

Tecnologias para Processamento no Lado do Cliente WWW



PUC-Rio / DI

TeleMídia

Plug-ins



PUC-Rio / DI

TeleMídia

Plug-ins

- ➔ Principal utilidade: exibir conteúdo cujo formato não é tratado pelo browser
 - conteúdos específicos das aplicações (PDF, PostScript, etc.), áudio, vídeo
- ➔ Permite também que aplicações existentes sejam facilmente integradas à Web (exemplo: plugin Gingga-NCL)



PUC-Rio / DI

TeleMídia

Plug-ins

- ➔ Tecnologia originalmente projetada pela Netscape
 - Netscape Navigator 2.0
 - Internet Explorer 3.0 passou também a oferecer suporte
- ➔ Módulo de código separado que se comporta como se fosse parte do browser
 - associado a um ou mais tipos de mídia (tipo MIME)
 - biblioteca de código nativo C
 - específico a uma plataforma (sistema operacional)
 - dependente da interface de programação do browser



PUC-Rio / DI

TeleMídia

Como inserir plug-ins em páginas HTML

- ➔ Elementos HTML utilizados para inserção de plug-ins
 - *OBJECT*
 - *EMBED* (não faz parte da especificação HTML 4.01)
- ➔ Modos de exibição de um plug-in
 - embutido
 - escondido
 - página inteira



PUC-Rio / DI

TeleMídia

Como inserir plug-ins em páginas HTML

- ➔ Elemento *OBJECT*
 - quando o browser não sabe tratar a especificação, o conteúdo do elemento deve ser apresentado
 - *Objects* podem ser aninhados

```
<object data="clock.avi" type="video/msvideo" height="100%" width="100%"  
classid="http://microsoft.com/plugins/" >  
  <object data="clock.gif" type="image/gif">  
  <p>Hora certa.  
  </object>  
</object>
```



PUC-Rio / DI

TeleMídia

Como inserir plug-ins em páginas HTML

➔ Principais atributos do elemento *OBJECT*

- *data*: URI do conteúdo a ser tratado pelo plug-in
- *type*: tipo MIME do dado
- *classid*: URI de onde encontrar um plug-in a ser baixado caso o browser não tenha o plug-in instalado
- *height* e *width*: tamanho da área para exibição
- *name* ou *id*: identificação para o plug-in



PUC-Rio / DI

TeleMídia

Elemento Object

➔ Container genérico para embutir conteúdo externo em páginas HTML (não se restringe a plug-ins)

- imagens, applets, plug-ins, documentos html embutidos, etc.
- deve substituir o elemento APPLET do HTML
- forma genérica para inclusão de imagens (elemento IMG) e outros documentos HTML (elemento IFRAME)

➔ Parâmetros podem ser passados para um OBJECT através do uso do elemento PARAM

- `<param name="nome_parâmetro" value="valor" type="mime_content_type" valuetype="tipo">`



PUC-Rio / DI

TeleMídia

Plug-ins - Modelo de Execução

- ➔ Plug-ins executam no mesmo espaço de memória do browser
 - DLLs, objetos compartilhados, bibliotecas compartilhadas, etc.
- ➔ Ciclo de vida de um plug-in está associado ao ciclo de vida da página que o aciona



Plug-ins - Modelo de Execução

- ➔ Quando o browser encontra em uma página uma referência (URI) para um arquivo que está associado a um Plug-in
 - browser carrega o código do plug-in na memória (se ainda não tiver feito)
 - cria uma nova instância do plug-in (o browser pode criar várias instâncias de um mesmo plug-in simultaneamente)



Plug-ins - Modelo de Execução

- ➔ Quando o browser sai da página que contém a referência para o plug-in ou tem sua janela fechada, a instância do plug-in é removida da memória
 - quando a última instância de um plug-in é removida, o código do plug-in é retirado da memória
- ➔ Quando um plug-in não está carregado em memória, o mesmo só ocupa espaço em disco



PUC-Rio / DI

TeleMídia

Plug-ins

- ➔ Dependentes de plataforma e browser
- ➔ Não permite interagir diretamente com o conteúdo HTML
 - substituir imagens (simular animações)
 - simular menus de opções
 - mudar características de apresentação do documento de acordo com a interação do usuário
 - acrescentar conteúdo dinamicamente



PUC-Rio / DI

TeleMídia

Scripts



PUC-Rio / DI

TeleMídia

Scripts

- ➔ Utilização de uma linguagem de script para adicionar funcionalidades dinâmicas a páginas HTML estáticas
- ➔ Página HTML carrega (de forma embutida ou através de uma referência) scripts que serão executados pelo browser
 - alterar a especificação de apresentação dos elementos
 - acrescentar conteúdo dinamicamente ao documento
 - verificar a entrada de dados em um formulário
 - controlar o browser



PUC-Rio / DI

TeleMídia

Linguagens de Script

➤ Principais linguagens de script utilizadas

- JavaScript (inicialmente chamado de LiveScript) - Netscape
- Jscript e VBScript - Microsoft
- Tcl

➤ Padrão para linguagens de script interpretadas no cliente

- ECMAScript (European Computer Manufacturers Association Script)
 - padrão de junho de 1997
 - JavaScript e JScript são implementações do padrão



PUC-Rio / DI

TeleMídia

ECMA Script

➤ Padrão para linguagens de script interpretadas no cliente

➤ Linguagem padrão dos middlewares declarativos

- DVB-HTML
- ACAP-X
- BML

➤ Linguagem também suportada no Ginga-NCL, onde a linguagem script padrão é Lua.



PUC-Rio / DI

TeleMídia

Por que Linguagens de Script?

- ➔ Interpretadas (não exigem compilação)
 - independência de plataforma
- ➔ Simples de programar
 - mais adequadas para usuários não experts em programação
 - ideais para tarefas simples
- ➔ Desvantagens
 - Ineficiência (interpretadas)
 - menos poderosas e por isso são indicadas apenas para tarefas não muito complexas

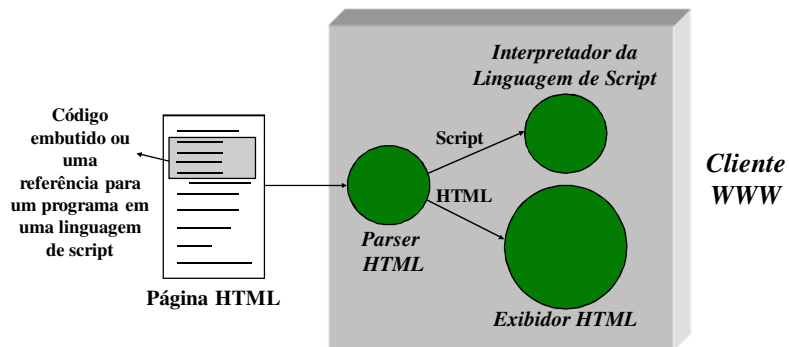


PUC-Rio / DI

TeleMídia

Scripts

- ➔ Para executar os scripts, o cliente WWW (browser) precisa de um interpretador da linguagem de script utilizada no documento



PUC-Rio / DI

TeleMídia

Como inserir Scripts em páginas HTML

➔ Elemento *SCRIPT*

- *src*: URI para o script propriamente dito
 - se não for definido, o conteúdo do elemento será tratado como sendo o código
- *type*: especifica a linguagem de script através de um tipo MIME
 - ex: text/javascript
- *charset*: apenas considerado quando o atributo *src* for definido
- *language*: desaconselhado, substituído pelo atributo *type*



PUC-Rio / DI

TeleMídia

Como inserir Scripts em páginas HTML

```
<p>Última atualização feita em:  
<script type="text/javascript">  
<!--  
document.write(document.lastModified);  
-->  
</script>
```

JavaScript

Comentários do HTML:
para evitar que browsers
que não dão suporte a
scripts exibam o conteúdo
do script na tela

Informa a data da última modificação do documento



PUC-Rio / DI

TeleMídia

Como inserir Scripts em páginas HTML

- ➔ Elemento pode aparecer várias vezes, tanto no *Head* como no *Body* do documento HTML
- ➔ Fica a cargo de cada linguagem de script oferecer uma sintaxe para referenciar elementos HTML no documento



PUC-Rio / DI

TeleMídia

Como inserir Scripts em páginas HTML

- ➔ Eventos intrínsecos: atributos de elementos HTML que permitem a definição de scripts para tratar eventos na apresentação do documento
 - Body e Frameset
 - onload, onunload
 - A, Area, Label, Input, Select, Textarea e Button
 - onfocus, onblur
 - Form
 - onsubmit, onreset



PUC-Rio / DI

TeleMídia

Como inserir Scripts em páginas HTML

➔ Eventos intrínsecos (continuação)

- Input e Textarea
 - onselect
- Input, Textarea e Select
 - onchange
- com a maioria dos elementos HTML
 - onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onkeypress, onkeydown, onkeyup



PUC-Rio / DI

TeleMídia

Exemplo de Script Associado a Eventos HTML

```
<form>  
<input type="button" value="Aperte aqui!"  
      onclick="alert('Você clicou no botão!')">  
</form>
```

- ➔ Exibe uma mensagem quando o usuário pressiona o botão
- ➔ Os eventos intrínsecos normalmente são utilizados em conjunto com funções declaradas na área de SCRIPT



PUC-Rio / DI

TeleMídia

Exemplo de Script Associado a Eventos HTML

```
<head>
<script type="text/javascript">
function my_onclick() {
...
}
</script>
</head>
<body>
...
<form>
<button type="button" name="mybutton" onclick="my_onclick()">
mybutton
</button>
</form>
</body>
```



PUC-Rio / DI

TeleMídia

Scripts (Exemplo DHTML)

- ➔ Sublinha a âncora só quando o mouse estiver sobre a mesma

```
<html><head>
<style>
.on {text-decoration:underline; color:blue;}
.off {text-decoration:none; color:black;}
</style>
</head><body>
<ul>
<li> <a href="apresentacoes.html" class="off"
onmouseover="this.className='on';"
onmouseout="this.className='off';">Apresentações</a>
<li> <a href="artigos.html" class="off"
onmouseover="this.className='on';"
onmouseout="this.className='off';">Artigos</a>
</ul>
</body></html>
```



PUC-Rio / DI

TeleMídia

O que pode ser feito com Scripts

- Acrescentar dinamicamente conteúdo a uma página HTML enquanto a página está sendo processada pelo browser
- Alterar o conteúdo de alguns elementos HTML
- Alterar características de apresentação dos elementos
- Obter a lista de elos do documento



PUC-Rio / DI

TeleMídia

O que pode ser feito com Scripts

- Controlar o comportamento do browser
 - exibir mensagens através de dialog boxes ou na linha de status do browser
 - abrir e fechar novas janelas do browser
 - navegar no histórico de documentos simulando o comportamento dos botões back e forward do browser
- Ler e escrever valores em elementos para entradas de dados em qualquer formulário do documento
- Ler e escrever em Cookies



PUC-Rio / DI

TeleMídia

O que NÃO pode ser feito com Scripts

- Linguagens não possuem recursos gráficos
- Por razões de segurança, não se tem acesso para ler ou escrever em arquivos no cliente
- Não oferece suporte para comunicação em rede a não ser baixar o conteúdo de uma URL
- Não tem capacidade para abrir múltiplas threads



PUC-Rio / DI

TeleMídia

ECMAScript em iTV

- Muitas das aplicações para TV digital baseadas em ECMAScript fazem parte do contexto de operação de DHTML (*Dynamic HTML*), permitindo converter uma página HTML estática em uma aplicação interativa.



PUC-Rio / DI

TeleMídia

Linguagens de Sistemas



PUC-Rio / DI

TeleMídia

Linguagens de Sistema

- ▶ Outra possibilidade para introduzir dinamismo em documentos na Web é através de uma linguagem de sistema
 - programas precisam ser distribuídos e executados em uma variedade de plataformas (clientes WWW)
 - programa (compilado) precisa ser independente de plataforma
- ▶ Java
 - linguagem orientada a objetos, independente de plataforma
 - desenvolvida pela Sun Microsystems

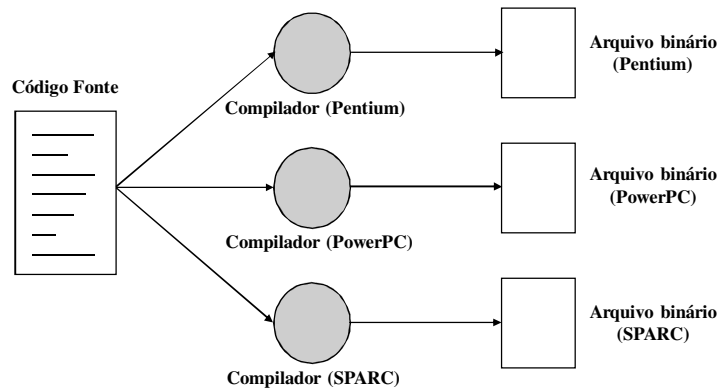


PUC-Rio / DI

TeleMídia

Linguagens de Sistema

➔ Programas compilados tradicionais (dependentes de plataforma)



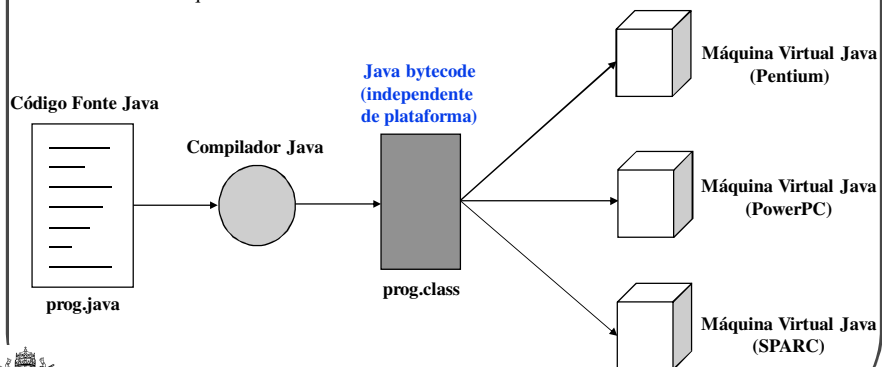
PUC-Rio / DI

TeleMídia

Linguagens de Sistema

➔ Programas Java (independentes de plataforma)

- desvantagem: desempenho => bytecode precisa ser interpretado pela máquina virtual Java



PUC-Rio / DI

TeleMídia

Linguagens de Sistema

➤ O que pode ser feito com Java?

- programas complexos
- tem recursos gráficos na linguagem
- suporta comunicação em rede
- tem capacidade para múltiplas threads

➤ Tipos de programas:

- aplicação stand-alone
- **applet Java - executados localmente no cliente WWW**



PUC-Rio / DI

TeleMídia

Applets Java



PUC-Rio / DI

TeleMídia

Applets Java

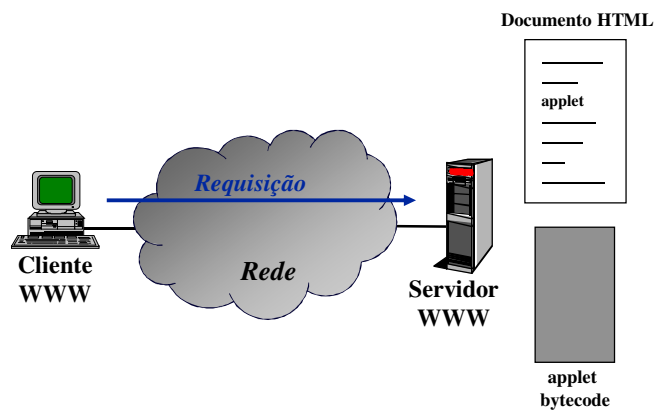
- ➔ Programas que podem ser baixados de qualquer servidor WWW
- ➔ Executados localmente em qualquer cliente WWW que saiba executar código Java
 - browsers possuem uma máquina virtual Java (JVM) embutida ou implementada como um plug-in
- ➔ Quando o browser carrega uma página Web que faz referência a um applet, ele traz o applet de um servidor WWW (bytecode) e o executa localmente



PUC-Rio / DI

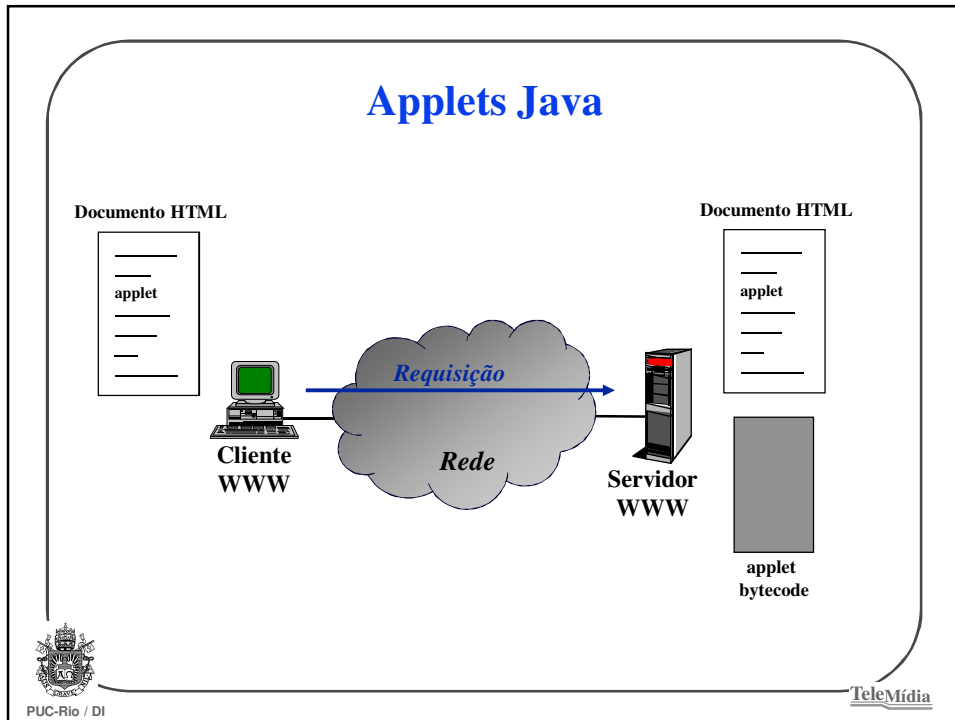
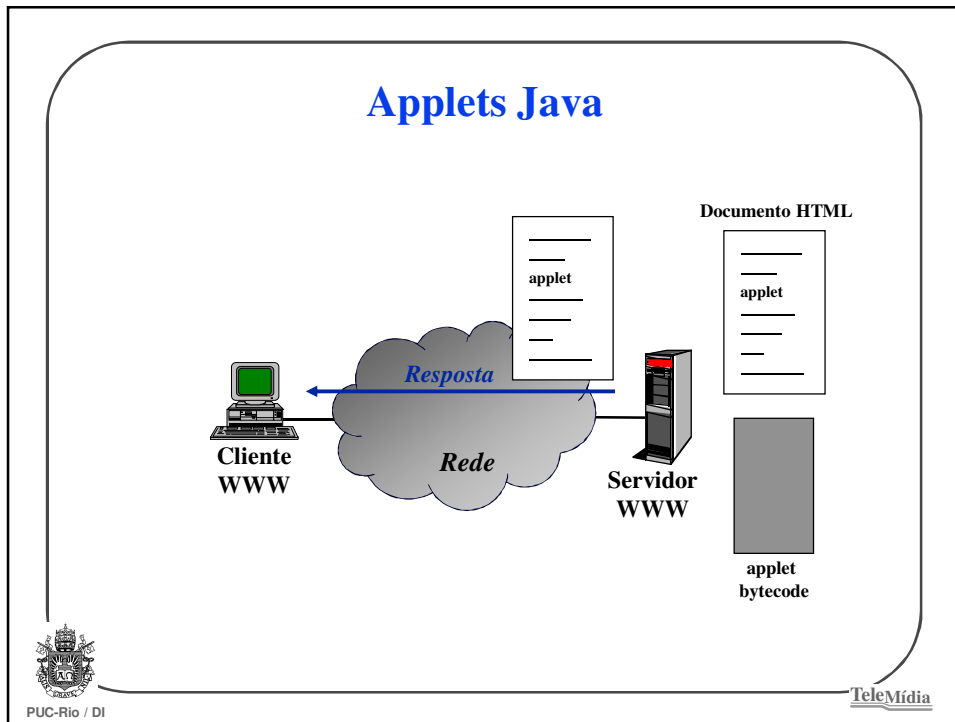
TeleMídia

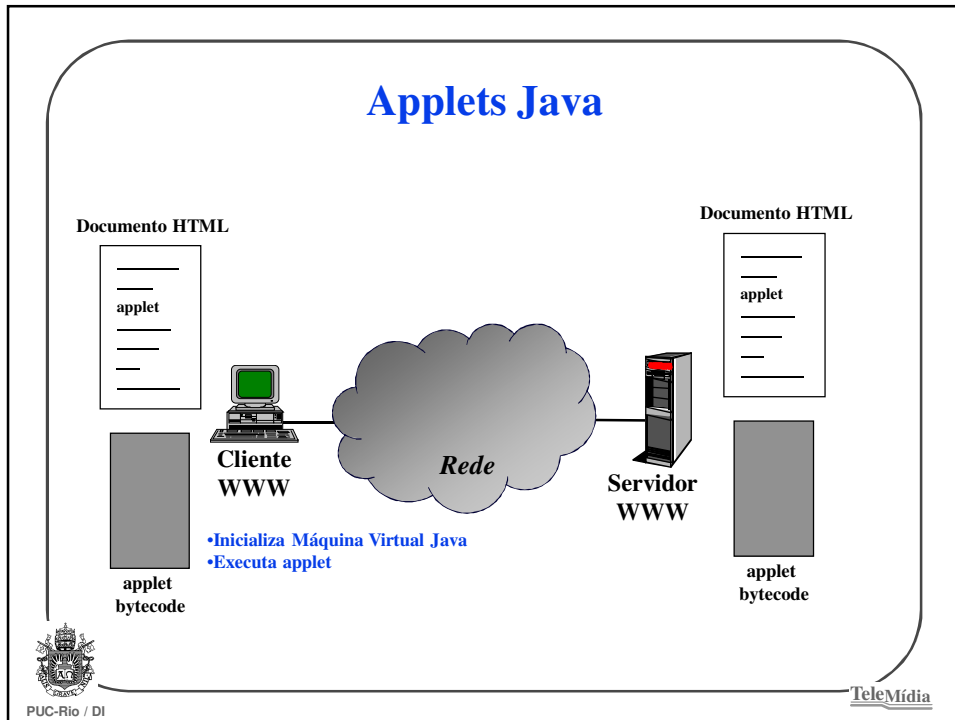
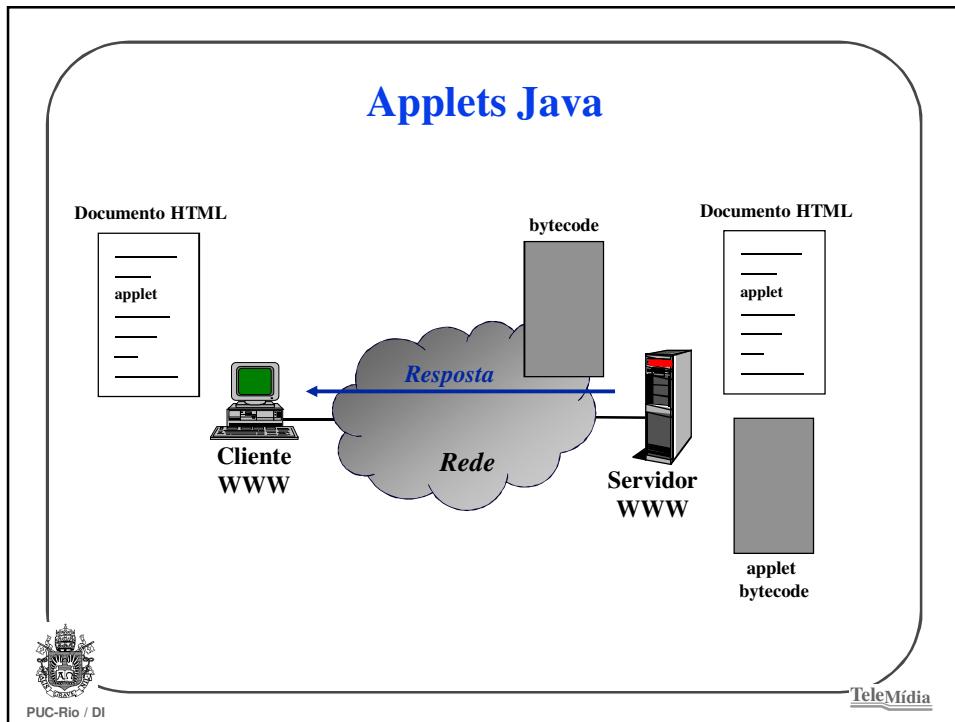
Applets Java



PUC-Rio / DI

TeleMídia





O que é um Applet

➔ Uma classe Java que estende a classe `Applet`

- pacote (package) `java.applet`

➔ Principais métodos definidos pela classe

- `public void init()`
- `public void start()`
- `public void stop()`
- `public void destroy()`
- `public void paint(Graphics g)`
- `public String getParameter(String name)`



PUC-Rio / DI

TeleMídia

Applet Java

```
import java.applet.*;

public class HelloAgain extends Applet {
    public void init() {
        System.out.println("init - Hello Again");
    }

    public void start() {
        System.out.println("start - Hello Again");
    }

    public void stop() {
        System.out.println("stop - Hello Again");
    }

    public void destroy() {
        System.out.println("destroy - Hello Again");
    }
}
```



PUC-Rio / DI

TeleMídia

Applet Java

```
import java.applet.*;
import java.awt.*;

public class HelloAgain extends Applet {
    private int cont;

    public void init() {
        cont= 0;}

    public void start() {
        cont= cont + 1;}

    public void paint(Graphics g) {
        g.drawString("Hello again: " + cont + "!", 5, 40);}
}
```



PUC-Rio / DI

TeleMídia

Como inserir Applets em páginas HTML

➔ Elemento *OBJECT*

- *codebase*: URI base para localizar o applet
- *codetype*: tipo MIME do objeto (“application/java”)
- *classid*: localização do applet (“java:<nome_classe>”)
- *height* e *width*: tamanho da área para exibição
- *name* ou *id*: nome de identificação para o applet



PUC-Rio / DI

TeleMídia

Como inserir Applets em páginas HTML

```
<object codetype="application/java"  
classid="java:HelloAgain" width="100" height="100">  
Texto caso o browser não trate applets.  
</object>
```

```
<object codetype="application/java"  
codebase="/applets/"  
classid="java:HelloAgain" width="100" height="100">  
Texto caso o browser não trate applets.  
</object>
```



PUC-Rio / DI

TeleMídia

Applets e Parâmetros

```
<object codetype="application/java"  
classid="java:HelloAgain" width="100" height="100">  
<param name="user" value="LFGS">  
Texto caso o browser não trate applets.  
</object>
```



PUC-Rio / DI

TeleMídia

O que NÃO é permitido com Applets Java

- ▶ Ler ou escrever no sistema de arquivos local do cliente
- ▶ Rodar programas executáveis no cliente
- ▶ Efetuar chamadas a métodos nativos
- ▶ Comunicar-se com um servidor WWW que não seja o servidor de onde foi feito download do applet (*originating host*)
- ▶ Consultar propriedades do sistema, exceto a versão da máquina virtual Java e nome/versão do sistema operacional



PUC-Rio / DI

TeleMídia

Integração das Tecnologias

- ▶ Como fazer a integração entre elementos HTML, scripts, applets e plug-ins?
 - depende do browser
 - Netscape => *LiveConnect*
 - Internet Explorer => *ActiveX*
- ▶ Permitem:
 - chamar métodos de applets e plug-ins a partir de JavaScript
 - chamar funções JavaScript a partir de Java
 - chamar métodos Java a partir de plug-ins



PUC-Rio / DI

TeleMídia

Xlets Java



PUC-Rio / DI

TeleMídia

Xlet

- ▶ A API JavaTV introduz o conceito de *xlets*, que é em linhas gerais bastante similar a *applets*. Assim como *applets*, *xlets* têm seu ciclo de vida controlado por algum elemento presente no ambiente de execução. No caso de *applets*, esse ambiente é tipicamente um navegador. No caso de *xlets*, há um *gerente de aplicações* residente no terminal de acesso.
- ▶ No entanto, há duas diferenças principais entre *applets* e *xlets*.
 - Em primeiro lugar, *xlets* podem ser suspensos (*paused*) e reiniciados (*resumed*). Como em um terminal de acesso várias aplicações podem estar executando concorrentemente, mas em geral somente uma delas pode estar visível em um dado instante, a suspensão de uma aplicação que não esteja visível permite que ela não consuma recursos do terminal de acesso, que podem então ser utilizados pela aplicação visível.
 - A segunda diferença é que *xlets* têm uma capacidade de interação com o seu ambiente de execução muito mais limitada que *applets*. No entanto, tais interações podem ser conseguidas indiretamente por meio de outras APIs.



PUC-Rio / DI

TeleMídia

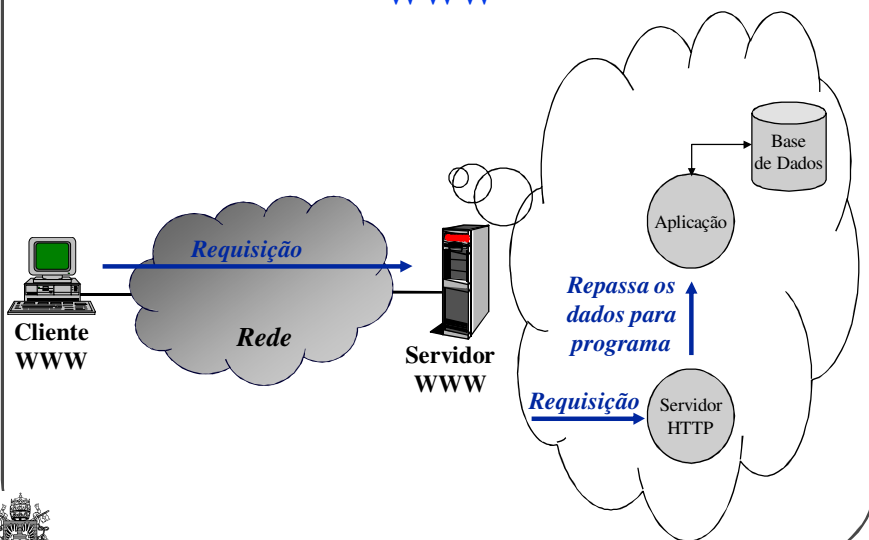
Geração de páginas dinâmicas em Servidores WWW



PUC-Rio / DI

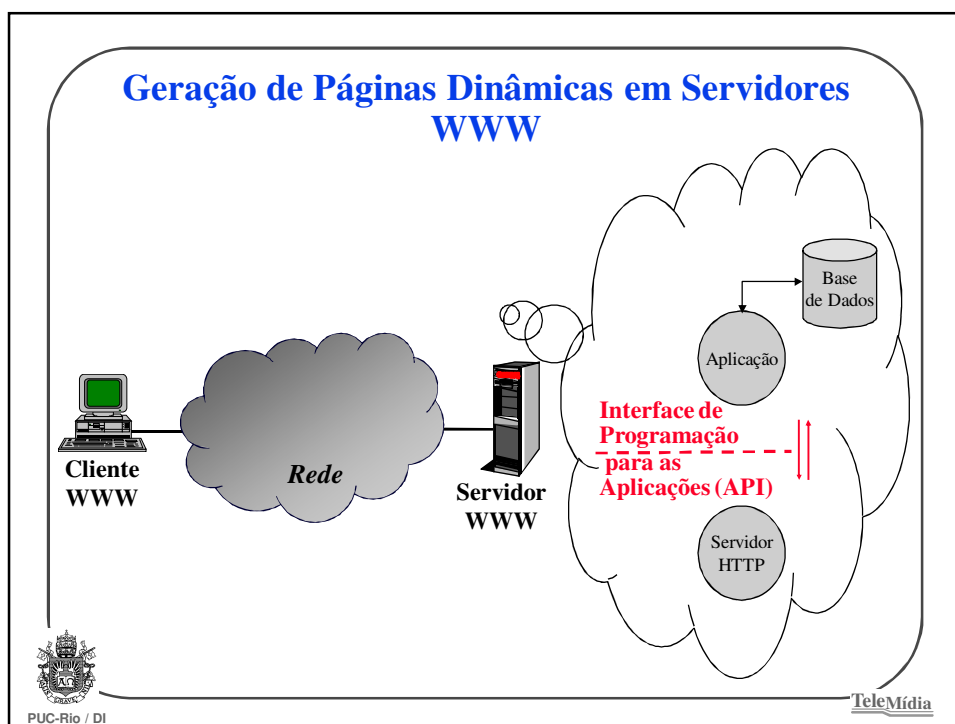
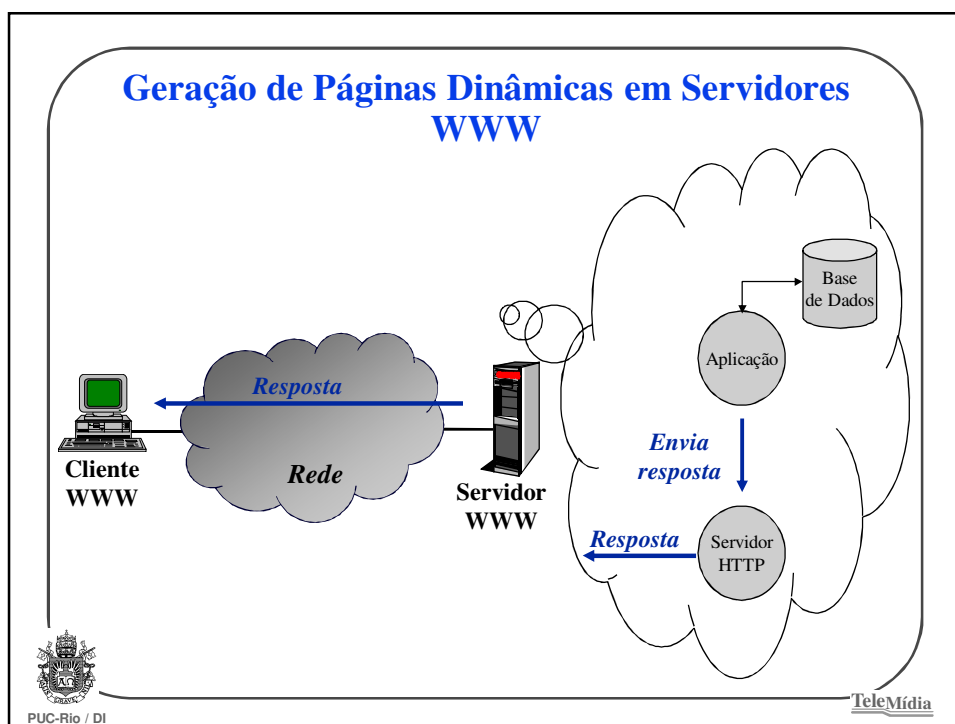
TeleMídia

Geração de Páginas Dinâmicas em Servidores WWW



PUC-Rio / DI

TeleMídia



Servidores WWW e APIs

▶▶ Padronizadas

- CGI (*Common Gateway Interface*)

▶▶ Proprietárias

- Apache API - Apache
- ISAPI (Internet Server API) - Microsoft
- NSAPI (Netscape Server API) - Netscape



PUC-Rio / DI

TeleMídia

Servidores WWW e APIs

▶▶ Padronizadas

- conferem alto grau de portabilidade às aplicações

▶▶ Proprietárias

- maior controle sobre o comportamento interno do servidor
- melhor desempenho
- portabilidade comprometida



PUC-Rio / DI

TeleMídia

Aplicações WWW no Servidor

➔ Objetivos

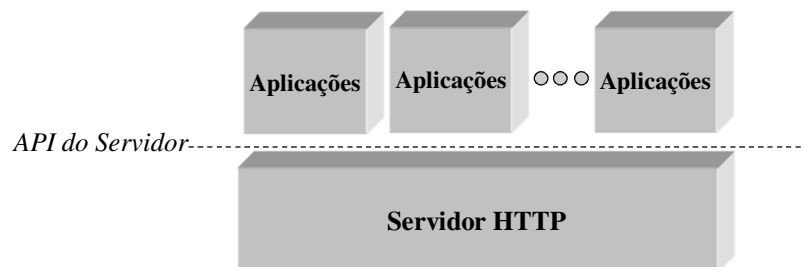
- *geração de páginas dinâmicas*
- outras funções que estendam a funcionalidade do servidor: autenticação, mapeamento de URI, log, etc.



PUC-Rio / DI

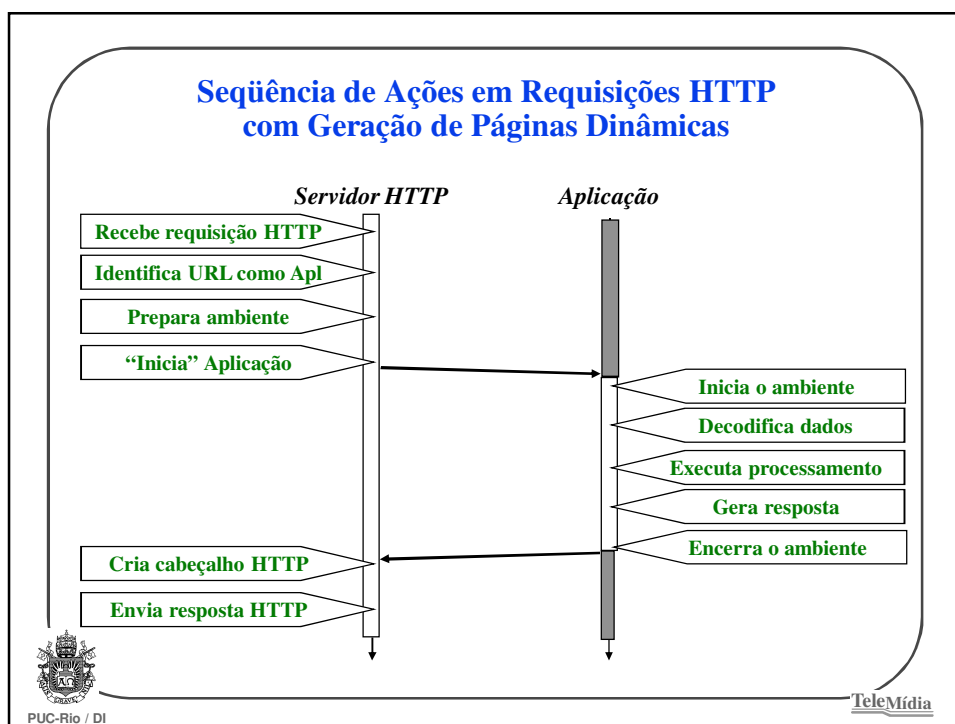
TeleMídia

Aplicações em Servidores WWW



PUC-Rio / DI

TeleMídia



Aplicações WWW no Servidor

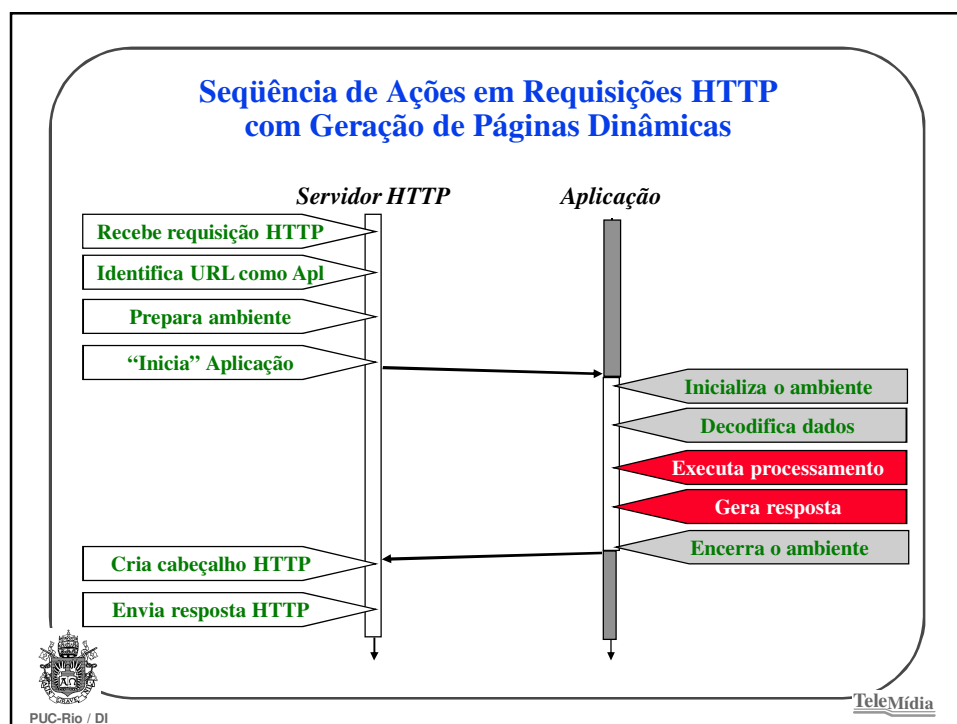
➔ Diversas tarefas comuns podem ser identificadas nas aplicações para geração de páginas dinâmicas

- iniciação do ambiente e liberação dos recursos
- tratamento (parser) dos dados de entrada

`nome=James+Bond&email=bond%40telemidia%2Epuc-rio%2Ebr`

- separação em tuplas (chave, valor)
- tratamento da codificação de caracteres de escape

PUC-Rio / DI TeleMídia



Aplicações WWW no Servidor

➔ Diversas tarefas comuns podem ser identificadas nas aplicações para geração de páginas dinâmicas

- inicialização do ambiente e liberação dos recursos

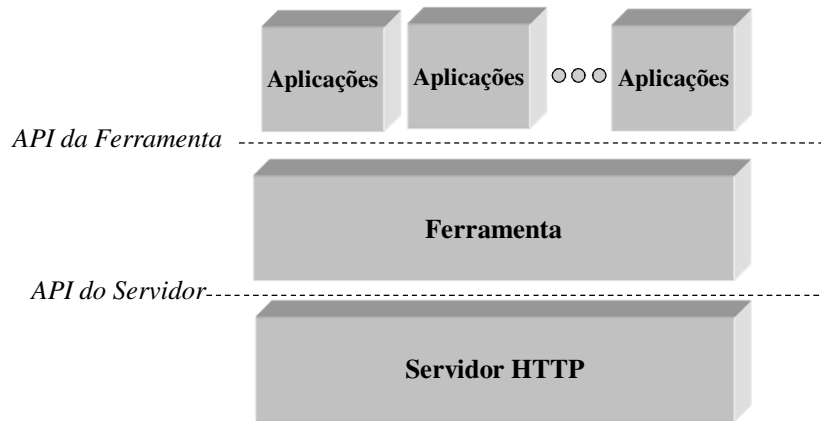
```
nome=James+Bond&email=bond%40telemidia%2Epuc-rio%2Ebr
```

- separação em tuplas (chave, valor)
- tratamento da codificação de caracteres de escape

➔ Implementação de *ferramentas* para suporte a tais funções

PUC-Rio / DI TeleMídia

Aplicações em Servidores WWW



PUC-Rio / DI

TeleMídia

Paradigmas para Programação de Páginas Dinâmicas

➔ Programa

➔ Template



PUC-Rio / DI

TeleMídia

Paradigmas para Programação de Páginas Dinâmicas

Programa

```
write("<HTML>");
write("<HEAD><TITLE>Script Hora</TITLE></HEAD>");
write("<BODY>");
write("<H1>Hora: " + getTime() + "</H1>");
write("</BODY>");
write("</HTML>");
```

Template

```
<HTML>
<HEAD><TITLE>Script Hora</TITLE></HEAD>
<BODY>
<H1>Hora:
$%| getTime() |%$
</H1>
</BODY>
</HTML>
```



PUC-Rio / DI

TeleMídia

Paradigmas para Programação de Páginas Dinâmicas

➔ Programa

- aplicações/scripts escritos através de programas completos
- forma normalmente oferecida em ferramentas baseadas em C/C++ e Perl

➔ Template

- página dinâmica é baseada em uma versão estática da página (em geral, HTML)
- marcações especiais no documento HTML delimitam comandos a serem processados dinamicamente
- paradigma tornou-se popular com ASP (CGILua e JSP também oferecem esse paradigma)



PUC-Rio / DI

TeleMídia

Paradigmas para Programação de Páginas Dinâmicas

➔ Programa

- Vantagens
 - maior expressividade
 - estilo de programação convencional
- Desvantagens
 - opera em um baixo nível de abstração
 - difícil para não programadores



PUC-Rio / DI

TeleMídia

Paradigmas para Programação de Páginas Dinâmicas

➔ Template

- Vantagens
 - permite uma separação entre o conteúdo estático e a lógica de programação
 - programador visual x programador da aplicação
 - permite utilizar editores HTML convencionais, deixando os pontos de processamento indicados
 - conveniente para não programadores
 - adequado a programadores quando existe muito código HTML envolvido na geração da página
- Desvantagens
 - dificulta o reuso



PUC-Rio / DI

TeleMídia

Paradigmas para Programação de Páginas Dinâmicas



Ideal

Que as ferramentas ofereçam suporte aos dois paradigmas



PUC-Rio / DI

TeleMídia

Ferramentas de Autoria



PUC-Rio / DI

TeleMídia

Ferramentas de Autoria SMIL



PUC-Rio / DI

TeleMídia

GRiNS

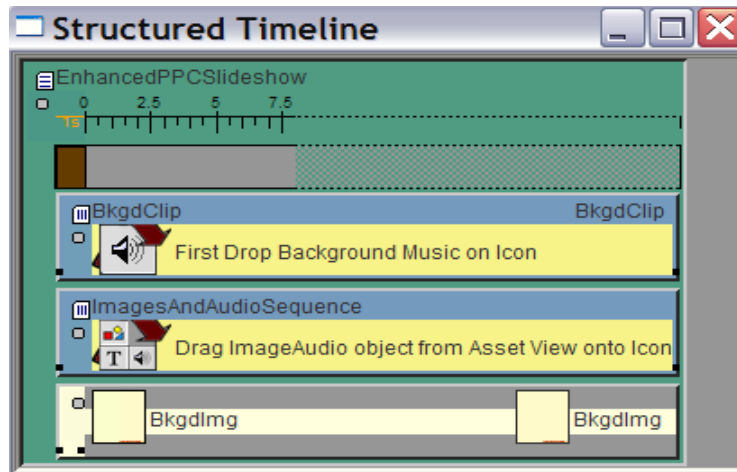
- ➔ Possui várias visões integradas (temporal, espacial e textual)
- ➔ Para que as modificações feitas possam ser aplicadas nas outras visões, é necessário que o usuário clique no botão *Apply*. Caso haja algum erro, o usuário é prontamente informado, e, se optar por desfazer as alterações, deve clicar no botão *Revert*.
- ➔ A existência de um *player* (exibidor) associado à ferramenta de autoria é de grande valia, pois, em tempo de criação, é possível ter noção de como está ficando a apresentação.
- ➔ Não há suporte gráfico para interatividade



PUC-Rio / DI

TeleMídia

GRiNS



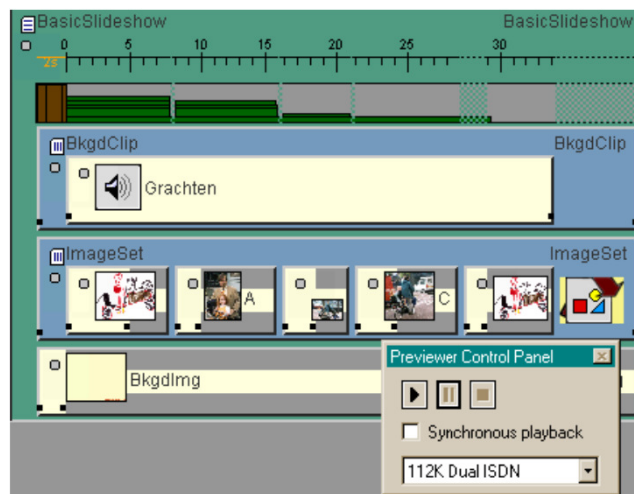
a green <par> (parallel) group and two blue <seq> (sequential) groups.



PUC-Rio / DI

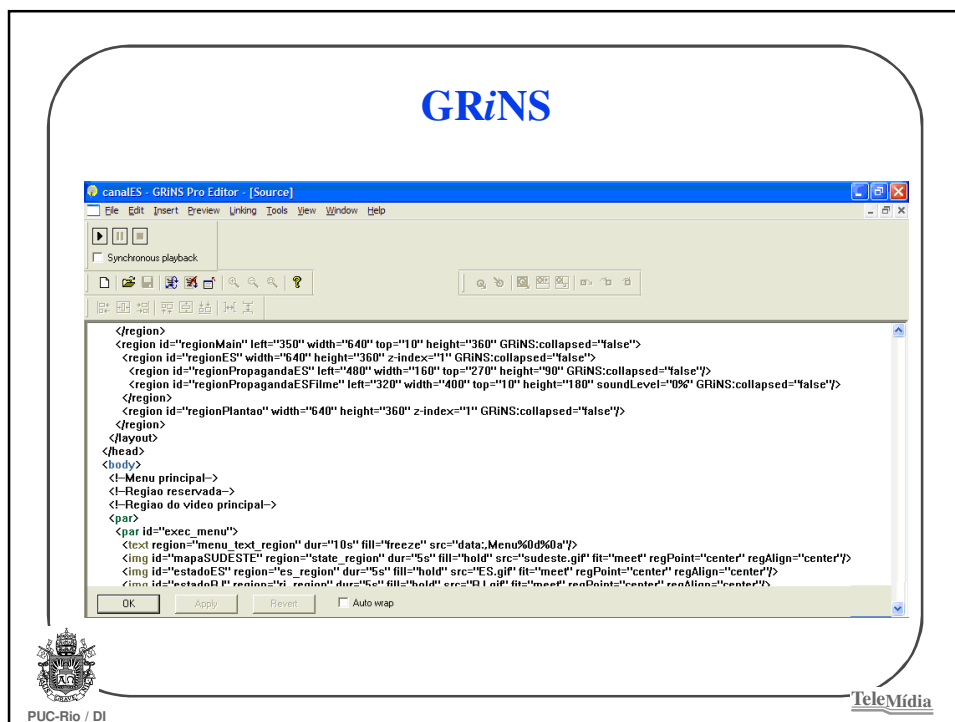
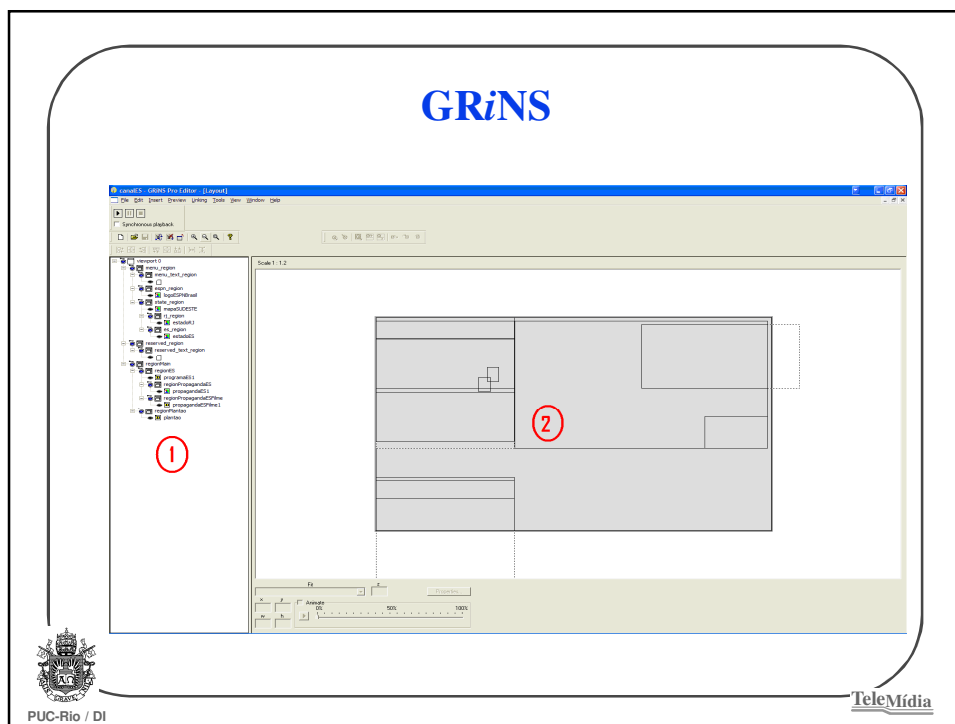
TeleMídia

GRiNS



PUC-Rio / DI

TeleMídia



LimSee2

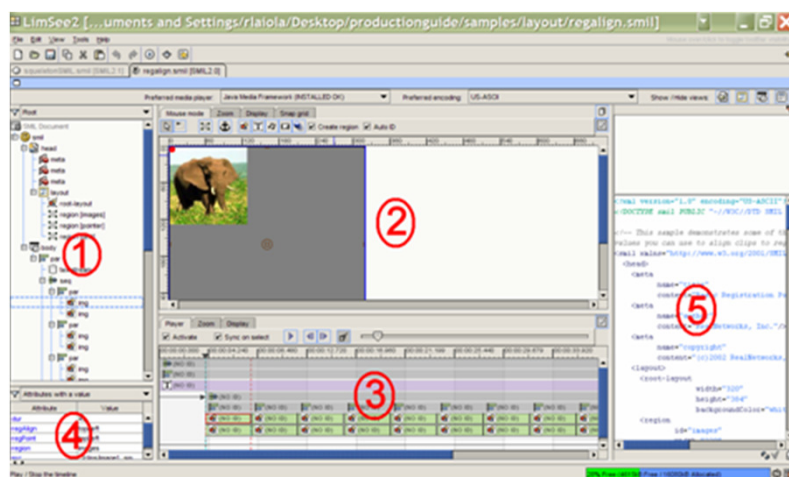
- Autoria para documentos SMIL desenvolvida pelo grupo de pesquisa WAM (*Web Adaptation and Multimedia*) do Institut National de Recherche en Informatique (INRIA).
- A região 1 (um) da Figura, é a visão estrutural
- Na visão temporal é possível especificar um determinado instante de tempo para visualizar na visão espacial o estado da apresentação.
- A região 4 (quatro) da Figura, chamada de visão de atributos, exhibe e permite a edição das propriedades de um elemento selecionado na visão temporal, estrutural ou espacial.
- As outras regiões são como no GRiNS
- Como no SMIL, não há suporte gráfico para interatividade



PUC-Rio / DI

TeleMídia

LimSee2



PUC-Rio / DI

TeleMídia

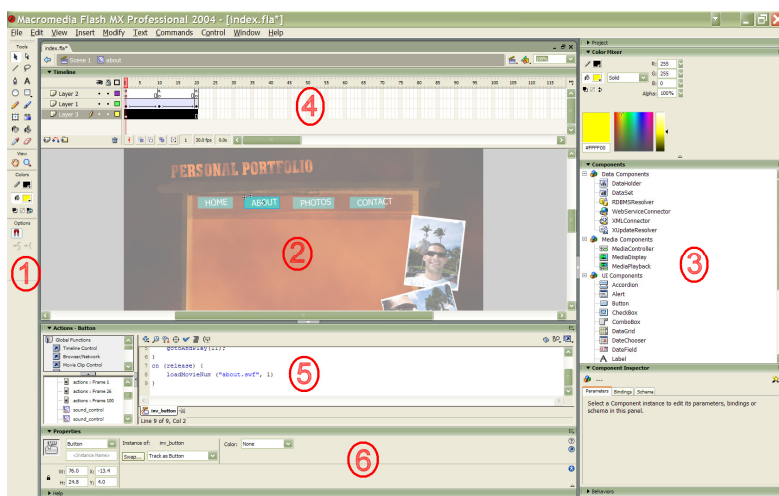
Ferramentas de Autoria Flash



PUC-Rio / DI

TeleMídia

Macromedia Flash



PUC-Rio / DI

TeleMídia

Ferramentas de Autoria para o MHP



PUC-Rio / DI

TeleMídia

JAME Author

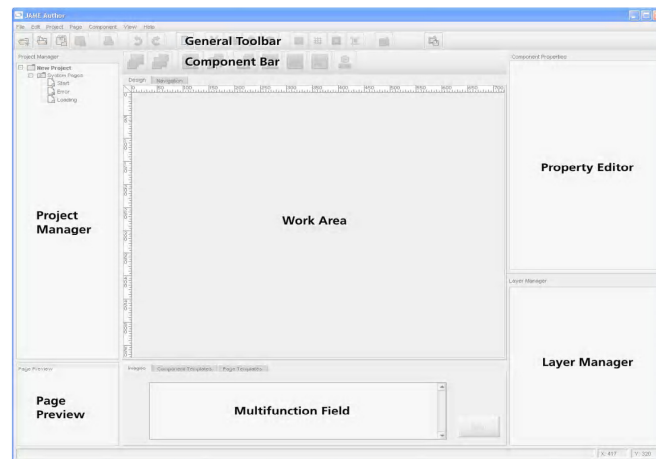
- MHP/OCAP
- Fraunhofer Institute for Media Communication IMK
- Aplicativos chamados de *page-based services*
- Páginas são descritas através de uma linguagem chamada JAME PDL (*JAME Page Description Language*), a qual é baseada em XML



PUC-Rio / DI

TeleMídia

JAME Author



PUC-Rio / DI

TeleMídia

JAME Author

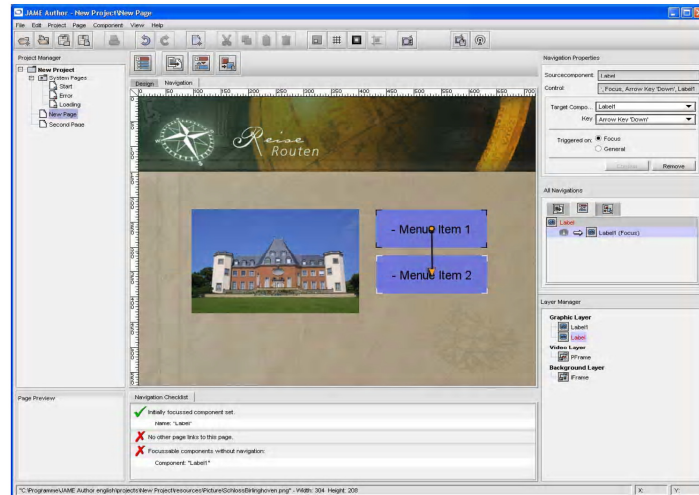
- *Project Manager* é a região onde são listadas as páginas de um aplicativo,
- *Page Preview* é a região de pré-visualização de uma página selecionada no *Project Manager*
- Quando um componente de uma página aberta na *Work Area* é selecionado, suas propriedades são exibidas na região *Property Editor*. Além de serem exibidas, essas propriedades também podem ser editadas.
- *Layer Manager* lista as camadas suportadas (*Graphic*, *Vídeo* e *Background Layer*), em concordância com o modelo de referência do MHP), e em qual delas cada componente está.



PUC-Rio / DI

TeleMídia

JAME Author



Como ilustração, a figura exibe um *link* que define a transferência de foco entre os componentes "Menu Item 1" e "Menu Item 2".



PUC-Rio / DI

TeleMídia

Cardinal Studio

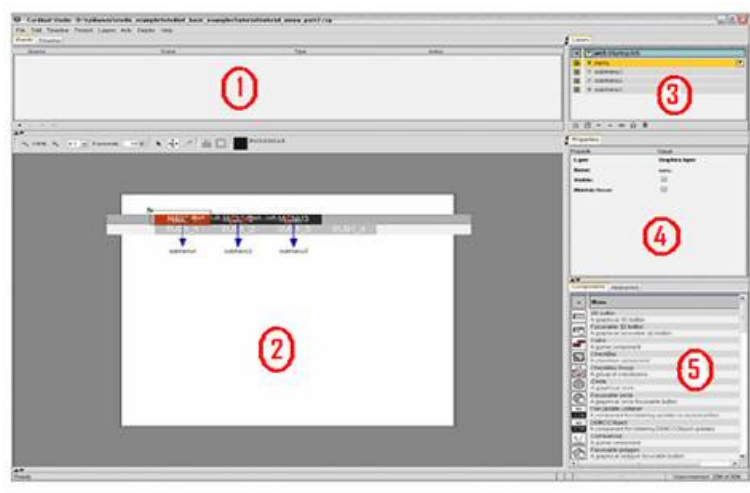
- Cardinal Systems, para a criação de aplicativos de TV digital interativa voltados para o *middleware* MHP
- O modelo de autoria tem como abstração de mais alto nível os "Atos" (*Acts*).
- Os atos são compostos por componentes que podem ser de dois tipos: visíveis e invisíveis.
- O primeiro grupo contempla elementos visuais básicos como painéis, áreas de texto e botões.
 - Esse grupo ainda é classificado em componentes de *background* e componentes focáveis. Praticamente todos os componentes de um conjunto possuem um componente equivalente no outro. A diferença é que componentes focáveis podem ser usados na navegação com o controle remoto, enquanto os componentes de *background* não.
- Componentes invisíveis, não são desenhados na tela. Alguns exemplos de componentes invisíveis são o canal de retorno, uma conexão HTTP, um *stream event*, um botão do controle remoto, entre outros.



PUC-Rio / DI

TeleMídia

Cardinal Studio



PUC-Rio / DI

TeleMídia

Ferramentas de Autoria NCL



PUC-Rio / DI

TeleMídia

NCL Composer

The screenshot displays the NCL Composer interface with the following components:

- NCL Textual View:** Shows XML code for defining media and link components.
- Structural View:** A central graphical editor showing a hierarchical tree structure of components.
- Outline View:** A tree view on the right side showing the hierarchy of media and link objects.
- Properties View:** A table at the bottom left showing attributes for the selected 'media.m1' object.
- Layout View:** A panel at the bottom center showing a grid layout with regions (rg1, rg2, rg3, rg4).
- Validator Plugin:** A panel at the bottom right showing a list of validation errors.

Attribute	Value
Descriptor	
id	m1
instance	
refer	
src	
type	video



PUC-Rio / DI

TeleMídia

NCL Eclipse

The screenshot displays the Eclipse IDE interface with the following components:

- Java Type Hierarchy:** A window at the top showing the hierarchy of Java classes.
- Project Explorer:** A sidebar on the left showing the project structure.
- XML Editor:** The main editor showing XML code for defining a media and link component.
- Problems View:** A panel at the bottom showing a list of validation errors.
- Console View:** A panel at the bottom showing the output of the application.



PUC-Rio / DI

TeleMídia

Programas ao Vivo



PUC-Rio / DI

TeleMídia

Linguagem de Templates



PUC-Rio / DI

TeleMídia