetura híbrida. Adaptado de Ghosh, Nafalski e Tweedale (2013) .....	40
Figura 4.1 - Criação de um agente extrator com origem de banco de dados.....	42
Figura 4.2 - Definição de condições para extração .....	42
Figura 4.3 - Funções de tratamento e formatação dos dados extraídos.....	43
Figura 4.4 - Representação de parâmetros para um agente atualizador .....	43
Figura 4.5 - Criação e visualização dos contextos em um agente atuador .....	44
Figura 4.6 - A interface do Mural.....	45
Figura 4.7 - Detalhes de um agente específico .....	45
Figura 4.8 - Agentes utilizados na solução para CEP único.....	47
Figura 4.9 - Agentes utilizados na solução de correção dos trechos de rede .....	48
Figura 4.10 - Gráfico representando a mudança após a correção dos trechos de rede.....	49
Figura 4.11 - Agentes utilizados na solução dos consumidores sem localização geográfica...51	
Figura 4.12 - Agente atuador AT003_1_Trata_Consumidor_Limbo.....	52
Figura 4.13 - Agente extrator EX003_2_Busca_Rota.....	53
Figura 4.14 - Agente atualizador AZ003_1_Atualiza_Poste.....	54

## **LISTA DE TABELAS**

Tabela 3.1 - Habitantes da cidade Villa.....	34
--	----

# SUMÁRIO

1	INTRODUÇÃO .....	2
1.1	JUSTIFICATIVA .....	4
1.2	OBJETIVOS .....	5
1.2.1	GERAIS .....	5
1.2.2	ESPECÍFICOS .....	5
1.3	METODOLOGIA .....	5
1.4	ESTRUTURA DA DISSERTAÇÃO .....	6
2	REFERENCIAL TEÓRICO .....	7
2.1	INTELIGÊNCIA ARTIFICIAL .....	7
2.2	INTELIGÊNCIA ARTIFICIAL CONEXIONISTA .....	7
2.3	INTELIGÊNCIA ARTIFICIAL SIMBÓLICA E SISTEMAS ESPECIALISTAS .....	10
2.3.1	SISTEMAS ESPECIALISTAS .....	11
2.3.2	COMPOSIÇÃO DOS SISTEMAS ESPECIALISTAS .....	12
2.3.3	USO DE REGRAS EM SISTEMAS ESPECIALISTAS .....	15
2.3.4	SISTEMAS ESPECIALISTAS E AMBIENTES MULTIAGENTES .....	16
2.4	OUTROS MODELOS COMPUTACIONAIS INTELIGENTES .....	18
2.5	NATUREZA DOS AMBIENTES .....	18
2.6	SISTEMAS MULTIAGENTES .....	19
2.7	AMBIENTE MULTIAGENTE .....	21
2.8	FORMALISMO VISUAL E O USO DE GRÁFICOS PARA REPRESENTAÇÃO DE ESTADOS .....	22
3	ABORDAGEM PROPOSTA .....	25
3.1	MODELO PROPOSTO .....	25
3.2	MURAL E SÍMBOLOS .....	27
3.3	AGENTE EXTRATOR .....	28
3.4	AGENTE ATUALIZADOR .....	28
3.5	AGENTE ATUADOR .....	29
3.5.1	ESTRUTURAÇÃO DE CONTEXTOS .....	30
3.5.2	PROTÓTIPO DO AGENTE ATUADOR .....	31
3.6	EXEMPLO UTILIZANDO A ABORDAGEM GRÁFICA .....	34
3.7	INTELIGÊNCIA HÍBRIDA .....	38
4	RESULTADOS OBTIDOS .....	41
4.1	INTERFACE DO AMBIENTE .....	41
4.1.1	PREPARAÇÃO PARA CRIAÇÃO DE SOLUÇÕES .....	41
4.1.2	CRIAÇÃO DAS SOLUÇÕES E SEUS AGENTES .....	41
4.1.3	REPRESENTAÇÃO GRÁFICA DO MURAL .....	45
4.2	APLICABILIDADE DO AMBIENTE .....	46
4.3	PUBLICAÇÕES .....	54
5	CONCLUSÕES .....	56
6	REFERÊNCIAS .....	58
	ANEXOS .....	62

# 1 INTRODUÇÃO

O uso da Inteligência Artificial (IA) está diretamente ligado a cenários de soluções complexas e diferenciadas uma das outras, sendo utilizada, muitas vezes, para tornar o processo de construção dessas soluções mais padronizado e flexível. Sendo assim, uma característica importante que o uso da IA traz aos sistemas computacionais é a flexibilidade (RUSSEL; NORVIG, 2010). Ela torna os sistemas capazes de assimilar modelos de comportamentos e, a partir desses modelos, resolver problemas variados. Entretanto, toda essa flexibilidade exige que o desenvolvedor possua conhecimentos aprofundados e, por vezes, domine assuntos complexos como redes neurais, algoritmos genéticos e lógica *Fuzzy*.

A Celg Distribuição S.A. (Celg D) é responsável pela comercialização de energia elétrica em cerca de 98,7% do estado de Goiás (CELG D, 2015). Dentre outras atividades, a Celg D é incumbida também pelo planejamento, instalação, manutenção e operação das redes elétricas. Ela também oferece diversos serviços informatizados como, por exemplo, retirada de segunda via de conta e ressarcimento para danos elétricos ou diversos. Serviços de podas de árvores e manutenções na rede elétrica são, na maioria das vezes, solicitados e exercidos sob demanda. Para melhor desenvolver essas atividades, a qualidade dos dados georreferenciados existentes em seus sistemas é de fundamental importância, a fim de que o que se tem registrado no banco de dados reflita a realidade de campo.

Durante o período de 2004 a 2006, a Celg D desenvolveu um banco de dados georreferenciado denominado Cadastro Técnico. Atualmente, a atualização das redes de distribuição no Cadastro Técnico é feita de forma automática via integração com os aplicativos da área comercial e da área de projetos, obras e manutenção, além de algumas funções de cadastro manual disponível aos diversos usuários do sistema.

O correto registro georreferenciado das estruturas que compõem a malha de distribuição de energia é um fator fundamental para a correta operação do sistema, para o planejamento da manutenção e crescimento desta malha, bem como para o atendimento à Agência Nacional de Energia Elétrica (ANEEL), por meio dos Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST), o qual determina o envio anual da base de dados geográfica da distribuidora (PRODIST, 2011). O não envio dessas informações ou o envio de informações incorretas pode acarretar advertências, multas, embargo de obras e, no pior dos

casos, caducidade da concessão ou permissão (ANEEL, 2004). Além dessas consequências, a imprecisão nas metragens de redes elétricas pode gerar ônus para a empresa, visto que, isso pode prejudicar o planejamento e a manutenção das mesmas. Unidades consumidoras e equipamentos incorretamente geolocalizados podem, ainda, gerar problemas para o setor de manutenção, pois, os mesmos podem ter problemas para encontrar essas localidades e, assim, gerando mais custo à empresa, seja quanto à alocação prolongada do colaborador, ou mesmo do veículo.

A alimentação dos sistemas da Celg D é feita por colaboradores de todas as partes do estado e, a partir dessas informações, é feita a integração com o Cadastro Técnico. Entretanto, por diversas razões, várias dessas informações não são preenchidas de forma ideal e podem estar incorretas. Em casos extremos, as informações sequer são preenchidas. Desta forma, os dados recebidos pelo Cadastro Técnico, sejam eles incorretos ou inexistentes, são utilizados por sistemas de planejamento e manutenção da rede elétrica. O setor de geoprocessamento é o único responsável por garantir, diariamente, a qualidade dos dados georreferenciados. Por ser um departamento pequeno, esse trabalho demanda esforço e cuidado constantes de cada colaborador.

No intuito de ajustar o Cadastro Técnico, percebeu-se que a solução para um determinado problema é diferente das soluções para os outros, dificultando a construção de soluções computacionais que atendam a todos os tipos de problemas identificados. No contexto deste trabalho realizado na Celg D, observou-se que todas as soluções para os problemas analisados têm uma estrutura comum que é a busca de informações nos diversos banco de dados dos diferentes sistemas da organização, seguido da análise dessas informações até a obtenção de uma conclusão e, finalmente, realizar uma atualização no banco de dados do Cadastro Técnico com o objetivo de torná-lo mais exato. Identificou-se nas soluções três responsabilidades principais: a extração de informação, o tratamento dessas e a manifestação de um resultado (uma atualização em banco de dados ou um relatório).

Para apresentar o conjunto de ações integradas que visam automatizar o processo de refinamento da qualidade do Cadastro Técnico da Celg D, este trabalho apresenta a proposta de utilização e desenvolvimento de um ambiente multiagente dinâmico para atualização dos dados desse cadastro. Com esse intuito são apresentados agentes especializados em extrair informações de diversas fontes diferentes, agentes inteligentes capazes de realizar inferências a partir dos dados trazidos pelos extratores e, finalmente, agentes capazes de atualizar

informações no banco de dados do Cadastro Técnico ou, simplesmente, gerar relatórios com advertências que apontam prováveis irregularidades do Cadastro Técnico.

Como parte do ambiente multiagente, é proposta uma interface gráfica inspirada pelo modelo de formalismo visual implementado por Harel (1987) denominado *statecharts*. Essa interface é responsável, portanto, por facilitar o mapeamento do conhecimento de usuários menos experientes na construção de soluções inteligentes.

## 1.1 JUSTIFICATIVA

Sabe-se 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 a cada nova possibilidade de relacionamento para a geração de informações georreferenciada sempre envolve análises complexas, implementação de refinadas consultas SQL (*Structured Query Language*) (NAVATHE; ELMSARI, 2011), além de uma série de condições que necessitam ser tratadas para constituir a solução.

A estrutura do paradigma multiagente faz-se necessária pois permite a gestão do conhecimento, administração da granularidade das soluções e garante, principalmente, flexibilidade na construção de soluções por meio da inteligência artificial. Desta forma, soluções complexas podem ser tratadas de forma padronizada. Isso garante que deixe de ocorrer, dentro da empresa, um isolamento do conhecimento no que diz respeito a criação de soluções para problemas que, por vezes, podem ser complexos. Esse isolamento gera redundância, que pode ser caracterizada como retrabalho, e problemas na gestão do conhecimento.

Para apresentar o conjunto de ações integradas que visam automatizar o processo de refinamento da qualidade do Cadastro Técnico da Celg D, propõe-se a utilização de um ambiente multiagente dinâmico baseado na abordagem simbolista para atualização de dados do Cadastro Técnico georreferenciado. O ambiente dispõe, ainda, de recursos adicionais para o uso de outros modelos de raciocínios, dando-lhe a capacidade de resolver problemas por meio de mecanismos híbridos de inteligência artificial. Por fim, esse ambiente apresenta um recurso gráfico que é capaz de abstrair do usuário final parte da complexidade inerente à criação de soluções, levando a definição de uma interface que exija menos esforço cognitivo.

## **1.2 OBJETIVOS**

### **1.2.1 GERAIS**

Desenvolver um ambiente multiagente dinâmico que vise a atualização de dados do Cadastro Técnico georreferenciado e possua uma interface gráfica capaz de abstrair parte da complexidade inerente à criação de soluções.

### **1.2.2 ESPECÍFICOS**

- Levantar o estado da arte acerca dos temas necessários para o desenvolvimento do ambiente;
- Analisar a forma de utilização do Cadastro Técnico em relação aos sistemas legados visando compreender sua relação;
- Propor um modelo de ambiente multiagente com arquitetura baseada na abordagem simbolista que seja capaz de se integrar ao Cadastro Técnico;
- Propor uma interface gráfica que abstraia parte da complexidade na criação de soluções multiagentes;
- Propor uma modificação da arquitetura de forma que a mesma seja capaz de receber inteligência híbrida;
- Avaliar a arquitetura proposta e a interface gráfica no ambiente desenvolvido;
- Apresentar soluções criadas pelos colaboradores e impactos das mesmas.

## **1.3 METODOLOGIA**

A fim de alcançar o resultado esperado, foi feito um levantamento sobre o estado da arte acerca dos temas utilizados na pesquisa, sendo eles Inteligência Artificial e Sistemas Especialistas, a utilização de regras em Sistemas Especialistas, outros modelos computacionais, a natureza dos ambientes, a estrutura de ambientes multiagentes e, por fim, sobre o formalismo visual. Em seguida, houve uma análise dos sistemas legados da empresa e seus impactos na manutenção do Cadastro Técnico. A partir do que foi obtido desse processo, um modelo de ambiente multiagente com base na abordagem simbolista foi proposto. De acordo com esse modelo, foi elaborada uma interface gráfica capaz de auxiliar na criação de soluções. Como parte da elaboração do ambiente multiagente, foi proposta uma modificação no mesmo a fim de que ele pudesse comportar inteligência híbrida na construção de suas soluções. Após o desenvolvimento, foi feita a avaliação da arquitetura proposta e da interface gráfica no produto desenvolvido, bem como treinamento dos colaboradores para utilização do

ambiente. Por fim, os resultados obtidos foram analisados por meio dos impactos gerados pelas soluções criadas pelos colaboradores.

#### **1.4 ESTRUTURA DA DISSERTAÇÃO**

Este trabalho apresenta, em seu primeiro capítulo, a introdução do trabalho, justificativa, objetivos gerais e específicos, bem como a metodologia utilizada. Em seguida, no segundo capítulo, o referencial teórico acerca dos elementos utilizados no desenvolvimento do ambiente multiagente é apresentado. Como terceiro capítulo, tem-se o detalhamento da abordagem proposta apresentando a concepção do ambiente e o conceito de sua interface. O capítulo quatro possui os resultados obtidos com a aplicação do ambiente desenvolvido. Tem-se as conclusões do trabalho no capítulo cinco. E, por fim, no capítulo seis são apresentadas as referências bibliográficas utilizadas ao longo do projeto.

## **2 REFERENCIAL TEÓRICO**

Neste capítulo, são apresentados os referenciais teóricos utilizados ao longo do trabalho. São mostrados conceitos e detalhes sobre cada um dos componentes abordados, sendo eles: Inteligência Artificial, Sistemas Especialistas, Sistemas e Ambientes Multiagentes, a relação entre os Sistemas Especialistas e os Ambientes Multiagentes.

### **2.1 INTELIGÊNCIA ARTIFICIAL**

A Inteligência Artificial pode ter sua origem datada do século 4 a.c., quando filósofos como Aristóteles começaram estudos na intenção de formular um conjunto de silogismos que permitissem gerar conclusões a partir de premissas. Um exemplo disso foi a formalização da lógica proposicional. Desde então, diversas áreas contribuíram para a evolução da Inteligência Artificial, como Matemática, Economia, Neurociência, Psicologia, Engenharia de Computadores, Teoria de Controle Cibernética e Linguística, dentre outras.

A essência da IA é compreender e reproduzir a forma de pensar e agir do ser humano. Conseqüentemente, existem diversas abordagens e técnicas de Inteligência Artificial e cada uma delas pode ser melhor aplicada a diversificados grupos de problemas.

Também não existe um conceito único para definir Inteligência Artificial, porém, independente do autor ou da linha de IA, as definições mais conhecidas variam ao longo de duas dimensões. A primeira delas considera os processos de pensamento e o raciocínio, que medem os sucessos considerando a fidelidade ao desempenho humano. A segunda refere-se ao comportamento e tem seu sucesso medido em função do conceito de racionalidade - forma ideal de inteligência (RUSSEL; NORVIG, 2010).

Diferentes áreas e disciplinas contribuíram para se chegar ao estado da arte dessa área de conhecimento. De um modo geral a Inteligência Artificial se divide em dois grandes grupos: Inteligência Artificial Simbólica e Inteligência Artificial Conexionista. É importante ressaltar que dentro de cada um desses grupos ainda existem diversas técnicas que podem ser aplicadas de acordo com o problema e natureza do ambiente.

### **2.2 INTELIGÊNCIA ARTIFICIAL CONEXIONISTA**

Sistemas Conexionistas são técnicas computacionais que propõem uma forma particular de processamento da informação com base na organização física do cérebro humano. Esses sistemas caracterizam-se pela reunião de uma grande quantidade de unidades processadoras

(células ou neurônios) interligadas por um número expressivo de conexões que processam as informações de forma paralela.

O estudo do conexionismo remete aos anos de 1943 a 1955, época chamada de Geração da Inteligência Artificial. Nesse período, McCulloch e Pitts (1943 apud BRAGA; CARVALHO; LUDERMIR, 1998) propuseram um modelo de neurônio artificial.

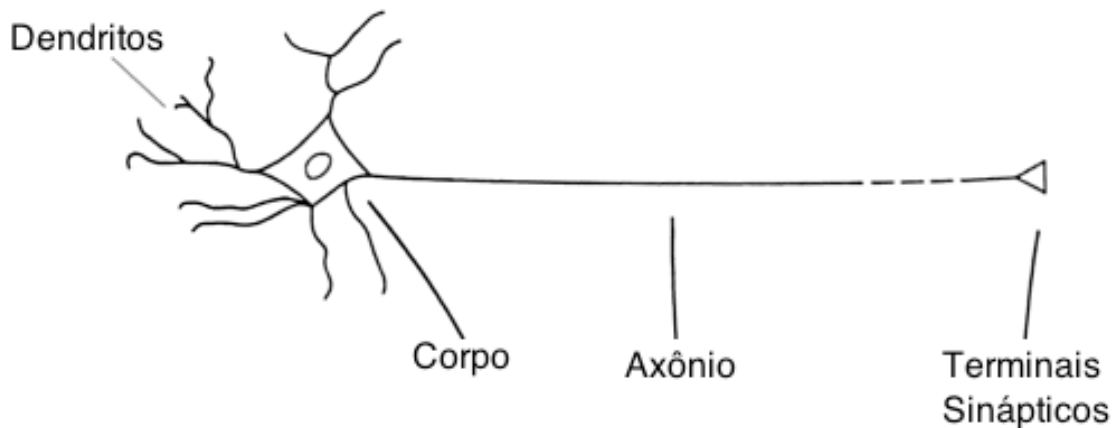


Figura 2.1 - Estrutura do neurônio humano. Adaptado de Levitan e Kaczmarek (2001)

Para propor esse modelo de neurônio artificial, McCulloch e Pitts basearam-se no conhecimento da fisiologia básica e da função dos neurônios do cérebro, na análise formal da lógica proposicional criada por Russell e Whitehead e na teoria da computação de Turing (RUSSELL; NORVIG, 2010).

Como base para o neurônio artificial, tem-se o neurônio humano que, de fato, é a unidade básica do cérebro. O neurônio artificial simula estímulos de entrada e saída como os que ocorrem no neurônio humano. A Figura 2.1 mostra o formato de um neurônio biológico.

McCulloch e Pitts (1943) descreveram um modelo artificial de neurônio como mostrado na Figura 2.2 e apresentaram suas capacidades computacionais, enquanto Donald Hebb (1949 apud BRAGA; CARVALHO; LUDERMIR, 1998) iniciou o estudo de técnicas de aprendizagem alguns anos depois. Esse último introduziu o conceito de variação sináptica na influência da elasticidade do neurônio (aprendizagem) e, também, criou uma regra para a atualização dos pesos nas sinapses.

Rosenblatt (1958) apresentou o modelo de rede neural Perceptron e propôs um algoritmo para treiná-lo. Em seu trabalho, foi capaz de demonstrar a capacidade do Perceptron de executar funções e classificar padrões após o treinamento.

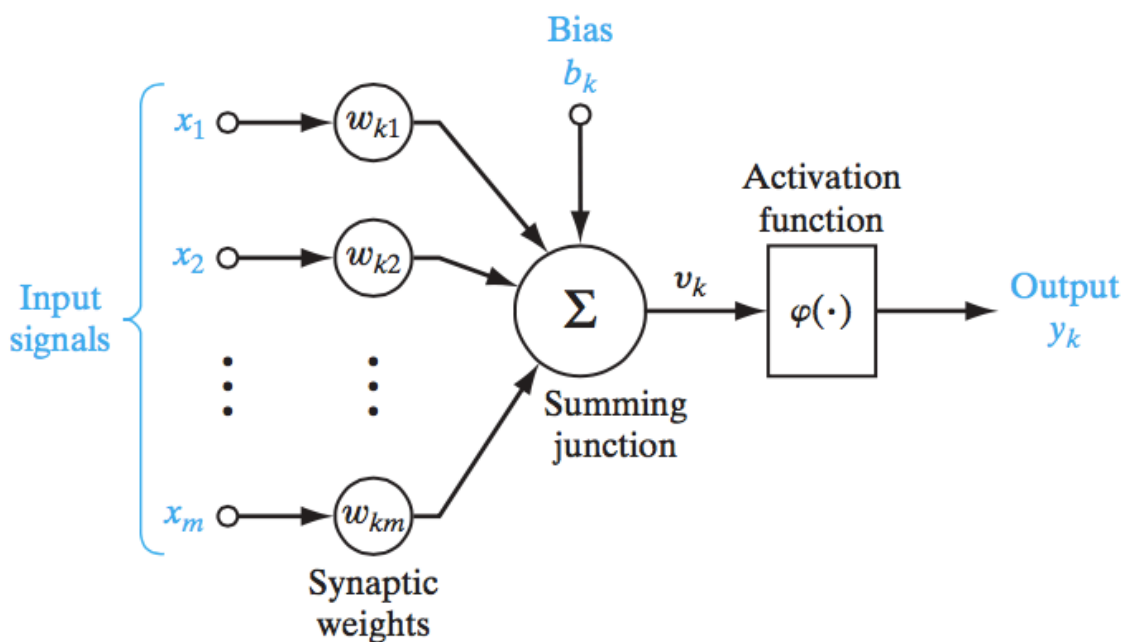


Figura 2.2 - Estrutura proposta por McCulloch e Pitts. Fonte: Haykin (2009)

Minsky e Papert (1969 apud BRAGA; CARVALHO; LUDERMIR, 1998), entretanto, foram responsáveis pela diminuição da visibilidade das redes neurais, pois, provaram que elas só eram capazes de resolver problemas linearmente separáveis.

Rumelhart, Hilton e Williams (1986) apresentaram um algoritmo que era capaz de resolver problemas linearmente separáveis e mais complexos, o que trouxe novamente o foco às redes neurais. Seu algoritmo utilizava uma técnica chamada de *back-propagation*. Aliado a isso, o fato da escola simbolista (inteligência artificial simbólica) não ter conseguido grandes avanços na solução de determinados tipos de problemas também contribuiu para o ressurgimento das redes neurais artificiais (RUSSELL; NORVIG, 2010).

Nos dias atuais, os diversos modelos de redes neurais desenvolvidos são utilizados nas mais diversas aplicações, sendo que alguns se adaptam melhor na solução de problemas de mineração de dados, outros no reconhecimento de voz, face etc. Graças a isso e aos avanços na área, o paradigma conexionista consolidou-se e atualmente é utilizado de modo comercial por indústrias e comércios, bem como por órgãos governamentais como a Receita Federal e Banco Central (CARVALHO, 2012).

Assim como ocorre com o cérebro humano, as redes neurais precisam aprender para em seguida tomar suas decisões. Todavia, para determinados tipos de problemas, não é possível

conhecer o resultado esperado. Sendo assim, os modelos conexionistas baseiam-se em duas grandes categorias: redes neurais artificiais (RNAs) supervisionadas e não supervisionadas.

Nos modelos de redes neurais supervisionadas é possível, a partir de dados anteriores, prever um determinado resultado. Um grande volume de entradas é submetido a uma rede neural artificial e, por meio de aprendizado, espera-se que ela seja capaz de responder a quaisquer dados posteriormente apresentados a ela. Esse volume de entradas costuma ser grande o suficiente de modo que modelos estatísticos ou matemáticos tradicionais não consigam explicar ou chegar ao resultado esperado. O nível de resposta a um conjunto de entradas de teste define o nível de assertividade da rede, ou seja, se ela consegue resolver o problema (HAYKIN, 2009). O modelo supervisionado mais conhecido é o Perceptron Multicamadas.

Alguns problemas não têm resultado conhecido e, nesses casos, o mais conveniente é utilizar modelos neurais não supervisionados. Nesses modelos a saída esperada não é conhecida e a quantidade de variáveis de entrada é muito grande. Um dos modelos de RNA não supervisionado mais conhecido é o modelo proposto por Kohonen (1997) pela sua especificidade de se auto-organizar, permitindo agrupamentos de características e perfis.

### **2.3 INTELIGÊNCIA ARTIFICIAL SIMBÓLICA E SISTEMAS ESPECIALISTAS**

A IA simbólica tem como um de seus pilares a necessidade de identificar o conhecimento do domínio (modelo do problema), representá-lo utilizando uma linguagem formal de representação e implementar um mecanismo de inferência para utilizar esse conhecimento. Essa implementação é, de fato, descrever na forma de regras preestabelecidas como se dará o processo de raciocínio do mecanismo que simula a inteligência artificial.

Na IA Simbólica existe a proposta de um sistema simbólico que é capaz de manifestar um comportamento inteligente, onde o comportamento inteligente global é simulado sem considerar os mecanismos responsáveis por este comportamento. Diversas técnicas são utilizadas na linha Simbólica como, por exemplo, a lógica proposicional, de predicados ou primeira ordem, modelos estatísticos e matemáticos, sistemas especialistas, entre outros.

Os conceitos da linha de pesquisa simbólica são encontrados no artigo *Physical symbol systems* de Newell (1980) e teve seu apogeu com o sucesso dos sistemas especialistas a partir da década de setenta.

### **2.3.1 SISTEMAS ESPECIALISTAS**

Os Sistemas Especialistas podem ser definidos como sistemas de inteligência artificial criados para resolver problemas em determinado domínio, cujo conhecimento utilizado é fornecido por pessoas que são especialistas naquele domínio.

Os sistemas convencionais são baseados em algoritmos e emitem resultados finais por meio do processamento de dados de maneira repetitiva, enquanto os Sistemas Especialistas são baseados em buscas heurísticas e trabalham com problemas que normalmente não possuem uma solução convencional organizada de forma algorítmica disponível, ou para montar esses algoritmos seria muito complexo e demorado. Segundo Russel e Norvig (2010), heurística pode ser denominada como: uma regra, simplificação, ou aproximação que reduz ou limita a busca por soluções em domínios que são difíceis e pouco compreendidos.

Com isso, pode-se afirmar que um Sistema Especialista é um sistema que foi desenvolvido e projetado para atender a uma aplicação determinada e limitada do conhecimento humano. Esse tipo de sistema também é capaz de emitir uma decisão apoiado em conhecimento justificado, a partir de uma base de informações, da mesma forma que algum especialista de determinada área do conhecimento humano seria capaz de justificar. Desta forma, o conhecimento construído entre a interação de um Sistema Especialista e os especialistas humanos de uma área torna-se permanente (TEIXEIRA, 1998). Além disso, Sistemas Especialistas não estão suscetíveis a estresse e cansaço, estão mais disponíveis que o especialista humano e, também, podem dar respostas mais rápidas e detalhadas (MAIA, 2012).

Para tomar decisões, os sistemas especialistas formulam hipóteses, sempre utilizando como base fatos que estão armazenados em suas bases de conhecimentos. Durante o processo de raciocínio, é verificada a importância dos fatos encontrados e estes são comparados com as informações já adquiridas sobre tal assunto. A descoberta de novos fatos influencia no processo de raciocínio; este que é baseado no conhecimento acumulado previamente.

Para garantir o sucesso dessa forma de raciocínio, os conhecimentos prévios existentes para o sistema especialista devem ser suficientes, pois, a falta de conhecimento pode induzir o sistema a chegar a uma conclusão errada. Por fim, Sistemas Especialistas não são influenciados por agentes externos, o que leva a conclusão de que, para situações idênticas ou parecidas, as decisões tomadas devem ser as mesmas.

### 2.3.2 COMPOSIÇÃO DOS SISTEMAS ESPECIALISTAS

Atualmente, existem diversas arquiteturas de sistemas especialistas sendo utilizadas, e dentre essas, a de mais fácil entendimento, é composta por três elementos principais: a base de conhecimento, o quadro negro e o mecanismo de inferência, conforme é mostrado na Figura 2.3. O modelo do quadro negro é baseado na metáfora criada por Newell (1962) em que ele faz uma alusão aos trabalhadores de uma empresa observando um mesmo quadro negro, sendo que cada trabalhador tem a capacidade de observar tudo que há no quadro e de julgar quando há algo relevante a ser adicionado nele.

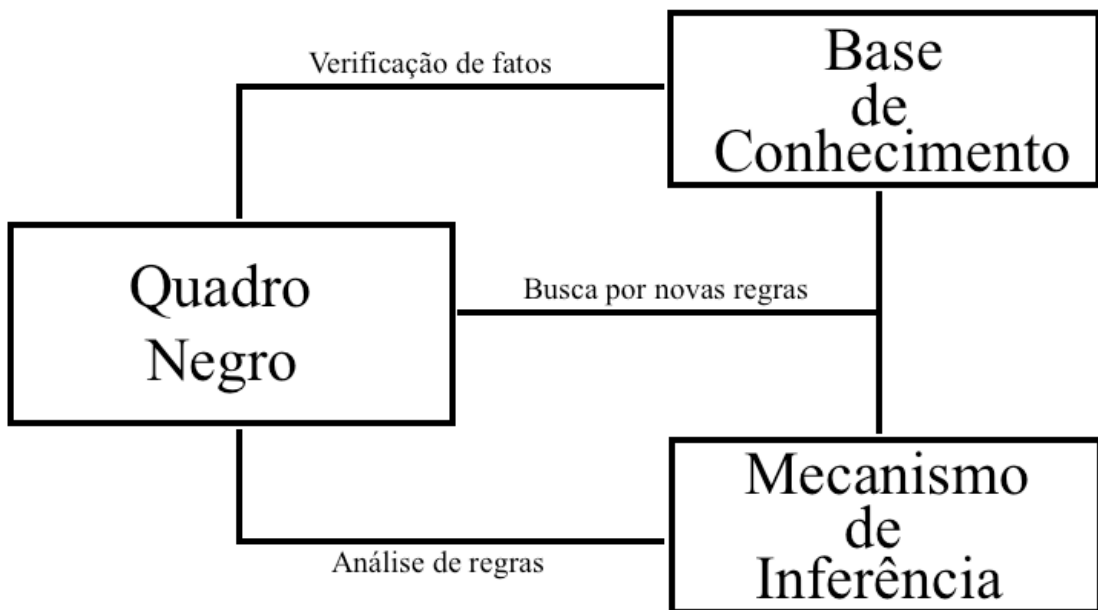


Figura 2.3 - Elementos básicos de um sistema especialista

#### *Base de conhecimento*

A principal responsabilidade da base de conhecimento dos sistemas especialistas é o armazenamento de informações, fatos e regras do domínio da especialidade do sistema. A base com dados, arquivos, registros e seus relacionamentos estáticos são substituídas por uma base de regras, fatos e heurísticas que correspondem ao conhecimento do especialista, ou dos especialistas do domínio sobre o qual foi construído o sistema. Sendo assim, a base de conhecimento não é apenas uma coleção de informações (NILSSON, 1982 apud MENDES, 1997). É importante observar que nem sempre uma base necessita conter todo o conhecimento de um domínio específico. De fato, cada base pode possuir uma gama de informações diferentes e independentes (CORKILL, 1991).

A base de conhecimento interage com a máquina de inferência e com o usuário, e chega às conclusões do problema que são solicitadas pelo próprio usuário (MAIA, 2012). Durante esse processo de interação, novas informações são requeridas ao usuário pelo sistema de forma que o caminho até a solução seja reduzido (MENDES, 1997). Por não funcionar com um agente ativo dentro do sistema, a base de conhecimento não precisa conhecer a existência de outras bases, entretanto, elas devem ser capazes de comunicarem entre si (CORKILL, 1991).

Um processo de aprendizagem automática pode acontecer no sistema especialista. Segundo Mendes (1997), ele "é capaz de analisar e gerar novas regras em sua base de conhecimento e/ou armazenar informações sobre novos fatos, ampliando assim a capacidade do sistema de resolver problemas, a cada vez que o mesmo for utilizado".

A construção das bases de conhecimento é, de fato, uma das partes mais importantes no desenvolvimento de um sistema especialista, visto que, toda a informação a ser utilizada na inferência e solução de problemas consiste nelas (NILSSON, 1982).

### *Quadro negro*

O quadro negro é o mecanismo de comunicação dos sistemas especialistas. As informações contidas no quadro negro são vistas e manipuladas pelos agentes do sistema como na metáfora de Newell (1962). Desta forma, é possível afirmar que o modelo do quadro negro é baseado em eventos de adição, exclusão ou alteração de informações (CORKILL, 1991).

Ao longo dos anos, foram desenvolvidas diferentes formas de comunicação utilizando esse modelo. O conceito de espaço de tuplas foi criado para auxiliar a comunicação. O espaço pode ser preenchido com informações em forma de tupla e os agentes podem ler, escrever, consumir ou criar novas tuplas nesse espaço (HUGET, 2003). Inicialmente a linguagem Linda utilizou esse conceito que, então, foi difundido para outras linguagens como Java (JavaSpaces) e Jason (CArtAgO). Fica claro, então, que é um modelo difundido para interação entre agentes em ambientes multiagente.

As informações são gravadas e apagadas no quadro negro pelo sistema especialista, em um processo de inferência até o momento em que o sistema consiga chegar a uma solução desejada, ou seja, o quadro-negro é uma memória usada para avaliação das regras que são

recuperadas da base de conhecimento para se chegar a uma solução (CRAIG, 1995). Portanto, as informações permanecem no quadro negro por um período curto.

Neste componente ficam armazenadas, também, as perguntas já respondidas pelo usuário, bem como possíveis soluções intermediárias. Desta forma, o quadro negro evita que o usuário responda a uma mesma pergunta duas vezes. A linha de raciocínio utilizada na busca da solução de um problema também é armazenada no quadro negro para utilização pelo sistema especialista no momento de justificar ao usuário como chegou àquela solução.

### *Mecanismo de Inferência*

O mecanismo de inferência é responsável por utilizar fatos e estratégias de resolução de problemas para chegar às soluções (TEIXEIRA, 1998). A base de conhecimento deve ser sólida e bem elaborada para que a interpretação dos fatos ocorra de forma devida e, então, soluções corretas sejam geradas.

O mecanismo de inferência examina o conteúdo da base de conhecimento e decide quais regras são satisfeitas pelos fatos ou objetos, priorizando as regras satisfeitas e executando as regras de maior prioridade. Em seguida, publica as informações provindas da inferência no quadro negro e faz perguntas ao usuário para dar continuidade ao processo de geração da solução. Uma agenda de prioridade de regras pode ser definida para auxiliar na definição da ordem de execução (MAIA, 2012). A tomada de decisão é feita por meio da análise das bases de conhecimento e processo de inferência.

Por meio das formas de funcionamento dos sistemas especialistas, é possível classificá-los em 10 categorias (HAYES-ROTH; WATERMAN; LENAT, 1983):

- a) Interpretação: são os sistemas que inferem descrições de situações à partir da observação de fatos fazendo uma análise de dados e procurando determinar as relações e seus significados;
- b) Diagnósticos: sistemas que detectam falhas oriundas da interpretação de dados;
- c) Monitoramento: interpreta as observações de sinais sobre o comportamento monitorado;
- d) Predição: a partir de uma modelagem de dados do passado e do presente, este sistema permite uma determinada previsão do futuro;

- e) Planejamento: o sistema prepara um programa de iniciativas a serem tomadas para se atingir um determinado objetivo;
- f) Projeto: sistema capaz de justificar a alternativa tomada para o projeto final, e de fazer uso dessa justificativa para alternativas futuras;
- g) Depuração: trata-se de sistemas que possuem mecanismos para fornecerem soluções para o mau funcionamento provocado por distorções de dados;
- h) Reparo: o sistema desenvolve e executa planos para administrar os reparos verificados na etapa de diagnóstico;
- i) Instrução: o sistema de instrução tem uns mecanismos para verificar e corrigir o comportamento do aprendizado dos estudantes;
- j) Controle: sistema que governa o comportamento geral de outros sistemas (não apenas de computação). É o mais completo, de um modo geral, pois deve interpretar os fatos de uma situação atual, verificando os dados passados e fazendo uma predição do futuro. Apresenta os diagnósticos de possíveis problemas, formulando um plano ótimo para sua correção. Este plano de correção é executado e monitorado para que o objetivo seja alcançado.

### 2.3.3 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 2.4.

```
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>
```

Figura 2.4 - Regras estruturando uma base de conhecimento

O usuário enuncia uma condição através da estrutura “SE”, 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” (FILHO; JÚNIOR, 2013). Essa sintaxe garante algumas características interessantes, como:

- a) Expressa de forma objetiva o conhecimento específico do agente;
- b) A representação facilita a inclusão e a exclusão de regras da base de conhecimento, permitindo o aprendizado;
- c) Viabiliza a execução paralela de regras existentes na base de conhecimento;
- d) 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 a base de conhecimento se torna muito extensa ou as condições para a execução das regras se tornam muito específicas, necessitando da criação de condições compostas.

Por outro lado, esse modelo de expressão de conhecimento pode se tornar complicado para usuários menos experientes quando:

- a) A base de conhecimento se torna muito extensa. Excesso de regras e de condições, bem como a combinação de condições para o disparo dessas regras;
- b) As condições para a execução das regras se tornam muito específicas, necessitando da criação de condições compostas;
- c) O encadeamento das atividades passa por conjunto de regras que, ora executam de forma paralela, ora estabelecem um sequenciamento temporal para a sua execução.

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.

#### **2.3.4 SISTEMAS ESPECIALISTAS E AMBIENTES MULTIAGENTES**

Nos Sistemas Especialistas, há uma separação entre o conhecimento acerca do domínio do problema e outras formas de conhecimento dentro do sistema. Outras formas de conhecimento seriam, por exemplo, como imprimir caracteres ou modificar dados em resposta a ações do usuário. Segundo Wooldridge (2009), algumas das diferenças entre agentes e Sistemas Especialistas são:

- Sistemas Especialistas "clássicos" dependem totalmente do comando de um usuário e funcionam como intermediário entre este e o ambiente em que a ação é tomada, enquanto agentes possuem uma forte noção de autonomia e são capazes de decidir se devem ou não executar uma requisição feita por outro agente.
- Sistemas Especialistas geralmente não são capazes de comportamento reativo ou proativo, enquanto agentes são flexíveis (reativos, proativos e sociais).
- Sistemas Especialistas geralmente não possuem habilidade de comunicação quando se trata de cooperação, coordenação e negociação, enquanto os agentes são sociais, independentes e colaborativos.

Conforme visto anteriormente, o conjunto de conhecimento a respeito do domínio do problema é chamado de base de conhecimento, enquanto o mecanismo de solução de problemas é chamado de máquina de inferência. Qualquer sistema organizado dessa forma é chamado de sistema baseado em conhecimento.

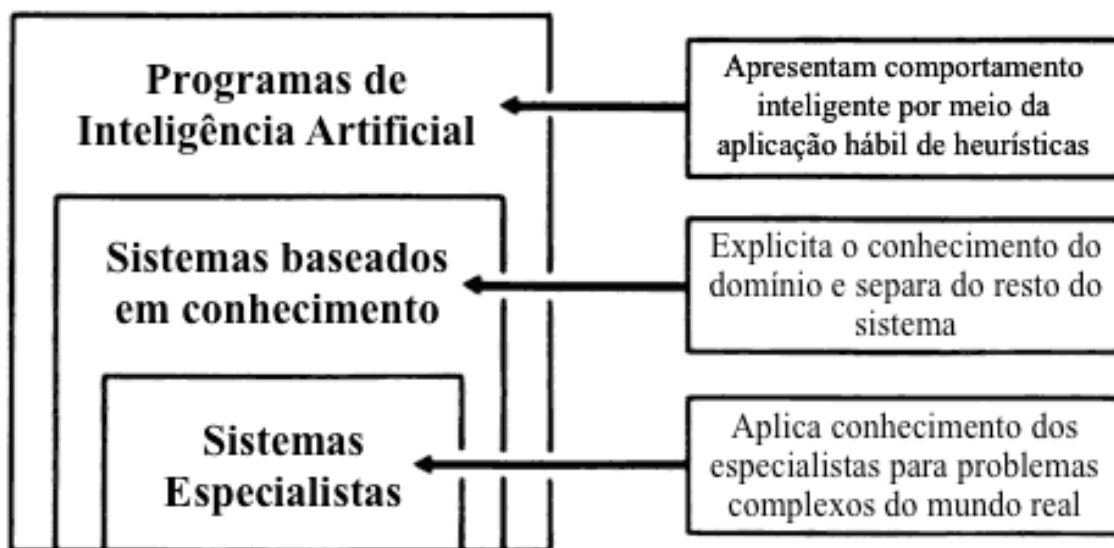


Figura 2.5 - Sistemas especialistas e baseados em conhecimento. Adaptado de Waterman (1986)

Podemos, então, afirmar que todo sistema especialista é, em sua essência, um sistema baseado em conhecimento, embora a recíproca nem sempre é verdadeira, uma vez que existem programas dentro do escopo de IA que apresentam esta estrutura de conhecimento, mas não podem ser considerados sistemas especialistas, conforme mostra a Figura 2.5.

Ao levantar as características dos sistemas especialistas é possível identificar que o ambiente multiagente proposto tem características bem próximas, pois, ambos possuem em suas estruturas uma máquina de inferência, uma base de conhecimento e um quadro negro, ou seja, utilizam a mesma logística de funcionamento.

## **2.4 OUTROS MODELOS COMPUTACIONAIS INTELIGENTES**

Além das RNAs supervisionadas e não supervisionadas, é possível encontrar outros modelos computacionais inteligentes pelo mundo. Cada um desses modelos é responsável por tratar de problemas de uma forma diferente, visando sempre o melhor resultado.

Dentro da computação evolutiva que, de fato, baseia-se na evolução de Darwin, tem-se os algoritmos genéticos. Um algoritmo genético, segundo Russel e Norvig (2010), é dado pela combinação de dois estados pais ao invés de serem gerados pela modificação de um único estado. Desta forma, parece-se com a reprodução sexuada dos seres vivos. Tal algoritmo é amplamente utilizado em problemas de otimização e busca local. É importante observar que dentro da computação evolutiva existem outras técnicas como, por exemplo, enxame de partículas, colônia de formigas e mapas auto-organizáveis.

No campo do raciocínio incerto, tem-se a lógica difusa ou lógica *Fuzzy*. Russel e Norvig (2010) dizem que pesquisadores em lógica difusa propuseram uma ontologia que permite imprecisão de forma que um evento pode ser "uma espécie de" verdadeiro. Sendo assim, a lógica difusa admite que um fato causa sérias dificuldades.

Portanto, percebe-se que, ao longo dos anos, diversas técnicas de inteligência artificial foram desenvolvidas para resolver diversos tipos de problemas. Algumas delas podem ser mais indicadas para certas ocasiões, porém, não tanto para outras. Por ser uma área em desenvolvimento, novas técnicas e heurísticas podem surgir todos os dias.

## **2.5 NATUREZA DOS AMBIENTES**

Para resolver problemas, um agente inteligente deve agir em um ambiente a partir da percepção que ele tem do mundo em que irá atuar e essa percepção pode ser obtida a partir dos seus sensores. Portanto, quanto mais complexo o ambiente, mais complexos devem ser os processos de inteligência e computação aplicados. Com o objetivo de facilitar a compreensão, são apresentadas abaixo algumas categorias de ambientes de tarefas (RUSSEL; NORVIG, 2010):

- a) Completamente observável *versus* Parcialmente observável: Completamente observável ocorre quando os sensores são capazes de ter acesso completo ao estado do ambiente, a todo instante. Quando isso não é possível, diz-se que o ambiente é parcialmente observável.
- b) Agente único *versus* Multiagente: Agente único é quando um único agente inteligente é capaz de resolver o problema. Todavia, em muitos dos casos é necessária uma solução multiagente, em que se encontram vários agentes inteligentes trabalhando em cooperação.
- c) Determinístico *versus* Estocástico: Determinístico ocorre quando o próximo estado do ambiente é determinado pelo estado atual e pela ação do agente. Quando isso não ocorre, temos um ambiente estocástico ou aleatório.
- d) Episódico *versus* Sequencial: Episódico ocorre quando as ações do agente podem ser divididas em episódios atômicos. Todavia quando uma ação atual pode afetar uma ação futura, dizemos o ambiente é sequencial.
- e) Estático *versus* Dinâmico: Um ambiente é estático se ele não mudar enquanto o agente está atuando e dinâmico se mudar.
- f) Discreto *versus* Contínuo: Em um ambiente discreto o estado do ambiente, o tempo e percepções do agente podem ser mensurados de forma finita. Um sistema de condução inteligente de automóvel, o ambiente é contínuo, uma vez que os elementos de percepção e tempo não podem ser mensurados a priori.
- g) Conhecido *versus* Desconhecido: Conhecido ou desconhecido refere-se ao estado de conhecimento do próprio agente. Quando as saídas esperadas (ou prováveis) são conhecidas, temos o conceito de conhecido, caso contrário o agente terá que aprender sobre o ambiente para então tomar as decisões.

## 2.6 SISTEMAS MULTIAGENTES

Dentro do contexto da Inteligência Artificial Distribuída, Torsun (1995) classifica-a em *Distributed Problem Solving* (DPS), *Multi-Agent Systems* (MAS) e *Parallel Artificial Intelligence* (PAI). A DPS ou Resolução Distribuída de Problemas decompõe o problema em módulos por meio de uma abordagem descendente (*top-down*) desenhada especificamente para um problema em particular, onde grande parte do raciocínio sobre a solução é inserido pelo próprio projetista. A PAI, Inteligência Artificial Paralela, interessa-se mais no

desempenho do que em avanços conceituais, preocupando-se principalmente em desenvolver linguagens e algoritmos de computação paralela. Por último, Os MAS ou Sistemas Multiagentes, caracterizam-se pelo estudo e implementação de agentes que interajam de forma autônoma e que trabalhem juntos para resolver um determinado problema ou alcançar um objetivo. Em suma, pode-se dizer que Sistemas Multiagentes são sistemas constituídos de múltiplos agentes que interagem ou trabalham em conjunto de forma a realizar um determinado conjunto de tarefas ou objetivos (WOOLDRIDGE, 2009). Esses objetivos podem ser comuns a todos os agentes ou não.

Os agentes dentro de um sistema multiagente podem ser heterogêneos ou homogêneos, colaborativos ou competitivos etc. A definição dos tipos de agentes depende da finalidade da aplicação em que o sistema multiagente está inserido.

Em sistemas multiagentes de agentes reativos, os mesmos apresentam apenas o comportamento de ação e reação, não possuindo, por vezes, nenhuma inteligência individual. Entretanto, em conjunto, os agentes reativos são capazes de gerarem soluções, visto que, a combinação de suas habilidades e conhecimentos pode resultar na ferramenta necessária para a solução de determinado problema. Em sistemas multiagentes de agentes cognitivos, os mesmos podem comunicar-se entre si, coordenar ações e negociar, visto que, esses agentes possuem conhecimentos, intenções, crenças e competências (FERBER; GASSER, 1991). A combinação desses tipos de agentes é chamado de agente híbrido e é explorado por Wooldridge e Jennings (1995) com desafio na comunicação entre eles e controle das ações executadas pelos mesmos.

Segundo Jennings, Sycara e Wooldridge (1998), algumas razões para o crescimento do interesse em pesquisas com sistemas multiagentes são:

- A capacidade de fornecer robustez e eficiência;
- A capacidade de permitir interoperabilidade entre sistemas legados;
- A capacidade de resolver problemas cujo dado, especialidade ou controle são distribuídos.

Apesar disso, existem muitos desafios e dificuldades em relação a projeto e implementação de agentes. Com base em Wooldridge (2009), pode-se apresentar alguns deles como:

- a) Como programar, descrever, decompor e alocar problemas e sintetizar resultados com um grupo de agentes inteligentes?
- b) Como permitir a comunicação e a interação entre agentes? Quais linguagens de comunicação e protocolos usar? O que e quando comunicar?
- c) Como assegurar que agentes atuam coerentemente tomando decisões ou executando ações?
- d) Como permitir agentes individuais atuar e raciocinar sobre ações, planos e conhecimento sobre outros agentes em ordem de coordená-los? Como raciocinar sobre o estado de seus processos coordenados?
- e) Como identificar e reconciliar pontos de vistas muito diferentes e intenções conflitantes em uma coleção de agentes que tentam coordenar suas ações?
- f) Como equilibrar efetivamente computação local e comunicação? Mais genericamente, como gerenciar alocação de recursos limitados?
- g) Como evitar ou minimizar comportamento prejudicial do sistema, tal como um comportamento caótico?
- h) Como projetar e construir sistemas multiagentes práticos? Como projetar plataformas tecnológicas e metodologias de desenvolvimento para sistemas multiagentes?

Podemos resumir em apenas três os principais desafios que um sistema multiagente possui:

- a) Como deve ser realizada a comunicação entre os agentes e que tipo de protocolos devem utilizar?
- b) Como a interação entre os agentes deve ocorrer? Que linguagem os agentes devem utilizar para interagirem entre si e combinar seus esforços?
- c) Como garantir a coordenação entre os agentes para que haja uma coerência na solução do problema que estão tentando resolver?

## **2.7 AMBIENTE MULTIAGENTE**

Ambientes de sistemas multiagentes, segundo Weiss (1999), provem uma infraestrutura especificando protocolos de comunicação e interação, são, geralmente, abertos e não possuem

apenas um projetista (não são centralizados), contém agentes que são autônomos e distribuídos e podem ser competitivos ou cooperativos.

Esses ambientes caracterizam-se pela existência de um contexto de suporte à sobrevivência dessas entidades (os agentes inteligentes), permitindo a eles atuarem de forma independente uns dos outros e realizarem tarefas específicas. É possível fazer uma analogia entre o Ambiente Multiagente e uma organização, onde seus funcionários atuam em conjunto, mas cada um desempenhando o seu papel específico é algo perfeitamente válido (GASSER, 1992). Os agentes inteligentes são, usualmente, compostos por uma máquina de inferência que tem a responsabilidade de "pensar", isto é, relacionar símbolos (dados) de entrada (fatos, eventos etc.), gerando como resultado uma conclusão representada por um ou mais símbolos de saída (ações, advertências etc.) (RICH; KNIGHT, 1993).

A inteligência do agente é estruturada através de um conjunto de regras que norteia o comportamento da máquina de inferência, sendo que essas regras são normalmente representadas por meio de operadores lógicos ou, eventualmente, da tradicional estrutura das linguagens de programação, o se-então-senão. O grau de flexibilidade de um agente é extremo, pois, são as regras armazenadas nele que definem seu comportamento. Essas regras são idealizadas e alimentadas pelo criador do agente (FERBER; GASSER, 1991).

## **2.8 FORMALISMO VISUAL E O USO DE GRÁFICOS PARA REPRESENTAÇÃO DE ESTADOS**

Harel (1987) introduziu seu conceito denominado *statechart* que foi tido como uma extensão do formalismo visual convencional dos diagramas de estados. Em seu trabalho, ele afirma que os *statecharts* são diagramas de estados, somados à profundidade, ortogonalidade e transmissão de comunicação. Em seu exemplo, Harel mostra a criação de um relógio (*Citizen Quarts Multi-Alarm III*) utilizando *statecharts* para mapear todas as ações e transições de estados possíveis do objeto. Tem-se, na Figura 2.6, a representação do relógio utilizado como exemplo e nela é possível observar os elementos de profundidade, ortogonalidade e comunicação anteriormente apontados e, ainda, perceber a capacidade de se aninhar estados. A contribuição de Harel é tão significativa que, mais tarde, seu conceito de diagrama de estados foi adotado pela UML (*Unified Modeling Language*) (LARMAN, 2005).

Mais tarde, Harel (1988) propôs um modelo de expressão de gráficos chamado de Higraphs que, por meio 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 (1988) está a aplicação na representação do conhecimento, aproximando-se do que está sendo proposto neste trabalho.

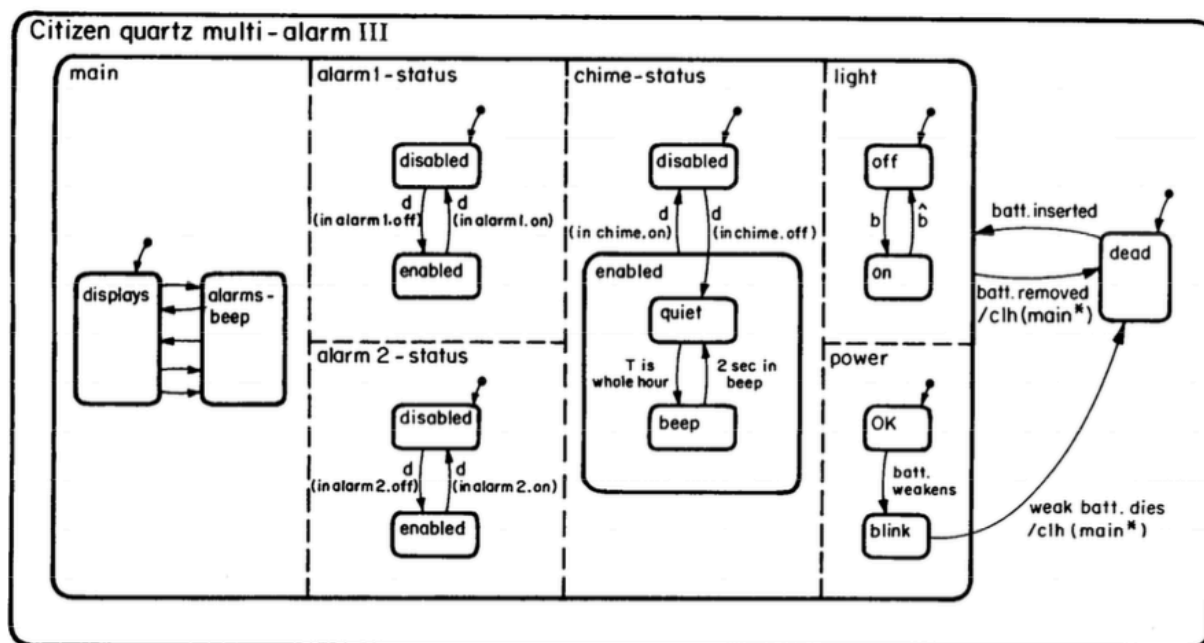


Figura 2.6 - Statechart do Citizen Quartz Multi-Alarm III. Fonte: Harel (1987)

Além do Higraphs, também existem outros modelos como o Digraph, que é menos genérico que o primeiro e permitem relações de inclusão de adjacência. Um dos modelos mais recentes é o de gráficos agrupados (*clustered graphs*), sendo que ele consiste de gráficos não direcionados e seu particionamentos recursivo em subgráficos. Esse último é relativamente genérico e pode lidar com várias aplicações que utilizam hierarquia (KASYANOV, 2012).

Outras abordagens utilizando-se de formalismo visual são encontradas para aplicação em engenharia de software como no trabalho de Jacob (1983), onde a expressão gráfica modela interfaces complexas. Em seu trabalho, ele utiliza o modelo formal para apresentar uma sugestão de melhoria no sistema de troca de mensagens militares. Como é possível observar na Figura 2.7, ele faz uso de diagrama de transição de estados para representar as formas que a interface do comando de "Login" pode assumir. Cada estado e suas transições representam uma condição que gerará uma instrução diferente levando, assim, ao próximo estado até que todo o comando seja finalizado.

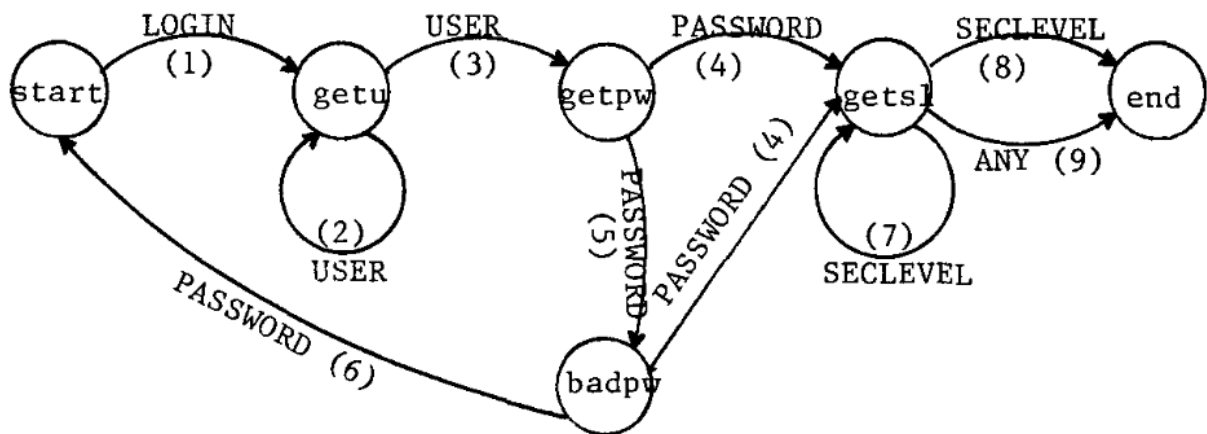


Figura 2.7 - Diagrama de Estados para o comando "Login". Fonte: Jacob (1983)

Kugler, Larjo e Harel (2010) propuseram uma linguagem denominada *Biocharts*, que é derivada dos *statecharts* de Harel. O modelo é focado para a construção de modelos de sistemas biológicos completamente executáveis e tem como um dos objetivos permitir que biólogos possam participar da construção de modelos computacionais de forma natural. Desta forma, eles pretendem, por meio de formalismo visual, diminuir a complexidade da construção de soluções complexas por pessoas com menos experiência em uma linguagem ou paradigma específicos.

Subburaj e Urban (2011) utilizaram o formalismo visual da AUML (*Agent Unified Modeling Language*) a partir da linguagem de especificação Descartes para desenvolvimento e configuração de agentes e suas transições.

Com uma abordagem genérica, Hilaire et al. (2000) desenvolveram um modelo de prototipação de ambientes multiagentes. Nesse modelo, eles utilizaram da linguagem Object-Z (SMITH, 2000) e *statecharts* para construir um modelo que fosse capaz de representar diferentes ambientes que resolvem os mais diversos problemas. Essa combinação de linguagem e formalismo visual permite, segundo eles, a especificação de aspectos reativos e transformacionais dos ambientes multiagentes, além de sua prototipação por simulação.

O modelo de estado proposto neste trabalho toma conhecimento dos conceitos e aplicações de outros modelos da literatura, porém, não se preocupa em encadear os estados porque entende que o comportamento do agente não é determinístico e depende 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 ABORDAGEM PROPOSTA

Neste capítulo é detalhado o modelo proposto para o ambiente multiagente. O modelo base e os elementos que compõem o modelo desenvolvido são detalhados, bem como o modo de funcionamento e intercomunicação dos mesmos. Por fim, um exemplo utilizando toda a estrutura é apresentado para melhor entendimento do produto final pretendido.

#### 3.1 MODELO PROPOSTO

O cenário de soluções complexas e diferenciadas 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. Essa conclusão irá gerar uma atualização em um outro banco de dados com o objetivo de torná-lo mais exato. A Figura 3.1 mostra o fluxo genérico de soluções que refinam o Cadastro Técnico a partir da busca de conhecimentos nos outros sistemas da organização.

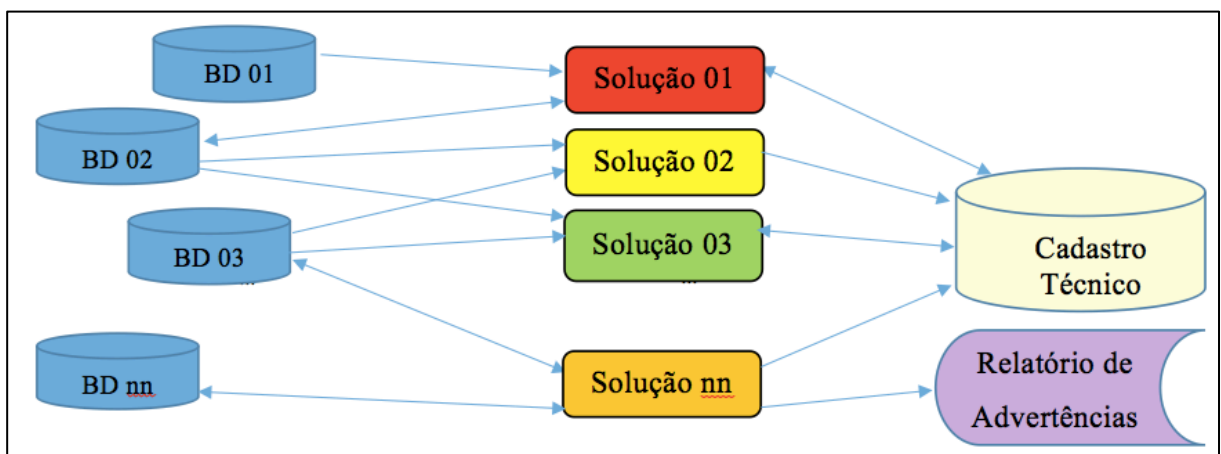


Figura 3.1 - Visão da arquitetura multiagente usada neste trabalho

Há a preocupação com a dificuldade do usuário final na construção de 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 baseado na abordagem simbolista (SCHALKOFF, 2009) e com recursos adicionais para o uso de outros modelos de raciocínios, dando-lhe a capacidade de resolver problemas por meio de

mecanismos híbridos de inteligência artificial (RUSSEL; NORVIG, 2010; NEGNEVISTKY, 2005). O objetivo é abstrair do usuário final parte da complexidade inerente à criação de soluções nos moldes descritos e ilustrados na Figura 3.1, levando a definição de uma interface que exija menos esforço cognitivo.

A interface aqui proposta é estruturada a partir da análise de problemas dentro da Celg D, onde soluções multiagente podem 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.

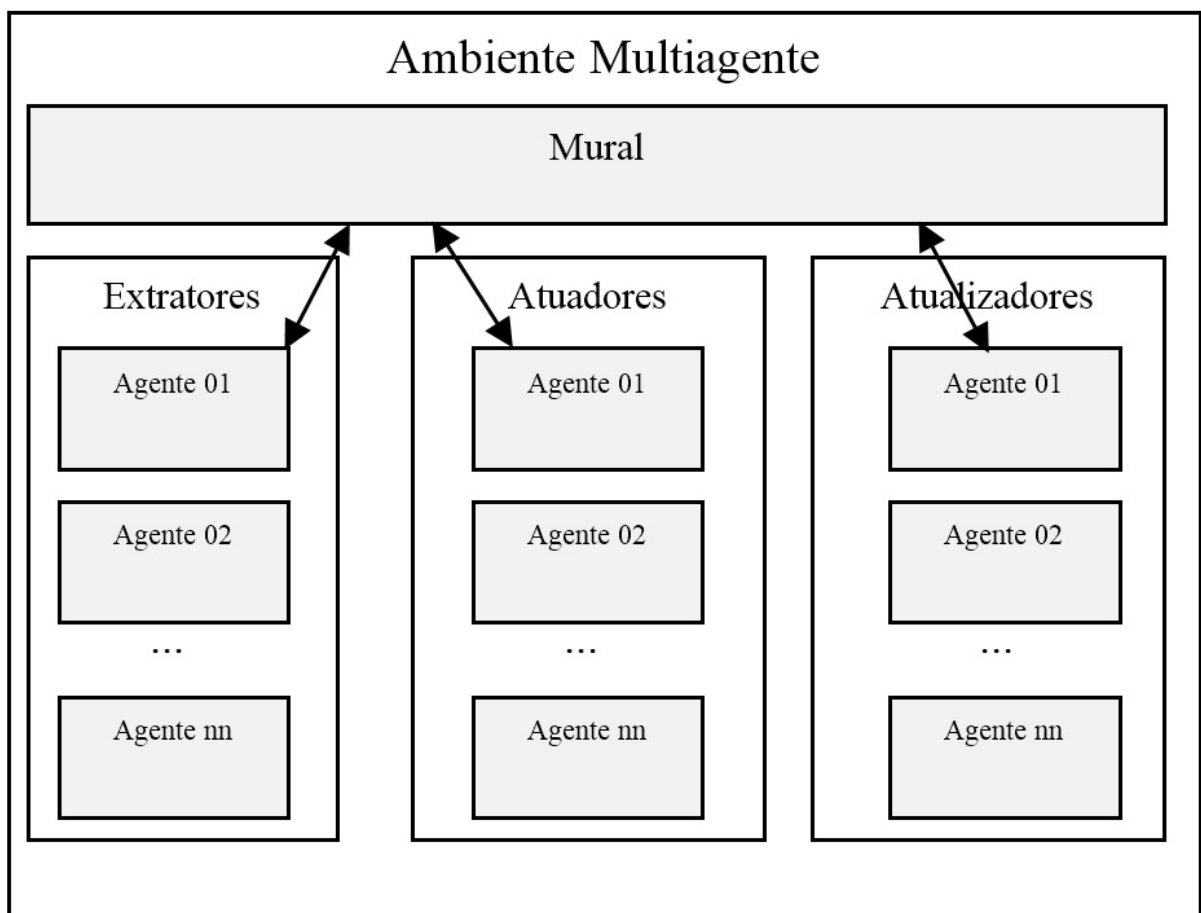


Figura 3.2 - Estrutura do ambiente multiagente proposto

Assim, a aplicação proposta é dividida em um conjunto de agentes inteligentes (D'INVERNO; LUCK, 2004) divididos em três camadas distintas, que deve 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 3.2, cujos elementos principais são: o Mural, que é a área para a comunicação entre agentes; Agentes Extratores, que são entidades capazes de buscar informações em bancos de dados, expondo tais informações no mural; Agentes Atuadores, que são as estruturas que realizarão as atividades de raciocínio; e Agentes Atualizadores, que são as entidades que executam atualizações em bancos de dados. Essa forma de abordagem utilizando três grupos de agentes (atuador, atualizador e extrator) apresenta-se como mais uma contribuição. Embora diferentes modelos possam ser encontrados na literatura e similaridades possam ser identificadas (MAALAL; ADDOU, 2011; ANDREADIS et al., 2014), nenhum deles é igual ao da abordagem deste trabalho.

A arquitetura multiagente criada é similar à suportada pelo JADE (BELLIFEMINE; CAIRE; GREENWOOD, 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.

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.

### **3.2 MURAL E SÍMBOLOS**

O primeiro elemento do ambiente multiagente é o Mural, que é a representação do quadro negro apresentado anteriormente. Ele é, de fato, a única forma de comunicação entre os agentes, visto que, é o único elemento que proporciona as condições necessárias para que isso aconteça. A exposição e retirada de símbolos do Mural é a única forma de troca de informações no ambiente. O símbolo é a estrutura básica que é exposta no Mural, sendo constituído por um nome que o identifica e 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.

### **3.3 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 ela se encontra.

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

- a) *Query by Example*: o uso da grade QBE (RAMAKRISHNAN; 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.
- b) 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.

Independente do tipo de configuração a ser realizada pelo usuário, o resultado é uma tabela obtida pela execução do SQL digitado ou construído a partir da grade QBE. Este resultado pode 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 indica o nome do símbolo de saída que é exposto no Mural com o resultado da consulta.

### **3.4 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.

- 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).
- 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.

### 3.5 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.

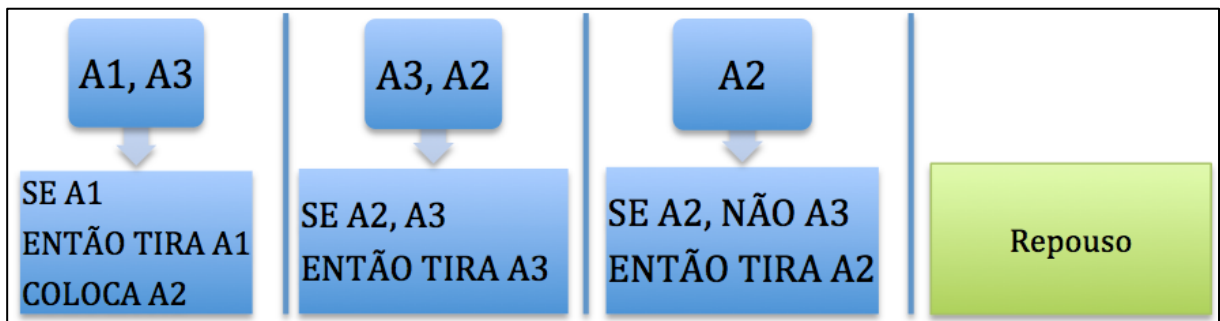


Figura 3.3 - Ciclo de inferência composto por três passos

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 3.3 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 que se iniciam mais simples e, à medida que ele avança na busca de uma solução, tornam-se mais complexos.

### 3.5.1 ESTRUTURAÇÃO DE CONTEXTOS

Na arquitetura proposta, a integração das três estruturas, a de conversão de informação em símbolo (agente extrator), inferências baseadas em regras da base de conhecimento (agente atuador) e conversão de símbolo em ação (agente atualizador), dá ao ambiente de agentes inteligentes a possibilidade de assumir várias configurações. A integração dessas três estruturas por meio de uma estruturação de contextos utilizando-se de uma forma visual é uma das contribuições deste trabalho.

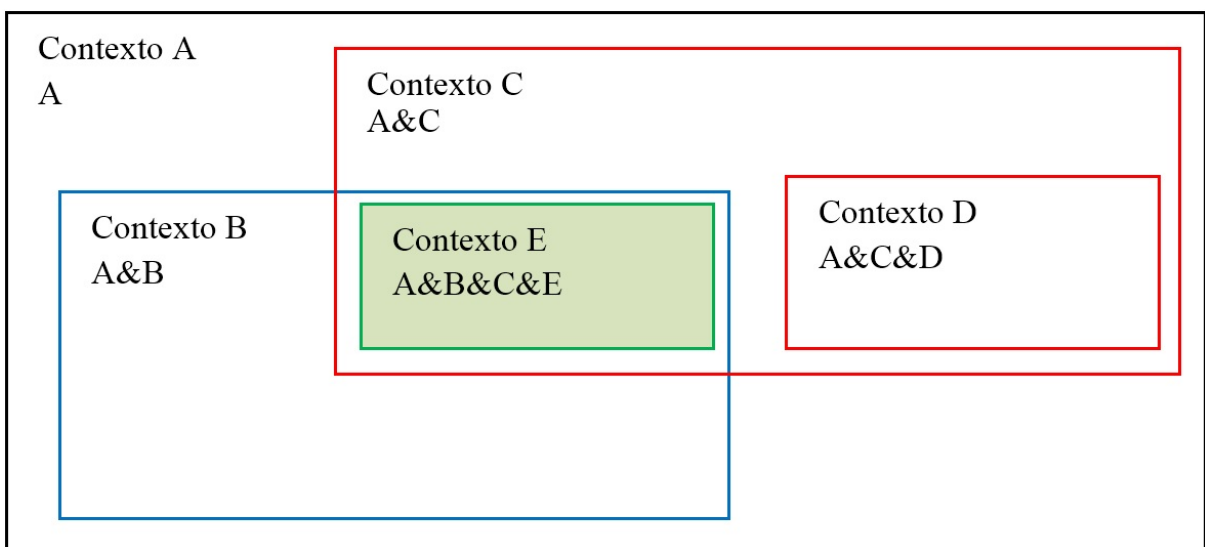


Figura 3.4 - Representação gráfica dos contextos

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 3.4 ilustra essa condição, onde uma região de contexto inicial pode ser a mais geral e simples, a região “A”.

É possível observar que as condições compostas e o “afunilamento” em um processo de inferência podem ser facilmente expressos pela expressão gráfica de regiões que, por sua vez, contém sub-regiões. O raciocínio inicia-se em “A”, algumas ações geram “B” no contexto, as regras do contexto A&B são ativadas e assim por diante.

O quadro do “Contexto E”, em destaque na Figura 3.4, 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. Por meio 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 3.4, “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.

### **3.5.2 PROTÓTIPO DO AGENTE ATUADOR**

Normalmente, uma solução pode ser estruturada com apenas um agente Atuador, embora, a critério do desenvolvedor, possam ser utilizados mais. Toda a complexidade inerente à solução pode ser exposta em um único diagrama de estados importantes de forma relativamente simples.

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:

- a) Identificar os símbolos envolvidos na estruturação de uma determinada solução;

- b) Estabelecer os contextos, de forma gráfica, conforme a Figura 3.4. 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;
- c) 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.

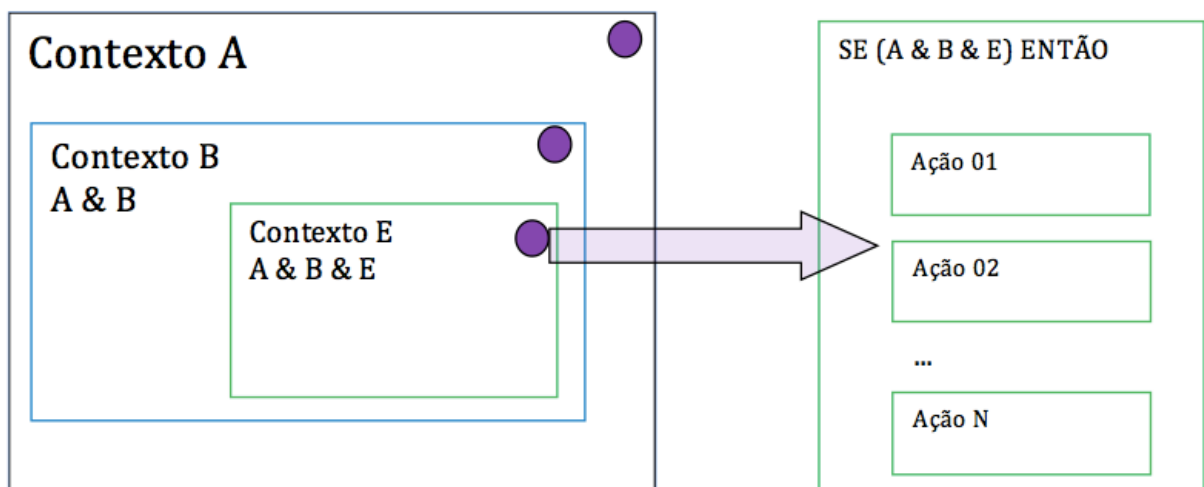


Figura 3.5 - Protótipo da tela de criação do agente atuador

Para o registro das regras, propõe-se neste trabalho que a partir de um ícone localizado no canto superior direito do diagrama de contextos, seja possível passar para outra interface, onde o conjunto de ações possa ser inserido. Para a definição das ações que o agente deve tomar, o usuário deve clicar no círculo no canto superior dos estados, conforme é mostrado na Figura 3.5. É possível, então, descrever a relação de ações que serão executadas pelo agente caso o estado se ative. O registro dessas ações completa o processo de alimentação da base de conhecimento. Se necessário, o usuário poderá inserir códigos Java (DEITEL; DEITEL, 2011) a fim de flexibilizar o ambiente em caso de soluções mais complexas. Os códigos devem estar sintaticamente corretos para que possam ser executados. O ambiente efetua a validação dos códigos antes de salvar os dados do agente.

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 Figura 3.6 mostra a ilustração de uma porção de código simplificado e traduzido a partir da Figura 3.4 que é gerado automaticamente pelo ambiente. Em níveis mais baixos, o código é, de fato, em linguagem Java. Esse código é parte da formação da base de conhecimento do agente Atuador e originado das informações contidas no diagrama de estados encadeados. É possível constatar que a criação dos contextos da Figura 3.4 mostra-se menos trabalhosa e menos complexa que o desenvolvimento da lógica representada pela Figura 3.6, mesmo quando utilizada linguagem traduzida.

```
SE Simbolo A
ENTÃO
    SE Simbolo B & Simbolo C
    ENTÃO
        SE Simbolo E
        ENTÃO ExecutarAcaoABCE();
    SENÃO
        SE Simbolo B
        ENTÃO ExecutarAcaoAB();
        SENÃO
            SE Simbolo C
            ENTÃO
                SE Simbolo D
                ENTÃO ExecutarAcaoACD();
                SENÃO ExecutarAcaoAC();
```

Figura 3.6 - Porção do código gerado a partir do diagrama de estados da Figura 3.4

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

Experimentos iniciais realizados com usuários sem conhecimento prévio de construções de regras lógicas em um sistema especialista demonstraram que eles são capazes de entender com facilidade e, também, construir as estruturas de estados encadeados. Quando, posteriormente, transformados em estruturas lógicas SE...ENTÃO, já não eram tão naturais de serem entendidas pelos próprios construtores da estrutura.

### 3.6 EXEMPLO UTILIZANDO A ABORDAGEM GRÁFICA

Para um melhor entendimento da proposta, foi elaborado um exemplo com dados fictícios que utiliza toda a estrutura de agentes extratores, atualizadores e atuadores. Nesse exemplo, é possível observar, também, protótipos de funcionalidades do ambiente, além da aplicação da estruturação de contextos na resolução de problemas.

No interior do estado de Goiás, existe a cidade Villa onde moram 6 habitantes. A cidade está dividida em dois bairros, o A e o B. Nela todos os habitantes que tenham idade superior a 30 anos e se chamarem João, obrigatoriamente, necessitam ter no último sobrenome o indicador de qual bairro eles moram. Portanto, qualquer João que possui mais de 30 anos e mora no bairro A deve ter o último sobrenome A, enquanto o que mora no bairro B deve ter o nome B como o último sobrenome.

Porém, com o passar do tempo percebeu-se que essa regra não está sendo seguida, conforme é exibido na Tabela 3.1. Desta forma, estruturam-se agentes inteligentes para resolver esse problema.

<b>Nome</b>	<b>Último Sobrenome</b>	<b>Bairro</b>	<b>Idade</b>	<b>RG</b>
Joaquina	Silva	5	20	123
Luísa	Ferreira	5	24	456
João	José	5	31	789
João	Maria	6	10	655
João	Silva	6	35	788
João	Ferreira	6	70	654

Tabela 3.1 - Habitantes da cidade Villa

São criados três agentes inteligentes: um extrator, um atuador e um atualizador. O agente extrator é responsável por encontrar os moradores da cidade Villa que tenham mais de 30 anos e que se chamem João. O agente atuador é responsável por verificar se o sobrenome do morador está igual ao bairro em que o mesmo mora e, caso não esteja, deve solicitar a

correção do sobrenome. Por último, o agente atualizador é responsável pela correção dos sobrenomes que estiverem errados.

*Passo 1 – Criação do agente extrator*

O banco de dados da cidade é composto pelas tabelas exibidas na Figura 3.7.

Cidade		
Cód.Cidade	Nome	Estado
1	Villa	Goiás

Bairro		
Cód.Cidade	Cód.Bairro	Nome
1	5	A
1	6	B

Morador				
Nome	Sobrenome	Cód.Bairro	Idade	RG
Joaquina	Silva	5	20	123
Luisa	Ferreira	5	24	456
João	José	5	31	789
João	Maria	6	10	655
João	Silva	6	35	788
João	Ferreira	6	70	654

Figura 3.7 - Estrutura do banco de dados exemplo

Para a criação do agente extrator é definido o contexto “Busca moradores João” e, dentro dele, são definidas as regras para a extração de informação conforme a Figura 3.8.

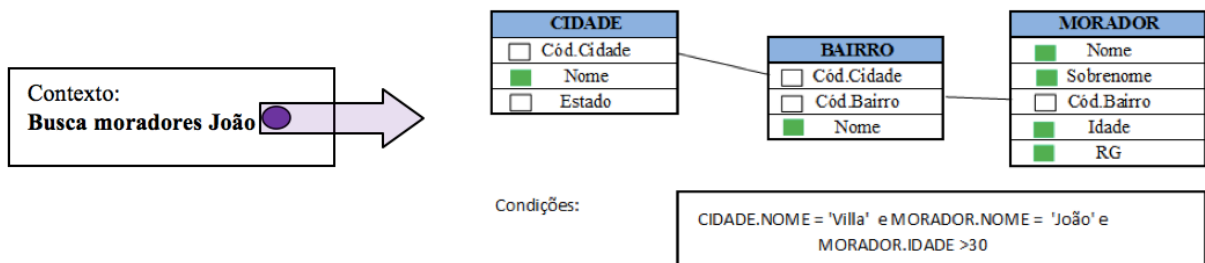


Figura 3.8 - Representação do agente extrator exemplo

O mural do ambiente é atualizado periodicamente e irá solicitar a ação desse agente extrator quando o mesmo for criado.

*Passo 2 – Criação do agente atuador*

O agente atuador é responsável por identificar se os moradores fixados no mural pelo agente extrator possuem o sobrenome correto e, em caso negativo, deve anexar que o sobrenome do morador precisa ser corrigido. Para a construção do agente atuador, é necessário pensar na solução lógica do problema e estruturar em contextos as condições e estados à medida que eles forem descobertos.

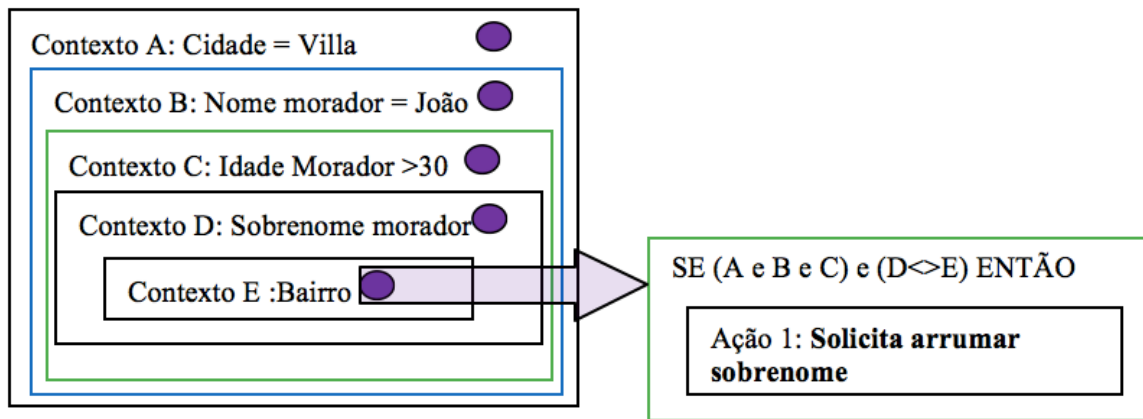


Figura 3.9 - Representação do agente atuador exemplo

Conforme a Figura 3.9, são identificados os seguintes contextos para a criação do agente atuador: cidade, morador, idade, bairro e sobrenome do morador. Apesar de existir a possibilidade de resolução desse problema com menos contextos, os mesmos foram separados para facilitar futuras manutenções nas regras.

*Passo 3 – Criação do agente Atualizador*

O agente atualizador é responsável por corrigir os sobrenomes dos moradores que o agente atuador identifica ter algum problema. Para isso, é criado somente o contexto "Arrumar sobrenome", conforme exibido na Figura 3.10.

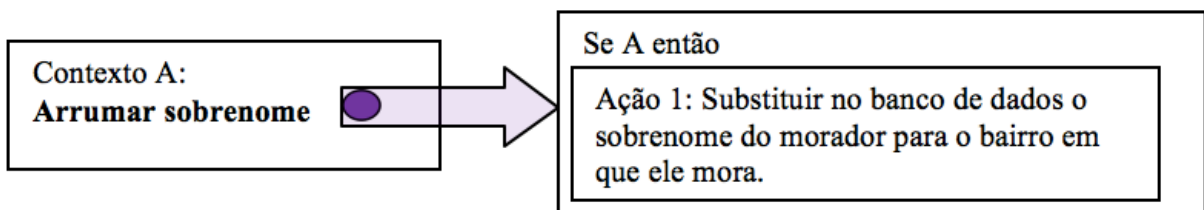


Figura 3.10 - Representação do agente atualizador exemplo

*Passo 4 – Execução*

Com a execução dos agentes criados para a resolução do problema, o seguinte resultado é obtido:

- 1) O mural emite o símbolo para que o agente extrator inicie suas atividades e, após a execução, o mesmo fixa no mural as informações que foram extraídas. As informações extraídas são:

- a. Agente extrator 1 - Cidade: Villa, Bairro: A, Nome: João, Sobrenome: José  
Idade: 31 RG: 789;
  - b. Agente extrator 2 - Cidade: Villa, Bairro: B, Nome: João, Sobrenome: Silva  
Idade: 31 RG: 788;
  - c. Agente extrator 3 - Cidade: Villa, Bairro: B, Nome: João, Sobrenome: Ferreira  
Idade: 31 RG: 654.
- 2) Com as informações fixadas no mural pelo agente extrator, o agente atuador entra em ação fazendo as verificações das regras conforme sua base de conhecimento.
- a. Agente atuador 1: verifica que o nome da cidade é Villa, que o morador de RG 789 se chama João e tem mais de 30 anos, porém, identifica que o sobrenome do morador está diferente do bairro em que ele mora. Sendo assim, fixa no mural que o sobrenome precisa ser corrigido.
  - b. Agente atuador 2: verifica que o nome da cidade é Villa, que o morador de RG 788 se chama João e tem mais de 30 anos, porém identifica que o sobrenome do morador está diferente do bairro em que ele mora. Sendo assim, fixa no mural que o sobrenome precisa ser corrigido.
  - c. Agente atuador 3: verifica que o nome da cidade é Villa, que o morador de RG 654 se chama João e tem mais de 30 anos, porém identifica que o sobrenome do morador está diferente do bairro em que ele mora. Sendo assim, fixa no mural que o sobrenome precisa ser corrigido.
- 3) Com as informações fixadas no mural pelo agente atuador, o agente atualizador entra em ação de faz as correções das regras conforme sua base de conhecimento:
- a. Agente atualizador 1: atualiza o sobrenome do morador de RG 789 para "A".
  - b. Agente atualizador 2: atualiza o sobrenome do morador de RG 788 para "B".
  - c. Agente atualizador 3: atualiza o sobrenome do morador de RG 564 para "B".

Por meio desse resultado, é possível observar que o problema é resolvido e que a estruturação dos contextos, embora simples, é capaz de corrigir todos os casos de sobrenome incorreto existentes no banco de dados, bem como casos que ocorrerem no futuro.

### 3.7 INTELIGÊNCIA HÍBRIDA

As diferentes técnicas da IA possuem seus pontos fortes e fracos, seja com lógica *Fuzzy*, algoritmos genéticos ou redes neurais. Visando melhores resultados, um ambiente multiagente híbrido pode ser criado. Desta forma, a combinação de uma ou mais técnicas com a estrutura multiagente é capaz de diminuir as limitações das mesmas quando individuais. Cada agente é responsável por aplicar sua devida técnica no conjunto de dados e devolver o resultado que, se necessário, pode ser utilizado por outro agente que aplica uma técnica diferente (KHOSLA, 1997; ZHANG; ZHANG, 2004). Vários exemplos de sistemas multiagentes com inteligência híbrida podem ser encontrados na literatura.

Wang, Abdulla e Salcic (2007) desenvolveram um ambiente multiagente híbrido que utiliza tanto lógica *Fuzzy* quanto redes neurais. Em seu trabalho, eles dispõem de uma sala chamada de DEIR (*Distributed Embedded Intelligence Room*) que possui sensores para controle de temperatura, pressão, fumaça e movimento. Por meio desses sensores e seu ambiente multiagente, a sala é capaz de monitoramento autônomo e controle de serviços em espaços inteligentes. Eles observaram que apenas o framework MAFIS (*Multi-Agent Fuzzy Inference System*) não era suficiente para monitoramento de comportamento e decidiram utilizar, também, redes neurais para auxiliar nessa tarefa. O modelo resultante foi chamado de MANNFIS (*Multi-Agent Neural Net Fuzzy Inference System*) e sua arquitetura é mostrada na Figura 3.11, onde é possível observar que, a partir dos dados numéricos, cada técnica executa seus procedimentos a fim de chegarem num resultado comum.

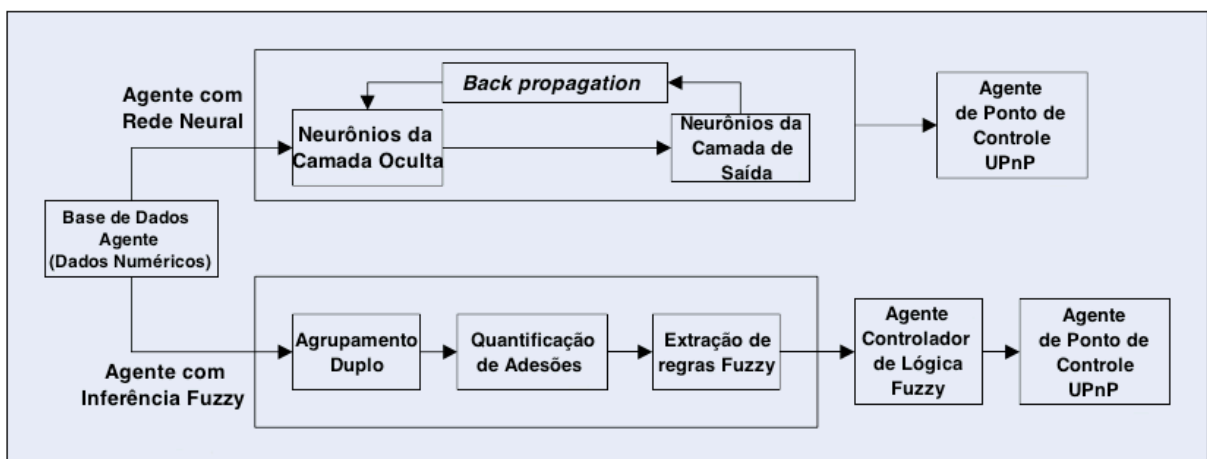


Figura 3.11 - Arquitetura MANNFIS. Adaptado de Wang, Abdulla e Salcic (2007)

Fowler e Hammell (2011) criaram um modelo híbrido para melhorar o processo de *data mining* de diversas fontes de dados. Eles planejam utilizar esse modelo para detecção de

atividades cibernéticas maliciosas. Em seu trabalho, eles não decidiram quais tipos de inteligência utilizar, entretanto, desenvolveram a arquitetura mostrada na Figura 3.12 que é capaz de receber diversos tipos de técnicas a fim de transformar dados diversos em conhecimento.

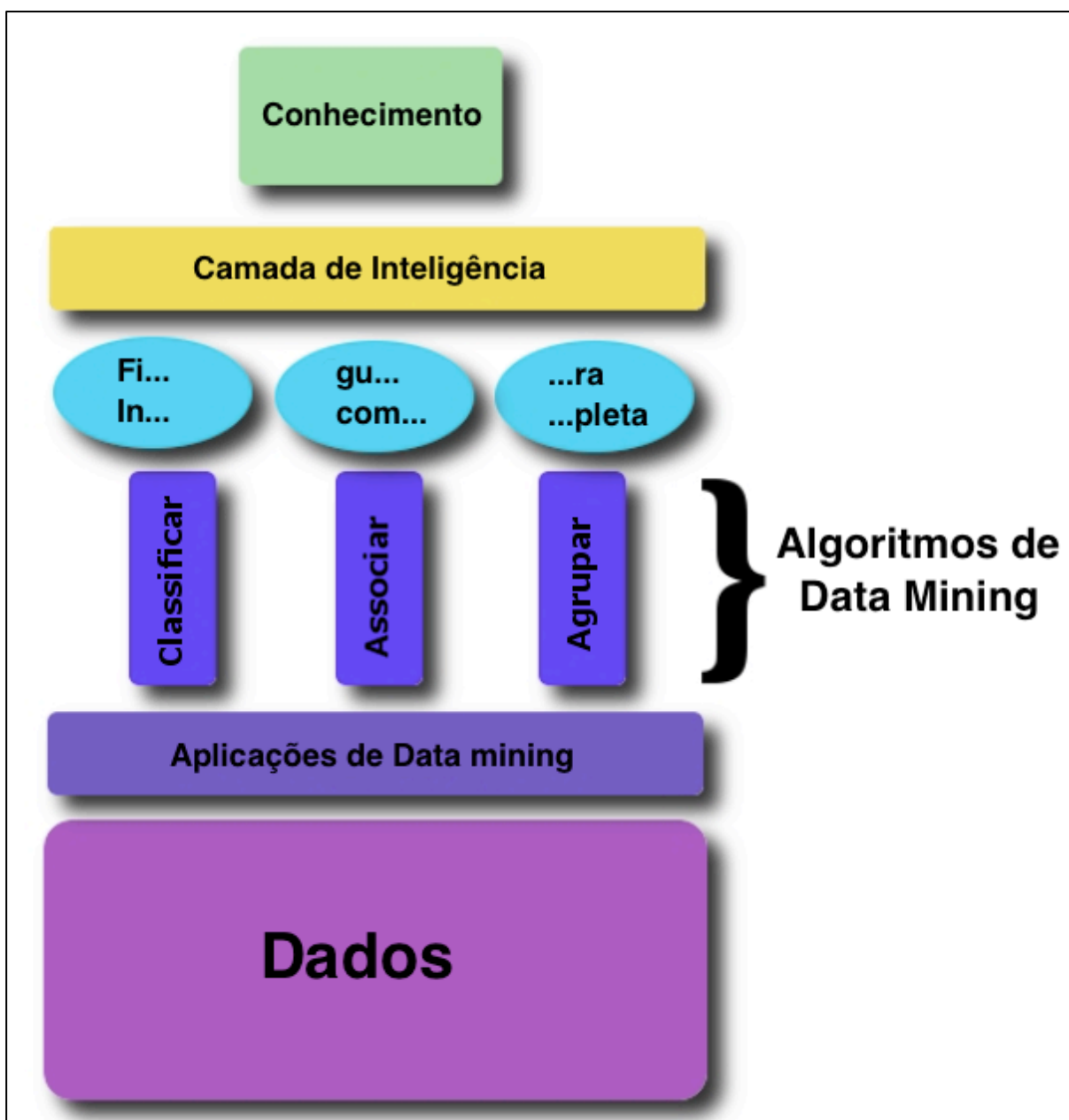


Figura 3.12 - Arquitetura híbrida para data mining. Adaptado de Fowler e Hammell (2011)

Ghosh, Nafalski e Tweedale (2013) desenvolveram um modelo chamado de IMADA (*Intelligent Multi-Agent Decision Analyzer*) que possui quatro agentes autônomos individuais, sendo um agente para manipulação de dados, um agente da base de conhecimento, um agente

de lógica *Fuzzy* e um agente de rede neural. Essa arquitetura pode ser vista na Figura 3.13 e foi utilizada para analisar dados em tempo real sobre estresse no trabalho.

Pešková et al. (2014) pensaram em um modelo híbrido para meta-aprendizado em um domínio de *data mining*. Seu sistema, por meio de diferentes técnicas de inteligência artificial, é capaz de recomendar técnicas de *data mining* ao usuário para cada conjunto de dados construindo, assim, uma base de meta-conhecimento.

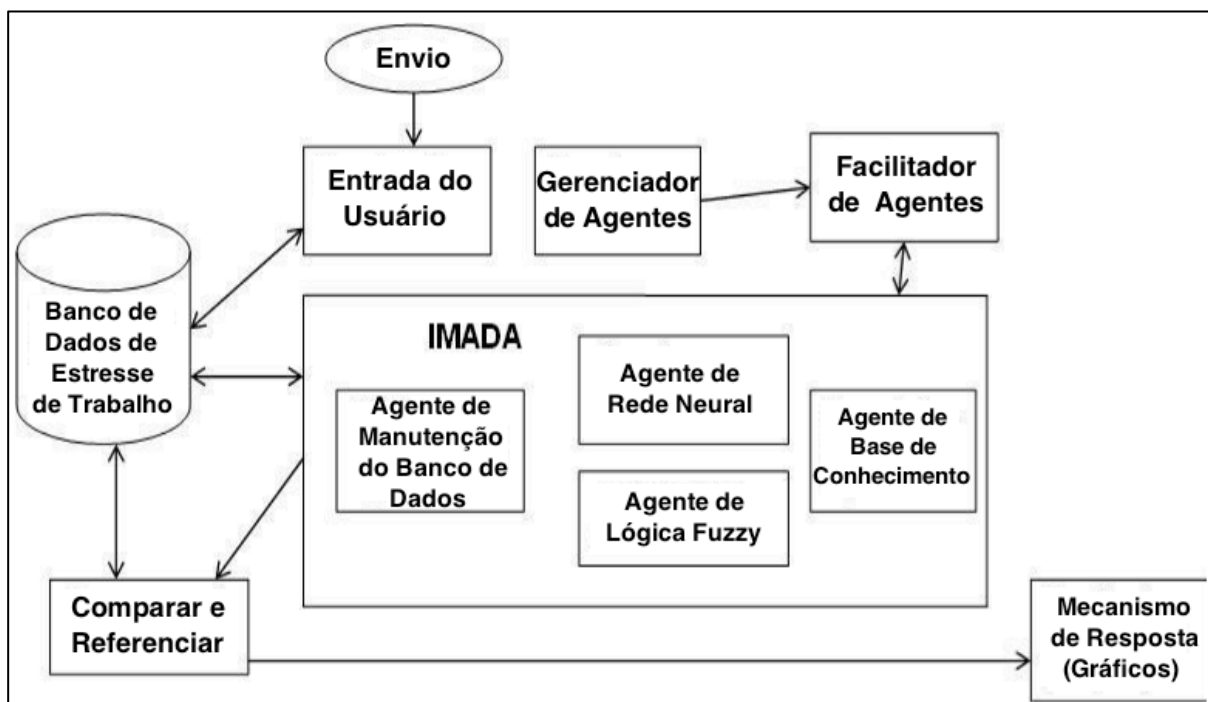


Figura 3.13 - Arquitetura híbrida. Adaptado de Ghosh, Nafalski e Tweedale (2013)

Tendo em vista os conceitos e os exemplos, acredita-se que o ambiente proposto neste trabalho é capaz de ser híbrido, combinando diferentes técnicas de inteligência artificial. Isso se dá pelo fato de que o símbolo, por conter uma estrutura de dados, pode receber e transmitir toda a informação necessária para um módulo que implementa outras abordagens. Nesse caso, a base de conhecimento da máquina de inferência do agente tomará as decisões por meio de conhecimento gerado por outra abordagem de IA, caracterizando, assim, a forma híbrida da solução. Em experimentos iniciais, a aplicação de lógica *Fuzzy* para resolução de problemas se mostrou possível. Entretanto, é importante observar que, para problemas mais complexos, adaptações podem ser necessárias para o funcionamento ideal de todos os recursos.

## **4 RESULTADOS OBTIDOS**

Neste capítulo são apresentados os resultados obtidos após o desenvolvimento do ambiente. Tem-se, a seguir, a interface do ambiente, ou seja, o mural e todos os elementos importantes para a criação dos agentes atuadores, extratores e atualizadores. Além disso, são apresentadas soluções reais e desenvolvidas pelos colaboradores da Celg D. Algumas dessas soluções foram desenvolvidas para resolver problemas observados pelos próprios colaboradores do setor de geoprocessamento.

### **4.1 INTERFACE DO AMBIENTE**

#### **4.1.1 PREPARAÇÃO PARA CRIAÇÃO DE SOLUÇÕES**

Os processos de extração e atualização podem ser feitos em bancos de dados, planilhas eletrônicas e arquivos de texto. Desta forma, antes de se criar uma solução, é necessário que essas origens de dados sejam especificadas. No caso de bancos de dados, são necessárias instruções de conexão com nome de usuário e senha. É importante observar a necessidade do envolvimento de uma equipe de Tecnologia da Informação no processo, visto que, essas informações de conexão, por exemplo, não são de domínio de todos os departamentos.

#### **4.1.2 CRIAÇÃO DAS SOLUÇÕES E SEUS AGENTES**

Tendo sido especificadas as origens de dados que serão utilizadas nas soluções, é dado início ao processo de construção das mesmas e, conseqüentemente, dos agentes que a compõem.

##### *A. Criação de um agente extrator*

Na Figura 4.1, é possível visualizar a tela de criação e definição de um agente extrator. Nesse caso específico, a origem de dados é um banco de dados e, portanto, são mostradas tabelas e suas respectivas ligações.

Todo esse trabalho de escolha de tabelas e ligação entre elas é feita de forma interativa (arrastar e soltar), o que elimina a utilização de comandos SQL. Isso torna o processo mais didático e requer menos conhecimento específico acerca de uma linguagem de alto nível, no caso o SQL.

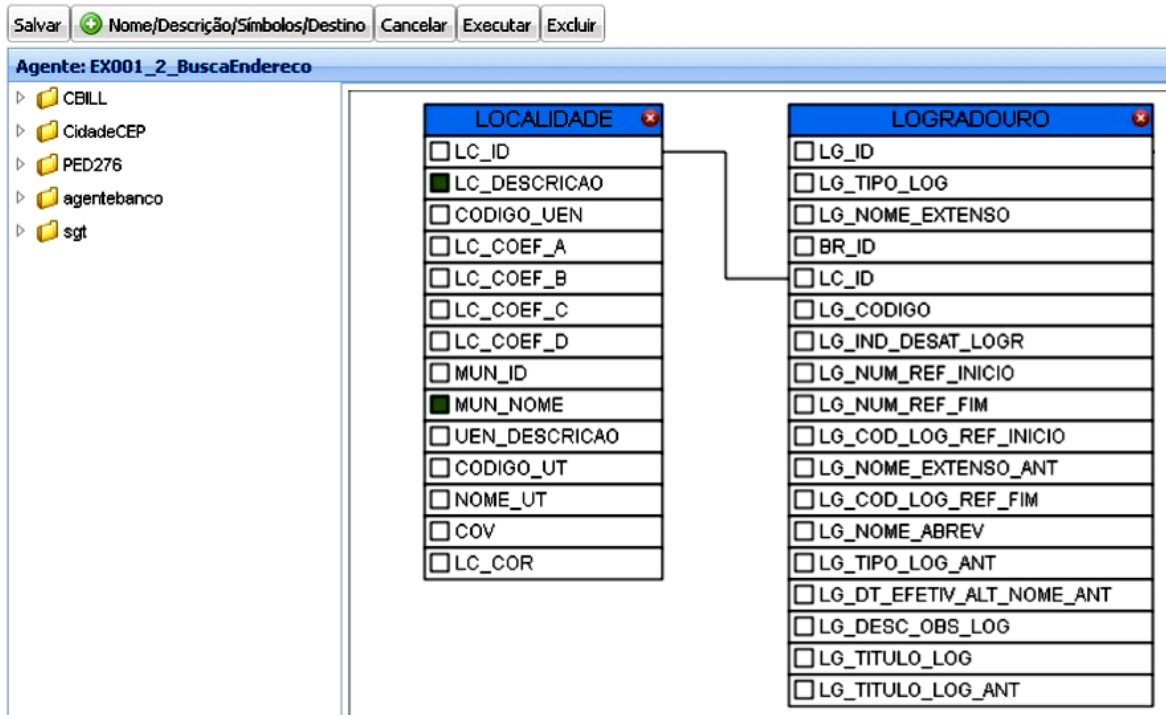


Figura 4.1 - Criação de um agente extrator com origem de banco de dados

Na Figura 4.2, temos a representação de regras e condições específicas para extração de dados a partir de um banco de dados que é, de certa forma, SQL em seu estado mais simples, entretanto, ainda mantendo a ideia de mínimo esforço cognitivo. No caso específico da figura, por exemplo, o retorno dos dados seria todas as linhas que possuam o nome de localidade igual ao parâmetro {cidadecepunico.cidade} que compõe o símbolo que ativou o agente atualizador.

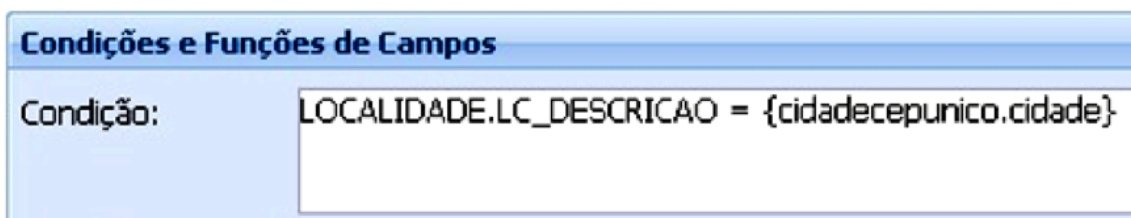


Figura 4.2 - Definição de condições para extração

A Figura 4.3 mostra que um tratamento de formatação dos dados pode ser feito logo após a extração da informação. A função que realiza essa formatação pode já estar pronto em uma biblioteca do ambiente ou pode ser criada (programada) pelo próprio usuário. Por exemplo: se o agente extrator de dados traz CPFs de clientes sem máscara de formatação (pontos e traço), o próprio agente pode incluir essa máscara de formatação, padronizando os dados a serem apresentados pelo processo de extração.

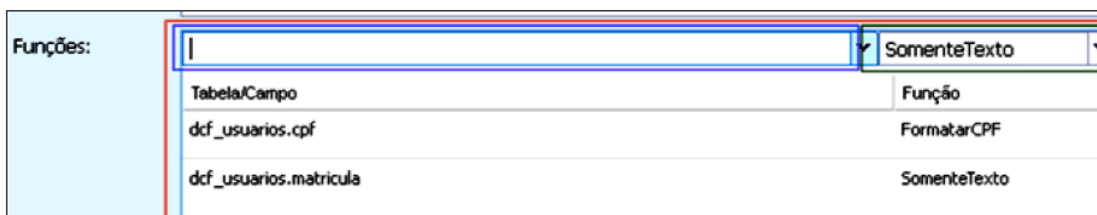


Figura 4.3 - Funções de tratamento e formatação dos dados extraídos

É importante lembrar que a criação dos agentes tem seu foco no usuário com baixo conhecimento de linguagens de programação, entretanto, o ambiente é flexível o suficiente para atender também aos que possuem tal conhecimento. Desta forma, agentes podem ser criados e melhorados com o tempo, atendendo aos problemas dos mais diversos níveis de complexidade.

#### B. Criação de um agente atualizador

Mais adiante, representado na Figura 4.4, temos a criação de um agente atualizador e que possui interface semelhante à do agente extrator, inclusive na forma de representar as condições da consulta. Na figura, por exemplo, os dados de localização de um consumidor serão atualizados no banco de dados.

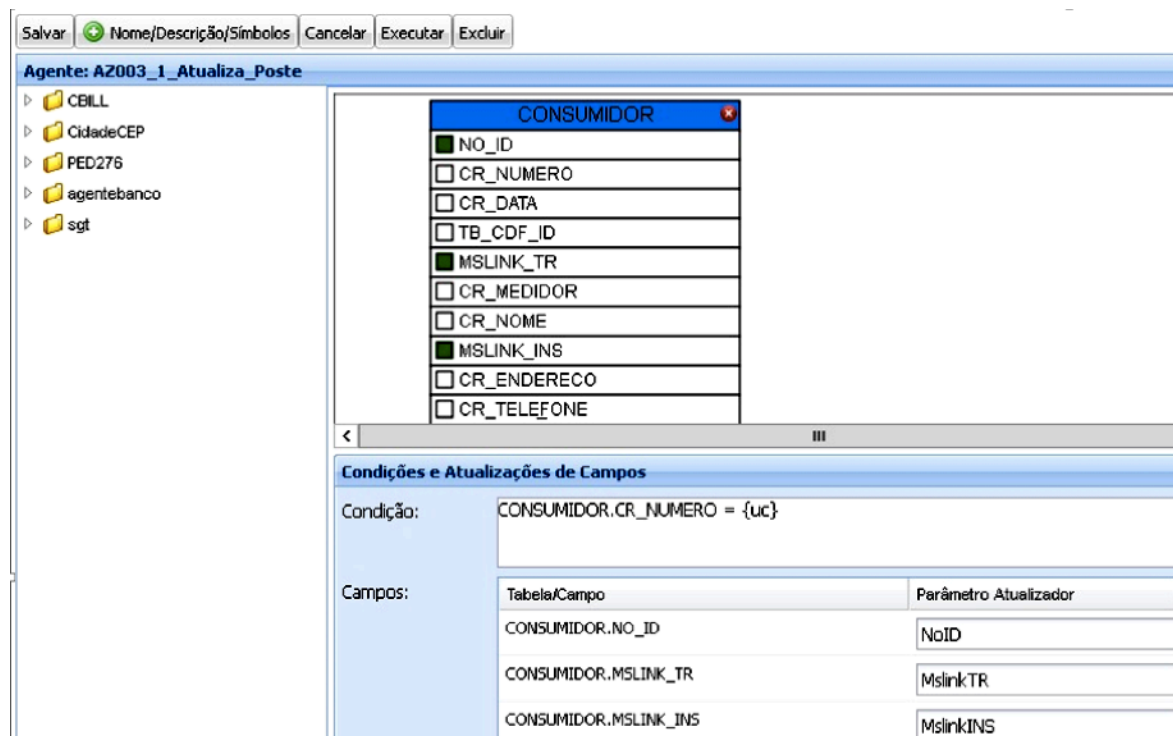


Figura 4.4 - Representação de parâmetros para um agente atualizador

Entretanto, por se tratar de uma atualização no banco de dados, é necessário informar quais campos serão atualizados e de qual tabela (Tabela/Campo) e, também, em qual

parâmetro do símbolo está contida a informação utilizada nessa ação de atualização (Parâmetro Atualizador). O agente atualizador, portanto, atualiza os campos escolhidos com os dados obtidos e armazenados no símbolo de acordo com a configuração feita pelo usuário. Esse tipo de configuração permite alta flexibilidade no processo de efetivação dos resultados da solução.

### C. Criação de um agente atuador

Após definir os dados básicos do agente atuador, como o nome, o próximo passo é criar a sua base de conhecimento. Inicia-se este processo pela estrutura hierárquica de estados que irá provocar a execução de ações que define o comportamento do agente. A Figura 4.5 mostra a interface criada para estruturar hierarquicamente os estados de atuação dos agentes. Após um processo interativo de arrastar e soltar o “Retângulo Branco” do Menu, no canto superior esquerdo, o agente o sistema irá criar o estado.

A sobreposição de estados terá como objeto final um conjunto de regras SE...ENTÃO como descrito neste trabalho. Desta forma, fica abstraída a necessidade de conhecimento de uma linguagem de programação de alto nível na criação de um contexto e execução de regras.

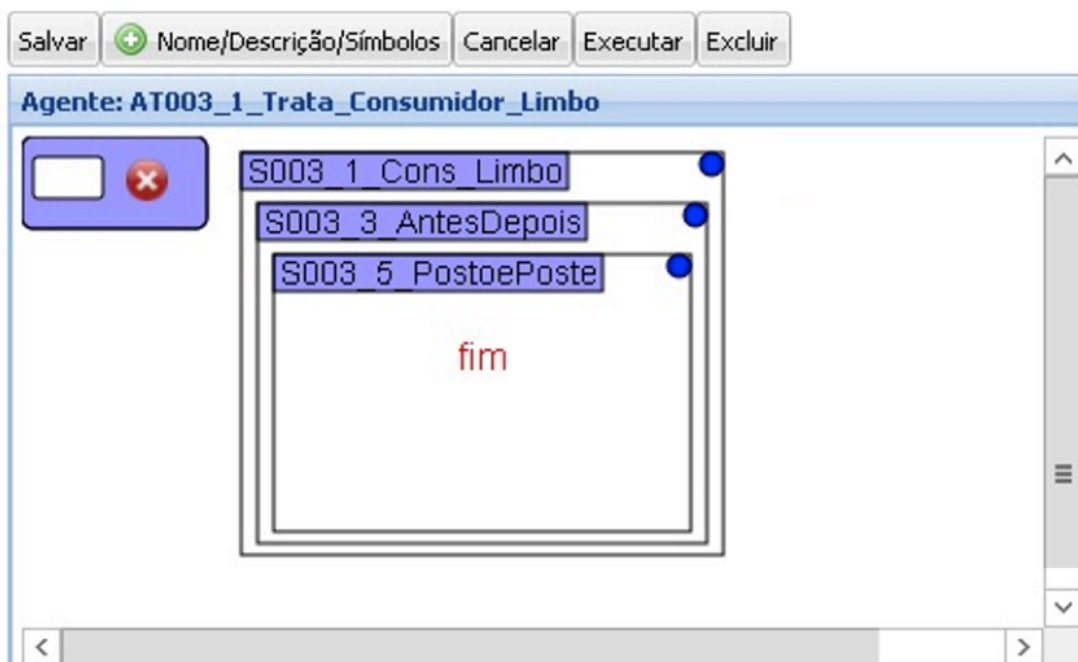


Figura 4.5 - Criação e visualização dos contextos em um agente atuador

No entanto, caso o usuário possua conhecimento na linguagem de alto nível Java, o mesmo poderá aplicá-lo, visto que, todos os agentes aceitam comandos nessa linguagem para tratamento das informações.

### 4.1.3 REPRESENTAÇÃO GRÁFICA DO MURAL

As várias soluções são executadas, bem como agentes exercem suas funções e símbolos são publicados, entretanto, todo esse processo ocorre de forma oculta e automática. No entanto, tem-se o Mural, que serve como área para monitoramento das atividades de todos os processos sendo executados pelo ambiente multiagente.

Agentes				
Extratores				
Nome	Estado	Execução e Tempo		
<a href="#">Extrator X</a>	Inativo	-		
<a href="#">Extrator A2</a>	Inativo	-		
<a href="#">Extrator com Solução X2 ca</a>	Parado	-		
<a href="#">Atualizador com solução X cad</a>	Inativo	-		

Símbolos				
Nome	Agente Publicador	Parâmetros	Agente Executando	
Símbolo A	<a href="#">Agente X1</a>	[A, B, C]	<a href="#">Agente X1</a>	1
Símbolo B	<a href="#">Agente A2</a>	[A, B, C]	<a href="#">Agente A2</a>	1

Figura 4.6 - A interface do Mural

O mesmo está representado na Figura 4.6, onde é possível visualizar os agentes extratores criados e seus respectivos estados (ativo, inativo, em execução ou parado).

Detalhes do Agente	
Nome:	Extrator X
Estado:	Inativo
Tipo:	Extrator
Descrição:	<div style="border: 1px solid gray; padding: 2px;">           Descrição do Extrator X         </div>
Solução:	Solução X
Execução e Tempo:	
Símbolos Iniciadores:	SímboloX
Símbolos de Saída:	SímboloSaidaA;SímboloSaidaB
Temporizador:	50min
Ativo:	Não

Figura 4.7 - Detalhes de um agente específico

Da mesma forma, é possível visualizar os demais agentes atualizadores e atuadores. Por fim, tem-se, também, os símbolos que estão publicados em determinado momento e acionam determinados agentes. Se necessário, o usuário pode ver informações específicas de um determinado agente a fim de entendê-lo melhor, como se pode observar na Figura 4.7.

Todas essas informações são atualizadas constantemente de forma a garantir um controle sobre todas as ações que estão sendo executadas pelo ambiente multiagente.

Portanto, é possível observar que a aplicação possui uma interface simples e limpa, tendo sempre o foco na interatividade e abstração de conhecimentos de linguagens e assuntos demasiado complexos para os não pertencentes à área de Tecnologia da Informação. Porém, não foi retirada a possibilidade de utilização pelos que possuem tais conhecimentos graças à flexibilidade do ambiente.

## **4.2 APLICABILIDADE DO AMBIENTE**

A ideia estruturada e detalhada nas seções anteriores viabiliza a construção de uma aplicação capaz de criar a base de dados para agentes inteligentes. Nesta seção são apresentados os resultados da implementação de aplicações criadas para a Celg D, algumas das quais foram descobertas e desenvolvidas pelos próprios colaboradores da empresa.

É importante observar que, nas soluções apresentadas a seguir, o refinamento e a melhora na qualidade das informações do Cadastro Técnico representa, de fato, economia de recursos financeiros, humanos e de tempo para a Celg D. Essa economia acontece, pois, ocorre uma diminuição de retrabalho e uma melhor gestão do conhecimento. Além disso, informações corretas acerca de metragens e localização de clientes e equipamentos são importantes e necessárias segundo as normas da ANEEL (ANEEL, 2004).

### *A. Localidades com CEP único*

O Cadastro Técnico enfrenta problemas de informações incorretas. Dentre esses problemas, destaca-se o do CEP único, que é um problema enfrentado tanto por consumidores da capital como de cidades do interior. Na data de apresentação da solução, existiam cerca de 90 mil endereços cadastrados com CEP 70.000-00. Esse problema ocasiona a devolução de milhares de objetos postais à empresa todos os meses, gerando custo financeiro, de pessoal e armazenamento no processo de reenvio.

A fim de corrigir o problema, foi criada uma solução no ambiente multiagente que executa a correção dos CEPs dos clientes. Tendo em vista que outros sistemas da empresa dispõem de endereço do cliente, percebeu-se que é possível fazer correção no Cadastro

Técnico por meio deles. Essa solução dispõe de quatro agentes, sendo dois agentes extratores, um agente atuador e um agente atualizador.

Como é possível observar na Figura 4.8, temos um agente extrator *EX001\_1\_Busca\_local* que busca as localidades com CEP único e as coloca no mural por meio do símbolo *S001\_1\_Localidade*. É ativado, então, o agente atuador *AT001\_1\_Trata\_Cep\_Unico* que trata as informações recebidas. O agente extrator *EXT001\_2\_Busca\_Endereco*, ativado pelo símbolo *S001\_2\_Pede\_End\_Local*, busca o endereço de localização de cada consumidor em bancos de dados de outros sistemas da empresa e os devolve ao Mural pelo símbolo *S001\_3\_Endereco*. O agente *AT001\_1\_Trata\_Cep\_Unico* é novamente ativado e envia o CEP correto pelo símbolo *S001\_4\_CEP\_Local* para que o agente atualizador *AZ001\_1\_Atualiza\_CEP* faça a correção do CEP no banco de dados do Cadastro Técnico.

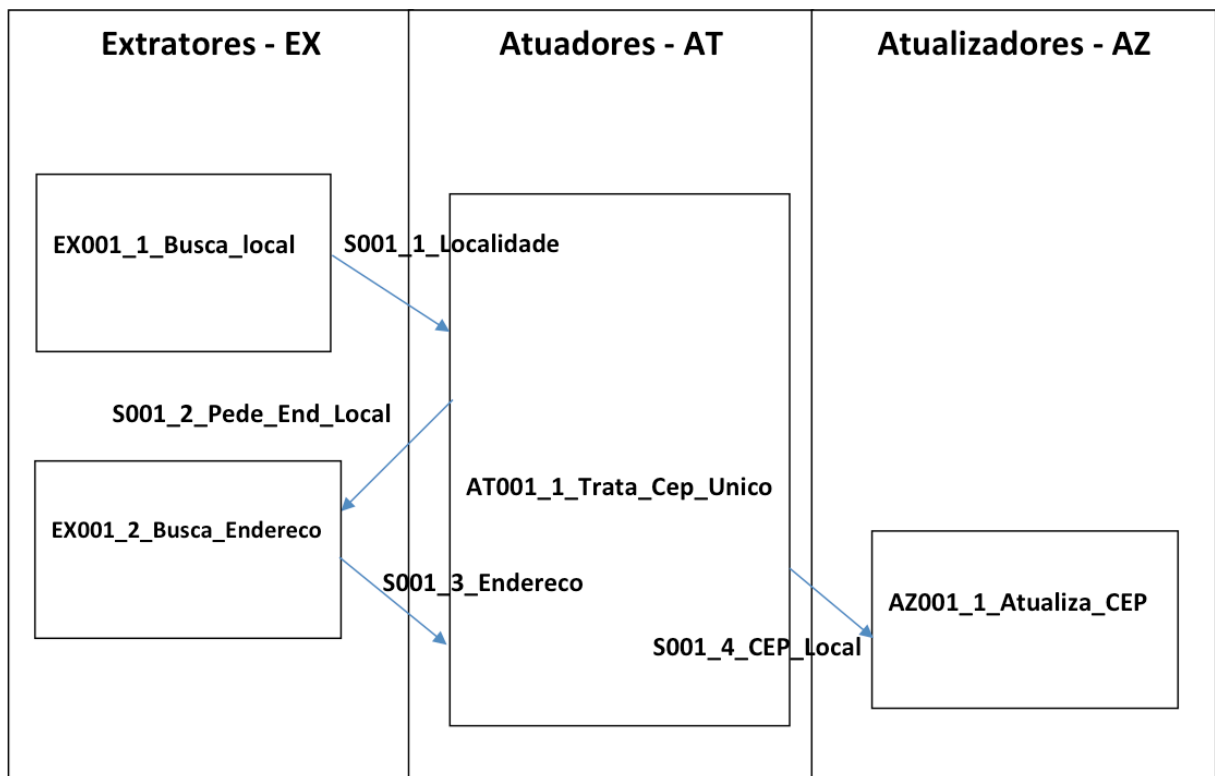


Figura 4.8 - Agentes utilizados na solução para CEP único

Em sua primeira execução, a solução foi capaz de resolver cerca de 80 mil casos de clientes com CEP único, ou seja, aproximadamente 90% dos casos do Cadastro Técnico. Esse número de casos resolvidos representa uma melhora significativa na consistência e confiabilidade dos dados do Cadastro Técnico, visto que, a correção de CEPs únicos pode contribuir para a eficiência e eficácia no planejamento de redes, bem como na manutenção da

rede elétrica nos endereços corrigidos. De fato, a solução mostrou-se eficiente para esse problema.

Essa solução possui certos desafios a serem explorados. Os colaboradores buscam expandi-la a fim de corrigir, também, nomes de bairros e ruas que tenham sido inseridos incorretamente. Por exemplo: tem-se uma rua de nome correto "Juscelino Kubitschek", enquanto, no sistema, existem endereços cadastrados com nome "Jucelino Kubitchek", "Jucelino Kubixek" etc. A correção e unificação de nomes de ruas e bairros pode ser feita e já é estudada pelos colaboradores.

### B. Correção dos trechos de rede

O setor de Geoprocessamento fornece a clientes internos e externos vários relatórios dos ativos da Celg D cadastrados no banco de dados do Sistema de Gestão Técnica (SGT). Entretanto, foi detectado que a informação do trecho de rede no sistema, que fornece a quilometragem de rede de baixa e média tensão, estava divergindo da apresentada em campo.

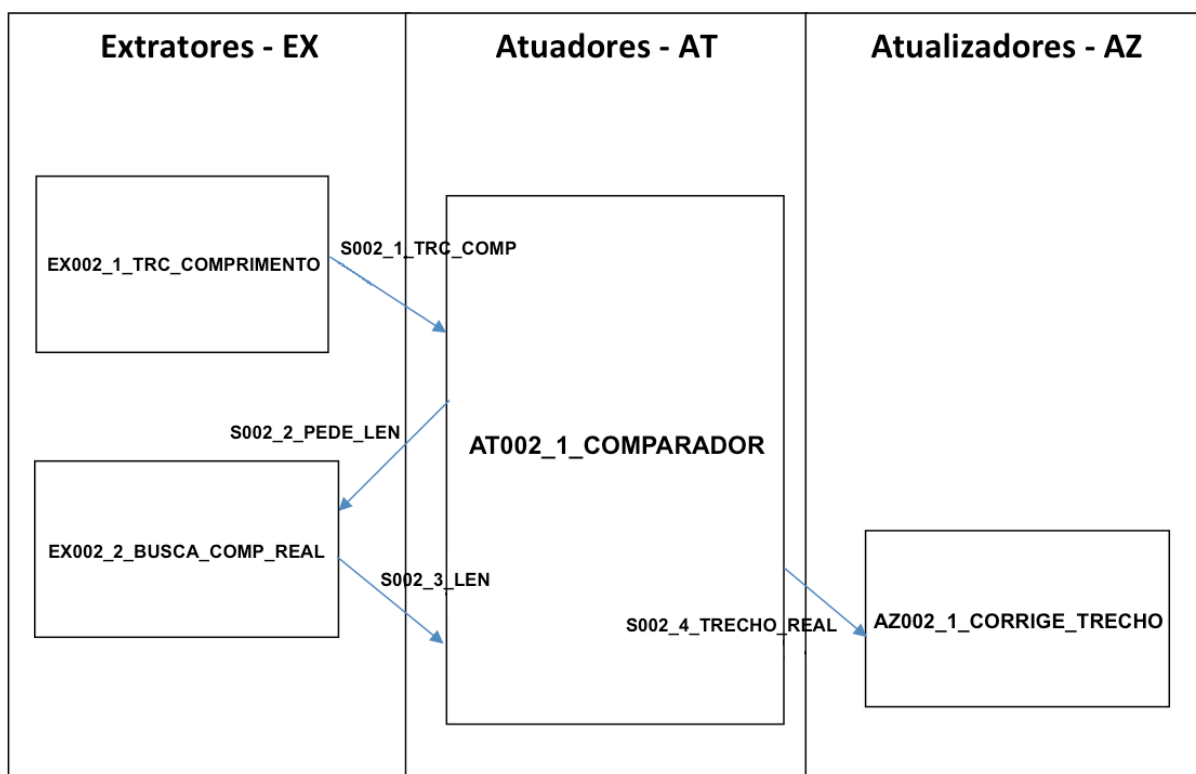


Figura 4.9 - Agentes utilizados na solução de correção dos trechos de rede

Para solucionar o problema nas extrações de dados foi utilizada uma tabela especial que apresenta o valor correto. Entretanto, mesmo após feita a correção via banco de dados, o erro voltou a aparecer no campo original causando problemas, especialmente, no módulo de

projetos do SGT (SGT-PROJ), sendo esse erro originado por algum dos sistemas de informação da Celg D.

Foi verificado que a medida incorreta do comprimento do trecho de rede pode ocasionar problemas na confiabilidade dos projetos realizados pelo SGJ-PROJ, bem como comprometer a qualidade dos relatórios tabulares e arquivos de banco de dados espaciais enviados para a ANEEL / AGR (ANEEL, 2004).

Antes da execução da solução, o sistema dispunha das seguintes informações:

- a) Redes de Baixa tensão (sem o neutro): 71.065,83 km
- b) Redes de Baixa tensão (apenas o neutro): 155.675,65 km
- c) Redes de Média tensão – 13,8kV: 164.283,12 km
- d) Redes de Média tensão – 34,5kV: 52.001,57 km

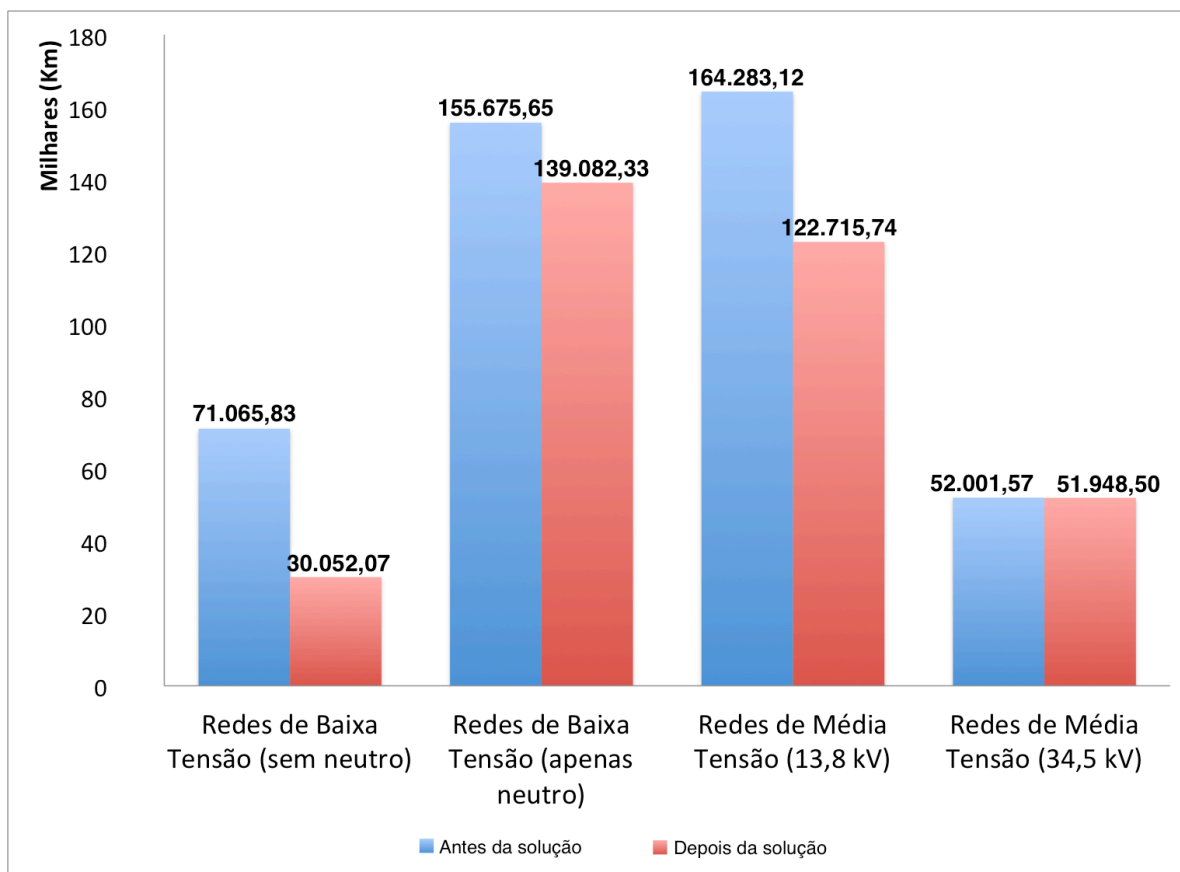


Figura 4.10 - Gráfico representando a mudança após a correção dos trechos de rede

A solução criada possui quatro agentes, sendo dois agentes extratores, um agente atuador e um agente atualizador. É possível observar na Figura 4.9 que um agente extrator

*EX002\_1\_TRC\_COMPRIMENTO* inicia o processo. Esse agente é responsável por buscar o trecho de rede incorreto e devolvê-lo ao mural por meio do símbolo *S002\_1\_TRC\_COMP*. Em seguida, o agente atuador *AT002\_1\_COMPARADOR* solicita o comprimento real do trecho de rede ao agente extrator *EX002\_2\_BUSCA\_COMP\_REAL* por meio do símbolo *S002\_2\_PEDE\_LEN*. Então, é novamente acionado o agente *AT002\_1\_COMPARADOR*, pelo símbolo *S002\_3\_LEN*, que verifica se os trechos são inconsistentes. Caso sejam, uma correção do comprimento de rede é solicitada ao agente atualizador *AZ002\_1\_CORRIGE\_TRECHO* pelo símbolo *S002\_4\_TRECHO\_REAL*.

Após a execução da solução, os comprimentos reais recolhidos após correção foram:

- a) Redes de Baixa tensão (sem o neutro): 30.052,07 km
- b) Redes de Baixa tensão (apenas o neutro): 139.082,33 km
- c) Redes de Média tensão – 13,8kV: 122.715,74 km
- d) Redes de Média tensão – 34,5kV: 51.948,50 km

Finalmente, por meio do gráfico mostrado na Figura 4.10, é possível observar uma redução de mais de 41 mil quilômetros nas redes de baixa tensão sem o neutro; mais de 16 mil quilômetros nas redes de baixa tensão apenas o neutro; mais de 41 mil quilômetros nas redes de média tensão de 13,8 kV; e de 56 quilômetros em redes de média tensão de 34,5 kV. Portanto, a correção de erros para as redes de baixa tensão (com e apenas neutro) e a de média tensão de 13,8 kV foi bastante significativa, garantindo mais confiabilidade nos dados georreferenciados do Cadastro Técnico.

### *C. Consumidores sem localização geográfica*

O Cadastro Técnico é extenso e enfrenta um problema de consumidores que não possuem a localização geográfica de seu poste registrada no sistema. Quando ocorrem esses erros as unidades consumidoras são alocadas no “Limbo”, ou seja, uma referência fictícia do Cadastro Técnico que comporta unidades consumidoras que não possuem localização geográfica correta. O Limbo é uma subestação criada unicamente para comportar unidades consumidoras que não possuem localização geográfica correta. Para que um cliente apareça corretamente no sistema, ele precisa estar conectado, preferencialmente, a um poste e, no mínimo, a um posto transformador.

O acúmulo desses erros provocou, ao longo dos anos, o envio de milhares de clientes de todo o estado ao Limbo. Ao analisar um a um, percebe-se que não existe um padrão para os resultados, variando de clientes situados desde a capital até a zona rural.

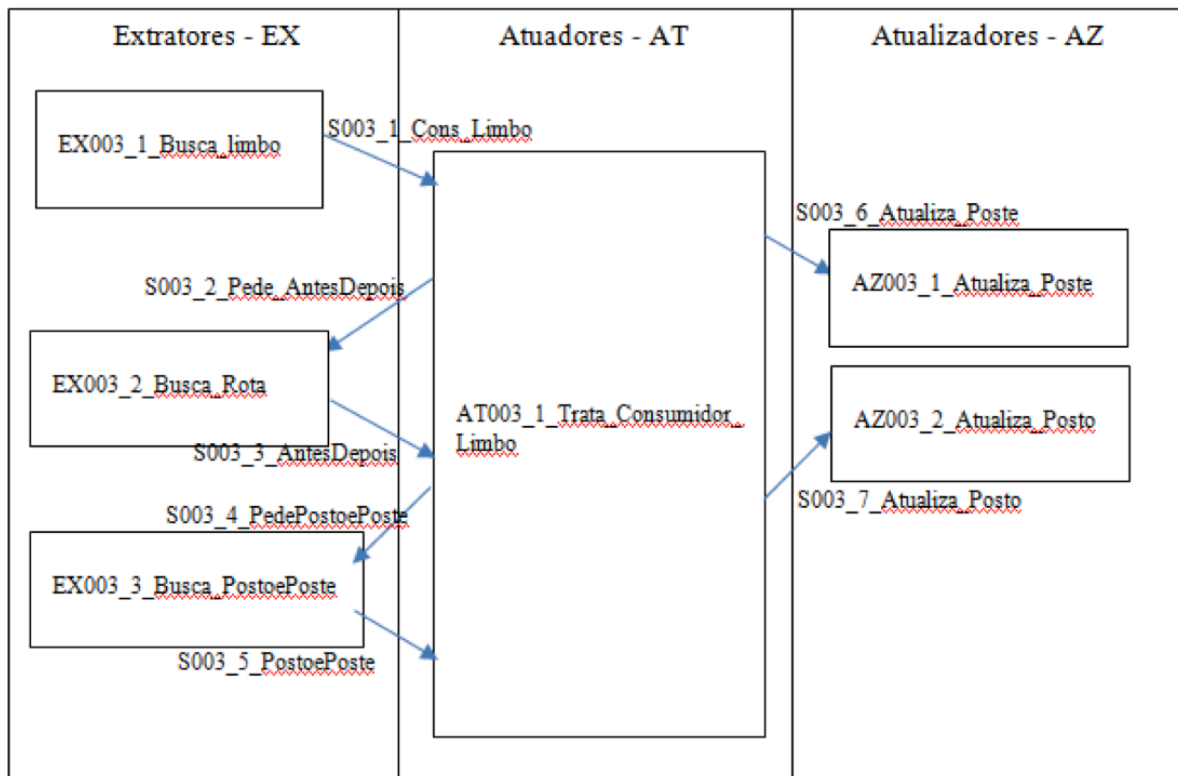


Figura 4.11 - Agentes utilizados na solução dos consumidores sem localização geográfica

Entretanto, sabe-se que no sistema comercial, por meio 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 faz as seguintes tarefas:

- a) Relaciona todos os consumidores sem posicionamento geográfico. Expõe no Mural cada um desses consumidores, um a um;
- b) Para cada consumidor exposto determina o consumidor anterior e posterior a partir da tabela de rota do sistema comercial;
- c) Encontra os dados do poste e posto a que os consumidores anterior e posterior estão ligados;
- d) Avalia se os três consumidores estão no mesmo poste, e, se estiverem, atualiza o posicionamento do consumidor avaliado.

Esta solução possui, no total, seis agentes, sendo três agentes extratores, um atuador e dois atualizadores.

Observando a Figura 4.11, percebe-se que o agente extrator de nome *EX003\_1\_Busca\_Limbo* executa uma busca de todos os consumidores no Limbo e envia o símbolo *S003\_1\_Cons\_Limbo* para o Mural. O agente atuador *AT003\_1\_Trata\_Consum\_Limbo* é acionado por esse símbolo e executa sua função, que é receber esses consumidores, um de cada vez. Por meio do símbolo *S003\_2\_Pede\_AntesDepois*, é iniciado um segundo agente extrator *EX003\_2\_Busca\_Rota* que tem por função executar uma consulta que retorna os consumidores anterior e posterior (ao consumidor do Limbo) de acordo com a rota de leitura. Em seguida, é acionado o agente *AT003\_1\_Trata\_Consum\_Limbo* novamente pelo símbolo *S003\_3\_AntesDepois*. Nesse momento, os consumidores anterior e posterior são tratados, seja verificando se existem, ou fazendo formatações nas informações. Feito isso, o agente *EX003\_3\_Busca\_PostoePoste*, acionado pelo símbolo *S003\_4\_PedePostoePoste*, é ativado e busca o poste e posto transformador de ambos os consumidores. O retorno é enviado pelo símbolo *S003\_5\_PostoePoste* que reativa o agente *AT003\_1\_Trata\_Consumidor\_Limbo*. Com os postes e postos transformadores, é feita a verificação se ambos são idênticos. Por fim, sabendo quais informações foram obtidas após a verificação, é feita a atualização do consumidor do Limbo com o novo poste e/ou posto transformador. Essa atualização é feita, pelos agentes *AZ003\_1\_Atualiza\_Poste*, acionado por *S003\_6\_Atualiza\_Poste*, e *AZ003\_2\_Atualiza\_Posto*, acionado pelo símbolo *S003\_7\_Atualiza\_Posto*.



Figura 4.12 - Agente atuador AT003\_1\_Trata\_Consumidor\_Limbo

Dentro do ambiente, a representação gráfica da solução por meio de contextos no agente atuador é representada na Figura 4.12. Como é possível observar, a estrutura final é simples de ser lida e possui contextos de fácil entendimento. Desta forma, pode-se constatar que uma solução relativamente complexa transformou-se em um esquema mais acessível a usuários com menor conhecimento acerca de sistemas multiagentes. De acordo com a lógica desenvolvida na estruturação de contextos, os agentes extratores e atualizadores são criados e seus estados são encadeados até o estado de repouso (fim) do agente atuador.

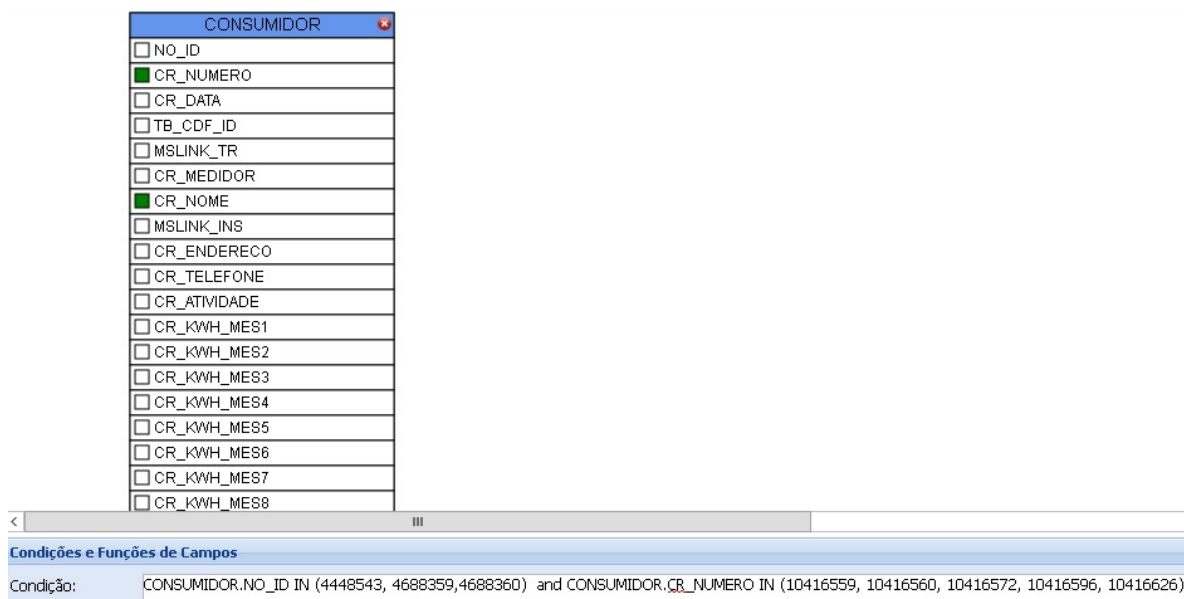


Figura 4.13 - Agente extrator EX003\_2\_Busca\_Rota

Como mostra a Figura 4.13, a extração torna-se um processo visual que exige menos desenvolvimento utilizando linguagem SQL.

O agente atualizador utilizado é apresentado na Figura 4.14 e mostra, novamente por meio visual, a pouca utilização de SQL no desenvolvimento da solução. Nesse caso, a atualização ocorre diretamente no banco de dados de acordo com os campos selecionados pelo usuário de acordo com a condição apresentada no campo "Condição".

Na data de aplicação inicial da solução, a subestação denominada Limbo possuía 16.187 unidades consumidoras. Após a aplicação da solução, esse número foi reduzido para 9.382, ou seja, 43% de redução em sua primeira execução. Isso representa uma melhora significativa nos dados georreferenciados e garante, dentre outras coisas, a eficiência no atendimento de casos de manutenção na rede elétrica nas proximidades de todas as 6.805 unidades consumidoras corrigidas pela solução.

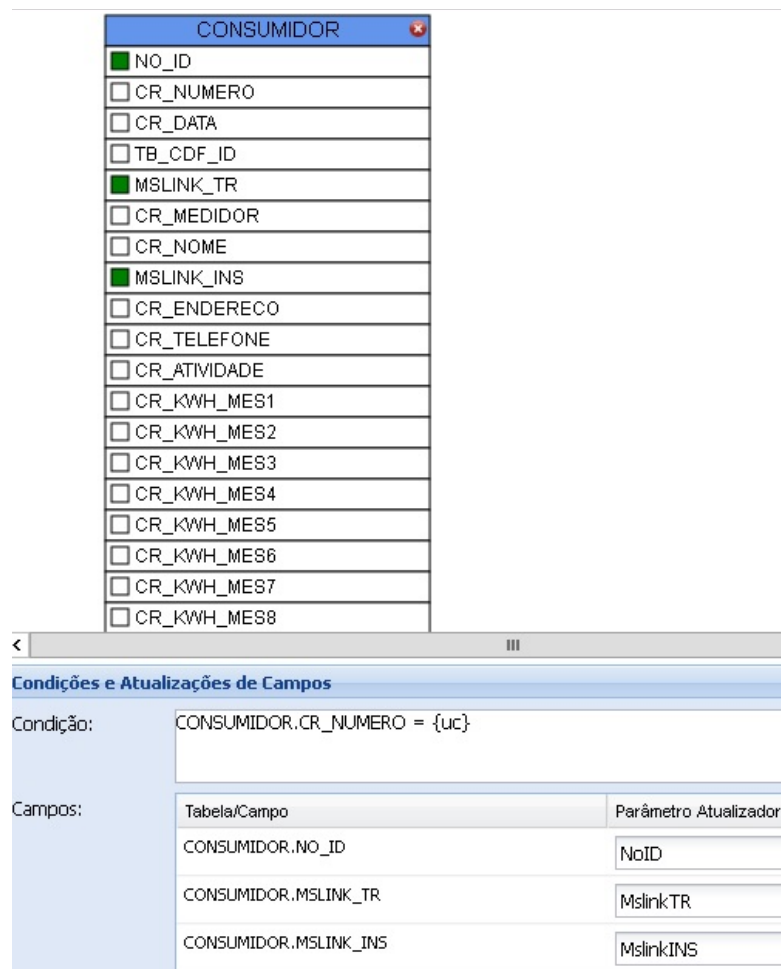


Figura 4.14 - Agente atualizador AZ003\_1\_Atualiza\_Poste

Portanto, é possível observar que após descobertos os problemas e aplicados no ambiente, as soluções apresentaram resultados satisfatórios quanto a correção dos dados do banco de dados original a partir de outras base de dados.

Durante a aplicação dos conhecimentos, os colaboradores demonstraram domínio do ambiente e facilidade na criação das soluções. De fato, a interface gráfica conseguiu extrair grande parte da dificuldade inerente ao desenvolvimento de uma lógica de pensamento para execução de cada agente de extração e atualização. Fica demonstrada pela Figura 4.12 a simplificação de um problema complexo dentro do departamento de georreferenciamento da Celg D e o sucesso na resolução do mesmo.

### 4.3 PUBLICAÇÕES

Como fruto deste trabalho, é possível destacar, também, as publicações em congressos e conferências, sendo que, ao todo, destacam-se cinco, sendo uma delas premiada. As mesmas são listadas a seguir e os textos na íntegra podem ser lidos nos Anexos deste trabalho.

- "Uma Proposta de Utilização de um Ambiente Multiagente Dinâmico para Atualização de Dados Técnicos em Distribuidoras de Energia" na XI Conferência de Estudos em Engenharia Elétrica (CEEL) em Uberlândia (MG).
- "Vinculação de Unidades Consumidoras ao Cadastro Técnico Georreferenciado Utilizando Otimização Geográfica: Aplicação de Agentes Inteligentes e Lógica Fuzzy" na XII Conferência de Estudos em Engenharia Elétrica (CEEL) em Uberlândia (MG).
- "Aplicando Conhecimento Mapeado por Meio de Diagrama de Estados em Agentes Inteligentes para Refinamento de Banco de Dados Georreferenciado" no XLII Congresso Brasileiro de Educação em Engenharia (COBENGE) em Juiz de Fora (MG).
- "Proposta de Utilização de um Ambiente Multiagente Dinâmico Para Atualização de Dados Técnicos em Distribuidoras de Energia" no XI Congresso de Pesquisa, Ensino e Extensão (CONPEEX) em Goiânia (GO).
- "Vinculação de Unidades Consumidoras ao Cadastro Técnico Georreferenciado Utilizando Otimização Geográfica: Aplicação de Agentes Inteligentes e Lógica Fuzzy" no III Encontro Nacional de Geoprocessamento do Setor Elétrico (ENGEEO) em Foz do Iguaçu (PR), sendo este premiado.

## 5 CONCLUSÕES

Com base no referencial teórico estudado, é evidente que a utilização de ambientes multiagentes tem sido amplamente difundida nos últimos anos a fim de resolver os mais diversos problemas. Diferentes aplicações e modelos tem sido explorados com o intuito de atender as diversidades causadas por esses problemas.

A fim de compreender as dificuldades enfrentadas pelo setor de Geoprocessamento da Celg D em relação ao Cadastro Técnico, foi feita uma análise acerca da estrutura de processos da empresa quanto à utilização de georreferenciamento. Desta forma, foram estudados os sistemas legados, suas formas de integração e, também, quem são as pessoas que alimentam esses sistemas e como o fazem. Por meio dessa análise, este trabalho mostrou uma proposta de estruturação de um ambiente multiagente capaz de solucionar problemas complexos, entretanto, exigindo do usuário um esforço cognitivo reduzido para a estruturação das soluções.

A abordagem proposta visa a solução de problemas e refinamento do Cadastro Técnico, porém, ao fim do projeto foi possível observar que sua forma de construção e flexibilidade extrapolam o objetivo inicial. Após a análise dos sistemas da empresa e início do desenvolvimento do ambiente, constatou-se que o ambiente criado pode ser aplicado a diferentes setores da empresa em diferentes soluções. Nota-se que a flexibilidade de um ambiente multiagente apresenta características importantes que contribuem para a resolução de problemas. Isto se deve, principalmente, à capacidade adaptativa e o desenvolvimento independente de agentes computacionais participantes desse ambiente. Ou seja, a utilização de ferramentas desenvolvidas por meio do conceito multiagente proporciona uma grande adaptabilidade frente a diferentes tipos de problemas que possam surgir. Os mesmos podem ser tratados dentro de um mesmo ambiente desenvolvido não sendo, portanto, necessário realizar novos investimentos em outros tipos de ferramentas à medida que novas demandas surgem.

Uma interface gráfica foi proposta a fim de eliminar parte do esforço cognitivo do usuário. Embora pareça complexa a ideia de contextualizar-se problemas em estados (construção de agentes que se comunicam por símbolos em um mural), este trabalho demonstrou, experimentalmente, que esse modelo de retratação de conhecimento é mais simples de ser operado por usuários pouco experientes do que o modelo de conhecimento

expresso com regras lógicas (por meio de scripts). Essa diminuição na complexidade se dá pelo fato de que os próprios usuários, que são colaboradores da empresa, foram os responsáveis pela idealização e estruturação das soluções no ambiente multiagente.

Durante a implantação do sistema e desenvolvimento de soluções iniciais, ficou evidenciada a capacidade do mesmo de se adaptar e receber inteligência híbrida. Em testes iniciais, soluções de teste simples utilizando lógica *Fuzzy* foram criadas a fim de provar isso. Entretanto, soluções mais complexas utilizando redes neurais artificiais, algoritmos genéticos ou modelos de computação evolutiva podem exigir alterações no ambiente para bom comportamento. Tais alterações podem fazer parte de trabalhos futuros a fim de melhorar o ambiente atual.

Por meio de uma avaliação dos recursos do ambiente após o seu desenvolvimento, foi possível constatar que a estrutura proposta utilizando três tipos de agentes e um mural atendeu às necessidades detectadas durante o processo de análise de problemas. A interface gráfica criada, por seguir os protótipos apresentados neste trabalho, apresentou os elementos necessários para facilitar a criação de soluções. A apresentação de um conjunto de soluções desenvolvidas pelos colaboradores da empresa Celg D demonstrou uma maior capacidade de abstração dos mesmos para com problemas complexos ao utilizarem um modelo gráfico de representação de estados.

Ainda no âmbito dos resultados obtidos, acredita-se que as publicações feitas com base neste trabalho também são resultados positivos e comprovam que a abordagem proposta inova e contribui.

Por fim, tendo proposto e desenvolvido o modelo, após a implantação e utilização do mesmo pelos colaboradores da Celg D, acredita-se que os resultados obtidos atenderam às expectativas e os objetivos traçados neste trabalho foram alcançados.

## 6 REFERÊNCIAS

- ANDREADIS, G. et al. *Classification and Review of Multi-Agents Systems in the Manufacturing Section*, Procedia Engineering 69, pp. 282–290, 2014.
- ANEEL. Agência Nacional de Energia Elétrica. Resolução Normativa N° 63, 12 de maio de 2004. Disponível em: <http://www.aneel.gov.br/cedoc/ren2004063.pdf>, Acesso em: 06 jan. 2014.
- BELLIFEMINE, F.; CAIRE, G.; GREENWOOD, D. *Developing Multi-Agent Systems with JADE*, 1st Ed., Wiley, 2007.
- BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDERMIR, T. B. *Fundamentos de redes neurais artificiais*. NCE/UFRJ, Rio de Janeiro, 1998.
- CARVALHO, S. D. *Modelo Híbrido de Sistema Tutor Inteligente Utilizando Conhecimento do Especialista e Mapas de Kohonen com Treinamento Automatizado*. 127 f. Tese (Doutorado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, Minas Gerais. 2012.
- CELG D. Goiás, Celg Distribuição S.A. Contém informações institucionais. Disponível em: <https://www.celg.com.br/paginas/institucional/institucional.aspx>. Acesso em: 18 jan. 2015.
- CORKILL, D. D. *Blackboard Systems*, AI Expert 6(9), pp. 40-47, September 1991.
- CRAIG, I. *Blackboard Systems*, Intellect Books, New Jersey, United States of America, pp. 3-9, 1995.
- D'INVERNO, M.; LUCK, M. *Understanding Agent Systems*, 2nd Ed., Springer, 2004.
- DEITEL, P.; DEITEL, H. *Java How To Program*, 9th Ed., DEITEL, 2011.
- FERBER, J.; GASSER, L. *Intelligence artificielle distribuée*, In: International Workshop on Expert Systems & Their Applications, 10, Avignon. Cours n. 9. France, 1991.
- FILHO, P. B.; JÚNIOR, A. H. *Fundamentos de Informática - Lógica Para Computação*, 1ª ed., LTC, 2013.
- FOWLER, C.; HAMMELL, R.J. *A Hybrid Intelligence/Multi-Agent System Approach for Mining Information Assurance Data*, in Proceedings of the 9th ACIS International Conference on Software Engineering, Research, Management, & Applications (SERA 2011), pp. 169-170, Baltimore, MD, pp. 10-12, August 2011.
- GASSER, L. *Boundaries, identity and aggregation: plurality issues in multiagent systems*, In: WERNER, E.; DEMAZEAU, Y. (Eds.). *Decentralized AI 3*. Amsterdam: North-Holland, 1992.
- GHOSH, A.; NAFALSKI, A.; TWEEDALE, W. J. *A Hybrid Intelligent Autonomous Model Developed Using Multi-Agent Systems*, International Journal of Electronics and Electrical Engineering Vol. 1, No. 1, March 2013.
- HAREL, D. *On Visual Formalisms*, Communication of the ACM, vol. 31, n° 5, pp. 514-530, 1988.

- HAREL, D. *Statecharts: A Visual Formalism for Complex Systems*, Science of Computer Programming, vol. 8, issue 3, pp. 231-274, North-Holland, 1987.
- HAYES-ROTH, F.; WATERMAN, D.; LENAT, D. B. *Building Expert Systems*, Addison-Wesley, Boston, MA, USA, 1983.
- HAYKIN, S. *Neural Networks and Learning Machines*, Pearson, 3rd Ed., Hamilton, Ontario, Canada, 2009.
- HILAIRE, V. et al. *Formal Specification and Prototyping of Multi-agent Systems*, Proceedings of the First International Workshop on Engineering Societies in the Agent World, pp. 114-127, 2000.
- HUGET, M. *Communication in Multiagent Systems: Agent Communication Languages and Policies*, Springer-Verlag Berlin Heidelberg, 1st Ed., Germany, 2003.
- JACOB, R. J. K. *Using formal specifications in the design of a human-computer interface*, Communications of the ACM, Vol. 26, n° 4, pp. 259-264, 1983.
- JENNINGS, N. R.; SYCARA, K.; WOOLDRIDGE, M. *A Roadmap of Agent Research and Development*, Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, pp. 7-38, July, 1998.
- KASYANOV, V. N. *Information Visualization on the Base of Hierarchical Graph Models*, Advances in Applied Information Science, Proceedings of the 12th WSEAS International Conference on Applied Informatics and Communications (AIC '12), Istanbul, Turkey, pp. 115-120, August 2012.
- KHOSLA, R. *Engineering Intelligent Hybrid Multi-Agent Systems*, Springer Science, 1st Ed., 1997.
- KOHONEN T. *Self-organizing maps*. Berlin. Springer-Verlag, 2nd Ed., 1997.
- KUGLER, H.; LARJO, A.; HAREL, D. *Biocharts: a visual formalism for complex biological systems*, J. R. Soc. Interface 7(48), pp. 1015-1024, July 2010.
- LARMAN, C. *Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*, Pearson, 3ª Ed., São Paulo, Brasil, pp. 494-495, 2005.
- LEVITAN, I. B.; KACZMAREK, L. K. *The Neuron: Cell and Molecular Biology*, Oxford University Press, 3rd Ed., New York, p. 11, 2001.
- MAALAL, S.; ADDOU, M. *A new approach of designing Multi-Agent Systems: With a practical sample*, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 11, 2011.
- MAIA, W. A. *Percepção & Inteligência Artificial: Conceitos, Considerações e Arquitetura*, 1ª ed., Biblioteca24horas, São Paulo - SP, 2012.
- MCCULLOCH, W. S.; PITTS, W. H. *A logical calculus of the ideas immanent in neural nets*. Bulletin of Mathematical Biophysics, v.5, 1943.
- MENDES, R. D. *Inteligência artificial: sistemas especialistas no gerenciamento da informação*, Ci. Inf., Brasília, v. 26, n. 1, pp. 39-45, jan./abr. 1997.

NAVATHE, S. B.; ELMASRI, R. *Sistemas de Banco de Dados*, 6ª ed., Pearson Education, São Paulo, 2011.

NEGNEVISTKY, M. *Artificial Intelligence: A Guide to Intelligent Systems*, 2nd ed., Pearson Education Limited, England, 2005.

NEWELL, A. *Some problems of the basic organization in problem-solving programs*, Proc. Second Conference on Self-Organizing Systems, pp. 393-423, Spartan Books, United States, 1962.

NEWELL, A. *Physical symbol systems*. Cognitive Science, 4, pp. 135-183, 1980.

NILSSON, J. N. *Principles of Artificial Intelligence*, Morgan Kaufmann, Palo Alto, California, United States of America, pp. 418-428, 1982.

PEŠKOVÁ, K. et al., *Hybrid Multi-Agent System for Metalearning in Data Mining*, CEUR Workshop Proceedings, Vol. 1201, pp. 53-54, 2014.

PRODIST – *Procedimento de Distribuição de Energia Elétrica no Sistema Elétrico Nacional*. Módulo 2. Revisão 3. Brasília, 2011. Disponível em: [http://www.aneel.gov.br/arquivos/PDF/Módulo2\\_Revisao\\_3.pdf](http://www.aneel.gov.br/arquivos/PDF/Módulo2_Revisao_3.pdf), Acesso em: 06 jan. 2014.

RAMAKRISHNAN, R.; GEHRKE, J. *Database Management Systems*, 3rd Ed., McGraw-Hill, 2002.

RICH, E.; KNIGHT, K. *Inteligência Artificial*, Makron Books, 2ª ed., São Paulo, 1993.

ROSENBLATT, F. *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological Review, v. 65, 1958.

RUMELHART, D. E.; HINTON G. E.; WILLIAMS, R. J. *Learning representations by back-propagation errors*. Nature, n. 323 pp. 533-536, 1986.

RUSSEL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*, 3rd Ed., Pearson Education Limited, United States of America, 2010.

SCHALKOFF, R. J. *Intelligent Systems: Principles, Paradigms and Pragmatics*, 1st Ed., Jones & Bartlett Publishers, 2009.

SMITH, G. *The Object-Z Specification Language*, Springer, 1st Ed., United States, 2000.

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, pp. 297-305, 2011.

TEIXEIRA, J. F. *Mentes e Máquinas: uma introdução à ciência cognitiva*, Artes Médicas Sul, 1ª ed., Porto Alegre - RS, pp. 51-66, 1998.

TORSUN, S. *Foundations of Intelligent Knowledge-based systems*, Academic Press, 1st Ed., London, 1995.

WANG, I. K.; ABDULLA, H. W.; SALCIC, Z. *Multi-agent System with Hybrid Intelligence Using Neural Network and Fuzzy Inference Techniques*, IEA/AIE 2007, LNAI 4570, pp. 473-482, 2007.

WATERMAN, D. A. *A Guide to Expert Systems*, Addison-Wesley, 1st Ed., Massachusetts, United States of America, 1986.

WOOLDRIDGE, M. *An Introduction to Multiagent Systems*, 2nd Ed., Wiley, 2009.

WOOLDRIDGE, M.; JENNINGS, N. R. *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review 10(2), pp. 115-152, 1995.

ZHANG, Z.; ZHANG, C. *Agent-Based Hybrid Intelligent Systems*, Springer-Verlag, 1st Ed., Germany, 2004.

## **ANEXOS**

# UMA PROPOSTA DE UTILIZAÇÃO DE UM AMBIENTE MULTIAGENTE DINÂMICO PARA ATUALIZAÇÃO DE DADOS TÉCNICOS EM DISTRIBUIDORAS DE ENERGIA

Kayo F. Pimentel

Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação,  
Goiânia – GO, [kayoken23@gmail.com](mailto:kayoken23@gmail.com)

Sirlon Thiago D. Lacerda

Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação,  
Goiânia – GO, [thiago.dfc@gmail.com](mailto:thiago.dfc@gmail.com)

Thyago C. Marques

Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação,  
Goiânia – GO, [thyagotem@gmail.com](mailto:thyagotem@gmail.com)

Sandrerley R. Pires

Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação,  
Goiânia – GO, [sandrerley@ufg.br](mailto:sandrerley@ufg.br)

**Resumo** - Este trabalho tem como objetivo apresentar uma proposta de utilização de um ambiente multiagente dinâmico para atualização de dados técnicos georreferenciados em Distribuidoras de Energia. Dada a expansão da malha de distribuição causada por um processo contínuo de crescimento de número de consumidores, ocorre uma grande quantidade de atualizações no cadastro técnico, levando a qualidade de suas informações a uma situação em que meras ações administrativas não são suficientes para manter os dados nos níveis de qualidade exigidos pela ANEEL. A arquitetura proposta neste trabalho procura estabelecer um ambiente com agentes inteligentes capazes de vigiar, corrigir e advertir situações em que o processo de atualização do cadastro técnico esteja sendo feito de maneira incorreta. A arquitetura atua também em problemas já existentes, refinando a qualidade do cadastro técnico. Capacidades especiais de acesso a banco de dados, formatação de saídas e raciocínio são dadas aos agentes para que estes sejam capazes de atuar eficientemente no ambiente descrito. Manter o cadastro técnico atualizado e confiável agiliza ações de expansão, manutenção e análise da operação da rede, além de evitar a geração de multas por parte da ANEEL, devido ao fornecimento de informações incorretas.

**Palavras-Chave** - Distribuidoras de Energia, sistemas inteligentes e Sistemas Multiagentes.

## A PROPOSAL TO UTILIZE A DYNAMIC MULTI-AGENT ENVIRONMENT TO UPDATE TECHNICAL DATA IN ENERGY DISTRIBUTOR

**Abstract** - This paper aims to present a proposal for use of a multi-agent environment for dynamic update of technical data georeferenced in Power Distribution. Given the expansion of the distribution network, caused by a continuous growth in the number of consumers, there is a lot of updates in the technical records, taking the quality of your information to a situation in which mere administrative actions are not sufficient to maintain data quality levels required by ANEEL. The architecture proposed in this paper seeks to establish an environment with intelligent agents able to monitor, warn and correct situations where the process of updating the technical registration is being done incorrectly. The architecture also operates in existing problems, refining the quality of technical records. Special capabilities to access the database, format outputs and ability to reason are given to agents so they are able to function effectively in such an environment. Keeping technical records updated and reliable, accelerates expansion actions, maintenance and analysis of network operation and avoids the generation of fines by ANEEL, due to providing incorrect information.

**Keywords** - Power distribution companies, intelligent systems, multi-agent systems.



XI CEEL – ISSN 2178-8308  
25 a 29 de novembro de 2013  
Universidade Federal de Uberlândia – UFU  
Uberlândia – Minas Gerais – Brasil

## I. INTRODUÇÃO

Dentre as diversas atividades de uma Distribuidora de Energia Elétrica, ela é responsável por realizar atividades de planejamento, instalação, manutenção e operação das redes elétricas. E, para melhor desenvolver essas atividades, a qualidade dos dados georreferenciados existentes em seus sistemas é de fundamental importância, a fim de que o que se tem registrado no banco de dados reflita a realidade de campo.

Durante o período de 2004 a 2006, a CELG D (Celg Distribuição S.A.) desenvolveu o banco de dados georreferenciado, denominado de Cadastro Técnico, com informações das redes primárias, secundárias, dos equipamentos e da vinculação das unidades consumidoras à rede. Esse banco de dados é utilizado em praticamente todas as atividades técnicas da distribuição, o que justifica os esforços realizados nos últimos anos, visando sempre aprimorar a qualidade das informações georreferenciadas constantes no Cadastro Técnico.

Dada a extensão da empresa e a abrangência da utilização do Cadastro Técnico, os processos de atualização, que ocorrem atualmente de forma descentralizada (existem algumas iniciativas de centralização do processo, mas sem um processo formal definido até o momento), devem ser rigorosamente controlados, pois, de outra forma, as atualizações podem introduzir erros no banco de dados deteriorando a sua qualidade. Portanto, obter e manter um cadastro com maior representatividade da situação física da rede elétrica é sempre desejado, uma vez que, além de assegurar o estabelecimento de tarifas justas aos consumidores de energia, também mantém a qualidade das informações dentro dos limites estabelecidos pela ANEEL (ANEEL, 2012).

Entretanto, o que se observa ao longo do tempo é o aparecimento de problemas das mais diversas ordens que tendem a prejudicar a qualidade do cadastro técnico da CELG D, onde diversas ações corretivas têm sido feitas, mas ainda assim erros tem se acumulado, podendo levar, em médio prazo, à necessidade de uma ação de levantamento de campo tão, ou mais, trabalhosa e custosa que a realizada na implantação do cadastro.

Para apresentar o conjunto de ações integradas que visam automatizar o processo de refinamento da qualidade do cadastro técnico da CELG D, este trabalho tem como objetivo mostrar a proposta de utilização de um ambiente multiagente dinâmico para atualização de dados técnicos do cadastro da CELG D. Agentes especializados em extrair informações de diversas fontes diferentes, aliado a agentes inteligentes capazes de realizar inferências a partir dos dados trazidos pelos extratores e, finalmente, agentes capazes de atualizar informações no banco de dados do cadastro técnico ou, simplesmente, gerar relatórios

com advertências que apontam prováveis irregularidades do cadastro técnico.

Este ambiente agregará um monitoramento automático, com ações ativas, que irá aproximar ainda mais a realidade desejada para o Cadastro Técnico e, conseqüentemente, uma significativa melhora na qualidade e na confiabilidade do mesmo.

## II. AGENTES INTELIGENTES

A metodologia está apoiada por uma arquitetura de software específica para a estruturação de um Ambiente Multiagente, onde as interações entre agentes inteligentes, dotados de capacidades específicas para atuar no ambiente de bancos de dados, serão capazes de produzir ações consistentes de atualização do cadastro técnico da CELG D.

A construção de um ambiente multiagente e, dentro deste, a criação de agentes inteligentes envolve um conjunto de conhecimentos do usuário que não são triviais como os que são exigidos para operar um sistema de informação tradicional. O desenvolvimento da lógica, a estruturação de um problema para que o mesmo seja tratado por agentes inteligentes, a estruturação de conhecimento através de regras e outras variáveis levam as equipes desenvolvedoras do projeto a encarar um desafio que é a criação de um ambiente capaz de resolver questões complexas, mas com um modelo de operação que seja o mais simples possível, permitindo aos usuários finais desenvolver um conjunto de agentes os quais sejam capazes de solucionar problemas específicos.

Para um melhor entendimento da proposta, alguns conceitos para uma arquitetura multiagente são enunciados abaixo. Maiores informações podem ser vistas em PIRES (2013).

- **Os Ambientes Multiagente:** os ambientes multiagentes se caracterizam pela existência de um contexto de suporte à *sobrevivência* dessas entidades (os agentes inteligentes), permitindo a eles atuarem de forma independente uns dos outros e realizarem tarefas específicas. É possível fazer uma analogia entre o Ambiente Multiagente e uma organização, onde seus funcionários atuam em conjunto, mas cada um desempenhando o seu papel específico é algo perfeitamente válido (GASSER, 1992).

- **Os agentes inteligentes:** são usualmente compostos por uma máquina de inferência que tem a responsabilidade de *pensar*, isto é, relacionar símbolos (dados) de entrada (fatos, eventos etc.), gerando como resultado uma conclusão representada por um ou mais símbolos de saída (ações, advertências etc.) (RICH e KNIGHT, 1993).

A inteligência do agente é estruturada através de um conjunto de regras que norteia o comportamento da máquina de inferência, sendo que essas regras são normalmente representadas através de operadores

lógicos ou eventualmente através da tradicional estrutura das linguagens de programação, o *se-então-senão*. O grau de flexibilidade de um agente é extremo, pois, são as regras armazenadas nele que definem seu comportamento. Essas regras são idealizadas e alimentadas pelo criador do agente (FERBER e GASSER, 1991).

### III. O MODELO PROPOSTO

A Figura 1 ilustra a arquitetura multiagente proposta neste trabalho. Além da tradicional máquina de inferência, o cérebro do agente, o modelo inclui em sua arquitetura de software estruturas marginais para a conversão de estímulos externos em símbolos e vice-versa.

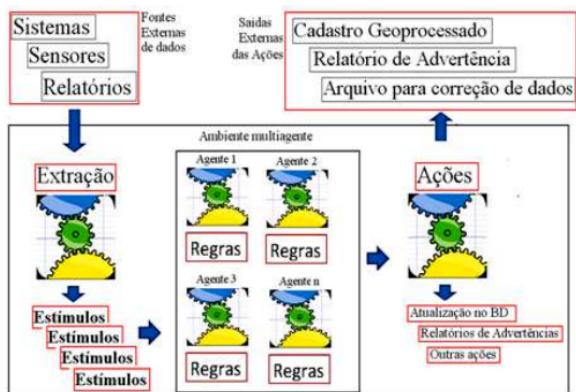


Figura 1 – Visão geral da arquitetura multiagente

São três os tipos de agentes que estarão atuando na estrutura multiagente. Todos os três tipos possuem todas as características de um agente tradicional, sendo que eles se diferenciam apenas pelas suas capacidades propiciadas por suas estruturas marginais. Os tipos de agentes são:

- **Os Agentes Extratores:** tem a responsabilidade de obter informações externas e transformá-las em símbolos no ambiente multiagente. Esses símbolos motivarão a atuação dos demais agentes do ambiente. As possíveis origens de informações que um agente extrator é capaz de acessar são os bancos de dados, planilhas Excel, arquivos de textos não estruturados e *web services*. O agente extrator é capaz de combinar essas diferentes origens em uma mesma operação de busca de informação. A princípio, mas não necessariamente dessa forma, os agentes extratores tendem a terem base de conhecimento limitada, pois, as suas principais ações são mapeadas a partir de acesso externo a informações.
- **Os Agentes Atualizadores:** tem a responsabilidade de externalizar uma decisão inferida pelo ambiente. Essa externalização pode ser feita através de uma atualização em banco de dados ou na criação de arquivos textos, planilhas ou até mesmos relatórios formatados para impressão e análise futura. Outra opção de externalização da saída é a

possibilidade de chamada de um *web service*, dando grande potencialidade à arquitetura, fato descrito mais adiante.

- **Os Agentes Atuadores:** são os agentes que estabelecem as ações que serão tomadas para tratar um determinado símbolo ou um conjunto desses. O conhecimento necessário para resolver o problema, nesta arquitetura proposta, é mapeado no agente atuador. A partir das ações de um atuador, novos extratores podem ser ativados, outros atuadores podem ser chamados a colaborar com o processo de solução e, finalmente, o controle pode ser transferido para um agente atualizador para efetivar o processo de solução do sistema.

Nessa arquitetura, a integração das três estruturas, a de conversão de informação em símbolo (agente extrator), inferências baseadas em regras da base de conhecimento (agente atuador) e conversão de símbolo em ação (agente atualizador), dá ao ambiente de agentes inteligentes a possibilidade de assumir várias configurações, extrapolando até o seu objetivo imediato que é refinar a qualidade do cadastro técnico da CELG D, conforme exemplificado na Tabela 1.

Tabela 1 – Exemplos de produtos

	1) Informações → Símbolo	2) Inferência	3) Símbolo → ação	4) Modelo de Sistema
1	Informações em BD	Regras	Atualização de BD	Refinador de informação
2	Informações em BD	Regras	Relatórios	Auditoria de Informações
3	Sensor	Regras	Atualização de BD	Sistema de vigília
4	Sensor	Regras	Reles	Sistema de Controle
5	Informações em BD	Regras	Relatórios	Sistema de Indicadores

Pode-se observar na Tabela 1 que, para a arquitetura proposta possa atender às exigências do cadastro técnico georreferenciado da CELG D, as funcionalidades de número 1 e 2 constantes na tabela bastariam, isto é, sistemas de auditoria e de integração de informações.

As combinações 3 e 4 levam a diversos tipos de sistemas de tempo real, mais precisamente aos sistemas de controle inteligente, sendo que, para uma empresa de distribuição de energia, sistemas inteligentes de controle da rede seriam plenamente apoiados pela arquitetura proposta.

Essa flexibilidade projetada para a arquitetura multiagente abre espaço para diversas outras aplicações, bastando para isso abranger uma nova área de conhecimento que não o escopo do cadastro georreferenciado. Esse modelo de ambiente de uma arquitetura multiagente, onde cada agente pode atuar de forma individualizada e independente dos demais, se enquadra como uma boa solução para o domínio gradual da complexidade do ambiente. Ou seja, é possível produzir um ambiente de software capaz de estruturar o ambiente multiagente e, a partir daí, passa-se a habitar o ambiente virtual com agentes que resolverão diversos tipos de problemas.

#### IV. O AMBIENTE MULTIAGENTE

Dentro da filosofia que a arquitetura foi criada, o ambiente desenvolvido deve ser capaz de integrar os diferentes sistemas de informação existentes dentro da organização. O motor de funcionamento dessa ferramenta será representado por um conjunto de agentes inteligentes, onde, através de interações entre esses agentes, será possível intercambiar e transformar informações entre sistemas de informação existentes na empresa distribuidora de energia. Diversos serão os tipos de agentes, pois, alguns poderão capturar informações, outros poderão se correlacionar e, finalmente, alguns terão a capacidade de armazenar informações. Para cada um desses tipos de agentes, diversas, também, serão as origens e os destinos para as informações manipuladas por eles.

O ambiente multiagente possuirá as funcionalidades esquematizadas na Figura 2. O extrator de símbolos demonstra como é feita a extração dos dados nos bancos e a conversão em informações passíveis de serem entendidas pela máquina de inferência.

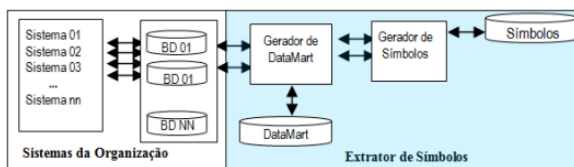


Figura 2 - Contexto do extrator de símbolos

As informações podem estar armazenadas em diferentes bancos de dados de diferentes sistemas internos da empresa e em fontes externas, como, por exemplo, imagens de satélites, planilhas e *web services*. O extrator de símbolos será capaz de buscar essas informações e estruturá-las de forma que a máquina de inferência possa entendê-las e interpretá-las.

Após a realização das inferências, as informações deverão se materializar em ações, que irão depender do processo que está sendo executado pelo agente inteligente. No caso de auditorias, a ação irá resultar em relatórios de auditoria, enquanto, no caso de ajustes de cadastros, serão geradas ações que resultarão em atualizações nos bancos de dados da organização. Extrapolando o contexto do sistema de geoprocessamento, pode-se citar uma saída do sistema como sendo a chamada de um *web service* que poderá exercer alguma ação em um equipamento da rede, levando a arquitetura proposta a atuar como um sistema de controle. Com a atividade de saída, encerra-se o ciclo de atuação do Agente inteligente.

Dentro do contexto do cadastro técnico georreferenciado da CELG D, as várias atualizações irão provocar uma melhoria na qualidade do banco de dados. A Figura 3 mostra o esquema deste recurso,

onde o agente de atualização executa as saídas através de atualizações em banco de dados.

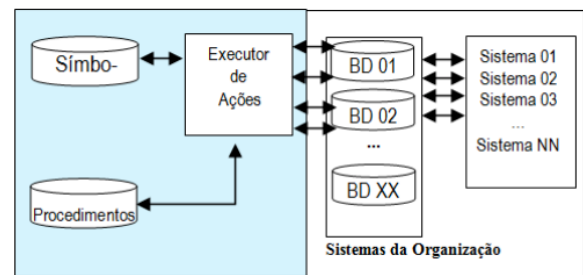


Fig. 3 - Contexto do atualizador

Este ambiente acima descrito tem como uma de suas várias funções potenciais, a capacidade de oferecer um mecanismo genérico de integração entre diferentes sistemas de informações, sincronizando informações entre eles.

#### V. APLICABILIDADE DO AMBIENTE MULTIAGENTE

A aplicabilidade do ambiente Multiagente, aliada ao processo metodológico para construção de Agentes Inteligentes, é definida pelas estruturas marginais à máquina de inferência. Isto é, o processo gerador de símbolos de entrada e as ações realizadas pelos símbolos de saída a partir de ordens geradas pela máquina de inferência é que definirão a aplicabilidade de um conjunto de agentes dentro do ambiente.

É essa visão que garante uma expansão enorme ao seu rol de aplicabilidade, podendo atuar desde um integrador de sistemas de informações até como controlador de algum dispositivo físico, sendo que, neste caso, o gerador de símbolo de entrada é um conjunto de sensores e as ações causadas pelos símbolos de saída são um conjunto de reles.

Como aplicabilidade imediata, será feito o refinamento de informações do cadastro técnico de georreferenciamento de uma distribuidora de energia. Dessa forma, em um primeiro momento, a Arquitetura Multiagente atuará como um integrador de sistemas de informações, correlacionando dados de um ou mais sistemas existentes e atualizando esses dados no banco de dados de um sistema destino, nesse caso, o Cadastro Técnico.

Os levantamentos realizados junto à área de geoprocessamento de uma determinada distribuidora de energia mostram que vários dos problemas existentes no Cadastro Técnico podem ser resolvidos, pelo menos de forma parcial, reunindo informações de outros sistemas de informações disponíveis na empresa. Embora as informações espaciais não existam no banco de dados dos outros sistemas da organização, os mesmos contém informações (dados de endereços, por exemplo) que permitem ao Agente Inteligente inferir sobre essa localização, estimando

os dados de posicionamento geográficos a partir das informações extraídas de diferentes sistemas da empresa distribuidora.

Seguem alguns exemplos de aplicabilidade da arquitetura utilizada, a saber:

- 1) Encontrar, pelo endereço, a provável localização de um consumidor que se encontra no *limbo*<sup>2</sup>;
- 2) Verificar se a quantidade de consumidores por transformador, por exemplo, é condizente com a capacidade instalada;
- 3) Localização de postos transformadores sem informação de equipamentos;
- 4) Geração e análise de indicadores numéricos, bem como a produção de advertências e/ou gráficos;
- 5) Estudo e enumeração de possibilidades de geração de informações de entrada utilizando recursos tecnológicos como internet 3G ou GSM, GPS, dispositivos fotográficos de alta resolução etc.;
- 6) Construção de uma interface para esses dispositivos e de um Agente Inteligente para tratar os símbolos produzidos por uma das interfaces enumeradas.
- 7) Determinar regiões em que os índices de DEC/FEC estão fora do padrão. A clusterização geográfica pode indicar pontos de gargalo do sistema e otimizar o processo de manutenção.

Portanto, um agente inteligente deve ser programado para realizar cada uma das atividades em cada um desses casos. Ainda assim, deve existir outro agente inteligente que receba todas estas informações e as consolida em um sistema estabelecido. Logo, percebe-se uma interação participativa entre cada um dos agentes em realizar determinadas tarefas.

## VI. CONCLUSÕES

Este trabalho teve como objetivo apresentar uma proposta de utilização de um Ambiente Multiagente Dinâmico para atualização de dados técnicos em distribuidoras de energia.

Nota-se que a flexibilidade de um ambiente multiagente apresenta características importantes que contribuem para a resolução deste ou outros problemas. Isso se deve, principalmente, à capacidade adaptativa e o desenvolvimento independente de agentes computacionais participantes desse ambiente. Ou seja, a utilização de ferramentas desenvolvidas através do conceito de multiagente proporciona uma grande adaptabilidade frente a diferentes tipos de problemas que possam surgir. Os mesmos podem ser tratados dentro de um mesmo ambiente desenvolvido, não sendo, portanto, necessário realizar novos

<sup>2</sup> Denomina-se “limbo” como sendo o local onde um determinado consumidor está posicionado quando o mesmo não possui localização geográfica no Cadastro Técnico.

investimentos em outros tipos de ferramentas à medida que novas demandas surgem.

Desta forma, com crescimento da malha de distribuição e, por vezes, a existência de uma falta de cuidados em registrar/cadastrar os devidos equipamentos que estão sendo utilizados em uma determinada região, gera uma defasagem técnica que, por sua vez, pode ocasionar multas às distribuidoras de energia por parte da ANEEL.

Logo, manter a base de dados técnica mais próxima do real utilizado se faz necessário, pois, assegura o estabelecimento de tarifas justas aos consumidores de energia e mantém a qualidade das informações dentro dos limites estabelecidas pela ANEEL.

## AGRADECIMENTOS

Esta pesquisa teve o suporte financeiro da FUNAPE/CELG D através de um projeto da realização de um Projeto de Pesquisa e Desenvolvimento.

## REFERÊNCIAS

- [1] ANEEL (2012). Agência Nacional de Energia Elétrica, <http://www.aneel.org.br>. Visitada em 20/01/2012.
- [2] FERBER, J. and GASSER, L. (1991). *Intelligence artificielle distribuée*, In: INTERNATIONAL WORKSHOP ON EXPERT SYSTEMS & THEIR APPLICATIONS, 10., Avignon. Cours n. 9. France.
- [3] GASSER, L. (1992). *Boundaries, identity and aggregation: plurality issues in multiagent systems*, In: WERNER, E.; DEMAZEAU, Y. (Eds.). Decentralized AI 3. Amsterdam: North-Holland.
- [4] Pires, K. G., R. S. P. e. M. T. C. (2013). *Relatório do Projeto P&D da CELG - Uma Proposta de um Ambiente Multi-Agente Dinâmico*, Relatório Técnico. NEPE/UFG.
- [5] RICH, E. and KNIGHT, K. (1993). *Inteligência Artificial*, São Paulo: Makron Books.

## DADOS BIOGRÁFICOS

**Kayo Fernandes Pimentel**, nascido em 26/02/1990 em Goiânia-GO, é engenheiro de computação (2013). Atualmente cursa o programa de mestrado da Escola de Engenharia Elétrica, Mecânica e de Computação da UFG e participa de projeto de P&D em execução nas Centrais Elétricas de Goiás.

**Sanderley Ramos Pires** possui graduação em Ciências da Computação pela Universidade Federal de Goiás (1988), especialização em Análise e Projeto de Sistemas pela Universidade Federal de Goiás

(1997), mestrado em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (1999) e doutorado em Engenharia Elétrica pela Universidade Federal de Uberlândia (2007). Atualmente é Professor Adjunto da Universidade Federal de Goiás. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação, atuando principalmente nos seguintes temas: Equações Diferenciais, Visualização 3D, Segmentação de Imagens, Interpolação de Imagens em 3D.

**Sirlon Thiago Diniz Lacerda** possui graduação em Sistemas de Informação pelas Faculdades Alves Faria (2011) e cursa o mestrado em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás.

**Thyago Carvalho Marques** possui graduação em Ciência da Computação pelo Instituto Unificado Ensino Superior Objetivo (1999), mestrado em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (2002), doutorado em Engenharia Elétrica e Computação e pós-graduação em Economia Financeira pela Universidade Estadual de Campinas (2006). Foi professor do Centro Federal de Educação Tecnológica de Goiás (CEFET-GO) de 2006 a 2008. Atualmente é professor Adjunto III da Universidade Federal de Goiás (UFG) na Escola de Engenharia Elétrica, Mecânica e de Computação (EMC).

# VINCULAÇÃO DE UNIDADES CONSUMIDORAS AO CADASTRO TÉCNICO GEORREFERENCIADO UTILIZANDO OTIMIZAÇÃO GEOGRÁFICA: APLICAÇÃO DE AGENTES INTELIGENTES E LÓGICA FUZZY

Paulo Afonso Lobato Fernandes, Marlene Ferreira Rodrigues Gomes  
CELG - Companhia Energética de Goiás, Departamento de Geoprocessamento, Goiânia – GO, paulo.afonso@celg.com.br, marlene.frg@celg.com.br

Sirlon Diniz de Carvalho  
Instituto Federal de Goiás, Goiânia – GO, sirlondiniz@gmail.com

Sandrerley R. Pires, Sirlon Thiago D. Lacerda  
Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação, Goiânia – GO, sandrerley@ufg.br, thiago.dcfc@gmail.com

**Resumo** - Com o objetivo de aumentar a qualidade e confiabilidade do cadastro técnico georreferenciado a Celg D está em processo de implantação de um projeto de Pesquisa e Desenvolvimento que compõe-se de uma metodologia e uma ferramenta computacional para o refinamento do cadastro técnico por meio do uso de agentes inteligentes. Dentre os vários problemas que podem ocorrer ao cadastro foi escolhido o da vinculação de unidades consumidoras à rede elétrica de distribuição. O correto posicionamento elétrico e geográfico de uma unidade consumidora depende basicamente das informações trazidas de campo após sua ligação ou outro tipo de intervenção. Se houver ruído nessas informações corre-se o risco de vincular erroneamente a unidade consumidora à rede ou até não ocorrer a vinculação, no cadastro técnico. A função do agente inteligente construído é reunir elementos que garantam a correta vinculação elétrica dessa unidade consumidora à rede elétrica utilizando-se de lógica fuzzy no processo.

**Palavras-Chave** - Agentes inteligentes, Cadastro técnico, Geoprocessamento, Lógica Fuzzy, Otimização geográfica.

## LINKING CONSUMERS TO THE GEOREFERENCED TECHNICAL RECORDS USING GEOGRAPHICAL OPTIMIZATION: APPLICATION OF



XII CEEL – ISSN 2178-8308  
13 a 17 de Outubro de 2014  
Universidade Federal de Uberlândia – UFU  
Uberlândia – Minas Gerais – Brasil

## FUZZY LOGIC AND INTELLIGENT AGENTS

**Abstract** - A project of research and development aiming to increase the quality and reliability of the georeferenced technical records of CELG D is in process. The project consists of a methodology and software for the refinement of technical records through the use of intelligent agents. The problem of linking consumers to the electrical distribution network was chosen among the various problems that can occur to the records. The correct electrical and geographical positioning of a consumer unit depends mainly on the information brought in from the field after his linking or other type of intervention. If there is noise on that information there is a risk of erroneously linking the consumer unit to the network or not linking at all to the technical records. The objective of the intelligent agent is to gather evidence to ensure the correct linking of the consumer to the electrical power grid using fuzzy logic in the process.

**Keywords** - Fuzzy logic, Geographical optimization, Geoprocessing, Intelligent agents, Technical records.

## I. INTRODUÇÃO

Em 2006, a Celg Distribuição S.A. – Celg D concluiu a montagem de seu cadastro técnico georreferenciado de redes de distribuição. De lá para cá a grande preocupação da área gestora do cadastro é quanto à manutenção da qualidade e confiabilidade das informações contidas nesse banco de dados.

Atualmente a atualização das redes de distribuição no cadastro técnico é feita de forma automática via integração com os aplicativos de projeto, de obras e

de manutenção. Para a vinculação de unidades consumidoras a integração é com o sistema comercial da empresa.

A vinculação elétrica, no cadastro técnico, das unidades consumidoras (UCs) à rede de distribuição se baseia nas informações buscadas no sistema comercial. Se para determinada unidade consumidora tais informações não são completas, estão erradas ou inexistentes implica em sua não vinculação elétrica no banco de dados e, por conseguinte, é como se essa unidade consumidora não existisse. No caso de, por exemplo, uma interrupção do fornecimento de energia elétrica a essa unidade, o aplicativo de operação não conseguirá triar a interrupção e fazer o despacho para atendimento à ocorrência.

Com intuito de controlar quantas e quais unidades consumidoras não estão vinculadas à rede elétrica, criou-se uma tabela dentro do cadastro técnico chamada “Limbo” onde cada uma dessas unidades consumidoras aguarda mais informações que permitam seu correto posicionamento na rede elétrica. Esse processo de retirada do “Limbo”, antes da construção da ferramenta, precisava de intervenção humana.

A fim de corrigir essas distorções que causam prejuízos financeiros, prejudicam seus indicadores de continuidade e causam problemas de diversas ordens aos clientes envolvidos é que optou-se por corrigir o chamado “problema do Limbo” de forma automatizada e utilizando a metodologia e a ferramenta de agentes inteligentes desenvolvida no projeto de P&D “Desenvolvimento de Ambiente Multiagente e de cultura organizacional para a construção de agentes inteligentes: Uma metodologia de monitoramento do Cadastro Técnico georreferenciado” [1].

A solução proposta engloba os dados disponíveis do sistema comercial, níveis de certeza e um algoritmo de otimização com lógica Fuzzy [2], de forma a posicionar as unidades consumidoras do “Limbo” em seus respectivos circuitos elétricos e postes, na maioria dos casos, sem a necessidade de ida ao campo.

## II. DESENVOLVIMENTO

Com o objetivo de proporcionar um melhor entendimento do processo de construção da metodologia e do trabalho de Pesquisa e Desenvolvimento, apresenta-se inicialmente os fundamentos teóricos das técnicas de Inteligência Artificial aplicadas no estudo de caso e, em seguida, o desenvolvimento da aplicação, o levantamento de requisitos e resultados obtidos.

### A. Agentes Inteligentes

A metodologia está apoiada por uma arquitetura de software específica para a estruturação de um Ambiente Multiagente, onde as interações entre agentes inteligentes, dotados de capacidades específicas para atuar no ambiente de bancos de

dados, serão capazes de produzir ações consistentes de atualização do cadastro técnico da Celg D.

A construção de um ambiente multiagente e, dentro deste, a criação de agentes inteligentes envolve um conjunto de conhecimentos do usuário que não são triviais como os que são exigidos para operar um sistema de informação tradicional. O desenvolvimento da lógica, a estruturação de um problema para que o mesmo seja tratado por agentes inteligentes, a estruturação de conhecimento através de regras e outras variáveis levam as equipes desenvolvedoras do projeto a encarar um desafio que é a criação de um ambiente capaz de resolver questões complexas, mas com um modelo de operação que seja o mais simples possível, permitindo aos usuários finais desenvolver um conjunto de agentes os quais sejam capazes de solucionar problemas específicos.

Para um melhor entendimento da proposta, alguns conceitos para uma arquitetura multiagente são enunciados abaixo. Maiores informações podem ser vistas em [3].

- Os Ambientes Multiagentes: os ambientes multiagentes se caracterizam pela existência de um contexto de suporte à sobrevivência dessas entidades (os agentes inteligentes), permitindo a eles atuarem de forma independente uns dos outros e realizarem tarefas específicas. É possível fazer uma analogia entre o Ambiente Multiagente e uma organização, onde seus funcionários atuam em conjunto, mas cada um desempenhando o seu papel específico [4].
- Os agentes inteligentes: são usualmente compostos por uma máquina de inferência que tem a responsabilidade de pensar, isto é, relacionar símbolos (dados) de entrada (fatos, eventos etc.), gerando como resultado uma conclusão representada por um ou mais símbolos de saída (ações, advertências etc.) [5].

A inteligência do agente é estruturada através de um conjunto de regras que norteia o comportamento da máquina de inferência, sendo que essas regras são normalmente representadas através de operadores lógicos ou eventualmente através da tradicional estrutura das linguagens de programação, o se-então-senão. O grau de flexibilidade de um agente é extremo, pois, são as regras armazenadas nele que definem seu comportamento. Essas regras são idealizadas e alimentadas pelo criador do agente [6].

### B. Lógica Fuzzy

A lógica Fuzzy, ou lógica difusa, parte do princípio da incerteza e, desta forma, é mais complexa que a lógica booleana (zero ou um). Pode-se dizer que a incerteza nem sempre foi bem vista pela comunidade científica, sendo um estado indesejado e que deveria ser evitado a todo custo. A lógica Fuzzy, de fato, implica a ideia de que existem graus para todas as coisas, como “muito devagar”, “um pouco devagar”, “muito rápido”, “um pouco rápido”, ao contrário da lógica booleana que é capaz apenas de dizer se é

rápido ou devagar. Para problemas que exigem extrema precisão, como o controle de um laser, por exemplo, a lógica difusa pode não ser a solução. Entretanto, para problemas humanos que não necessitem de tal precisão, a utilização desse tipo de lógica pode contribuir para resultados mais satisfatórios. É possível dizer que soluções muito precisas são caras, além de necessitarem de longos períodos de desenvolvimento e produção. Em casos em que a imprecisão seja aceitável, o cientista deverá analisar as condições e, se houver tolerância a isso, utilizar lógica difusa.

Entre os objetivos da Fuzzy está a de se aproximar da forma com que o raciocínio humano relaciona as informações, com o objetivo de buscar respostas aproximadas para os problemas. Sendo assim, tem-se foco dessa lógica a solução de problemas cujas informações presentes sejam incertas, tais como as incertezas levantadas nesta solução [7].

Para melhor compreensão de como funciona a lógica Fuzzy e sua aplicação, no que tange ao comportamento não digital, isto é, o tratamento do “nem não nem sim, talvez talvez!”, podemos recorrer às Figuras 1 (a) a 1 (c).

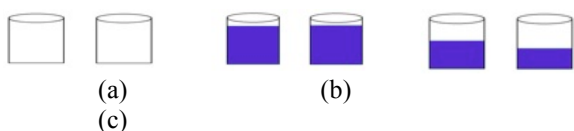


Fig. 1. (a) Exemplo de copos vazios; (b) Exemplo de copos cheios e (c) Exemplo de copos com medida variada.

Para a Figura 1 (a) pode-se questionar se os copos estão cheios ou vazios e a resposta, naturalmente, é que eles estão vazios. Para a Figura 1 (b) a mesma pergunta pode ser feita e ter-se-á que os copos estão efetivamente cheios. Para os casos anteriores é fácil observar que não há qualquer incerteza com relação à decisão, pois ou os copos estão vazios, ou estão cheios. Todavia, ao aplicar a mesma pergunta para a Figura 1 (c) não há uma resposta precisa. Pode-se afirmar, por exemplo, que o copo da esquerda está meio vazio e que o da direita está meio cheio, mas sem precisar essa resposta. Todavia, é possível estabelecer escalas de medidas que precisem o nível de maneira não discreta, flexibilizando a decisão. Para estes casos, a lógica Fuzzy apresenta-se como uma solução, remetendo-se ao caso da Tabela 1.

Não se deve confundir, entretanto, ao se dizer que a lógica Fuzzy raciocina de forma imprecisa. Na verdade, ela toma os valores imprecisos e afirma que eles são variantes entre zero e um, por exemplo, ao invés de apenas verdadeiro ou falso. Para trabalhar com essa lógica, utiliza-se os conjuntos Fuzzy, que são conjuntos de valores, ou predicados, que fazem parte de um conjunto sem limites definidos. Desta forma, a lógica difusa é um método de se raciocinar por meio de expressões lógicas que descrevem os conjuntos Fuzzy.

Sistemas Fuzzy são aproximadores universais. É possível dizer que essa lógica é bastante útil em dois

casos: em situações envolvendo sistemas altamente complexos cujos comportamentos não são conhecidos, ou em situações onde uma solução aproximada, porém rápida, é garantida. A construção de um sistema Fuzzy é feita seguindo os seguintes passos:

1. Especificar o problema e definir variáveis linguísticas.
2. Determinar os conjuntos Fuzzy.
3. Deduzir e construir as regras Fuzzy
4. Codificar os conjuntos Fuzzy, regras Fuzzy e procedimentos para realizar inferência difusa no sistema especialista.
5. Avaliar e ajustar o sistema.

Sendo assim, têm-se como primeira tarefa para construção de um sistema que utilize a lógica Fuzzy a análise do problema e a transformação dos dados em entradas variáveis. Após esta transformação deve-se determinar também as funções de pertinência, ou seja, os graus de certeza e incerteza, tal como exemplificado na Tabela 1.

A partir da aplicação desses conceitos, torna-se possível fazer inferências diversas, considerando não somente a totalidade ou formato discreto da lógica proposicional, mas também os níveis de certeza variados, também comuns no pensamento humano.

Apesar de a lógica Fuzzy não ser perfeita e ainda estar em constante aperfeiçoamento, acredita-se que para o caso de otimização geográfica ela se faz suficiente e é capaz de prover resultados satisfatórios.

### C. Levantamento de requisitos

São, principalmente, dois os bancos de dados envolvidos na solução do “problema do Limbo”: o cadastro técnico georreferenciado, chamado Sistema de Gestão Técnica – SGT e o cadastro comercial, chamado CBILL.

Frequentemente ocorrem erros em cadastros de unidades consumidoras em postes ou em circuitos elétricos incorretos. Quando ocorrem esses erros as unidades consumidoras são alocadas no “Limbo”, ou seja, uma referência fictícia do cadastro técnico que comporta unidades consumidoras que não possuem localização geográfica correta. Para que uma unidade consumidora apareça corretamente no SGT, ela precisa estar conectada, no mínimo, a um posto transformador, porém, preferencialmente a um poste.

O acúmulo desses erros provocou, ao longo dos anos, o envio de milhares de unidades consumidoras de toda a área de concessão da Celg D para o “Limbo”. Ao analisar-se uma a uma, percebe-se que não existe um padrão para os resultados, variando de unidades consumidoras situadas desde a capital até a zona rural.

Tendo isso em mente, é possível perceber o quão prejudicial à empresa é o “Limbo”. Em caso de manutenção corretiva solicitada por um cliente, cuja unidade consumidora esteja no “Limbo” o Centro de Operação da Distribuição - COD, por vezes, necessita executar trabalhos de pesquisa a fim de descobrir a localização dessa unidade. Além do custo de tempo e

recursos humanos da tarefa, existe, ainda, o risco de a equipe ser enviada a um local incorreto, gerando ainda mais prejuízo.

No entanto, sabe-se que os endereços das unidades consumidoras que têm leitura e faturamento normais estão disponíveis no CBILL. Desta forma, a utilização desses endereços na solução é imprescindível e, junto ao ambiente multiagente, engloba todas as informações e recursos necessários para a implementação da solução.

O ambiente multiagente é o responsável pela extração das informações dos sistemas existentes, bem como atualização pós execução do algoritmo de otimização, que será tratado também como uma ação de um agente com inteligência híbrida, isto é, regras simbólicas na forma de um sistema especialista [8][9] e a realização de inferência via lógica Fuzzy. A fusão das informações constantes nos três sistemas, sendo essas utilizadas pela solução multiagente, visam a diminuição problema do “Limbo”. É importante notar em alguns erros mais complexos a visita em campo faz-se necessária.

A fim de melhorar as informações do cadastro técnico da Celg D propõe-se uma solução de otimização geográfica. Essa solução utiliza informações do CBILL, do SGT, bem como de um conjunto de agentes do sistema multiagente desenvolvido no citado projeto de P&D, sendo que, esse último também irá contribuir na extração e atualização das informações.

Para a construção da solução, é utilizada uma tecnologia de geolocalização como o Google Maps. Os endereços das unidades consumidoras do “Limbo” e que estão corretos no CBILL possibilitam a geração de uma área capturada, que será chamada de buffer, como o apresentado na Figura 2.

A partir desse buffer, são encontrados os postos transformadores e postes no SGT. As coordenadas desses elementos de rede, que estão em coordenadas poliocônicas, são convertidas para coordenadas aceitas pelo Maps seguindo uma série de regras de projeções. Em grande parte dos casos, o buffer gerado engloba muitos postes e, por vezes, mais de um posto transformador. Isso dificulta a identificação do poste e torna a alocação de uma unidade consumidora um processo complicado e sujeito a ação humana.



Fig. 2. Exemplo de área capturada (buffer) pelo Google Maps

Por meio do buffer gerado e das informações extraídas por um dos agentes, é possível identificar todos os postos transformadores e postes da região, dando início ao processo de estudo para alocação nos locais corretos.

Tendo em mãos os pontos resgatados pela extração, parte-se para a segunda parte, que é a análise de opções para definição das regras e seus níveis de certeza. Esse trabalho é feito em forma de uma tabela de níveis de certeza. Essa tabela começou a ser construída a partir de reuniões com as equipes da Fundação de Apoio à Pesquisa (FUNAPE) e da Celg D. Diversas reuniões foram necessárias para se chegar a esse modelo preliminar de decisão e envolveu pessoas de níveis de supervisão, operacional e gerencial. A Tabela 1 apresenta o modelo de decisão que será o ponto inicial para a construção da solução do “Limbo”.

Tabela I - Regras e níveis de certeza

Situação provável	Desdobramento	Ação provável	Nível de certeza (precisão)
É possível fazer a localização geográfica a a partir do endereço	1. e encontra-se um único poste com posto transformador próximo	Posicionar unidade consumidora (UC) no poste próximo	10
	2. e encontra-se um único poste próximo (com baixa tensão)	Verificar se há seccionamento no poste e procurar a partir dos vizinhos (roteiro). Se ambos estão no mesmo posto transformador, posicionar no poste	10
	3. e encontra-se mais de um poste próximo (com baixa tensão)	Verificar se há seccionamento no poste e procurar a partir dos vizinhos (roteiro). Havendo	9

		seccionamento, sugerir visita "in loco". Não havendo, posicionar no poste mais próximo.	
Não é possível fazer localização geográfica a partir do endereço	1. mas tem-se a localização geográfica das UCs anterior e posterior (roteiro de leitura) com quadra e poste comum entre todos esses	Não havendo seccionamento no poste, posicionar no poste	10
	2. mas tem-se localização geográfica das UCs anterior e posterior (roteiro de leitura) com poste comum entre esses	Não havendo seccionamento no poste, posicionar no poste	10
	3. mas tem-se localização geográfica das UCs anterior e posterior (roteiro de leitura), cada UC com poste diferente, no entanto, com vizinhos em posto transformador comum	Posicionar no posto transformador	10
	4. mas tem-se localização geográfica das UCs anterior e posterior (roteiro de	Sugerir visita "in loco" para decidir qual posto transformador posicionar	10

	leitura), cada UC com poste diferente, no entanto, com vizinhos em postos transformadores diferentes		
	5. As duas UCs anteriores e posteriores no roteiro de leitura também encontram-se no limbo	Sugerir visita "in loco" com maior prioridade (possível classificação de prioridade A/B/C/D/E, sendo A com maior prioridade)	10

Por meio dessa tabela e do levantamento das situações prováveis, desdobramentos, ações prováveis e níveis de certeza, é possível desenvolver um algoritmo que irá otimizar e promover o reposicionamento de cada unidade consumidora do "Limbo". O algoritmo irá obedecer cada uma das regras e será capaz de reduzir o buffer completamente até que seja possível alocar a unidade consumidora em um posto transformador e/ou poste ou, em último caso, gerar uma tabela com os casos que não solucionados, indicando ações mais precisas para visitas "in loco" pelas equipes de cadastro técnico. Observe que mesmo para estes últimos casos, também haverá uma otimização, uma vez que haverá um número reduzido de ações e os casos serão elucidados pelo sistema multiagente que indicará qual o problema de posicionamento, antes completamente desconhecido.

O problema apresentado provocou a equipe a novos desdobramentos de estudos com o objetivo de definir qual a técnica de Inteligência Artificial mais adequada para resolver o problema. Há que se observar, entretanto, que trata-se de uma tabela que apresenta os problemas e soluções mais triviais e identificadas durante as reuniões entre FUNAPE e Celg D. Todavia, acredita-se que com a aplicação da solução, novas regras de reposicionamento surgirão e deverão ser integradas da forma mais simples possível ao modelo multiagente. Diante de tal situação, a equipe FUNAPE entendeu que a utilização de lógica Fuzzy, ou lógica difusa é a mais indicada para utilização no modelo multiagente híbrido, pois ela é capaz de interpretar e trabalhar com incertezas. As incertezas da lógica Fuzzy podem ser mais favoráveis e ela pode ser capaz de fornecer resultados mais satisfatórios, visto que trata-se de posicionamento

geográfico com níveis de incertezas não discretos, ou seja, com variações de níveis de decisão, tão comum ao pensamento humano.

Espera-se como resultado, então, que o algoritmo seja capaz de executar as regras levantadas para cada uma das unidades consumidoras e, de forma automatizada, consiga reduzir a área de buffer a ponto de conseguir aloca-las em seus postos transformadores e/ou postes corretos.

No entanto, é importante ressaltar que, em certos casos, o algoritmo não é capaz de identificar apenas um posto transformador ou poste. Para esses casos, como já mencionado, relatórios devem ser gerados para que sejam analisados por colaboradores especializados e, provavelmente, uma visita de campo far-se-á necessária.

Tem-se como limitação do algoritmo, também, o fato de que ele será mais eficaz em grandes cidades e cidades médias, pois, as muito pequenas não dispõem de um sistema de CEP completo. Mesmo assim, para esses casos o sistema disponibiliza informações que poderão ser utilizadas, tais como razão (data de leitura), rotas e roteiros utilizados pelas equipes de medição e entrega de faturas. Ainda assim, para áreas rurais o algoritmo fica impossibilitado de ser executado, visto que a localização do endereço torna-se impossível a priori. Soluções para essas limitações fazem parte de estudos futuros.

O algoritmo funciona como um agente do sistema multiagente desenvolvido no projeto de P&D e consistirá de um serviço automatizado que é executado constantemente de acordo com as regras estabelecidas, sendo que as mesmas podem ser mudadas ao longo da execução.

As informações a serem utilizadas pelo algoritmo deverão ser providas por outros agentes do mesmo sistema, os quais efetuam a extração de dados em um processo de mineração de dados.

Após a extração, análise e otimização feita pelo algoritmo, as informações são atualizadas na base de dados da empresa e, como resultado, tem-se uma diminuição satisfatória do número de unidades consumidoras no “Limbo”.

Como resultado esperado feito após análise inicial, estipula-se que a diminuição de unidades consumidoras do “Limbo” no primeiro período poderá chegar a 50%. No decorrer das ações e refinamento das informações contidas no sistema georreferenciado, esse número pode aumentar.

### **C. Resultados obtidos**

Com o objetivo de avaliar a aplicabilidade do modelo proposto a equipe do projeto preparou um ambiente computacional de testes, utilizando um computador PC de 2GHz com 500GB de disco e 8GB de memória principal. Neste computador foram instalados o banco de dados Oracle, o servidor web Tomcat e todo o ambiente de desenvolvimento Java, incluindo a IDE Eclipse. Além disso, o ambiente multiagente foi instalado nesta máquina servidora para a estruturação da solução multiagente.

Depois de preparado o ambiente, a equipe identificou as UCs do Limbo para então realizar os testes. De posse dessas UCs, a equipe retirou uma amostra de 2.500 UCs para o estudo de caso.

Nos testes iniciais a equipe aplicou as regras que tratam somente do cruzamento de informações entre os sistemas comercial e o técnico. As premissas que envolvem a utilização das soluções do Google estão em fase de conclusão no ambiente. Portanto, utilizando-se somente a situação provável “Não é possível fazer localização geográfica a partir do endereço”, conforme indicado na Tabela 1, o sistema mostrou-se capaz de reposicionar 1.304 unidades das 2.500 amostradas, totalizando 52,16%.

Diante desses resultados e considerando ainda que nem todas as regras foram implementadas, espera-se que esse número possa ser melhorado de forma significativa. Ainda assim, observa-se que esse resultado supera a expectativa inicial prevista para o projeto, que era de 50%.

## **III. CONCLUSÕES**

Os estudos e levantamentos feitos para construção da solução do “problema do Limbo” permitiram a identificação da possibilidade de reposicionamento de UCs do “Limbo” sem a necessidade de visita “in loco” para um percentual significativo de unidades consumidoras. Por outro lado, a descrição dos problemas e o levantamento de requisitos desafiou a equipe de pesquisa na identificação de soluções de Inteligência Artificial que pudessem atender aos aspectos levantados, remetendo à construção de modelos multiagentes híbridos. A análise do problema indicou que a lógica Fuzzy pode ser a mais adequada para tratar o problema do limbo, pois permite fazer inferências mesmo que os níveis de certeza de decisão sejam variados. Sendo assim, apresentaram-se os encaminhamentos de decisões e a tecnologia a ser adotada em parte dos agentes do sistema.

Uma base cadastral de redes confiável é premissa básica para uma gestão técnica de distribuição com qualidade. A preocupação, não só da Celg D, mas de todas as concessionárias de distribuição é como manter seu cadastro técnico sempre atualizado e com o mínimo de inconsistências.

A metodologia e a ferramenta computacional descritas neste artigo apresentam-se como aliadas poderosas no sentido de reduzir consideravelmente as falhas contidas no banco de dados, e não só nesse aspecto de vinculação de unidades consumidoras à rede elétrica, uma vez que tais ferramentas podem ser adaptadas para solução de problemas de outras naturezas que afligem o cadastro técnico.

Esse esforço, em última análise, representa uma considerável melhoria na qualidade do serviço prestado pela Celg D.

## **AGRADECIMENTOS**

Os autores agradecem à Celg Distribuição S/A, aos colaboradores do departamento de Geoprocessamento pela colaboração neste trabalho e à Agência Nacional de Energia Elétrica (ANEEL).

#### REFERÊNCIAS

- [1] K. F. Pimentel, S. T. D. Lacerda, T. C. Marques, S. R. Pires, "Uma proposta de utilização de um ambiente multiagente dinâmico para atualização de dados técnicos em distribuidoras de energia", XI CEEL, Uberlândia-MG, Brasil, 2013.
- [2] T. J. Ross, *Fuzzy Logic With Engineering Applications*, 2 ed., John Wiley & Sons Ltd., England, 2004.
- [3] K. G. Pires, S. R. Pires, T. C. Marques, *Relatório do Projeto P&D da CELG - Uma Proposta de um Ambiente Multiagente Dinâmico*, Relatório Técnico. NEPE/UFG, 2013.
- [4] L. Gasser, *Boundaries, identity and aggregation: plurality issues in multiagent systems*, In: WERNER, E.; DEMAZEAU, Y. (Eds.). *Decentralized AI 3*. Amsterdam: North-Holland, 1992.
- [5] E. Rich, K. Knight, *Inteligência Artificial*, São Paulo: Makron Books, 1993.
- [6] J. Ferber, L. Gasser, *Intelligence artificielle distribuée*, In: INTERNATIONAL WORKSHOP ON EXPERT SYSTEMS & THEIR APPLICATIONS, 10, Avignon. Cours n. 9. France, 1991.
- [7] A. G. Aguado, M. A. Cantanhede, "Lógica Fuzzy", Congresso UNICAMP, Limeira, 2010.
- [8] S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3 ed., Pearson Education Limited, United States of America, 2010.

- [9] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*, 2 ed., Pearson Education Limited, England, 2005.

#### DADOS BIOGRÁFICOS

**Paulo Afonso Lobato Fernandes**, possui graduação em Engenharia Elétrica pela Universidade Federal Goiás (1983). Atualmente é Gerente do setor de Geoprocessamento na Celg Distribuição S/A.

**Marlene Ferreira Rodrigues Gomes**, possui graduação em Ciências da Computação pela PUC-GO (2002). Atualmente é Eletrotécnica na Celg Distribuição S/A.

**Sirlon Diniz de Carvalho**, doutor em Engenharia Elétrica pela Universidade Federal de Uberlândia (2013), mestre em Engenharia de Computação pela Universidade Federal Goiás (2002). Atualmente é Professor do Instituto Federal de Goiás.

**Sanderley Ramos Pires**, possui graduação em Ciências da Computação pela Universidade Federal de Goiás (1988), especialização em Análise e Projeto de Sistemas pela Universidade Federal de Goiás (1997), mestrado em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (1999) e doutorado em Engenharia Elétrica pela Universidade Federal de Uberlândia (2007). Atualmente é Professor Adjunto da Universidade Federal de Goiás.

**Sirlon Thiago Diniz Lacerda**, possui graduação em Sistemas de Informação pelas Faculdades Alves Faria (2011) e cursa o mestrado em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás. Atualmente é Professor no Instituto Federal de Goiás.

## **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.dcfc@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.*

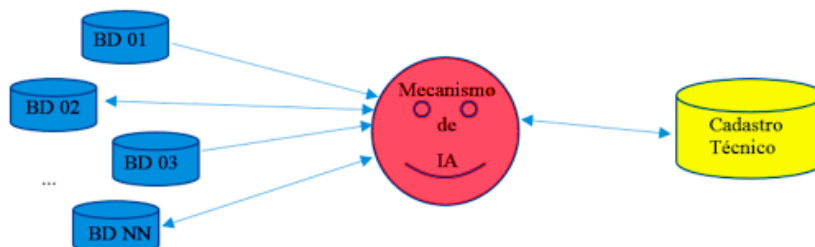
### **1. INTRODUÇÃO**

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 tem 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 do usuário no momento de estruturar a sua solução.

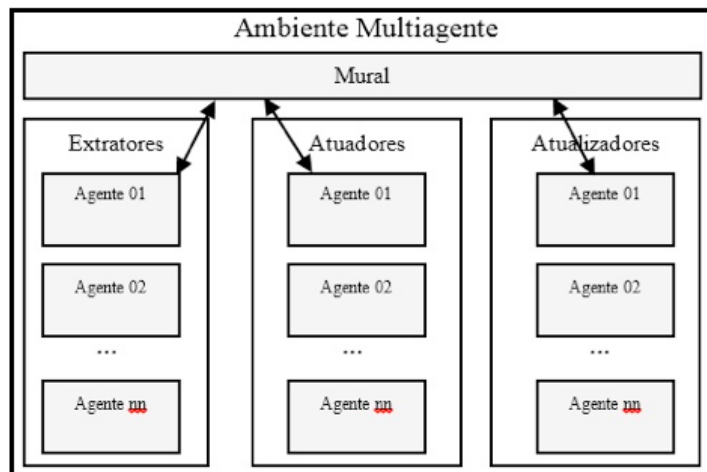
A interface aqui proposta se estrutura a partir de análise em problemas dentro da Celg D, onde soluções multiagente possam ser úteis para o refinamento de informações no cadastro

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. Mural: área comum para a comunicação entre agentes. Os agentes escrevem símbolos no mural para expor alguma informação. Símbolos podem ativar o comportamento de agentes do ambiente.
2. Agentes Extratores: são entidades capazes de buscar informações em bancos de dados, planilhas, arquivos de textos e Web Services, expondo tais informações no mural;
3. Agentes Atuadores: são as estruturas que realizarão as atividades de raciocínio, correlacionando os produtos dos extratores, analisando-os e chegando a conclusões que serão repassadas, via Mural, aos atualizadores;
4. Agentes Atualizadores: entidades que executam atualizações em bancos de dados ou, adicionalmente, podem gerar relatórios de advertência para os usuários;

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.

Segundo Weiss (1999), os ambientes de sistemas multiagente provêm uma infraestrutura especificando protocolos de comunicação e interação; são geralmente abertos e não possuem apenas um projetista – não são centralizados; contêm agentes que são autônomos e distribuídos, os quais podem ser competitivos ou cooperativos.

### 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).

Entretanto, este trabalho, quando utiliza-se de gráficos, está focado na determinação de estados importantes para ativar as ações de um agente inteligente. Pode-se considerar que parte da estrutura aqui proposta, principalmente a que determina os estados importantes, pode ser classificada como um subconjunto do modelo proposto por Harel e com algumas modificações para descrever os comportamentos do agente. Ressalta-se que, ao contrário do modelo de Harel, a estrutura aqui proposta não se preocupa com qual seria o próximo estado do agente após a realização das ações previstas em seu comportamento.

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, usando este conhecimento para direcionar o usuário inexperiente a construir seu processo de extração de informação.
2. SQL: para usuários mais experientes é possível a configuração do agente Extrator de informação por meio da construção de um comando SQL para realizar a extração de informação dos 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). A origem da informação para a atualização dos dados no BD estarão presentes no símbolo que ativou o agente Atualizador.

- 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. Os dados que comporão o relatório serão retirados do símbolo que ativou o agente Atualizador.

### 3.4. O Agente Atuador

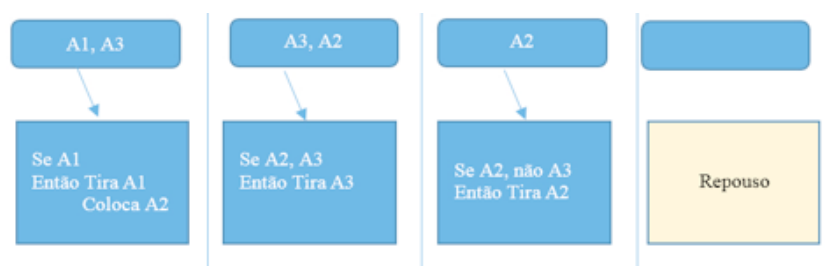
Um dos aspectos mais difíceis no momento de compor uma solução multiagente é a criação de uma base de conhecimento. Para se criar essa base é necessário entender qual será o correto encadeamento de estados durante o processo de raciocínio até se atingir uma conclusão. Os estados são definidos pela configuração 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. Sempre que os símbolos do ambiente satisfizerem alguma estrutura “SE”, o agente reagirá modificando os símbolos do ambiente até o momento em que nenhum “SE” seja mais satisfeito.

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 o 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 realiza 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

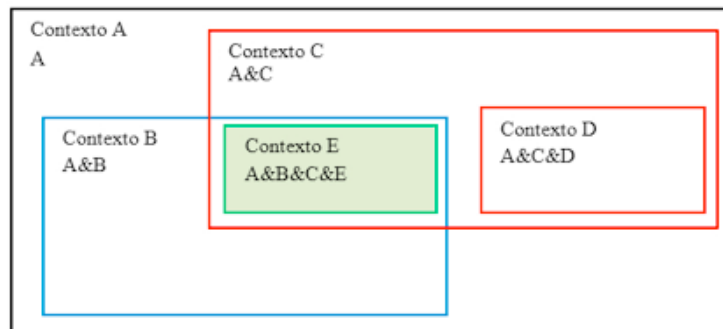


#### *A estruturação de contextos*

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.

Diversos outros contextos podem existir na base de conhecimento do agente, fazendo que mesmo graficamente a expressão do conhecimento seja complexa, mas sempre terá uma forma mais simples que a de scripts.

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. Isto é, se no ambiente encontram-se os símbolos “A” e “C”, o conjunto de ações mapeadas na região onde A&C é ativado, levando ao disparo das ações ligadas a essa região de ativação.

***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 uma 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 de forma gráfica 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. É provável que o desenvolvimento de um raciocínio vá ocorrer através de um fluxo de saída dos contextos mais simples e caminha, nos ciclos de inferência, para os contextos mais complexos;

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.

Para o registro das regras, propõe-se neste trabalho que a partir de um ícone localizado no canto superior direito do diagrama de contextos, seja possível passar para outra interface para inserir o conjunto de ações. 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. O que este trabalho propõe é que o usuário analise os estados possíveis de forma gráfica, pois assim a sua visão sistêmica do problema será facilitada.

#### 4. RESULTADOS EM UMA APLICAÇÃO EXEMPLO

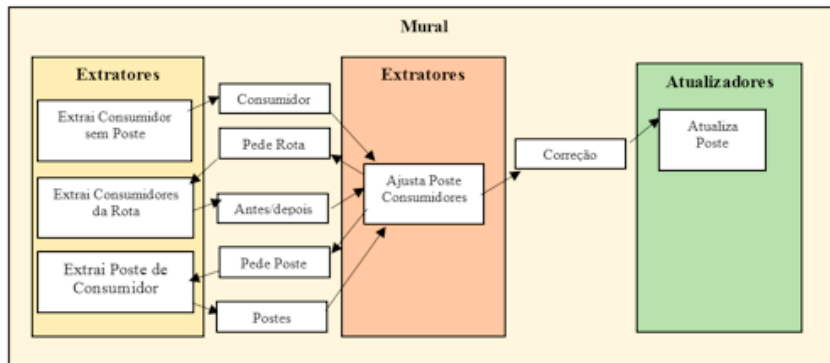
A ideia estruturada e detalhadas nas seções anteriores viabiliza a construção de uma aplicação capaz de criar a base de dados para agentes inteligentes. Neste artigo são apresentados os resultados da implementação de uma aplicação criada para a Celg D.

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. Diversos outros aspectos poderiam ser tratados nesta solução, mas para efeito de demonstrar a funcionalidade do ambiente multiagente usaremos o escopo da solução já exposta.

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 “Ajusta 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. As informações obtidas pelos três agentes extratores será suficiente para realizar todo o processo de análise da solução.

##### *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*

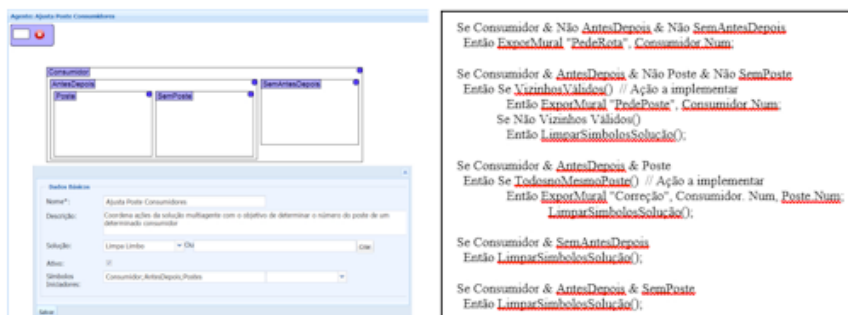
Normalmente, uma solução poderá se estruturada com apenas um agente Atuador, embora, a critério do desenvolvedor, possam ser utilizados mais que um agente. Neste exemplo, um único agente foi criado cujo nome é “Ajusta Poste Consumidor”, mas no

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. Observe o nível de facilidade oferecido ao usuário com a abordagem proposta.

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 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 Workshops*, p. 297-305, 2011.

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

WEISS, G. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

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:** *Georefencing, Intelligent Agents, Thought Structuring*

## **Proposta de Utilização de um Ambiente Multiagente Dinâmico Para Atualização de Dados Técnicos em Distribuidoras de Energia**

Sirlon Thiago D. Lacerda; Sandrerley Ramos Pires; Gélson da Cruz Júnior

Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), UFG,

Goiânia, Goiás.

[sirlonthiago@yahoo.com.br](mailto:sirlonthiago@yahoo.com.br); [sandrerley@ufg.br](mailto:sandrerley@ufg.br); [gcruz@emc.ufg.br](mailto:gcruz@emc.ufg.br)

### **Resumo**

Este trabalho tem como objetivo apresentar uma proposta de utilização de um ambiente multiagente dinâmico para atualização de dados técnicos georreferenciados em Distribuidoras de Energia. Dada a expansão da malha de distribuição causada por um processo contínuo de crescimento de número de consumidores, ocorre uma grande quantidade de atualizações no cadastro técnico, comprometendo a qualidade das informações, às vezes não sendo suficiente para manter os dados nos níveis de qualidade exigidos pela ANEEL. A arquitetura proposta neste trabalho procura estabelecer um ambiente com agentes inteligentes capazes de vigiar, corrigir e advertir situações em que o processo de atualização do cadastro técnico esteja sendo feito de maneira incorreta.

**Palavras-chave** - Distribuidoras de Energia, sistemas inteligentes e Sistemas Multiagentes.

### **Introdução**

Dentre as diversas atividades de uma Distribuidora de Energia Elétrica, ela é responsável por realizar atividades de planejamento, instalação, manutenção e operação das redes elétricas. E, para melhor desenvolver essas atividades, a qualidade dos dados georreferenciados existentes em seus sistemas é de fundamental importância, a fim de que o que se tem registrado no banco de dados reflita a realidade de campo.

Durante o período de 2004 a 2006, a CELG D (Celg Distribuição S.A.) desenvolveu o banco de dados georreferenciado, denominado de Cadastro Técnico. Esse cadastro possui informações das redes primárias, secundárias, dos equipamentos e da vinculação das unidades consumidoras à rede. Uma maior representatividade da situação física da rede elétrica é sempre desejada, uma vez que, além de assegurar o estabelecimento de tarifas justas aos consumidores de energia, também mantém a qualidade das informações dentro dos limites estabelecidos pela ANEEL (ANEEL, 2012). Entretanto, o que se observa ao longo do tempo é o aparecimento de problemas das mais diversas ordens que tendem a prejudicar a qualidade do cadastro técnico da CELG D.

Para apresentar o conjunto de ações integradas que visam automatizar o processo de refinamento da qualidade do cadastro técnico da CELG D, este trabalho tem como objetivo mostrar a proposta de utilização de um ambiente multiagente dinâmico para atualização de dados técnicos do cadastro da CELG D. Agentes especializados em extrair informações de diversas fontes diferentes, aliado a agentes inteligentes capazes de realizar inferências a partir dos dados trazidos pelos extratores e, finalmente, agentes capazes de atualizar informações no banco de dados do cadastro técnico ou, simplesmente, gerar relatórios com advertências que apontam prováveis irregularidades do cadastro técnico.

### **Agentes Inteligentes**

A metodologia está apoiada por uma arquitetura de software específica para a estruturação de um Ambiente Multiagente, onde as interações entre agentes inteligentes, dotados de capacidades específicas para atuar no ambiente de bancos de dados, serão capazes de produzir ações consistentes de atualização do cadastro técnico da CELG D.

É possível fazer uma analogia entre o Ambiente Multiagente e uma organização, onde seus funcionários atuam em conjunto, mas cada um desempenhando o seu papel específico é algo perfeitamente válido (GASSER, 1992). Os agentes inteligentes são, usualmente, compostos por uma máquina de inferência que tem a responsabilidade de pensar, isto é, relacionar símbolos (dados) de entrada (fatos, eventos etc.), gerando como resultado uma conclusão representada por um ou mais símbolos de saída (ações, advertências etc.) (RICH e KNIGHT, 1993).

A inteligência do agente é estruturada através de um conjunto de regras que norteia o comportamento da máquina de inferência, sendo que essas regras são normalmente representadas através de operadores lógicos ou eventualmente através da tradicional estrutura das linguagens de programação, o se-então-senão. O grau de flexibilidade de um agente é extremo, pois, são as regras armazenadas nele que definem seu comportamento. Essas regras são idealizadas e alimentadas pelo criador do agente (FERBER e GASSER, 1991).

### **Modelo Proposto**

No modelo proposto, três tipos de agentes atuarão na estrutura multiagente. Todos eles possuem as características de um agente tradicional, porém diferenciam-se pelas suas capacidades propiciadas por suas estruturas marginais.

Os Agentes Extratores obtêm informações externas e transformá-las em símbolos no ambiente multiagente. Esses símbolos motivarão a atuação dos demais agentes do ambiente. As possíveis origens de informações que um agente extrator é capaz de acessar são os bancos de dados, planilhas, arquivos de textos não estruturados e web services. O agente extrator é capaz de combinar essas diferentes origens em uma mesma operação de busca de informação.

Os Agentes Atualizadores tem a responsabilidade de externalizar uma decisão inferida pelo ambiente. Essa externalização pode ser feita através de uma atualização em banco de dados ou na criação de arquivos textos, planilhas ou até mesmos relatórios formatados para impressão e análise futura.

Os Agentes Atuadores estabelecem as ações que serão tomadas para tratar um determinado símbolo ou um conjunto desses. O conhecimento necessário para resolver o problema, nesta arquitetura proposta, é mapeado no agente atuador. A partir das ações de um atuador, novos extratores podem ser ativados, outros atuadores podem ser chamados a colaborar com o processo de solução e, finalmente, o controle pode ser transferido para um agente atualizador para efetivar o processo de solução do sistema.

Nessa arquitetura, a integração das três estruturas, a de conversão de informação em símbolo (agente extrator), inferências baseadas em regras da base de conhecimento (agente atuador) e conversão de símbolo em ação (agente

atualizador), dá ao ambiente de agentes inteligentes a possibilidade de assumir várias configurações, extrapolando até o seu objetivo imediato que é refinar a qualidade do cadastro técnico da CELG D.

### **Aplicabilidade do Ambiente**

Como aplicabilidade imediata, será feito o refinamento de informações do cadastro técnico de georreferenciamento de uma distribuidora de energia. Dessa forma, em um primeiro momento, a Arquitetura Multiagente atuará como um integrador de sistemas de informações, correlacionando dados de um ou mais sistemas existentes e atualizando esses dados no banco de dados de um sistema destino.

A seguir, tem-se alguns exemplos de aplicabilidade da arquitetura utilizada.

1. Encontrar, pelo endereço, a provável localização de um consumidor que se encontra sem localização geográfica;
2. Verificar se a quantidade de consumidores por transformador, por exemplo, é condizente com a capacidade instalada;
3. Localização de postos transformadores sem informação de equipamentos;
4. Geração e análise de indicadores numéricos, bem como a produção de advertências e/ou gráficos.

Portanto, um agente inteligente deve ser programado para realizar cada uma das atividades em cada um desses casos. Ainda assim, deve existir outro agente inteligente que recebe todas estas informações e as consolida em um sistema estabelecido. Logo, percebe-se uma interação participativa entre cada um dos agentes em realizar determinadas tarefas.

### **Conclusões**

Este trabalho teve como objetivo apresentar uma proposta de utilização de um Ambiente Multiagente Dinâmico para atualização de dados técnicos em distribuidoras de energia.

Nota-se que a flexibilidade de um ambiente multiagente apresenta características importantes que contribuem para a resolução deste ou outros problemas. Isso se deve, principalmente, à capacidade adaptativa e o desenvolvimento independente de agentes computacionais participantes desse ambiente. Ou seja, a utilização de ferramentas desenvolvidas através do conceito de

multiagente proporciona uma grande adaptabilidade frente a diferentes tipos de problemas que possam surgir. Os mesmos podem ser tratados dentro de um mesmo ambiente desenvolvido, não sendo, portanto, necessário realizar novos investimentos em outros tipos de ferramentas à medida que novas demandas surgem.

Desta forma, com crescimento da malha de distribuição e, por vezes, a existência de uma falta de cuidados em registrar/cadastrar os devidos equipamentos que estão sendo utilizados em uma determinada região, gera uma defasagem técnica que, por sua vez, pode ocasionar multas às distribuidoras de energia por parte da ANEEL. Logo, manter a base de dados técnica mais próxima do real utilizado se faz necessário, pois, assegura o estabelecimento de tarifas justas aos consumidores de energia e mantém a qualidade das informações dentro dos limites estabelecidos pela ANEEL.

### **Agradecimentos**

Esta pesquisa teve o suporte financeiro da FUNAPE/CELG D através de um projeto da realização de um Projeto de Pesquisa e Desenvolvimento.

### **Referências**

ANEEL (2012). Agência Nacional de Energia Elétrica, <http://www.aneel.org.br>. Visitada em 20/01/2012.

FERBER, J. and GASSER, L. (1991). Intelligence artificielle distribuée, In: INTERNATIONAL WORKSHOP ON EXPERT SYSTEMS & THEIR APPLICATIONS, 10., Avignon. Cours n. 9. France.

GASSER, L. (1992). Boundaries, identity and aggregation: plurality issues in multiagent systems, In: WERNER, E.; DEMAZEAU, Y. (Eds.). Decentralized AI 3. Amsterdam: North-Holland.

Pires, K. G., R. S. P. e. M. T. C. (2013). Relatório do Projeto P&D da CELG - Uma Proposta de um Ambiente Multi-Agente Dinâmico, Relatório Técnico. NEPE/UFG.

RICH, E. and KNIGHT, K. (1993). Inteligência Artificial, São Paulo: Makron Books.

# VINCULAÇÃO DE UNIDADES CONSUMIDORAS AO CADASTRO TÉCNICO GEORREFERENCIADO UTILIZANDO OTIMIZAÇÃO GEOGRÁFICA: APLICAÇÃO DE AGENTES INTELIGENTES E LÓGICA FUZZY

Paulo Afonso Lobato Fernandes \*  
Sirlon Thiago Diniz Lacerda \*\*  
Sirlon Diniz de Carvalho \*\*\*  
Sandrerley Ramos Pires \*\*\*\*  
Marlene Ferreira Rodrigues Gomes \*\*\*\*\*

**RESUMO:** Com intuito de aumentar a qualidade e confiabilidade de seu cadastro técnico georreferenciado, a Celg D está em processo de implantação de um projeto de Pesquisa e Desenvolvimento intitulado “Desenvolvimento de Ambiente Multiagente e de cultura organizacional para a construção de agentes inteligentes: Uma metodologia de monitoramento do Cadastro Técnico georreferenciado” que compõe-se de uma metodologia e uma ferramenta computacional que objetiva o refinamento do cadastro técnico através de soluções construídas com agentes inteligentes.

Dentre os vários problemas que podem ocorrer ao cadastro foi escolhido o da vinculação de unidades consumidoras à rede elétrica de distribuição. O correto posicionamento elétrico e geográfico de uma unidade consumidora depende basicamente das informações trazidas de campo após sua ligação ou outro tipo de intervenção.

Se houver ruído nessas informações corre-se o risco de vincular erroneamente a unidade consumidora à rede ou até não ocorrer a vinculação, no cadastro técnico. A função do agente inteligente construído é, ao utilizar soluções de otimização geográfica, reunir elementos que garantam a correta vinculação elétrica dessa unidade consumidora à rede elétrica.

Atualmente o percentual, da Celg D, de unidades consumidoras não vinculadas à rede está em torno de 0,85%, ou 22.500 em números absolutos, com os testes realizados com o agente conseguiu-se um sucesso de 52,16%.

**Palavras-chave:** Geoprocessamento. Cadastro técnico. Vinculação elétrica de unidades consumidoras. Otimização geográfica. Agentes inteligentes. Lógica Fuzzy.

## 1 INTRODUÇÃO

---

\* Engenheiro Eletricista, Especialista em Gestão Técnica de Concessionárias de Energia Elétrica. paulo.afonso@celg.com.br, Rua 2, 505, s. 301-A, Jardim Goiás - 74805-180 - Goiânia-GO.

\*\* Formado em Sistemas de Informação pelas Faculdades Alves Faria e mestrando em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás. sirlonthiago@yahoo.com.br.

\*\*\* Mestre em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (UFG) e doutor em Engenharia Elétrica pela Universidade Federal de Uberlândia (UFU), sirlondiniz@gmail.com, IFG, Câmpus Goiânia, Goiás.

\*\*\*\* Mestre em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (UFG) e doutor em Engenharia Elétrica pela Universidade Federal de Uberlândia (UFU), sandrerley@hotmail.com, Escola de Engenharia Elétrica e Computação, UFG, Goiânia, Goiás.

\*\*\*\*\* Eletrotécnica, Cientista da computação. marlene.frg@celg.com.br, Rua 2, 505, S. 301-A, Jardim Goiás - 74805-180 - Goiânia-GO.

Em 2006 a Celg Distribuição S.A. – Celg D concluiu a montagem de seu cadastro técnico georreferenciado de redes de distribuição. De lá para cá a grande preocupação da área gestora do cadastro é quanto à manutenção da qualidade e confiabilidade das informações contidas nesse banco de dados.

Atualmente a atualização das redes de distribuição no cadastro técnico é feita de forma automática via integração com os aplicativos de projeto, de obras e de manutenção. Para a vinculação de unidades consumidoras a integração é com o sistema comercial da empresa.

A vinculação elétrica, no cadastro técnico, das unidades consumidoras (UCs) à rede de distribuição se baseia nas informações buscadas no sistema comercial. Se para determinada unidade consumidora tais informações não são completas, estão erradas ou inexistentes implica em sua não vinculação elétrica no banco de dados e, por conseguinte, é como se essa unidade consumidora não existisse. No caso de, por exemplo, uma interrupção do fornecimento de energia elétrica a essa unidade, o aplicativo de operação não conseguirá triar a interrupção e fazer o despacho para atendimento à ocorrência.

Com intuito de controlar quantas e quais unidades consumidoras não estão vinculadas à rede elétrica, criou-se uma tabela dentro do cadastro técnico chamada “Limbo” onde cada uma dessas unidades consumidoras aguarda mais informações que permitam seu correto posicionamento na rede elétrica. Esse processo de retirada do “Limbo”, antes da construção da ferramenta, precisava de intervenção humana.

Com intuito de corrigir essas distorções que causam prejuízos financeiros, prejudicam seus indicadores de continuidade e causam problemas de diversas ordens aos clientes envolvidos é que optou-se por corrigir o chamado “problema do Limbo” de forma automatizada e utilizando a metodologia e a ferramenta de agentes inteligentes desenvolvida no projeto de P&D “Desenvolvimento de Ambiente Multiagente e de cultura organizacional para a construção de agentes inteligentes: Uma metodologia de monitoramento do Cadastro Técnico georreferenciado”.

A solução proposta engloba os dados disponíveis do sistema comercial, níveis de certeza e um algoritmo de otimização com lógica Fuzzy (ROSS, 2004), de forma a posicionar as unidades consumidoras do “Limbo” em seus respectivos circuitos elétricos e postes, na maioria dos casos, sem a necessidade de ida ao campo.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Agentes Inteligentes**

A metodologia está apoiada por uma arquitetura de software específica para a estruturação de um Ambiente Multiagente, onde as interações entre agentes inteligentes, dotados de capacidades específicas para atuar no ambiente de bancos de dados, serão capazes de produzir ações consistentes de atualização do cadastro técnico da Celg D.

A construção de um ambiente multiagente e, dentro deste, a criação de agentes inteligentes envolve um conjunto de conhecimentos do usuário que não são triviais como os que são exigidos para operar um sistema de informação tradicional. O desenvolvimento da lógica, a estruturação de um problema para que o mesmo seja tratado por agentes inteligentes, a estruturação de conhecimento através de regras e outras variáveis levam as equipes desenvolvedoras do projeto a encarar um desafio que é a criação de um ambiente capaz de resolver questões complexas, mas com um modelo de operação que seja o mais simples

possível, permitindo aos usuários finais desenvolver um conjunto de agentes os quais sejam capazes de solucionar problemas específicos.

Para um melhor entendimento da proposta, alguns conceitos para uma arquitetura multiagente são enunciados abaixo. Maiores informações podem ser vistas em PIRES (2013).

- Os Ambientes Multiagentes: os ambientes multiagentes se caracterizam pela existência de um contexto de suporte à sobrevivência dessas entidades (os agentes inteligentes), permitindo a eles atuarem de forma independente uns dos outros e realizarem tarefas específicas. É possível fazer uma analogia entre o Ambiente Multiagente e uma organização, onde seus funcionários atuam em conjunto, mas cada um desempenhando o seu papel específico (GASSER, 1992).
- Os agentes inteligentes: são usualmente compostos por uma máquina de inferência que tem a responsabilidade de pensar, isto é, relacionar símbolos (dados) de entrada (fatos, eventos etc.), gerando como resultado uma conclusão representada por um ou mais símbolos de saída (ações, advertências etc.) (RICH e KNIGHT, 1993).

A inteligência do agente é estruturada através de um conjunto de regras que norteia o comportamento da máquina de inferência, sendo que essas regras são normalmente representadas através de operadores lógicos ou eventualmente através da tradicional estrutura das linguagens de programação, o se-então-senão. O grau de flexibilidade de um agente é extremo, pois, são as regras armazenadas nele que definem seu comportamento. Essas regras são idealizadas e alimentadas pelo criador do agente (FERBER e GASSER, 1991).

## 2.2 Lógica Fuzzy

A lógica *Fuzzy*, ou lógica difusa, parte do princípio da incerteza e, desta forma, é mais complexa que a lógica booleana (zero ou um). Pode-se dizer que a incerteza nem sempre foi bem vista pela comunidade científica, sendo um estado indesejado e que deveria ser evitado a todo custo. A lógica *Fuzzy*, de fato, implica a ideia de que existem graus para todas as coisas, como "muito devagar", "um pouco devagar", "muito rápido", "um pouco rápido", ao contrário da lógica booleana que é capaz apenas de dizer se é rápido ou devagar. Para problemas que exigem extrema precisão, como o controle de um laser, por exemplo, a lógica difusa pode não ser a solução. Entretanto, para problemas humanos que não necessitem de tal precisão, a utilização desse tipo de lógica pode contribuir para resultados mais satisfatórios. É possível dizer que soluções muito precisas são caras, além de necessitarem de longos períodos de desenvolvimento e produção. Em casos em que a imprecisão seja aceitável, o cientista deverá analisar as condições e, se houver tolerância a isso, utilizar lógica difusa.

Entre os objetivos da *Fuzzy* está a de se aproximar da forma com que o raciocínio humano relaciona as informações, com o objetivo de buscar respostas aproximadas para os problemas. Sendo assim, tem-se foco dessa lógica a solução de problemas cujas informações presentes sejam incertas, tais como as incertezas levantadas nesta solução (AGUADO e CANTANHEDE, 2010).

Para melhor compreensão de como funciona a lógica *Fuzzy* e sua aplicação, no que tange ao comportamento não digital, isto é, o tratamento do “nem não nem sim, talvez talvez!”, podemos recorrer às Figuras 1 (a) a 1 (c) mostradas abaixo.

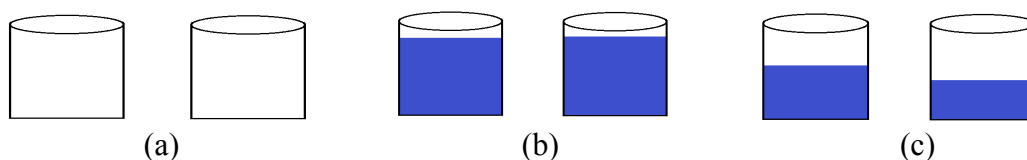


Figura 1 (a) Exemplo de copos vazios; (b) Exemplo de copos cheios e (c) Exemplo de copos com medida variada.

Para a Figura 1 (a) pode-se questionar se os copos estão cheios ou vazios e a resposta, naturalmente, é que eles estão vazios. Para a Figura 1 (b) a mesma pergunta pode ser feita e ter-se-á que os copos estão efetivamente cheios. Para os casos anteriores é fácil observar que não há qualquer incerteza com relação à decisão pois ou os copos estão vazios, ou estão cheios. Todavia, ao aplicar a mesma pergunta para a Figura 1 (c) não há uma resposta precisa. Pode-se afirmar, por exemplo, que o copo da esquerda está meio vazio e que o da direita está meio cheio, mas sem precisar essa resposta. Todavia, é possível estabelecer escalas de medidas que precisem o nível de maneira não discreta, flexibilizando a decisão. Para estes casos, a lógica *Fuzzy* apresenta-se como uma solução, remetendo-se ao caso da Tabela 1.

Não se deve confundir, entretanto, ao se dizer que a lógica *Fuzzy* raciocina de forma imprecisa. Na verdade, ela toma os valores imprecisos e afirma que eles são variantes entre zero e um, por exemplo, ao invés de apenas verdadeiro ou falso. Para trabalhar com essa lógica, utiliza-se os conjuntos *Fuzzy*, que são conjuntos de valores, ou predicados, que fazem parte de um conjunto sem limites definidos. Desta forma, a lógica difusa é um método de se raciocinar por meio de expressões lógicas que descrevem os conjuntos *Fuzzy*.

Sistemas *Fuzzy* são aproximadores universais. É possível dizer que essa lógica é bastante útil em dois casos: em situações envolvendo sistemas altamente complexos cujos comportamentos não são conhecidos, ou em situações onde uma solução aproximada, porém rápida, é garantida. A construção de um sistema *Fuzzy* é feita seguindo os seguintes passos:

1. Especificar o problema e definir variáveis linguísticas.
2. Determinar os conjuntos *Fuzzy*.
3. Deduzir e construir as regras *Fuzzy*.
4. Codificar os conjuntos *Fuzzy*, regras *Fuzzy* e procedimentos para realizar inferência difusa no sistema especialista.
5. Avaliar e ajustar o sistema.

Sendo assim, têm-se como primeira tarefa para construção de um sistema que utilize a lógica *Fuzzy* a análise do problema e a transformação dos dados em entradas variáveis. Após esta transformação deve-se determinar também as funções de pertinência, ou seja, os graus de certeza e incerteza, tal como exemplificado na Tabela 1.

A partir da aplicação desses conceitos, torna-se possível fazer inferências diversas, considerando não somente a totalidade ou formato discreto da lógica proposicional, mas também os níveis de certeza variados, também comuns no pensamento humano.

Apesar de a lógica *Fuzzy* não ser perfeita e ainda estar em constante aperfeiçoamento, acredita-se que para o caso de otimização geográfica ela se faz suficiente e é capaz de prover resultados satisfatórios.

### **3 DEFINIÇÃO DA TÉCNICA DE INTELIGÊNCIA APLICADA NA DECISÃO**

#### **3.1 Levantamento de requisitos**

São, principalmente, dois os bancos de dados envolvidos na solução do “problema do Limbo”: o cadastro técnico georreferenciado, chamado Sistema de Gestão Técnica – SGT e o cadastro comercial, chamado CBILL.

Frequentemente ocorrem erros em cadastros de unidades consumidoras em postes ou em circuitos elétricos incorretos. Quando ocorrem esses erros as unidades consumidoras são alocadas no “Limbo”, ou seja, uma referência fictícia do cadastro técnico que comporta unidades consumidoras que não possuem localização geográfica correta. Para que uma unidade consumidora apareça corretamente no SGT, ela precisa estar conectada, no mínimo, a um posto transformador, porém, preferencialmente a um poste.

O acúmulo desses erros provocou, ao longo dos anos, o envio de milhares de unidades consumidoras de toda a área de concessão da Celg D para o “Limbo”. Ao analisar-se uma a uma, percebe-se que não existe um padrão para os resultados, variando de unidades consumidoras situadas desde a capital até a zona rural.

Tendo isso em mente, é possível perceber o quão prejudicial à empresa é o “Limbo”. Em caso de manutenção corretiva solicitada por um cliente, cuja unidade consumidora esteja no “Limbo” o Centro de Operação da Distribuição - COD, por vezes, necessita executar trabalhos de pesquisa a fim de descobrir a localização dessa unidade. Além do custo de tempo e recursos humanos da tarefa, existe, ainda, o risco da equipe ser enviada a um local incorreto, gerando ainda mais prejuízo.

No entanto, sabe-se que os endereços das unidades consumidoras que têm leitura e faturamento normais estão disponíveis no CBILL. Desta forma, a utilização desses endereços na solução é imprescindível e, junto ao ambiente multiagente, engloba todas as informações e recursos necessários para a implementação da solução.

O ambiente multiagente é o responsável pela extração das informações dos sistemas existentes, bem como atualização pós execução do algoritmo de otimização, que será tratado também como uma ação de um agente com inteligência híbrida, isto é, regras simbólicas na forma de um sistema especialista (RUSSEL, 2010) (NEGNEVISTKY, 2005) e a realização de inferência via lógica Fuzzy. A fusão das informações constantes nos três sistemas, sendo essas utilizadas pela solução multiagente, visam a diminuição problema do “Limbo”. É importante notar em alguns erros mais complexos a visita em campo faz-se necessária.

A fim de melhorar as informações do cadastro técnico da Celg D propõe-se uma solução de otimização geográfica. Essa solução utiliza informações do CBILL, do SGT, bem como de um conjunto de agentes do sistema multiagente desenvolvido no citado projeto de P&D, sendo que, esse último também irá contribuir na extração e atualização das informações.

Para a construção da solução, é utilizada uma tecnologia de geolocalização como o Google Maps. Os endereços das unidades consumidoras do “Limbo” e que estão corretos no CBILL possibilitam a geração de um *buffer*, como apresentado na Figura 2.

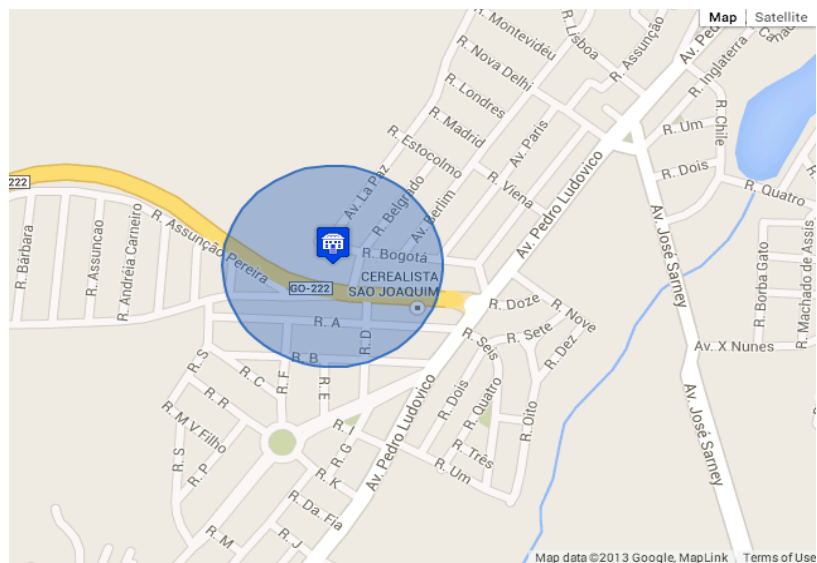


Figura 2: Buffer gerado pelo Google Maps.

A partir desse *buffer*, são encontrados os postos transformadores e postes no SGT. As coordenadas desses elementos de rede, que estão em coordenadas polícônicas, são convertidas para coordenadas aceitas pelo *Maps* seguindo uma série de regras de projeções. Em grande parte dos casos, o *buffer* gerado engloba muitos postes e, por vezes, mais de um posto transformador. Isso dificulta a identificação do poste e torna a alocação de uma unidade consumidora um processo complicado e sujeito a ação humana.

Por meio do *buffer* gerado e das informações extraídas por um dos agentes, é possível identificar todos os postos transformadores e postes da região, dando início ao processo de estudo para alocação nos locais corretos.

Tendo em mãos os pontos resgatados pela extração, parte-se para a segunda parte, que é a análise de opções para definição das regras e seus níveis de certeza. Esse trabalho é feito em forma de uma tabela de níveis de certeza. Essa tabela começou a ser construída a partir de reuniões com as equipes da FUNAPE e da Celg D. Diversas reuniões foram necessárias para se chegar a esse modelo preliminar de decisão e envolveu pessoas de níveis de supervisão, operacional e gerencial. A Tabela 1 apresenta o modelo de decisão que será o ponto inicial para a construção da solução do “Limbo”.

Tabela 1: Avaliação posicionamentos e de níveis de certeza

Situação provável	Desdobramento	Ação provável	Nível de certeza (precisão)
É possível fazer a localização geográfica a partir do endereço	1. e encontra-se um <b>único poste com posto transformador</b> próximo	Posicionar UC no poste próximo	10
	2. e encontra-se um <b>único poste</b> próximo (com baixa tensão)	Verificar se há seccionamento no poste e procurar a partir dos vizinhos (roteiro) se ambos estão no mesmo posto transformador, posicionar no poste	10

	3. e encontra-se <b>mais de um poste</b> próximo (com baixa tensão)	Verificar se há seccionamento no poste e procurar a partir dos vizinhos (roteiro). Havendo seccionamento, sugerir visita “in loco”, não havendo posicionar no poste mais próximo.	9
<b>Não é possível</b> fazer <b>localização geográfica</b> a partir do endereço	1. mas têm-se a localização geográfica das UCs anterior e posterior (roteiro de leitura), com Quadra e poste comum entre todos esses	Não havendo seccionamento no poste, posicionar no poste	10
	2. mas têm-se localização geográfica das UC's anterior e posterior (roteiro de leitura), com poste comum entre esses	Não havendo seccionamento no poste, posicionar no poste	10
	3. mas têm-se localização geográfica das UC's anterior e posterior (roteiro de leitura), cada UC com poste diferente, no entanto com vizinhos em posto transformador comum	Posicionar no posto transformador	10
	4. mas têm-se localização geográfica das UC's anterior e posterior (roteiro de leitura), cada UC com poste diferente, no entanto com vizinhos em postos transformadores diferentes	Sugerir visita “in loco” para decidir qual posto transformador posicionar	10
	5. As duas UCs anteriores e posteriores no roteiro de leitura também encontram-se no limbo	Sugerir visita “in loco” com maior prioridade (possível classificação de prioridade A/B/C/D/E, sendo A com maior prioridade)	10

Por meio dessa tabela e do levantamento das situações prováveis, desdobramentos, ações prováveis e níveis de certeza, é possível desenvolver um algoritmo que irá otimizar e promover o reposicionamento de cada unidade consumidora do “Limbo”. O algoritmo irá obedecer cada uma das regras e será capaz de reduzir o *buffer* completamente até que seja possível alocar a unidade consumidora em um posto transformador e/ou poste ou, em último caso, gerar uma tabela com os casos que não solucionados, indicando ações mais precisas para visitas “in loco” pelas equipes de cadastro técnico. Observe que mesmo para estes últimos casos, também haverá uma otimização, uma vez que haverá um número reduzido de ações e os casos serão elucidados pelo sistema multiagente que indicará qual o problema de posicionamento, antes completamente desconhecido.

O problema apresentado provocou a equipe a novos desdobramentos de estudos com o objetivo de definir qual a técnica de Inteligência Artificial mais adequada para resolver o problema. Há que se observar, entretanto, que trata-se de uma tabela que apresenta os problemas e soluções mais triviais e identificadas durante as reuniões entre FUNAPE e Celg D. Todavia, acredita-se que com a aplicação da solução, novas regras de reposicionamento surgirão e deverão ser integradas da forma mais simples possível ao modelo multiagente. Diante de tal situação, a equipe FUNAPE entendeu que a utilização de lógica *Fuzzy*, ou lógica difusa é a mais indicada para utilização no modelo multiagente híbrido, pois ela é capaz de interpretar e trabalhar com incertezas. As incertezas da lógica *Fuzzy* podem ser mais favoráveis e ela pode ser capaz de fornecer resultados mais satisfatórios, visto que trata-

se de posicionamento geográfico com níveis de incertezas não discretos, ou seja, com variações de níveis de decisão, tão comum ao pensamento humano.

Espera-se como resultado, então, que o algoritmo seja capaz de executar as regras levantadas para cada uma das unidades consumidoras e, de forma automatizada, consiga reduzir a área de *buffer* a ponto de conseguir alocá-las em seus postos transformadores e/ou postes corretos.

No entanto, é importante ressaltar que, em certos casos, o algoritmo não é capaz de identificar apenas um posto transformador ou poste. Para esses casos, como já mencionado, relatórios devem ser gerados para que sejam analisados por colaboradores especializados e, provavelmente, uma visita de campo far-se-á necessária.

Tem-se como limitação do algoritmo, também, o fato de que ele será mais eficaz em grandes cidades e cidades médias, pois, as muito pequenas não dispõem de um sistema de CEP completo. Mesmo assim, para esses casos o sistema disponibiliza informações que poderão ser utilizadas, tais como razão (data de leitura), rotas e roteiros utilizados pelas equipes de medição e entrega de faturas. Ainda assim, para áreas rurais o algoritmo fica impossibilitado de ser executado, visto que a localização do endereço torna-se impossível a priori. Soluções para essas limitações fazem parte de estudos futuros.

O algoritmo funciona como um agente do sistema multiagente desenvolvido no projeto de P&D e consistirá de um serviço automatizado que é executado constantemente de acordo com as regras estabelecidas, sendo que as mesmas podem ser mudadas ao longo da execução.

As informações a serem utilizadas pelo algoritmo deverão ser providas por outros agentes do mesmo sistema, os quais efetuam a extração de dados em um processo de mineração de dados.

Após a extração, análise e otimização feita pelo algoritmo, as informações são atualizadas na base de dados da empresa e, como resultado, tem-se uma diminuição satisfatória do número de unidades consumidoras no “Limbo”.

Como resultado esperado feito após análise inicial, estipula-se que a diminuição de unidades consumidoras do “Limbo” no primeiro período poderá chegar a 50%. No decorrer das ações e refinamento das informações contidas no sistema georreferenciado, esse número pode aumentar.

### 3.2 Resultado obtidos

Com o objetivo de avaliar a aplicabilidade do modelo proposto a equipe do projeto preparou um ambiente computacional de testes, utilizando um computador PC de 2GHz com 500GB de disco e 8GB de memória principal. Neste computador foram instalados o banco de dados Oracle, o servidor web *TomCat* e todo o ambiente de desenvolvimento Java, incluindo a IDE Eclipse. Além disso, o ambiente multiagente foi instalado nesta máquina servidora para a estruturação da solução multiagente.

Depois de preparado o ambiente, a equipe identificou as UCs do Limbo para então realizar os testes. De posse dessas UCs, a equipe retirou uma amostra de 2.500 UCs para o estudo de caso.

Nos testes iniciais a equipe aplicou as regras que tratam somente do cruzamento de informações entre os sistemas comercial e o técnico. As premissas que envolvem a utilização das soluções do Google estão em fase de conclusão no ambiente. Portanto, utilizando-se somente a situação provável “**Não é possível fazer localização geográfica a partir do endereço**”, conforme indicado na Tabela 1, o sistema mostrou-se capaz de reposicionar 1.304 unidades das 2.500 amostradas, totalizando 52,16%.

Diante desses resultados e considerando ainda que nem todas as regras foram implementadas, espera-se que esse número possa ser melhorado de forma significativa. Ainda

assim, observa-se que esse resultado supera a expectativa inicial prevista para o projeto, que era de 50%.

### 3 CONCLUSÃO

Os estudos e levantamentos feitos para construção da solução do “problema do Limbo” permitiram a identificação da possibilidade de reposicionamento de UCs do “Limbo” sem a necessidade de visita “in loco” para um percentual significativo de unidades consumidoras. Por outro lado, a descrição dos problemas e o levantamento de requisitos desafiou a equipe de pesquisa na identificação de soluções de Inteligência Artificial que pudessem atender aos aspectos levantados, remetendo à construção de modelos multiagentes híbridos. A análise do problema indicou que a lógica *Fuzzy* pode ser a mais adequada para tratar o problema do limbo, pois permite fazer inferências mesmo que os níveis de certeza de decisão sejam variados. Sendo assim, apresentou-se os encaminhamentos de decisões e a tecnologia a ser adotada em parte dos agentes do sistema.

Uma base cadastral de redes confiável é premissa básica para uma gestão técnica de distribuição com qualidade. A preocupação, não só da Celg D, mas de todas as concessionárias de distribuição é como manter seu cadastro técnico sempre atualizado e com o mínimo de inconsistências.

A metodologia e a ferramenta computacional descritas neste artigo apresentam-se como aliadas poderosas no sentido de reduzir consideravelmente as falhas contidas no banco de dados, e não só nesse aspecto de vinculação de unidades consumidoras à rede elétrica, uma vez que tais ferramentas podem ser adaptadas para solução de problemas de outras naturezas que afligem o cadastro técnico.

Esse esforço, em última análise, representa uma considerável melhoria na qualidade do serviço prestado pela Celg D.

### REFERÊNCIAS

AGUADO, A. G., CANTANHEDE, M. A. **Lógica Fuzzy**. Congresso UNICAMP, Limeira, 2010.

FERBER, J. and GASSER, L. **Intelligence artificielle distribuée**, In: INTERNATIONAL WORKSHOP ON EXPERT SYSTEMS & THEIR APPLICATIONS, 10, Avignon. Cours n. 9. France, 1991.

GASSER, L. **Boundaries, identity and aggregation: plurality issues in multiagent systems**, In: WERNER, E.; DEMAZEAU, Y. (Eds.). **Decentralized AI 3**. Amsterdam: North-Holland, 1992.

NEGNEVISTKY, M. **Artificial Intelligence: A Guide to Intelligent Systems**, 2 ed., Pearson Education Limited, England, 2005.

PIRES, K. G., R. S. P. e. M. T. C. **Relatório do Projeto P&D da CELG - Uma Proposta de um Ambiente Multiagente Dinâmico**, Relatório Técnico. NEPE/UFG, 2013.

RICH, E. and KNIGHT, K. **Inteligência Artificial**, São Paulo: Makron Books, 1993.

ROSS, T. J. **Fuzzy Logic With Engineering Applications**, 2 ed., John Wiley & Sons Ltd, England, 2004.

RUSSEL, S.; Norvig, P. **Artificial Intelligence: A Modern Approach**, 3 ed., Pearson Education Limited, United States of America, 2010.