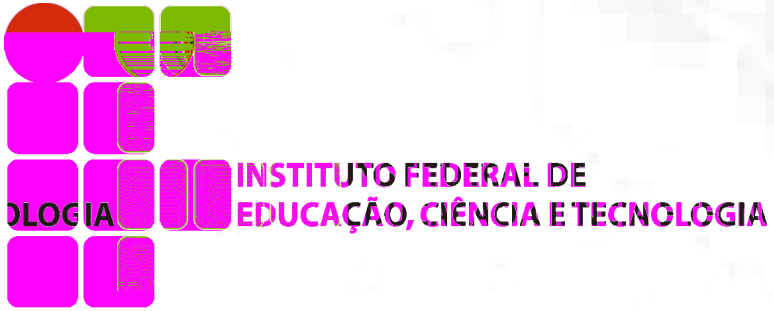
INF016 – Arquitetura de Software

07 - Análise

Sandro Santos Andrade
sandroandrade@ifba.edu.br

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



Introdução

- Modelos mais formais de arquiteturas de *software* trazem um conjunto de benefícios:
  - Fazem com que o arquiteto resolva problemas que seriam provavelmente ignorados
  - Permitem uma comunicação mais precisa entre os *stakeholders*
  - Constituem um modelo sólido para a construção, implantação, execução e evolução do *software*
  - Apresentam mais detalhes sobre a arquitetura do que modelos informais e, portanto, questões podem ser analisadas de forma mais precisa

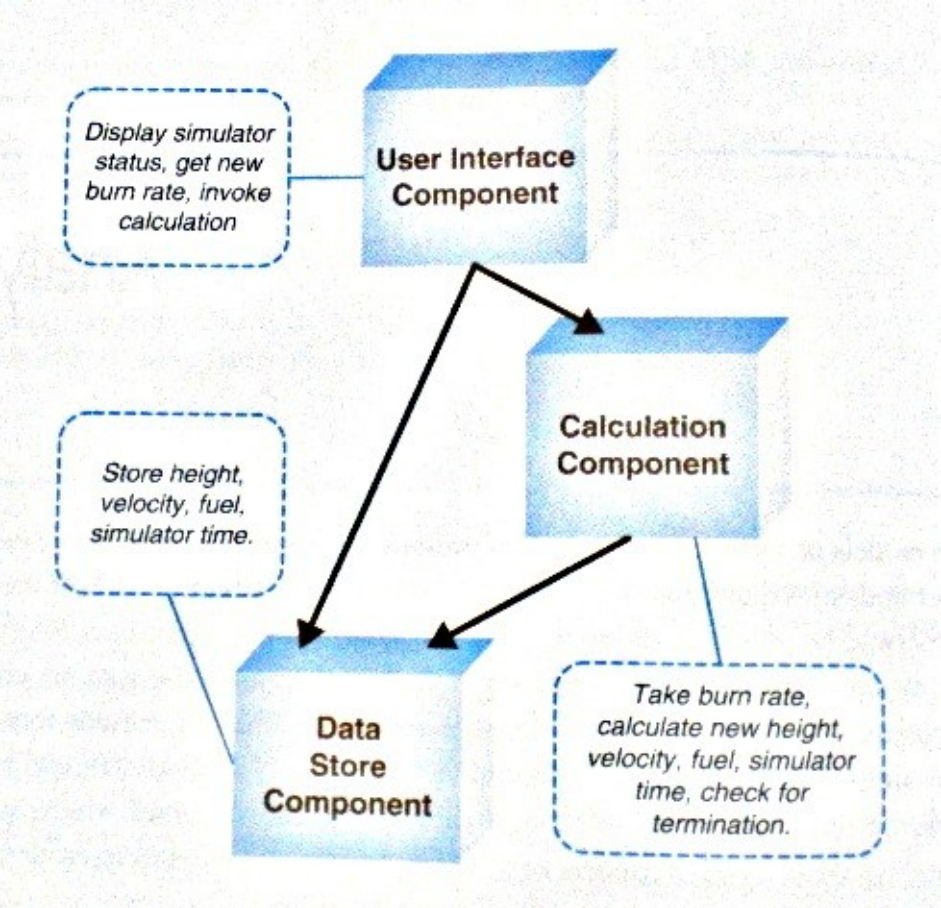
Introdução

Análise Arquitetural: atividade de descoberta de propriedades importantes do sistema a partir dos seus modelos arquiteturais

- Tais propriedades ajudam a identificar decisões inapropriadas ou incorretas antes de serem propagadas para o sistema
- Considerações:
  - Quais perguntas sobre a arquitetura devem ser realizadas ?
  - Porque e como realizar a pergunta ?
  - Como garantir que serão apropriadamente respondidas a partir da interpretação e extrapolação do modelo ?

Introdução

- Exemplo – o diagrama ao lado:
  - Pode ajudar o arquiteto a obter informações sobre o sistema
  - Pode ser informalmente analisado para garantir que o escopo do projeto é apropriado
  - Não informa sobre como os componentes interagem, onde são implantados e a natureza das suas interações



Introdução

- Exemplo – o diagrama ao lado:
  - Pode garantir que os componentes serão integrados conforme descritos no modelo
  - Permite que componentes individuais sejam analisados para verificar a possível compatibilidade com componentes *COTS*
  - Pode dar suporte a uma ferramenta para geração automática de código

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

type Calculation is interface
  action in SetBurnRate();
            DoSetValues();
  behavior
            action CalcNewState();
  begin
            SetBurnRate CalcNewState DoSetValues();
end Calculation;

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;
```

Introdução

- O que será visto sobre Análise Arquitetural
  - Metas (para que ?)
  - Escopos (quanto ?)
  - Aspectos (o que ?)
  - Grau de Formalidade
  - Tipo
  - Grau de Automação
  - *Stakeholders*
  - Técnicas

Análise Arquitetural: Metas

- Variedade de metas:
  - Estimativa do tamanho, complexidade e custo do sistema
  - Verificação de conformidade com restrições e diretrizes de projeto
  - Verificação de satisfação de requisitos funcionais e não-funcionais
  - Avaliação da corretude do sistema implementado em relação à sua arquitetura documentada
  - Avaliação de oportunidades de reuso

Análise Arquitetural: Metas

- Quatro categorias podem entretanto ser identificadas:
  - Completude (*Completeness*)
  - Consistência (*Consistency*)
  - Compatibilidade (*Compatibility*)
  - Corretude (*Correctness*)

Análise Arquitetural: Metas

- Completude:
  - Completude externa:
    - Externa em relação aos requisitos do sistema. Estabelece se a arquitetura captura adequadamente todos os requisitos funcionais e não-funcionais principais do sistema
    - É uma tarefa não-trivial em sistema grandes, complexos, dinâmicos e de longa vida
    - Em tais sistemas são utilizadas diversas notações com diferentes graus de rigor e formalidade
    - Os requisitos e a arquitetura são geralmente construídos de forma incremental e, portanto, o arquiteto deve identificar pontos de avaliação da completude externa

Análise Arquitetural: Metas

- Completude:
  - Completude interna:
    - Estabelece se todos os elementos do sistema foram totalmente capturados em relação à notação de modelagem utilizada e ao sistema sendo projetado
    - A completude interna em relação à notação garante que o modelo inclui todas as informações demandadas pelas regras sintáticas e semânticas da notação
      - Ex: na linguagem RAPIDE uma ação *out* de um componente deve ser conectada numa ação *in* de outro componente, visto que conectores não são explicitamente declarados
    - A completude interna em relação ao sistema sendo projetado requer a identificação de componentes, conectores, interfaces, protocolos e caminhos de interação e dependências ausentes na arquitetura

Análise Arquitetural: Metas

- Consistência:
  - Inconsistência de Nome
    - Ocorre em relação a componentes e conectores ou nos seus elementos constituintes, tais como os serviços exportados por um componente
    - Múltiplos elementos/serviços do sistema podem ter nomes similares
    - Nas linguagens de programação características tais como ligação estática, checagem de tipos e comunicação síncrona ponto-a-ponto ajudam a detectar tais inconsistências
    - Arquitetura desacopladas (*publish/subscribe* ou *broadcast* de eventos), adaptáveis e dinâmicas tornam a detecção dessas inconsistências um trabalho mais difícil

Análise Arquitetural: Metas

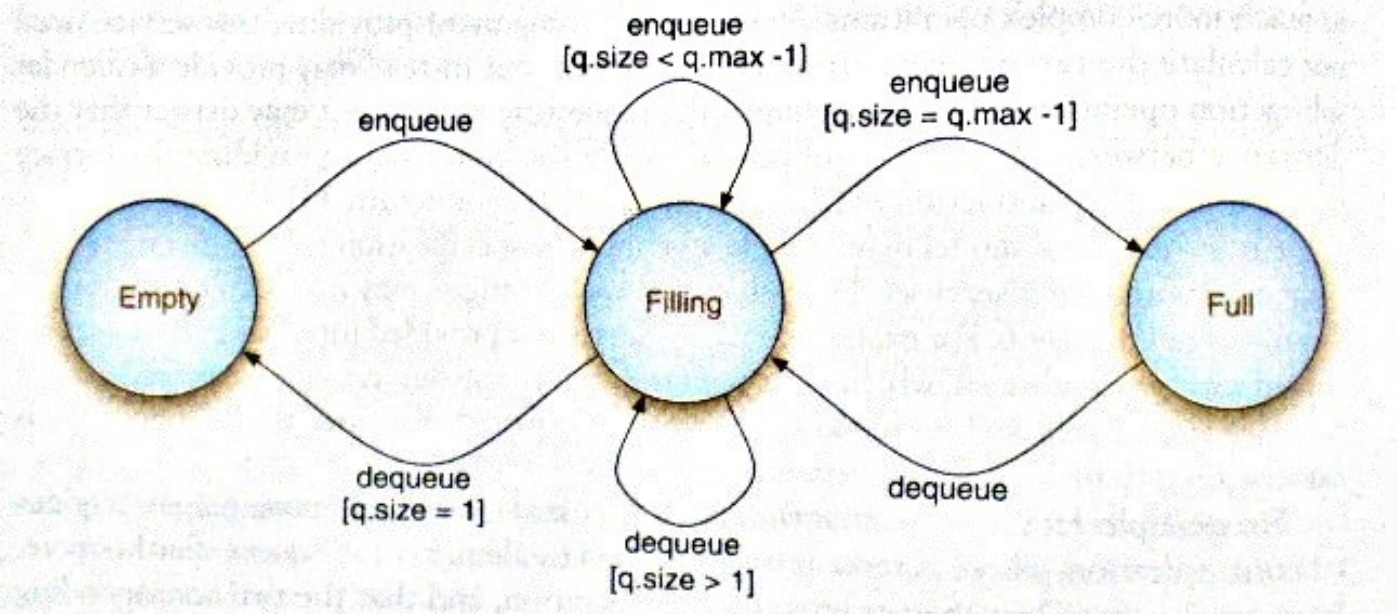
- Consistência:
  - Inconsistência de Interface
    - Contém todas as inconsistências de nome
    - O nome de um serviço requerido pode ser o mesmo de um serviço provido porém os parâmetros (tipos e quantidade) e retorno podem ser diferentes
    - Exemplo:
      - ReqInt: getSubQ (Natural first, Natural last, Boolean remove) returns FIFOQueue;
      - ProvInt1: getSubQ (Index first, Index last) returns FIFOQueue;
      - ProvInt2: getSubQ (Natural first, Natural last, Boolean remove) returns Queue;

Análise Arquitetural: Metas

- Consistência:
  - Inconsistência Comportamental
    - Ocorre entre componentes que requerem e disponibilizam serviços cujos nomes e interfaces são compatíveis porém seus comportamentos não são
    - Exemplo:
      - ProvInt: subtract (Integer x, Integer y) returns Integer;
      - Subtração aritmética ou de datas (427-27=331) ?
      - O modelo arquitetural pode conter informações comportamentais, por exemplo pré- e pós-condições:
        - ReqInt:front() returns Integer;
          - *precondition*: q.size >= 0;
          - *postcondition*: ~q.size = q.size;
        - ProvInt: front() returns Integer;
          - *precondition*: q.size >= 1;
          - *postcondition*: ~q.size = q.size -1;

Análise Arquitetural: Metas

- Consistência:
  - Inconsistência de Interação
    - Ocorre quando as operações disponibilizadas por um componente são acessadas de um modo que viola certas restrições de interação, tais como a ordem na qual as operações devem ser invocadas (protocolo)

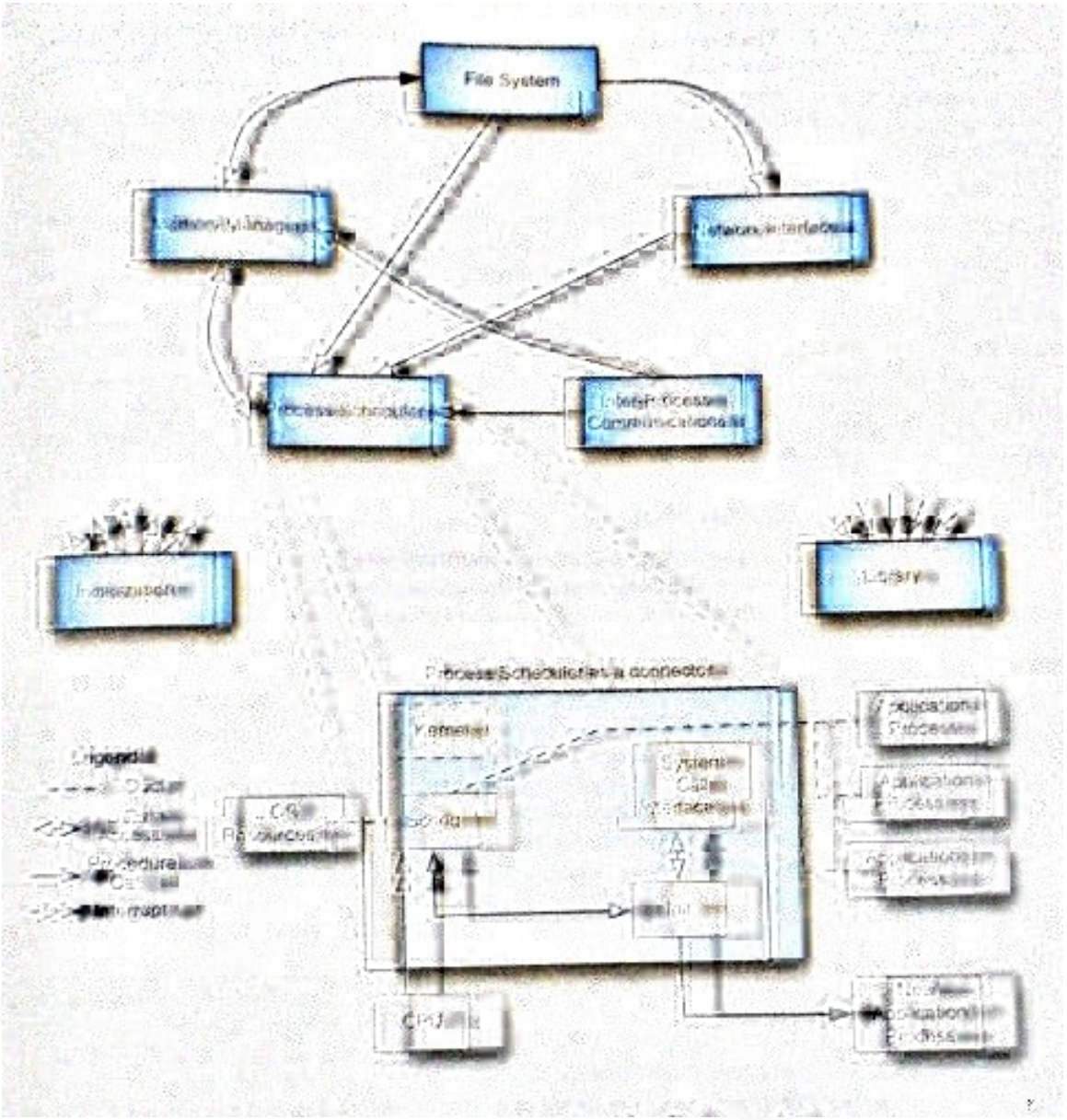


Análise Arquitetural: Metas

- Consistência:
  - Inconsistência de Refinamento
    - Surgem devido ao fato que arquiteturas são frequentemente capturadas em múltiplos níveis de abstração
    - Para que um modelo seja analisado em relação a inconsistências de refinamento, três condições devem ser atendidas:
      - Elementos do modelo de mais alto nível devem estar presentes no modelo de mais baixo nível
      - Propriedades chave do modelo de mais alto nível devem ter sido preservadas no modelo de mais baixo nível (decisões não foram omitidas, modificadas ou violadas)
      - Os detalhes novos introduzidos no modelo de mais baixo nível são consistentes com os detalhes do modelo de mais alto nível

Análise Arquitetural: Metas

- Consistência:
  - Inconsistência de Refinamento

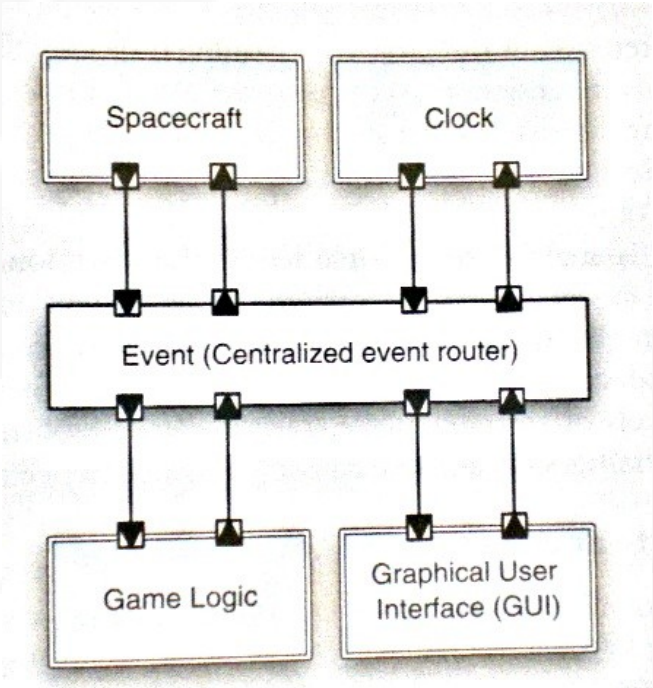


Análise Arquitetural: Metas

- Compatibilidade:
  - Propriedade externa que verifica se o modelo arquitetural está em conformidade com as restrições e diretrizes impostas pelo estilo arquitetural, arquitetura de referência ou padronização arquitetural em uso
  - Se o modelo é formal (ex: numa arquitetura de referência) verificar a compatibilidade é trivial
  - Pode requerer consistência de refinamento nos casos de instanciações de arquiteturas de referência

Análise Arquitetural: Metas

- Compatibilidade:



  - C2, *Blackboard* ou *Event-Based* ?
  - Demanda mais informação, conhecimento tácito e uso combinado de mais de uma técnica de análise

Análise Arquitetural: Metas

- Corretude:
  - Propriedade externa que verifica se as decisões arquiteturais de projeto realizam totalmente a especificação externa do sistema (corretude arquitetural)
  - A implementação é correta em relação à arquitetura se a implementação captura e realiza todas as decisões principais de projeto que compõem a arquitetura (corretude de implementação)
  - A corretude é relativa: é o resultado da avaliação da arquitetura em relação a outro artefato
    - Ou o artefato foi criado para reificar a arquitetura ou a arquitetura foi criada para reificar o artefato

Análise Arquitetural: Escopo

- A arquitetura pode ser analisada a partir de diferentes perspectivas e níveis de abstração:
  - Avaliação de propriedades de componentes e conectores individuais, ou de seus elementos constituintes
  - Verificação de propriedades exibidas por composições de componentes e conectores
  - Propriedades identificadas na troca de dados entre elementos
  - Comparação entre modelos do mesmo sistema (em diferentes níveis) ou modelos de sistemas diferentes

Análise Arquitetural: Escopo

- Escopos possíveis:
  - Componente-Conector
  - Sistema-Subsistema
  - Troca de Dados
  - Arquiteturas em Diferentes Níveis de Abstração
  - Comparação de Duas ou mais Arquiteturas

Análise Arquitetural: Escopo

- Componente-Conector
  - O tipo mais simples de análise no nível Componente-Conector é verificar se o componente/conector disponibiliza o serviço dele esperado
  - Verificação de consistências de nome e interface podem não ser suficientes

```
component{
  id = "datastore";
  description = "Data Store";
  interface{
    id = "datastore.getValues";
    description = "Data Store Get Values Interface";
    direction = "in";
  }
  interface{
    id = "datastore.storeValues";
    description = "Data Store Store Values Interface";
    direction = "in";
  }
}
```

```
connector Pipe =
  role Writer = write → Writer ⊓
                close → √
  role Reader =
    let ExitOnly = close → √
    in let DoRead = (read → Reader □
                     read-eof → ExitOnly)
    in DoRead ⊓ ExitOnly
  glue = let ReadOnly = Reader.read → ReadOnly □
             Reader.read-eof → Reader.close → √ □
             Reader.close → √
  in let WriteOnly = Writer.write → WriteOnly □
         Writer.close → √
  in Writer.write → glue □
     Reader.read → glue □
     Writer.close → ReadOnly □
     Reader.close → WriteOnly
```

Análise Arquitetural: Escopo

- Sistema-Subsistema
  - Mesmo que componentes e conectores individuais possuam propriedades desejadas, nada garante que a sua composição será correta ou desejada pois o inter-relacionamento entre eles pode ser complexo
  - A análise sistema-subsistema mais simples é a consideração de componentes dois a dois
  - Um próximo passo seria a consideração de diversos componentes ligados através de um único conector
  - Espera-se que uma combinação de componentes que apresentem propriedades α, β, γ etc possua uma combinação destas propriedades

Análise Arquitetural: Escopo

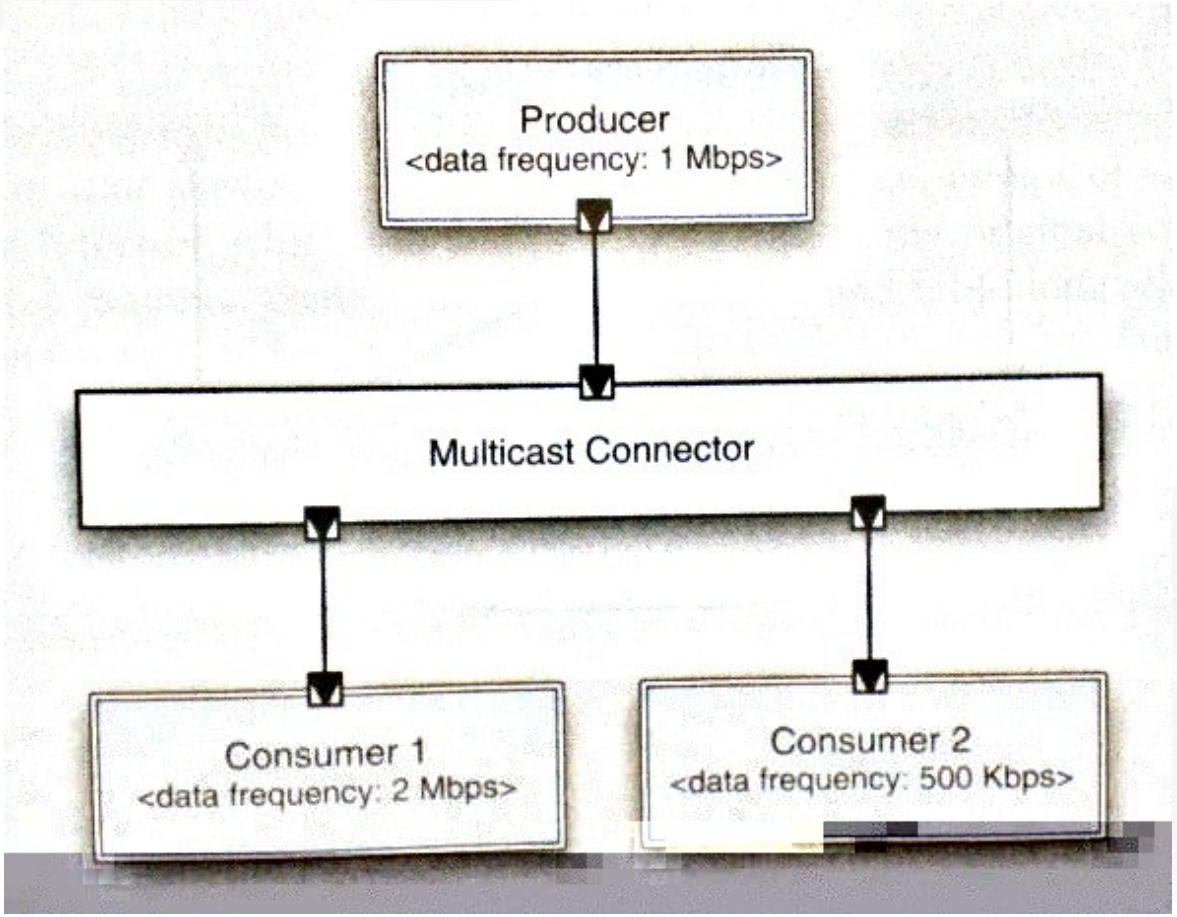
- Sistema-Subsistema
  - Entretanto, o inter-relacionamento dos componentes geralmente contribui para melhorar ou piorar esta combinação:
    - Exemplos de melhora: em sistemas críticos o desempenho pode ser sacrificado para se obter melhor segurança. Sistemas de tempo-real podem introduzir componentes para acrescentar atrasos nas requisições. Nestes casos, o sistema é maior que a soma das partes
    - Nos casos de piora geralmente este resultado não é intencional. Ex: combinação de dois componentes que assumem que possuem a *thread* de controle, modelos de concorrência incompatíveis, interação de componentes *CPU-Bound* e *Memory-Bound*, etc
    - “Síndrome do hamburguer assado com mel”

Análise Arquitetural: Escopo

- Troca de Dados
  - *Data-Intensive Systems*: computação científica, aplicações *web*, comércio eletrônico e multimídia
  - A avaliação dos elementos de dados inclui:
    - Estrutura dos dados: *typed* x *untyped*, *discrete* x *streamed*
    - Fluxo de dados no sistema: ponto-a-ponto vs *broadcast*
    - Propriedades da Troca de Dados: consistência, segurança, latência, etc

Análise Arquitetural: Escopo

- Troca de Dados
  - Exemplo:

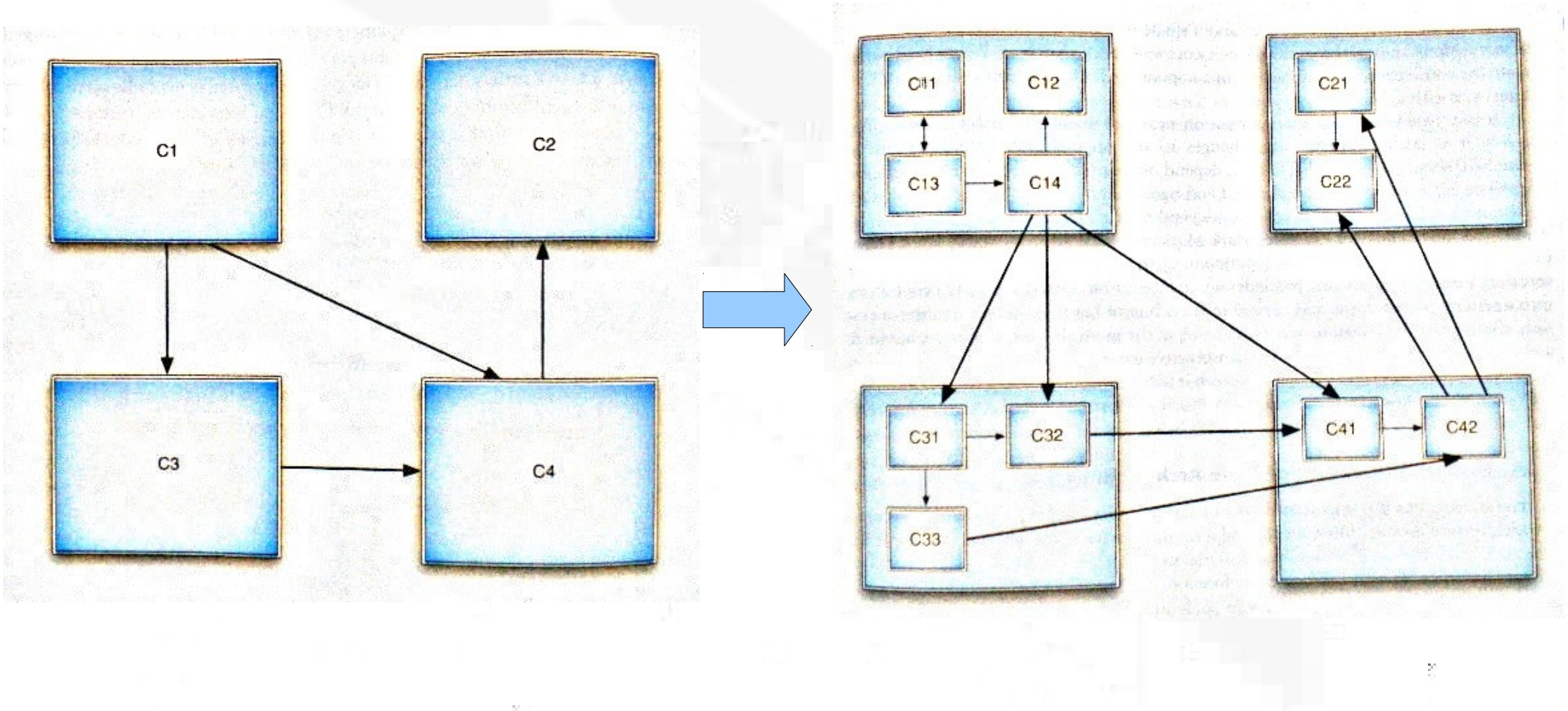


Análise Arquitetural: Escopo

- Arquiteturas em Diferentes Níveis de Abstração
  - O projeto arquitetural é geralmente feito de forma incremental
  - Exemplos de violações:
    - Invalidação direta de uma decisão arquitetural
    - Violação de restrição: interações entre componentes de mais alto nível devem ter um único destino e um único alvo
    - Tais informações não estão presentes em gráficos informais
    - Política de Refinamento
      - Ex: Extensão Conservativa: impede que um arquiteto introduza novas características. Somente é permitida a elaboração ou eliminação de características já presentes no modelo arquitetural de mais alto nível

Análise Arquitetural: Escopo

- Arquiteturas em Diferentes Níveis de Abstração



Análise Arquitetural: Escopo

- Comparação de Duas ou mais Arquiteturas
  - É interessante comparar uma arquitetura de interesse com uma outra arquitetura (de referência)
  - A arquitetura de referência pode ser:
    - Uma arquitetura que o arquiteto conhece
    - Uma arquitetura encontrada na literatura
    - Uma arquitetura de um produto relacionado
    - Uma arquitetura recuperada a partir de um sistema que possui as mesmas propriedades desejadas
  - Pode requerer a avaliação de componentes, conectores, troca de dados e composições
  - As arquiteturas podem estar em níveis de abstração diferentes

Análise Arquitetural: Aspectos

- Diferentes facetas podem ser analisadas:
  - Características Estruturais:
    - Exemplos:
      - Conectividade entre componentes e conectores
      - Presença de elementos de baixo nível em elementos *composite* de alto nível
      - Possíveis pontos de distribuição em rede
      - Potenciais arquiteturas de implantação
    - Determinam se a arquitetura é bem formada
    - Possíveis problemas:
      - Componentes ou sub-sistemas desconectados do resto da arquitetura
      - Caminhos de interação inexistente entre componentes e conectores que espera-se que se comuniquem
      - Encapsulamento de componentes e conectores que devem estar visíveis em níveis mais altos de abstração

Análise Arquitetural: Aspectos

- Diferentes facetas podem ser analisadas:
  - Características Estruturais:
    - Pode também estabelecer conformidade com restrições, padrões e estilos arquiteturais
    - Ajudam na análise de diferentes aspectos de distribuição e concorrência, visto que relacionam os elementos de *software* com as plataformas de *hardware* nas quais eles executam

Análise Arquitetural: Aspectos

- Diferentes facetas podem ser analisadas:
  - Características Comportamentais:
    - Analisar as características comportamentais de uma arquitetura envolve:
      - Considerar o comportamento interno de componentes individuais
      - Considerar a estrutura arquitetural, com o objetivo de avaliar comportamentos *composite*
    - Especialmente em caso de uso de elementos *COTS* propriedades comportamentais inferidas a partir da interface pública do elemento podem fazer com que muitos problemas comportamentais permaneçam não identificados

Análise Arquitetural: Aspectos

- Diferentes facetas podem ser analisadas:
  - Características de Interação:
    - São definidas pela quantidade e tipos de conectores utilizados, bem como pelos valores atribuídos às suas diferentes dimensões de variação
    - Exemplo: um conector *non-buffering* pode resultar em um sistema onde um dos componentes recebe somente, no máximo, metade dos dados
    - Pode envolver análise dos protocolos de interação e comportamento interno dos conectores
    - Possíveis objetivos:
      - Detectar se um componente ligado a um determinado conector será eventualmente acessado
      - Detectar se um conjunto de componentes apresentam *deadlock*

Análise Arquitetural: Aspectos

- Diferentes facetas podem ser analisadas:
  - Características Não-Funcionais:
    - Geralmente são obtidas através da inter-relação de múltiplos componentes e conectores, o que dificulta sua análise
    - São frequentemente compreendidas de forma inapropriada, qualitativas por natureza e suas dimensões são parciais ou informais
    - Ao mesmo tempo em que são importantes e desafiadoras, técnicas de análise arquitetural de propriedades não-funcionais são raras

Análise Arquitetural: Grau de Formalidade

- Relacionamento simbiótico entre modelos arquiteturais e análise arquitetural:
  - Modelos Informais:
    - Sujeitos a análises informais e manuais
    - Exemplo: determinação das necessidades básicas de pessoal para o projeto
    - Devem ser apreciadas com cuidado devido à sua inerente ambiguidade e falta de detalhes

Análise Arquitetural: Grau de Formalidade

- Relacionamento simbiótico entre modelos arquiteturais e análise arquitetural:
  - Modelos Semi-Formais
    - Maioria dos modelos utilizados
    - Alta precisão e formalidade X Expressividade e facilidade de compreensão
    - Sujeitos tanto a análises manuais quanto automáticas
    - Sua imprecisão parcial os tornam inadequados a análises mais sofisticadas

Análise Arquitetural: Grau de Formalidade

- Relacionamento simbiótico entre modelos arquiteturais e análise arquitetural:
  - Modelos Formais
    - São sujeitos a análises formais e automatizadas
    - Geralmente utilizados pelos *stakeholders* técnicos do projeto
    - Documentar toda a arquitetura através de modelos formais pode ser inviável
    - Modelos formais geralmente sofrem de problemas de escalabilidade

Análise Arquitetural: Tipo

- Pode ser classificada em:
  - Análise Estática
  - Análise Dinâmica
  - Análise Baseada em Cenários

Análise Arquitetural: Tipo

- Análise Estática:
  - Envolve a descoberta de propriedades, a partir dos seus modelos, sem requerer a execução destes modelos
  - Exemplo: determinar se o sistema está em conformidade com as regras sintáticas da notação
  - Pode ser automática (ex: compilação) ou manual (ex: inspeção)
  - Todas as notações podem sofrer análise estática
  - Notações formais, entretanto, produzem conclusões mais precisas e sofisticadas. Ex: notações axiomáticas, notações algébricas e notações baseadas em lógica temporal

Análise Arquitetural: Tipo

- Análise Dinâmica:
  - Envolve a execução ou simulação do modelo do sistema
  - Para isso, a fundamentação semântica do modelo deve ser “executável” ou passível de simulação
  - Exemplos: diagramas de transição de estados, eventos discretos, redes de filas e redes de Petri

Análise Arquitetural: Tipo

- Análise Baseada em Cenários:
  - Para sistemas grandes e complexos é inviável avaliar uma propriedade para todo o sistema
  - Neste caso, identifica-se *use-cases* específicos que representam os cenários mais importantes ou frequentes
  - Pode ser tanto estática quanto dinâmica
  - Deve-se ter cuidado com as inferências realizadas a partir da evidência limitada apresentada pela análise

Análise Arquitetural: Grau de Automação

- O grau de automação depende da formalidade e completude do modelo e da propriedade sendo analisada
- Uma propriedade que é bem compreendida e facilmente quantificável é mais fácil de ser avaliada automaticamente
  - Em particular, a maioria das propriedades não-funcionais são compreendidas em termos de intuição e diretrizes informais

Análise Arquitetural: Grau de Automação

- Graus de Automação:
  - Manual:
    - É custoso pois requer um significativo envolvimento humano
    - Entretanto, pode ser realizada em modelos de diversos níveis de detalhe, rigor, formalidade e completude
    - Além disso, pode levar em consideração o *rationale*, que é frequentemente um conhecimento tácito
    - Pode ser necessária quando propriedades potencialmente conflitantes devem ser garantidas
    - Técnicas:
      - Inspeções (ex: *Architecture Trade-Off Analysis Method – ATAM*)
    - Resultados são tipicamente qualitativos

Análise Arquitetural: Grau de Automação

- Graus de Automação:
  - Manual:
    - Geralmente aplicada em Análises Baseadas em Cenários
    - Um aspecto crítico é como tornar a análise confiável e repetível

Análise Arquitetural: Grau de Automação

- Graus de Automação:
  - Parcialmente Automático:
    - A maioria das análises arquiteturais podem ser parcialmente automatizadas (*software* + intervenção humana)
    - Exemplos:
      - Um modelo em xADL pode ser analisado em relação a regras de inter-conectividade específicas de um estilo
      - Um modelo Wright pode ser analisado em relação à presença de *deadlocks* entre componentes que se comunicam através de um conector
    - A impossibilidade de total automação é geralmente causada pela não captura de parâmetros relevantes à propriedade sendo analisada

Análise Arquitetural: Grau de Automação

- Graus de Automação:
  - Totalmente Automático:
    - As respostas obtidas através das análises parcialmente automatizadas são geralmente parciais ou incompletas
    - As técnicas de análise totalmente automáticas são utilizadas em conjunto com aquelas particialmente automáticas, para prover respostas mais completas

Análise Arquitetural: *Stakeholders*

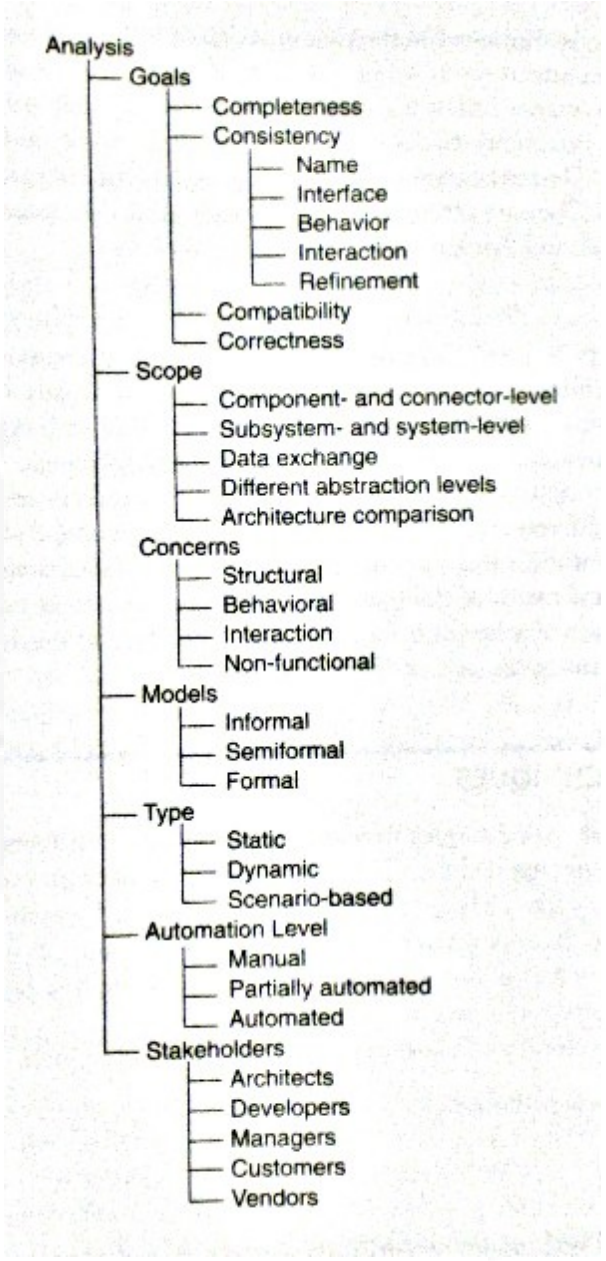
- Diferentes *stakeholders* possuem diferentes objetivos:
  - Arquitetos: visão global do sistema (4C's)
  - Desenvolvedores: visão mais limitada do sistema: compatibilidade dos módulos por eles desenvolvidos com os estilos, padrões e arquiteturas de referência
  - Gerentes: interessados em completude e corretude
  - Clientes: estão construindo o sistema correto ? Estão construindo corretamente o sistema ?
  - Distribuidores: interessados na facilidade de composição e compatibilidade com padrões e arquiteturas de referência

Análise Arquitetural: Técnicas

- As técnicas de análise arquitetural podem ser divididas em três categorias:
  - Baseadas em Inspeções/Revisões
  - Baseadas em Modelos
  - Baseadas em Simulação

Análise Arquitetural: Técnicas

- Dimensões de análise arquitetural utilizadas na descrição das técnicas

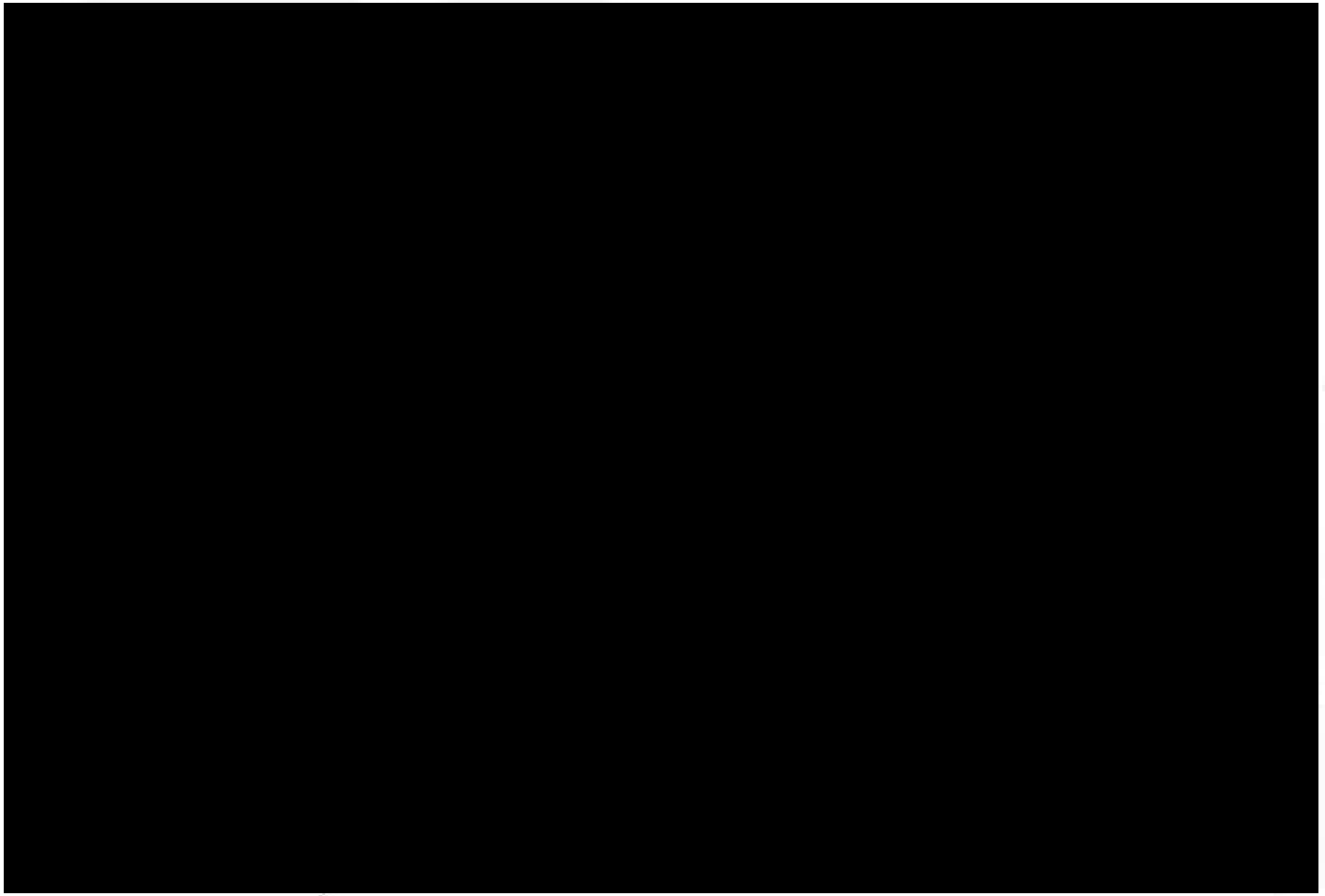


Análise Arquitetural: Técnicas

- Baseadas em Inspeções/Revisões
  - Conduzidas por diferentes *stakeholders* para garantir uma variedade de propriedades na arquitetura
  - Diferentes modelos arquiteturais podem ser estudados
  - Geralmente ocorre em *Review Boards*
  - Geralmente é uma técnica manual e cara
  - Entretanto, é util em descrições informais e parciais
  - Pode ter qualquer uma das quatro metas apresentadas
  - O escopo e aspecto pode variar amplamente
  - Geralmente são análise estáticas e baseadas em cenários

Análise Arquitetural: Técnicas

- Baseadas em Inspeções/Revisões
  - Ex: ATAM – *Architecture Trade-Off Analysis Method*



Análise Arquitetural: Técnicas

- Baseadas em Inspeções/Revisões
  - Ex: ATAM – *Resumo*

Goals	Completeness
	Consistency
	Compatibility
	Correctness
Scope	Subsystem- and system-level
	Data exchange
Concern	Non-functional
Models	Informal
	Semiformal
Type	Scenario-driven
Automation Level	Manual
Stakeholders	Architects
	Developers
	Managers
	Customers

Análise Arquitetural: Técnicas

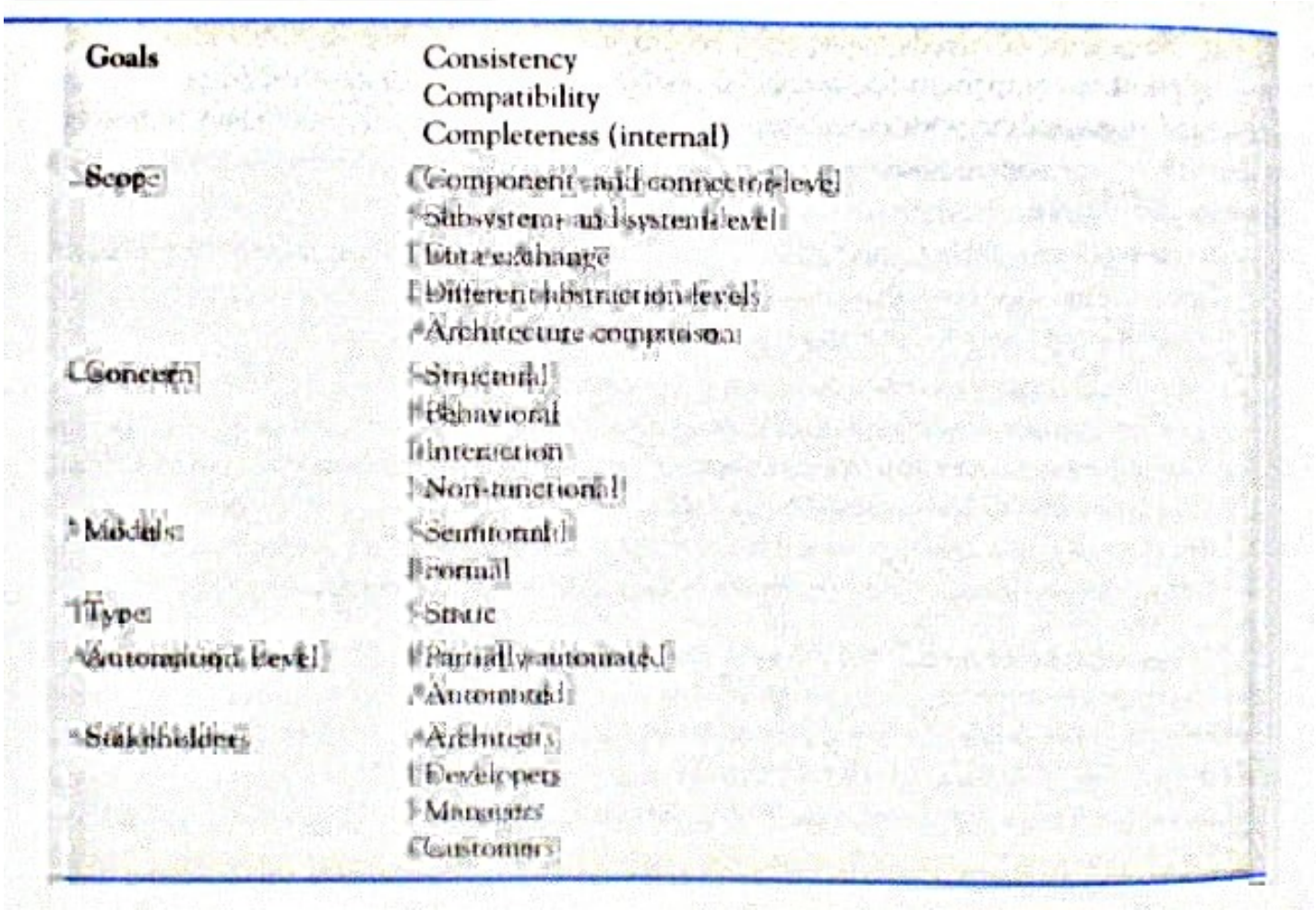
- Baseadas em Modelos
  - Dependem somente do descrição arquitetural do sistema e a manipula para descobrir propriedades de interesse
  - Envolve ferramentas de diferentes níveis de sofisticação, frequentemente guiadas pelos arquitetos, que interpretam os resultados intermediários e indicam o caminho da análise
  - É mais barato que inspeções/revisões porém só pode ser utilizado para avaliar propriedades que podem ser formalmente descritas
  - Geralmente focam em um único aspecto, por exemplo corretude sintática, ausência de *deadlock* ou conformidade com um determinado estilo arquitetural

Análise Arquitetural: Técnicas

- Baseadas em Modelos
  - Principal *trade-off* destas ferramentas:
    - Escalabilidade X Precisão (confiança)
    - Pode-se obter alta precisão e confiança ao se analisar sistemas pequenos. Para sistemas grandes algo tem que ser sacrificado
  - Produzem resultados parciais e são comumente empregadas em conjunto com técnicas das outras duas categorias

Análise Arquitetural: Técnicas

- ADLs como suporte à análise baseada em modelos:



Goals	Consistency Compatibility Completeness (internal)
Scope	Component- and connector-level Subsystem- and system-level Data exchange Different abstraction levels Architecture comparison
Concern	Structural Behavioral Interaction Non-functional
Models	Semiformal Formal
Type	Static
Automation level	Partially automated Automated
Stakeholders	Architects Developers Managers Customers

Análise Arquitetural: Técnicas

- Análise de confiabilidade baseada em modelos:

Goals	Consistency
	Compatibility
	Correctness
Scope	Component- and connector-level
	Subsystem- and system-level
	Data exchange
Concern	Structural
	Behavioral
	Interaction
	Non-functional
Models	Formal
Type	Dynamic
	Scenario-based
Automation Level	Automated
Stakeholders	Architects
	Developers
	Managers
	Customers
	Vendors

Análise Arquitetural: Técnicas

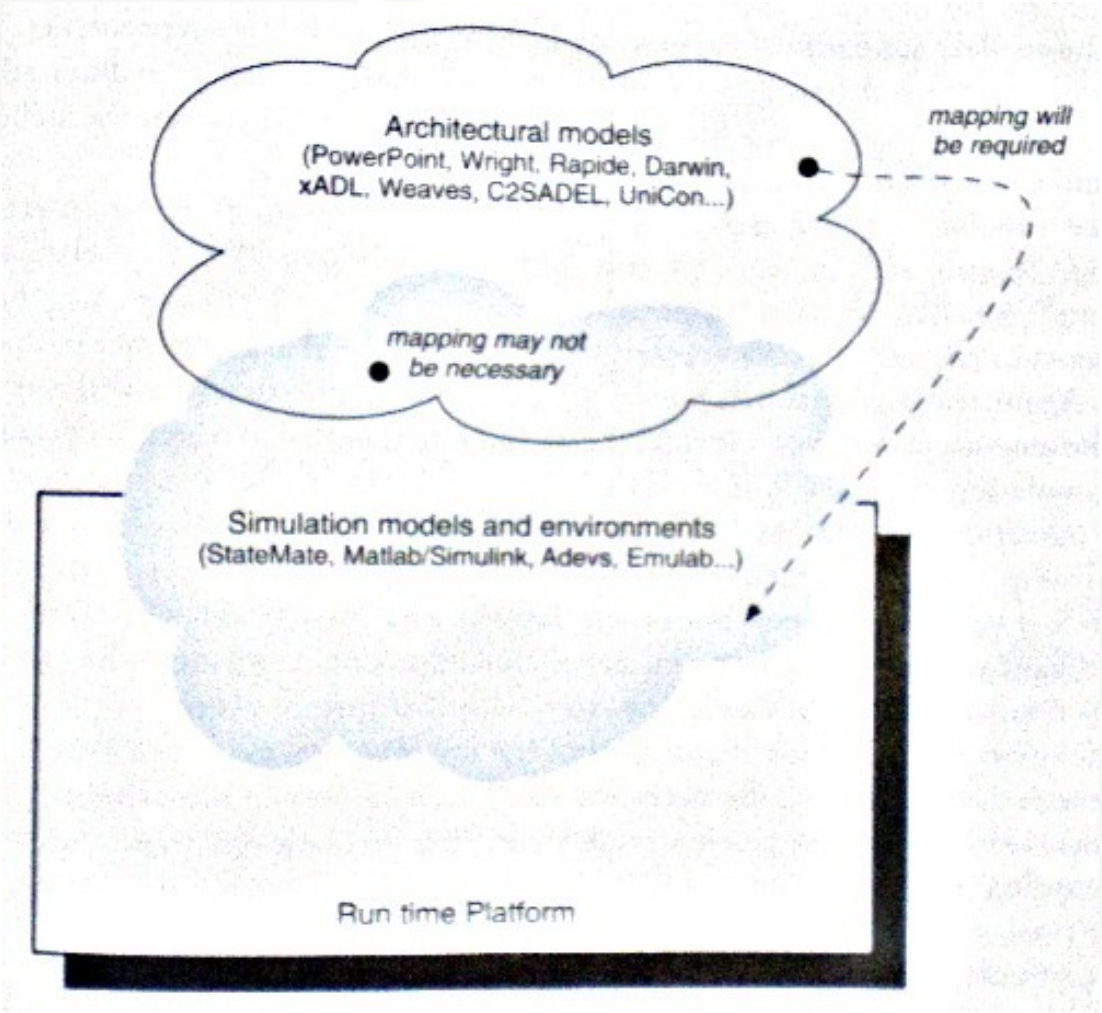
- Baseada em simulação:
  - Requer a produção de um modelo dinâmico e executável do sistema (ou de parte dele), possivelmente a partir de um outro modelo não-executável
  - Ex: geração de *traces* de execução a partir da descrição dos protocolos de interação
  - Simulações arquiteturais dão indícios somente de sequências de eventos, tendências gerais e faixas de valores
  - Em contraste, uma execução da implementação do sistema pode ser vista como uma simulação altamente precisa

Análise Arquitetural: Técnicas

- Baseada em simulação:
  - Mesmo os modelos passíveis de simulação podem precisar ser aumentados com um formalismo externo que viabiliza a sua execução
    - Ex: modelos de arquiteturas baseadas em eventos não produzem a geração e processamento de eventos nem as frequências de resposta
    - Tais modelos devem ser mapeados em um formalismo de simulação de eventos discretos ou em uma rede de filas
    - Cada informação adicional, entretanto, pode introduzir imprecisões no modelo arquitetural
  - Visto que existe um bom suporte por ferramentas, diferentes valores podem ser experimentados nos parâmetros do modelo

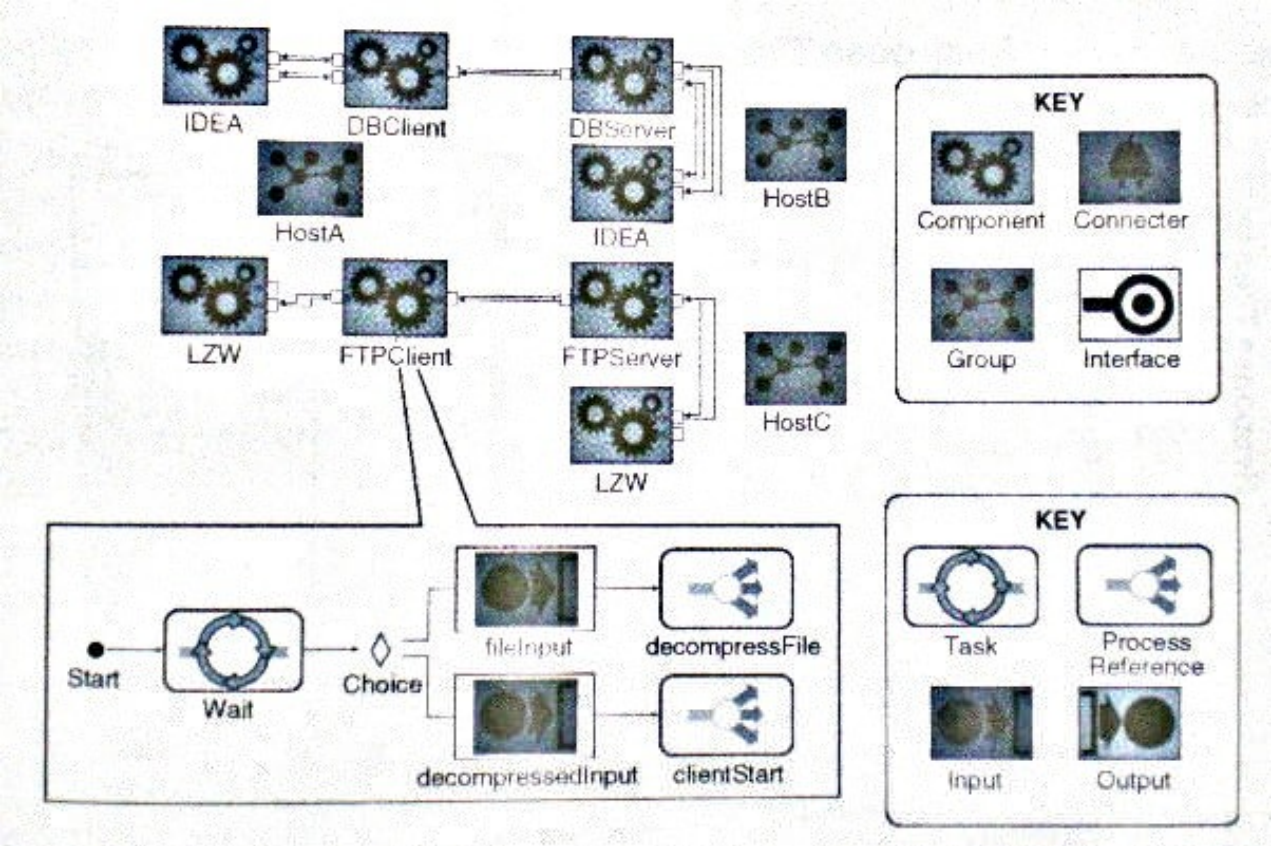
Análise Arquitetural: Técnicas

- Baseada em simulação:



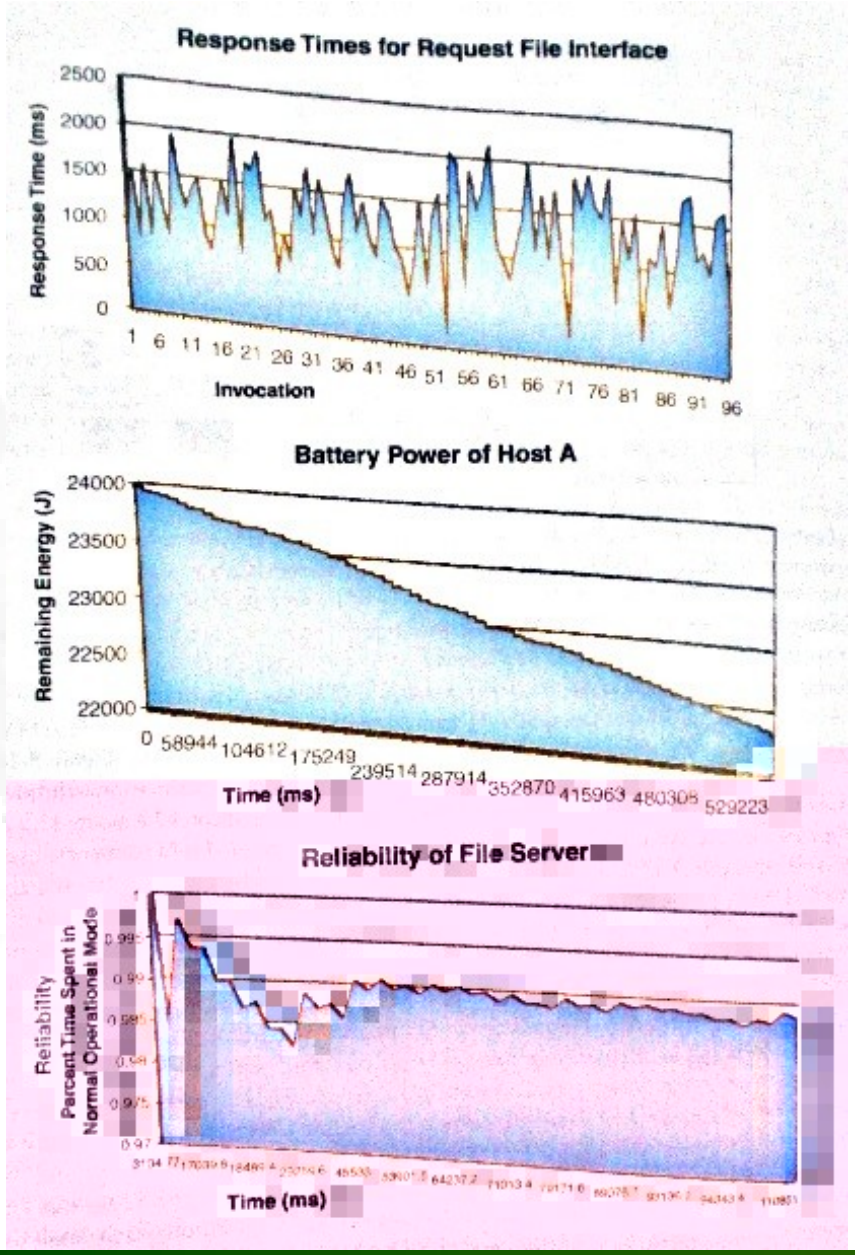
Análise Arquitetural: Técnicas

- Baseada em simulação:
  - Ex: XTEAM (*eXtensible Tool-chain for Evaluation of Architectural Models*)



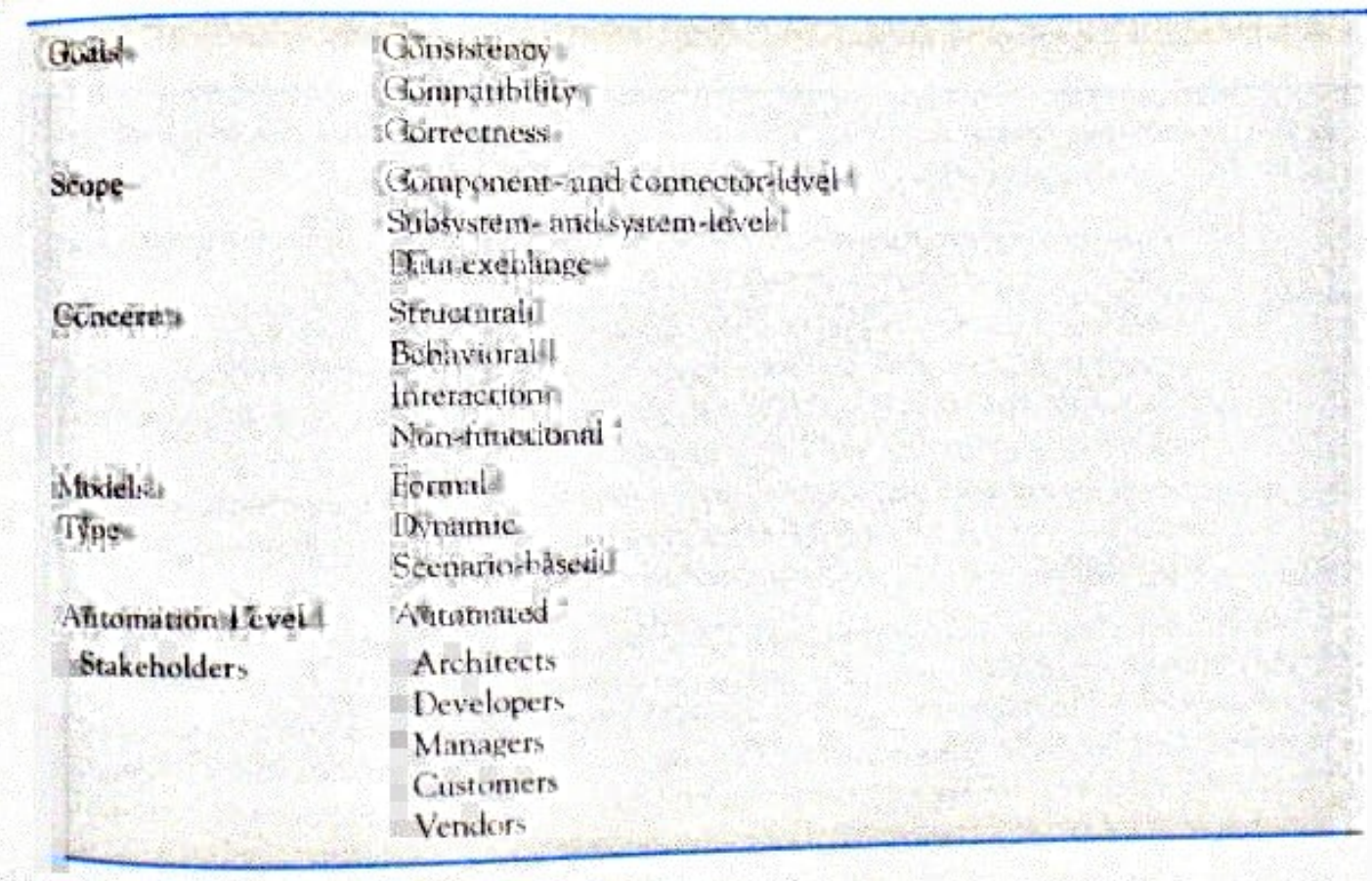
Análise Arquitetural: Técnicas

- Baseada em simulação:
  - Ex: resultados de uma simulação no XTEAM



Análise Arquitetural: Técnicas

- Baseada em simulação:
  - XTEAM - resumo



Goals	Consistency
	Compatibility
	Correctness
Scope	Component- and connector-level
	Subsystem- and system-level
	Data exchange
Concerns	Structural
	Behavioral
	Interaction
	Non-functional
Models	Formal
Type	Dynamic
	Scenario-based
Automation Level	Automated
Stakeholders	Architects
	Developers
	Managers
	Customers
	Vendors

INF016 – Arquitetura de Software

07 - Análise

Sandro Santos Andrade
sandroandrade@ifba.edu.br

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