6
Edição e Distribuição de Apresentações Hipermídia

Para que uma apresentação possa ser atualizada durante a sua execução, uma sintaxe de transferência é proposta neste capítulo. Essa sintaxe é formada por comandos que, quando recebidos nos clientes, preservam tanto a especificação da aplicação quanto o HTG que representa o seu sincronismo.

O HTG, durante uma apresentação, também pode ter suas partes distribuídas para diferentes dispositivos, que passam a controlar, cada um, uma parte da apresentação original. Dessa forma, torna-se possível que vários usuários, de forma escalável, possam interagir através dos seus dispositivos individuais, sem interromper a apresentação coletiva. Na proposta desta tese, cada dispositivo pode redistribuir a apresentação que ele controla, definindo um verdadeiro modelo distribuído hierárquico, que será apresentado neste capítulo.

6.1.
Edição Simultânea à Apresentação

Nos casos em que os conteúdos das aplicações não são conhecidos *a priori*, não é possível ao autor especificar completamente uma aplicação antes que a sua apresentação seja iniciada. Esse é o caso, por exemplo, das aplicações contendo conteúdos produzidos “ao vivo”, como foi exemplificado no Capítulo 1.

Nesta tese, uma sintaxe de transferência é proposta, formada por comandos de edição que podem ser enviados do servidor (ambiente de autoria da aplicação) aos clientes receptores e exibidores da aplicação, com o objetivo de modificar tanto a especificação dessa aplicação quanto o fluxo da sua apresentação. De fato, os comandos atuam sobre o HTG, modificando sua estrutura em tempo de exibição (Costa, 2006). Caso as modificações realizadas ao vivo sobre as apresentações não fossem registradas simultaneamente nas especificações das aplicações, anteriormente enviadas ao sistema de apresentação, uma posterior apresentação dessas aplicações não levaria em conta tais alterações.

PUC-Rio - Certificação Digital Nº 0521497/CA

6.1.1.
Edição ao Vivo nas Aplicações NCL

Comandos para edição permitem inserir, remover ou editar as especificações de uma aplicação NCL. Em geral, esses comandos têm a seguinte sintaxe:

- add*Entity*(*baseId*, *documentId*, *xml*); e
- remove*Entity*(*baseId*, *documentId*, *entityId*).

Nesses comandos, a palavra "*Entity*" deve ser substituída pelo nome da entidade NCL (*Connector, Rule, Descriptor, Link* etc.) que o comando pretende incluir, remover ou editar. A aplicação NCL que será modificada é especificada no lugar do parâmetro "*documentId*". Além de especificar a aplicação, o comando de edição também precisa identificar a base privada onde essa aplicação está armazenada. Essa informação deve estar especificada no lugar do parâmetro "*baseId*".[21]

Caso um comando de edição seja utilizado para inserir ou editar uma entidade NCL, a especificação dessa entidade deve estar no lugar do parâmetro "*xml*". A especificação das entidades (sintaxe) deve obedecer ao XML Schema do perfil da NCL para comandos de edição (*NCL 3.0 Editing Commands Profile*) (ABNT, 2009). Um comando para adicionar uma entidade, que já existe na especificação da aplicação é utilizado, de fato, para atualizar (editar) essa entidade.

No caso dos comandos utilizados para remover entidades em uma aplicação NCL, o identificador único dessa entidade deve estar especificado no lugar do parâmetro "*entityId*".

A ordem de recepção dos comandos de edição nos clientes é importante para que a consistência de uma aplicação seja preservada. Como exemplo, caso um elo fosse adicionado a uma aplicação (*addLink*), antes da construção dos nós e das interfaces referenciadas por esse elo, um especificação inválida seria obtida. Para evitar esse problema, quando as entidades referenciadas por um comando de edição não estiverem presentes na aplicação, uma definição padrão para essas entidades deve ser construída. Posteriormente, essas entidades, construídas de forma padrão, poderão ser substituídas, através de outros comandos de edição (Costa, 2006).

[21] Como mencionado no capítulo anterior, as aplicações NCL recebidas nos clientes são preservadas em uma base privada (Moreno, 2009).

PUC-Rio - Certificação Digital Nº 0521497/CA

Também em relação à recepção dos comandos, se o autor quiser construir âncoras temporais que fazem menção ao instante no tempo em que o comando é recebido, o valor dessa âncora temporal deve fazer referência ao valor "*now*". Nos clientes, esse valor é substituído pelo instante no tempo em que o comando é recebido.

A especificação dos comandos de edição NCL pode ser encontrada no Apêndice B. Nesse apêndice encontra-se a definição completa desses comandos, incluindo o trabalho proposto por (Moreno, 2009). Nesse outro trabalho, esses comandos são estendidos para poderem identificar também os recursos (as estruturas de dados), especificados como parâmetros dos comandos, transmitidos externamente às estruturas de dados que carregam esses comandos. Nesse mesmo trabalho também é definido como o ciclo de vida das aplicações NCL pode ser controlado com base nos comandos de edição.

PUC-Rio - Certificação Digital Nº 0521497/CA

6.1.2. Edição ao Vivo no HTG

Um HTG pode ser completamente construído através dos comandos de edição. Também é possível, através desses comandos, remover ou editar uma parte de um HTG. Como mencionado no início deste capítulo, os comandos de edição são especificados em relação às entidades da NCL para que, além do HTG, a especificação de uma aplicação também permaneça atualizada.

Para inserir, remover ou editar vértices e arestas que representam um evento de apresentação em uma parte de um HTG, os seguintes comandos, descritos na Tabela 6.1, são propostos.

Comandos	Descrição
addNode(baseId, documentId, compositeId, {uri, id}+)	Adiciona um nó NCL (<media>, <context> ou <switch>) a um nó de composição (*compositeId*) em uma aplicação NCL (*documentId*) localizada uma base privada (*baseId*)
removeNode(baseId, documentId, compositeId, Id)	Remove um nó NCL (<media>, <context> ou <switch>) (*Id*) de um nó de composição (*compositeId*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

addInterface(baseId, documentId, nodeId, xml)	Adiciona uma interface (<port>, <area>, <property> ou <switchPort>)(*xml*) a um nó (<media>, <body>, <context> ou <switch>) (*nodeId*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
removeInterface(baseId, documentId, nodeId, Id)	Remove uma interface (<port>, <area>, <property> ou <switchPort>)(*Id*) de um nó (<media>, <body>, <context> ou <switch>) (*nodeId*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

Tabela 6.1 - Comandos para inserir, remover ou editar vértices e arestas que representam um evento de apresentação ou a adaptação do conteúdo.

Um comando *addNode*, quando especificado sobre um nó de mídia (<media>), pode provocar a inserção de dois vértices em um HTG, correspondentes ao início e ao fim do evento de apresentação desse nó. Caso essa mídia tenha uma duração, esses vértices devem estar unidos por uma aresta, que deve ter como condição essa duração, tão logo ela seja conhecida. Caso esse comando seja especificado sobre um nó de mídia já existente na aplicação, a condição da aresta que representa a sua duração pode ser modificada (editada), recebendo um novo valor. Também é possível que essa aresta seja removida, caso a duração da apresentação dessa mídia deixe de ser especificada. O comando *removeNode* é utilizado para remover esses vértices e a aresta entre eles.

Um comando *addInterface*, quando especificado sobre uma âncora de conteúdo (<area>) de um nó de mídia (<media>), pode provocar a inserção de dois vértices em um HTG, correspondentes ao início e ao fim do evento de apresentação dessa âncora. O vértice que representa essa transição de fim somente será construído caso exista uma duração explícita nessa âncora (especificada no parâmetro *xml*). Nesse caso, uma aresta entre esses vértices será construída, tendo como condição a duração da apresentação dessa âncora. Caso esse comando seja especificado sobre uma âncora já existente na aplicação, a condição da aresta que representa a sua duração pode ser modificada (editada) por um novo valor. Também é possível que essa aresta e o seu vértice destino sejam removidos, caso essa âncora temporal seja modificada, deixando de ter uma duração explícita. A remoção desses vértices e da aresta entre eles é realizada através do comando *removeInterface*.

PUC-Rio - Certificação Digital Nº 0521497/CA

O comando *addInterface*, quando especificado sobre uma âncora de conteúdo (<area>), também provoca no HTG a construção de uma aresta que representa o relacionamento entre o objeto de mídia e a sua âncora de conteúdo. O vértice que representa o início da apresentação desse objeto é a origem dessa aresta, que tem como destino o vértice que representa a transição de início da apresentação sobre a âncora especificada. Essa aresta é removida através do comando *removeInterface*, que remove também os vértices e arestas do HTG descritos no parágrafo anterior.

Os comandos *addNode*, *removeNode*, *addInterface* e *removeInterface* também podem ser utilizados para inserir, remover ou editar vértices e arestas de parte de um HTG que representam uma adaptação do conteúdo em uma aplicação. Nesse caso, um elemento <switch> é especificado como parâmetro dos comandos *addNode* ou *removeNode.* Através desses comandos são inseridos (*addNode*) ou removidos (*removeNode*) vértices no HTG correspondentes às transições de início e de fim da apresentação de cada componente filho do elemento <switch>. Se forem definidas interfaces específicas nesses componentes (*addInterface* e *removeInterface*), arestas representando os relacionamentos com a participação dessas interfaces são inseridas (*addInterface*) ou removidas (*removeInterface*) no HTG. A essas arestas devem estar associadas condições para a escolha do conteúdo entre as opções para adaptação. Os comandos de edição que permitem inserir, remover ou editar essas condições sobre as arestas são apresentados na Tabela 6.2.

PUC-Rio - Certificação Digital Nº 0521497/CA

Comandos	Descrição
addRule (baseId, documentId, xml)	Adiciona uma regra (<rule>)(*xml*) a uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
removeRule (baseId, documentId, Id)	Remove uma regra (<rule>)(*Id*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
addRuleBase (baseId, documentId, xml)	Adiciona uma base de regras (<ruleBase>)(*xml*) a uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

removeRuleBase (baseId, documentId, Id)	Remove uma base de regras (<ruleBase>)(*xml*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

Tabela 6.2 - Comandos para inserir, remover ou editar condições nas arestas que representam a adaptação do conteúdo ou da apresentação.

Uma condição, associada a uma aresta do HTG, pode ser inserida ou modificada por um comando para adicionar uma regra (ou uma base de regras) (*addRule, addRuleBase*). Condições também pode ser removidas (*removeRule*, *removeRuleBase*). Além da adaptação do conteúdo, regras também são necessárias para inserir, remover ou editar condições nas arestas do HTG que representam a adaptação da apresentação de um objeto. Quando a apresentação de um objeto pode ser realizada com diferentes durações (explícitas), uma aresta deve ser construída para cada uma dessas durações. É a condição associada a essas arestas que define, em tempo de apresentação, qual será a duração da apresentação desse objeto.

Para inserir, remover ou editar cada uma das arestas que tem como condição diferentes durações explícitas, os comandos *addDescriptor* (inserir ou editar) e *removeDescriptor* (remover), especificados na Tabela 6.3, são propostos.

Comandos	Descrição
addDescriptor (baseId, documentId, xml)	Adiciona um descritor (<descriptor>)(*xml*) a uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
removeDescriptor (baseId, documentId, Id)	Remove um descritor (<descriptor>)(*Id*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
addDescriptorSwitch (baseId, documentId, xml)	Adiciona um *switch* de descritores (<descriptorSwitch>)(*xml*) a uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
removeDescriptorSwitch (baseId, documentId, Id)	Remove um *switch* de descritores (<descriptorSwitch>)(*Id*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
addDescriptorBase(baseId, documentId, xml)	Adiciona uma base de descritores (<descriptorBase>)(*xml*) a uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

PUC-Rio - Certificação Digital Nº 0521497/CA

removeDescriptorBase(baseId, documentId, Id)	Remove uma base de descritores (<descriptorBase>)(*Id*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

Tabela 6.3 - Comandos para inserir, remover ou editar arestas que representam a adaptação da apresentação.

Descritores devem estar especificados em uma base (*addDescriptorBase*, *removeDescriptorBase*). Quando diferentes alternativas para apresentação são oferecidas (adaptação da apresentação), os descritores de cada uma dessas alternativas devem estar agrupados em um *switch* de descritores (*addDescriptorSwitch*, *removeDescriptorSwitch*).

As arestas do HTG que representam relacionamentos causais entre as transições de eventos (vértices) podem ser inseridas, removidas ou modificadas, através dos comandos da Tabela 6.4. Quando a ação de um relacionamento corresponde a uma transição de um evento de atribuição, um vértice, destino da aresta que representa esse relacionamento, também pode ser inserido, removido ou modificado, através dos comandos da Tabela 6.4.

PUC-Rio - Certificação Digital Nº 0521497/CA

Comandos	Descrição
addConnector (baseId, documentId, xml)	Adiciona um conector (<connector>)(*xml*) a base de conectores de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
removeConnector (baseId, documentId, Id)	Remove um conector (<connector>)(*Id*) da base de conectores de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
addConnectorBase (baseId, documentId, xml)	Adiciona uma base de conectores (<connectorBase>)(*xml*) a uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
removeConnectorBase (baseId, documentId, Id)	Remove uma base de conectores (<connectorBase>)(*Id*) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)
addLink(baseId, documentId, compositeId, xml)	Adiciona um elo (<link>)(*xml*) a um nó de composição NCL (<body>, <context> ou <switch>) em uma aplicação NCL (*documentId*) localizada em uma base privada (*baseId*)

removeLink(baseId, documentId, compositeId, Id)	Remove um elo (<link>)(*Id*) de um nó de composição NCL (<body>, <context> ou <switch>) de uma aplicação NCL (*documentId*) em uma base privada (*baseId*)

Tabela 6.4 - Comandos para inserir, remover ou editar vértices e arestas que representam aos relacionamentos de uma apresentação.

Os comandos para adicionar ou modificar um conector (*addConnector* e *addConnectorBase*) e para adicionar ou modificar um elo (*addLink*) atuam em conjunto sobre o HTG. Quando esses comandos definem um relacionamento que tem uma condição e uma ação simples, uma única aresta é construída no HTG, unindo os vértices correspondentes à condição e à ação desse relacionamento. Caso o relacionamento especificado nos comandos de edição defina ações compostas, várias arestas são construídas no HTG, cada qual tendo como destino o vértice correspondente a cada ação desse relacionamento. Caso esse relacionamento defina condições compostas, condições do HTG serão associadas a cada aresta construída para representar esse relacionamento. A remoção dos vértices e arestas do HTG, que representam os relacionamentos mencionados, é ser realizada através dos comandos *removeConnector*, *removeConnectorBase* e, principalmente, *removeLink*.

6.2. Apresentação Distribuída através de Múltiplos Dispositivos

O uso de múltiplos dispositivos pode agregar várias vantagens à apresentação das aplicações hipermídia (Costa, 2009; Soares, 2009b). Em cenários com assistência coletiva, por exemplo, dispositivos individuais podem ser utilizados para evitar que uma apresentação seja interrompida em função das interações realizadas individualmente pelos usuários.

Além de evitar que a assistência coletiva seja prejudicada, o uso de múltiplos dispositivos pode permitir diversas navegações individuais. Nos STVDI, por exemplo, vários usuários telespectadores poderiam interagir simultaneamente, o que seria impossível utilizando apenas a TV e o controle remoto tradicional. Dependendo da aplicação, cada telespectador poderia escolher uma opção diferente, e visualizar, individualmente, o resultado da sua escolha.

PUC-Rio - Certificação Digital Nº 0521497/CA

As escolhas realizadas pelos usuários, através dos seus dispositivos, podem resultar em diferentes apresentações. Caso o controle dessas apresentações seja realizado de forma centralizada, o dispositivo que contém a especificação da aplicação, chamado, a partir deste ponto, de dispositivo base, precisa controlar cada interação realizada através de cada dispositivo individual. Nessa forma de controle, também pode ser necessário que o dispositivo base conheça as características de cada dispositivo, para que as adaptações especificadas nas aplicações possam ser realizadas. Assim, o controle centralizado das apresentações distribuídas pode exigir recursos (armazenamento, processamento, comunicação etc.) no dispositivo base que, em muitos cenários, incluindo o cenário formado por receptores em um STVDI, podem não estar disponíveis.

Caso os dispositivos individuais tenham a capacidade de executar uma apresentação, uma solução mais interessante, proposta nesta tese, consiste em distribuir, para cada dispositivo, o controle da sua própria apresentação (parte da apresentação "global"). Nesse caso, cabe ao dispositivo base controlar a distribuição da apresentação aos dispositivos individuais, como será discutido no exemplo a seguir.

A aplicação NCL especificada na Figura 4.25 (Capítulo 4) poderia ter sua apresentação realizada de forma distribuída. Nessa aplicação, uma animação ("*animation*") e uma música ("*music*") são executadas em paralelo, desde o ínicio da sua apresentação. Também nessa aplicação, após 45 segundos do início da apresentação da animação, a imagem de uma chuteira ("*icon*") pode ser apresentada, mas apenas se ações interativas forem permitidas. Caso o usuário selecione essa imagem, uma propaganda ("*shoes*") e um formulário ("*form*") são apresentados. Para que essas outras mídias possam ser apresentadas na tela, único dispositivo de apresentação no exemplo original, a animação precisa ser redimensionada, como resultado da seleção da imagem da chuteira.

Caso cada usuário tivesse seu próprio dispositivo, a apresentação descrita no parágrafo anterior poderia ser executada sem que a assistência coletiva fosse prejudicada, como acontece, por exemplo, no redimensionamento do filme da animação. Nesse caso, o filme da propaganda e o formulário para aquisição do produto poderiam ser apresentados somente nos dispositivos dos usuários que realizaram a ação interativa. A Figura 6.1 ilustra esse cenário, onde a apresentação da aplicação NCL, escolhida como exemplo, é realizada de forma distribuída.

PUC-Rio - Certificação Digital Nº 0521497/CA

Figura 6.1 - Múltiplos dispositivos para exibição.

No cenário proposto na Figura 6.1, a imagem da chuteira é apresentada em todos os dispositivos individuais. Se essa imagem for selecionada em um dispositivo específico, terá início à apresentação, nesse mesmo dispositivo, do vídeo relativo à propaganda da chuteira e do formulário. Na Figura 6.1, dois usuários realizaram essa seleção. Através dos seus dispositivos, esses usuários podem fazer escolhas individuais em paralelo, o que seria impossível com apenas um único dispositivo. Nesse exemplo, um dos usuários já finalizou sua compra, enquanto o outro ainda está avaliando o produto.

O controle da apresentação ilustrada na Figura 6.1 pode ser realizado tanto de forma centralizada, no dispositivo base, quanto de forma distribuída, em cada dispositivo individual. No controle centralizado, no entanto, além dos requisitos já mencionados nesta seção, para a apresentação do exemplo ilustrado na Figura 6.1, o dispositivo base precisaria controlar todas as informações individuais dos usuários necessárias para a compra de cada produto (modelo, tamanho, forma de pagamento etc.).

Independente da forma como a apresentação é controlada (centralizada ou distribuída), para que ela possa ser realizada de forma distribuída, na especificação da aplicação devem ser definidos quais objetos de mídia deverão ser apresentados nos dispositivos individuais. Essa especificação pode ser realizada pelo autor ou ser definida de forma automática, como proposto em (Cesar, 2007). Em ambos os casos, é desejável que essa especificação possa ser realizada de forma adaptável. Dessa forma, uma apresentação pode ser realizada de forma distribuída apenas se dispositivos individuais estiverem disponíveis no ambiente de apresentação (Costa, 2009; Soares, 2009b).

PUC-Rio - Certificação Digital Nº 0521497/CA

Distribuir para cada dispositivo, o controle da sua própria apresentação, requer a fragmentação da especificação de uma aplicação durante à sua apresentação. Nesta tese, o HTG é proposto como a estrutura de dados que oferece suporte a essa fragmentação, que é realizada selecionando, no HTG da aplicação, os vértices que representam transições de eventos sobre objetos que devem ser apresentados nos dispositivos individuais. As arestas com origem e destino sobre os vértices selecionados também devem ser selecionadas para a construção desse novo HTG, parte do "HTG original".

Voltando ao exemplo ilustrado na Figura 6.1, considerando que nessa aplicação foi especificado que a imagem da chuteira ("*icon*"), o filme da propaganda ("*shoes*") e o formulário ("*enForm*" e "*ptForm*") deverão ser apresentados através dos dispositivos individuais disponíveis, o HTG, apresentado na Figura 6.2, é construído para o controle dessa apresentação nos dispositivos. Esse HTG foi obtido com base no "HTG original" dessa aplicação, apresentado no Capítulo 4 (Figura 4.26).

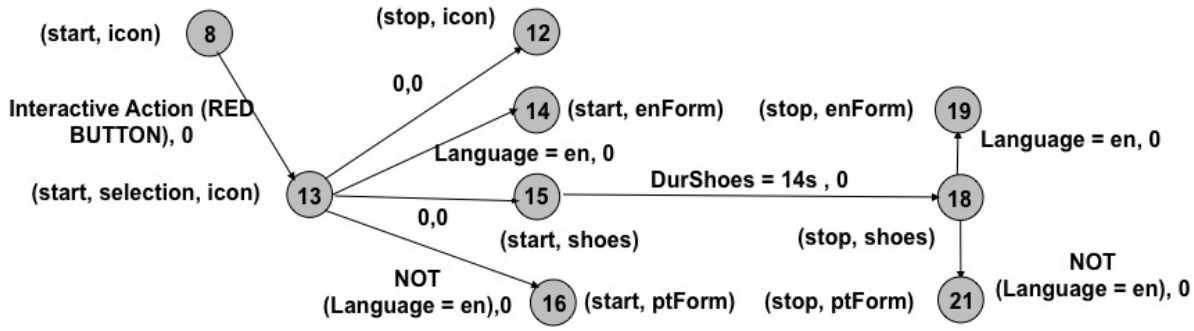

Figura 6.2 - HTG para controle da apresentação nos dispositivos individuais.

No HTG da Figura 6.2 são representadas todas as transições de eventos e os seus relacionamentos necessários à execução da apresentação nos dispositivos individuais da Figura 6.1. Na Figura 6.2, o ponto de início do HTG é o vértice 8 (início da apresentação da imagem da chuteira). Seguindo as especificações desse HTG, a imagem da chuteira é apresentada e pode ser selecionada. Essa seleção, por sua vez, finaliza a apresentação dessa imagem e inicia a apresentação do formulário e da propaganda. Em cada dispositivo, um formulário, em um idioma diferente (português ou inglês), pode ser apresentado, dependendo das características do usuário.

Durante a apresentação, o dispositivo base deve enviar, aos dispositivos individuais, todos os conteúdos das mídias que podem ser apresentados nesses

PUC-Rio - Certificação Digital Nº 0521497/CA

dispositivos. Também devem ser enviados o HTG para o controle da apresentação (Figura 6.2) e as especificações espaciais necessárias à apresentação dos conteúdos (Costa, 2009).

Caso existam relacionamentos entre os objetos apresentados nos dispositivos individuais, e os demais objetos, apresentados no dispositivo base, eventos devem ser gerados para essa comunicação (Costa, 2009). No exemplo da Figura 6.1, um evento para iniciar a apresentação da imagem da chuteira (vértice 8) deve ser enviado pelo dispositivo base aos dispositivos individuais. O mesmo acontece para terminar a apresentação dessa imagem (vértice 12).[22] Esses relacionamentos podem ser identificados no HTG da aplicação, através das arestas com origem e destino em vértices sobre objetos apresentados em dispositivos de diferentes classes, que são definidas a seguir.

Um classe de dispositivos define um conjunto de dispositivos que sempre executam uma mesma parte de uma apresentação. Esse é o caso, por exemplo, dos dispositivos individuais da Figura 6.1. Classes onde os dispositivos controlam suas próprias apresentações são definidas como “ativas”.[23] Diversos dispositivos podem fazer parte de uma mesma classe e inúmeras classes podem ser definidas, dependendo da distribuição desejada para uma apresentação.

Dispositivos em uma mesma classe “ativa” visualizam, normalmente, a mesma apresentação apenas no seu início. Caso transições de eventos interativos e adaptações estejam presentes em uma apresentação, esses pontos de imprevisibilidade poderão levar a apresentações completamente diferentes em cada dispositivo. Esse é o caso, por exemplo, das apresentações realizadas nos dispositivos individuais da Figura 6.1, onde um usuário já finalizou a sua compra, enquanto o outro, através do seu dispositivo, ainda está analisando o produto.

Em alguns casos, no entanto, pode ser desejável sincronizar os dispositivos que fazem parte de uma mesma classe, para que eles estejam visualizando a mesma apresentação em um determinado instante da duração dessa apresentação. Esse caso pode acontecer quando, por exemplo, usuários compartilham o mesmo dispositivo e, a partir de um determinado instante, cada um desses usuários deseja

[22] Nesse exemplo, a apresentação da imagem da chuteira pode ser interrompida caso o usuário realize uma ação interativa. Assim, um evento de termino da apresentação da imagem da chuteira recebido posteriormente seria desprezado, considerando os estados possíveis para um evento (Figura 3.1).

[23] Esse termo é proposto para diferenciar os dispositivos que apenas reproduzem informações (“passivos”), sem controlar, de fato, a apresentação.

PUC-Rio - Certificação Digital Nº 0521497/CA

usar o seu próprio dispositivo. Nesse caso, um dispositivo, ao se registrar em uma classe, pode receber o plano de apresentação construído em um outro dispositivo, dessa mesma classe. Dessa forma, ambos os dispositivos terão suas apresentações posicionadas no mesmo instante no tempo da apresentação. A partir desse ponto, os usuários desses dispositivos podem interagir de forma independente, podendo obter, como resultado, diferentes apresentações individuais.

Cada dispositivo em uma classe "ativa" pode redistribuir a apresentação que ele controla. Na aplicação ilustrada na Figura 6.1, por exemplo, um grupo de usuários poderia desejar usar um outro dispositivo, com uma tela maior, para visualizar o vídeo da propaganda. Nesse exemplo, ilustrado na Figura 6.3, o preenchimento do formulário para a compra do produto continuaria sendo realizado através dos dispositivos individuais.

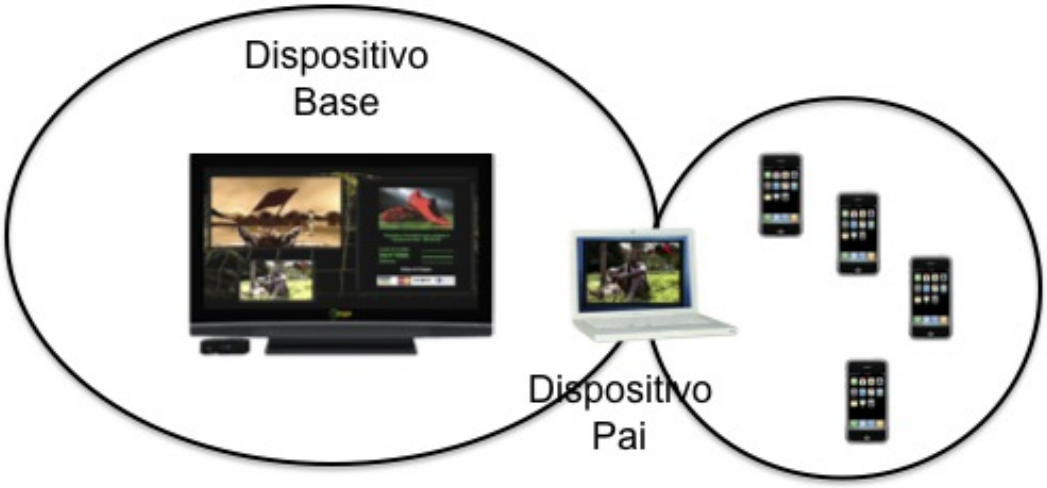

Figura 6.3 - Distribuição de dispositivos em domínios.

Na Figura 6.3, dois domínios são representados. Cada domínio define uma organização hierárquica, onde um dispositivo controla a distribuição da apresentação para os demais integrantes desse domínio. O dispositivo que realiza esse controle é chamado de dispositivo pai. O dispositivo base é um dispositivo pai que controla a distribuição da apresentação completa.

No exemplo da Figura 6.3, parte da apresentação é distribuída para ser executada em um computador. Essa parte, é a mesma apresentada nos dispositivos individuais da Figura 6.1. Porém, nesse exemplo, o formulário será apresentado em outros dispositivos. No computador será apresentada a imagem da chuteira. Caso essa imagem seja selecionada, o vídeo da propaganda será apresentado nesse mesmo dispositivo. O formulário para aquisição desse produto, porém, será

PUC-Rio - Certificação Digital Nº 0521497/CA

redistribuído, a partir do computador, para ser apresentado em outros dispositivos individuais.

O computador faz parte de um domínio, onde a TV é o dispositivo base. O computador também faz parte de um outro domínio, onde ele é o dispositivo pai. Nesse outro domínio, é o computador que faz a distribuição da apresentação aos demais dispositivos. Assim, nesse exemplo, pode ser observada a hierarquia para a distribuição das apresentações. Essa distribuição é iniciada na TV (dispositivo base) que passa para o computador (dispositivo pai em um domínio) parte da apresentação que, por fim, é entregue aos dispositivos individuais.

PUC-Rio - Certificação Digital Nº 0521497/CA