

DOMÓTICA INTELIGENTE: AUTOMAÇÃO BASEADA EM COMPORTAMENTO

MARCELO TAKIUCHI, ÉRICA MELO E FLAVIO TONIDANDEL

Centro Universitário da FEI – UniFEI

Av. Humberto de A. Castelo Branco, 3972

09850-901 - São Bernardo do Campo - SP - Brasil

E-mails: mtakiuchi@uol.com.br; erica_credilatina@hotmail.com; flaviot@fei.edu.br

Abstract— Research in the Building Automation field has been increased over the last years. Intelligent Domotics, as the Intelligent Building Automation has been named, intends to extend the automation concept with the introduction of intelligent decisions, like resource management (water and power consuming, temperature, etc) and fault control. However, the most research and modern systems have been focused on static domotics, which concept is: inhabitants must adapt their lives to the automation system (automation-based architecture). In this paper, it is proposed an inversion of this concept: the Automation system must adapt their rules to fit the inhabitants behaviors (behavior-based architecture). The system proposed, called ABC, takes use of a learning technique of the Artificial Intelligence field to create and update their automation rules, redefining the intelligent domotics concept.

Keywords— Buildings Automation, Intelligent domotics, Learning, Artificial Intelligence

Resumo— As pesquisas na área de automação predial, ou domótica, vêm crescendo nos últimos anos, tanto em nível nacional quanto internacional. A domótica inteligente, atualmente, visa estender o conceito de automação predial com a inserção de tomadas de decisão inteligentes, como por exemplo, gerenciamento de recursos (energia, água, etc) e controle e solução de falhas. Entretanto, a grande maioria das pesquisas e sistemas atuais focam em sistemas de domótica estáticos, onde o conceito envolvido visa que o habitante se adapte à automação implantada (arquitetura baseada na automação). Neste artigo, propõe-se a inversão desse conceito, onde os sistemas de automação devem se adaptar ao comportamento dos habitantes, i.e., uma arquitetura baseada no comportamento. Para isto, apresenta-se uma arquitetura de um sistema de automação, chamado ABC (Automação Baseada em Comportamento) que possui técnicas de aprendizado para criação e atualização automática de regras de automação predial, redefinindo assim o conceito de domótica inteligente.

Palavras-chave— Automação predial, domótica inteligente, aprendizado, inteligência artificial

1. Introdução

A automação predial, também conhecida como domótica, vem crescendo nos últimos anos. Conceitos como casas e prédios inteligentes estão sendo definidos pela aplicação de mecanismos automáticos e, algumas vezes, pela aplicação de técnicas de Inteligência Artificial.

As atuais pesquisas na área de automação predial se concentram, em sua maioria, na rede de comunicação entre de sensores e atuadores, nos protocolos usados nessa comunicação e ainda em sistemas de gerenciamento e otimização do uso e aproveitamento de recursos (Mariotoni; Andrade, 2002). A arquitetura de tais sistemas geralmente é dedicada e centralizada.

O conceito de domótica inteligente (DI) deve ir além de um sistema centralizado e dedicado e seu foco deve ir além dos sensores e atuadores. Diferentemente, deve ser uma arquitetura descentralizada e adaptativa às necessidades dos habitantes e às resoluções de problemas e gerenciamento de recursos. Atualmente, os pesquisadores que se interessam por domótica inteligente, muitas delas com conceitos limitados quanto à inteligência que tais sistemas devem possuir. O conceito de domótica inteligente em (Mota, 2003), por exemplo, se resume ao gerenciamento automático de recursos (temperatura, energia, etc) e aplicação de técnicas de IA na predição e resolução de falhas. O

conceito em aplicações industriais e comerciais de casas e prédios inteligentes é ainda mais limitado, se resume apenas ao uso de sistemas automáticos sem qualquer característica de adaptabilidade ou mesmo de predição de falhas (e.g. assento sanitário aquecido, interruptor automático de luz, etc).

A Domótica inteligente deve, por sua vez, analisar os dados obtidos pelos sensores de modo a adaptar suas regras de automação ao comportamento dos habitantes. Isto se faz necessário, pois o comportamento dos seres humanos muda ao longo do tempo, bem como cada indivíduo possui sua própria preferência, algumas gostam de temperaturas mais quentes do que outras, ou mesmo outras gostam de luminosidade menor do que outras.

Este artigo foca na definição e modelagem de um sistema de domótica inteligente adaptativo, chamado sistema ABC (Automação Baseada em Comportamento) através do uso de técnicas de aprendizado no ajuste das regras de automação ao comportamento das pessoas que habitam uma casa ou prédio inteligente. Esse ajuste se dá de forma automática e sem intervenção do indivíduo, i.e., nenhuma configuração manual é necessária.

O artigo se divide da seguinte forma: a próxima seção discutirá os trabalhos correlatos da área; a seção 3 introduzirá o sistema ID3 usado como sistema de aprendizado e apresentará a definição do sistema de automação inteligente proposto ressaltando como

capturar o comportamento dos habitantes e como redefinir as regras de automação; a seção 4 apresenta a implementação de teste realizada e a seção 5 finalmente conclui o artigo.

2. Domótica inteligente

As pesquisas na área de automação predial inteligente, também chamada de domótica inteligente, são recentes. Por este motivo, o conceito de domótica inteligente, por vezes, se confunde com o conceito de automação (ou simplesmente domótica). Deve-se por domótica inteligente o processo de automação que incorpore algum mecanismo automático de tomada de decisão baseada em técnicas de Inteligência Artificial.

Poucos trabalhos de pesquisa focam em domótica inteligente. O trabalho de Rutishauser and Schafer (2002), por exemplo, modela um sistema de domótica inteligente baseada em agentes e lógica fuzzy. Entretanto, a parte principal do projeto, que é tornar o sistema adaptativo ao ambiente e ao comportamento dos habitantes, ainda está sendo desenvolvido. Os autores apontam o uso de técnicas de Aprendizado por Reforço (*Reinforcement Learning*) (Michell, 1994) para tal finalidade, em trabalho interessante em domótica inteligente, que integra teoria e aplicação, é (Brooks, 1997), do *MIT Artificial Intelligence Lab*. Tal trabalho é restrito à uma sala de reuniões onde o sistema de domótica inteligente ajuda as pessoas atingirem seus objetivos, utilizando para isto reconhecimento de fala e percepção visual (visão computacional). A Ada (Eng et al, 2002) foca em um outro paradigma de domótica inteligente. O sistema trata o ambiente como um organismo artificial que possui estados emocionais. O objetivo do sistema Ada é modificar dinamicamente o ambiente de modo a melhorar sua funcionalidade e qualidade. Essa modificação é feita por diálogos com os visitantes.

Neste artigo, assim como os demais artigos acima citados, é proposta uma inversão de responsabilidades no gerenciamento de uma casa inteligente. Ao invés dos habitantes terem de se adaptar ao funcionamento pré-programado de uma edificação automatizada e, para isto, terem que mudar seu modo de vida ou de trabalho, o sistema é que se preocupa em mudar sua atuação de modo a satisfazer às necessidades e se adaptar ao comportamento dos habitantes da edificação. Essa diferença de foco e de paradigma é a principal diferença existente entre domótica e domótica inteligente. Este artigo se diferencia dos demais por abordar a parte de adaptação do sistema de domótica através de técnicas de aprendizado e por ser um modelo genérico e não específico a apenas um tipo de ambiente e, ao invés de se adaptar por diálogos ou interatividade, se adaptar com base no comportamento dos habitantes.

3. Automação Baseada em Comportamento

A automação baseada em comportamento proposta neste artigo concentra-se no algoritmo de aprendizado conhecido como ID3 (Michell, 1997). Este algoritmo é o responsável pelo processo de adaptação do sistema de automação.

A escolha do algoritmo ID3 se deu pela sua simplicidade e seu alto poder de generalização de regras. É bem conhecido na literatura que tal algoritmo possui certas limitações, mas essas não afetam nossos testes preliminares apresentados neste artigo.

3.1. O algoritmo ID3

O algoritmo ID3 (Michell, 1997) é uma técnica de aprendizado que consiste na indução de uma descrição geral a partir de um conjunto de exemplos, chamado de conjunto de treinamento. O ID3 é um dos algoritmos mais simples que generaliza regras em formato de uma árvore de decisão. Cada regra é um conjunto de instâncias que, por sua vez, possui certos atributos discretos.

Um exemplo de instância pode ser *luz_da_sala* cujos atributos são acesa ou apagada. Outro exemplo pode ser a instância *tempo*, cujos atributos são manhã, tarde e noite.

Para criar a árvore de decisão, o algoritmo ID3 utiliza, primeiramente, uma fórmula que calcula a entropia do conjunto de treinamento considerado. A equação da entropia é a seguinte:

$$Entropia(S) = \sum -p_i \log_2 p_i \quad (1)$$

Onde:

S = conjunto de treinamento.

P_i = proporção de exemplos cuja valor da resposta é *i*

Com a equação 1 é possível determinar o quanto há de exemplos para cada atributo de resposta. A tarefa do sistema ID3 é classificar cada atributo de cada instância relacionando com cada atributo da resposta. Isto é, o sistema deve escolher um atributo como raiz da árvore de decisão que possa separar, da melhor maneira possível, o conjunto de treinamento inicial em outros conjuntos de treinamento menores porém mais homogêneos. Para isto, o sistema usa a equação do ganho:

$$Gain(S,A) = Entropia(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} Entropia(S_v) \quad (2)$$

Onde *S_v* = é o conjunto de instâncias cujo valor do atributo *A* é *v*.

O ID3 começa por escolher uma instância, aquela que produzirá o maior ganho (equação 2), a partir da qual efetua uma partição dos dados, dividindo os exemplos de treinamento conforme cada atributo dessa instância.

O melhor ganho de uma instância significa que a mesma pode dividir o conjunto de treinamento em sub-conjuntos mais homogêneos do que outras instâncias que possuem valor de ganho menor. Com isso o ID3 consegue decidir quais serão os nós e a raiz da árvore de decisão.

A escolha de cada instância é aplicada recursivamente para cada sub-conjunto de treinamento, até que os exemplos contidos nesse sub-conjunto pertençam todos à mesma classe de um atributo de resposta. Sendo r o atributo dessa resposta, o ramo dessa árvore forma uma regra de decisão para que r seja válida ou considerada.

A construção da árvore só chega ao fim quando todos os exemplos estiverem convenientemente desclassificados não havendo possibilidade de ambigüidade.

3.2. Arquitetura do sistema ABC

O ID3 aplicado ao sistema de automação implica nas seguintes definições:

Definição 1 (Conjunto de Treinamento): Cada conjunto de treinamento no sistema ID3 representará um atuador do sistema de automação.

Definição 2 (Resposta): Cada atributo de resposta de um conjunto de treinamento no sistema ID3 será um estado do atuador cujo conjunto de treinamento representa.

Definição 3 (Instâncias e atributos): Cada sensor de um sistema de automação é uma instância cujos atributos formam o *status* deste sensor.

Com as definições acima, propõe-se uma arquitetura, que consiste de atuadores, sensores e um banco de regras criados pelo ID3, apresentada na Figura 1. Nessa figura, os sensores₁ formam o grupo de sensores que agem diretamente com os atuadores e que podem sofrer interferência humana em seu *status*. Exs.: interruptor luz e interruptor Ar-condicionado. Já os sensores do grupo sensores₂ agem sem a interferência humana exs.: Luminosidade e Temperatura externa.

Na arquitetura proposta para o sistema (fig 1), o controlador monitora constantemente os sensores de cada ambiente, formados pelo grupo de sensores 1 e 2, verificando se existe alguma regra de automação armazenada em seu banco de dados ou nas regras de segurança correspondente ao *status* dos sensores.

Este controlador verifica, preferencialmente e com base nos sensores, se alguma situação de segurança, como incêndio ou vazamento de gás

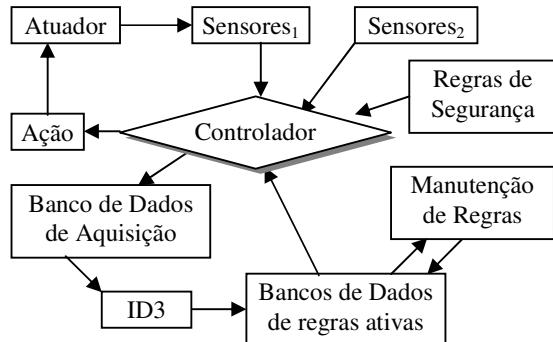


Figura 1 – Arquitetura do sistema ABC

Por exemplo, está ocorrendo para assim acionar uma ação pré-determinada pelas regras de segurança. Estas regras de segurança são estáticas e não podem ser alteradas no modelo proposto.

Não sendo nenhuma situação de segurança, o controlador passa a verificar se existe alguma regra no banco de dados que deve ser acionada pela configuração de *status* dos sensores. Se existir alguma regra, a solução a ser adotada é enviada ao atuador que executa a ação. O processo se repete toda vez que algum sensor tiver seu *status* modificado.

3.3. Aquisição de dados de comportamento

As regras para o conjunto de treinamento do algoritmo ID3 são geradas a partir de um banco de dados de todas as ações tomadas pelo habitante. Por exemplo, sempre que o *status* de um sensor do conjunto de sensores₁ mudar, e esta mudança não for ocasionada pela ação de uma regra, ficará configurado para o sistema um novo comportamento a ser aprendido. Para isto, guarda-se no banco de dados uma regra de ação com o *status* de todos os sensores (1 e 2) e com a ação que resulta na mudança de *status* detectada. As regras a serem armazenadas no banco de dados são gravadas pelo sistema a partir de um código, que consiste da combinação dos sensores do ambiente analisado.

Cada sensor possui valores discretos, chamados de *status*, e cada regra representa uma regra *if..then* para tomada de decisão. O formato geral das regras é o seguinte:

If (sensor₁ = Vx₁ and ... and sensor_n = Vx_n)
then ação_y ∈ A.

Onde

- ◆ A = Conjunto de Ações possíveis;
- ◆ Sensor_i = sensores pertencentes ao grupos de sensores 1 e 2;
- ◆ Vxi = Valor de cada sensor conforme tabela discreta de *status*;

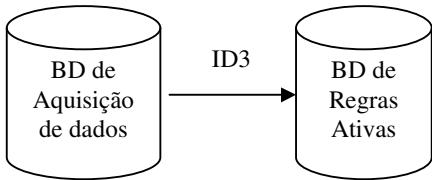


Figura 2 – Banco de Dados usados no sistema ABC

Cada atuador controlado pela casa possui sua tabela de banco de dados com todas as regras definidas. Assim sendo, toda vez que uma situação de valores é verificada pelo sistema, este pesquisa em todos os bancos de dados a ação a ser tomada para cada atuador da casa.

3.4. Adaptação ao comportamento

O sistema trabalha com banco de dados que armazemam a configuração das regras de automação. Conforme figura 2, existem dois conjuntos de banco de dados (BD): um chamado BD de aquisição de dados e outro BD de regras ativas.

Cada conjunto possui uma quantidade de tabelas tal que cada atuador tem seu próprio conjunto de dados, i.e., cada BD possui uma tabela de dados para cada atuador.

O BD de aquisição de dados é alimentado por informações do comportamento do habitante. Toda vez que o habitante toma uma ação é criado uma nova regra e armazenado no BD de aquisição de dados. O algoritmo ID3 é acionado toda vez que o Banco de dados de um certo atuador alcançar um certo número de regras genéricas que garanta uma boa amostragem do comportamento.

O acionamento do ID3 transforma regras genéricas de comportamento em regras ativas, específicas do comportamento do habitante. O ID3, por si só, determina os *status* dos sensores relevantes e responsáveis pelo comportamento analisado.

As novas regras geradas pelo ID3 são armazenadas no BD de regras ativas e são utilizadas normalmente pelo controlador. Um exemplo de geração de regras via ID3 é apresentada na figura 3.

Pela figura 3, as seguintes regras são adicionadas no banco de dados de regras ativas do atuador a que elas respondem:

Regra 1: **If** (Sensor1=V1 **and** Sensor2=V4 **and** Sensor3=V6) **then** ação1

Regra 2: **If** (Sensor1=V1 **and** Sensor2=V5 **and** Sensor3=V7) **then** ação2

Regra 3: **if** (Sensor1 = V2 **and** Sensor2 = V4) **then** ação 3

Regra 4: **If** (Sensor1 = V3) **then** ação4

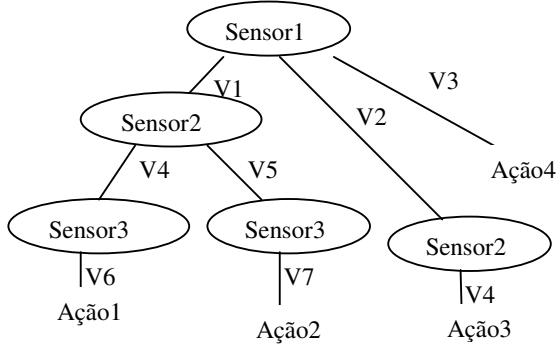


Figura 3 – Exemplo de árvore de decisão gerada pelo ID3 aplicado à automação

As ações mencionadas nas regras geradas pelo ID3 podem ser as mesmas ou não. Estas foram inseridas de forma ilustrativa na figura 3 indicando que para a situação apresentada, uma ação é tomada.

Para o conjunto de regras geradas pela árvore de decisão, podemos observar que com o acionamento de apenas um sensor (regra 4), é possível fazer com que uma ação seja tomada, neste caso o sensor 2 e o sensor 3 não são relevantes para a ação tomada. Na regra 3 temos que o sensor 3 não é relevante para que uma ação seja tomada e nas regras 1 e 2 todos os sensores são relevantes para ação.

Além de gerar novas regras, o sistema deve ainda manter sua base de regras atualizada e consistente. Para isto, um módulo de manutenção das regras deve ser adicionado ao sistema.

3.5. Manutenção da base de regras

A partir das soluções adotadas pelo seu habitante, o sistema ABC gera novas regras, podendo assim criar algumas que o sistema avalia serem corretas, mas que o habitante nem cogitou ensinar.

O sistema é programado para que as regras que entraram em desuso pelo sistema sejam excluídas automaticamente a partir do momento em que atingem um determinado tempo a ser definido pelo habitante. A base de dados armazena a data da última utilização da regra, e caso a regra não seja utilizada por um determinado período de tempo estipulado pelo habitante, ela é removida automaticamente.

Caso o sistema gere uma regra que não agrade o habitante, ele pode acessar o sistema e eliminar a regra manualmente, esta regra vai para um banco de dados de regras rejeitadas e o sistema ao gerar novas regras ID3, não cria esta regra novamente. Caso o habitante queira que a regra passe a existir, ele terá de retirar a regra da tabela de regras rejeitadas.

Outro ponto importante na manutenção de regras é deixar a base de dados consistente. Isso significa evitar *loopings* acidentais. Isso ocorre quando uma regra A modifica o *status* de um sensor

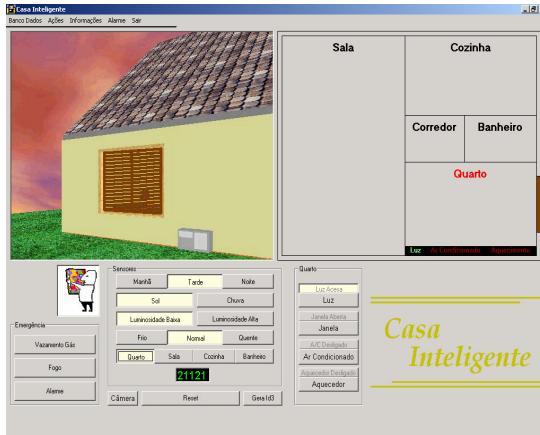


Figura 4 – Simulador de uma Casa Inteligente para testes do modelo proposto neste artigo

Essa modificação aciona outra regra B, que ao modificar o *status* de outro sensor, acaba acionando a regra A novamente.

Esse tipo de *looping* de recorrência requer uma solução complexa principalmente quando o *looping* acontece entre diversas regras, e não apenas duas.

A detecção de tal problema é verificada pelo seguinte procedimento: se uma regra A, após ter sido acionada, volta a ser acionada após a mudanças de *status* dos sensores do grupo $sensores_1$ ocasionados única e exclusivamente por outras regras. Entretanto, a solução do problema não é trivial e detectar qual regra deva ser modificada para evitar o *looping* é uma tarefa complexa. Por isso, tal solução não é tratada neste artigo e será abordada e pesquisada em trabalhos futuros.

4. Testes empíricos

Para verificação do modelo criado, foi desenvolvido um simulador de uma casa inteligente. Para a simulação, considerou-se os seguintes sensores:

$$\begin{aligned}
 \text{Período} &= \{\text{Manhã, Tarde, Noite}\}; \\
 \text{Tempo} &= \{\text{Sol, Chuva}\}; \\
 \text{Luminosidade} &= \{\text{Alta, Baixa}\}; \\
 \text{Temperatura} &= \{\text{Frio, Normal, Quente}\};
 \end{aligned}$$

Para o uso do sistema ID3, não definiu-se nenhuma quantidade determinada de regras para sua aplicação de forma automática, este pode ser acionado manualmente por um botão.

A figura 4 mostra a tela do simulador e a figura 5 apresenta o banco de dados de regras ativas, alimentado pelo algoritmo ID3 após análise da base de dados de regras de comportamento. Para este ambiente de simulação, botões permitem a configuração dos *status* dos sensores.

Tabelas Auxiliares Gera ID3								
Sar								
Código	regra	situação	sensores	período	ceu	luminosidade	temperatura	status luz
115	201	11221	Manhã	Llimpo	Alta	Normal	0	
117	202	31121	Noite	Llimpo	Baixa	Normal	1	
118	203	21111	Tarde	Llimpo	Baixa	Frio	1	
119	204	21221	Tarde	Llimpo	Alta	Normal	1	
120	205	21231	Tarde	Llimpo	Alta	Quente	1	
121	206	22121	Tarde	Chuva	Baixa	Normal	1	
122	0	10001	Manhã	Nulo	Nulo	Nulo	0	
123	0	20001	Tarde	Nulo	Nulo	Nulo	1	
124	0	30001	Noite	Nulo	Nulo	Nulo	1	
125	0	20011	Nulo	Chuva	Nulo	Nulo	1	
126	0	101	Nulo	Nulo	Baixa	Nulo	1	
127	0	11	Nulo	Nulo	Nulo	Frio	1	
128	0	31	Nulo	Nulo	Nulo	Quente	1	
129	0	21201	Tarde	Llimpo	Alta		1	

Figura 5 – Exemplo de tabela com regras ativas e ações para a Luz do quarto (*status da luz* \rightarrow 1 acender e 0 apagar). O *status* Nulo significa que o *status* do sensor não é aplicado naquela regra.

O processo de simulação é extremamente simples e serve apenas para validação do modelo apresentado. Após ser carregado com regras *default*, pode-se simular situações de sensores e aplicar certas ações para serem aprendidas e que não estavam previstas no sistema

Essas situações simuladas são armazenadas na tabela da figura 5, onde a situação dos sensores e o *status* da ação executada são armazenados.

Após um período, o ID3 é acionado e novas regras são criadas com base nas situações simuladas e armazenadas. Perceba que as regras não são, necessariamente, as mesmas sempre. Sistema não aprende por repetibilidade, mas sim por generalização de regras e situações inclusive diferentes.

Como era esperado, nos testes simulados, o sistema conseguiu aprender regras novas e aplicá-las todas as vezes que certa configuração de alguns sensores fosse atingida. Essas regras passam a fazer parte do sistema de automação

Embora o ambiente de simulação seja simples, permitiu verificar que o modelo funciona e pode, portanto, ser estendido para aplicações reais e muito mais complexas.

5. Conclusão e trabalhos futuros

Este artigo visou apresentar o modelo de um sistema de automação predial e residencial, conhecido como domótica, que adapta e remodela suas regras conforme o comportamento do habitante (usuário) do sistema.

Este novo paradigma de automação predial e residencial é definido como domótica inteligente, pois evita que os usuários tenham que modificar seu modo de vida ou trabalho.

O modelo de sistema apresentado neste artigo se baseia em técnicas de aprendizado da área de Inteligência Artificial para determinar novas regras de automação. Este sistema, altamente reativo, permite

que o processo de automação seja flexível para que se adapte ao comportamento do usuário.

Entretanto, tal modelo necessita de ajustes e aplicações em campo para determinar a quantidade mínima de regras necessária para criar uma árvore de decisão precisa. Essa quantidade depende da quantidade de sensores e atuadores, bem como da probabilidade de mudança do *status* de cada sensor. Outro problema é como se comportar e criar regras para um ambiente com diversos usuários de comportamentos diferentes. Uma técnica é trabalhar com “processo de votação”, onde o comportamento da maioria – ou repetido mais vezes – se sobrepõe ao comportamento da minoria. Estes dois pontos em aberto serão investigados em trabalhos futuros.

Além de tudo isso, outras técnicas e algoritmos de aprendizado poderão ser aplicados no futuro, e seus resultados podem ser confrontados com o algoritmo ID3. Permitindo assim encontrar um algoritmo que permita uma adaptabilidade quando aplicado a sistemas de domótica inteligente.

Agradecimentos

Agradecemos a Claudia V. Tavares e a Daniel Witkowski pela ajuda na realização deste trabalho.

Referências Bibliográficas

- Brooks, A.R. (1997). The intelligent Room Project. In: 2th International Cognitive Technology Conference (ICT'97). Proceedings. Aizu, Japão, 1997.
- Eng, K et al. (2002). Ada: Constructing a synthetic organism. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002). Proceedings. Lausanne. Suíça, 2002.
- Mariotoni C. A. & Andrade E. P. (2002). Descrição de Sistemas de Automação Predial Baseados em Protocolos PLC Utilizados em Edifícios de Pequeno Porte e Residências. CTAI - Revista de Automação e Tecnologia da Informação. Volume 01 nº 1 Janeiro/Junho 2002
- Michell, T. (1997). Machine Learning. McGraw Hill, 414 p. 1997.
- Mota, J. A. A. J. (2003). Casas Inteligentes. EditoraCentro Atlântico Ltda. Lisboa. Portugal. 145 p. 2003.
- Rutishauser, U. & Schafer, A. (2002). Adaptive Building Automation: A multi-Agent approach. Research project. University of Applied Science Rapperswil. Suíça, July 2002. Disponível em: <http://www.easc.ch/aha/>