

Sandro Santos Andrade

**Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas**



Introdução



software



modelos

modelagem

Modelo Arquitetural:

Modelagem Arquitetural:

Introdução

-
-
- **notações de modelagem de arquiteturas**

Notação de Modelagem Arquitetural:

Introdução



UML

Conceitos de Modelagem



Stakeholder

Conceitos de Modelagem

Modelagem Dirigida a *Stakeholder*

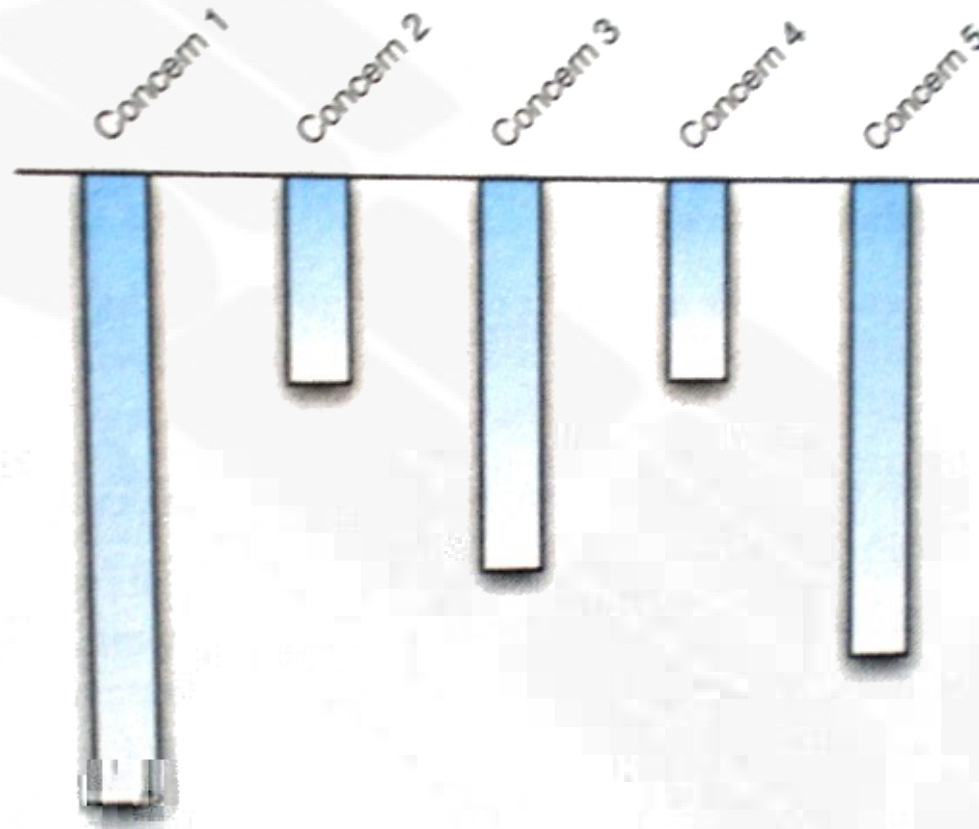


stakeholders



Conceitos de Modelagem

Modelagem Dirigida a *Stakeholder*



Conceitos de Modelagem

Modelagem Dirigida a *Stakeholder*



software

Conceitos de Modelagem

Modelagem Dirigida a *Stakeholder*



stakeholder

software

bugs

Conceitos de Modelagem

Modelagem Dirigida a *Stakeholder*



Conceitos de Modelagem

Conceitos Arquiteturais Básicos

-
-
-
-
-
- *Rationale*

Conceitos de Modelagem

Conceitos Arquiteturais Básicos

-
-
- *Building blocks*

-
- *Building blocks*

Conceitos de Modelagem

Conceitos Arquiteturais Básicos

-
-
-
- *Rationale*
-

Conceitos de Modelagem

Conceitos Arquiteturais Básicos

-
-
-
- *rationale*
- *stakeholders*
- *rationale*

Conceitos de Modelagem

Conceitos Arquiteturais Básicos



Conceitos de Modelagem

Conceitos Arquiteturais Básicos



Foco central da modelagem arquitetural

Conceitos de Modelagem

Conceitos Arquiteturais Básicos

-
-
- *World Wide Web*
-
-
-
- *cases*
-
- *use-*

Conceitos de Modelagem

Elementos de um Estilo Arquitetural



Conceitos de Modelagem

Elementos de um Estilo Arquitetural



Conceitos de Modelagem

Elementos de um Estilo Arquitetural



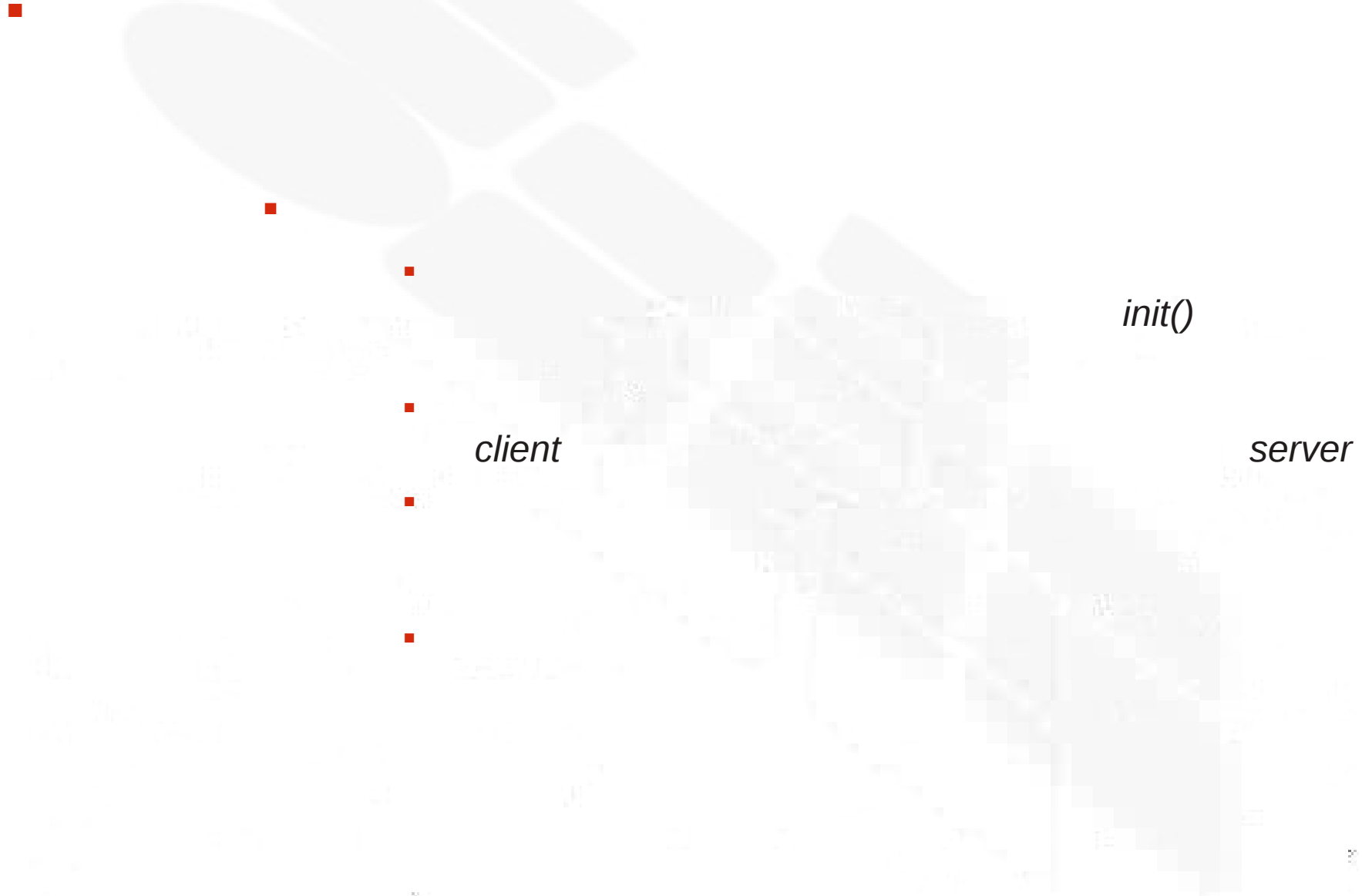
Conceitos de Modelagem

Elementos de um Estilo Arquitetural



Conceitos de Modelagem

Elementos de um Estilo Arquitetural



Conceitos de Modelagem

Elementos de um Estilo Arquitetural



Conceitos de Modelagem

Aspectos Estáticos e Dinâmicos

-
-
-
-

hosts

hosts

run-time

Conceitos de Modelagem

Aspectos Estáticos e Dinâmicos

- 

run-time

aspectos
modelos

Conceitos de Modelagem

Aspectos Estáticos e Dinâmicos

- **aspecto**
- **modelo**
- *statechart*
- *statechart*

Conceitos de Modelagem

Aspectos Estáticos e Dinâmicos

read-write

machine-

readable machine-writable

on-the-fly

Conceitos de Modelagem

Aspectos Funcionais e Não-Funcionais

- o que
- sistema imprime

como

rapidamente e confidencialmente

Conceitos de Modelagem

Aspectos Funcionais e Não-Funcionais

■

■

■

■

■

■

Conceitos de Modelagem

Aspectos Funcionais e Não-Funcionais

■

■

■

■

■

cache

Conceitos de Modelagem

Aspectos Funcionais e Não-Funcionais

- *rationale*
-
-

Conceitos de Modelagem

Ambiguidade, Exatidão e Precisão



Conceitos de Modelagem

Ambiguidade, Exatidão e Precisão

- *ambiguity*

ambíguo ambiguous

- *bugs*

- *stakeholders*

Conceitos de Modelagem

Ambiguidade, Exatidão e Precisão



principais **toda**



stakeholders



Conceitos de Modelagem

Ambiguidade, Exatidão e Precisão

■ *accuracy*

precision

exato *accurate*

preciso *precise*

Conceitos de Modelagem

Ambiguidade, Exatidão e Precisão

accuracy

precision



(a)



(b)



(c)



(d)

Conceitos de Modelagem

Ambiguidade, Exatidão e Precisão



Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



- *stakeholders*

- *stakeholders*

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*

- *Views* *Viewpoints*

view

viewpoint

view

- *view*

viewpoint

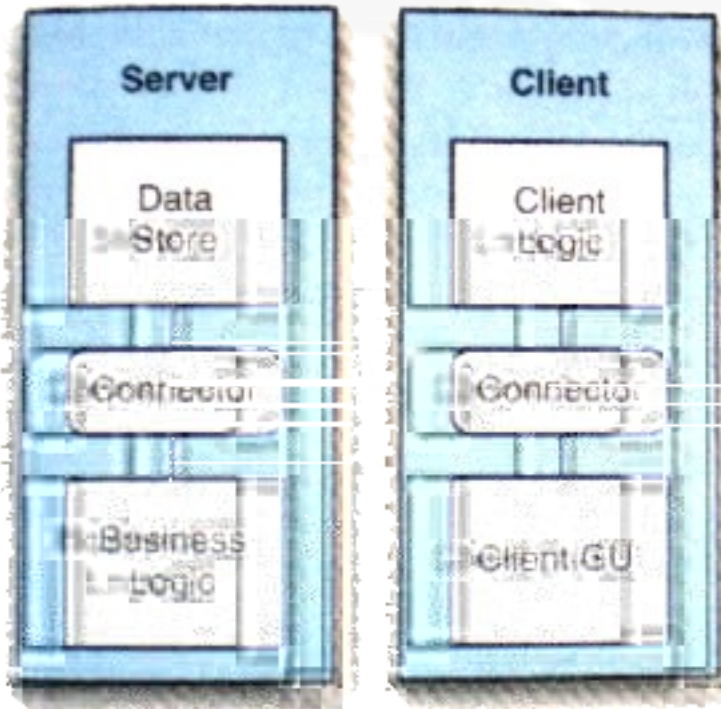
- *viewpoint*

view

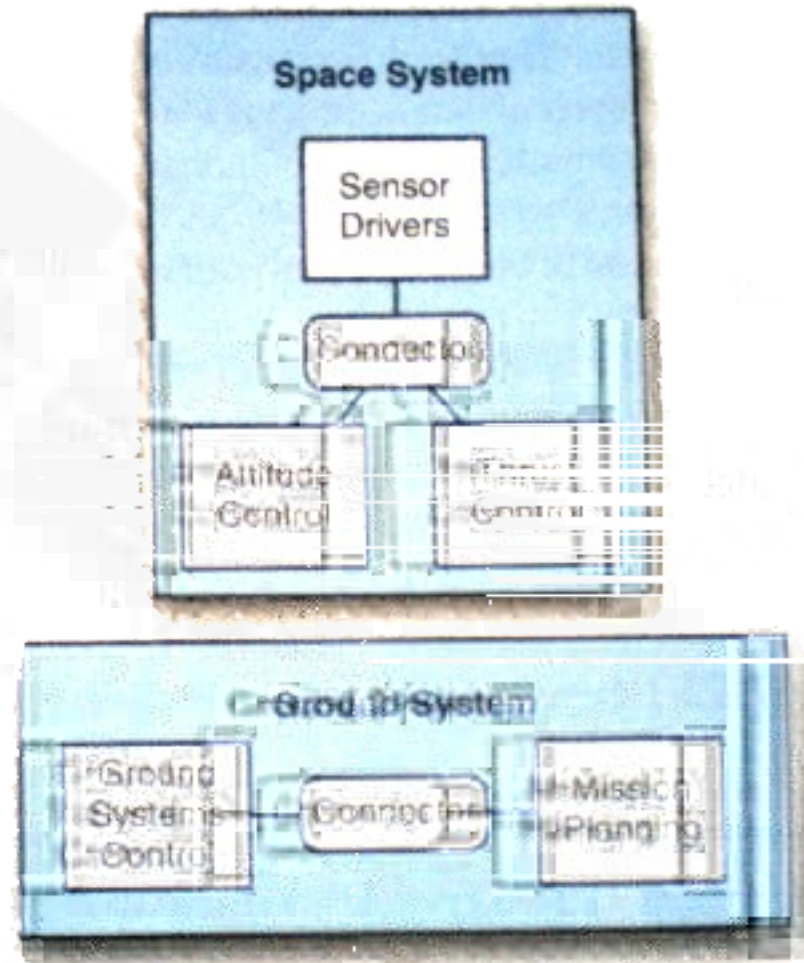
Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*

- *Deployment View*



(a)



(b)

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*

- *viewpoints*

- *Logical Viewpoint*

software

- *Physical Viewpoint*

hardware

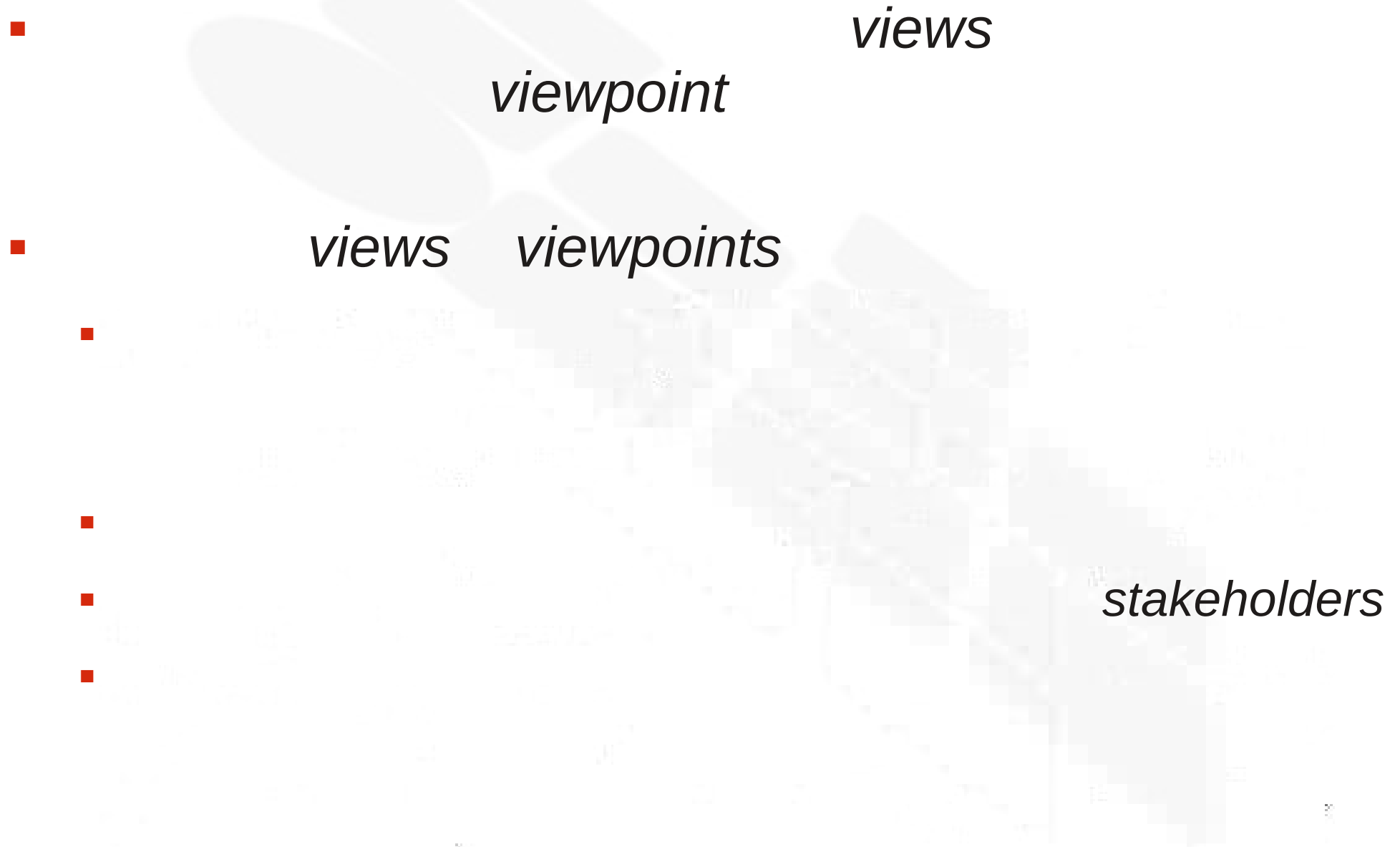
- *Deployment Viewpoint*

- *Concurrency Viewpoint*
multi-threading

- *Behavioral Viewpoint*

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



views



views

■ *Views*



views



Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



views

hosts

hosts

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



views

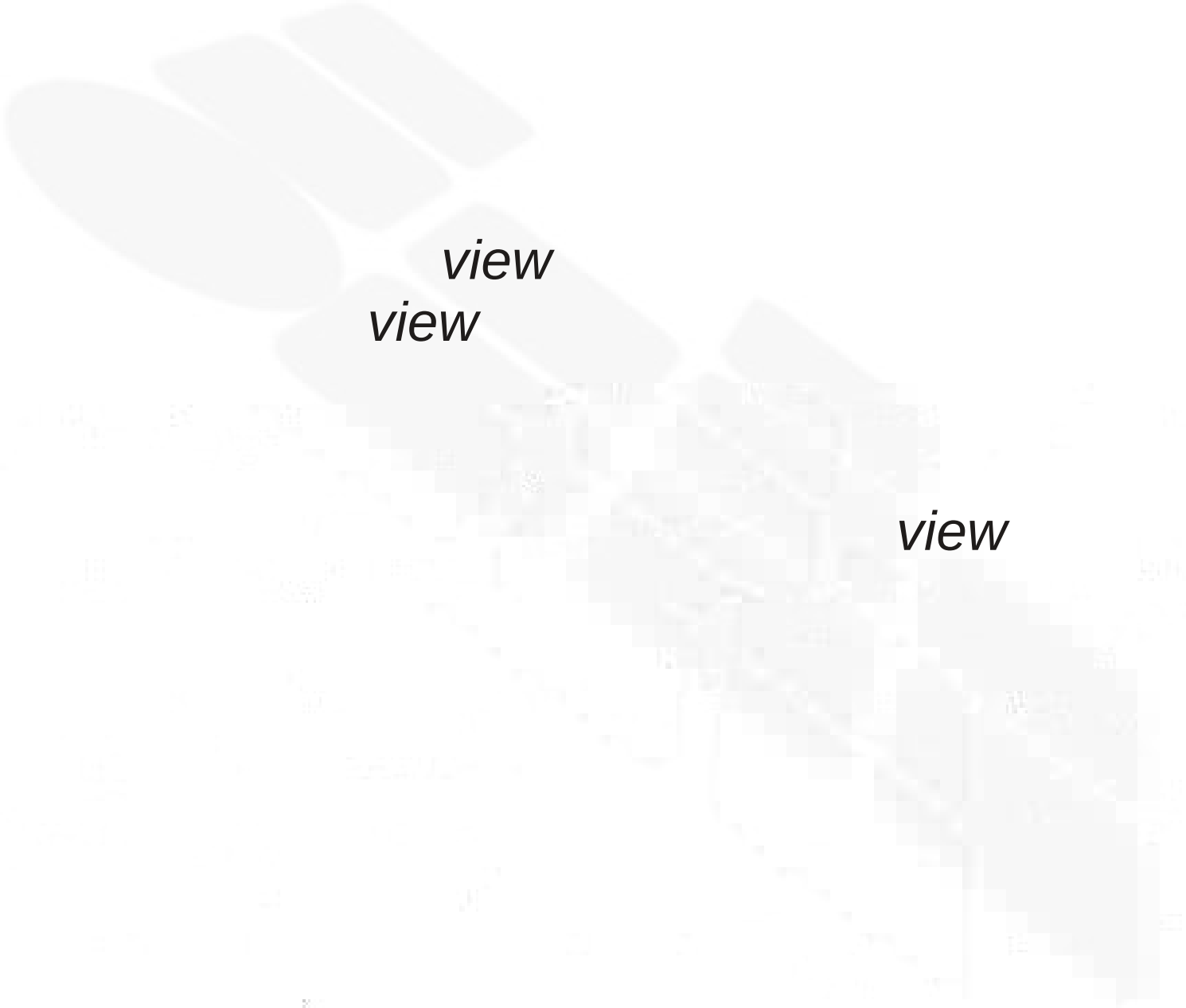
view

view

views

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



view
view

view

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



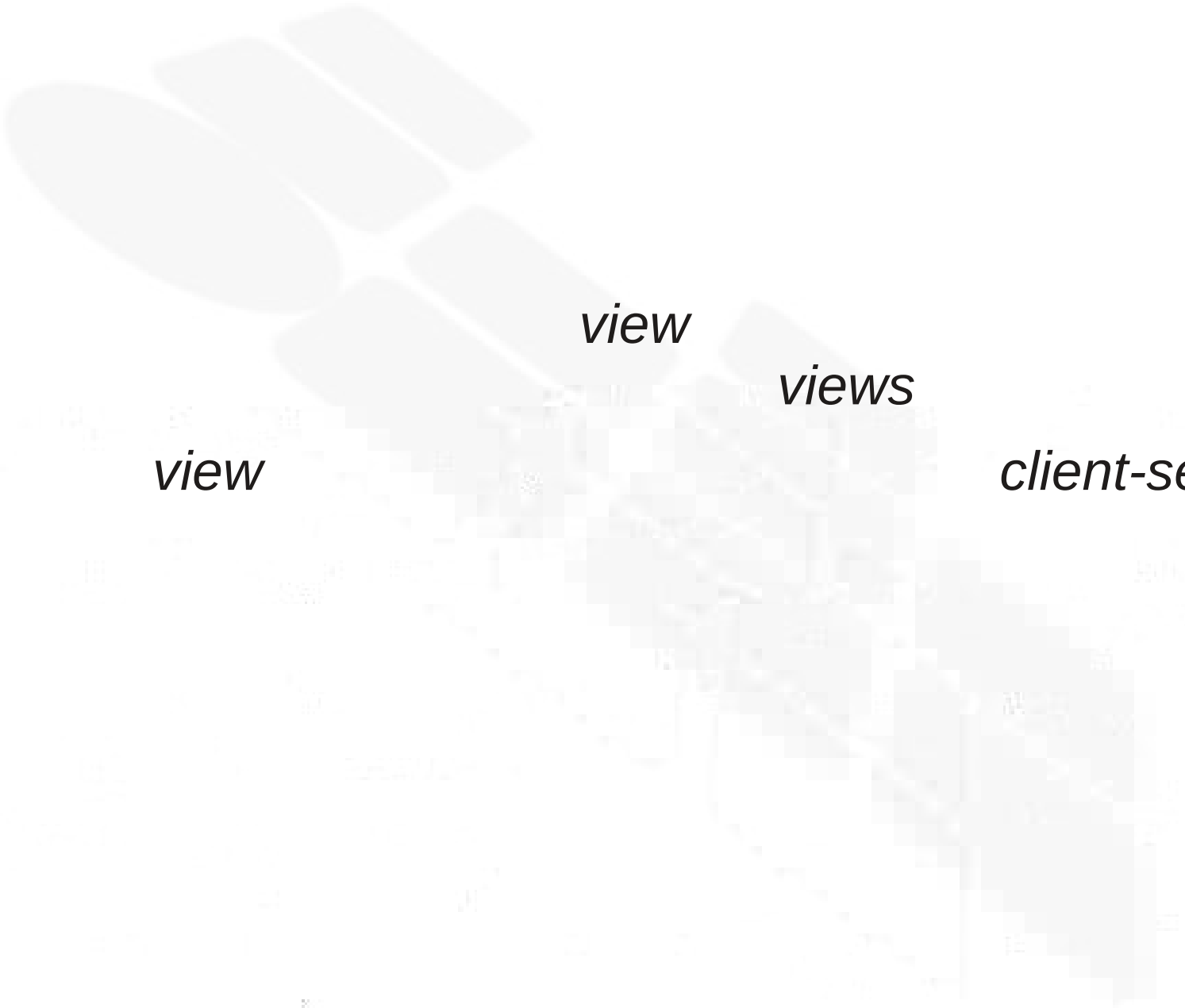
views



statecharts

Modelagem Complexa

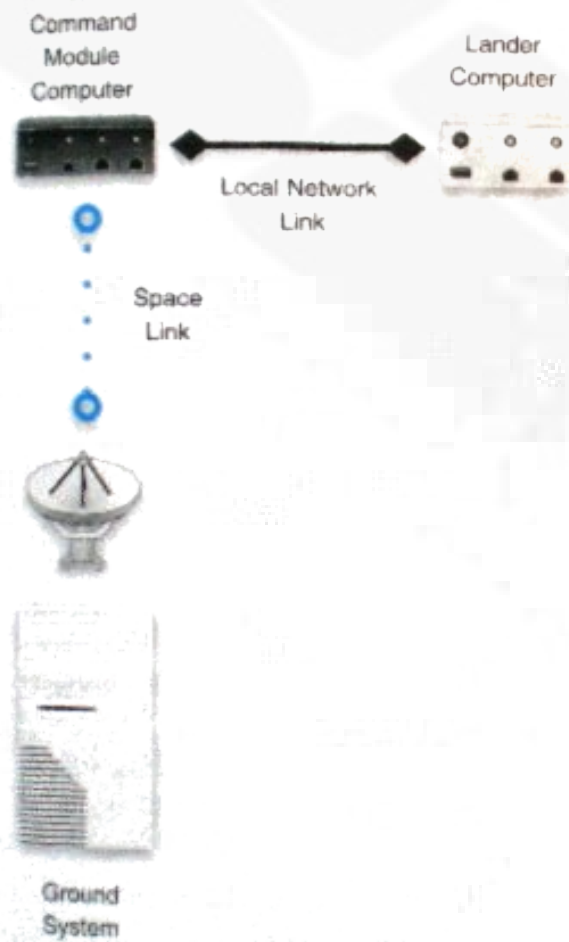
Múltiplos Conteúdos – Multiplas *views*



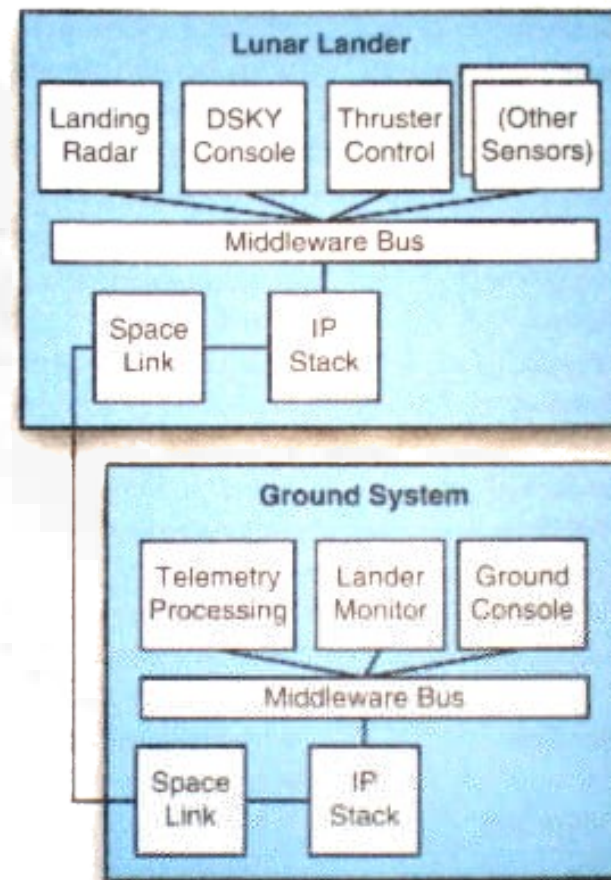
-
-
- *view*
- *view* *views* *client-server view*

Modelagem Complexa

Múltiplos Conteúdos – Multiplas *views*



(a)



(b)

Técnicas Específicas de Modelagem



Técnicas Específicas de Modelagem

- - - - *Unified Modeling Language UML*
 - *Architecture Description Languages ADLs*
 - - *Darwin*
 - *Rapide*
 - *Wright*
 - -
 -
 - *Architecture Analysis and Design Language*

Técnicas Específicas de Modelagem

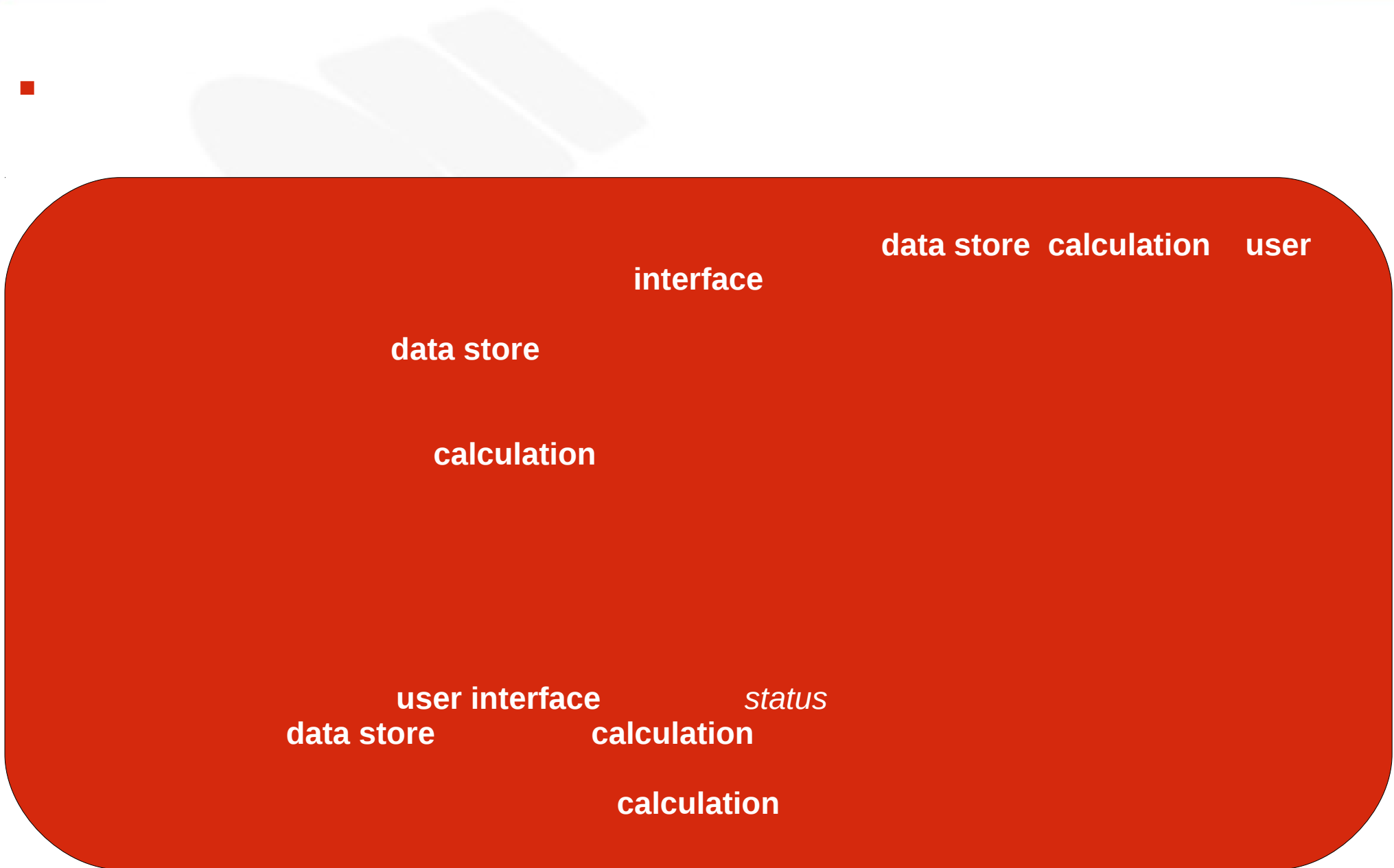
Técnicas Genéricas



Escopo e Propósito	
Elementos Básicos	
Estilo	
Aspectos Estáticos e Dinâmicos	
Modelagem Dinâmica	
Aspectos Não-Funcionais	
Ambiguidade	<i>Templates</i>
Exatidão	
Precisão	
<i>Viewpoints</i>	
Consistência entre <i>Views</i>	

Técnicas Específicas de Modelagem

Técnicas Genéricas



Técnicas Específicas de Modelagem

Técnicas Genéricas



Técnicas Específicas de Modelagem

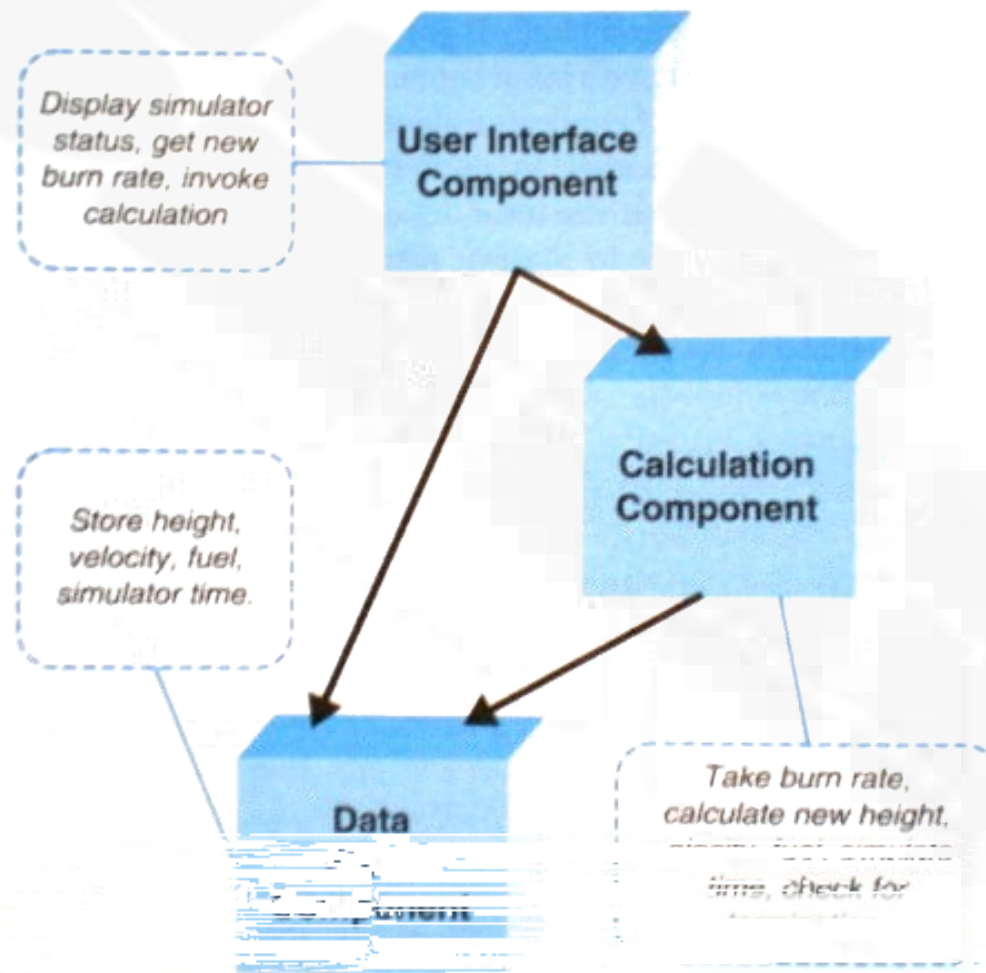
Técnicas Genéricas



Escopo e Propósito		
Elementos Básicos	<i>splines</i>	<i>clip-art</i>
Estilo		
Aspectos Estáticos e Dinâmicos		
Modelagem Dinâmica		
Aspectos Não-Funcionais		
Ambiguidade		
Exatidão		
Precisão	<i>slide</i>	
Viewpoints <i>viewpoints</i>		<i>viewpoint</i>
Consistência entre Views		

Técnicas Específicas de Modelagem

Técnicas Genéricas



Técnicas Específicas de Modelagem

Técnicas Genéricas



Técnicas Específicas de Modelagem

Técnicas Genéricas

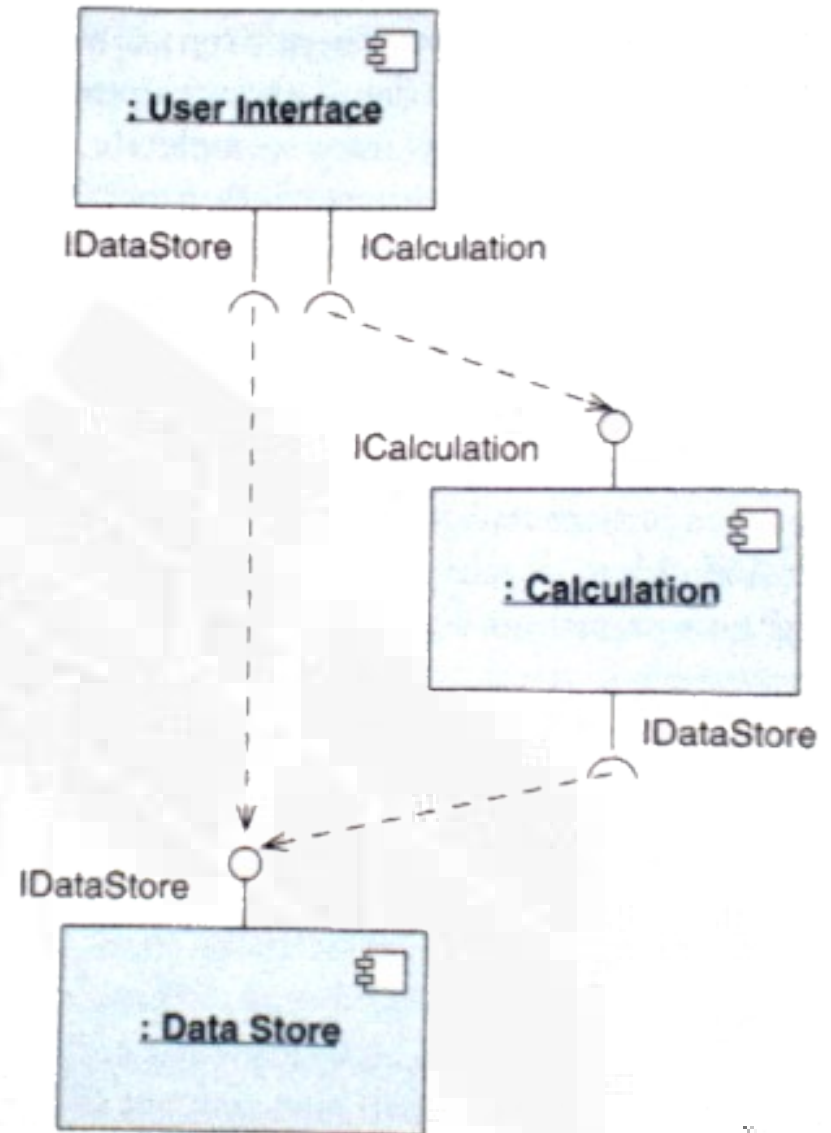
■ *Unified Modeling Language UML*

Escopo e Propósito	<i>software</i>		
Elementos Básicos		<i>composite</i>	<i>OCL</i>
Estilo	<i>OCL</i>		
<i>viewpoints</i>			
Aspectos Estáticos e Dinâmicos			
Modelagem Dinâmica			
Aspectos Não-Funcionais			
Ambiguidade	<i>UML</i> <i>profiles UML</i>	<i>stereotypes</i>	<i>tagged values</i>
Exatidão	<i>OCL</i>		
Precisão			
<i>Viewpoints</i>	<i>viewpoint</i>		
Consistência entre <i>Views</i>	<i>OCL</i>		

Técnicas Específicas de Modelagem

Técnicas Genéricas

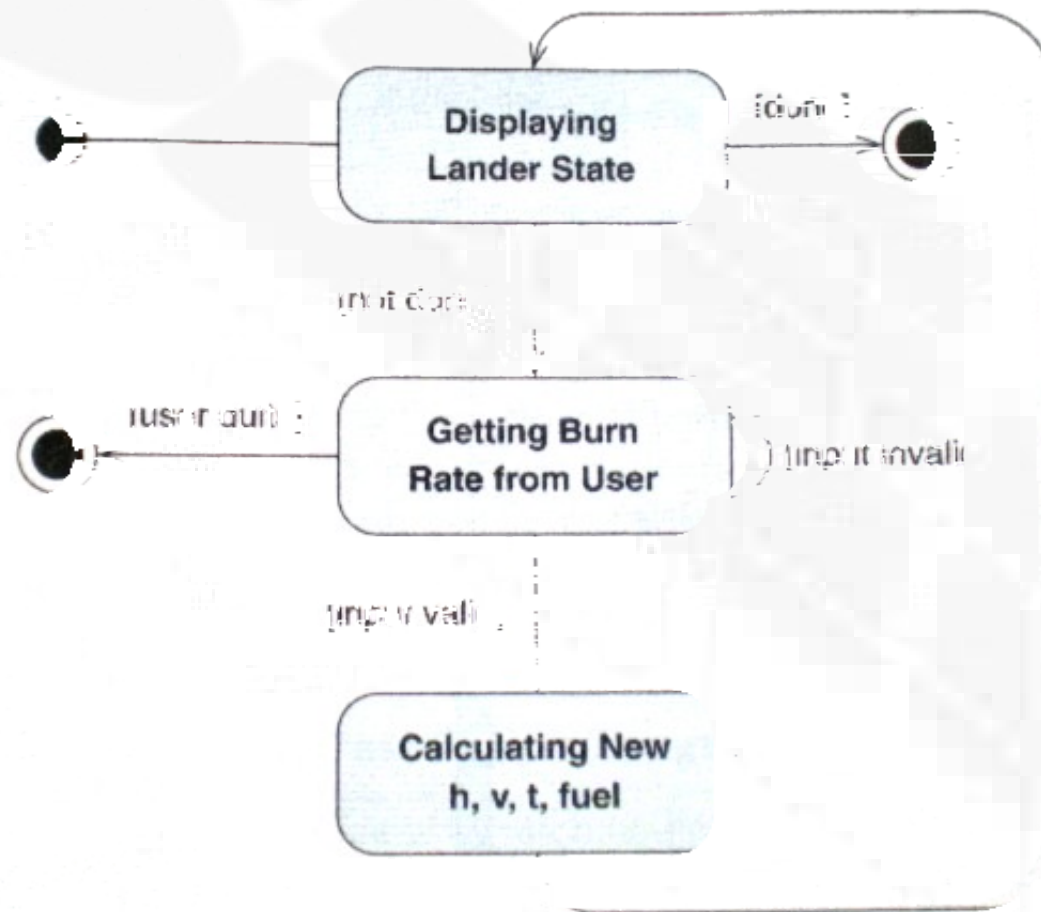
- *Unified Modeling Language*



Técnicas Específicas de Modelagem

Técnicas Genéricas

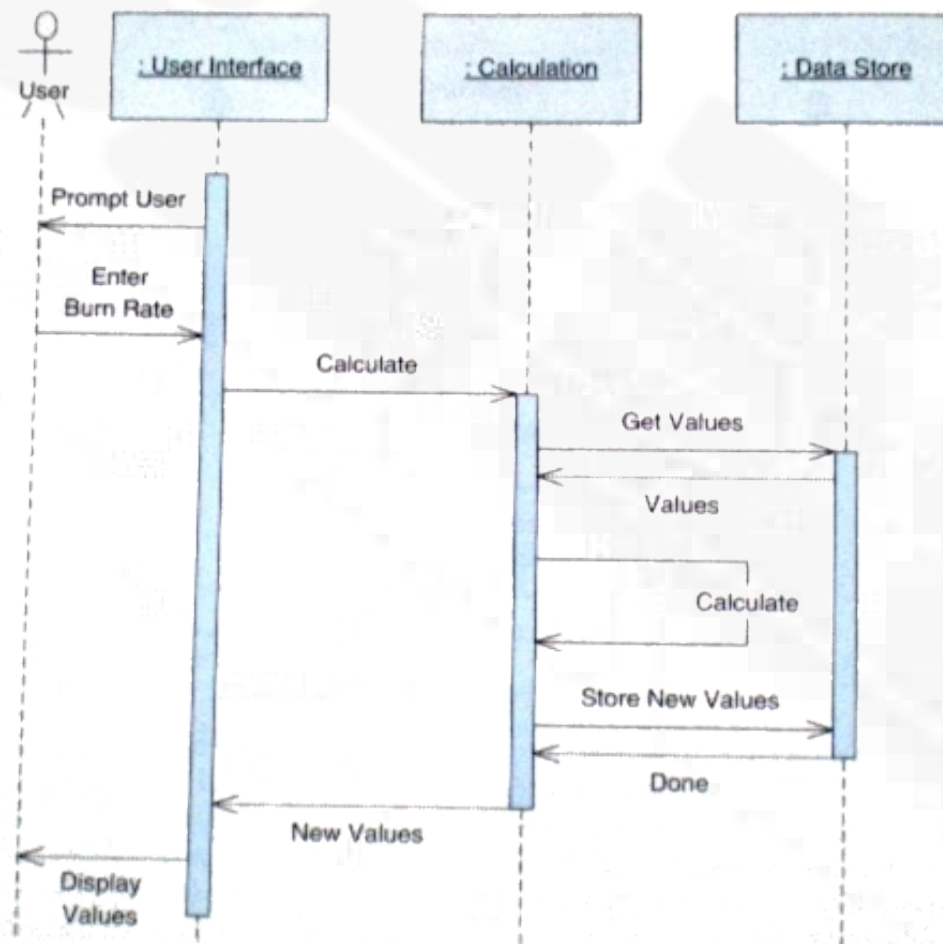
- *Unified Modeling Language*



Técnicas Específicas de Modelagem

Técnicas Genéricas

- *Unified Modeling Language*



Técnicas Específicas de Modelagem

Técnicas Genéricas

- *Unified Modeling Language*



- *Profiles*

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

- - *Darwin*
 - *Rapide*
 - *Wright*

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

- *Darwin*

- *ADL*

- *required ports*
 - *provided*

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Darwin

Escopo e Propósito	
Elementos Básicos	<i>required provided</i>
Estilo	
Aspectos Estáticos e Dinâmicos	<i>views lazy/dynamic instantiation/binding</i>
Modelagem Dinâmica	
Aspectos Não-Funcionais	
Ambiguidade	<i>stakeholders</i>
Exatidão	<i>pi-calculus</i>
Precisão	
<i>Viewpoints</i>	
Consistência entre Views	

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Darwin

```
component DataStore{
    provide landerValues;
}

component Calculation{
    require landerValues;
    provide calculationService;
}

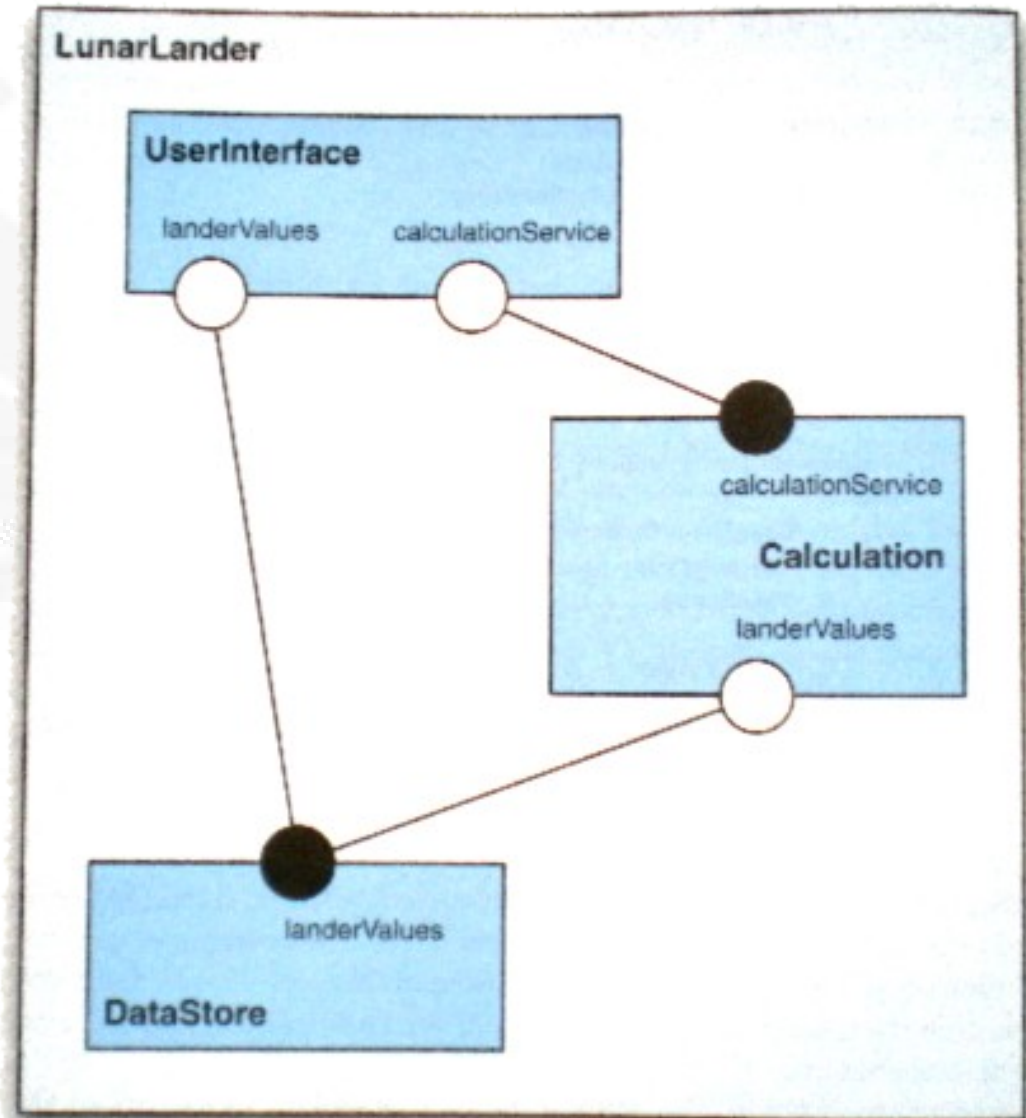
component UserInterface{
    require calculationService;
    require landerValues;
}

component LunarLander{
    inst
        U: UserInterface;
        C: Calculation;
        D: DataStore;
    bind
        C.landerValues -- D.landerValues;
        U.landerValues -- D.landerValues;
        U.calculationService -- C.calculationService;
}
```

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

■ *Darwin*



Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Darwin

```
component WebServer{
    provide httpService;
}

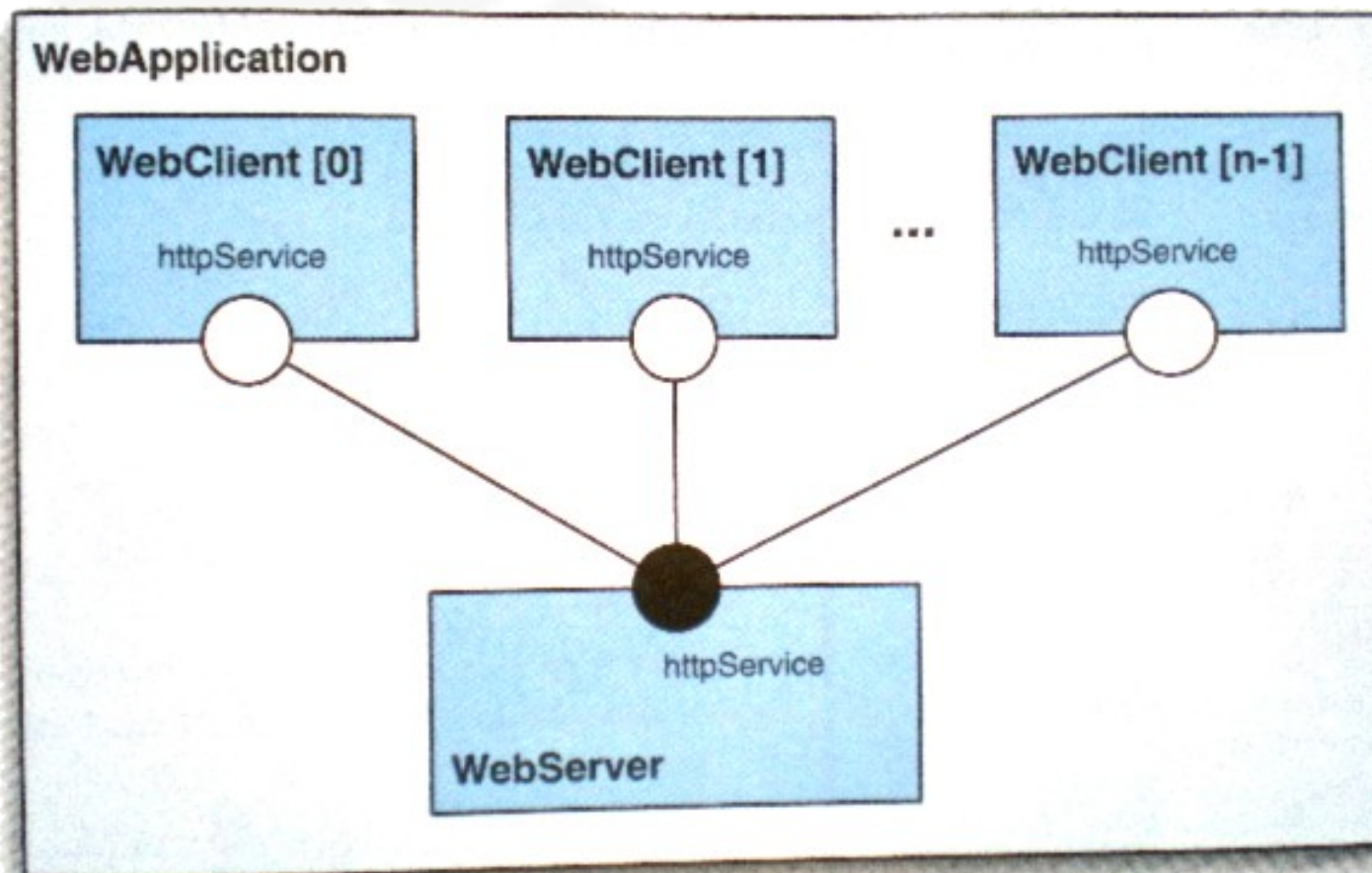
component WebClient{
    require httpService;
}

component WebApplication(int numClients){
    inst S: WebServer;
    array C[numClients]: WebClient;
    forall k:0..numClients-1{
        inst C[k] @ k;
        bind C[k].httpService -- S.httpService;
    }
}
```

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Darwin



Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

■ *Rapide*

Escopo e Propósito	
Elementos Básicos	
Estilo	
Aspectos Estáticos e Dinâmicos	
Modelagem Dinâmica	<i>run-time</i>
Aspectos Não-Funcionais	
Ambiguidade	
Exatidão	
Precisão	
<i>Viewpoints</i>	<i>viewpoint</i>
Consistência entre <i>Views</i>	

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Rapide

```
type DataStore is interface
  action in SetValues();
  out NotifyNewValues();
  behavior
  begin
    SetValues => NotifyNewValues();
  end DataStore;

type Calculation is interface
  action in SetBurnRate();
  out DoSetValues();
  behavior
  action CalcNewState();
  begin
    SetBurnRate => CalcNewState(); DoSetValues();
  end Calculation;
```

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Rapide

```
type Player is interface
  action out DoSetBurnRate();
  in NotifyNewValues();
  behavior
    TurnsRemaining : var integer := 1;
    action UpdateStatusDisplay();
    action Done();
  begin
    (start or UpdateStatusDisplay) where \
      ($TurnsRemaining > 0) => \
      if ( $TurnsRemaining > 0 ) then \
        TurnsRemaining := $TurnsRemaining - 1; \
        DoSetBurnRate(); \
      end if;;
    NotifyNewValues => UpdateStatusDisplay();
    UpdateStatusDisplay where $TurnsRemaining == 0 \
      => Done();
  end UserInterface;
```

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Rapide

```
architecture lander() is
  P1, P2 : Player;
  C : Calculation;
  D : DataStore;
connect
  P1.DoSetBurnRate to C.SetBurnRate;
  P2.DoSetBurnRate to C.SetBurnRate;
  C.DoSetValues to D.SetValues;
  D.NotifyNewValues to P1.NotifyNewValues();
  D.NotifyNewValues to P2.NotifyNewValues();
end LunarLander;
```

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Rapide

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Wright

Escopo e Propósito		
Elementos Básicos	<i>ports roles</i>	<i>attachments</i>
Estilo		
Aspectos Estáticos e Dinâmicos		
Modelagem Dinâmica		
Aspectos Não-Funcionais		
Ambiguidade	<i>Wright</i>	<i>CSP</i>
Exatidão <i>deadlocks</i>	<i>CSP</i>	
Precisão		
Viewpoints		
Consistência entre Views		<i>CSP</i>

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Wright

```
Component DataStore
  Port getValues (behavior specification)
  Port storeValues (behavior specification)
  Computation (behavior specification)

Component Calculation
  Port getValues (behavior specification)
  Port storeValues (behavior specification)
  Port calculate (behavior specification)
  Computation (behavior specification)

Component UserInterface
  Port getValues (behavior specification)
  Port calculate (behavior specification)
  Computation (behavior specification)

Connector Call
  Role Caller =  $\overline{call} \rightarrow \overline{return} \rightarrow Caller[]$  §
  Role Callee =  $call \rightarrow \overline{return} \rightarrow Callee[]$  §
  Caller.call  $\rightarrow$  Callee.call  $\rightarrow$  Glue
  Glue =  $[] Callee.return \rightarrow \overline{Caller.return} \rightarrow Glue$ 
  [] §
```

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Wright

```
Configuration LunarLander
Instances
  DS : DataStore
  C : Calculation
  UI : UserInterface
  CtoUIgetValues, CtoUIstoreValues, UItoC, UItoDS : Call

Attachments
  C.getValues as CtoUIgetValues.Caller
  DS.getValues as CtoUIgetValues.Callee

  C.storeValues as CtoUIstoreValues.Caller
  DS.storeValues as CtoUIstoreValues.Callee

  UI.calculate as UItoC.Caller
  C.calulate as UItoC.Callee

  UI.getValues as UItoDS.Caller
  DS.getValues as UItoDS.Callee
End LunarLander.
```



Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

-
-
-
- *Architecture Analysis and Design Language*

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Koala

Escopo e Propósito
Elementos Básicos <i>diversity interfaces switches multiplexers</i>
Estilo
Aspectos Estáticos e Dinâmicos
Modelagem Dinâmica <i>run-time</i>
Aspectos Não-Funcionais
Ambiguidade
Exatidão
Precisão
<i>Viewpoints</i>
Consistência entre <i>Views</i> <i>views</i>

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

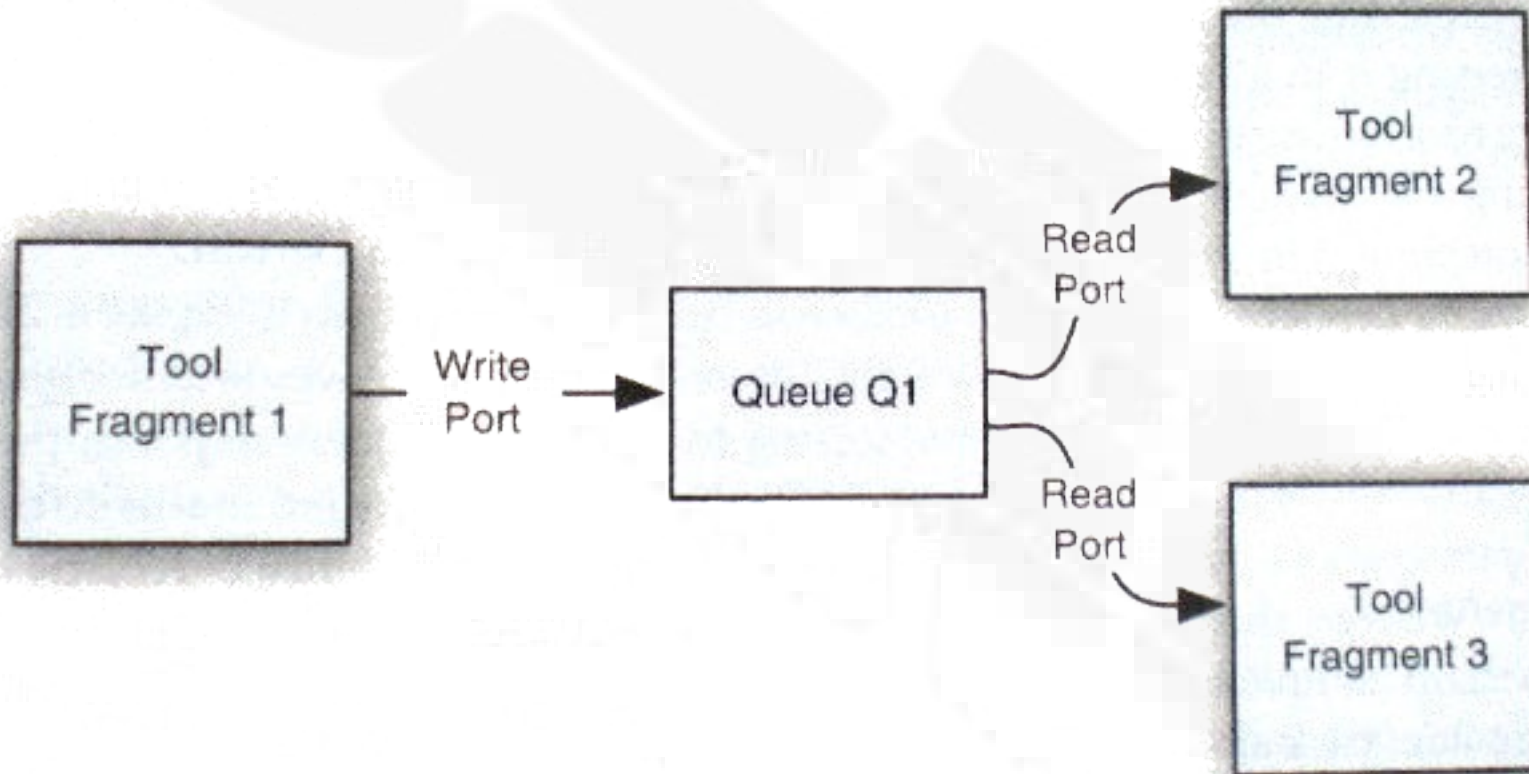
Weaves

Escopo e Propósito	<i>Weaves</i>
Elementos Básicos	<i>queues</i>
Estilo	<i>Weaves</i>
Aspectos Estáticos e Dinâmicos	
Modelagem Dinâmica	
Aspectos Não-Funcionais	
Ambiguidade	<i>Weaves</i>
Exatidão	<i>Weaves</i>
Precisão	
<i>Viewpoints</i>	
Consistência entre <i>Views</i>	<i>views</i>

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

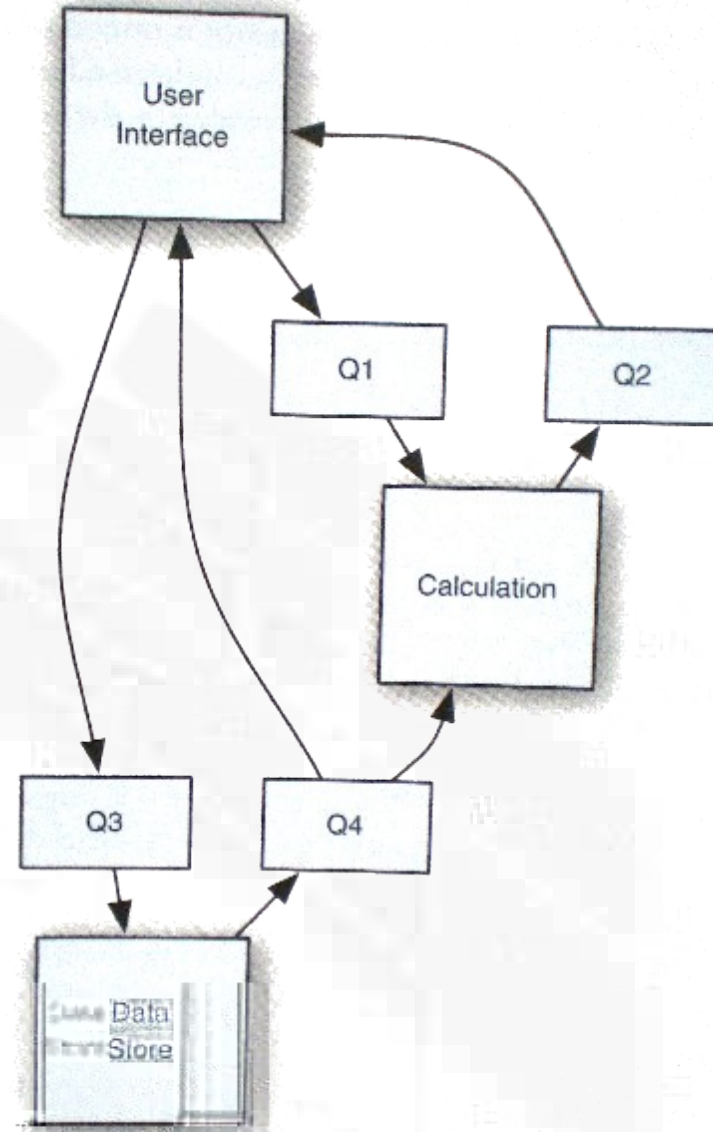
Weaves



Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)

Weaves



Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)



Escopo e Propósito	<i>hardware</i>	<i>software</i>	
Elementos Básicos	<i>hardware</i>	<i>software</i>	<i>threads</i>
Estilo			
Aspectos Estáticos e Dinâmicos			
Modelagem Dinâmica			
Aspectos Não-Funcionais			
Ambiguidade			
Exatidão			
Precisão			
Viewpoints	<i>viewpoints</i>	<i>hardware</i>	<i>software</i> <i>threads</i>
Consistência entre Views	<i>Open Source AADL Tool Environment</i>		<i>plugins</i>

Técnicas Específicas de Modelagem

Architecture Description Languages (ADLs)



Sandro Santos Andrade

**Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas**

