

Uma Abordagem para o Desenvolvimento de Serviços para o Sistema Cossack

Giuliano Araujo Bertoti, Felipe Afonso de Almeida, Davi D'Andréa Baccan

Laboratório de Interação, Comunicação e Mídia (LINCOM)
Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50, 12228-900
São José dos Campos, SP, Brasil

gbertoti@hotmail.com, felal@ida.liu.se, davi@comp.ita.br

Abstract. *One of the possible solutions that are being proposed to the problem of digital exclusion is based on the fact that most of Brazilians have a television set. The solution is the use of interactive TV systems to offer services to population that nowadays are only being offered through the Internet. However, interactive TV systems currently available in Brazil both impose high subscription fee and do not offer services such as education and health. Moreover, they do not allow owners of local business to make their services available through television. This paper presents the interactive TV system called Cossack System, that provides a solution to the existing problems of current iTV systems. Also, it presents an approach that has been created to the development of services to the Cossack System.*

Resumo. *Uma das soluções que estão sendo propostas para o problema da exclusão digital no Brasil está baseada no fato de que a grande maioria da sua população possui pelo menos um aparelho de televisão. Trata-se da adoção de sistemas de televisão interativa para a prestação de serviços à população que atualmente, somente são prestados através da Internet. Porém, os sistemas de televisão interativa disponíveis atualmente no Brasil cobram taxas elevadas de assinatura, não prestam serviços nas áreas da educação e saúde, e não permitem que os comerciantes locais disponibilizem seus serviços através da televisão. Este artigo apresenta o Sistema Cossack, que visa solucionar os problemas existentes nos sistemas atuais de televisão interativa, e também a abordagem que está sendo criada para o desenvolvimento de serviços para este Sistema.*

1. Introdução

O acesso a computadores reflete as desigualdades econômicas e sociais do Brasil. De acordo com o “Mapa da Exclusão Digital” [FGV 2003], divulgado pela Fundação Getúlio Vargas (FGV), apenas 12,46% dos brasileiros têm computador em casa, sendo que este percentual é reduzido para 8,31% se considerados os computadores conectados à internet. Isso significa que uma grande parcela da população brasileira não possui acesso a uma série de serviços gratuitos que são disponibilizados através da Internet. Dentre estes serviços pode-se citar os prestados por Universidades, ONG’s, Portais de utilidade pública, Governo Federal e outros.

Uma das possíveis soluções para este problema é a utilização de sistemas de televisão interativa. Estes sistemas podem ser denominados como sistemas resultantes da convergência entre sistemas televisivos e sistemas computacionais nos quais o telespectador, através de uma televisão conectada a um aparelho chamado set-top box [Consulting 2003], pode ter acesso aos serviços citados acima. Exemplos de sistemas de televisão interativa, atualmente sendo utilizados no Brasil, são DirecTV [DIRECTV 2004] e SKY [SKY 2004]. Esta solução tem como justificativa o fato de que, segundo dados do [IBGE 2004], mais de 146 milhões de habitantes possuem pelo menos um aparelho de TV em casa. Além disso, o uso da TV apresenta algumas vantagens. O custo dos aparelhos set-top box, necessários para a implantação de sistemas de TV interativa, é menor quando comparado ao custo de um computador e freqüentemente estes aparelhos são emprestados, pelo próprio sistema de televisão interativa, sem custos. Outro fator importante a ser considerado é que, diferentemente do computador, a TV representa um aparelho familiar com o qual a população brasileira convive e interage a décadas.

Porém, os sistemas de televisão interativa implantados atualmente no Brasil apresentam alguns problemas em relação a prestação de serviços:

- Os serviços fornecidos ao telespectador são freqüentemente relacionados ao comércio eletrônico e entretenimento, privando o telespectador da oportunidade de acessar serviços importantes prestados através da internet, como por exemplo, educação à distância;
- a mensalidade cobrada dos assinantes de sistemas de televisão interativa é elevada, principalmente quando considera-se que existem muitos provedores de internet gratuitos;
- os serviços são prestados de forma global e uniforme, ou seja, um mesmo serviço é oferecido para todo o país. Com isso a cultura e o comércio da população de cada região, principalmente de regiões remotas, são prejudicados;
- os serviços de comércio eletrônico são freqüentemente prestados pelo próprio sistema de televisão interativa, caracterizando assim um monopólio que enfraquece o comércio local.

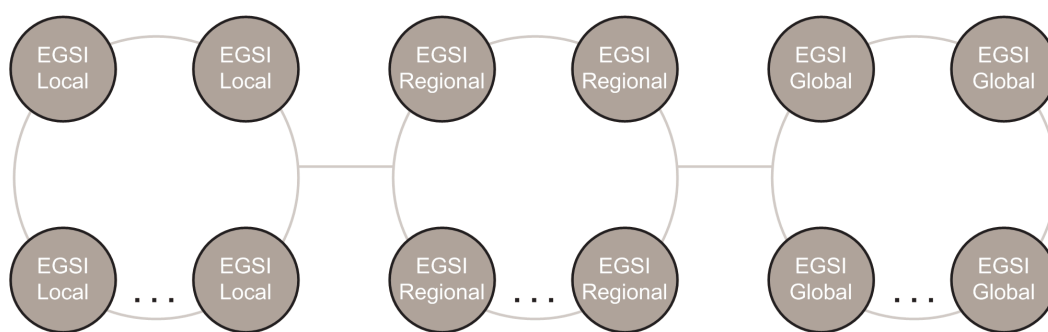
Com o objetivo de resolver os problemas apresentados acima e, portanto, utilizar sistemas de televisão interativa como uma possível solução para a prestação de serviços para a população brasileira foi proposto o Sistema Cossack [Almeida 1999]. O Sistema Cossack, que será apresentado com mais detalhes na seção 2, é um sistema de televisão

interativa que visa fornecer uma plataforma para que Universidades, ONG's, Portais de utilidade pública, Governo Federal, comerciantes locais, dentre outros, possam fornecer, através da televisão, serviços de forma local, ou seja, cada região ou cidade tem a liberdade para prestar os serviços mais adequados para sua população. A concepção da prestação de serviços proposta pelo Sistema Cossack é diferente da concepção dos demais sistemas de televisão interativa, e portanto uma nova forma de desenvolver estes serviços deve ser criada.

Os objetivos deste artigo são, mostrar como os serviços locais prestados pelo Sistema Cossack podem efetivamente resolver os problemas dos serviços não prestados pelos sistemas de televisão interativa atuais e apresentar uma abordagem para permitir o desenvolvimento de serviços locais do Sistema Cossack. O artigo está dividido como segue: Na seção 2 serão apresentados os principais conceitos do Sistema Cossack bem como sua abordagem de serviços locais. Na seção 3 serão apresentados os principais conceitos da Arquitetura de Software e sua relação com o Sistema Cossack. Na seção 4 será apresentada a abordagem que está sendo criada para o desenvolvimento dos serviços Cossack. Finalmente, a seção 5 conclui o artigo.

2. O Sistema Cossack

O Sistema Cossack é um sistema de televisão interativa [Almeida 2004] que se diferencia dos demais sistemas existentes no que diz respeito à prestação de serviços. Os serviços são disponibilizados e prestados de forma hierárquica. Esta hierarquia é composta por Estações de Gerenciamento de Serviços Interativos (EGSI's), que podem ser divididas em EGSI's locais, EGSI's regionais, e EGSI's globais. A rede de EGSI's locais (as quais prestam serviços para pequenas cidades ou bairros de grandes cidades), está conectada com a rede de EGSI's regionais (as quais prestam serviços para regiões englobando pequenas cidades, ou uma grande cidade), que por sua vez está conectada a rede de EGSI's globais (as quais prestam serviços para estados ou países). A Figura 1 ilustra como esta hierarquia está estruturada.



**Figura 1. Hierarquia do Sistema Cossack:
Redes de EGSI's Locais, Regionais e Globais.**

Essa abordagem hierárquica permite que um telespectador acesse serviços, por exemplo de educação, saúde, comércio e entretenimento, disponíveis em uma EGSI local, que presta serviços para sua cidade ou bairro. Desta forma, tanto as particularidades de cada região quanto o comércio local são fortalecidos, conforme veremos nos exemplos de serviços da subseção 2.1.2. Porém, se o serviço requisitado

pelo telespectador não for encontrado em alguma EGSi local, este serviço pode ser solicitado para a rede de nível hierárquico superior. Por exemplo, se algum serviço requisitado não for prestado por alguma EGSi Local, esta requisição pode ser feita para alguma EGSi Regional, e por fim para alguma EGSi Global.

Uma Estação de Gerenciamento de Serviços Interativos, responsável pela prestação de serviços a população através da televisão, possui um Provedor de Serviços Interativos (PSI) e um Provedor de Serviços Interativos-TV (PSITV):

- **PSI:** é responsável por desenvolver e prestar os serviços (como aqueles que serão descritos em 2.1.2).
- **PSITV:** é responsável por transmitir não somente o áudio e o vídeo de um programa de TV convencional, mas também banners (conforme será explicado na subseção 2.1.1.) sincronizados com este programa. Cabe ao PSITV informar aos PSI's sobre a programação a ser disponibilizada para que a veiculação de serviços possa ser feita. A Figura 2 ilustra a arquitetura de uma EGSi.

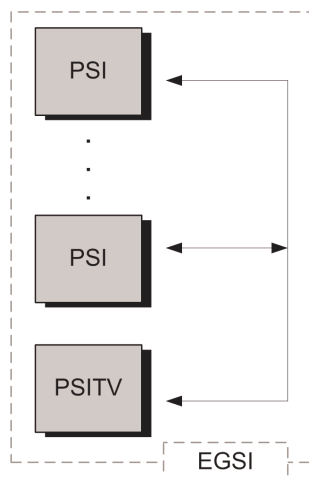


Figura 2: Arquitetura de uma EGSi.

2.1. Serviços Cossack

No contexto do Sistema Cossack, os serviços são aplicações nas mais diversas áreas (saúde, educação, comércio e entretenimento). Estes serviços são desenvolvidos e prestados pelos PSI's. Os telespectadores podem acessar estes serviços através de um programa de televisão.

2.1.1. Acesso aos Serviços Cossack

Conforme visto anteriormente, o PSITV transmite o vídeo e áudio, correspondentes a um programa de televisão convencional, e os banners. Os banners contêm hiperlinks (ou hiperelos, ou ainda, elos), que são endereços para determinados serviços. Quando o telespectador aciona um destes banners, por exemplo através de um controle remoto, ele ativa um serviço fornecido por um determinado PSI, conforme ilustrado na Figura 3. Um programa de televisão que possui banners contendo “hiperlinks” é chamado Hiperprograma.

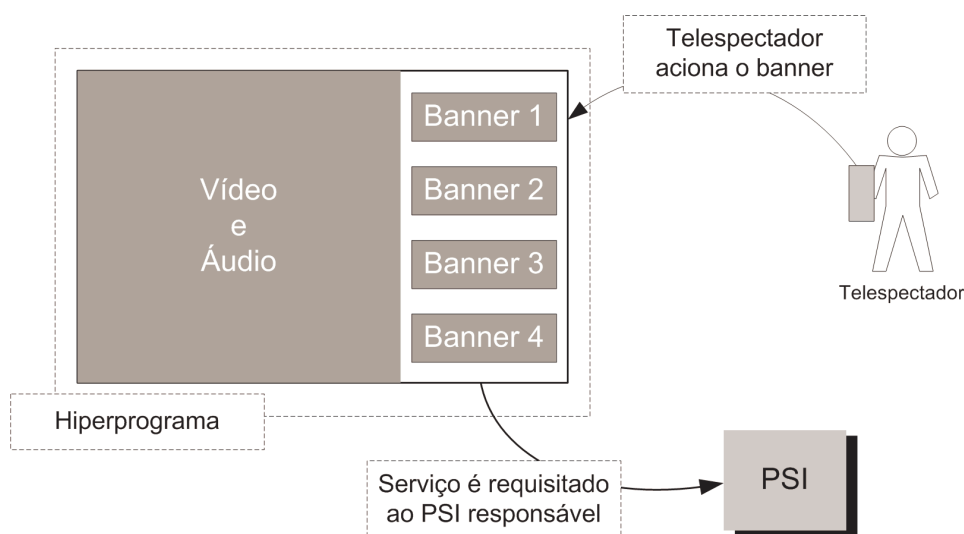


Figura 3: Relação entre Telespectador, Hiperprograma e PSI's.

2.1.2. Exemplos de Serviços Cossack

A seguir serão apresentados exemplos de algumas áreas onde os serviços poderão ser desenvolvidos e prestados pelos PSI's:

- **Saúde:** Enquanto o PSITV transmite para o telespectador um Hiperprograma sobre cuidados com a saúde, o PSI local poderá prestar um serviço que permite ao telespectador obter informações sobre prevenção e tratamento de doenças existentes na região do país onde ele vive. Por exemplo, se este telespectador residir na floresta amazônica ele poderá receber informações sobre prevenção e tratamento de doenças como malária e febre amarela, enquanto se o telespectador morar no sertão do Piauí ele poderá receber informações sobre a utilização de plantas nativas para o combate a anemia. Além disso, o PSI pode prestar um serviço de Telemedicina, ou seja, um serviço no qual o telespectador assume o papel de paciente e requisita ao PSI uma consulta com um médico local. Se o médico requisitado pelo telespectador não estiver disponível no PSI local, a requisição deste serviço poderá ser feita ao PSI regional e assim por diante.
- **Educação:** Os PSI's podem realçar os programas educacionais oferecendo material didático adicional e exercícios. Cada PSI local pode fornecer um tipo de material didático de acordo com sua população. Por exemplo, em um programa de TV sobre história do Brasil os telespectadores que moram na Bahia poderiam receber do PSI material didático sobre a Revolta de Canudos, enquanto os telespectadores que moram em Minas Gerais poderiam receber informações sobre a Inconfidência Mineira.
- **Comércio-Eletrônico:** Com o objetivo de fortalecer o comércio local, os PSI's podem fornecer aos seus telespectadores serviços de compras prestados pelos comerciantes locais. Como exemplo, ao invés do telespectador requisitar um serviço de vídeo sob demanda fornecido pelos sistemas de TV interativa convencionais, o telespectador alugaria um vídeo em uma locadora da sua região através de um serviço prestado por um PSI Local, por exemplo. A diferença é que,

ao invés de pagar pelo vídeo para um grande sistema de televisão interativa, o telespectador pagaria para uma locadora da sua cidade (local). Além disso, os PSI's poderiam cobrar dos comerciantes locais pequenas taxas para a disponibilização dos serviços, as quais seriam revertidas para a produção e prestação gratuita de serviços nas áreas da saúde e educação, isentando os telespectadores do pagamento de taxas de assinatura.

- **Entretenimento:** Os serviços de entretenimento fornecidos por PSI's podem ser informações culturais sobre cada região ou até mesmo jogos regionais. Por exemplo, telespectadores de Pernambuco poderiam requisitar aos PSI's locais músicas de maracatu e frevo.

2.2. Necessidade de uma Nova Abordagem para o Desenvolvimento de Serviços

Os serviços prestados pelos sistemas de televisão interativa disponíveis atualmente no Brasil são desenvolvidos por ambientes de autoria [Launchlot 2004] [OpenTV 2004] que utilizam-se de uma abordagem baseada na Orientação a Objetos (O.O.) [Horstmann 2003]. Porém, os serviços prestados pelos PSI's do Sistema Cossack possuem características diferentes (apresentadas ao longo da seção 2) baseadas em outros paradigmas e, portanto, não podem ser desenvolvidos da mesma forma. A seguir são apresentadas as razões que impedem que ambientes de autoria orientados a objeto sejam utilizados para desenvolver os serviços Cossack:

- Conforme apresentado, no Sistema Cossack os comerciantes locais podem disponibilizar seus serviços através da televisão (veja subseção 2.1.2). Portanto, a abordagem utilizada para o desenvolvimento dos serviços deve permitir e prover meios para que os desenvolvedores de serviço de um PSI comuniquem-se com os comerciantes locais. Porém, como isso não ocorre atualmente, os ambientes de autoria O.O. utilizados pelos sistemas atuais não possuem formas de comunicação entre os responsáveis pelo desenvolvimento de serviços (pessoal técnico) e os comerciantes (pessoal não técnico);
- a prestação de serviços locais é uma característica do Sistema Cossack (conforme dito ao longo da seção 2). Porém os sistemas de televisão interativa atuais não permitem esta forma de prestação de serviço e, portanto, os ambientes de autoria utilizados por eles não suportam o desenvolvimento de serviços locais;
- os ambientes de autoria atuais não permitem o reuso de funcionalidades importantes no desenvolvimento de serviços de televisão interativa, como sincronização temporal, obrigando os desenvolvedores a implementar estas funcionalidades diversas vezes [Medvidovic 1998];
- os ambientes de autoria atuais não dão suporte à especificação formal dos serviços, obrigando o desenvolvedor a possuir ferramentas para realizar esta tarefa. Além disso, os ambientes e métodos formais não são específicos para o desenvolvimento de serviços de televisão interativa;
- os PSI's precisam de ferramentas de análise específicas para os serviços do Sistema Cossack como, por exemplo, simuladores que verificam o re-direcionamento de serviços de um PSI local para um regional.

3. Arquitetura de Software

Arquitetura de Software é uma área emergente da Engenharia de Software que trata do desenvolvimento de sistemas de software com base em sua estrutura global, considerando questões referentes à arquitetura do sistema, por exemplo, como a configuração dos componentes que compõem o sistema (a forma com que os componentes que formam este sistema estão interconectados), os protocolos de comunicação, sincronização, desempenho, evolução, distribuição, dentre outras [Shaw 1996] [Garlan 2000]. De acordo com [Kerschbaumer 2002], esta definição inclui alguns aspectos importantes:

- **Abstração:** arquiteturas de software fornecem abstrações de alto-nível de um sistema, pois omitem os detalhes internos dos componentes, que não afetam suas relações com outros componentes, como, por exemplo, estrutura de dados ou algoritmos.
- **Visões Arquiteturais:** um mesmo sistema pode ser especificado através de diferentes abstrações, chamadas de visões arquiteturais [Kruchten 1995]. Como exemplo, um sistema pode ser visto através de uma visão de tempo-real e uma visão de implementação. Deve ser notado que os conceitos dados aos componentes e conectores são diferentes de acordo com a visão arquitetural considerada. Por exemplo, em uma visão de tempo-real um componente pode ser um processo, enquanto, em uma visão de implementação o mesmo componente pode ser considerado um trecho de código fonte.
- **Propriedades de Componentes:** uma arquitetura não define somente a estrutura global, mas também as propriedades dos seus componentes que influenciam a comunicação destes com os outros componentes do sistema.

3.1. Estilos Arquiteturais

Um estilo arquitetural é a definição dos componentes, conectores e interfaces existentes em uma determinada família de sistemas e as restrições para que estes componentes e conectores possam ser combinados para compor a arquitetura do sistema [Perry 1992]. Conectores são unidades de comunicação, componentes são unidades de computação e interfaces são pontos de conexão presentes em conectores e componentes. A seguir são apresentados alguns exemplos de estilos arquiteturais:

- **Cliente-Servidor:** é um estilo para famílias de sistemas distribuídos, onde os componentes podem ser clientes ou servidores e os conectores são protocolos de comunicação [Fielding 2000]. Por exemplo, um componente cliente pode requisitar um serviço para um componente servidor através de um conector de chamada remota de procedimentos (RPC).
- **Camadas:** é um estilo para sistemas organizados hierarquicamente. Cada camada é um componente e fornece serviços para a camada superior e atua como cliente da camada inferior. Os conectores são definidos por protocolos que determinam como será a interação entre as camadas [Kerschbaumer 2002].
- **Heterogêneo:** é um estilo desenvolvido a partir da combinação entre diversos estilos. Por exemplo, uma combinação entre Cliente-Servidor e Camadas. A maioria dos sistemas utiliza estilos arquiteturais heterogêneos [Shaw 1996].

3.2. Linguagens de Descrição de Arquitetura

Linguagens de Descrição de Arquitetura, conhecidas como ADLs (Architecture Description Languages), têm sido definidas para suportar a especificação de sistemas utilizando os conceitos de Arquitetura de Software [Clements 1996]. ADLs possuem notações textuais e gráficas que permitem a especificação da arquitetura de um sistema, baseada em um ou mais estilos arquiteturais. Uma ADL permite que um desenvolvedor especifique os componentes, conectores e a configuração de um sistema [Medvidovic 1997].

Além disso, ADLs freqüentemente fornecem um conjunto de ferramentas que auxiliam no desenvolvimento de sistemas, como ferramentas que permitem a especificação de um sistema de forma gráfica, geradores de código e ambientes de simulação [Shaw 1996]. Tanto as notações textuais e gráficas quanto as ferramentas fornecidas por uma ADL são de domínio específico, ou seja, são destinadas ao desenvolvimento de um tipo específico de sistema. A seguir são apresentados alguns exemplos de ADLs:

- **Darwin:** é uma ADL criada pelo Distributed Software Engineering Group – Imperial College, para o desenvolvimento de sistemas distribuídos [Magee 1995]. Uma ferramenta chamada SAA é disponibilizada para a especificação gráfica destes sistemas.
- **C2SADEL:** é uma ADL criada especialmente para o estilo arquitetural C2 [Medvidovic 1996] baseado em eventos, que suporta a definição de arquiteturas dinâmicas e distribuídas. Esta ADL permite a geração de código executável em C/C++, ADA e Java.

3.3. xADL: Uma linguagem de Descrição de Arquitetura Extensível

Com a finalidade de facilitar o desenvolvimento de novas ADLs, a Universidade da Califórnia (UCI, Irvine, USA) criou uma ADL chamada xADL [Dashofy 2002]. xADL facilita o desenvolvimento de novas ADLs pois é baseada em XML, um padrão que fornece capacidades de definir linguagens extensíveis [Bray 2000], e porque é definida de forma genérica, ou seja, não segue nenhum estilo arquitetural específico. Com isso, xADL pode ser estendida para suportar o desenvolvimento de sistemas de acordo com qualquer estilo arquitetural [Dashofy 2001]. Uma ADL criada com base em xADL permite descrever um sistema sob quatro visões arquiteturais diferentes:

- **Visão de Design:** nesta visão os componentes e conectores de um sistema serão especificados de acordo com suas informações básicas (descrição textual, autor, identificação e outras), seus comportamentos no sistema e suas restrições de uso.
- **Visão de tempo-real:** esta visão contém informações sobre o estado corrente dos componentes e conectores (executando, bloqueado, suspenso e outros), e onde os componentes estão sendo executados em um sistema distribuído (qual máquina, qual processador e outros).
- **Visão de Evolução:** esta visão contém informações sobre os componentes que devem ser adicionados, removidos e substituídos, em tempo-real, no sistema.

- **Visão de Implementação:** esta visão contém informações para que as três visões anteriores sejam utilizadas na geração de código-fonte executável.

3.4. Benefícios da Arquitetura de Software para o Desenvolvimento de Serviços Cossack

A seguir são apresentados os benefícios de se desenvolver serviços Cossack utilizando uma abordagem baseada nos conceitos da Arquitetura de Software:

- **Base para a Comunicação:** a Arquitetura de Software serve como base para a comunicação entre as diferentes partes envolvidas no projeto de um sistema de software, pois fornece abstrações de alto-nível que permitem que clientes (pessoas que estão comprando o sistema) entendam de forma correta os desenvolvedores. Com isso, uma abordagem baseada em Arquitetura de Software facilitará a comunicação entre os PSI's (desenvolvedores dos serviços) e os comerciantes locais (clientes) [Garlan 2000].
- **Reuso:** um dos grandes benefícios da Arquitetura de Software é a separação entre computação e comunicação. Em Arquitetura de Software, conectores são entidades de primeira-classe (conectores possuem especificação e implementação próprias) [Clements 1996], removendo dos componentes a responsabilidade de saber como eles estão conectados. Portanto, os componentes e conectores não terão interdependências, fazendo com que a prática do reuso seja mais eficaz quando comparada à abordagem orientada a objetos [Medvidovic 1998].
- **Configuração:** conforme apresentado no item anterior, os componentes e conectores em Arquitetura de Software não possuem interdependências, ou seja, eles estão prontos para o uso (off-the-shelf) [Ren 2003]. Com isso, o desenvolvedor precisa apenas escolher os componentes e conectores corretos e configurá-los (conectá-los através de links) para compor um sistema [Kramer 1992].
- **Evolução:** a Arquitetura de Software é uma abordagem ideal para suportar evolução de sistemas, pois os componentes e conectores não possuem interdependências e, portanto, operações de adição, remoção e alteração de componentes ou conectores são realizadas sem alterar o código-fonte [Shaw 1996]. No Sistema Cossack este benefício auxilia a evolução de serviços, uma vez que, os serviços não encontrados em uma rede de PSI's podem ser requisitados a rede de nível hierárquico superior.
- **Especificação, Análise e Implementação:** por serem baseadas em estilos arquiteturais que definem os componentes, conectores e as restrições de um sistema, as ADLs podem fornecer ferramentas de especificação adequadas ao domínio dos sistemas, ferramentas de análise específicas (simuladores, por exemplo) e geradores automáticos de código [Magee 1995].

4. Uma Linguagem de Descrição de Arquitetura e suas Ferramentas para o Desenvolvimento de Serviços Cossack

Nesta seção será apresentada a abordagem que vem sendo criada para o desenvolvimento de serviços para o sistema Cossack. Conforme as razões apresentadas

na subseção 3.5, esta abordagem é baseada nos conceitos de Arquitetura de Software e, é composta de duas partes:

- Uma Linguagem de Descrição de Arquitetura (ADL), chamada xCossack, utilizada para a especificação dos serviços Cossack;
- um conjunto de ferramentas baseadas em xCossack, chamado CossacKit, contendo um ambiente para a especificação gráfica dos serviços, simuladores e geradores automáticos de código.

4.1. xCossack

A ADL xCossack está sendo desenvolvida com base em três diferentes estilos arquiteturais:

- ***Layered-Client-Cache-Stateless-Server (LC\$SS)***: este estilo arquitetural heterogêneo, apresentado por [Fielding 2000], integra ao Estilo Cliente-Servidor as características do Estilo de Camadas (apresentados na subseção 3.1). Com isso, o Estilo LC\$SS permite a especificação de sistemas cliente-servidor onde um servidor que não possui um determinado serviço requisitado por um cliente pode redirecionar esta requisição para um servidor da camada acima. Portanto, utilizando este estilo a ADL xCossack permitirá a especificação de serviços locais Cossack, onde o telespectador requisita um serviço para um PSI local e, se este PSI não prestar o serviço, a requisição do telespectador pode ser direcionada ao PSI regional e depois ao PSI global. A seguir são apresentados exemplos de componentes, conectores e restrições deste estilo:
 - Componentes: Clientes e servidores.
 - Conectores: Chamadas Remotas de Procedimentos (RPC).
 - Restrições: Limitar a comunicação entre determinadas camadas adjacentes.
- **hyHAL**: este estilo arquitetural, definido em [Muchaluat-Saade 2001], é destinado a especificação de aplicações hipermídia. Por isso, será utilizado para a especificação da interface hipermídia de serviços Cossack, representada por banners que são disponibilizados ao telespectador através da televisão. A seguir são apresentados exemplos de componentes e conectores deste estilo:
 - Componentes: Mídias como banners, textos e animações.
 - Conectores: Sincronização temporal entre as mídias.
- **Main Program + Subroutine**: este estilo, utilizado por algumas linguagens de programação existentes, é uma decomposição hierárquica das funcionalidades de um sistema em subrotinas, controladas por um programa principal, onde cada uma resolve uma tarefa específica [Shaw 1996]. A seguir são apresentados exemplos de componentes e conectores deste estilo:
 - Componentes: Procedimentos.
 - Conectores: Chamadas de Procedimentos.

A especificação de um serviço Cossack realizada através da ADL xCossack envolve a utilização destes três estilos. A Figura 4 demonstra a especificação de um serviço utilizando a notação gráfica da xCossack.

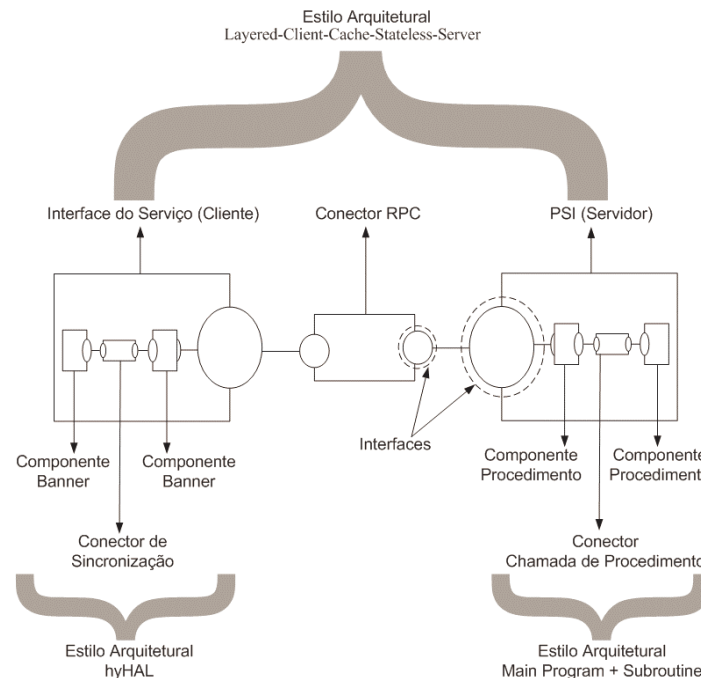


Figura 4: Especificação de um serviço utilizando a notação gráfica da ADL xCossack.

Como xCossack está sendo desenvolvida com base na ADL xADL [Dashofy 2002], apresentada na subseção 3.3, a notação textual da xCossack especifica um serviço utilizando quatro diferentes visões arquiteturais: visão de design, visão de tempo-real, visão de evolução e visão de implementação. A Figura 5 apresenta um exemplo da especificação textual de um componente banner, que faz parte da interface de um serviço, utilizando a visão de design da xCossack:

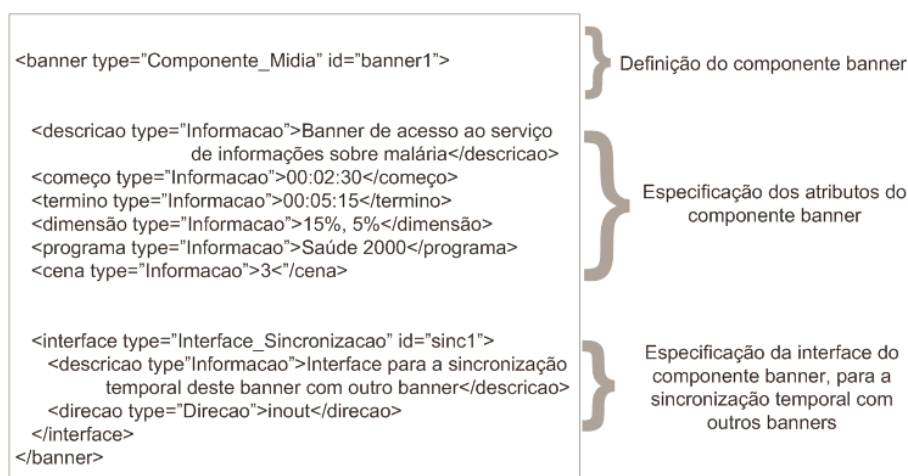


Figura 5: Exemplo da especificação textual de um componente banner usando a ADL xCossack.

4.2. CossacKit

O CossacKit é um ambiente (mostrado na Figura 6) que fornece as seguintes ferramentas para o desenvolvimento de serviços Cossack:

- **Editor-Cossack:** Esta ferramenta, que está sendo desenvolvida com base no software Sofaa [SofAA 2002], permite que os PSI's especifiquem os serviços de forma gráfica (utilizando a notação gráfica mostrada na Figura 4) e utilizando componentes e conectores prontos para o uso. Com isso, os PSI's não precisam especificar um serviço utilizando linhas de código da notação textual de xCossack. Além de facilitar a especificação dos serviços, a especificação gráfica desenvolvida através do Editor-Cossack melhora a comunicação entre os PSI's (com conhecimentos técnicos) e os comerciantes locais (sem conhecimentos técnicos), pois a especificação gráfica desenvolvida no Editor-Cossack é uma abstração de alto-nível de um serviço. O Editor-Cossack deve, a partir da especificação gráfica de um serviço, gerar automaticamente a especificação textual deste.
- **Simula-Cossack:** A partir da especificação textual de um serviço, gerada automaticamente pela Editor-Cossack, a ferramenta Simula-Cossack faz simulações da execução de um serviço, detectando, por exemplo, problemas de Qualidade de Serviço (QoS) [Sztajnberg 2000].
- **Code-Cossack:** A partir da especificação textual de um serviço, gerada automaticamente pela Editor-Cossack, a ferramenta Code-Cossack gera código executável deste serviço.

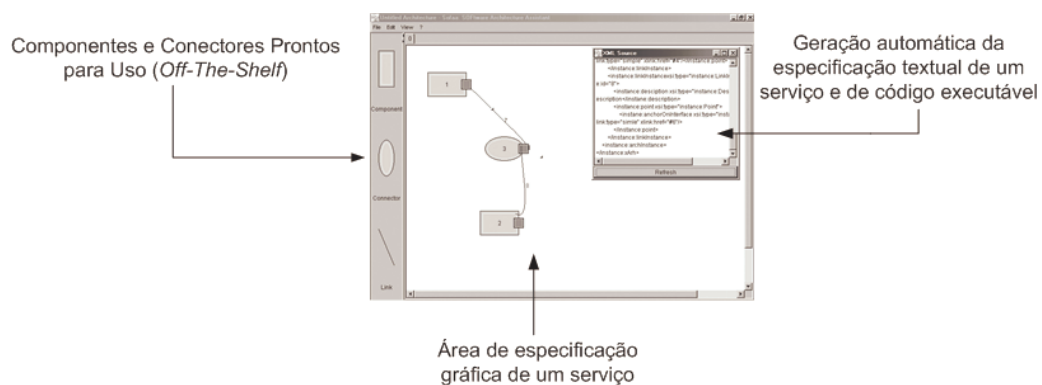


Figura 6: Tela principal do CossacKit.

5. Conclusão

Este artigo apresentou uma abordagem para o desenvolvimento de serviços do Sistema Cossack. Este Sistema de televisão interativa pode vir a ser uma solução para o problema da exclusão digital brasileira, já que oferece serviços através da televisão. Estes serviços podem ser prestados de forma local, preservando desta forma as particularidades de cada região. Para o desenvolvimento destes serviços foi proposta uma abordagem baseada nos conceitos de Arquitetura de Software. Esta abordagem é composta de uma linguagem para a especificação de serviços, chamada xCossack, e um ambiente de desenvolvimento chamado CossacKit.

Utilizando o CossaKit os Provedores de Serviços Interativos (PSI's) poderão especificar os serviços de acordo com a linguagem xCossack, e fazer simulações da execução dos serviços e gerar automaticamente, a partir da especificação, código executável dos serviços. A ADL xCossack e o CossackKit serão disponibilizados gratuitamente, com o código aberto, ou seja, além de não acarretar em custos adicionais, os PSI's poderão estender as funcionalidades da xCossack e do CossackKit para o desenvolvimento de novos serviços.

6. Agradecimentos

Os autores Giuliano Araujo Bertoti e Davi D'Andréa Baccan gostariam de agradecer, respectivamente, as agências financiadoras CAPES e CNPq.

Referências

- Almeida, Felipe Afonso. Cossack project, Scientific Final Report. Technological Institute of Aeronautics, Computer Science Division, 1999.
- Almeida, F. A., Fortes, R. S., Baccan, D. D., "The Cossack System: A Platform for Interactive TV", 2nd European Conference on Interactive Television: Enhancing the Experience, Brighton, UK, in prelo, 2004.
- Bray, T., Paoli, J., Sperberg, C. M. M., Maler, E., "Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 6 October 2000.
- Clements, P. C., "A Survey of Architecture Description Languages", Proceedings of the 8th International Workshop on Software Specification and Design, 1996.
- Consulting, S., "iTV Hanbook: Technologies and Standards", Prentice Hall, 1ª Edição – 2003.
- Dashofy, Eric M., Hoek, A., Taylor, R. N., "A Highly-Extensible, XML-Based Architecture Description Language", In Proceedings of the Working IEEE/IFIP Conference on Software Architectures (WICSA 2001), Amsterdam, Netherlands.
- Dashofy, Eric M., Hoek, A., Taylor, R. N., "An Infrastructure for the Rapid Development of XML-based Architecture Description Languages", Proceedings of the 24th international conference on Software engineering, Orlando, Florida, 2002.
- DIRECTV, <http://www.directv.com.br/>, 2004.
- Fielding, Roy T., "Architectural Styles and the Design of Network-based Software Architectures", University of California, Irvine, 2000.
- Garlan, David, "Software Architecture: a Roadmap", ICSE 2000, 22nd International Conference on Software Engineering, Future of Software Engineering Track, June 4-11, 2000, Limerick Ireland, ACM, 2000.
- Horstmann, C. S., "Object-Oriented Design and Patterns", IE-WILEY, 1ª Edição - 2003.
- IBGE, Instituto Brasileiro de Geografia e Estatística, "Censo Demográfico 2000", disponível em <http://www.ibge.com.br/>, 2004.
- Kerschbaumer, A., "Behavioral Refinement of Software Architectures", Technischen Universit. at Graz, August 2002.

- Kramer, J., "Configuration Programming - Exploiting Component Reuse in Distributed Systems", in 'Software Reuse and Reverse Engineering in Practice', ed. P. Hall, (Chapman and Hall and UNICOM), 1992, 415-442.
- Kruchten, Philippe B., "The 4+1 View Model of Architecture", IEEE Computer Society Press, 1995.
- Launchalot, "iTV authoring with Lois", <http://www.launchalot.com>, 2004.
- Magee, J., Dulay, N., Eisenbach, S., Kramer, J., "Specifying Distributed Software Architectures", In Proceedings of 5th European Software Engineering Conference (ESEC 95), Sitges, Spain, September 1995.
- FGV, Fundação Getúlio Vargas, "Mapa da Exclusão Digital", disponível em: http://www2.fgv.br/ibre/cps/mapa_exclusao/apresentacao/apresentacao.htm, Abril, 2003.
- Medvidovic, N., "Using Object-Oriented Typing to Support Architectural Design in the C2 Style", In Proceedings of the ACM SIGSOFT '96 Fourth Symposium on the Foundations of Software Engineering. p.24-32, ACM SIGSOFT. San Francisco, CA, October, 1996.
- Medvidovic, N., "A Framework for Classifying and Comparing Architecture Description Languages", Proceedings of the Sixth European Software Engineering Conference (ESEC/FSE 97), 1997.
- Medvidovic, N., Taylor, R. N., "Separating Fact from Fiction in Software Architecture." In Proceedings of the Third International Software Architecture Workshop (ISAW-3), pages 105-108, Orlando, FL, November 1-2, 1998.
- Muchaluat-Saade, D.C., Soares, L.F.G., "Towards the Convergence Between Hypermedia Authoring Languages and Architectural Description Languages", I ACM Symposium on Document Engineering - DocEng2001 Atlanta, USA – November, 2001.
- OpenTV, <http://www.opentv.com/solutions/developers/itvtools.html>, 2004.
- Perry, D. E., Wolf, A. L., "Foundations for the study of software architecture", ACM SIGSOFT Software Engineering Notes, v.17 n.4, p.40-52, Oct. 1992.
- Ren, J., Taylor, R. N., "Visualizing Software Architecture with Off-The-Shelf Components", Proceedings of the 15th International Conference on Software Engineering & Knowledge Engineering, pp.132-141, San Francisco, California, USA, July 1-3, 2003.
- Sztajnberg, A., Loques, O., "Bringing QoS to the Architectural Level", accepted for presentation, ECOOP 2000 Workshop on QoS on Distributed Object Systems, Cannes, France, June, 2000.
- Shaw, Mary, Garlan, David, "Software Architecture: Perspectives on an Emerging Discipline", Prentice Hall, 1996.
- SKY, <http://www.skyweb.com.br/>, 2004.
- SofAA (Sof[tware] A[rchitecture] A[ssistant]), "A graphical tool designed to create/edit/view software architecture descriptions", <http://sofaa.sourceforge.net/>, 2002.