

Recuperação e Tolerância a Falhas em Apresentações de Aplicações Hiperfídia Distribuídas

Carlos Eduardo C. F. Batista
Departamento de Informática
PUC-Rio
Rio de Janeiro, RJ, Brasil
cbatista@inf.puc-rio.br

Marcio Ferreira Moreno
Departamento de Informática
PUC-Rio
Rio de Janeiro, RJ, Brasil
mfmoreno@inf.puc-rio.br

Luiz Fernando Gomes Soares
Departamento de Informática
PUC-Rio
Rio de Janeiro, RJ, Brasil
lfgs@inf.puc-rio.br

ABSTRACT

The NCL language supports the creation of hypermedia presentations using resources from multiple connected devices simultaneously, through a high-level declarative description. This paper proposes protection mechanisms to address fundamental characteristics of the distributed presentation environment of the Ginga-NCL subsystem, which is seen as a failure-prone distributed system using asynchronous communication to orchestrate devices with heterogeneous hardware and software specifications. As proof of concept the discussed mechanisms were incorporated to the reference implementation of Ginga in order to make it capable of recovering from and adjusting to faults or configuration changes during the presentation of distributed hypermedia applications.

RESUMO

A linguagem NCL oferece suporte à especificação de apresentações hiperfídia que podem utilizar simultaneamente recursos fornecidos por múltiplos dispositivos conectados, através de uma descrição declarativa de alto-nível. O presente trabalho propõe mecanismos de proteção relacionados às características fundamentais do ambiente de execução distribuída do subsistema Ginga-NCL, que é tratado como um sistema distribuído propenso a falhas, levando em consideração a comunicação assíncrona entre dispositivos com especificações de hardware e software heterogêneas. Como prova de conceito, os mecanismos discutidos foram incorporados à implementação de referência do middleware Ginga, viabilizando a recuperação e ajuste a falhas ou mudanças de configuração durante a apresentação de aplicações hiperfídia distribuídas.

Categories and Subject Descriptors

I.7.2 [Document Preparation]: Languages and systems, Hypertext/hypermedia, Markup languages, Standards.

General Terms

Management, Design, Languages.

Keywords

Fault-tolerant Systems, Ginga-NCL, Distributed Hypermedia Applications.

1. INTRODUÇÃO

O desenvolvimento de aplicações hiperfídia distribuídas possui alta complexidade, uma vez que o sincronismo entre os dispositivos envolvidos pode ser dificultado por características inerentes à comunicação em rede. Para garantir a viabilidade e eficiência da execução de tais aplicações, os ambientes que as executam devem oferecer mecanismos que visem reduzir o esforço necessário para a autoria de aplicações multimídia distribuídas robustas.

Muitas plataformas se propõem a oferecer suporte ao desenvolvimento e a execução de aplicações hiperfídia que podem utilizar recursos (tela, alto falantes, armazenamento etc.) de dispositivos conectados em rede.

O segmento das plataformas direcionadas a interligar dispositivos em redes domésticas (*Home-Area Networks – HAN*) se expandiu sensivelmente na última década, com a popularização de diversos dispositivos capazes de se comunicar em rede e de reproduzir artefatos multimídia. UPnP [1] e OSGi [2] são alguns exemplos de plataformas com essas características.

Entre os dispositivos presentes em uma rede doméstica comumente tem-se um receptor de TV Digital. Com a possibilidade de executar aplicações interativas concomitantemente ao conteúdo audiovisual de TV, é natural que seja possível para essas aplicações utilizar os recursos disponibilizados por outros dispositivos conectados à rede local do receptor. Diferentes sistemas já suportam o desenvolvimento de aplicações que podem utilizar os recursos de dispositivos conectados que fazem parte do contexto de uma aplicação interativa [3][4].

Para esse tipo de desenvolvimento, a linguagem NCL (*Nested Context Language*) [5] define dois tipos de classes de dispositivos. A classe de dispositivos que deve apresentar o mesmo conteúdo sobre controle único é chamada de classe passiva. Por outro lado, a classe de dispositivos que apresenta o mesmo conteúdo inicial, mas que oferece controle individual e independente a partir de cada dispositivo é chamada de classe ativa.

O suporte a apresentação de aplicações especificadas na linguagem NCL é realizado pelo ambiente declarativo do middleware Ginga, denominado Ginga-NCL. NCL e Ginga-NCL fazem parte do padrão do sistema ISDB-T (*International Standard for Digital Broadcasting*) [6] e da recomendação ITU-T para serviços IPTV [7].

A implementação de referência do Ginga-NCL oferece a possibilidade de se definir classes de dispositivos através da descrição de suas características de hardware, software e de rede [8], contemplando também mecanismos para registro de dispositivos nas classes definidas. O modelo de descrição de

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Full Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9642.
SBC - Brazilian Computer Society

classes é extensível e genérico, uma abordagem que, por exemplo, permite que aplicações NCL possam explorar recursos oferecidos por dispositivos compatíveis com outras tecnologias que visem interligar dispositivos em redes domésticas (como UPnP [1] ou OSGi [2]).

A eficiência de uma apresentação multimídia distribuída depende de uma combinação de fatores relacionados às características de hardware, software e de rede dos dispositivos envolvidos em sua especificação. É importante que os aspectos que visem a eficiência da realização da apresentação sejam tratados de forma automática ou semiautomática pelo ambiente de execução.

O principal objetivo deste trabalho consiste em desenvolver e incorporar mecanismos de recuperação e tolerância a falha na implementação de referência do Ginga-NCL, que serão utilizados quando da apresentação de aplicações NCL para múltiplos dispositivos. Tais mecanismos estão associados aos parâmetros de descrição das classes de dispositivos suportadas pela implementação de referência, como detalhado na Seção 5, de forma que o autor da aplicação lide apenas com abstrações de alto-nível relacionadas às características de uso dos recursos e serviços distribuídos associados à aplicação NCL.

O trabalho é organizado como descrito a seguir: a segunda seção apresenta informações sobre o suporte às aplicações multi-dispositivos do Ginga-NCL; a terceira seção descreve a modelagem das entidades funcionais da arquitetura do Ginga-NCL que incorporam mecanismos de recuperação e tolerância a falhas na execução de aplicações NCL; a quarta seção apresenta uma análise sobre as falhas na execução de aplicações NCL distribuídas que serão tratadas; a implementação das entidades funcionais que visam dar suporte ao tratamento e recuperação das falhas apresentadas na seção anterior é apresentada na quinta seção, implementação que é testada e validada; o trabalho conclui na sexta seção, com perspectivas futuras e algumas considerações quanto à relevância do trabalho proposto.

2. APLICAÇÕES MULTI-DISPOSITIVO NO GINGA-NCL

Aplicações sendo executadas no Ginga-NCL podem conter objetos de mídia que devem ser exibidos em dispositivos conectados, registrados em classes de dispositivos, que agregam dispositivos conectados e registrados ao dispositivo base. A associação de um objeto de mídia a uma classe de dispositivos define que, apesar de orquestrado pelo dispositivo base, o conteúdo de tal objeto será reproduzido em todos os dispositivos registrados àquela classe em particular [9].

Para que um dispositivo secundário faça parte da execução de uma aplicação distribuída, ele deverá se registrar a uma classe definida por um dispositivo base, que é aquele que contém o código completo e as referências para todos os objetos de mídia da aplicação NCL a ser executada de maneira distribuída. Esse registro de dispositivos secundários em dispositivos base poderá ser feito de maneira recursiva (dispositivos secundários atuando como base para outros), criando domínios e subdomínios da aplicação. O dispositivo que controla a classe em que um dispositivo secundário (filho) está registrado é chamado de dispositivo pai. O modelo de controle hierárquico define que uma classe não pode estar sobre o controle de mais de um dispositivo pai por vez. A hierarquia dentro do domínio de aplicações é mantida e um dispositivo não pode ser pai ou filho de si mesmo.

Para a implementação de referência do Ginga-NCL foi especificado um mecanismo genérico de registro de dispositivos

nas classes disponíveis em um determinado domínio de aplicação. Tal mecanismo é composto por três passos, descritos a seguir:

1. Um dispositivo secundário torna-se visível para o dispositivo base, não importa se por pareamento, descoberta de serviços ou simplesmente por estar acessível através de uma rede comum. O dispositivo base é responsável por orquestrar a apresentação com recursos associados às classes de dispositivos.

2. O dispositivo base verifica se o dispositivo secundário é compatível com alguma das definições de classes de dispositivos associadas à apresentação corrente, e em caso de múltiplas possibilidades positivas, o usuário do dispositivo secundário deve ser requisitado a escolher qual classe se registrar.

3. O dispositivo base então configura todos os serviços de rede necessários para a comunicação com o dispositivo secundário, de forma que os recursos fornecidos pelo dispositivo secundário possam ser orquestrados pelo dispositivo base durante a execução da apresentação.

NCL suporta a definição de dois tipos de classes de dispositivos: as classes passivas, nas quais o mesmo conteúdo é exibido em todos os dispositivos registrados ao mesmo tempo (*multicast*), com controle de navegação compartilhado; e as classes ativas, onde a exibição da aplicação é controlada individualmente (*unicast*). As classes são associadas a objetos de mídia, sinalizando para o orquestrador que o exibidor associado é remoto, e através da identificação das classes o orquestrador deve ser capaz de determinar o mecanismo de comunicação com o dispositivo (passo 3 na sequência apresentada anteriormente).

As classes de dispositivo do NCL devem servir de abstração para os recursos que os dispositivos que as compõem oferecem. Portanto, é interessante facilitar o acoplamento de adaptadores que permitam a integração de diferentes dispositivos (e seus serviços de rede associados) à plataforma de execução de aplicações Ginga-NCL distribuídas, de tal forma que uma aplicação NCL possa explorar quaisquer recursos multimídia que os dispositivos disponham. O acoplamento de novas classes de dispositivos deve ser feito com base nas características dos mesmos (hardware, software, rede), o que requer um mecanismo para descrição de tais classes.

O mecanismo de descrição de classes proposto para a implementação de referência do Ginga-NCL [8] é baseado em algumas premissas, que visam acomodar as diferentes características associadas aos dispositivos multimídia conectáveis em rede, e da comunicação com os serviços que permitem a exploração de seus recursos multimídia.

Assim, deve ser possível também integrar, em uma aplicação Ginga-NCL distribuída, dispositivos que são compatíveis com outras plataformas de integração de dispositivos multimídia, como as apresentadas na seção anterior. As premissas consideradas para a descrição de classes de dispositivos são listadas abaixo:

- Possibilidade de estabelecer um limite (máximo e mínimo) de dispositivos para uma classe, associada a um mecanismo de identificação único para cada dispositivo.
- Possibilidade de descrição das capacidades específicas que os dispositivos filhos devem possuir para poderem ser associados pelo dispositivo pai a uma classe. Essas capacidades podem utilizar semântica de descrição de uma plataforma arbitrária, e a requisição por tais informações deve ser suportada pela plataforma (através de adaptadores).

Limitar a quantidade de dispositivos registrados é importante para aplicações que só façam sentido com um limite máximo ou mínimo de usuários. Ações relacionadas a um objeto de mídia de uma classe com tal limitação só serão notificadas pelo dispositivo pai aos dispositivos filhos quando a classe contiver uma quantidade de dispositivos compatíveis com sua definição (por exemplo, usuários de uma classe recebem a sinalização para inicialização de um objeto de mídia só se o número mínimo de dispositivos conectados é alcançado).

A identificação única de dispositivos poderá ser utilizada na definição de elos (elemento `<link>`) contendo objetos de mídia de uma classe registrada, através do elemento `<bindParam>` com o atributo `name` contendo `"deviceId"`, e o atributo `value` a variável de identificação do dispositivo. Usando identificação única é possível limitar a notificação de eventos apenas para um dispositivo secundário filho dentre os registrados na classe. O identificador único é acessível de acordo com sua ordem de registro (exemplo: a variável `ClasseAtivaNCL.device(1)` contém o identificador único para o primeiro dispositivo da classe `ClasseAtivaNCL`).

A descrição das capacidades de dispositivos em uma classe é recuperada pelo dispositivo base antes da distribuição da exibição de uma aplicação, através de chamadas específicas para os serviços do dispositivo filho. Note que a descrição das capacidades de uma classe deve ser comparada com a descrição das capacidades (serviços oferecidos) do dispositivo quando do registro em uma classe. O uso de mesma semântica e sintaxe para essas especificações simplifica muito a comparação.

Para a implementação de referência do Ginga-NCL, uma extensão da especificação CC/PP (na sua implementação UAProf [10]) foi definida para dar suporte à descrição das classes de dispositivos NCL. A extensão contempla o acréscimo de alguns elementos específicos do modelo de múltiplos dispositivos proposto para o Ginga-NCL, de forma a incorporar outras semânticas de descrição de capacidades (como a descrição UPnP [1]).

A definição das classes de dispositivos NCL foi implementada utilizando os elementos `HardwarePlatform`, `SoftwarePlatform` e `NetworkCharacteristics` do UAProf (que são subclasses do componente genérico CC/PP [11]), e uma nova classe `DeviceGroup`. Um novo namespace é usado para a definição da classe, bem como os novos parâmetros específicos do Ginga-NCL e atributos genéricos para comportar definições de capacidades com semântica associada a algum outro modelo de descrição.

O módulo que dá suporte à comunicação com múltiplos dispositivos da implementação de referência do Ginga-NCL é apresentado na Figura 1, através de um diagrama de segmentação em camadas que compreende suas principais entidades funcionais e a relação com o formatador NCL. Os componentes do módulo de integração de dispositivos devem ser responsáveis por realizar as etapas necessárias para descoberta, registro e orquestração dos serviços oferecidos pelos dispositivos secundários, de forma que tais serviços possam ser utilizados por uma aplicação NCL. A seguir, os componentes do módulo implementado são descritos:

- Gerente de Integração de Dispositivos – componente responsável pela comunicação com o formatador NCL. Traduz as notificações do formatador para mensagens de controle dos serviços oferecidos pelos dispositivos secundários e as notificações advindas dos dispositivos secundários para transições das máquinas de estado

associadas aos objetos de mídia em execução nos dispositivos secundários.

- Adaptador de Processador Remoto – componente que oferece uma abstração através da qual os serviços dos dispositivos filhos conectados ao dispositivo pai podem ser utilizados, para a execução de objetos de mídia de uma aplicação NCL. Para tal, deve-se estabelecer uma conexão (através do Módulo de Comunicação) com o dispositivo filho e também alocar recursos do dispositivo pai (através do Proxy de Recurso Local) que são utilizados durante a orquestração do dispositivo filho.
- Módulo de Registro – módulo responsável por oferecer mecanismos de registro de dispositivos secundários, de acordo com suas capacidades. Tal módulo contém adaptadores que podem interpretar semânticas de descrição de plataformas de integração de dispositivos multimídia arbitrárias.
- Gerente de Classes de Dispositivos – módulo responsável por gerenciar as classes de dispositivos, que podem ser incorporadas à plataforma a partir de uma descrição RDF associada a uma aplicação NCL. O módulo realiza o registro de dispositivos filhos em classes suportadas pelo dispositivo pai.

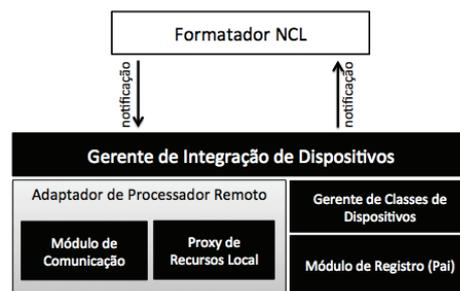


Figura 1. Arquitetura de software do módulo de integração de dispositivos

A seguir será apresentado o plano de recuperação do Ginga-NCL, que a princípio considera apenas seus componentes de software locais, mas que, como resultado do presente trabalho, contemplará também recuperação de falhas ocorridas durante a execução de aplicações NCL distribuídas.

3. PLANO DE RECUPERAÇÃO DO GINGA-NCL

O plano de recuperação do Ginga-NCL tem por objetivo tornar resiliente não apenas o subsistema em si, mas também a apresentação de aplicações interativas NCL que o subsistema executa. Com o objetivo de definir conjuntos destinados à aplicação de técnicas de recuperação, os módulos da arquitetura do Ginga-NCL [5] foram classificados da seguinte maneira: módulos de Risco, módulos de Apresentação, módulos de Controle e módulos de Recuperação [12].

Os módulos de Risco são aqueles que podem comprometer a confiabilidade da execução do Ginga-NCL, por agregarem bibliotecas de terceiros ou serem atualizados no processo de evolução dinâmica de um middleware de TV Digital [5] (exemplo: módulos dos exibidores, *parser* XML, máquina Lua, sintonizador etc.).

Os módulos de Recuperação do Ginga-NCL são aqueles responsáveis pela detecção, controle e recuperação de falhas de software. O módulo Gerente de Recuperação é o único módulo desse conjunto, de fato um super-módulo que, internamente, divide-se em outros módulos [12].

A Figura 2 [12] ilustra como o conjunto dos módulos de Recuperação atuam sobre os outros conjuntos de módulos para criar o plano de recuperação a falhas. A arquitetura apresentada na Figura 2 é dividida de acordo com as técnicas de recuperação: Proativa e Reativa.

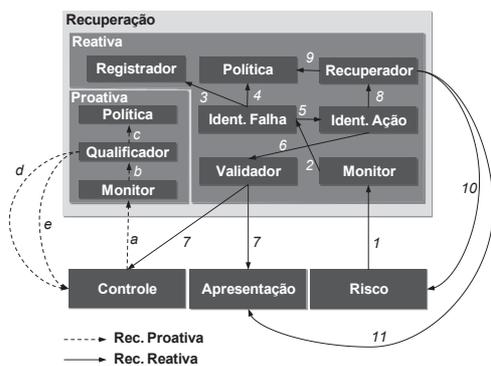


Figura 2. Arquitetura Plano de Recuperação Ginga-NCL.

Na recuperação proativa as técnicas de recuperação são aplicadas pelo componente Qualificador através de rejuvenescimento dos módulos de Controle. Quando um módulo de Controle é rejuvenescido, os módulos de Apresentação e os módulos de Risco, controlados por esse módulo de Controle, são, indiretamente, rejuvenescidos.

Para saber quando realizar o rejuvenescimento de cada módulo de Controle, sem comprometer a disponibilidade do sistema, foi definido um componente Monitor, que deve ser cadastrado como ouvinte do comportamento dos módulos de Controle. Quando alguma operação que possibilita rejuvenescimento é permitida nesses módulos, o Monitor é notificado (na Figura 2, seta “a”). O Monitor então repassa a informação para que o Qualificador avalie se há alguma ação de recuperação a ser realizada (na Figura 2, seta “b”).

Na avaliação realizada pelo Qualificador, pode ser levada em consideração uma política de recuperação proativa especificada para o dispositivo receptor, através de uma consulta ao componente Política (na Figura 2, seta “c”). Por exemplo, um dispositivo receptor capaz de apresentar apenas uma aplicação interativa por vez deve possuir como política a existência de apenas um Formatador instanciado.

Antes de executar a ação de recuperação (na Figura 2, seta “e”), o Qualificador consulta o estado dos módulos de Controle para certificar-se que o rejuvenescimento pode ser realizado (na Figura 2, seta “d”).

Na recuperação reativa, um Monitor de falhas foi definido para receber notificações de falhas ocorridas nos módulos de Risco (na Figura 2, seta “1”). Para isso, o Monitor é cadastrado como ouvinte de falhas dos módulos de Risco. Ao receber uma notificação, o Monitor repassa a informação de falha ao Identificador de Falha (na Figura 2, seta “2”).

De posse das informações sobre a falha, o Identificador de Falha solicita ao Registrador que a falha seja registrada no sistema (na Figura 2, seta “3”). O Registrador de falhas retorna ao Identificador de Falha quantas vezes a falha ocorreu em um

intervalo de tempo. Com essa informação, o Identificador de Falha consulta a Política de recuperação reativa para avaliar se a falha deve ser tratada (na Figura 2, seta “4”). Por exemplo, um dispositivo receptor pode definir como política que uma determinada falha não seja mais tratada após um determinado número de tentativas de recuperação. Caso o Identificador de Falha defina que a falha não seja tratada, nenhuma ação de recuperação será realizada. Caso contrário, o Identificador de Falha repassa as informações sobre a falha para o Identificador de Ação (na Figura 2, seta “5”).

Ao receber as informações sobre a falha ocorrida, o Identificador de Ação avalia qual a ação de recuperação a ser realizada. O Identificador de Ação então solicita ao Validador que o estado do módulo a ser recuperado seja validado (na Figura 2, seta “6”). Para isso, o Validador consulta os módulos de Controle e de Apresentação, retornando as informações para o Identificador de Ação (na Figura 2, seta “7”). O estado retornado pelo Validador pode indicar que, apesar da falha ocorrida, o módulo que consiste na origem da falha realizou suas operações corretamente. Nesse caso, nenhuma ação de recuperação será realizada. Caso contrário, após definir a ação de recuperação e adquirir as informações necessárias para realizá-la, o Identificador de Ação aciona o Recuperador (na Figura 2, seta “8”).

O Recuperador é responsável por garantir que a ação de recuperação recebida através do Identificador de Ação seja realizada. Antes de executar a ação de recuperação, no entanto, o Recuperador consulta a Política de recuperação reativa para determinar os requisitos da ação (na Figura 2, seta “9”). Um exemplo de requisito é o intervalo de tempo em que a ação de recuperação é válida. De posse de todas as informações necessárias, o Recuperador realiza a ação de recuperação sobre o módulo de Risco origem da falha (na Figura 2, seta “10”) e, em seguida, notifica o módulo de Apresentação, que controla esse módulo de Risco, que uma recuperação foi realizada (na Figura 2, seta “11”).

4. FALHAS EM APLICAÇÕES NCL MULTI-DISPOSITIVO

Uma plataforma para execução de aplicações hipermídia multi-dispositivo é um sistema distribuído propenso a um conjunto significativo de falhas, muitas das quais são passíveis de tratamento e recuperação. Sistemas distribuídos estão suscetíveis a falhas arbitrárias nas entidades computacionais envolvidas, falhas nos meios de comunicação, atrasos na comunicação e também suscetíveis a problemas de consenso entre as entidades distribuídas. Partindo dessa perspectiva, o tratamento das falhas relacionadas à comunicação entre as entidades é de suma importância quando o sistema distribuído em questão dá suporte à execução de uma aplicação hipermídia, principalmente para a manutenção da consistência temporal definida para a aplicação.

Como discutido na seção anterior, a implementação de referência do subsistema Ginga-NCL suporta a tolerância e recuperação de falhas, especificando um plano de recuperação. Tal plano torna a implementação de referência resiliente, oferecendo também resiliência para as aplicações interativas em execução. O tratamento das falhas discutido, no entanto, contempla apenas os componentes do Ginga-NCL executados localmente.

Esta seção discorre sobre um conjunto representativo de falhas que podem acontecer durante a apresentação de uma aplicação Ginga-NCL multi-dispositivo, as quais foram consideradas para a implementação de mecanismos de proteção adicionais para a implementação de referência do Ginga-NCL. Munido de tais

mecanismos, o plano de recuperação disponível para um dispositivo base tratará também falhas causadas por dispositivos secundários quando da execução de aplicações NCL multi-dispositivo. Tais falhas serão apresentadas de acordo com os dois diferentes modelos de comunicação do Ginga-NCL: o ativo e o passivo.

4.1 Falhas de comunicação: classes passivas

Para realizar a orquestração de dispositivos registrados em classes passivas, o formatador NCL instancia apenas um objeto de apresentação a partir do objeto de mídia (associado a uma classe de dispositivos passiva) em um documento NCL. O objeto de apresentação é transmitido (transcodificado ou em formato de codificação cru) para todos os dispositivos registrados. O tratamento das notificações de entrada (interações do usuário) geradas pelos dispositivos em classes passivas não é individualizado (durante a apresentação a navegação é compartilhada) e um mesmo conteúdo é reproduzido nos dispositivos.

A implementação de referência do Ginga-NCL conta com mecanismos de resiliência, recuperando-se e tolerando falhas em seus módulos exibidores [12], o que contempla também o tratamento local dos componentes que manipulam (renderizam ou transcodificam) objetos de mídia a serem transmitidos para dispositivos passivos.

Um conjunto de possíveis falhas durante a transmissão do objeto de apresentação do dispositivo base para os dispositivos passivos e também durante a comunicação necessária para a correta orquestração dos dispositivos foi elencado para tratamento através de mecanismos adicionais a serem incorporados ao plano de recuperação do Ginga-NCL.

Levando em consideração as características da comunicação com dispositivos de classe passiva, a implementação de referência do Ginga-NCL [5] realiza a comunicação com dispositivos passivos através de datagramas UDP, utilizando IP *multicast* para criação de grupos de dispositivos que integram uma classe passiva (outros protocolos, como RTP, também podem ser utilizados para dar suporte ao modelo de comunicação passiva). Uma vez que o dispositivo base inicia a execução de um objeto de mídia associado a uma classe de dispositivos passiva, o fluxo de bits oriundo do adaptador apropriado (um renderizador ou transcodificador) é empacotado em datagramas e enviado em uma taxa adequada para o grupo *multicast* que foi criado para agregar os dispositivos da classe.

Dispositivos que se registrarem no grupo *multicast* depois da inicialização de um objeto de mídia associado a uma classe passiva passarão a receber o fluxo do objeto de apresentação a partir da primeira amostra disponível para o grupo no momento de sua entrada. Falhas de sincronização (i.e. dispositivos em uma mesma classe passiva visualizando instantes diferentes de um conteúdo transmitido pelo dispositivo base) podem ocorrer, como são muitos os fatores envolvidos (capacidade de processamento dos dispositivos, largura de banda de rede disponível etc.) é virtualmente impossível garantir precisamente que todos os dispositivos reproduzam o objeto de apresentação concomitantemente, amostra por amostra. Em redes locais e com dispositivos com especificações equivalentes, a implementação de referência do Ginga-NCL apresentou resultados satisfatórios de sincronismo distribuído com o modelo de comunicação passiva [9], em testes com objetos de mídia que geraram taxas de transmissão baixas. Mecanismos de QoS podem ser associados a

uma plataforma de exibição NCL distribuída em particular, porém este cenário foge do escopo deste trabalho.

O dispositivo base deve ser capaz de detectar quando uma classe passiva não possui mais membros (por exemplo, em grupos IP *multicast* através de uma mensagem IGMP "*group specific query*") [9], o que deve ser tratado como uma falha quando essa classe está associada a um objeto de mídia em execução. Além disso, as de notificações de entrada (interações do usuário) originadas pelos dispositivos passivos podem gerar falhas, visto que, dado que os dispositivos podem consumir o objeto de apresentação com diferentes taxas, as notificações de interação de um dispositivo que visualiza o fluxo de bits de um determinado objeto de apresentação com um alto atraso podem chegar ao dispositivo base em um momento que o estado do objeto de apresentação já foi modificado (por interações recebidas anteriormente, enviadas por dispositivos com menor latência na comunicação). Para que a ação de navegação atrasada no objeto (transições nos eventos de seleção ou atribuição) não aconteça, a notificação com latência em excesso deve ser tratada como uma falha.

4.2 Falhas de comunicação: classes ativas

Na comunicação com os dispositivos registrados em classes ativas, o formatador NCL controla remotamente exibidores de mídia, determinando quais objetos de mídia estes reproduzirão com controle local. Cada dispositivo filho associado a uma classe ativa instancia seu próprio objeto de apresentação, navegando individualmente e notificando o formatador no dispositivo pai sobre as transições das máquinas de estado dos eventos de apresentação (ocorrência de âncoras temporais em mídias contínuas, por exemplo), de seleção e de atribuição dos objetos de mídia orquestrados remotamente. O dispositivo pai pode entregar os objetos de mídia ou indicar como o dispositivo filho pode recuperar os objetos de forma que estejam disponíveis quando orquestrados (vide o plano de pré-busca apresentado em [13]).

A comunicação necessária para a orquestração (envio e recebimento de notificações) de dispositivos de classe ativa é feita através de um canal *unicast* (troca de mensagens via TCP na implementação de referência do Ginga-NCL [5]) e cada dispositivo possui um identificador único. A quantidade máxima e mínima de dispositivos em uma classe ativa pode ser delimitada [8]. Como um objeto de apresentação é instanciado em cada dispositivo filho registrado em uma classe ativa, as falhas que ocorrerem nos exibidores remotos devem ser detectadas e notificadas ao orquestrador no dispositivo pai.

Falhas na orquestração de dispositivos registrados em classes ativas podem acontecer no registro dos dispositivos e durante a troca de mensagens durante a apresentação distribuída. Uma situação de falha se dá quando um dispositivo ativo se registra em uma classe e a mesma já possui objetos de mídia em execução, e o mesmo não recebe as notificações de orquestração anteriores ao seu registro. A mesma inconsistência de estado pode ser causada por falhas arbitrárias nos exibidores durante a apresentação nos dispositivos ativos, o que deve ser detectado pelo orquestrador no dispositivo base e tratado como uma falha.

Como mencionado em [8], um objeto de mídia associado a uma classe ativa só pode receber notificações de transição de seu evento de apresentação se a quantidade de dispositivos registrados na classe for compatível com as da definição da classe (e maior que zero). O valor é dinâmico e durante uma apresentação distribuída, por um conjunto de fatores não determinísticos, o

limite inferior pode ser violado, o que deve ser tratado como uma falha.

5. INTEGRAÇÃO E TESTES

A recuperação proativa é realizada para o rejuvenescimento dos serviços de comunicação necessários para a comunicação com múltiplos dispositivos. Os serviços que dão suporte ao módulo de integração de dispositivos (apresentado na seção 2) são monitorados pelo módulo de recuperação proativa que, quando necessário (por exemplo, quando um serviço de rede torna-se indisponível) realiza rejuvenescimento do serviço.

Os mecanismos adicionais que serão descritos nesta seção foram desenvolvidos no intuito de viabilizar que o Gerente de Integração de Dispositivos notificasse o Gerente de Recuperação (componente do do Módulo de Recuperação apresentado na seção 3) das falhas ocorridas na comunicação com dispositivos secundários, viabilizando a recuperação e tratamento de falhas durante a apresentação de uma aplicação hipermídia distribuída, baseado em políticas pré-definidas. As falhas consideradas durante a comunicação para a orquestração de dispositivos secundários requerem recuperação reativa.

As falhas analisadas na seção anterior são detectadas pelo Gerente de Integração de Dispositivos e notificadas ao Gerente de Recuperação. Ao receber uma notificação de falha, o Módulo de Recuperação as classifica conforme os tipos apresentados na coluna “Falha” da Tabela 1 e realiza seu registro e consulta da política de recuperação reativa. A definição da política oferece apenas a opção de quantas falhas podem ocorrer e em um período de tempo. Caso esses valores não sejam definidos, não é realizada a validação da política. Após a consulta da política de recuperação, de acordo com o tipo de falha, é definida a ação, como apresentado na coluna “Ação” na Tabela 1.

Tabela 1. Tipos de comunicação, Tipos de Falhas e Respectivas Ações.

Classe	Falha	Ação
Ambas	Não há mais dispositivos registrados a uma classe que possui uma mídia associada em execução.	Mídia é abortada.
Passiva	Notificação de seleção ou atribuição chega atrasada	Remover dispositivo da classe.
Ativa	Número de dispositivos da classe torna-se inferior ao de sua definição.	Mídia é abortada.
Ativa	Dispositivo não responde verificação de estado no tempo definido (<i>timeout</i>).	Remover dispositivos da classe.
Ativa	Dispositivo apresenta estado inválido (inconsistência temporal).	Dispositivo base sincroniza dispositivo secundário de acordo com andamento da apresentação.

Um objeto de mídia deve ser abortado (Tabela 1) quando a classe de dispositivos ao qual está associado passa a ter um número de dispositivos inferior ao de sua definição (no caso das classes ativas) ou quando o número de dispositivos torna-se zero (em ambas as classes). O Gerente de Recuperação implementado possui uma interface de comunicação com o Formattador NCL (Figura 3) e notifica-o para garantir que os relacionamentos do objeto não sejam mais considerados na apresentação.

Durante a orquestração dos dispositivos ativos e passivos, a latência do canal de comunicação entre o dispositivo pai e um

dispositivo filho pode exceder o limite necessário para a manutenção da consistência temporal da aplicação. O Gerente de Integração de Dispositivos detecta o excesso de latência em uma conexão com um dispositivo secundário dependendo do tipo da classe do dispositivo (vide Tabela 1). Nesse caso, e também quando a conexão com um dispositivo é finalizada durante a orquestração, o Gerente de Recuperação deve notificar o Gerente de Integração de Dispositivos (Figura 3) que, através de seu submódulo Gerente de Classes de Dispositivos, removerá o registro do dispositivo causador da falha junto à sua classe de dispositivo.

Para que o estado dos dispositivos filhos não seja inválido, eles devem receber informações sobre o andamento da apresentação do dispositivo pai. Assim a consistência temporal é mantida, o que é verificado periodicamente durante a orquestração. O Módulo de Recuperação se comunica com o Formattador NCL (Figura 3) e o Módulo de Integração de Dispositivos, para que o dispositivo pai possa sincronizar o dispositivo filho, o que preferencialmente deve ser feito através cadeias temporais e Grafos Temporais Hipermídia (GTH) [13], que oferecem suporte ao início ou retomada síncrona da apresentação. No entanto, na versão atual da implementação de referência, o suporte ao GTH não está implementado, de forma que o sincronismo é mantido pelo Gerente de Classes de Dispositivos do dispositivo pai através do registro das mensagens de orquestração das mídias associadas a uma classe de dispositivos, que é entregue a um dispositivo secundário em caso de detecção de inconsistência temporal. A Figura 3 apresenta o Gerente de Recuperação Reativa do Ginga-NCL (*ReactiveRecoveryManager*) e as interfaces de monitoramento de falhas com componentes dos módulos de risco do Ginga-NCL (um deles é o Gerente de Integração de Dispositivos – *MultiDeviceManager*).

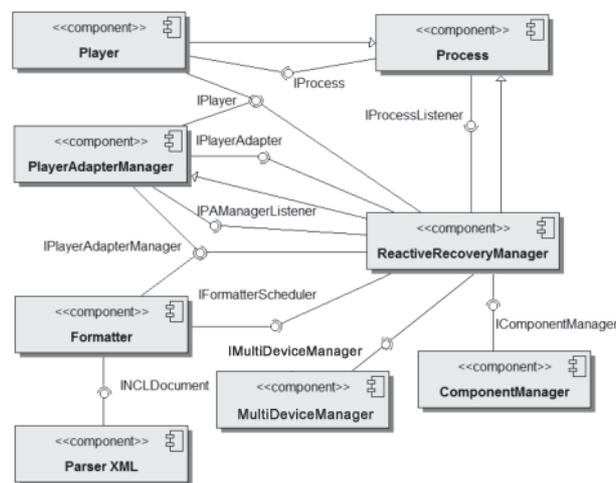


Figura 3. Diagrama de Componentes para Recuperação Proativa do Ginga-NCL

5.1 Cenário de Testes

Duas aplicações NCL multi-dispositivo são usadas como cenário de testes para os mecanismos de proteção propostos no presente trabalho.

A primeira aplicação é um jogo da velha distribuído, onde dispositivos de classe passiva podem marcar os espaços adequados, intercalando ‘O’ e ‘X’ para marcação, sem restrição de que dispositivo controla que símbolo; a ordem da jogada determina qual símbolo é usado e como trata-se de uma classe

passiva não há restrição de quantas jogadas cada dispositivo pode fazer em sequência.

A segunda aplicação é uma apresentação distribuída para dispositivos de classe ativa, onde um conteúdo pode ser visualizado tanto no dispositivo base quanto nos dispositivos filhos. Posteriormente um jogo para dois jogadores é executado, onde dois dispositivos ativos atuam com diferentes papéis em uma disputa de pênalti (um ataca e o outro defende) e o resultado é exibido na tela do dispositivo base.

5.1.1 Comunicação Passiva

Uma aplicação NCL contendo um “Jogo da Velha” é embutida em uma outra aplicação em execução em um dispositivo base, e é associada a uma classe de dispositivos passivos, cujos usuários podem navegar na aplicação (realizar jogadas) através de toques na tela dos dispositivos. As mensagens geradas pela navegação no objeto de apresentação são notificadas ao dispositivo base, que atualiza o estado do objeto e o envia aos dispositivos.

Duas falhas são testadas no cenário de execução para validar os mecanismos de recuperação reativa implementados. A primeira falha considera o término da conexão entre todos os dispositivos passivos e o dispositivo pai (dispositivo pai detecta que grupo *multicast* está vazio). A segunda falha foi a injeção de uma latência artificial para o envio das mensagens de notificação de navegação em um dos dispositivos passivos.

A aplicação inicial foi executada em um dispositivo base MiniPC AOpen (Linux), com a implementação de referência do Ginga-NCL; foram utilizados como dispositivos passivos dois celulares iPhone com o Ginga-NCL para dispositivos passivos [9].

A aplicação inicial define que um vídeo seja exibido na tela do dispositivo base enquanto a aplicação NCL embutida é exibida nos dispositivos de classe passiva e também em parte da tela do dispositivo base (um mesmo objeto de apresentação, através do esquema de URL “ncl-mirror://” suportado pelo NCL [6]).

Durante a primeira execução da aplicação, dois dispositivos passivos se registram junto ao dispositivo base e recebem as primeiras amostras referentes à renderização da parte da aplicação NCL referente ao “Jogo da Velha”. A aplicação possui um elo com condição de ativação “onAbort” do objeto de mídia associado à classe passiva (o NCL embutido do “Jogo da Velha”), que inicia a exibição de uma imagem (com uma mensagem de erro) na tela do dispositivo base. Enquanto o objeto de mídia do jogo ainda está em execução, os dois dispositivos passivos são desconectados da rede (desligados), o que ativa o elo da imagem com a mensagem de erro (que é exibida na tela do dispositivo base) pois o Formatador NCL é notificado pelo Módulo de Recuperação após a detecção da falha.

A segunda execução de teste também utiliza dois dispositivos passivos, que se registram junto ao dispositivo base executando a mesma aplicação que embute o “Jogo da Velha”. Um dos dispositivos passivos possui uma variação da implementação protótipo de dispositivo passivo [9] que injeta um atraso para envio de notificações de navegação. Uma vez que durante a execução da aplicação os dispositivos passivos passam a receber as amostras referentes à aplicação do “Jogo da Velha”, apenas o celular sem a versão alterada do cliente Ginga-NCL começa a interagir (através de toques na tela referentes às jogadas a serem realizadas na matriz 3x3), o que faz com que o dispositivo base atualize o estado do objeto de apresentação nos dois celulares. Quando o celular com a versão alterada do cliente Ginga-NCL classe passiva (com injeção de latência) começa a interagir simultaneamente com o objeto de apresentação, o dispositivo base

detecta a latência em excesso do dispositivo e solicita o fim do seu registro junto à classe passiva (envia uma notificação ao dispositivo passivo que faz com que o dispositivo se desconecte do grupo *multicast*). O jogo passa a ser exibido apenas no dispositivo passivo sem a latência injetada na comunicação com o dispositivo base.

5.1.2 Comunicação Ativa

Durante a Copa do Mundo de 2010 na África do Sul, o Brasil apresentou algumas inovações tecnológicas tendo em vista o que pode ser esperado para as transmissões da Copa do Mundo de 2014. Uma dessas apresentações foi a aplicação interativa “Brazil 14-Bis”, sendo executada no Ginga-NCL com múltiplos dispositivos de apresentação (Figura 4). O vídeo principal da aplicação é composto por clipes de edições passadas da Copa do Mundo FIFA e durante a exibição o uso de múltiplos dispositivos é extensivamente explorado. Dois desses momentos são usados como cenário para injeção de falhas e teste para os mecanismos propostos por este trabalho.

No primeiro momento, uma parte da aplicação é exibida na tela do dispositivo base, apresentando graficamente informações sobre os países que já foram campeões da Copa do Mundo. Os dispositivos secundários exibem uma aplicação que oferece uma opção (Figura 4) que, se selecionada, fará com que a aplicação sendo exibida no dispositivo base seja finalizada e uma versão adaptada da aplicação seja exibida apenas nos dispositivos secundários. Este momento será utilizado para injeção de falhas de sincronismo no registro de dispositivos ativos e falhas de desconexão de todos os dispositivos registrados em classes ativas.

O segundo momento considerado é um jogo para dois usuários, que simula uma disputa de pênaltis. O primeiro usuário a se registrar atuará como o jogador de linha e o segundo usuário como o goleiro. Em um momento específico o dispositivo base requisita que os usuários, através das telas dos seus dispositivos ativos, escolham para que lado chutar ou pular, e o resultado da disputa (gol ou defesa) é apresentado na tela do dispositivo base. Este momento será utilizado para injeção de falha de violação do limite inferior de número de dispositivos associados a uma classe.

Foi utilizado, como dispositivo base, um MiniPC AOpen (Linux) com a implementação de referência do Ginga-NCL e como dispositivos ativos dois celulares Android (Motorola Milestone e HTC Legend, com a API Android 2.1) com uma versão adaptada do Ginga-NCL para Android [14].



Figura 4. Execução da aplicação “Brazil 14-Bis”.

O primeiro trecho testado da aplicação inicialmente apresenta um vídeo principal e um conjunto de imagens e animações na tela do dispositivo base com informações relacionadas às seleções campeãs de todas as Copas do Mundo (até 2010). Inicialmente apenas um dispositivo ativo se conecta ao dispositivo base. Em um determinado momento uma aplicação é exibida no dispositivo ativo: uma imagem (um botão) que se selecionado altera o valor

de uma variável no nó de definições (*settings*) do Ginga-NCL [6]. A aplicação no dispositivo base, através de um elo com ativação “*onEndAttribution*” [6] associado a essa variável, finaliza a exibição da parte da aplicação que estava sobre o vídeo principal na tela do dispositivo base e inicia a apresentação de uma versão adaptada de tal parte no dispositivo ativo. Em um momento posterior um segundo dispositivo ativo se conecta ao dispositivo base. O dispositivo base detecta o estado inconsistente do segundo dispositivo ativo (dispositivo não está apresentando nenhum objeto de mídia), e então realiza o envio das mensagens de orquestração dos objetos de mídia associados à classe ativa. Logo após o registro o segundo dispositivo passa a apresentar a aplicação correntemente associada aos dispositivos ativos.

A aplicação continua, com os dois dispositivos exibindo a versão adaptada da aplicação com informações sobre os campeões da copa do mundo. Enquanto a aplicação ainda está em execução, os dois dispositivos são desconectados da classe ativa (desligados). A aplicação no dispositivo base possui um elo com condição de ativação “*onAbort*” do objeto de mídia associado à classe ativa: quando os dois dispositivos ativos são desligados, o elo faz com que a parte da aplicação com informações sobre os campeões das copas seja exibida novamente na tela do dispositivo base.

Durante a execução do segundo trecho da aplicação a ser considerado para os testes, é iniciada uma aplicação NCL em dois dispositivos ativos conectados, que, de acordo com a ordem de registro na classe ativa, exibem uma versão de ataque ou defesa para uma simulação de disputa por pênaltis. A classe foi definida para dois dispositivos (máximo e mínimo) [8]. Antes que a jogada seja realizada, um dos dispositivos ativos se desconecta (é desligado) do dispositivo base, que aborta o objeto de mídia associado à classe ativa. Então, um elo com condição de ativação “*onAbort*” da aplicação NCL associada aos dispositivos ativos faz com que uma imagem seja exibida no dispositivo base notificando o fim do jogo. Além disso, o Formatador NCL faz com que a aplicação NCL sendo executada pelo dispositivo ativo que permaneceu conectado seja finalizada.

6. CONSIDERAÇÕES FINAIS

Diversas plataformas dão suporte a execução de aplicações interativas distribuídas no âmbito da TV Digital. No entanto, com exceção do Ginga-NCL, o suporte é oferecido através da definição de uma API imperativa, como é o caso do ambiente de execução imperativo baseado em Java Ginga-J [3] e também as API Java do ARIB [15]. Nesses casos, a tolerância falhas não é oferecida pela plataforma, e o desenvolvedor das aplicações, através de instruções da linguagem, deve tratar exceções na comunicação.

A maioria dos trabalhos que discutem tolerância a falhas e recuperação no âmbito das aplicações multimídia distribuídas versam sobre provisionamento de recursos (distribuídos) através de mecanismos de QoS [16], e até onde as pesquisas dos autores deste artigo alcançaram, nenhum dos trabalhos se propunha a realizar o sincronismo entre dispositivos envolvidos na execução de uma aplicação hipermídia distribuída em uma rede doméstica.

A evolução deste trabalho poderá considerar um escopo mais amplo de falhas e flexibilizar os mecanismos de proteção, permitindo, o uso de parâmetros para as políticas de recuperação na definição das classes de dispositivos. No presente trabalho, as falhas de comunicação consideradas estão relacionadas a um ambiente de execução de aplicações NCL distribuídas baseadas em redes domésticas, porém outras infraestruturas podem ser consideradas. A definição de classes atualmente utilizada [8] deve

ser modificada para dar suporte a tais definições, o que será objeto de trabalhos futuros.

7. REFERÊNCIAS

- [1] MILLER B. A., et al. Home Networking with Universal Plug and Play. IEEE Communications Magazine, pp105-109 (2001).
- [2] Open Service Gateway initiative Alliance. Disponível em <http://www.osgi.org> (Acesso em Abril de 2011).
- [3] SILVA L. D. N. e, et al. Suporte para desenvolvimento de aplicações multiusuário e multidispositivo para TV Digital com Ginga. T&C Amazônia Magazine. N. 12, ISSN 1678-3824, pp 75-84. Manaus, AM : s.n., 2007.
- [4] LIN C.-L. et al.: A Wrapper and Broker Model for Collaboration between a Set-Top Box and Home Service Gateway. IEEE Transactions on Consumer Electronics, Vol. 54, No. 3, pp 1123-1128, 2008.
- [5] SOARES L. F. G. et al, 2007. Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System. Journal of the Brazilian Computer Society, v. 12, p. 37-46 (2007).
- [6] ABNT NBR 15606-2 Associação Brasileira de Normas Técnicas. Digital Terrestrial Television Standard 06: Data Codification and Transmission Specifications for Digital Broadcasting, Part 2 – GINGA-NCL: XML Application Language for Application Coding.
- [7] ITU-T Recommendation H.761, 2009. Nested Context Language (NCL) and Ginga-NCL for IPTV Services. Genebra, Abril, 2009.
- [8] BATISTA, C.E.C.F, et al. Estendendo o uso das classes de dispositivos Ginga-NCL. Proc. of WebMedia 2010. Belo Horizonte, Brasil (2010).
- [9] SOARES, L. F. G., et al. Multiple Exhibition Devices in DTV Systems. Proc. of ACMMultimedia2009. Beijing, China (2009).
- [10] WAP Forum, WAG UAProf. Technical Report WAP-248-UAPROF-20011020-a, 2001.
- [11] KLYNE G., REYNOLDS F., WOODROW C., OHTO H., HJELM J., BUTLER M. H., TRAN L. Composite Capability/Preference Profiles (CC/PP): Structure and vocabularies. W3C working draft, 2004.
- [12] MORENO, M.F., SOARES, L. F. G. Resilient Hypermedia Presentations. Proc. of XXI IEEE International Symposium on Software Reability Engineering (Workshop W2, Software Aging and Rejuvenation). Novembro, 2010.
- [13] Costa, R. et al. “Intermedia Synchronization Management in DTV Systems”. In Proc. of DocEng, ACM Symposium on Document Engineering, São Paulo, 2008.
- [14] DAHER, G., et al. Ginga-NCL em Dispositivos Portáteis: Uma Implementação para a Plataforma Android. Proc. of WebMedia 2010. Belo Horizonte, (2010).
- [15] Association of Radio Industries and Businesses. ARIB STD-B23 Version 1.2: Application Execution Engine Platform for Digital Broadcasting. 2004.
- [16] VEERAVALLI, B. et al. Fault-tolerant analysis for multiple servers movie retrieval strategy for distributed multimedia applications. Springer Multimedia Tools and Applications, Volume 32, Number 1, 1-27.