

# Domótica Inteligente: Automação Residencial baseada em Comportamento

Julio André Sgarbi, Flavio Tonidandel

Centro Universitário da FEI – UniFEI  
Av. Humberto de A. C. Branco, 3972 - CEP:09850-901 - São Bernardo do Campo – SP - Brasil  
[unimjsgarbi@fei.edu.br](mailto:unimjsgarbi@fei.edu.br), [flaviot@fei.edu.br](mailto:flaviot@fei.edu.br)

**Resumo.** Muitos avanços têm acontecido em automação residencial, entretanto pouco destaque é dado à automação residencial baseada no comportamento dos habitantes. Este artigo descreve o estudo de um sistema cujo objetivo é observar e aprender regras em uma casa de acordo com o comportamento de seus habitantes. O sistema proposto, Sistema ABC+ (Automação Baseada em Comportamento) utiliza o conceito de aprendizado com regras de indução.

**Palavras-chave:** Domótica Inteligente, Aprendizado, Inteligência Artificial.

## 1 Introdução

Existe uma série de trabalhos que abordam o tema automação residencial. Os avanços tecnológicos e a busca por conforto e segurança vêm fazendo com que a cada dia as residências tenham mais e melhores sistemas automatizadores.

Algumas vezes estes sistemas são implementados com técnicas de inteligência artificial, entretanto a maioria deles possui apenas mecanismos automáticos [4].

A principal ênfase destes sistemas está no gerenciamento de recursos, segurança, conforto e atualmente em entretenimento [7].

Muitos dos trabalhos publicados alegam que seus sistemas são inteligentes. Alguns deles detectam eventos como a presença dos habitantes e suas ações, ou possuem sensores que conseguem capturar mudanças nas condições do ambiente; a estes eventos são aplicadas reações que estão previamente configuradas [4]. O conceito de inteligência deve ir além de automatizar ou simplesmente aplicar regras pré-estabelecidas; os sistemas inteligentes devem interagir com os habitantes da casa e aprender com seus comportamentos.

Automação residencial deve evoluir para o conceito de Domótica Inteligente, onde se entende que dados obtidos dos sensores da casa devem ser avaliados de modo a adaptar as regras de automação do ambiente ao comportamento dos habitantes [10].

Os seres humanos estão em constante mudança; o que é uma regra ou rotina hoje, amanhã pode não ser. Os hábitos, horários e atividades mudam com o passar do tempo. Os sistemas têm de aprender e se adaptar a isto.

Neste trabalho será estudado um sistema de automação residencial inteligente que cria regras em função do aprendizado com o comportamento dos habitantes de uma casa, dando ênfase à obtenção, tratamento e manutenção das regras.

O documento é composto pela seção 2, que contém resumidamente os sistemas de Aprendizado Automático; a seção 3 apresenta o conceito de Domótica; na seção 4 é descrito o sistema ABC (Automação Baseada em Comportamento); a seção 5 detalha o sistema ABC+ proposto; a seção 6 contém o trabalho proposto para a dissertação e a seção 7 conclui o artigo.

## 2 Sistemas de Aprendizado Automático

O aprendizado automático é o campo dedicado ao desenvolvimento de métodos computacionais para os processos de aprendizagem e a aplicação de sistemas informáticos de aprendizagem a problemas práticos [6]. O aprendizado automático se desenvolve através de programas computacionais capazes de automaticamente adquirir conhecimento de alto nível e/ou estratégias para resolver problemas a partir de exemplos. Com os exemplos gerados por um tutor e o conhecimento de base ou prévio um sistema de aprendizagem consegue criar descrições gerais de conceitos.

Para se evitar sistemas de aprendizado automático tipo caixa preta, no qual as representações dos conceitos não são compreensíveis aos humanos, alguns pesquisadores procuraram por sistemas capazes de induzir conceitos que sejam representados por uma linguagem facilmente compreensível.

Os sistemas de aprendizado simbólico são aqueles nos quais se constroem representações simbólicas de um conceito para aprender, através de análise de exemplos e contra-exemplos desse conceito [3]. As representações simbólicas geralmente estão na forma de expressão lógica, árvores de decisão, regras (ou rede semântica), sendo estas duas últimas as mais estudadas atualmente.

Uma maneira de representar conceitos é a lógica de regras de decisão associativas, a qual pode ser facilmente entendida por humanos, pois tem a seguinte forma: Se Y é verdade e X é falso, então classe A [3].

Existem várias maneiras de adquirir conhecimento a partir de dados; para determinar qual é a melhor maneira para um determinado conjunto de dados, é necessário definir como avaliar os métodos de aprendizagem.

Podem-se citar alguns indicadores de desempenho de sistemas de aprendizado [6]:

- Precisão: quantidade de exemplos positivos e negativos avaliados corretamente.
- Eficiência: um sistema deve ser capaz de gerar descrições corretas com um número mínimo de exemplos.
- Compreensibilidade: é importante que conceitos gerados sejam compreensíveis ao usuário, já que a finalidade destes sistemas é que o usuário aprenda algo deles.
- Robustez: contra o ruído e contra os exemplos incompletos.
- Requerimentos Especiais: alguns domínios requerem que um sistema aprenda à medida que chegam os exemplos, isto é conhecido como aprendizado incremental.

**Algoritmo C4.5.** O algoritmo C4.5 [9] é um sistema de aprendizado que constrói árvores de decisão a partir de um conjunto de exemplos. Estes exemplos são eventos

compostos por vários atributos e uma única classe. O domínio de cada atributo destes eventos está limitado a um conjunto de valores.

O algoritmo C4.5 gera uma árvore de decisão a partir dos dados mediante partições realizadas recursivamente. A árvore é construída mediante a estratégia de busca em profundidade (depth-first) [5]. O algoritmo considera todas as provas possíveis que podem dividir o conjunto de dados e seleciona a prova que resulta no maior ganho de informação. Para cada atributo discreto, se considera uma prova com  $n$  resultados, sendo  $n$  o número de valores possíveis que pode tomar o atributo. O C4.5 permite trabalhar com valores contínuos para os atributos, separando os possíveis resultados na árvore em duas partes: uma para aqueles  $A_i \leq N$  e outra para  $A_i > N$ , onde  $A_i$  é o  $i$ -ésimo atributo e  $N$  é algum valor dentro dos limites de valores do atributo  $A_i$ . A vantagem dos atributos contínuos é a eliminação da restrição de valores discretos. Para cada atributo contínuo, se realiza uma prova binária sobre cada um dos valores que toma o atributo nos dados.

Outro aspecto importante é a possibilidade de trabalhar com atributos desconhecidos. Neste caso os dados são aproveitados e consegue generalizar regras.

### 3 Domótica

A palavra Domótica é a junção da palavra latina Domus (casa) e do termo Robótica [2]. O significado está relacionado à instalação de tecnologia em residências, com o objetivo de melhorar a qualidade de vida, aumentar a segurança e viabilizar o uso racional dos recursos para seus habitantes.

Um sistema domótico é dividido em vários subsistemas, cada qual atua especificamente em um campo de controle. Atualmente estes sistemas são informatizados e computadorizados.

**Domótica Inteligente.** Pode-se citar como características fundamentais num sistema inteligente: ter memória; ter noção temporal; fácil interação com os habitantes; capacidade de integrar todos os sistemas do ambiente; atuar em várias condições; facilidade de reprogramação e capacidade de auto-correção [1].

A Domótica inteligente não é simplesmente prover a uma residência um sistema dotado de controle central que possa aperfeiçoar certas funções inerentes à operação e administração da mesma. Pode-se imaginar que uma residência inteligente é algo como uma residência com vida própria, portanto os sistemas de Domótica inteligente devem ter as características de um sistema inteligente e devem interagir com os habitantes da residência, aprendendo dinamicamente com seus comportamentos. Este aprendizado é permanente, pois os habitantes estão sempre mudando.

### 4 Sistema ABC

O processo normal de criação de regras é aquele onde o habitante é quem cria as regras, inserindo-as em um sistema. O sistema ABC (Automação Baseada em Comportamento) [10], o qual aprende regras em função do comportamento do

habitante, foi testado e demonstrou através de simulações que é possível reverter o processo normal de criação de regras.

A arquitetura do sistema ABC, que será resumidamente descrita, define a existência de sensores (detectores de presença, medidores de temperatura, medidores de luminosidade, etc.), atuadores (interruptores de luz, ar-condicionado, etc.), bancos de dados e demais elementos necessários para criação e controle das regras.

Para cada atuador da casa existe um banco de dados de aquisição de comportamento, o qual é alimentado com eventos e os respectivos dados dos sensores vinculados ao atuador. Como exemplo, pode-se ter o atuador “Ar-condicionado” e os sensores “Temperatura”, “Luminosidade”, “Horário” e “Presença” (o nome do sensor é utilizado como nome do atributo, portanto o sensor “Presença” que identifica se existe a presença ou não do habitante, terá seu valor refletido no atributo “Presença”, neste caso com “Sim” ou “Não”). Quando o estado do Ar-condicionado muda, por ação do habitante, os dados do próprio atuador e mais os dados dos sensores são armazenados em uma linha do banco de dados de aquisição de comportamento.

Quando o banco de dados atinge o valor configurado de eventos armazenados, o mesmo é inserido no algoritmo de aprendizado com árvores de indução ID3 [8], o qual generaliza os dados e cria regras. As regras são armazenadas no banco de dados de regras ativas. Uma regra aprendida poderia ser: SE Temperatura=Alta E Horário=Noite E Luminosidade=Alta E Presença=Sim ENTÃO Ar-condicionado=Ligado. Assim, quando os sensores indicam os valores presentes na regra o ar-condicionado é automaticamente ligado pelo Sistema ABC. Ou seja, a partir do momento em que novos eventos acontecem, de acordo com as ações do habitante, é feita uma varredura no banco de dados de regras ativas para avaliar se alguma regra deve ser aplicada e realizar uma ação no atuador.

Existe outro banco de dados onde estão as regras de segurança, nele podem existir regras do tipo: Fogo=Sim ENTÃO Energia=Desligada.

A manutenção das regras é simples, quando uma regra fica sem ser utilizada por um valor de tempo pré-configurado, ela é removida. Também é possível remover regras manualmente pelo habitante.

Apesar de muito interessante, o sistema possui deficiências. Ele não detecta seqüências causais de eventos no tempo, se um evento de atuador acontece depois de pouco ou muito tempo de um evento de sensor isto não é considerado. Neste sistema, com o uso do ID3, é possível trabalhar somente com variáveis lógicas, não é possível trabalhar com valores contínuos. Outra deficiência é o fato de regras criadas pelo ID3 se tornarem diretamente ativas, isto pode desagradar o habitante da casa.

## **5 Sistema ABC+ Proposto**

O sistema ABC, em sua versão inicial, marcou o uso de regras em automação residencial através da observação do comportamento de um habitante em uma casa, entretanto é possível observar que ele possui limitações que podem ser atenuadas.

Com o intuito de corrigir o sistema ABC e diminuir suas limitações, foi proposto o sistema ABC+, o qual possui arquitetura parecida (fig. 1) com o sistema inicial, entretanto as funcionalidades e lógicas de ambos são bastante diferentes.

As principais diferenças são a janela de observação de eventos, as regras embrionárias, C4.5 [9] e o novo processo de manutenção das regras.

Um quarto de uma casa onde existem um sensor de entrada no ambiente, um sensor de saída do ambiente e um atuador para ligar e desligar uma lâmpada servirá de exemplo para entender as implementações feitas no sistema ABC+.

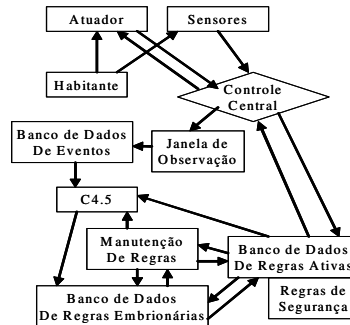


Fig. 1. Arquitetura do Sistema ABC+

## 5.1 A Janela de Observação

Nas condições em que o habitante entra no quarto e imediatamente acende a lâmpada ou quando sai do quarto e imediatamente desliga a lâmpada, pode-se facilmente assimilar regras, que seriam: SE habitante entra ENTÃO acenda lâmpada; SE habitante sai ENTÃO apague lâmpada.

Note que se o habitante entra no quarto e após uma hora ele acende a lâmpada, os sensores detectarão os mesmos dados da situação em que ele imediatamente acende a lâmpada após entrar no quarto. Isto leva a criação de uma regra errada, ou seja, a regra que será criada acende a lâmpada logo após o habitante entrar no quarto, mas o habitante não fez isto, sendo assim o sistema não deve criar tal regra.

Para eliminar esta limitação a janela de observação é utilizada. A janela consiste em se armazenar e comparar cada evento anterior e posterior ao evento em análise, inclusive com horários. Os eventos em análise são eventos de atuador, ou seja, eventos em que o atuador muda de estado, no exemplo a lâmpada. Eventos de sensores não são utilizados para gerar regras, mas sim para ativarem uma regra, se a mesma existir. Na janela os eventos anterior e posterior podem ser eventos de sensores, pois servirão de comparação e não para criação de regra.

No exemplo, usando uma janela de observação configurada com valor de 5 segundos, quando o habitante entra no ambiente ele dispara um evento de sensor, que é armazenado temporariamente com seu devido horário. Após uma hora ele acende a lâmpada. Este evento de atuador é armazenado e comparado com o evento anterior, se o anterior aconteceu dentro de 5 segundos, então significa que o evento do atuador está vinculado ao evento anterior, neste caso o evento do atuador é armazenado para gerar uma futura regra. Quando existe vinculo com o evento anterior não se realiza a

comparação com o posterior e o sistema volta a situação de início. Porém, se foi constatado que não existe vínculo entre evento do atuador e o evento anterior, é necessário verificar o evento posterior. Novamente se compara a diferença de horários entre eles e se o posterior aconteceu dentro de 5 segundos, então significa que o evento do atuador está vinculado ao evento posterior, caso contrário não existe vínculo. No caso de estarem vinculados armazenam-se os dados do evento de sensor posterior para gerar futura regra, pois é este evento que reflete as reais condições para o atuador ter sido mudado.

## 5.2 As Regras Embrionárias

No sistema ABC, sempre que o ID3 cria uma regra, ela é imediatamente colocada como regra ativa. Sendo assim, não existe uma etapa de validação da regra. A primeira vez que o habitante gerar as condições dos sensores iguais às condições da regra, a mesma será executada e alguma ação será feita. Se o habitante contrariar a regra nada ocorre, e se ele quiser acabar com a regra terá de removê-la manualmente.

Ao criar uma regra, deve existir uma etapa de validação, ou seja, ter certeza que a regra é aceita pelo habitante. No sistema ABC+, toda vez que novas regras são criadas, elas são armazenadas inicialmente no banco de regras embrionárias.

As regras embrionárias são regras que estão em validação. Quando o habitante gerar condições dos sensores iguais às condições de uma regra embrionária, o que acontece é que a regra ganha pontos positivos. Somente quando a regra atinge uma certa pontuação superior é que ela passa para o banco de regras ativas.

Por outro lado, se o habitante contrariar a regra, ela ganha pontos negativos e ao atingir uma pontuação inferior a mesma é eliminada do banco de regras embrionárias.

## 5.3 A Manutenção das Regras

A maneira como se mantêm as regras em uma casa influencia a interação do sistema com o habitante.

A inserção e remoção de regras têm de ser o mais sutil possível, caso contrário trará desconforto ao usuário e podem incorrer em desestabilização das próprias regras.

Para o desenvolvimento e manutenção de regras no sistema ABC+ são utilizados três bancos de dados para cada atuador da casa.

**Tabela 1.** Eventos por Atuador (A1, A2, .....An).

Evento	Sensor 1 (S1)	Sensor 2 (S2)	.....	Sensor N (Sn)	Atuador An
1					
2					
3					
....					

Serão utilizadas as seguintes definições:

BDEventos – Banco de Dados de Eventos por Atuador.

BDAtivas – Banco de Dados de Regras Ativas.

ATIV – Campo do BDAtivas utilizado para pontuar positivamente uma regra.

EXC - Campo do BDAtivas utilizado para pontuar negativamente uma regra.

BDEmbrio – Banco de Dados de Regras Embrionárias.

OK - Campo do BDEmbrio utilizado para validar uma regra.

NOK - Campo do BDEmbrio utilizado para excluir uma regra.

Os novos eventos que surgem nos atuadores da casa, e passam pelo crivo da janela de observação, são armazenados no Banco de Dados de Eventos por Atuador (tab. 1).

Na tabela 1 cada linha representa um evento de atuador armazenado. A informação que consta na linha é o valor do atributo atuador e os valores dos atributos dos sensores para um determinado evento. Um exemplo de evento e atributos pode ser: atributo atuador igual a “Lâmpada” e seus valores “Acesa” e “Apagada”, atributo sensor 1 igual a “Temperatura” e seus valores “Alta”, “Normal” e “Baixa”, atributo sensor 2 igual a “Luminosidade” e seus valores “Alta”, “Normal” e “Baixa”, atributo sensor 3 igual a “Umidade” e seus valores “Alta”, “Média” e “Baixa” e atributo sensor 4 igual a “Porta” e seus valores “Entrada” e “Saída”. Um evento pode ser:

**Tabela 2.** Exemplo de evento armazenado no BDEventos.

Evento	Temperatura	Luminosidade	Umidade	Porta	Lâmpada
1	Alta	Alta	Média	Entrada	Acesa

Quando o banco de dados de eventos de atuador atinge um número pré-determinado de eventos ocorre a sua inserção no algoritmo C4.5 para a geração de novas regras. Estas novas regras são inicialmente comparadas com as regras ativas e embrionárias, para que as regras repetidas sejam eliminadas.

As novas regras geradas, que não forem repetidas, vão para o banco de regras embrionárias. O Banco de Regras Embrionárias tem o mesmo formato da tabela 1, porém existem para cada regra, além dos campos de sensores e atuador, os campos OK e NOK, os quais servem para o controle da validação da regra. Quando uma regra embrionária é gerada ela possui os campos OK e NOK iguais a zero.

Sempre que ocorre um novo evento de atuador este é comparado primeiro com as regras ativas, para verificar se alguma regra ativa está sendo contrariada ou não, depois este evento é comparado com as regras embrionárias. Caso exista regra embrionária relacionada ao evento é necessário dar pontos à mesma. Se o evento confirmou a regra, ao valor do campo OK é somado mais um, caso o evento contrarie a regra o valor do campo NOK recebe mais um. Quando o campo OK atinge certo valor, a regra embrionária é transformada em regra ativa. Quando o campo NOK atinge certo valor, a regra embrionária é removida do banco de regras.

Os campos OK e NOK trabalham como porcentagens de acertos e erros. Por exemplo, se o valor no campo OK para a regra ser validada é oito e o valor no campo NOK para a regra ser excluída é dois, significa que quando temos 80% ou mais de acertos a regra é validada ou quando temos 20% ou mais de erros a regra é excluída.

O Banco de Regras Ativas também possui uma lógica de manutenção. Ele também tem o mesmo formato da tabela 1, porém para cada regra possui além dos campos de sensores e atuador, os campos ATIV e EXC. É neste banco que estão as regras de segurança, as quais não sofrem modificações, e as regras a serem executadas na casa. Sempre que algum evento de sensores acontece o banco de regras ativas é consultado, se existir alguma regra relacionada ao evento a mesma é aplicada, assim uma ação é feita no atuador. Neste caso soma-se ao valor do campo ATIV o valor um.

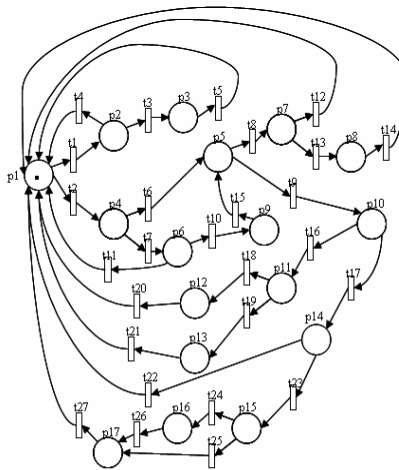
O banco de regras ativas possui uma quantidade limitada de regras. As regras estão ordenadas de acordo com o valor do campo ATIV. Quando uma nova regra é inserida no BDAtivas e este está com sua capacidade máxima, então é necessário retirar a regra com menor pontuação ATIV e colocar esta regra de volta ao BDEmbrio.

Um evento de atuador também recebe atenção por parte do BDAtivas. Caso o evento atuador seja contrário a alguma regra existente no BDAtivas, soma-se ao valor do campo EXC desta regra em questão o valor um. Se o campo EXC ultrapassar um valor pré-determinado, esta regra é excluída do BDAtivas.

Um ponto importante na manutenção das regras é a criação de novas regras. Quando o C4.5 recebe o BDEventos para criar novas regras, uma função mais nobre é executada por ele. São inseridas no C4.5, além das informações do BDEventos, as regras constantes no BDAtivas. Com isto pode-se ter a generalização de algumas regras existentes. Após a criação destas novas regras verifica-se se as mesmas já constam em BDAtivas e BDEmbrio, para evitar duplicação de regras. Tal processo torna necessário que as regras BDEmbrio ao serem promovidas a BDAtivas sejam avaliadas com o objetivo de verificar se são uma generalização de alguma regra existente, caso sejam as correspondentes regras do BDAtivas são eliminadas.

**POSIÇÕES**

- P1 – Estado inicial de espera
- P2 – Busca por RA existente
- P3 – Realização da ação X, fazer ATIV=ATIV+1, com ATIV máximo=T e ordenação do BRA por ATIV
- P4 – Busca por ES anterior recente (R segundos)
- P5 – Busca por RA existente
- P6 – Busca por ES posterior recente (R segundos)
- P7 – Avaliação se a regra é igual ou contrária
- P8 – Fazendo EXC=EXC+1, se EXC>K a regra é excluída do BRA
- P9 – Armazenamento dos dados do ES posterior
- P10 – Busca por RE existente
- P11 – Avaliação se a regra é igual ou contrária
- P12 – Fazendo OK+OK+1, se OK=Z regra vira RA
- P13 – Fazendo NOK=NOK+1, se NOK>Q regra é excluída do BRE
- P14 – Colocação do evento no BE
- P15 – C4.5 gera novas regras a partir de BE+BRA
- P16 – As regras repetidas são ignoradas
- P17 – Regras antigas do BRE com OK<W são excluídas; coloca regras novas no BRE, para toda RA fazer ATIV=ATIV-1; se RA tem ATIV<J coloca-a no BRE



**TRANSIÇÕES**

- T1 – Novo evento de sensor
- T2 – Novo evento de atuador
- T3 – Existência de RA
- T4 – Não existência de RA
- T5 – Finalização da ação em P3
- T6 – Existência de ES anterior
- T7 – Não existência de ES anterior
- T8 – Existência de RA
- T9 – Não existência de RA
- T10 - Existência de ES posterior
- T11 - Não existência de ES posterior
- T12 – A regra é igual
- T13 – A regra é contrária
- T14 – Finalização da ação em P8
- T15 – Finalização da ação em P9
- T16 - Existência de RE
- T17 – Não existência de RE
- T18 - A regra é igual
- T19 – A regra é contrária
- T20 - Finalização da ação em P12
- T21 - Finalização da ação em P13
- T22 – BE não tem Y eventos
- T23 – BE tem Y eventos
- T24 – Alguma(s) nova(s) regra(s) está(ão) em BRA ou BRE
- T25 – Nenhuma nova regra está em BRA ou BRE
- T26 - Finalização da ação em P16
- T27 – Finalização da ação em P17

**LEGENDA**

- EA=Evento Atuador
- ES=Evento Sensor
- BE=Banco de Eventos
- RA=Regra Ativa
- BRA=Banco de Regras Ativas
- RE=Regra Embrionária
- BRE=Banco de Regras Embrionárias

**Fig. 2.** Máquina de estado em Rede de Petri do Sistema ABC+.

Na criação de novas regras também é checado se alguma regra do BDAtivas ficou velha. Para isto subtrai-se o valor um do campo ATIV de todas as regras. As regras que tiverem seu campo ATIV abaixo de certo valor serão rebaixadas ao BDEmbrio.

Para que o sistema possa ser estendido para aplicações reais e mais complexas será ainda necessário identificar como lidar com os possíveis loops nas regras criadas.



Um *loop* pode fazer com que uma regra ao ser ativada leve à ativação direta de uma outra regra ou ativação indireta de regra por mudança de algum sensor.

A Máquina de estado em Rede de Petri do sistema ABC+ (fig. 2) reflete as informações descritas nesta seção.

## 6 Trabalho em Desenvolvimento

Um simulador foi desenvolvido para confirmar o funcionamento do sistema proposto. O ambiente simulado possui apenas um cômodo (quarto) com um atuador (lâmpada) e alguns sensores.

Para a implementação do simulador foi utilizado o software Delphi. O algoritmo C4.5 utilizado é o original [9], porém codificado para o Windows. Os sensores podem ter seu estado alterado através de botões, conforme a figura 3. As simulações foram feitas manipulando estes botões e assim gerando eventos.

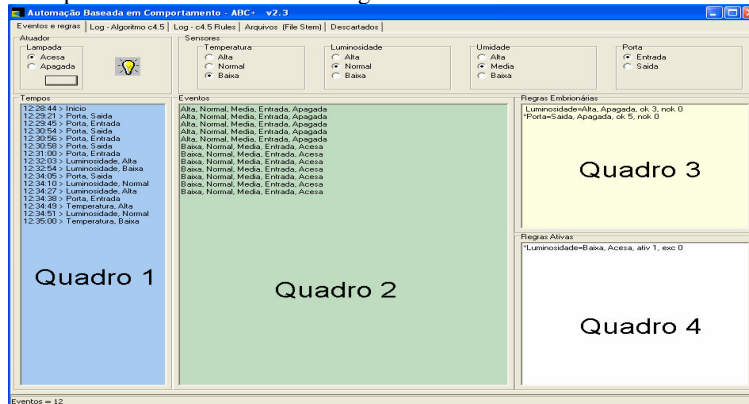


Fig. 3. Simulador do sistema ABC+.

Um exemplo é clicar no valor Normal, do sensor Temperatura. Isto gera um evento de sensor que fica armazenado temporariamente, inclusive com horário, e pode ser visualizado no quadro um do simulador. Este evento pode: levar à execução de uma das regras ativas, as quais podem ser observadas no quadro quatro do simulador; atuar na validação ou rejeição de uma das regras embrionárias, as quais podem ser vistas no quadro três do simulador; servir de referência na janela de observação.

Outro exemplo é clicar no valor Apagada do atuador Lâmpada. Isto gera um evento de atuador, o qual passa pela lógica da janela de observação e se for validado é armazenado no banco de dados de eventos do atuador. No quadro dois do simulador é possível ver os eventos de atuador armazenados.

**Motivação.** Com a implementação do simulador surgiram vários pontos em aberto que necessitam ser desenvolvidos para que se encontre o melhor estado de funcionamento do sistema de automação residencial baseado no comportamento.

O primeiro ponto a ser desenvolvido é a identificação dos parâmetros de erro e desempenho que melhor permitem avaliar o sistema.

Outros pontos são: Determinar quando uma regra deve ser criada, se em todo fim de dia ou após n eventos (mudanças de estado). Definir qual o número de regras que se deve ter. Identificar se após a criação da regra, nas n primeiras atuações da mesma o evento imediatamente posterior à mesma deve ser ignorado ou não (adaptação do usuário a regra). Em resumo a estes pontos, é necessário identificar os valores ótimos das variáveis mais o valor máximo de regras em BDAtivas presentes no sistema.

Para as simulações será criado um banco de dados de eventos que reflitam as atividades de um usuário durante um período de tempo; através de um agente estes eventos serão imputados no sistema, observando-se o comportamento das regras que são produzidas. As simulações serão feitas com variações dos valores das variáveis presentes no sistema e os parâmetros de erro e desempenhos serão avaliados.

## 7 Conclusão

O trabalho a ser desenvolvido terá como objetivo apresentar um novo sistema de automação residencial, sistema ABC+, o qual aprende regras por observação do comportamento do habitante de um ambiente.

Como existem habitantes que se adaptam facilmente a inovações e outros não, os mesmo irão interagir diferentemente com o sistema ABC+. Portanto a identificação de parâmetros para mensurar o desempenho do sistema e a análise do comportamento das variáveis servirá para entender como o sistema pode ser moldado de forma a atender diversos perfis de habitante, com hábitos e gostos diferentes.

## Referências

1. Introdução a Domótica. Disponível em <http://www.din.uem.br/ia/intelige/domotica/int.htm>. Acessado em 06/08/2005.
2. Angel, P. M. Introducción a la domótica; Domótica: controle e automação. Escuela Brasileño-Argentina de Informática. EBAI. (1993)
3. Batista, G.E. Um ambiente de Avaliação de Algoritmos de Aprendizado de Máquina utilizando exemplos. Dissertação (mestrado), Universidade de São Paulo, São Carlos. (1997)
4. Bolzani, C.A.M. Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes : uma introdução aos sistemas domóticos. Dissertação (mestrado), Universidade de São Paulo, São Paulo. (2004)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.. Introduction to Algorithms. Second Edition. MIT Press and McGraw-Hill. Section 22.3: Depth-first search, 540–549. (2001)
6. Michalski, R.S., Bratko, I., Kubat M. Machine Learning and Data Mining. Methods and Applications. Wiley & Sons Ltd., EE.UU. (1998)
7. Muratori, J.R. As tendências do mercado de Automação Residencial. Congresso Habitar - Congresso de Automação Residencial e Tecnologias para Habitação, São Paulo. (2005)
8. Quinlan, J.R. Induction of Decision Trees. Machine Learning, Capítulo 1, Morgan Kaufmann. (1990). 81-106.
9. Quinlan, J.R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers. San Mateo, California, EE.UU. (1993)
10. Tonidandel, F., Takiuchi, M., Melo, E. Domótica Inteligente: Automação baseada em comportamento. Congresso Brasileiro de Automática. (2004)