

6

NCL em Ambientes de Computação Ubíqua

Os cenários descritos no Capítulo 1 mostram diferentes tipos de aplicações compondo as apresentações. Tradicionalmente, sistemas multimídia utilizam mídia contínua (e.g. áudio e vídeo) e mídia estática (e.g. imagens e páginas web) em apresentações. Todavia, este trabalho, além de mídia contínua e estática, propõe o uso de aplicações *single-user*, como um navegador tridimensional e o Microsoft PowerPoint®, para compor a apresentação multimídia, que na verdade são aplicações que possuem ilimitadas possibilidades de execução que se refletem em infinitos estados. Ao mesmo tempo que tais aplicações são extremamente interessantes em uma apresentação, elas criam novos desafios de representação.

Os requisitos de representação levantados pelos nossos protótipos são mais facilmente alcançados através de NCL do que de SMIL. O principal motivo é a existência de atributos. O segundo motivo é a facilidade de ligação entre componentes de mídia oferecido por NCL. Isso ocorre porque a representação de relações complexas de sincronia em SMIL através dos tipos básicos *par*, *seq* e *excl* é geralmente feita através de aninhamento [42] [58]. Através da flexibilidade de criação de elos de ligação oferecida por NCL, é possível representar tais relações complexas de sincronia de forma mais simples do que SMIL. Isso é especialmente verdadeiro em ambientes de computação ubíqua que possuem uma grande variedade de aplicações e estados de execução. Desta forma, é possível fazer com que certas ações ocorram apenas em determinadas condições de disparo.

O conector hipermídia XConnector, um dos módulos de NCL, permite a separação entre relações e participantes, possibilitando que usuários mais experientes criem uma biblioteca de relações, enquanto usuários menos experientes apenas as utilizem nas apresentações. Desta forma, XConnector também pode facilitar a autoria de documentos, por exemplo através de uma interface gráfica que ofereça um conjunto de conectores para que o usuário escolha qual conector utilizar e como ligar os papéis aos diferentes tipos de mídia. Mais ainda, XConnector oferece a possibilidade de criação

de expressões para as condições de disparo.

Outra característica do XConnetor é a utilização de uma interface comum de conexão entre as aplicações através da definição de papéis (*roles*) e de uma cola (*glue*). Essa interface uniformizada permite que cada aplicação seja responsável por verificar quando uma âncora é atingida, facilitando a adição de novas aplicações.

Portanto, propomos a representação de apresentações multimídia em espaços ativos através de NCL. Nas próximas seções são apresentadas propostas de utilização e adaptações à linguagem NCL para que seja possível acomodar os diversos requisitos apresentados.

6.1

Dispositivos de Execução

Ambientes de computação ubíqua, por definição, utilizam múltiplos dispositivos, e heterogêneos. Nota-se que da mesma forma que os computadores pessoais possuem uma enorme variedade de hardware, em ambientes de computação ubíqua essa variedade se torna ainda mais acentuada. Salas computacionais diversas possuem diferentes números de computadores, dispositivos de entrada e de saída, processadores, aplicações, entre outras características.

Portanto, a representação de uma apresentação requer a descrição de dispositivos através de características que possam ser mapeadas entre diferentes ambientes computacionais. Uma forma de se fazer isso é, por exemplo, utilizarmos características do dispositivos como tamanho da tela ou capacidade de processamento ao invés de seus nomes.

Propomos uma solução na qual a representação de tais características é feita através de tuplas nome-valor e representadas através de arrays associativos. As características especificadas podem ser verificadas através de um Serviço de Detecção Semântica (Cap. 2). Além disso, cada dispositivo pode também definir regiões de tela, da mesma forma que NCL originalmente permite.

Outra representação útil em aplicações multimídia é a reserva de recursos dos dispositivos. Atributos de reserva de recurso, apesar de opcionais, podem fazer parte da base de descritores das mídias. A infra-estrutura *ActivePresentation* pode ser facilmente adaptada para seguir tais descrições através do sistema DSRT [14]. Na Seção 6.5 é possível verificar como a representação da reserva de recursos pode ser incluída em NCLua.

6.2 O Tipo Replica

Ambientes de computação ubíqua são por definição ambientes que possuem múltiplos dispositivos de entrada e saída. Todavia, gostaríamos que a definição do documento multimídia não fosse restrita pelo aparato computacional de apenas uma sala, assim como não gostaríamos de utilizar um conjunto reduzido de dispositivos para garantir a execução da apresentação em qualquer ambiente. A solução que adotamos para este problema foi o uso de replicação da execução de aplicações. Réplicas são facilmente introduzidas ou retiradas de uma apresentação e, quando aliadas a uma sala rica em dispositivos, auxiliam a imersão da platéia na apresentação. Desta forma, a utilização de réplicas também ajuda o usuário a obter um melhor aproveitamento da apresentação. Entretanto a utilização de um número grande de mídias e conteúdos distintos pode fazer com que a audiência não seja capaz de absorver o conteúdo passado. Ainda é preciso analisar novas formas de interação humano-computador nesses ambientes de computação ubíqua.

A implementação de sincronia entre réplicas em diferentes máquinas é muito dependente do tipo de mídia utilizado. Considere como exemplo a reprodução de um vídeo. É necessário que a sincronização entre os diferentes computadores seja capaz de reproduzir os vídeos com uma diferença de tempo da ordem de poucos milisegundos [31]. Nossos experimentos mostram que 3 mili-segundos já podem ser percebidos visualmente.

Tal refinamento de sincronização não pode ser realizado no nível de descrição do documento, pois requer a construção de um sistema específico para este fim. No exemplo de um grupo de vídeo, é necessário levar em consideração atrasos de rede e as características inerentes da aplicação usada para reproduzir o vídeo.

Portanto, para a representação de um conjunto de réplicas propomos a criação de um novo tipo de composição. Similar ao módulo *BasicComposite*, propomos o módulo *BasicReplica*. Esse tipo deve definir como atributos o tipo do elemento a ser executado, o identificador do grupo, opcionalmente o arquivo de dados a ser apresentado e a lista de âncoras. Além disso, é necessário indicar os componentes do grupo de réplicas. Note que, como já havíamos citado, o tipo *replica* pode conter componentes opcionais. Por exemplo, na Seção 6.5.2 veremos um grupo de réplicas em que a segunda instância do PowerPoint® depende da existência de um laptop com resolução de 1024x768 no ambiente computacional onde a apresentação é mostrada.

Um importante requisito que também deve ser satisfeito é a substituição do conteúdo apresentado em um dispositivo. Em NCL isso já era permitido através da substituição de conteúdo em uma região espacial. Porém, principalmente no caso de réplicas que são muitas vezes descartáveis durante uma apresentação, essa possibilidade se torna ainda mais interessante em ambientes de computação ubíqua. Nota-se que na nossa representação, através de um mapeamento para dentro do elemento *replica*, é possível terminar um nó individualmente.

Futuramente, o tipo *replica* pode ser modificado para que a organização espacial das réplicas seja especificada através de semânticas, oferecendo mais adaptação ao documento. Por exemplo, poderíamos especificar que vídeos fossem sempre executados em computadores dispostos lado a lado.

6.3 Novas Aplicações

Em NCL os descritores são responsáveis pela descrição da aplicação enquanto os componentes da apresentação apenas definem e usam os tipos de mídia. Por exemplo, ao invés de tratar o Windows Media Player como uma aplicação na descrição do documento, o tipo *video* é descrito no módulo *BasicMedia*. Dessa forma, vídeos podem ser apresentados através de qualquer aplicação, contanto que sigam as diretrizes de apresentação da mídia em questão.

Uma das características dos ambientes de computação ubíqua é a heterogeneidade das aplicações. Neste trabalho a maioria das aplicações utilizadas exigiu a criação de um novo tipo de mídia. Criamos para o PowerPoint® o tipo *ppt* (Seção 6.5.2); para obter informações relativas ao espaço físico, criamos o tipo *activespace* (Seção 6.5.3); para a aplicação de visualização 3D utilizamos o tipo *vis3d* (Seção 6.5.4).

Esses nomes podem ser utilizados para que o formatador da apresentação inicie as aplicações nos diferentes dispositivos. Nota-se porém que, para que a aplicação possa ser executada no dispositivo remoto, ela precisa receber informações sobre as âncoras, avisando ao formatador quando estas são alcançadas. Na próxima Seção abordaremos essa questão com mais profundidade.

6.4 Conexão Entre as Aplicações

Com a adição de novos tipos de aplicações, novas formas de mídia precisam ser tratadas. Essas novas mídias exigem também novas formas de segmentar seu conteúdo. Por exemplo, uma aplicação de localização introduz o conceito de um espaço tridimensional que atualmente não está especificado na linguagem NCL. Além disso, em algumas aplicações as âncoras são representadas por intervalos enquanto em outros casos elas são valores absolutos. Com o objetivo de relacionar essas novas mídias é necessário que o documento possua expressividade para representar também as regiões das novas aplicações, oferecendo uma interface para o nó através da sua lista de âncoras.

Para representar as novas regiões estendemos o módulo *MediaInterface* para a criação dos novos tipos de âncoras. Os exemplos apresentados na Seção 6.5 exigiram a definição do tipo *boundingbox* com dois pontos com três coordenadas cada. Esses pontos criam fronteiras tridimensionais que podem ser utilizadas pelas aplicações, por exemplo, para detectar se uma entidade entrou ou saiu de dentro de uma fronteira. Além do tipo *boundingbox* precisamos do tipo *slides*, que possui um valor unidimensional correspondente ao número do slide PowerPoint®.

Nota-se que a semântica das âncoras é dependente da aplicação. Por exemplo, a aplicação de rastreamento de coordenadas pode utilizar as coordenadas passadas para verificar se um determinado dispositivo de localização está contido dentro de uma área ou se todos os dispositivos de localização estão dentro dessa área. Observamos portanto que, à medida que as aplicações são utilizadas em apresentações multimídia distribuídas, novas formas de utilização das âncoras irão surgir.

Outro ponto é a necessidade de parâmetros na âncora. Usando novamente o exemplo da aplicação de localização de objetos espaciais, percebemos que podemos fazer uso de diversos localizadores distintos. Através de parâmetros gostaríamos de representar no arquivo que, para que uma âncora seja acionada, um dispositivo em particular precisa estar dentro de uma região.

Da mesma forma que o XConnector permite a criação de novas formas de ligação entre aplicações, um módulo para estender a inclusão de aplicações também pode permitir que usuários mais experientes escrevam tais arquivos e os usuários menos experientes possam usá-los. Nota-se que em nossos exemplos não tratamos dessa questão em detalhes e deixamos

para que trabalhos futuros possam tratar disso.

Além das âncoras que representam regiões do conteúdo, cada aplicação pode possuir atributos. Por exemplo, um atributo da aplicação de reprodução de áudio especifica se gostaríamos de visualizar uma barra de controle de tempo ou não. Para que possamos ajustar esses atributos individualmente em cada computador, descrevemos esse atributo como configuração da aplicação. Novamente, isso é um indicativo de que uma maior flexibilização da descrição dos tipos de aplicação se faz necessário. Por exemplo, em nosso protótipo do trailer de *starwars* (Seção 6.5.2), usamos um controlador da posição de vídeo dentro do Windows Media Player® para retroceder ou avançar a apresentação. Da mesma forma, os atributos de controle de volume são parte integrante da aplicação, uma vez que cada aplicação pode possuir uma configuração independente.

6.5 Representações NCLua

Apresentamos a seguir exemplos de documentos multimídia distribuídos baseados na linguagem NCL 2.1, porém representados em NCLua com as extensões propostas (com exceção do exemplo da Seção 6.5.1 que é uma versão simplificada do formato). A utilização de Lua para a representação do documento multimídia distribuído foi feita por conveniência. Da mesma forma que documentos XML, a linguagem Lua [7, 41] é capaz de representar documentos semi-estruturados através de tabelas associativas. Os documentos apresentados podem facilmente ser convertidos de Lua para XML.

6.5.1 O Trailer de Starwars - original

Como o objetivo deste trabalho não é a construção de um sistema de execução completo, mas sim averiguar as necessidades de execução e representação de apresentações multimídia distribuídas em espaços ativos, construímos um sistema simplificado de um formatador capaz de interpretar o documento a seguir.

Outra versão deste documento, que segue o formato NCLua, pode ser encontrado na Seção 6.5.2.

```
datadir = "c:\\activepres-documents\\"
```

```
presentation = {
  head = {
    layout = {
      resources = {
        {
          id = "plasma1",
          properties={hostname="plasma1-3105.ad.uiuc.edu"},
        },
        {
          id = "plasma2",
          properties={hostname="plasma2-3105.ad.uiuc.edu"},
        },
        {
          id = "plasma3",
          properties={hostname="plasma3-3105.ad.uiuc.edu"},
        },
        {
          id = "plasma4",
          properties={hostname="plasma4-3105.ad.uiuc.edu"},
        },
        {
          id = "mark",
          properties={hostname="stramp"},
        },
      }
    },
    appbase = {
      {
        id="wmpgroup",
        group=true,
        application="WMP::IWMPCore",
        src=datadir.."starwars.mpg",
        areas={
          {begin=0,    finish=12.5},    -- 20th century fox
          {begin=12.6, finish=21},     -- A long time ago...
          {begin=21.1, finish=31},     -- description
          {begin=31.1, finish=36},     -- empire image
          {begin=36.1, finish=38.6},   -- Darth Vader
          {begin=38.7, finish=47},     -- Destroy the Jedi
        }
      }
    }
  }
}
```

```

        {begin=47,    finish=59},    -- lava slide
        {begin=59.1, finish=63},    -- yes master
        {begin=63.1, finish=76},    -- join the empire
        {begin=76.1, finish=76.9},  -- black
        {begin=77,    finish=92},    -- action
        {begin=92.1, finish=100},    -- revenge of the sith
        {begin=100.1, finish=107.8}, -- credits
        {begin=107.9, finish=110},  -- kill everything
    },
},
{
    id="pptgroup",
    group=true,
    application="PPT::SlideShowView",
    src=datadir.."starwars.ppt",
    areas={
        {begin=1,finish=1},
        {begin=2,finish=2},
        {begin=3,finish=3},
        {begin=4,finish=4},
        {begin=5,finish=5},
        {begin=6,finish=6},
        {begin=7,finish=7},
        {begin=8,finish=8},
        {begin=9,finish=9},
        {begin=10,finish=10},
        {begin=11,finish=11},
        {begin=12,finish=12},
        {begin=13,finish=13}}
    },
},
},
body={
    components = {
        {
            id="1",
            base="pptgroup",
            machine="plasma1",
        },
    },
},

```



```
{
  id="2",
  base="wmpgroup",
  machine="plasma2",
},
{
  id="3",
  base="pptgroup",
  machine="plasma3",
},
{
  id="4",
  base="wmpgroup",
  machine="plasma4",
},
{
  id="5",
  base="wmpgroup",
  machine="mark",
},
},
linkBase={
  driver="wmpgroup",
  {
    {appbase="wmpgroup", area = "1"}, -- Restrição área 1 ser apresentada
    {appbase="pptgroup", area = "1"}, -- junta ao mesmo tempo do slide 1
    driver="wmpgroup",
  },
  {
    {appbase="wmpgroup", area = "2"},
    {appbase="pptgroup", area = "2"},
    driver="wmpgroup",
  },
  {
    {appbase="wmpgroup", area = "3"},
    {appbase="pptgroup", area = "3"},
    driver="wmpgroup",
  },
  {
```

```
{appbase="wmpgroup", area = "4"},
{appbase="pptgroup", area = "4"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "5"},
{appbase="pptgroup", area = "5"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "6"},
{appbase="pptgroup", area = "6"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "7"},
{appbase="pptgroup", area = "7"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "8"},
{appbase="pptgroup", area = "8"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "9"},
{appbase="pptgroup", area = "9"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "10"},
{appbase="pptgroup", area = "10"},
driver="wmpgroup",
},
{
{appbase="wmpgroup", area = "11"},
{appbase="pptgroup", area = "11"},
driver="wmpgroup",
},
},
```

```

    {
      {appbase="wmpgroup", area = "12"},
      {appbase="pptgroup", area = "12"},
      driver="wmpgroup",
    },
    {
      {appbase="wmpgroup", area = "13"},
      {appbase="pptgroup", area = "13"},
      driver="wmpgroup",
    },
    {
      {appbase="wmpgroup", area = "14"},
      driver=nil,
    },
  },
},
}

```

6.5.2 O Trailer de Starwars - revisto

Abaixo apresentamos um arquivo NCLua capaz de representar o mesmo conjunto de informações apresentadas na Seção anterior. A maior diferença entre este exemplo e o anterior é o uso dos conectores hipermídia (XConnector). Os conectores utilizados neste e nos próximos exemplos são mostrados no apêndice (Cap. A).

```

presentation = {
  head = {
    resources = {
      {
        id = "bigscreen1",
        properties={sound="yes",resolution="1024x768"},
        layout = {
          id="window1", width="1024", height="768",
          region={id="1", width="100%", height="100%"}
        }
      },
      {
        id = "bigscreen2",

```

```
        properties={sound="no",resolution="1024x768"},
        layout = {
            id="window1", width="1024", height="768",
            region={id="1", width="100%", height="100%"}
        }
    },
    {
        id = "laptop1",
        properties={resolution="1024x768"},
        layout = {
            id="window1", width="1024", height="768",
            region={id="1", width="100%", height="100%"}
        }
    },
    {
        id = "laptop2",
        properties={resolution="1024x768"},
        layout = {
            id="window1", width="1024", height="768",
            region={id="1", width="100%", height="100%"}
        }
    },
    },
    descriptorBase={
        {id="loc1", machine="bigscreen1", region="window1",
        process-priority="HIGH"},
        {id="loc2", machine="bigscreen2", region="window1",
        process-priority="HIGH"},
        {id="loc3", machine="laptop1", region="window1"},
        {id="loc4", machine="laptop2", region="window1"},
    }
    },
    body={
        {type="port", id="startup", component="starwars", port="wmpgroup"},
        {type="composite", id="starwars",
        {type="replica",
        {type="video",
        id="wmpgroup",
        src="starwars.mpg",
```

```

    areas={{id=1, begin=0,    finish=12.5}, -- 20th century fox
           {id=2, begin=12.6, finish=21},  -- A long time ago...
           {id=3, begin=21.1, finish=31},  -- description
           {id=4, begin=31.1, finish=36},  -- empire image
           {id=5, begin=36.1, finish=38.6}, -- Darth Vader
           {id=6, begin=38.7, finish=47},  -- Destroy the Jedi
           {id=6, begin=47,    finish=59},  -- lava slide
           {id=7, begin=59.1, finish=63},  -- yes master
           {id=8, begin=63.1, finish=76},  -- join the empire
           {id=9, begin=76.1, finish=76.9}, -- black
           {id=10,begin=77,    finish=92},  -- action
           {id=11,begin=92.1, finish=100},  -- revenge of the sith
           {id=12,begin=100.1,finish=108}} -- credits 107.8
  },
  components={{id="wmp1",descriptor="loc1"},
              {id="wmp2",descriptor="loc2"}}
},
{type="replica",
  {type="ppt",
    id="pptgroup",
    src="starwars.ppt",
    enableTimeBar="yes",
    slides={{id=1, number=1},
            {id=2, number=2},
            {id=3, number=3},
            {id=4, number=4},
            {id=5, number=5},
            {id=6, number=6},
            {id=7, number=7},
            {id=8, number=8},
            {id=9, number=9},
            {id=10,number=10},
            {id=11,number=11},
            {id=12,number=12},
            {id=13,number=13}},
    components={{id="ppt1",descriptor="loc3"},
                {id="ppt2",descriptor="loc4",optional="yes"}}
  },
},
},

```

```

linkBase={
  {
    id="link1", xconnector="starts.xml",
    {component="wmpgroup", port="1",
      role="on_x_presentation_begin"},
    {component="pptgroup", port="1", role="start_y"},
  },
  {
    id="link2", xconnector="starts.xml",
    {component="wmpgroup", port="2",
      role="on_x_presentation_begin"},
    {component="pptgroup", port="2", role="start_y"},
  },
  --...
},
},
},
}

```

6.5.3 Olhos Que Seguem

Uma das cenas do projeto CyberOpera foi a utilização de um vídeo com imagens de olhos que seguem um sensor de localização preso a um ator. Abaixo apresentamos o arquivo NCLua capaz de representar tal cena.

Tentamos desenvolver essa idéia através de um protótipo construído sobre a infra-estrutura *ActivePresentation*, porém a implementação mostrou resultados insatisfatórios devido a limitações do Windows Media Player®. Para este exemplo, precisamos de um reprodutor de vídeo que seja capaz de eficientemente avançar e retroceder sua reprodução rapidamente. Apesar de oferecer opções para fazer isso, o Windows Media Player® não ofereceu desempenho satisfatório para a realização desta atividade em tempo real.

```

presentation = {
  head = {
    resources = {
      {
        id = "enormousscreen",
        properties={resolution="4096x3072",memory="16MB",

```

```

        cpu="3GHz"},
    layout = {
        id="window1",width="4096",height="3072",
        region={id="r1",width="100%",height="100%"}
    }
},
{
    id = "ubisense-server",
    properties={hostname="ubisense-server"},
},
},
descriptorBase={
    {id="loc1",machine="enormousscreen",region="window1",
        process-priority="HIGH"},
    {id="loc2",machine="ubisense-server"},
}
},
body={
    {type="port",id="startup",component="followingeyes",
        port="space1"},
    {type="composite",id="followingeyes",
        {type="activespace",
            id="space1",
            descriptor="loc2",
            boundingbox={{id="1", p1x=0, p1y=0, p1z=0,
                p2x=20, p2y=20, p2z=0},
                {id="2", p1x=0, p1y=20, p1z=40,
                p2x=40, p2y=20, p2z=40}}},
        {type="video",
            id="eyes",
            src="eyes.mpg",
            descriptor="loc1",
            areas={{id="1",begin=0, finish=20},--eye moves from 1 to 2
                {id="2",begin=20.1,finish=30},--eye moves from 2 to 1
                {id="3",begin=30.1,finish=40},--eye sits in 1
                {id="4",begin=40.1,finish=50}}--eye sits in 2
        },
    linkBase={
        {

```

```

        id="link1", xconnector="starts.xml",
        {component="space1",port="1",
          role="on_x_presentation_begin"},
        {component="eyes",port="1",role="start_y"}},
    },
    {
        id="link2",xconnector="starts.xml",
        {component="space1",port="2",
          role="on_x_presentation_begin"},
        {component="eyes",port="2",role="start_y"}},
    },
    {
        id="link3",xconnector="meets-start.xml",
        {component="eyes",port="1",
          role="on_x_presentation_end"},
        {component="eyes",port="4",role="start_y"}},
    },
    {
        id="link4",xconnector="meets-start.xml",
        {component="eyes",port="2",
          role="on_x_presentation_end"},
        {component="eyes",port="3",role="start_y"}},
    },
  },
},
}

```

6.5.4 Ambiente virtual e Ubisense

Neste exemplo, construímos um arquivo NCLua para interligar o ambiente virtual e o Ubisense. No protótipo da Seção 3.4 apresentamos uma aplicação capaz de representar a localização de sensores no modelo virtual de uma das salas inteligentes do Departamento de Computação da Universidade de Illinois em Urbana-Champaign.

Para este exemplo, dividimos a sala em regiões. Quando o sensor entra em uma das regiões, a aplicação de representação tridimensional recebe uma ação para modificar a posição do sensor no modelo.


```
presentation = {
  head = {
    resources = {
      {
        id = "machine1",
        properties={resolution="1024x768",memory="1MB",
          cpu="3GHz"},
        layout = {
          id="window1", width="4096", height="3072",
          region={id="r1", width="100%", height="100%"}
        }
      },
      {
        id = "ubisense-server",
        properties={hostname="ubisense-server"},
      },
    },
    descriptorBase={
      {id="loc1", machine="machine1", region="window1",
        process-priority="HIGH"},
      {id="loc2", machine="ubisense-server"},
    }
  },
  body={
    {type="port", id="startup", component="vis3d",
      port="sensors"},
    {type="composite", id="vis3d",
      {type="activespace",
        id="sensors",
        descriptor="loc2",
        boundingbox={{id="plasma1", p1x=0, p1y=0, p1z=50,
          p2x=50, p2y=50, p2z=100},
          {id="plasma2", p1x=50, p1y=0, p1z=50,
            p2x=100,p2y=50, p2z=100},
          {id="plasma3", p1x=100,p1y=0, p1z=50,
            p2x=100,p2y=50, p2z=100},
          {id="plasma4", p1x=100,p1y=0, p1z=0,
            p2x=150,p2y=50, p2z=50 },
          {id="table", p1x=50, p1y=0, p1z=0,
```

```

                p2x=100,p2y=50, p2z=100}},
    {type="vis3d",
      id="roommodel",
      descriptor="loc1",
      src="vis\data\smartroom\smartroom.3DS",
      boundingbox={{id="plasma1", p1x=0, p1y=0, p1z=50,
                    p2x=50, p2y=50, p2z=100},
                  {id="plasma2", p1x=50, p1y=0, p1z=50,
                    p2x=100,p2y=50, p2z=100},
                  {id="plasma3", p1x=100,p1y=0, p1z=50,
                    p2x=100,p2y=50, p2z=100},
                  {id="plasma4", p1x=100,p1y=0, p1z=0,
                    p2x=150,p2y=50, p2z=50 },
                  {id="table", p1x=50, p1y=0, p1z=0,
                    p2x=100,p2y=50, p2z=100}}},
    },
  },
},
linkBase={
  {
    id="link1", xconnector="starts.xml",
    {component="sensors", port="plasma1",
      role="on_x_presentation_begin"},
    {component="roommodel", port="plasma1",
      role="start_y"},
  },
  {
    id="link2", xconnector="starts.xml",
    {component="sensors", port="plasma2",
      role="on_x_presentation_begin"},
    {component="roommodel", port="plasma2",
      role="start_y"},
  },
  {
    id="link3", xconnector="starts.xml",
    {component="sensors", port="plasma3",
      role="on_x_presentation_begin"},
    {component="roommodel", port="plasma3",
      role="start_y"},
  }
}

```

```
},
{
  id="link4", xconnector="starts.xml",
  {component="sensors", port="plasma4",
    role="on_x_presentation_begin"},
  {component="roommodel", port="plasma4",
    role="start_y"},
},
{
  id="link5", xconnector="starts.xml",
  {component="sensors", port="table",
    role="on_x_presentation_begin"},
  {component="roommodel", port="table",
    role="start_y"},
},
{
  id="link6", xconnector="finishes.xml",
  {component="sensors", port="plasma1",
    role="on_x_presentation_end"},
  {component="roommodel", port="plasma1",
    role="stop_y"},
},
{
  id="link7", xconnector="finishes.xml",
  {component="sensors", port="plasma2",
    role="on_x_presentation_end"},
  {component="roommodel", port="plasma2",
    role="stop_y"},
},
{
  id="link8", xconnector="finishes.xml",
  {component="sensors", port="plasma3",
    role="on_x_presentation_end"},
  {component="roommodel", port="plasma3",
    role="stop_y"},
},
{
  id="link9", xconnector="finishes.xml",
  {component="sensors", port="plasma4",
```

```
        role="on_x_presentation_end"},
    {component="roommodel", port="plasma4",
      role="stop_y"},
  },
  {
    id="link10", xconnector="finishes.xml",
    {component="sensors", port="table",
      role="on_x_presentation_end"},
    {component="roommodel", port="table",
      role="stop_y"},
  },
},
}
```