

Sistema DomoBus

Prof. Renato Nunes

Instituto Superior Técnico / Inesc-Id

renato.nunes@inesc-id.pt

Especificação XML de um Sistema DomoBus

V1.6 - 13/12/2006

(Ver historial no fim do documento)

1. Introdução

Este documento detalha a especificação XML usada para definir um sistema DomoBus. Essa especificação divide-se em duas partes: uma que define os tipos abstractos de dispositivos e suas propriedades, e outra parte que define um sistema concreto, explicitando a estrutura física de uma habitação, serviços existentes e dispositivos reais instalados. Toda esta informação está guardada num ficheiro por forma a facilitar a gestão da sua coerência. Deverão ser previstos mecanismos que permitam importar a definição dos tipos abstractos de dispositivos associados a um sistema, facilitando a sua reutilização noutros sistemas.

Procurou-se efectuar uma especificação bastante completa, não obviando no entanto a que possa ser expandida no futuro. Existem alguns aspectos, de maior detalhe, que poderão não ser essenciais em todas as implementações, consoante os objectivos das aplicações.

O presente documento assume que o leitor conhece o essencial do modelo de abstracção seguido na abordagem DomoBus (ver [1]).

2. Definição dos Tipos Genéricos de Dispositivos

Existe uma especificação dos tipos genéricos de dispositivos que existem num sistema DomoBus. Essa especificação pode ser reutilizada em múltiplos sistemas concretos (através de um mecanismo de importação dessa informação). A especificação envolve a definição dos tipos de dispositivos, tipos de propriedades, tipos de valores das propriedades e outras entidades que se detalham sem seguida.

2.1 Classes de dispositivos

Para permitir uma estruturação dos diferentes tipos de dispositivos, foi criada a noção de classe de dispositivo. Por razões de simplicidade, assume-se nesta fase que um dispositivo só pode pertencer a uma classe.

Definição de classes de dispositivos

```
<DeviceClassList>
  <DeviceClass ID="#" Name="x" />
</DeviceClassList>
```

Exemplo:

```
<DeviceClassList>
  <DeviceClass ID="1" Name="Iluminação" />
  <DeviceClass ID="2" Name="Aquecimento" />
  <DeviceClass ID="3" Name="Segurança" />
</DeviceClassList>
```

Notar que foi dada preferência ao uso de atributos pois permite um formato muito mais compacto. Se em vez de atributos fossem usados elementos vinha:

```
<DeviceClassList>
  <DeviceClass>
    <DeviceClassID> # </DeviceClassID>
    <DeviceClassName> x </DeviceClassName>
  </DeviceClass>
</DeviceClassList>
```

Comparar com:

```
<DeviceClassList>
  <DeviceClass ID="#" Name="x" />
</DeviceClassList>
```

2.2 Conversão de valores

Os dispositivos possuem várias propriedades e cada propriedade possui um valor. Os valores são representados internamente como grandezas inteiras ou sequências de bytes. Como vista a poder oferecer aos utilizadores outros tipos de representação, existem fórmulas e objectos de conversão.

As fórmulas são representadas como sequências de caracteres em que a grandeza a converter é representada pela letra 'X' (exemplos: "2+X/10", "123+2*X^3-X"). Para cada valor a converter são fornecidas duas fórmulas. Uma (identificada por "UserToSystem") permite converter a representação usada pelo utilizador na representação interna e a outra ("SystemToUser") permite efectuar a conversão inversa.

Os cálculos são efectuados em vírgula flutuante sendo o resultado visualizado com um número de casas decimais que é explicitado usando o atributo "DecimalPlaces" e devidamente arredondado. Se o número de casas decimais for zero, o valor é convertido para inteiro (adequadamente arredondado).

Antecipando situações mais complexas, prevê-se a existência de objectos de conversão disponíveis no código da aplicação e que podem ser referenciados na especificação em XML. Esta abordagem, embora permita oferecer um elevado grau de funcionalidade, é menos flexível na medida em que envolve código e, se for necessário um novo tipo de objecto de conversão, obriga a alterar o código e a gerar uma nova versão da aplicação (em Java poderá recorrer-se ao carregamento dinâmico de classes; em C# poderá recorrer-se a uma DLL delimitando o código a modificar).

Conversão de valores

```
<ConversionFormulaList>
  <ConversionFormula ID="#" Name="x"
    UserToSystem="x" SystemToUser="x" DecimalPlaces="#" />
</ConversionFormulaList>

<ConversionObjectList>
  <ConversionObject ID="#" Name="x"
    UserToSystemObj="#" SystemToUserObj="#" DecimalPlaces="#" />
</ConversionObjectList>
```

Exemplos:

```
<ConversionFormulaList>
  <ConversionFormula ID="1" Name="Converte para dobro"
    UserToSystem="2*X" SystemToUser="X/2" DecimalPlaces="0" />
  <ConversionFormula ID="2" Name="10*X+200"
    UserToSystem="10*X+200" SystemToUser="(X-200)/10" DecimalPlaces="1" />
  <!-- valor real varia entre -20.0 e 60.0. O valor 23.4 é representado por 434 -->
</ConversionFormulaList>

<ConversionObjectList>
  <ConversionObject ID="1" Name="Intensidade - lux"
    UserToSystemObj="1" SystemToUserObj="2" DecimalPlaces="2" />
</ConversionObjectList>
```

2.3 Valores das propriedades

Existem três tipos possíveis para os valores das propriedades:

- `ScalarValueType` - Escalar. Valor inteiro que pode ser representado por 8 bits (valores entre 0 e 255) ou por 16 bits (valores entre 0 e 65535).
- `EnumValueType` - Enumerado. Corresponde a pares "nome / valor" em que o valor é uma grandeza de 8 bits.
- `ArrayValueType` - Sequência de bytes (array).

Para as grandezas escalares é possível definir qual o valor mínimo e máximo que podem ter e qual o valor mínimo de incremento.

As grandezas escalares e os arrays permitem a indicação do tipo de conversão a que devem ser sujeitas ("FORMULA" ou "OBJ") e a referência da entidade de conversão. Notar que os arrays apenas suportam conversões através de objectos. Esclarece-se que os arrays podem ser usados em múltiplas situações e com múltiplas finalidades, e que também podem ser usados em casos particulares para permitir a troca de valores representados em vírgula flutuante, caso essa necessidade venha a surgir. Se os valores não necessitam de conversão é usado o tipo "NONE".

Tipos de valores

```
<ScalarValueTypeList>
  <ScalarValueType ID="#" Name="x" NumBits="#" MinValue="#" MaxValue="#" Step="#">
    <ValueConversion Type="x" Ref="#" />
  </ScalarValueType>
</ScalarValueTypeList>
```

```

<EnumValueTypeList>
  <EnumValueType ID="#" Name="x">
    <Enumerated Name="x" Value="#" />
    <Enumerated Name="x" Value="#" />
  </EnumValueType>
</EnumValueTypeList>

<ArrayValueTypeList>
  <ArrayValueType ID="#" Name="x" MaxLen="x">
    <ValueConversion Type="x" Ref="#" />
  </ArrayValueType>
</ArrayValueTypeList>

```

Exemplos:

```

<ScalarValueTypeList>
  <ScalarValueType ID="1" Name="Porcentagem (0-100)"
    NumBits="8" MinValue="0" MaxValue="100" Step="1">
    <ValueConversion Type="NONE" Ref="" />
  </ScalarValueType>
  <ScalarValueType ID="2" Name="Potência"
    NumBits="16" MinValue="0" MaxValue="800" Step="10">
    <ValueConversion Type="FORMULA" Ref="2" />
  </ScalarValueType>
</ScalarValueTypeList>

<EnumValueTypeList>
  <EnumValueType ID="1" Name="On-Off">
    <Enumerated Name="Off" Value="0" />
    <Enumerated Name="On" Value="1" />
  </EnumValueType>
  <EnumValueType ID="2" Name="Comando Ar Condicionado">
    <Enumerated Name="Desligado" Value="0" />
    <Enumerated Name="Aquecer" Value="1" />
    <Enumerated Name="Arrefecer" Value="2" />
  </EnumValueType>
</EnumValueTypeList>

<ArrayValueTypeList>
  <ArrayValueType ID="1" Name="Nome Empresa" MaxLen="10">
    <ValueConversion Type="NONE" Ref="" />
  </ArrayValueType>
  <ArrayValueType ID="2" Name="Float IEEE" MaxLen="8">
    <ValueConversion Type="OBJ" Ref="1" />
  </ArrayValueType>
</ArrayValueTypeList>

```

2.4 Tipos de dispositivos

Os dispositivos são caracterizados por uma coleção de propriedades. Essas propriedades possuem um nome e um modo de acesso que indica se podem ser lidas e escritas ("RW"), só lidas ("RO") ou só escritas ("WO"). Adicionalmente, e como já foi visto antes, as propriedades podem ser de vários tipos consoante os respectivos valores, sendo discriminadas usando "SCALAR", "ENUM" ou "ARRAY".

Tipos de dispositivos

```
<DeviceTypeList>
  <DeviceType ID="#" Name="x" RefDeviceClass="#" Description="x">
    <PropertyTypeList>
      <PropertyType ID="#" Name="x" AccessMode="x" ValueType="x" RefValueType="#" />
    </PropertyTypeList>
  </DeviceType>
</DeviceTypeList>
```

Exemplo:

```
<DeviceTypeList>
  <DeviceType ID="1" Name="Lâmpada-Regulada" RefDeviceClass="1" Description="-">
    <PropertyTypeList>
      <PropertyType ID="1" Name="On-Off" AccessMode="RW"
        ValueType="ENUM" RefValueType="1" />
      <PropertyType ID="2" Name="Intensidade" AccessMode="RW"
        ValueType="SCALAR" RefValueType="1" />
    </PropertyTypeList>
  </DeviceType>

  <DeviceType ID="2" Name="Sensor-Temperatura" RefDeviceClass="2" Description="-">
    <PropertyTypeList>
      <PropertyType ID="3" Name="Temperatura" AccessMode="RO"
        ValueType="SCALAR" RefValueType="1" />
    </PropertyTypeList>
  </DeviceType>
</DeviceTypeList>
```

3. Especificação de um Sistema Concreto

A especificação de um sistema concreto envolve a definição da estrutura da habitação, a definição dos serviços que estão disponíveis e a definição dos dispositivos domóticos existentes. Estes aspectos são detalhados em seguida.

Inclui-se também informação sobre os utilizadores e respectivos privilégios de acesso.

3.1 Definição de utilizadores e níveis de acesso

Permite-se que múltiplos utilizadores possam aceder ao sistema. Cada utilizador possui um identificador, um nome, uma palavra passe e um nível de acesso. Um dado utilizador só consegue aceder a dispositivos que tenham um nível de acesso inferior ou igual ao seu. Um utilizador pode bloquear o acesso a um dado dispositivo, caso em que apenas ele poderá efectuar o respectivo desbloqueio ou então um utilizador com nível de privilégio superior (mas não igual).

O mecanismo de privilégios descrito é simples mas permite um bom grau de funcionalidade. Futuramente poderão ser concebidos mecanismos mais complexos.

Na abordagem proposta os níveis de acesso são fixos, correspondendo a um número inteiro que pode variar entre 0 e 10 e em que o nível 0 é o que possui menos privilégios. O valor superior pode ser aumentado, embora não se antevêja que tal venha a ser necessário. Todos os valores dentro da gama definida são válidos e absolutos (o nível 5 é sempre superior ao nível 4).

Para facilitar o uso dos níveis de acesso existe uma secção que permite associar a cada nível um nome. Caso um nível não tenha um nome definido, será usado um nome por omissão (por exemplo, para o nível 8, poderá ser usado "Access Level 8").

Utilizadores e níveis de acesso

```
<AccessLevelList>
  <AccessLevel Level="#" Name="x" />
</AccessLevelList>

<UserList>
  <User ID="#" Name="x" Password="x" AccessLevel="#" />
</UserList>
```

Nota: Sugere-se que campos "Name" e "Password" conttenham valores cifrados usando, por exemplo, 3DES.

Exemplos:

```
<AccessLevelList>
  <AccessLevel Level="0" Name="Free Access" />
  <AccessLevel Level="1" Name="Guest" />
  <AccessLevel Level="2" Name="Common User - Child" />
  <AccessLevel Level="3" Name="Common User - Parent" />
  <AccessLevel Level="10" Name="Administrator" />
</AccessLevelList>

<UserList>
  <User ID="1" Name="abcd" Password="efgh" AccessLevel="2" />
  <User ID="2" Name="ijkl" Password="mnop" AccessLevel="10" />
</UserList>
```

3.2 Definição da Estrutura da Habitação

Considera-se que uma habitação é constituída por um ou mais pisos e que cada piso pode possuir diversas divisões. Cada piso terá um nome e uma ordem que é usada para conhecer a ordem relativa dos pisos em altura (o sótão terá uma ordem superior ao r/c e este terá uma ordem superior à cave). Cada divisão possui um nome e um nível de acesso. Este nível de acesso será usado como valor por omissão para o nível de acesso dos dispositivos aí localizados. Os identificadores das divisões são globais (cada divisão possui um identificador único, independentemente do piso a que está associada).

A presente especificação foi adoptada pela sua simplicidade e não contempla relações de vizinhança ou de ligação entre divisões e entre pisos. Estas relações poderão ser necessárias para certo tipo de aplicações, havendo na altura que proceder à expansão desta especificação.

Estrutura da habitação

```
<House ID="#" Name="x" Address="x" Phone="x">
  <FloorList>
    <Floor ID="#" Name="x" HeightOrder="#">
      <DivisionList>
        <Division ID="#" Name="x" AccessLevel="#" />
      </DivisionList>
    </Floor>
  </FloorList>
</House>
```

Exemplo:

```
<House ID="1" Name="Vivenda Silva" Address="xxxxx" Phone="xxxx">
  <FloorList>
    <Floor ID="1" Name="Cave" HeightOrder="-1">
      <DivisionList>
        <Division ID="1" Name="Cave" AccessLevel="1" />
      </DivisionList>
    </Floor>
    <Floor ID="2" Name="R/C" HeightOrder="0">
      <DivisionList>
        <Division ID="2" Name="Cozinha" AccessLevel="3" />
        <Division ID="3" Name="Sala" AccessLevel="2" />
        <Division ID="4" Name="Hall" AccessLevel="2" />
      </DivisionList>
    </Floor>
    <Floor ID="3" Name="Sótão" Order="1">
      <DivisionList>
        <Division ID="5" Name="Sótão" AccessLevel="2" />
      </DivisionList>
    </Floor>
  </FloorList>
</House>
```

3.3 Definição de serviços

É usado o conceito de serviço para agrupar dispositivos associados a uma dada funcionalidade. É possível um dispositivo estar afecto a mais de que um serviço (Exemplo: detector de presença pode estar associado ao serviço de detecção de intrusão e ao serviço de iluminação).

Definição de serviços

```
<ServiceList>
  <Service ID="#" Name="x" />
</ServiceList>
```

Exemplo:

```
<ServiceList>
  <Service ID="1" Name="Aquecimento">
  <Service ID="2" Name="Iluminação">
  <Service ID="3" Name="Alarme">
  <Service ID="4" Name="Tomadas">
  <Service ID="5" Name="Irrigação">
</ServiceList>
```

3.4 Dispositivos Domóticos

Os dispositivos domóticos existentes são definidos nesta secção e fazem referência aos serviços a que estão associados, à divisão em que estão localizados e ao seu nível de acesso. Este nível de acesso pode discriminar operações de monitorização e comando. Para tal, são indicados dois valores separados por uma vírgula. O primeiro valor refere-se à capacidade de monitorização e o segundo à capacidade de comando. Se for apenas indicado um valor ele aplica-se a ambas as capacidades. Um utilizador pode bloquear o acesso (leitura, escrita ou ambos) a um dispositivo, desde que o seu nível de privilégio seja adequado (maior ou igual ao do dispositivo). Quando isso ocorre, apenas ele próprio ou alguém com mais privilégios pode desbloquear o acesso ao dispositivo. O atributo "UserBlocked" pode conter '-' ou o identificador do utilizador que bloqueou o acesso para leitura e/ou escrita. A partir dos identificadores dos utilizadores é possível conhecer os seus níveis de privilégio.

Para além dos valores já descritos, um dispositivo inclui o seu endereço físico que é necessário para poder comunicar com ele.

Definição dos dispositivos

```
<DeviceList>
  <Device ID="#" RefDeviceType="#" Name="x"
    Address="#" RefDivision="#" AccessLevel="#,#" UserBlocked="#,#">
    <DeviceServiceList>
      <DeviceService RefService="#" />
    </DeviceServiceList>
  </Device>
</DeviceList>
```

Exemplo:

```
<DeviceList>
  <Device ID="1" RefDeviceType="1" Name="Candeeiro-Mesa"
    Address="#0100" RefDivision="2" AccessLevel="3,5" UserBlocked="-,27" />
  <!-- Dispositivo não bloqueado para monitorização (-) e bloqueado para comando pelo utilizador
    com o id 27; só o utilizador 27 pode desbloquear ou alguém com privilégio superior a ele -->
  <DeviceServiceList>
    <DeviceService RefService="1" />
    <DeviceService RefService="4" />
  </DeviceServiceList>
</Device>
</DeviceList>
```

4. Monitorização e Comando de uma Habitação

Tendo por base a especificação de um sistema domótico, descrita atrás, torna-se possível desenvolver aplicações de interface genéricas que permitem o comando e a monitorização de qualquer sistema. Essas interfaces funcionarão de forma dinâmica, interpretando o conteúdo XML do ficheiro relativo às características de base do sistema DomoBus e o conteúdo do ficheiro que explicita as características específicas de uma dada habitação e de um dado sistema em concreto.

As aplicações de interface devem permitir um acesso aos dispositivos usando uma lógica espacial (localização dos dispositivos) ou uma lógica em que os dispositivos

são organizados em função do serviço a que estão associados. São também possíveis abordagens mistas. Por exemplo, no caso de uma divisão com um número muito elevado de dispositivos, em vez de os listar a todos, poderiam ser apenas mostrados os dispositivos de determinado serviço.

4.1 Navegação espacial

Processo de navegação em que é representada a habitação e a sua decomposição em pisos (se for o caso). Ao seleccionar um piso são mostradas as divisões que o compõem. E, ao seleccionar uma divisão são visualizados os dispositivos existentes. Ao escolher um dispositivo são visualizadas as suas propriedades com os respectivos valores. Se o utilizador pretender alterar o estado de um dispositivo, basta modificar o valor da propriedade relevante.

4.2 Navegação por serviço

Em alternativa à navegação espacial, pode existir uma outra forma de aceder ao dispositivos em que é seleccionado um dado serviço, sendo então visualizados os dispositivos afectos a esse serviço.

4.3 Preferências dos utilizadores

Em certos casos pode fazer sentido o utilizador poder explicitar divisões ou dispositivos que usa com mais frequência, permitindo acelerar o acesso a eles. Pode-se conceber uma zona de "favoritos" à qual o utilizador pode adicionar ou remover divisões e adicionar ou remover dispositivos. As divisões referidas podem ser físicas ou lógicas servindo neste último caso como meio de agregação de dispositivos.

Esta informação é específica para cada utilizador e, a existir, será objecto de uma especificação XML própria.

4.4 Alarmes

A vertente da segurança na habitação é muito importante. É pois relevante a possibilidade de identificar dispositivos que, ao mudarem de estado, assinalam ocorrências das quais os utilizadores deverão ser informados. Estes alarmes, deverão ser visualizados em qualquer ecrã da interface, permitindo aceder-lhes rapidamente e tomar conhecimento das situações existentes.

Este aspecto será objecto de uma futura especificação.

4.5 Acções compostas - Macros

Tem interesse poder dispor de elementos da interface que possam desencadear múltiplas acções com uma única actuação. Normalmente a interface apenas permite aceder aos dispositivos e às suas propriedades de uma forma individual. Porém, é fácil conceber entidades que se comportam como "macros" e que, quando activadas, desencadeiam múltiplas acções. O uso de macros permite lidar facilmente com situações tais como "ver TV", "preparar para dormir", "sair de casa", etc, sendo óbvia a sua utilidade.

Ilustra-se em seguida a definição de macros.

Definição de macros

```
<MacroList>
  <Macro ID="#" Name="x">
    <ActionList>
      <Action ID="#" Name="x" RefDevice="#" RefProperty="#" Value="x" />
    </ActionList>
  </Macro>
</MacroList>
```

Exemplo:

```
<MacroList>
  <Macro ID="1" Name="Ver TV">
    <ActionList>
      <Action ID="1" Name="Fecha Estore" RefDevice="1" RefProperty="0" Value="0" />
      <Action ID="2" Name="Candeeiro 50%" RefDevice="5" RefProperty="1" Value="50" />
      <Action ID="3" Name="LampTecto 40%" RefDevice="9" RefProperty="1" Value="40" />
    </ActionList>
  </Macro>
</MacroList>
```

5. Estado dos dispositivos

Quando se acede a um dispositivo é visualizada informação sobre o seu estado actual, ou seja, sobre os valores das suas propriedades. Assume-se que essa informação está disponível numa entidade externa. Para saber o valor de uma propriedade de um dado dispositivo deverá ser enviada uma mensagem para a entidade adequada que fornecerá a informação pretendida e que é mostrada ao utilizador.

Para mais detalhes sobre as comunicações consultar o documento [2]. Para simplificar o teste das aplicações, o objecto de comunicação descrito em [2] pode ser simulado, sendo usada uma representação interna do estado de todos os dispositivos. Esse estado pode ser guardado em ficheiro e lido, permitindo simular um efeito de persistência de acções efectuadas pelos utilizadores. Abaixo propõe-se um possível formato para o registo da informação de estado dos dispositivos. Notar que foi contemplado um campo para simular a operacionalidade ou não dos dispositivos (ver secção 4 de [2]).

Sobre a actuação, ou seja, a modificação de valores de propriedades, deverá ser seguida a abordagem descrita a seguir. Quando um utilizador define um novo valor para uma dada propriedade deverá ser assinalado, em termos da interface, que a actuação foi desencadeada. No entanto, não deverá ser assumido imediatamente o sucesso da actuação. Deverá ser mostrada uma informação indicando ao utilizador para aguardar, sendo enviada a mensagem adequada para o dispositivo. Se a mensagem for confirmada, então o novo estado é efectivo. Caso contrário, deverá ser apresentada uma indicação de erro ao utilizador, desconhecendo-se o estado em que se encontra o dispositivo.

Informação auxiliar para registo do estado de um sistema

```
<DeviceStateList>
  <DeviceState RefDevice="#" RefProperty="#" Value="x" OperationalStatus="#" />
</DeviceStateList>
```

Se "OperationalStatus" for 1 isso indica um dispositivo operacional. Se for 0 indica um dispositivo que o qual não é possível comunicar (desconhece-se o seu estado e não é possível efectuar actuações sobre ele).

Caso se justifique, pode ser adicionada informação adicional como, por exemplo, a data da última actuação e identificação do utilizador.

Referências

- [1] Renato Nunes, "Modelo de Especificação e Programação de um Sistema Domótico", IADIS Conferência Ibero-Americana WWW/Internet 2004, Madrid, Spain, October 2004, pp. 377-384.
- [2] Renato Nunes, "API de Comunicação do Sistema DomoBus", Relatório Interno, IST, Abril 2005. (<http://mega.ist.utl.pt/~ic-eid/papers>)

Historial das revisões deste documento

V1.6 - 13/12/2006

- Na secção 3.2 foi explicitado que os identificadores das divisões são globais (cada divisão possui um identificador único).

V1.5 - 22/11/2006

- Foi acrescentado o atributo "Name" aos elementos "ConversionFormula" e "ConversionObject" (secção 2.2) e aos elementos "ScalarValueType", "EnumValueType" e "ArrayValueType" (secção 2.3). A existência do novo atributo visa simplificar a identificação desses elementos pelo utilizador aquando da sua criação e posterior referência. Os exemplos foram ajustados passando a incluir o novo atributo.
- Por razões análogas às descritas acima foi também acrescentado o atributo "Name" aos elementos "Action" (secção 4.5).
- Foram eliminadas as referências ao uso de dois ficheiros para conterem a especificação de um sistema DomoBus: um ficheiro com a especificações dos tipos de dispositivos e informação associada; e outro ficheiro com a especificação de um sistema concreto - utilizadores, topologia da habitação, dispositivos domóticos concretos, etc. A experiência demonstra que é mais robusto e evita problemas de coerência da informação usar apenas um ficheiro com toda a informação (de outro modo haveria que garantir que os dois

V1.6

ficheiros usados correspondiam à mesma versão e eram coerentes). Para permitir reutilizar a especificação de aspectos genéricos que tipicamente serão comuns à generalidade dos sistemas DomoBus, deverão ser previstos mecanismos que permitam importar definições que estejam presentes na especificação de outro sistema.

- Foi clarificado o exemplo relativo ao bloqueio de acesso a dispositivos domóticos por um utilizador (atributo "UserBlocked") - ver secção 3.4.

V1.4 - 09/12/2005

- Foram efectuadas pequenas correcções nos exemplos relativos aos tipos de dispositivos (secção 2.4) e à definição da estrutura da habitação (secção 3.2).

V1.3 - 15/11/2005

- Foi corrigido o exemplo relativo aos tipos de valores (secção 2.3).

V1.2 - 04/11/2005

- Foi modificada a secção relativa aos valores das propriedades (secção 2.3). Esses valores passaram a ser de três tipos: escalares, enumerados e "arrays". Os valores escalares podem ser representados por 8 bits (valores entre 0 e 255) ou por 16 bits (valores entre 0 e 65535). A secção 2.4 foi modificada de modo a ficar coerente com as alterações feitas na secção 2.3.

V1.1 - 09/06/2005

- Foi modificada a secção relativa à definição de utilizadores e níveis de acesso (secção 3.1), detalhando-se melhor a abordagem proposta. Foram mudados os nomes de alguns atributos: atributo "ID" do elemento AccessLevel passou a "Level" e o atributo "RefAccessLevel" (existente em vários elementos) passou a "AccessLevel".
- Foi corrigido o XML do exemplo da Estrutura da Habitação (secção 3.2) no que refere aos IDs do último "Floor". Cada piso (*Floor*) tem um ID único. Cada divisão (*Division*) tem também um ID único (independentemente do piso a que está associada).

V1.0 - 26/04/2005

- Versão inicial do documento