

# Tratamento de Documentos Multimídia/Hipermídia



PUC-Rio / DI

TeleMídia

## Objetos de Dados

- A definição do conteúdo do nó e de suas âncoras são contidas em objetos chamados **objetos de dados** — OD.
- Um ambiente de autoria, contudo, deve também permitir a definição do comportamento esperado de cada componente, isto é, como e com que ferramentas o objeto de dados associado será apresentado.
- Estas definições devem ser realizadas, preferencialmente, independente dos objetos de dados.
- No NCM, objetos **descriptor** contêm tais informações.



PUC-Rio / DI

TeleMídia

## Descritores

- ➔ A independência entre descritores e objetos de dados vai permitir o melhor **reuso de objetos**, permitindo apresentações diferentes do mesmo dado.
- ➔ Descritores diferentes podem levar a durações diferentes de apresentação das unidades de informação de um objeto de dados, a qualidades de apresentações diferentes e a requisitos de plataforma diferentes. Assim, a definição de todos esses quesitos deve também fazer parte do descritor, e não do objeto de dados.



PUC-Rio / DI

TeleMídia

## Objeto de Representação

- ➔ A agregação de um objeto de dados e um descritor, com a finalidade de apresentar um componente, é chamado **objeto de representação** — OR e será considerado uma **versão de representação** do objeto de dados.



PUC-Rio / DI

TeleMídia

## Objetos de Armazenamento

- ➔ Um objeto de dados é criado ou como um novo objeto totalmente novo, ou como uma versão local de outro objeto de dados, ou ainda como uma versão local de objetos persistentes anteriormente criados, chamados de **objetos de armazenamento** — OA.
- ➔ Nesse último caso, o objeto de dados é uma cópia do objeto de armazenamento, com a possível adição de novos atributos (não-persistentes) que são dependentes da aplicação.
- ➔ Objetos de dados criados a partir de outros objetos são considerados **versões de dados**.
- ➔ Objetos de armazenamento, de dados e de representação estão relacionados através de operações de versionamento.



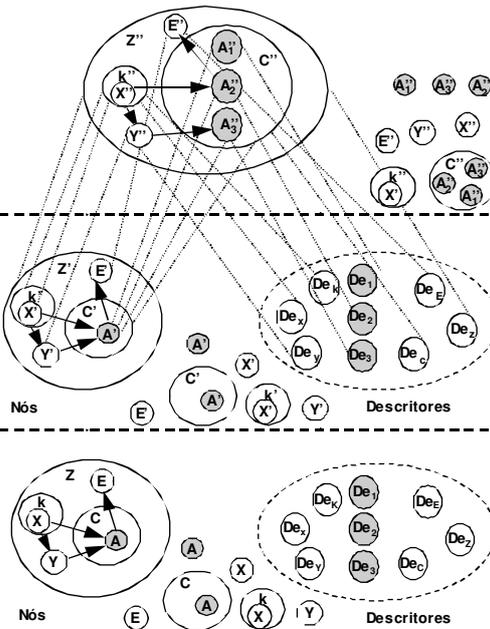
PUC-Rio / DI

TeleMídia

Plano de  
Objetos de  
Representação

Plano de  
Objetos de Dados

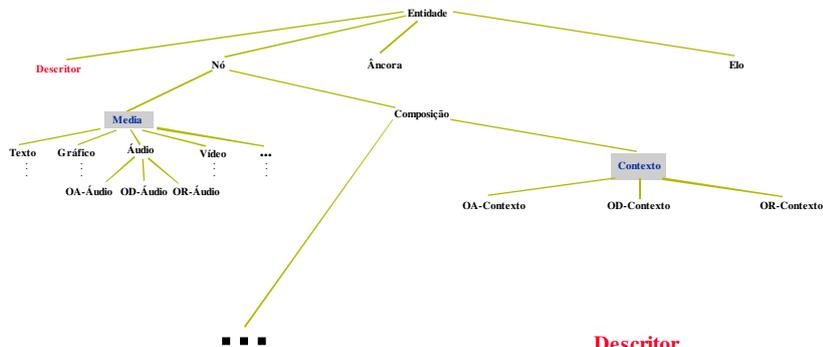
Plano de  
Objetos de  
Armazenamento



PUC-Rio / DI

TeleMídia

## Modelo Conceitual



## Descritor - Especificação de Iniciação

- ➔ Contém as informações necessárias para iniciar a apresentação de uma entidade. Em particular, ela define os métodos para exibição ou edição das entidades.
- ➔ Ela deve definir todos os parâmetros (**valores de propriedades**) necessários para a criação de um objeto representação a partir de um objeto de dados.
- ➔ Uma especificação de iniciação possui também uma lista ordenada de operações que devem ser executadas para preparar a exibição do nó.



## Descritor - Especificação de Iniciação

- ➔ A noção exata do que constitui uma especificação de iniciação depende da classe do nó ao qual o descritor será associado.
- ➔ Por exemplo, para um nó texto, uma especificação de iniciação pode determinar o editor de texto “word,” pode passar, para a criação do objeto de representação, o tamanho da janela de apresentação do texto e sua posição na tela, etc.
- ➔ No caso especial de uma composição, o método default de exibição é o definido na composição, ou redefinido nas subclasses (por exemplo o método para exibição da hiperbase pública e da base privada).



PUC-Rio / DI

TeleMídia

## Especificação de Término

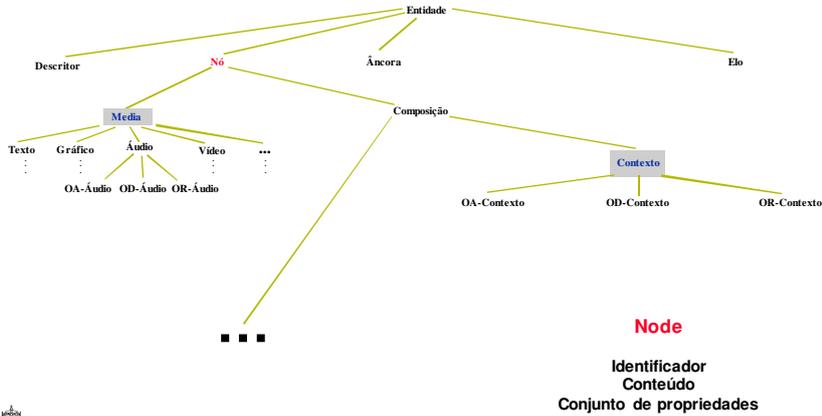
- ➔ Uma especificação de término contém as informações necessárias para finalizar a apresentação de uma entidade. Em particular, ela define os métodos que devem ser executados ao final de uma exibição.
- ➔ Uma especificação de término possui também uma lista ordenada de operações que devem ser executadas ao finalizar a exibição do nó.



PUC-Rio / DI

TeleMídia

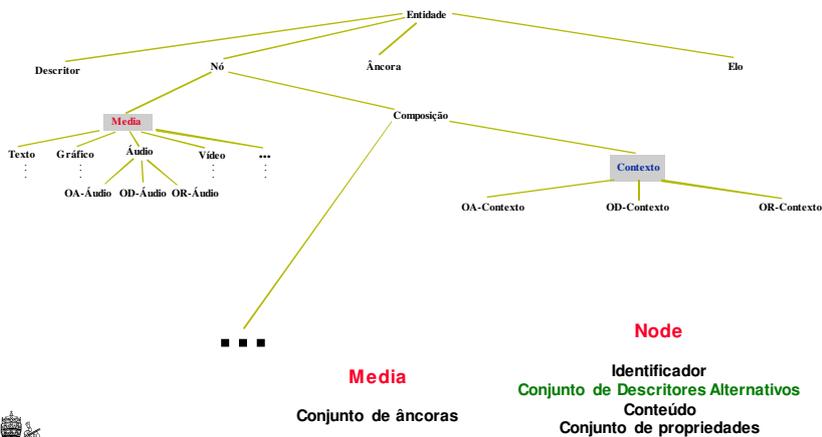
## Modelo Conceitual



PUC-Rio / DI

TeleMídia

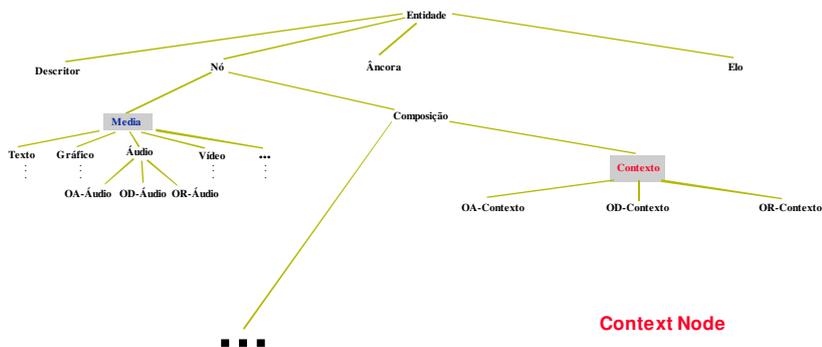
## Modelo Conceitual



PUC-Rio / DI

TeleMídia

## Modelo Conceitual



**Context Node**  
 Coleção de Apresentação  
 Conteúdo: Conjunto de *Media* or *Context Nodes*  
 Conjunto de *Links*  
 Lista ordenada de âncoras  
 Estado

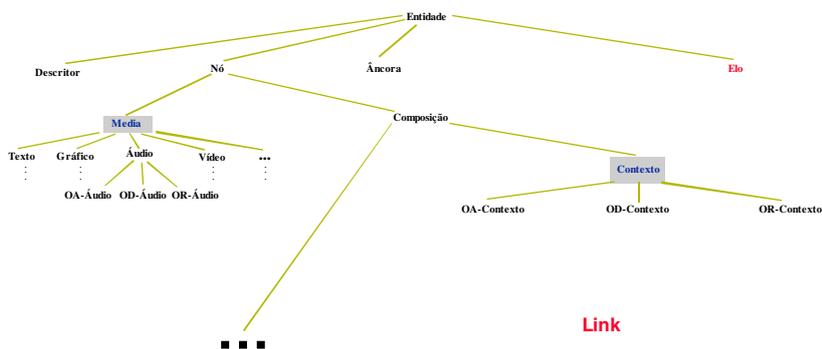


PUC-Rio / DI

TeleMídia

## Modelo Conceitual

$PT: \langle (N_k, \dots, N_2, N_1), \alpha, \mathcal{D} \rangle$



**Link**  
 Conjunto de Pontos Terminais Fonte  
 Conjunto de Pontos Terminais de Destino  
 Conteúdo



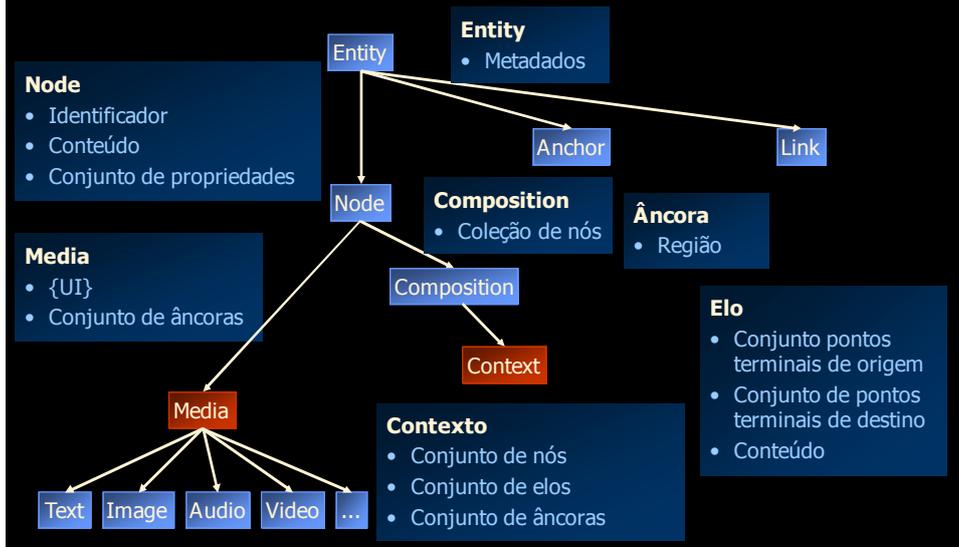
PUC-Rio / DI

TeleMídia

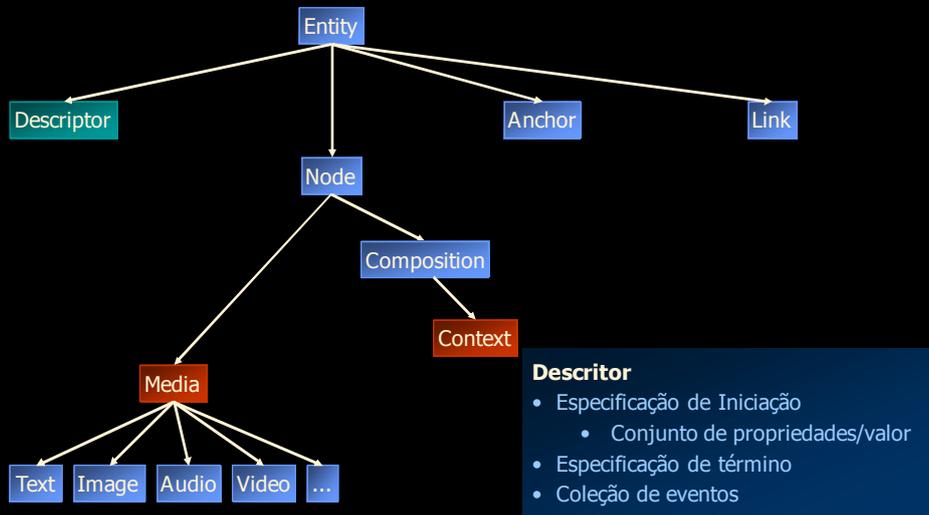
# Recordando



## Modelo Conceitual



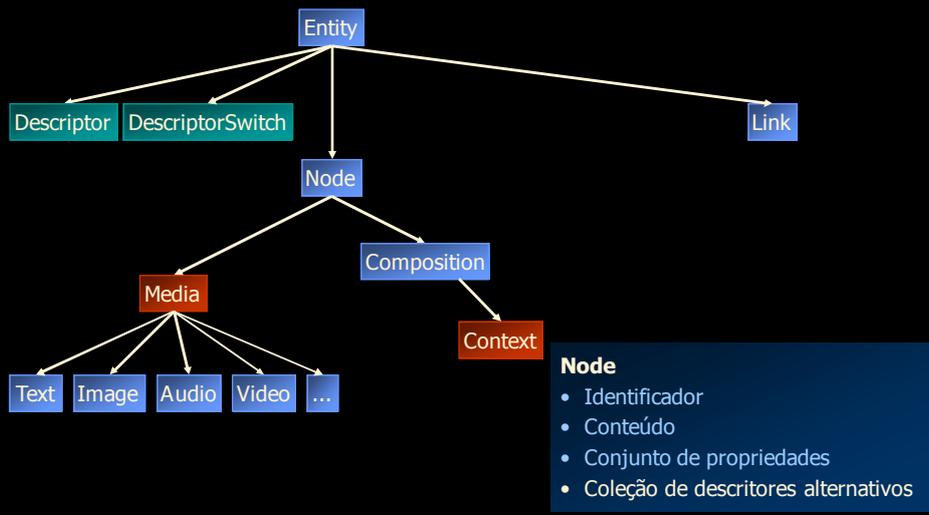
## Modelo Conceitual



17



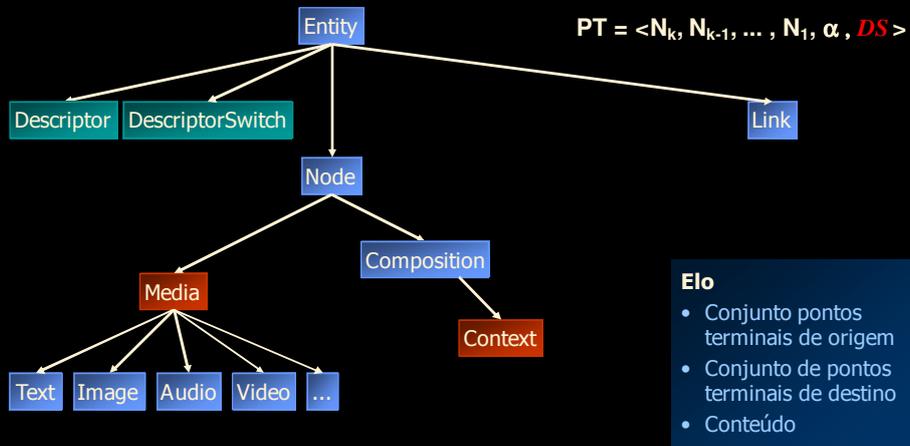
## Modelo Conceitual



18



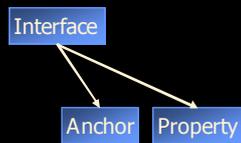
## Modelo Conceitual



19



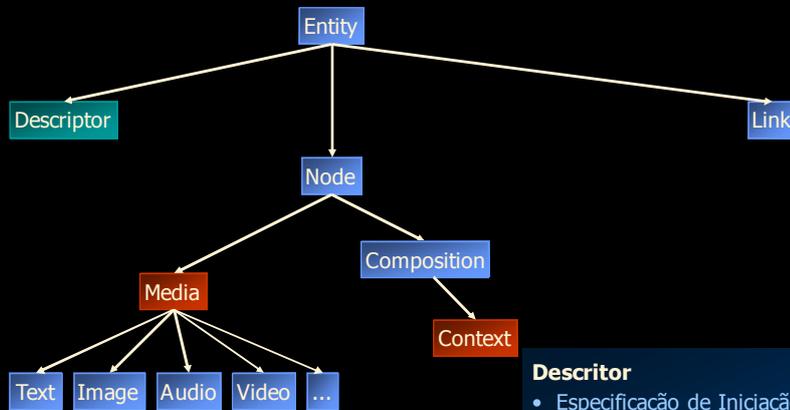
## Modelo Conceitual



20



## Modelo Conceitual



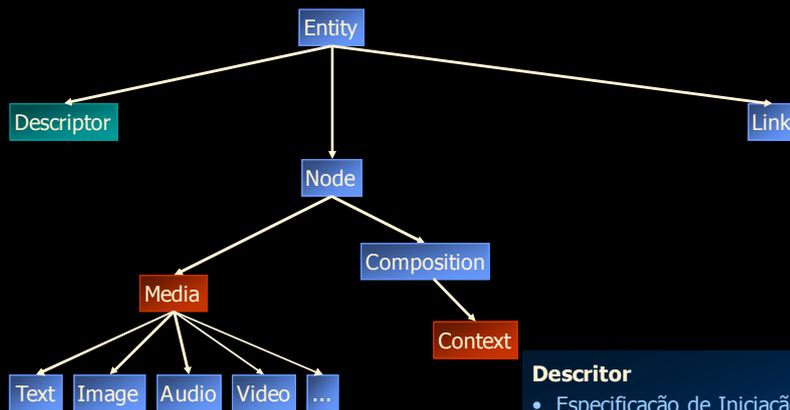
### Descritor

- Especificação de Iniciação
  - Conjunto de propriedades/valor
- Especificação de término
- Coleção de eventos

21



## Modelo Conceitual

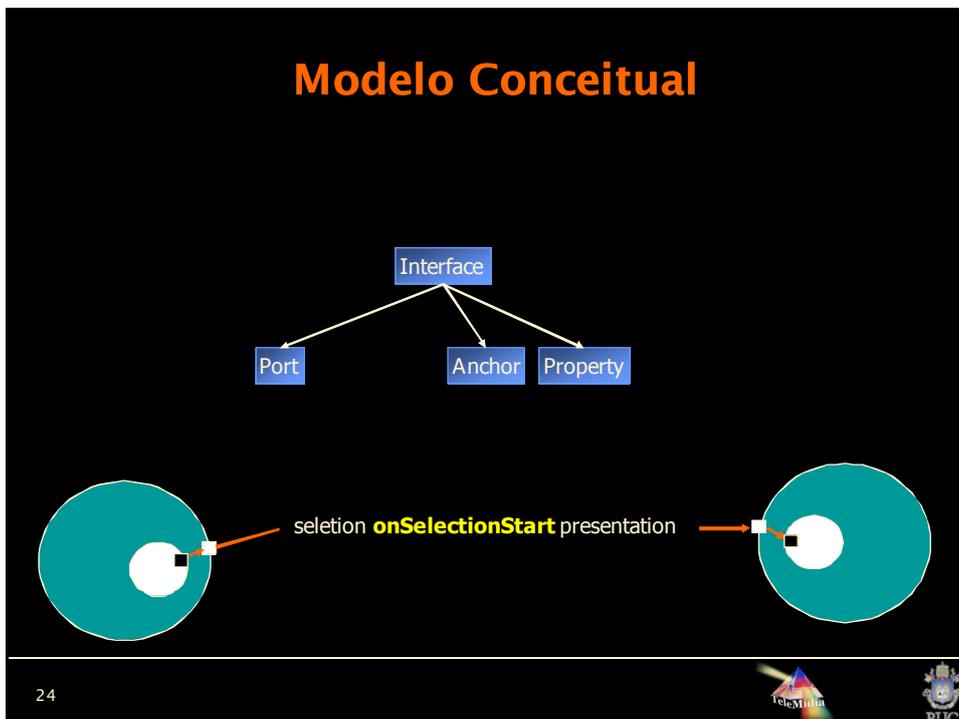
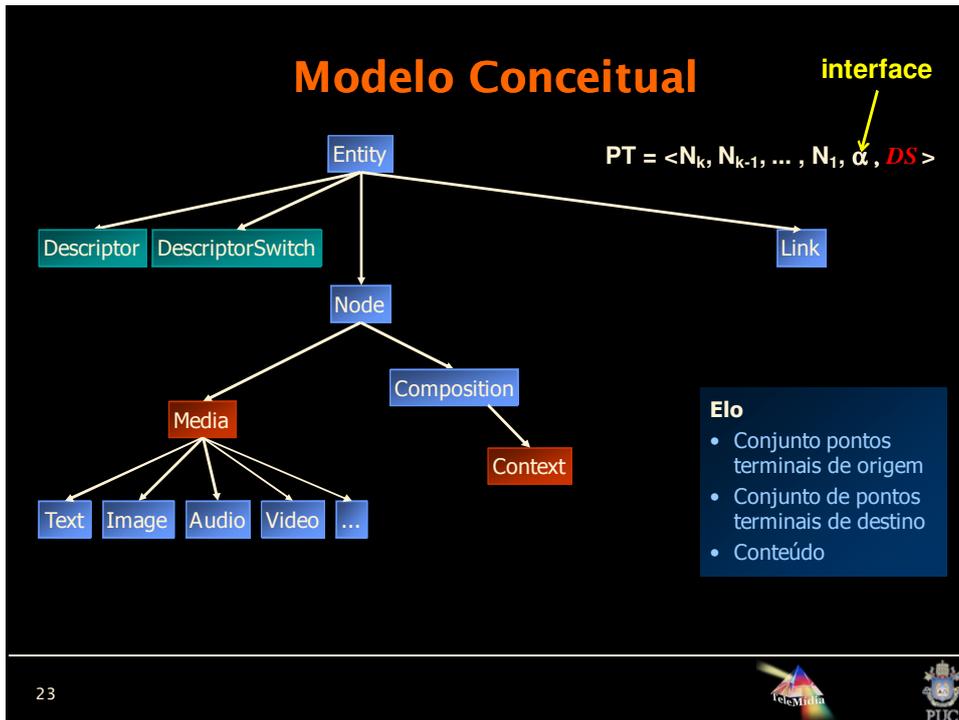


### Descritor

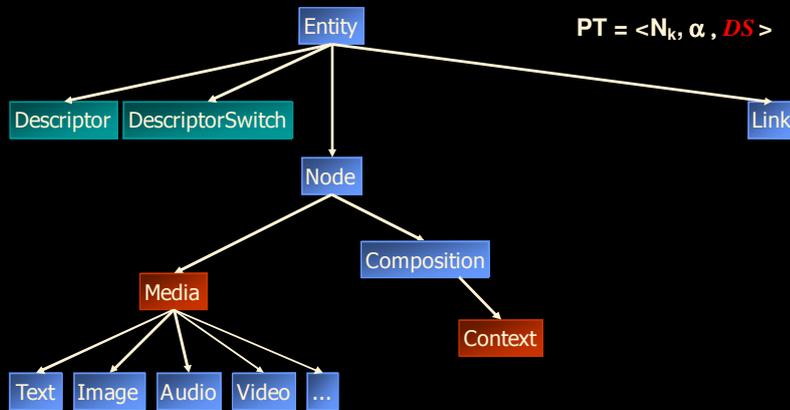
- Especificação de Iniciação
  - Conjunto de propriedades/valor
- Especificação de término

22





## Modelo Conceitual



25



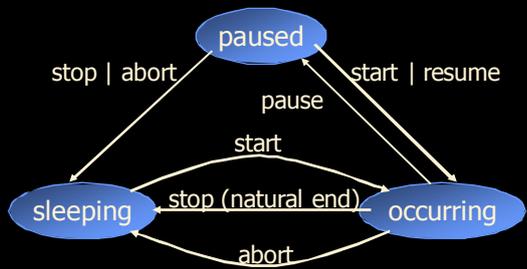
## Eventos

- Seguindo a definição de Perez-Luque, um evento é uma ocorrência no tempo que pode ser instantânea ou durar um período de tempo.
- **Evento de pré-busca:** busca de uma coleção não vazia de unidades de informação
- **Evento de exibição:** exibição de uma coleção não vazia de unidades de informação
- **Evento de seleção:** seleção de uma coleção não vazia de unidades de informação
- **Evento de atribuição:** mudança de um atributo de um nó ou da condição de habilitação das mudanças de comportamento definidas no objeto descritor (visto mais adiante)
- O início ou fim de um evento é instantâneo e é denominado ponto de sincronização. Note que o ponto de sincronização é também um evento.

26



# Máquina de Estados de Eventos

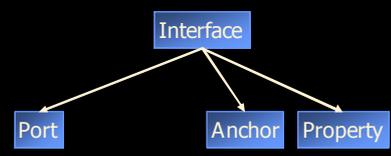


Transition (caused by action)
<i>sleeping</i> → <i>occurring</i> (start)
<i>occurring</i> → <i>sleeping</i> (stop or natural end)
<i>occurring</i> → <i>sleeping</i> (abort)
<i>occurring</i> → <i>paused</i> (pause)
<i>paused</i> → <i>occurring</i> (resume or start)
<i>paused</i> → <i>sleeping</i> (stop)
<i>paused</i> → <i>sleeping</i> (abort)

27



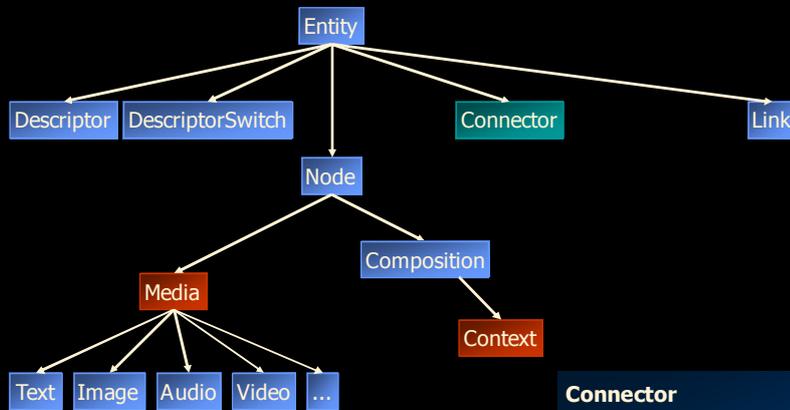
# Modelo Conceitual



28



## Modelo Conceitual



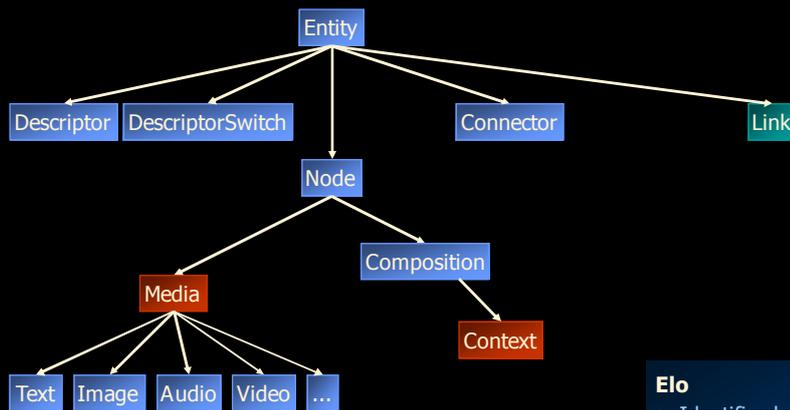
### Connector

- Identificador
- Conjunto de papéis (*roles*)
- *Glue expression*

29



## Modelo Conceitual



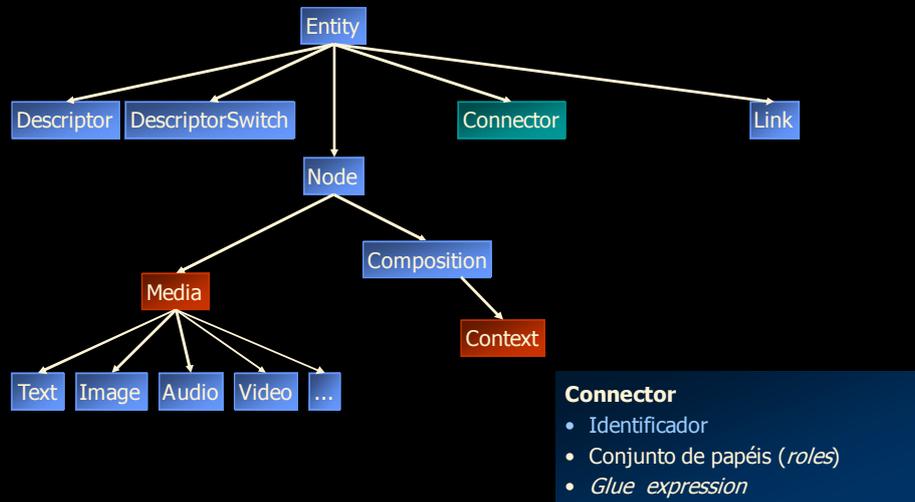
### Elo

- Identificador
- Conjunto de *binds*
- Connector

30



## Modelo Conceitual



31



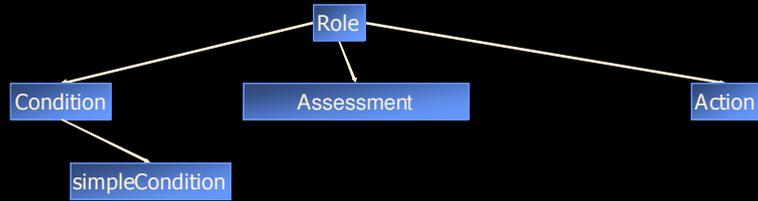
## Papel (Role)

- Identificador
- Tipo de evento
- Cardinalidade: especifica o número mínimo e máximo de eventos que podem exercer o papel, em um elo.
- Observação: Se o tipo de evento de um papel for de **atribuição**, o papel pode também identificar o **valor**.

32



# Modelo Conceitual



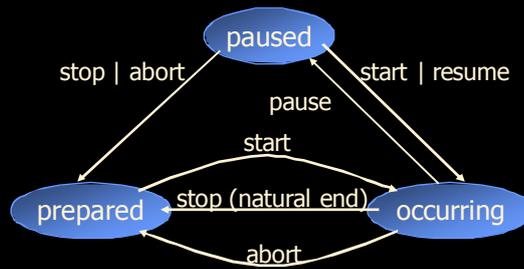
**Role**  
 Identificador  
 Tipo de Evento  
 Cardinalidade  
 Qualificador  
 valor

simpleCondition role

transition  
 delay



# Máquina de Estados de Eventos



Condições Simples
<i>sleeping</i> → <i>occurring</i> (start)
<i>occurring</i> → <i>sleeping</i> (stop or natural end)
<i>occurring</i> → <i>sleeping</i> (abort)
<i>occurring</i> → <i>paused</i> (pause)
<i>paused</i> → <i>occurring</i> (resume or start)
<i>paused</i> → <i>sleeping</i> (stop)
<i>paused</i> → <i>sleeping</i> (abort)



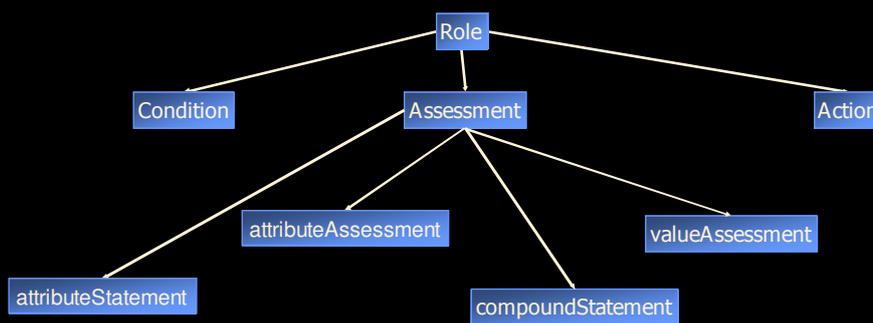
## Valores reservados de papéis de condição

Role Value	Transition Value	Event Type
	<i>Starts</i>	<i>presentation</i>
<i>onEnd</i>	<i>Stops</i>	<i>presentation</i>
<i>onAbort</i>	<i>Aborts</i>	<i>presentation</i>
<i>onPause</i> <i>onBegin</i> <i>se</i>	<i>Pauses</i>	<i>presentation</i>
<i>onResume</i>	<i>Resumes</i>	<i>presentation</i>
<i>onSelection</i>	<i>Stops</i>	<i>selection</i>
<i>onAttributionBegin</i>	<i>Starts</i>	<i>attribution</i>
<i>onEndAttribution</i>	<i>Stops</i>	<i>attribution</i>

35



## Modelo Conceitual



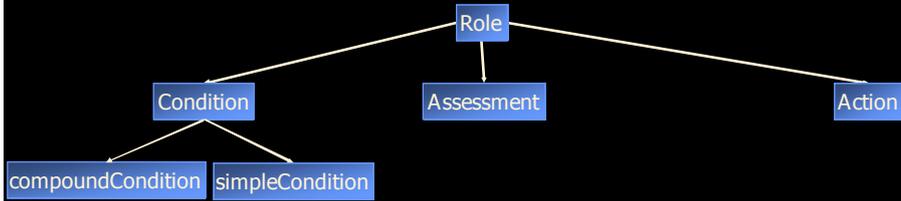
### Role

Identificador  
Tipo de Evento  
Cardinalidade  
Qualificador  
Valor

36



# Modelo Conceitual



**Role**  
Identificador  
Tipo de Evento  
Cardinalidade  
Qualificador  
Valor

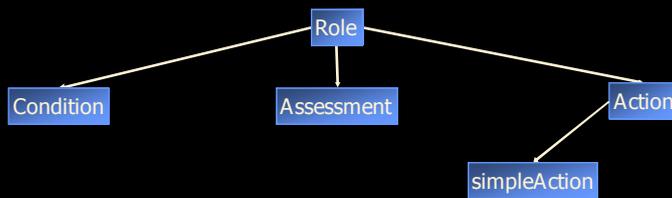
**compoundCondition**

operator  
delay

37



# Modelo Conceitual



**Role**  
Identificador  
Tipo de Evento  
Cardinalidade  
Qualificador  
Valor

**simpleAction**

actionType  
delay

38



## Action Roles

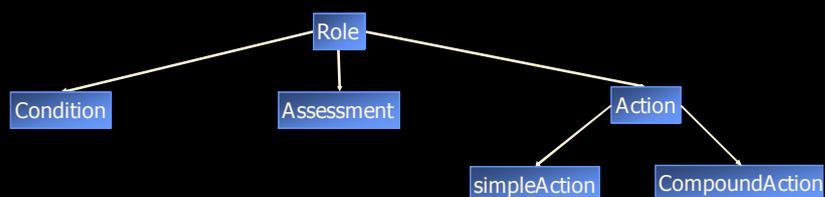
- Capturam ações que podem ser executadas sobre os eventos.
- Tipo da ação (valores reservados de papéis)

Action type	Event type
<i>Start</i>	<i>presentation</i>
<i>Stop</i>	<i>presentation</i>
<i>Abort</i>	<i>presentation</i>
<i>Pause</i>	<i>presentation</i>
<i>Resume</i>	<i>presentation</i>
<i>Start</i>	<i>attribution</i>

39



## Modelo Conceitual



**Role**  
 Identificador  
 Tipo de Evento  
 Cardinalidade  
 Qualificador  
 Valor

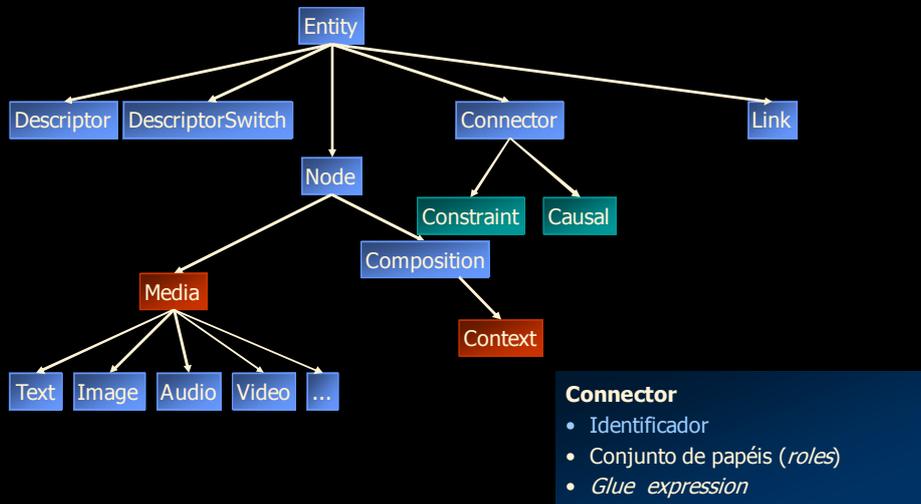
**compoundAction**

**actionType**  
 delay

40



## Modelo Conceitual



41



## Modelo Conceitual



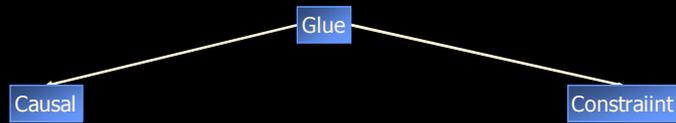
### Causal

simpleCondition | compoundCondition  
simpleAction | compoundAction

42



## Modelo Conceitual



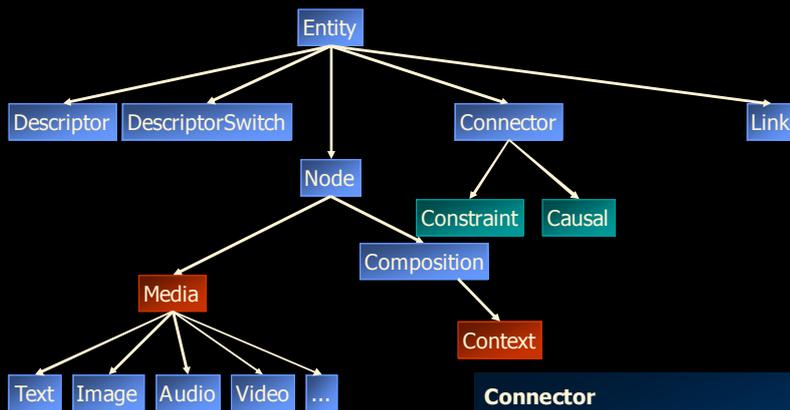
Constraint

statementExpression

43



## Modelo Conceitual



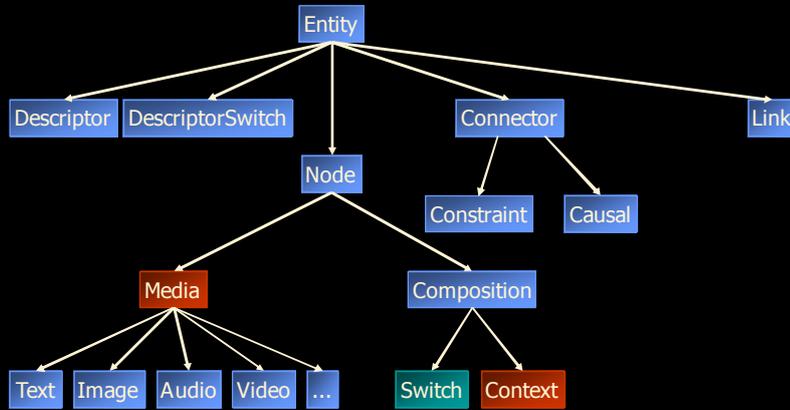
### Connector

- Identificador
- Conjunto de papéis (*roles*)
- *Glue expression*

44



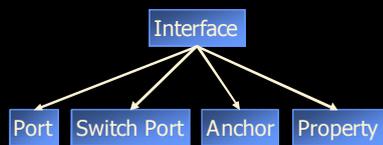
## Modelo Conceitual



45



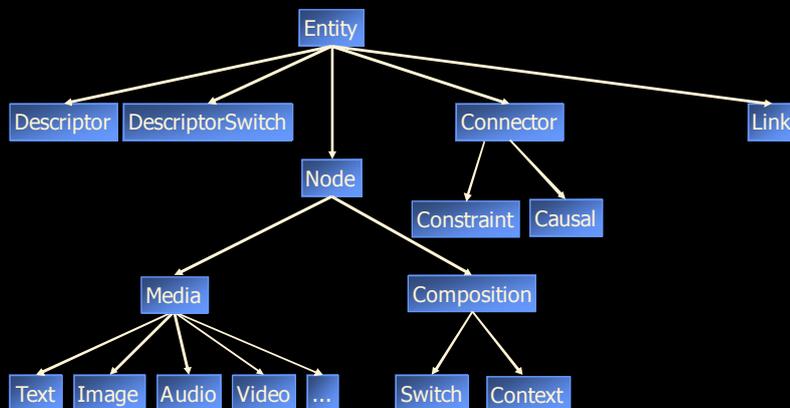
## Modelo Conceitual



46



## Modelo Conceitual



47



## Paradigmas de Programação

- **Imperativo (procedural)**
  - Especificação: como fazer
  - Maior poder de expressão
- **Declarativo**
  - Especificação: a intenção final
  - Especificação em mais alto nível

48



## Linguagem de Especificação

- **Simple de ser entendida e usada**
- **Leve**
- **Expressiva**
- **Linguagem Declarativa de Domínio Específico DSL**

49



## Funcionalidades – Suporte declarativo a:

- Sincronismo de mídia
  - Interatividade como um caso particular
- Adaptabilidade
- Múltiplas redes de distribuição
- Múltiplos dispositivos de exibição
- Edição ao vivo

50



Declarativos  
NCL  
HTML  
MHEG

Imperativo:  
Lua  
ECMAScript  
Java

51

TeleMídia  
PUC

Declarativo

+ eficiente  
+ fácil de usar

Imperativo

+ geral

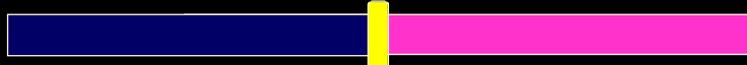
52

TeleMídia  
PUC

## Declarativo X Imperativo

Declarativo

Imperativo



53



## Declarativo X Imperativo

Declarativo

Imperativo



54



## Declarativo X Imperativo

Declarativo

Imperativo



55



## Declarativo X Imperativo

Declarativo

Imperativo



HTML

56



## Por que NCL?

# NCL

57



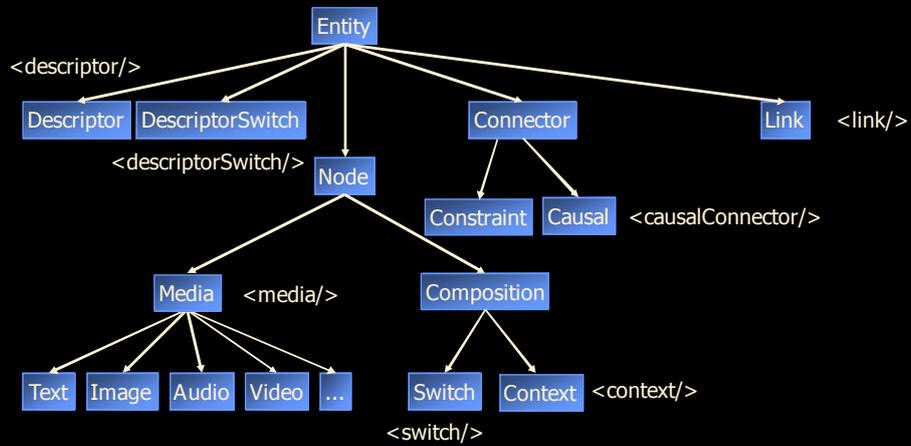
## NCL – Nested Context Language

- Suporte à sincronização
  - Sincronização baseada na estrutura
  - Suporte a canal de retorno
- Suporte a múltiplas redes de distribuição
- Suporte a múltiplos dispositivos
- Suporte à adaptação do conteúdo e da apresentação
- Suporte à edição ao vivo
- **NCL é software livre**

58



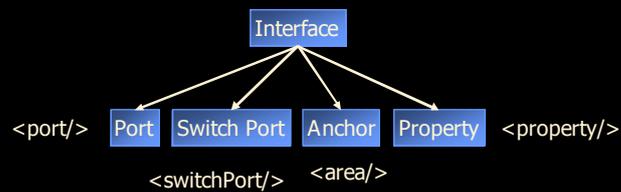
## Modelo Conceitual



59



## Modelo Conceitual



60



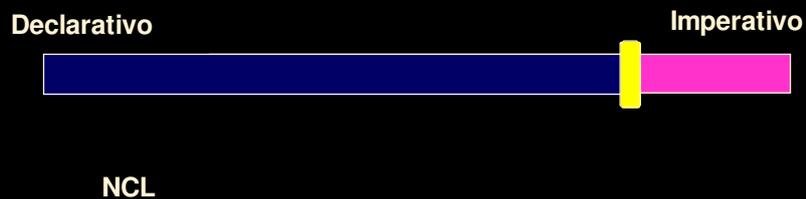
## NCL Nested Context Language

- Inovação do Sistema Brasileiro de TV Digital:
  - Linguagem declarativa NCL (Nested Context Language)
  - Sua linguagem de script Lua
  - seu ambiente de execução *middleware* Ginga-NCL.
- ITU-T H.761 Recommendation for IPTV services
- ITU-R BT 1691-1 Recommendation for Terrestrial DTV

61



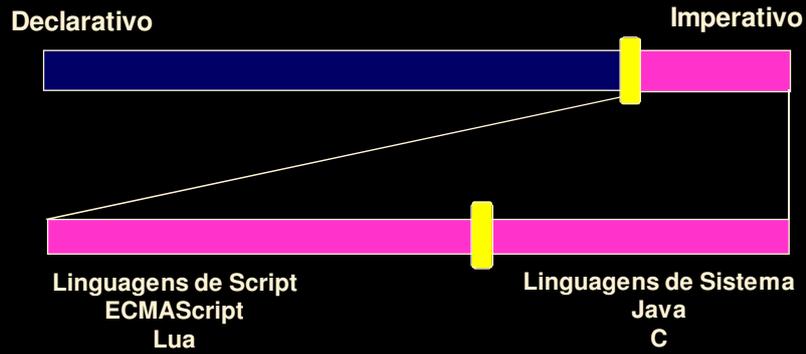
## Declarativo X Imperativo



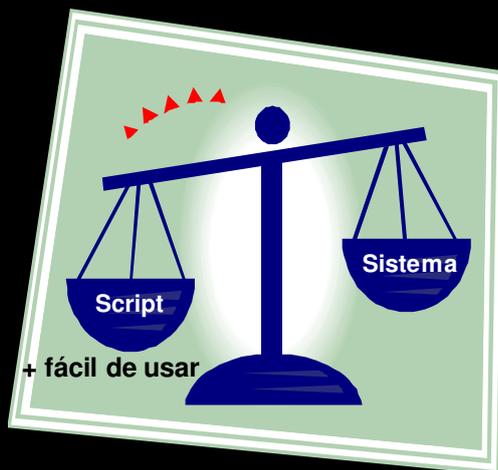
62



# Declarativo X Imperativo



63



Eficiência?  
Footprint?

**Depende do Problema**

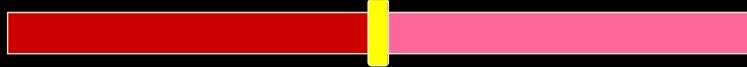
64



# Script X Sistema

Script

Sistema



65



# Script X Sistema

Script

Sistema



66



## Script X Sistema



67



## Opções Ginga



Tarefas de pequena e média complexidade: Lua  
Tarefas de grande complexidade: Java

68



## Opções Ginga

Lua

Java



69



## Por que Lua?



70



## Por que Lua?

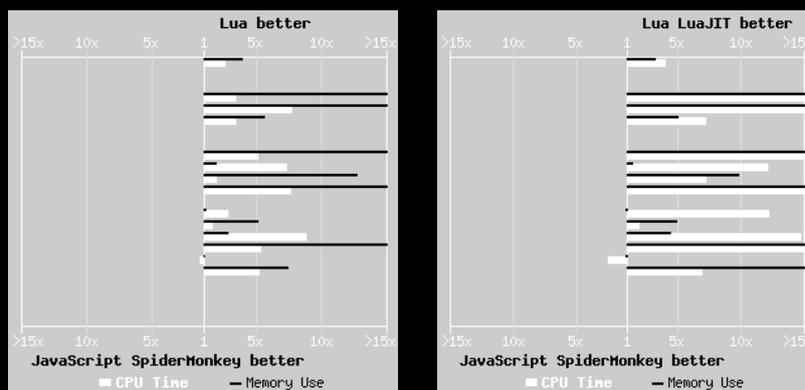


- ✓ Lua é simples e expressiva
- ✓ Lua é portátil
- ✓ Lua é fácil de embarcar
- ✓ Lua é rápida
- ✓ Lua é robusta
- ✓ Lua é software livre

71



<http://shootout.alioth.debian.org>



JavaScript SpiderMonkey = 936 Kbytes  
Lua = 120 Kbytes  
LuaJIT = 150 Kbytes

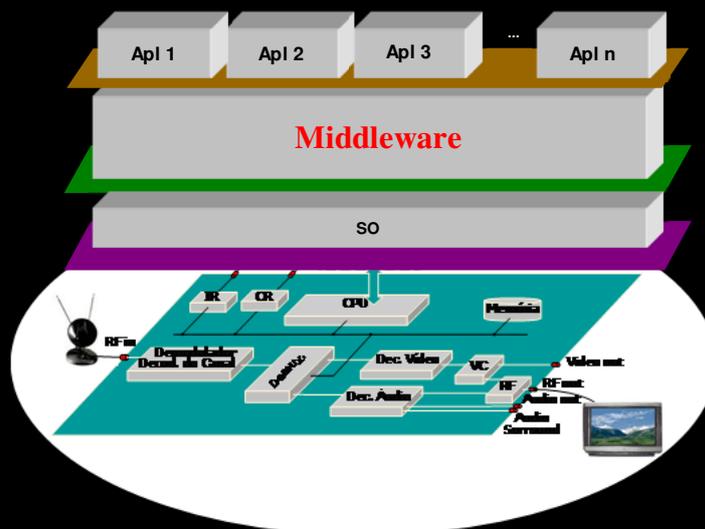
72



# Alguns Jogos Usando Lua

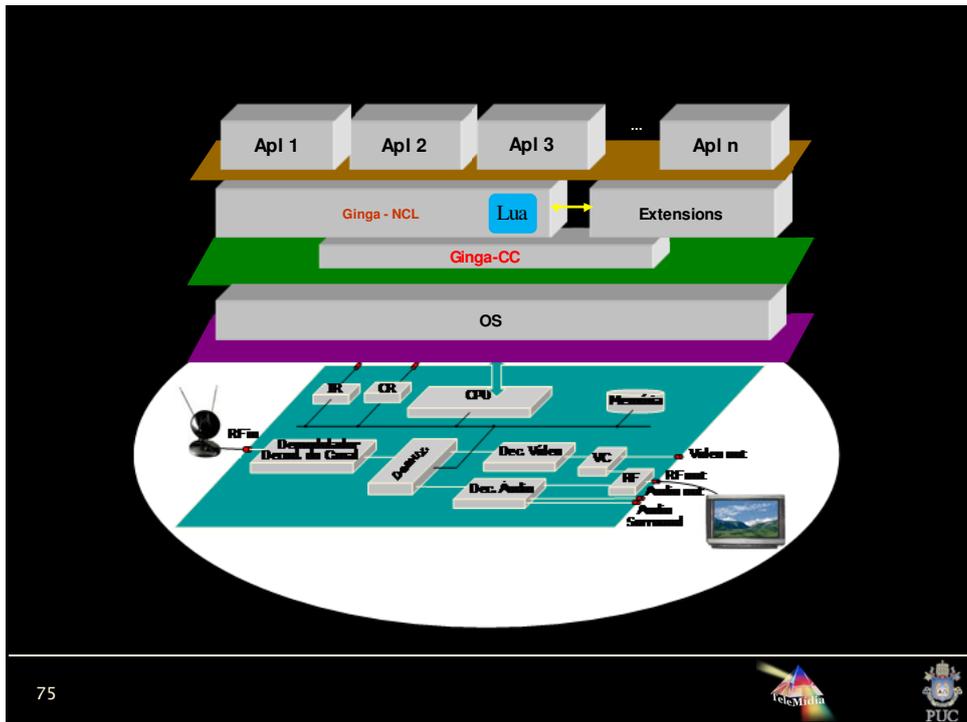


73

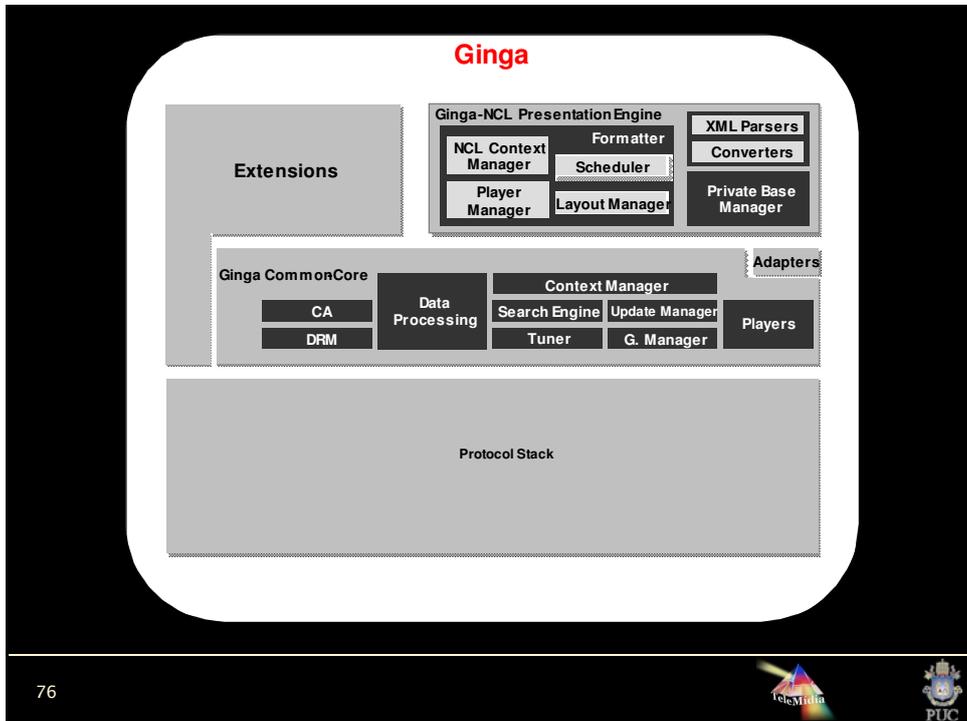


74



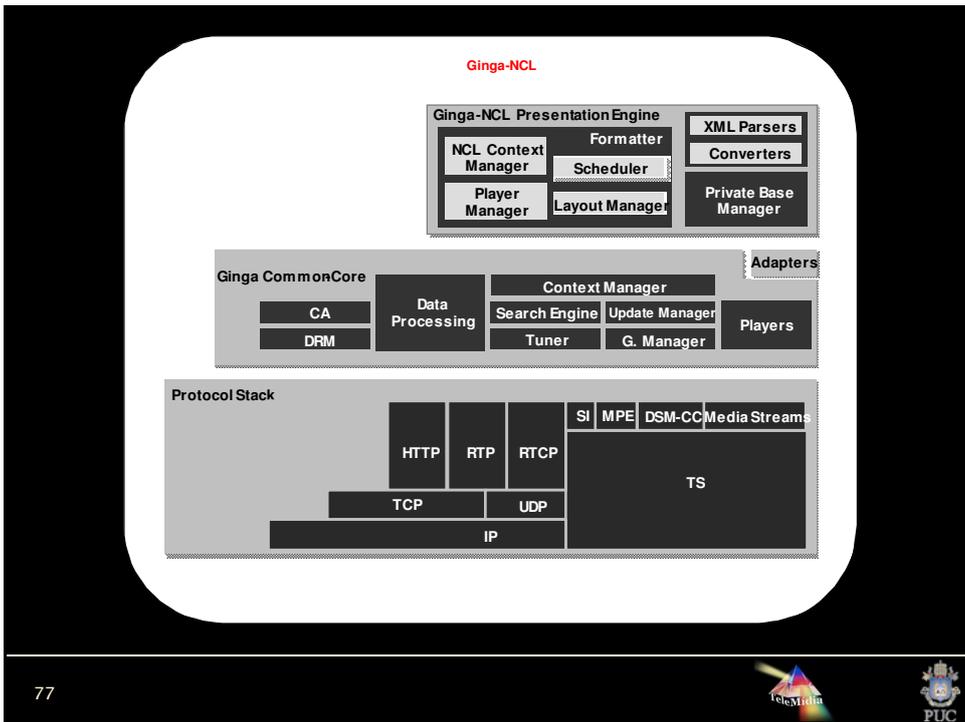


75

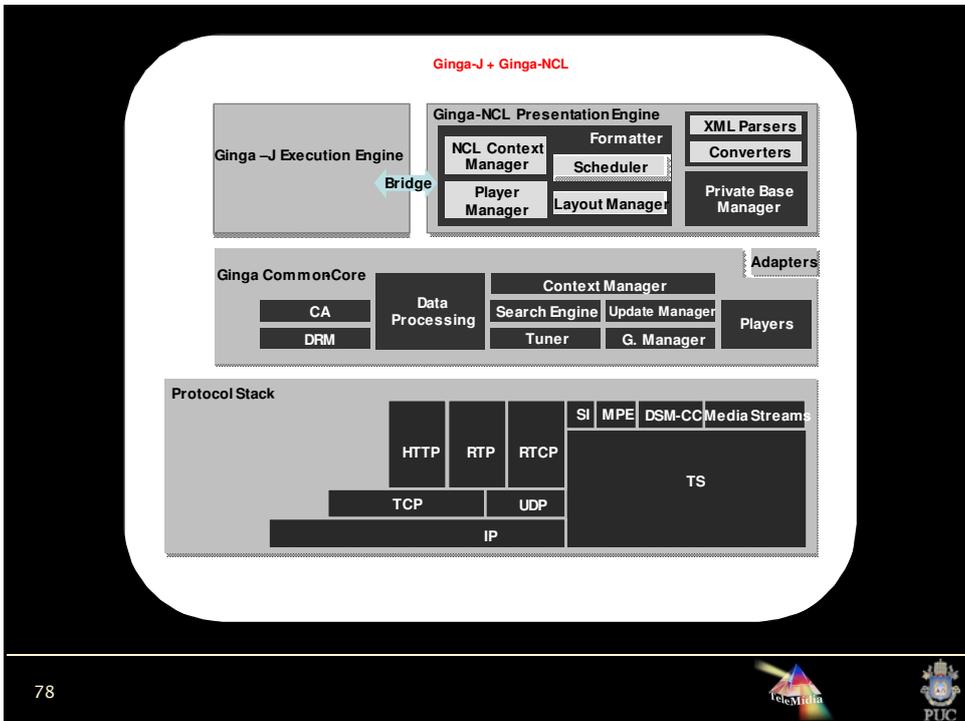


76





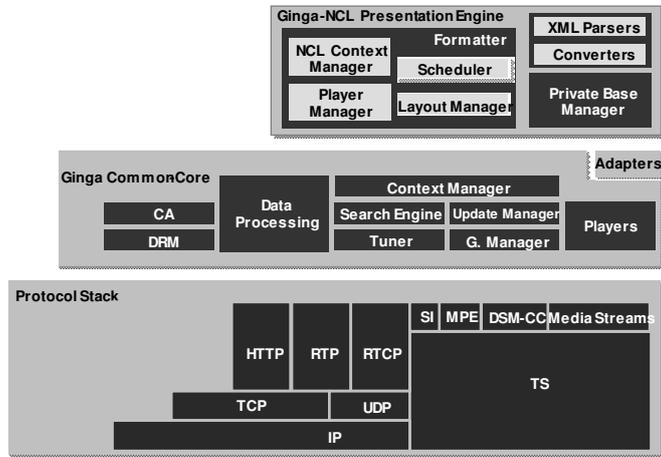
77



78



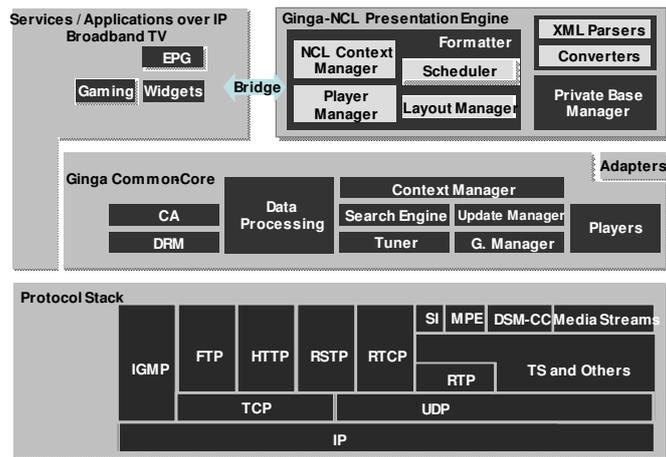
Broadband TV + Ginga-NCL



79



Broadband TV + Ginga-NCL



80



