

MODELO DE ESPECIFICAÇÃO E PROGRAMAÇÃO DE UM SISTEMA DOMÓTICO

Renato Jorge Caleira Nunes
Instituto Superior Técnico / INESC-ID
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
renato.nunes@dei.ist.utl.pt

RESUMO

Neste artigo aborda-se a especificação de um sistema domótico e a programação do seu comportamento. É proposta uma abordagem baseada numa aplicação WWW que oferece uma solução genérica e independente da tecnologia domótica utilizada. Para permitir o grau de generalidade desejado foi concebido um modelo de abstracção de um dispositivo domótico, que é detalhado. É também apresentado um modelo simples que permite descrever a estrutura física de uma habitação e identificar os dispositivos existentes e a sua localização. Com a metodologia proposta torna-se possível descrever qualquer sistema e qualquer habitação.

É ainda apresentada uma solução para o problema da definição do comportamento de um sistema domótico. Essa solução baseia-se no uso de cláusulas *IF...THEN* que são simples de compreender e aplicar por um utilizador comum.

Por último descreve-se sucintamente a implementação de uma aplicação WWW, designada *DomoConfig*, que permite a definição da composição e características de um sistema domótico e a programação do seu comportamento.

PALAVRAS-CHAVE

Domótica, Automação Residencial, Modelo de abstracção, Aplicação WWW, Interoperação.

1. INTRODUÇÃO

A domótica, ou automação residencial, tem tido uma divulgação crescente dado os benefícios que oferece. Ela pode facilitar tarefas diárias tais como controlar a rega do jardim, ligar a iluminação exterior quando escurece, estabelecer níveis de temperatura e iluminação adequados quando se chega a casa após o trabalho, descer ou subir os estores a determinadas horas do dia ou desligar lâmpadas em divisões que ficaram vazias. Deste modo a domótica pode contribuir para oferecer melhores níveis de conforto e, ao mesmo tempo, otimizar os gastos energéticos e conduzir a poupanças nos consumos de electricidade, gás e água.

A domótica permite também aumentar a nossa segurança, detectando e sinalizando situações de emergência (e.g., incêndio, fugas de gás, inundação) ou situações de intrusão. Pode ainda ser de grande utilidade para pessoas idosas ou pessoas com mobilidade reduzida ou outro tipo de limitação física, permitindo-lhes uma maior autonomia e qualidade de vida. Por exemplo, o uso de um telecomando adequado ou de um dispositivo que reconhece mensagens de voz pode facilitar imensas tarefas como sejam o abrir a porta da rua, o ligar a iluminação ou o desligar o forno, evitando que a pessoa tenha de se deslocar.

Actualmente existem disponíveis múltiplas tecnologias domóticas. Várias estão normalizadas ou são *abertas* (por exemplo, X10 [1], EIB [2], LonWorks [3] e CEBus [4]), e muitas outras são propriedade de empresas. Estas tecnologias são incompatíveis entre si e não oferecem mecanismos próprios de suporte à interoperação. Outras tecnologias, tais como UpnP [5], OSGi [6] e Jini [7], abordam o problema da interoperação mas impõem restrições (por exemplo, o uso obrigatório de TCP/IP ou o uso exclusivo da linguagem Java e respectivas máquinas virtuais) e não são particularmente adequadas para a domótica [8].

Na nossa opinião, a existência de múltiplas tecnologias incompatíveis constitui um factor de grande importância que tem dificultado uma maior divulgação da domótica.

Outra limitação das tecnologias actuais refere-se ao facto de não oferecerem formas normalizadas de abordagem ao problema da especificação de um dado sistema e sua adaptação a uma habitação em concreto e

aos requisitos específicos dos utilizadores. Nalguns casos, como sucede com o EIB, existem disponíveis aplicações (e.g., ETS2) mas estas são específicas para uma dada tecnologia e destinam-se a ser usadas por pessoal técnico especializado e não por utilizadores comuns.

E, ainda mais limitador, é a inexistência de aplicações genéricas de monitorização e controlo que sejam capazes de lidar com alterações e permitam ao utilizador modificar o comportamento do sistema e adaptá-lo a novas necessidades ou a novas preferências. As soluções mais comuns consistem em aplicações específicas, desenvolvidas para uma dada tecnologia, uma dada habitação e com um comportamento predefinido. Estas soluções tendem, a médio ou longo prazo, a tornar-se insatisfatórias, conduzindo a um sentimento de frustração equivalente a descobrir que um interruptor foi instalado num local incorrecto e nada é possível fazer quanto a isso, a não ser recorrendo a obras de vulto (no caso em apreciação, nova versão da aplicação).

É interessante notar que a tecnologia X10 constitui uma excepção neste panorama. A sua simplicidade permite a um utilizador comum adquirir módulos que podem ser instalados facilmente na sua habitação, bastando ligá-los nas vulgares tomadas de energia eléctrica. Isto conduziu à divulgação desta tecnologia e à necessidade de dispor de soluções genéricas para definir quais as acções desejadas em certas circunstâncias e permitir efectuar programações horárias. A necessidade de flexibilidade era óbvia uma vez que cada utilizador pode ter um sistema com uma configuração diferente e que essa configuração pode mudar ao longo do tempo. Existem várias aplicações que oferecem bons níveis de funcionalidade, nomeadamente, *ActiveHome*, *HomeSeer*, *Home Director* e *HomeControl*.

No entanto, a tecnologia X10 constitui um caso particular dada a sua simplicidade, reduzido número de tipos de módulos e funcionalidade básica (ligar / desligar, aumentar / reduzir intensidade luminosa). Os sistemas X10 são também, normalmente, de reduzida dimensão. Por estas razões, as soluções disponíveis para a tecnologia X10 não podem ser facilmente expandidas ou adaptadas a outras tecnologias com mais capacidades e a sistemas de grande dimensão.

Neste artigo abordamos o problema descrito e propomos uma solução baseada numa aplicação WWW que oferece um elevado grau de generalidade, flexibilidade e independência face à tecnologia domótica utilizada.

Começamos por apresentar um modelo de abstracção de um dispositivo domótico, o que constitui um elemento chave da nossa proposta. Em seguida é descrita uma forma simples de representar a estrutura física de uma habitação, com o objectivo de permitir localizar onde estão instalados os diversos dispositivos. Após dispor de um modo para descrever de forma genérica a constituição de um sistema domótico abordamos o problema da definição do comportamento desejado para o sistema. Apresentamos uma proposta que usa a noção de *cenário*, o que consiste num conjunto de acções que são executadas sempre que determinada condição é satisfeita. Essa condição pode ser tão complexa quanto desejado. Este modelo baseia-se no uso de cláusulas *IF...THEN*, o que é fácil de compreender e usar por um utilizador comum.

A finalizar, descrevemos de forma muito sucinta as opções de base tomadas na implementação da aplicação *DomoConfig* que dá corpo à aplicação prática dos modelos propostos.

2. MODELO DE ABSTRACÇÃO DE DISPOSITIVOS DOMÓTICOS

Um dos nossos principais objectivos é abordar a domótica (automação residencial) de um modo genérico e flexível, permitindo representar qualquer dispositivo. Essa representação deverá ser tão simples quanto possível mas suficientemente poderosa para descrever as capacidades essenciais de um dispositivo e ser independente da sua tecnologia.

Com esse objectivo, propomos uma solução em que cada dispositivo é caracterizado por um conjunto de propriedades, tendo cada propriedade um valor. Por exemplo, um sensor de temperatura pode ter apenas uma propriedade - *Temperatura* - que contém o valor actual da temperatura medida pelo dispositivo. No caso de um dispositivo que controla uma lâmpada, ele pode possuir duas propriedades: *EstadoLigadoDesligado* e *IntensidadeLuminosa*. A primeira propriedade indica se a lâmpada está ligada ou desligada e a segunda propriedade indica a sua intensidade (numa escala de 0 a 100, por exemplo). O uso de duas propriedades permite desligar a lâmpada sem alterar a intensidade previamente definida. Deste modo, quando se voltar a ligar a lâmpada, esta pode assumir o valor de intensidade anterior.

Os dispositivos mencionados são simples mas é fácil descrever dispositivos mais complexos como, por exemplo, um televisor. Neste caso, e considerando apenas as funções mais óbvias e úteis, faria sentido

considerar quatro propriedades: *EstadoLigadoDesligado*, *Canal*, *Volume* e *ModoSilencioso (mute)*. No entanto, seria muito simples adicionar mais propriedades para representar, por exemplo, o brilho, a cor, agudos e graves.

Os exemplos anteriores ilustram a nossa abordagem, a qual será detalhada em seguida usando um modelo expresso em UML [9] e que se encontra presente na figura 1.

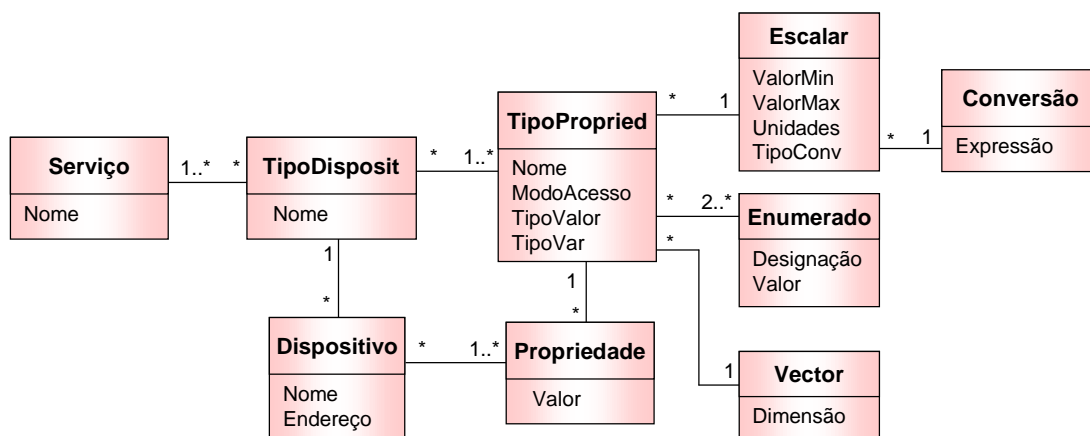


Figura 1. Modelo de um dispositivo domótico.

A entidade *TipoDispositivo* abstrai dispositivos de um dado tipo, tais como lâmpadas de intensidade ajustável, sensores de temperatura, ar-condicionados, interruptores comuns, tomadas de energia, motores de estores, electroválvulas, detectores de fumo, sensores de presença, aparelhagem de alta fidelidade, TV, DVD, etc. Convém notar que é possível, em qualquer momento, adicionar novos tipos de dispositivos, desde que seja possível descrever as suas características usando propriedades tal como descrito anteriormente.

Antes de detalhar as características de *TipoDispositivo* começamos pela classe *Serviço*. Esta classe permite agrupar tipos de dispositivos que partilham uma relação funcional. Para exemplificar, referimos que é possível ter serviços de iluminação, climatização, segurança, entretenimento, etc. Como se observa no modelo, é possível um dispositivo estar associado a mais do que um serviço. Esta possibilidade é útil, por exemplo, no caso de um sensor de presença, o qual pode pertencer quer ao serviço de segurança (para detectar intrusos) quer ao serviço de iluminação (para controlar a iluminação de uma divisão em função da presença de pessoas). Um dado serviço pode possuir qualquer número de dispositivos associados a ele. A classe *Serviço* é útil sempre que alguém deseja aceder aos dispositivos existentes em função da sua funcionalidade. Por exemplo, a selecção do serviço de climatização permite aceder a todos os equipamentos com ele relacionados, tais como ar-condicionados, aquecedores e sensores de temperatura.

A caracterização de um *TipoDispositivo* é efectuada usando uma ou mais *TipoPropriedade* (ver figura 1). O mesmo *TipoPropriedade* pode ser usado por qualquer número de *TipoDispositivo*. Esta particularidade é útil pois permite partilhar propriedades que são comuns a múltiplos dispositivos como, por exemplo, *EstadoLigadoDesligado*.

Na figura 1 estão representados os principais atributos da classe *TipoPropriedade*. O atributo *Nome* contém a designação do tipo de propriedade, podendo tomar valores tais como *temperatura*, *intensidade-luminosa* e *canal*. Para cada tipo de propriedade é possível definir um modo de acesso que pode assumir valores tais como *só-leitura*, *só-escrita* ou *leitura-escrita*. Para exemplificar o uso deste atributo, refere-se que a propriedade *temperatura* de um sensor faz todo o sentido que seja *só-leitura*, enquanto a propriedade *intensidade-luminosa* de uma lâmpada é de *leitura-escrita*. Neste último caso, é possível ler o valor da propriedade para conhecer a intensidade actual da lâmpada em questão e é possível escrever nessa propriedade para alterar a sua intensidade. Embora ainda não tivesse sido claramente expresso, neste momento deverá ser aparente que o modelo proposto assume que todas as interacções com os dispositivos físicos são realizadas exclusivamente lendo ou escrevendo valores em propriedades.

Outro atributo de *TipoPropriedade* é *TipoValor*, que permite distinguir três tipos de valores: *Escalar*, *Enumerado* e *Vector*. O tipo *Escalar* identifica um valor que pode variar entre um valor mínimo (*ValorMin*) e

um valor máximo (*ValorMax*). Esse valor pode ter um certo tipo de unidades (por exemplo, graus Celsius, Watt, Lux), pode corresponder a uma percentagem (escala entre 0 e 100) ou simplesmente não possuir unidades. Isso é especificado no atributo *Unidades*. Um valor escalar pode ainda estar sujeito a uma conversão (indicada por *TipoConv*), caso em que existirá uma associação com uma classe *Conversão* a qual contém a expressão simbólica a usar. Esta possibilidade é útil em múltiplas situações. Por exemplo, considere-se que a temperatura é medida com uma precisão de um décimo de grau mas é representada por um inteiro. Neste caso, uma gama de temperaturas entre $-25,0^{\circ}\text{C}$ e $65,0^{\circ}\text{C}$ seria representada por um valor entre -250 e 650, pelo que a expressão de conversão seria simplesmente de $0,1x$. Seria também possível converter graus Fahrenheit em Celsius, usando a expressão $(x-32)*5/9$, ou efectuar qualquer outro tipo de conversão.

Nalguns casos uma propriedade pode ter apenas um pequeno número de valores. Por exemplo, a propriedade *EstadoLigadoDesligado* pode assumir apenas os valores 0 (desligado) e 1 (ligado). Relativamente ao estado de funcionamento de um ar-condicionado, podem-se considerar quatro possibilidades: desligado (0), ventilar (1), arrefecer (2) e aquecer (3). Para estes casos, e outros semelhantes, é muito mais amigável usar uma designação para representar o valor da propriedade do que o respectivo valor numérico. Com esse objectivo, foi criado o tipo *Enumerado*. Quando uma propriedade é deste tipo, ela estará associada a, pelo menos, dois objectos da classe *Enumerado*. O atributo *Designação* dessa classe contém o nome associado a um dado valor.

Uma propriedade pode também ser do tipo *Vector*. Este tipo é usado em propriedades que contêm uma lista de caracteres (representando um texto ou outro tipo de informação).

Até ao momento foi ilustrado como definir os tipos base de dispositivos, cada um caracterizado por um conjunto de propriedades. O passo seguinte consiste na definição dos dispositivos concretos existentes numa habitação. É esse o objectivo da classe *Dispositivo*. Por exemplo, um dispositivo designado *LâmpadaQuarto* pode ser criado, sendo explicitado que é do tipo *LâmpadaRegulada* definido anteriormente. Deste modo, torna-se imediatamente conhecido quais as propriedades que possui. Dado que este dispositivo corresponde a uma entidade física, necessita guardar valores específicos das suas propriedades. É essa a função da classe *Propriedade*. O dispositivo específico será associado com tantas propriedades quantas as que foram definidas aquando da criação do *TipoDispositivo*.

O modelo apresentado é pois muito flexível e eficaz. Após a definição dos tipos de dispositivos existentes, torna-se simples adicionar os dispositivos concretos que se pretendem instalar numa dada habitação. A definição dos tipos de dispositivos pode ser reutilizada em múltiplas situações, só necessitando ser modificada caso se torne necessário representar um novo dispositivo.

O modelo descrito permite também representar o tempo, que é uma entidade fundamental na definição de programações horárias. Consideramos que existe um dispositivo relógio, caracterizado por possuir as seguintes propriedades: minutos, horas, dia da semana, dia, mês e ano. É claro que podem ser usadas outras propriedades para representar, por exemplo, se é dia ou noite, se o presente dia é um feriado ou um dia normal.

A capacidade de representar o dispositivo relógio exactamente da mesma forma que qualquer outro dispositivo é muito importante, como se verá adiante, para permitir oferecer um modo genérico e totalmente uniforme de programação do comportamento de um sistema.

3. REPRESENTAÇÃO DE UMA HABITAÇÃO

Na secção anterior foi apresentado um modelo que permite a especificação de qualquer tipo de dispositivo domótico. Essa especificação é feita em duas fases. Primeiro são definidos tipos genéricos de dispositivos e são especificadas as suas propriedades. Em seguida são indicados os dispositivos concretos que existem numa dada habitação, sendo explicitado de que tipo são.

É também importante indicar onde os dispositivos estão instalados fisicamente e ter um modo de descrever a estrutura de uma habitação. Para atingir este objectivo propõe-se um modelo muito simples, que está representado na figura 2.

Nesse modelo, considera-se uma habitação como uma entidade que possui um ou mais pisos. Isto é particularmente útil no caso de vivendas em que podem existir vários pisos e até uma cave ou um sótão.

Cada piso contém um ou mais divisões e cada divisão pode ter qualquer número de dispositivos domóticos. A classe *Dispositivo* corresponde à classe com o mesmo nome presente na figura 1.

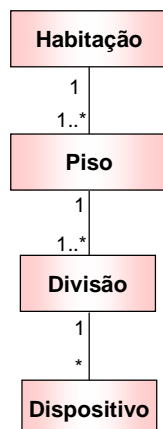


Figura 2. Modelo de uma habitação e respectivos dispositivos domésticos.

O modelo indicado permite, de uma forma simples mas eficaz, representar qualquer habitação. Convém notar que a entidade *Divisão* não necessita corresponder exactamente ao espaço físico de uma divisão, podendo referir-se a um espaço lógico que tenha interesse identificar. Por exemplo, no caso de uma sala que tenha uma zona de jantar e uma zona de lazer, pode ter interesse representar essas duas regiões. No modelo foi usada a designação *Divisão* por se considerar que é mais sugestiva.

4. MONITORIZAÇÃO E COMANDO DE UM SISTEMA DOMÓTICO

Usando os modelos descritos anteriormente torna-se possível começar do zero e definir qualquer tipo de dispositivo e associá-lo a um dado serviço. Podemos também representar qualquer habitação e especificar que dispositivos existem e onde estão localizados.

Isto oferece uma base de suporte à monitorização e controlo de qualquer habitação. Em termos de interacção com o utilizador, podem ser oferecidas duas formas distintas de navegação: uma através da estrutura da habitação e outra que tem em conta a organização funcional dos dispositivos em serviços. No primeiro caso, torna-se possível mostrar os vários pisos que constituem a habitação e permitir que o utilizador escolha um deles. Em seguida são mostradas as divisões (ou zonas) que estão definidas e, após selecção de uma delas, são visualizados os dispositivos existentes. No segundo caso, o utilizador visualiza os serviços existentes (por exemplo, iluminação, climatização, segurança, entretenimento, etc) e selecciona um deles. Isso dá acesso a todos os dispositivos associados a esse serviço, o que é particularmente útil quando se pretende monitorizar ou executar acções sobre vários dispositivos similares (por exemplo, desligar todos os ar-condicionados).

Os dois modos básicos de navegação podem ser misturados de modo a aumentar a usabilidade, permitindo ao utilizador, em cada momento, escolher qual o modo que deseja usar. Por exemplo, após aceder a uma dada divisão, o utilizador pode escolher visualizar apenas os dispositivos de um serviço específico, o que é particularmente útil se essa divisão possuir muitos dispositivos de tipos diferentes.

Após selecção de um dispositivo o utilizador acede às suas propriedades, as quais podem ser consultadas e alteradas. A forma como o utilizador visualiza o valor de uma propriedade depende do seu tipo. Se é uma propriedade escalar, o utilizador vê o seu valor. Para a alterar, o utilizador pode indicar directamente um novo valor, pode usar dois botões (um para aumentar e outro para diminuir) ou pode usar um controlo de rodar ou deslizar. São verificados os valores mínimos e máximos aceitáveis.

Se o valor é do tipo enumerado, o utilizador visualiza a designação que corresponde ao valor actual (por exemplo, desligado, arrefecer, a gravar). Para alterar um valor enumerado o utilizador acede a uma lista e escolhe um dos valores disponíveis.

5. DEFINIÇÃO DO COMPORTAMENTO DE UM SISTEMA

Antes de descrever alguns pormenores de implementação da nossa aplicação, abordamos em seguida um modelo de suporte à definição do comportamento desejado para uma habitação.

Propomos um modelo baseado na noção de *cenário*. Um cenário tem um nome que pode ser, por exemplo, "Ver TV", "Ir deitar", "Levantar de manhã", "Regar jardim", "Activar segurança", etc.

Um cenário corresponde essencialmente a uma cláusula do seguinte tipo:

IF *condição* THEN *lista-acções*

O termo *condição* pode ser uma expressão tão complexa quanto se queira, a qual pode envolver o tempo ou o valor de qualquer propriedade de um dispositivo. O tempo é tratado do mesmo modo que qualquer outro valor, pois corresponde a propriedades do dispositivo relógio (tal como foi explicado anteriormente).

A *lista-acções* é uma sequência de acções, em que cada uma consiste na atribuição de um valor a uma propriedade de um dispositivo.

Esta forma de especificar o comportamento de uma habitação é simples mas eficaz. É fácil de compreender por uma pessoa comum e simplifica o processo de manter o comportamento desejado uma vez que, tipicamente, os cenários são entidades independentes e não interagem ou dependem uns dos outros.

Para além do que foi definido, os cenários podem também possuir uma *lista de acções de desactivação*. Esta lista contém acções que são executadas quando uma condição de activação de um cenário deixa de se verificar. Para ilustrar os benefícios desta capacidade analisemos um cenário relativo à rega de um jardim e consideremos a seguinte condição (nota: é usada uma notação em que se identifica um dispositivo e a respectiva propriedade - *Dispositivo.propriedade*).

Relógio.hora >= 9:30 AND *Relógio.hora* <= 10:00 AND *SensorHumidadeSolo.humidade* < 30%

Quando esta condição é avaliada como verdadeira, uma lista de acções como a que se indica em seguida seria executada:

ElectroválvulaRelva.estado_aberta_fechada = aberta
ElectroválvulaCanteiro.estado_aberta_fechada = aberta

Como pode ser observado, se nada mais for efectuado, uma vez abertas as electroválvulas estas manter-se-ão nesse estado indefinidamente. A lista de acções de desactivação permite ultrapassar este problema, possibilitando, por exemplo, especificar as seguintes acções:

ElectroválvulaRelva.estado_aberta_fechada = fechada
ElectroválvulaCanteiro.estado_aberta_fechada = fechada

Como nota, referimos que deverão existir mecanismos de segurança que ofereçam salvaguardas para situações em que uma falha possa conduzir a problemas ou a comportamentos indesejados. Para o exemplo apresentado, os próprios actuadores das electroválvulas teriam um mecanismo de protecção que asseguraria o seu desligar automático caso deixasse de existir comunicação durante um período de tempo longo ou passasse um tempo pré-definido sem receber um comando de desligar.

O uso de acções de desactivação é opcional. Em muitas situações pode interessar que as acções continuem até indicações em contrário. Por exemplo, o cenário "Chegar a casa do trabalho" pode ligar o ar-condicionado, o qual só será desligado pelo cenário "Ir deitar".

Um cenário pode ser suspenso, deixando de operar, e pode ser reactivado posteriormente.

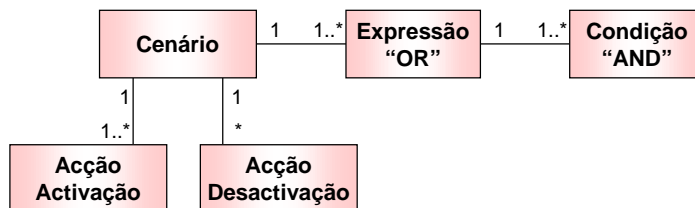


Figura 3. Modelo de um cenário.

Na figura 3 apresenta-se um modelo de um cenário, sintetizando-se os conceitos já apresentados. Como se observa, um cenário pode ter associada uma ou mais acções de activação e qualquer número de acções de desactivação (zero ou mais). A activação de um cenário é controlada por uma condição a qual pode ser tão complexa quanto se queira. Para atingir este objectivo considera-se que essa condição é definida por uma ou mais expressões entre as quais é efectuada a operação de disjunção ("OR"). Por sua vez, cada uma dessas expressões é obtida a partir da conjunção ("AND") de uma ou mais condições (condições simples envolvendo apenas o teste de uma propriedade de um dispositivo).

6. A APLICAÇÃO DOMOCONFIG

Após a descrição do método proposto para representar genericamente os dispositivos domésticos, uma habitação e especificar o comportamento desejado para o sistema, afluíram em seguida as principais opções tomadas na implementação da aplicação *DomoConfig*, a qual segue os modelos de abstracção apresentados.

A nossa escolha por uma aplicação do tipo WWW deveu-se à vantagem que lhe é inerente de permitir um acesso local e remoto à habitação, para monitorização, actuação e programação de acções. Actualmente é cada vez mais comum dispor de uma ligação em permanência à Internet (por cabo de TV ou ADSL) o que torna viável esta solução e permite a uma pessoa aceder à sua habitação de qualquer ponto do globo e usando um multiplicidade de equipamentos que suportem um *browser* comum.

A figura 4 ilustra a arquitectura proposta para a aplicação *DomoConfig*, a qual foi desenvolvida usando a plataforma .NET, a linguagem C# e o servidor IIS (*Internet Information Services*) da Microsoft.

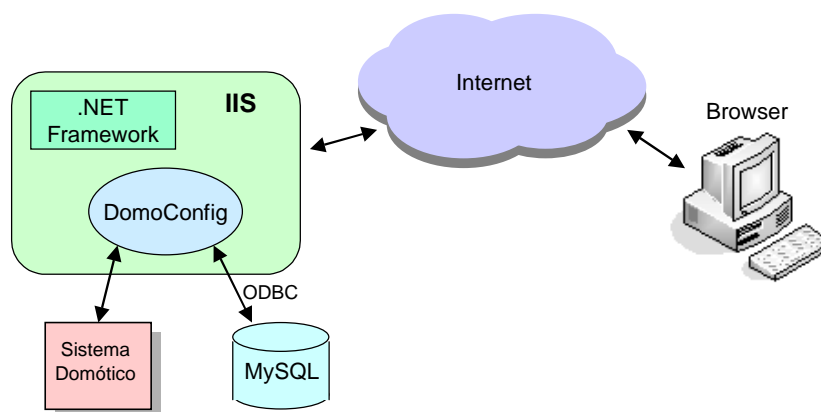


Figura 4. Arquitectura da aplicação.

Todos os dados relativos à especificação dos tipos de dispositivos domésticos e suas propriedades, dispositivos existentes, estrutura da habitação e cenários programados (condições, listas de acções e acções de desactivação), estão armazenados numa base de dados MySQL. A interface com a base de dados é efectuada usando ODBC. Encontra-se fora do âmbito do presente artigo detalhar o modelo de dados usado, convindo referir que foram seguidas as abstracções apresentadas nas figuras 1, 2 e 3.

A interacção com o sistema doméstico usa um serviço de mensagens baseado em *sockets*, sendo o conteúdo das mensagens expresso em XML. Existem dois tipos fundamentais de mensagens: *GET* e *SET*. No primeiro caso (*GET*) é identificado um dispositivo (através do seu endereço) e uma propriedade de que se pretende conhecer o valor. No segundo caso (*SET*) é identificado um dispositivo e uma propriedade, sendo indicado o valor a atribuir a essa propriedade. Na nossa implementação foi realizada uma interface com o sistema DomoBus [10].

A aplicação suporta três perfis de utilizadores: Administrador, Utilizador privilegiado e Utilizador comum. Um *Administrador* tem acesso a todas as funções e, em particular, pode adicionar ou remover utilizadores e alterar os seus perfis. O *Utilizador privilegiado* pode realizar qualquer acção no que concerne à definição de um sistema doméstico (tipos de dispositivos e propriedades, estrutura da habitação e dispositivos

existentes), monitorização e actuação (ler e escrever valores em propriedades de dispositivos) e programação de acções (criação e modificação de cenários). O *Utilizador comum* não pode modificar a definição de um sistema mas pode monitorizar ou actuar sobre dispositivos e criar, apagar ou modificar cenários.

7. CONCLUSÃO

Neste artigo foi descrita uma aplicação WWW que oferece uma forma genérica e flexível de especificar e comandar um sistema domótico, permitindo também programar o comportamento desejado. As capacidades oferecidas assentam num modelo de abstracção de dispositivos domóticos, que foi descrito. Este modelo revelou-se simples mas muito genérico e eficaz, sendo um elemento chave para o suporte de diferentes tecnologias domóticas e abrindo caminho à interoperação e à capacidade de construir sistemas heterogéneos. Idealmente esta possibilidade conduzirá a uma maior divulgação da domótica e permitirá aos utilizadores adequar as tecnologias existentes às suas necessidades e preferências.

REFERÊNCIAS

- [1] X10, <http://www.x10.com>
- [2] EIB - European Installation Bus, <http://www.eiba.com>
- [3] LonWorks (ANSI/EIA 709.1-A), <http://www.echelon.com>
- [4] CEBus - Consumer Electronics Bus (EIA-600), <http://www.cebus.org>
- [5] UPnP - Universal Plug and Play, <http://www.upnp.org>
- [6] OSGi - Open Services Gateway Initiative, <http://www.osgi.org>
- [7] Jini Network Technology, <http://java.sun.com>
- [8] Sérgio Silva, 2003. *A Modular System for Monitoring and Controlling Home Automation Systems*, MSc Thesis, Instituto Superior Técnico - Technical University of Lisbon.
- [9] Grady Booch, Ivar Jacobson, James Rumbaugh, 1998. *The Unified Modeling Language User Guide*. Addison-Wesley.
- [10] Renato Nunes, 2003. DomoBus - A new approach to home automation. *8CLEEE - 8th International Congress on Electrical Engineering*, Portugal, pp.2.19-2.24.
- [11] Renato Nunes, Sérgio Silva, Tiago Lima, 2003. A Generic Abstraction Model for Home Automation Devices. *CCCT 2003 - International Conference on Computer, Communication and Control Technologies*, Orlando, USA, pp. 242-245.