



PMR 5020

Metodologia do *Projeto* de Sistemas

Aula 11: Axiomatic Systems Design

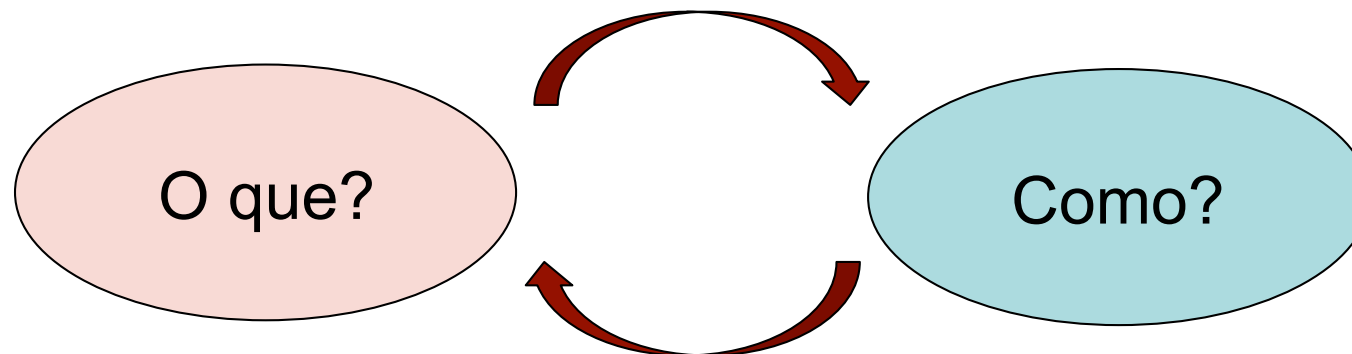
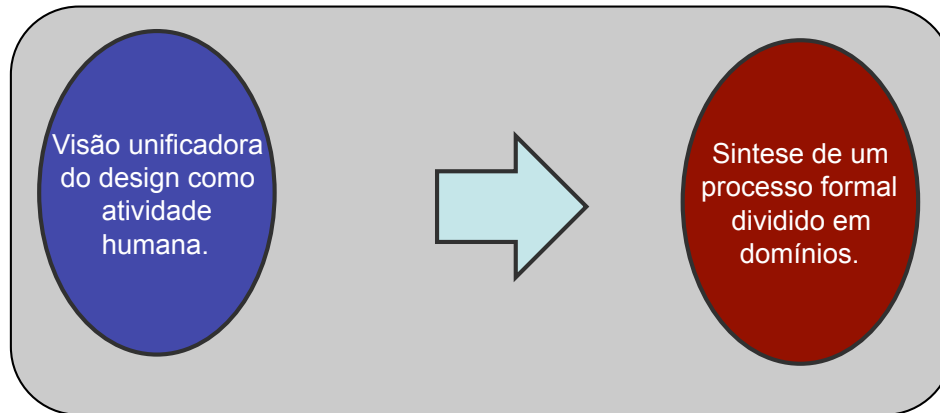


Prof. Dr. José Reinaldo Silva
reinaldo@poli.usp.br





The Design Metaphor

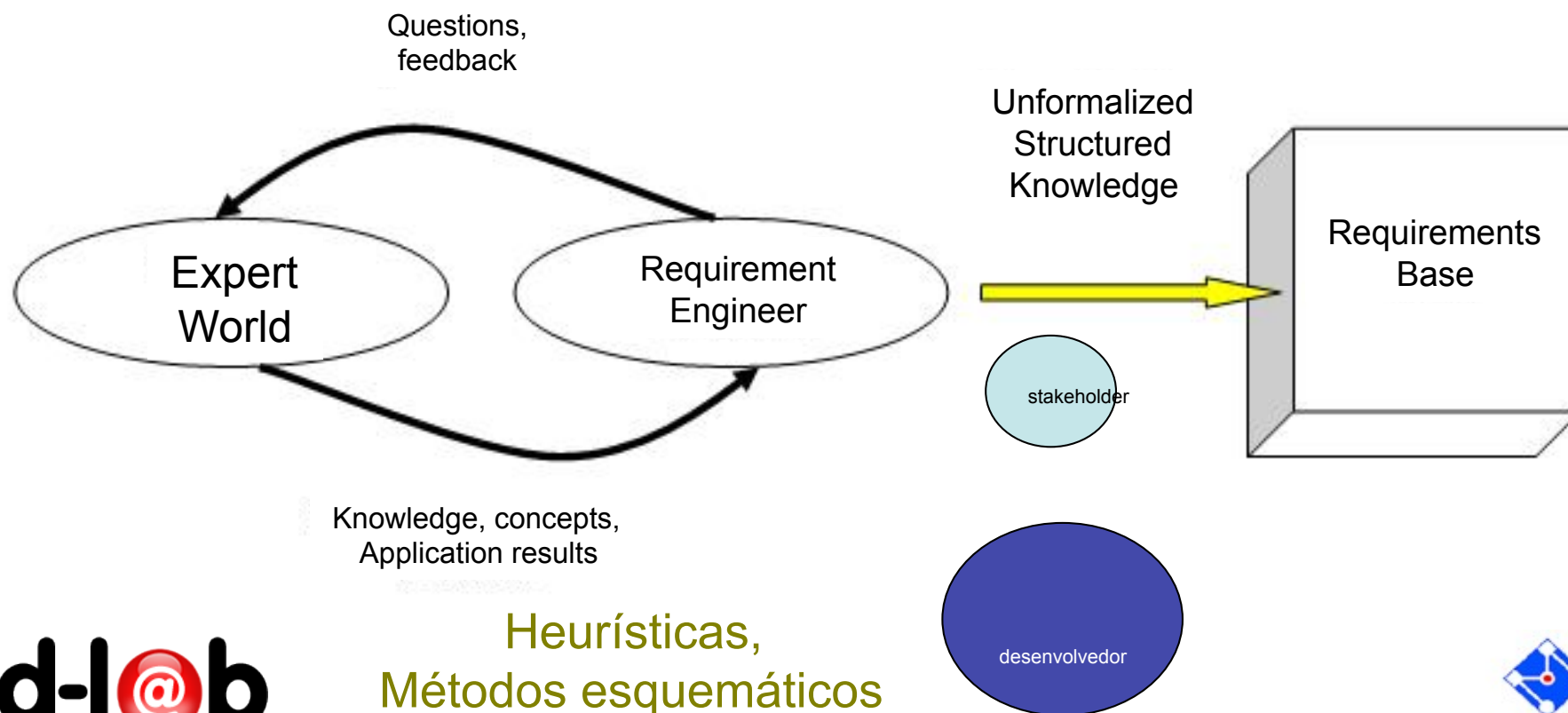
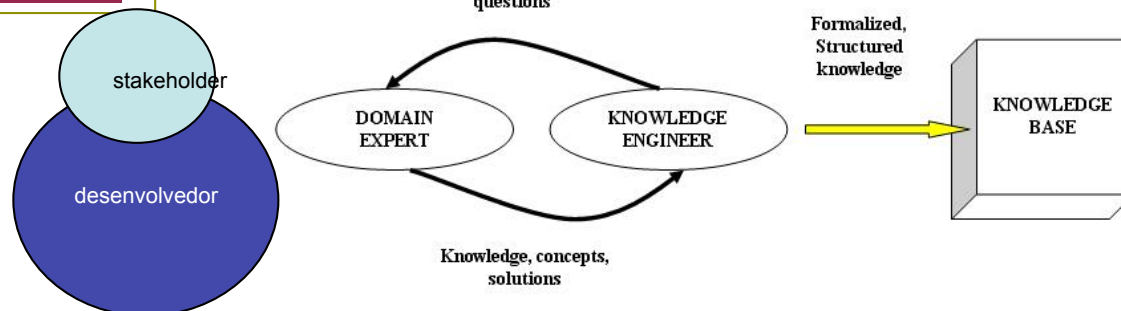


Convergencia?



O Processo de eliciação

IA,
Modelagem formal





Axiomatic Design

Os balizadores do Axiomatic Design são:

CA – Costumer Attributes

FR – Functional Requirements

DP – Design Parameters

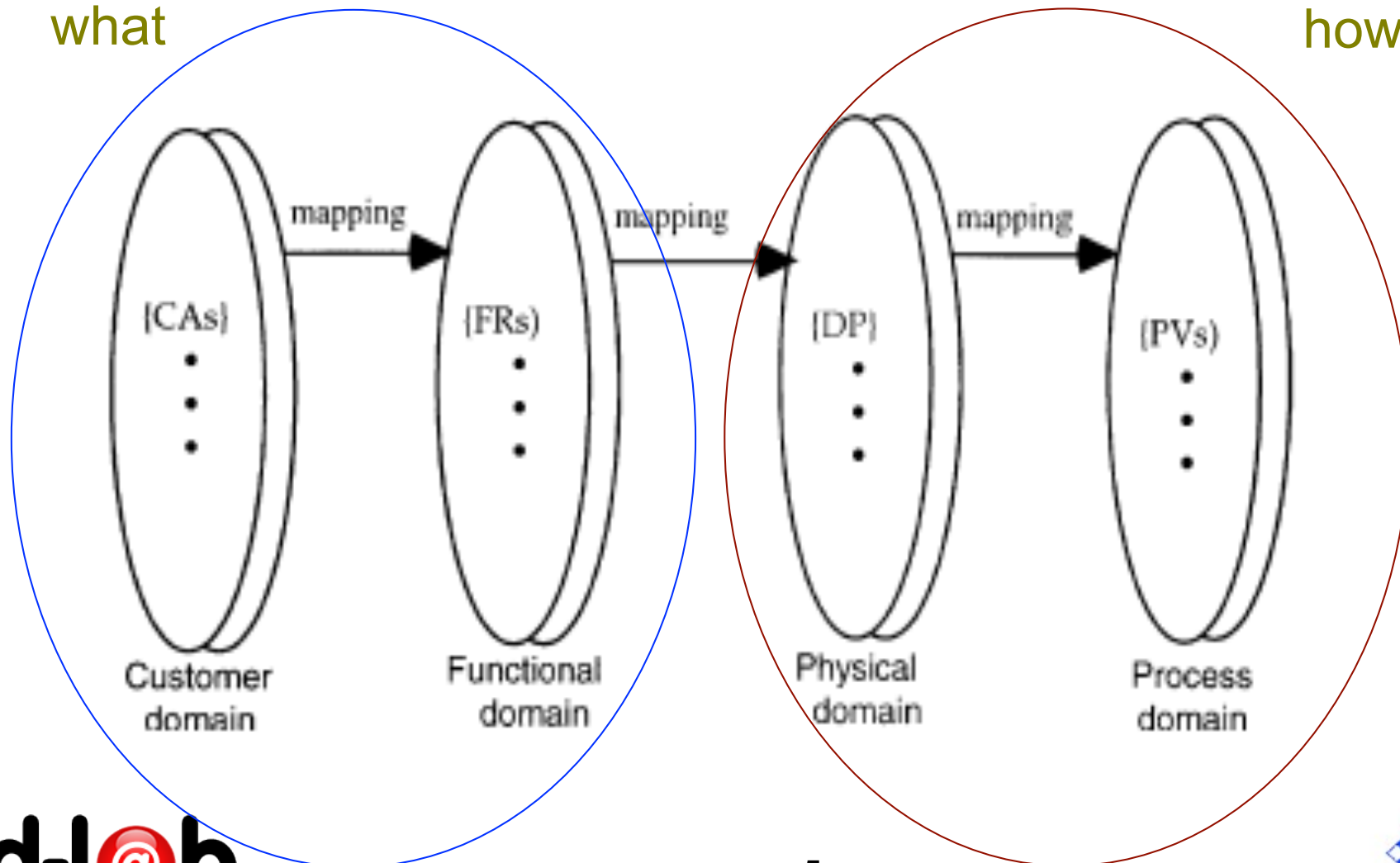
PV – Process Variables



Domínios na ADT

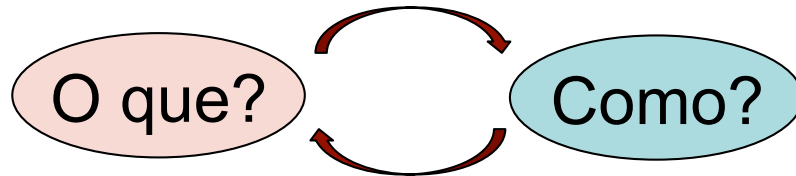
what

how





Os Axiomas



Axioma 1: *Axioma da Independência*

Mantenha a independência entre os FRs;

Axioma 2: *Axioma da Mínima Informação*

O Melhor design é aquele que minimiza a informação sobre o artefato.



Functional Requirement:

Functional requirements (FRs) are a minimum set of independent requirements that completely characterize the functional needs of the product (or software, organizations, systems, etc.) in the functional domain. By definition, each FR is independent of every other FR at the time the FRs are established.

Constraint:

Constraints (Cs) are bounds on acceptable solutions. There are two kinds of constraints: input constraints and system constraints. Input constraints are imposed as part of the design specifications. System constraints are constraints imposed by the system in which the design solution must function.



Design parameter:

Design parameters (DPs) are the key physical (or other equivalent terms in the case of software design, etc.) variables in the physical domain that characterize the design that satisfies the specified FRs.

Process variable:

Process variables (PVs) are the key variables (or other equivalent term in the case of software design, etc.) in the process domain that characterizes the process that can generate the specified DPs.



A fase de “design”

A fase reconhecida como design (how) é baseada em um mapeamento entre os FRs e os DPs. Assim, o design está associado à montagem da seguinte equação entre os elementos do desenvolvimento:

$$FR = [A] DP$$

Onde FR e DP representam vetores e A, chamada *matriz de design*, é uma matriz n x m que estabelece a correlação entre estes vetores.



Definindo os processos

A fase final do processo de design (no método axiomático) consiste em mapear os DPs nas PVs. Como anteriormente a equação é a mesma e sujeita uma análise semelhante, onde agora uma matriz B .

$$DP = [B] PV$$



A matriz de design

Algumas considerações sobre a matriz de design:

- i) Seja a matriz $\mathbf{A}$, $m \times n$, onde n é o número de DPs e m é o número de FRs. Se $m > n$ então temos mais equações que incógnitas e o sistema é redundante.
- ii) Seja a matriz $\mathbf{A}$, $m \times n$, onde n é o número de DPs e m é o número de FRs. Se $m < n$ então temos mais incógnitas que equações e o sistema é indeterminado.
- iii) A situação ideal é quando $m = n$ e $\mathbf{A}$ é uma matriz quadrada. Neste caso é possível achar uma solução única para o design.



Teoremas do AD

Teorema 1] Seja um design $\mathcal{D}$ onde o número de DPs é menor que o número de FRs. O design é acoplado ou os FRs não podem ser satisfeitos.

Dem] Seja o design denotado pela equação $FR = [A]_{m \times n} DP$, onde $m > n$. Supondo que o design desacoplado até a ordem n , isto é, que se possa gerar uma sub-matriz diagonal $n \times n$. Os demais $m-n$ FRs serão definidos pela combinação dos mesmos DPs e o design será acoplado. Se alguma linha $r > m-n$ for nula, então FR_r não poderá ser satisfeito.

$$\begin{pmatrix} FR_1 \\ \vdots \\ FR_n \\ \vdots \\ FR_m \end{pmatrix} = \begin{bmatrix} X & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & X \\ \cdots & \cdots & \cdots & \cdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \begin{pmatrix} DP_1 \\ \cdots \\ DP_N \end{pmatrix}$$



Teoremas do AD

Teorema 2] Seja um design $\mathcal{D}$ acoplado, onde o número de DPs é menor que o número de FRs. Se $\mathcal{D}$ é acoplado, então pode ser desacoplado pela inserção de novos DPs design pode ser desacoplado pelo acréscimo de DPs até que $|FR|=|DP|$ e triangularizando esta nova matriz.

Corolário 1] Um design $\mathcal{D}$ acoplado deve ser dividido em partes, bem como a sua solução.

Corolário 2] O conjunto $[FR]$ deve ser minimal.



Teoremas do AD

Teorema 3] Seja um design $\mathcal{D}$ acoplado, onde o número de DPs é maior que o número de FRs. Neste caso o design pode ser acoplado ou redundante.

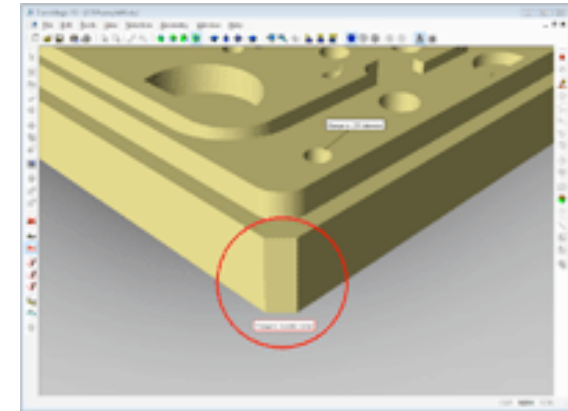
No caso do design redundante, este pode ou não violar o Axioma da Independência, dado que a única forma de resolver as equações com um número maior de incógnitas seria fixar o valor de alguns DPs. Dependendo da escolha, o design pode ser desacoplado, parcialmente acoplado, ou completamente acoplado.

$$\begin{pmatrix} FR_1 \\ \vdots \\ FR_m \end{pmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ \vdots & \cdots & \cdots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \begin{pmatrix} DP_1 \\ \vdots \\ DP_m \\ \vdots \\ DP_n \end{pmatrix}$$



Feature Integration

design features. Concept introduced by C. F. Hockett in the 1960s of a set of key properties of language not shared or not known to be shared, as a set, with systems of communication in any other species. Their number and names vary from one account to ...

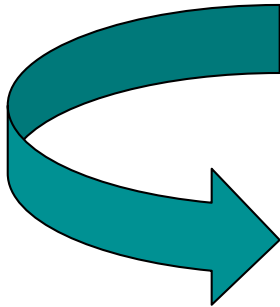


Corolário 3] É possível integrar *design features* em um único componente físico se os FRs puderem ser satisfeitos independentemente nesta nova solução.



Teoremas do AD

Teorema 4] Um design $\mathcal{D}$ é dito ideal se o número de DPs é igual ao número de FRs, e estes estão definidos de forma independente em relação aos DPs, isto é, a matriz de design é diagonal.

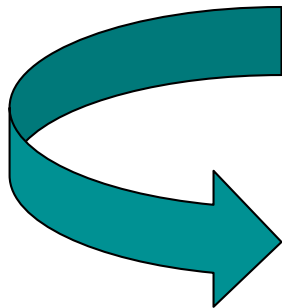


O Design ideal



Teoremas do AD

Teorema 5] Seja um design $\mathcal{D}$. Se o conjunto dos *functional requirements* é modificado pela inserção de um novo FR, ou pela substituição de alguns dos FRs, ou ainda pela escolha de um novo conjunto completamente diferente, a solução associado aos DPs originais não pode mais satisfazer ao novo conjunto de FRs. Portanto uma nova solução deve ser buscada.



Design Revision



Ambientes de design

Axiomatic Design Solutions, Inc.

Curso: Metodologia do Projeto ... WebHome < DLab < TWiki USP :: Sistemas USP - Departamento... Axiomatic Design Solutions, Inc.

http://www.axiomaticdesign.com/

Most Visited Getting Started Latest Headlines Apple Yahoo! Google Maps YouTube Wikipedia Noticias Bookmarks

GAME & APPS axiomatic design Web Search Login 24°C

AXIOMATIC DESIGN
SOLUTIONS, INC.

Support / WebEx / eTraining / Contact Us

Click to view the Acclaro DFSS Software Demonstration

Next Generation DFSS Technology

- >> Product of the Year with Axiomatic Design
- >> General Dynamics picks Acclaro DFSS
- >> Latest *Design for Lean Six Sigma* Book
- >> Acclaro DFSS expands QFD functions
- >> DFSS Projects save \$300k on Average
- >> ICAD 2009 Best Papers Awarded

Solutions Services Software News/Events Clients Partners Technology Publications

Functional Requirements Analysis and Risk Mitigation for software, business process and product development

We reduce development times by 50% or more by integrating MIT's remarkable Axiomatic Design technology into a pre-emptive Design for Six Sigma (DFSS) process framework that enables robust design and risk mitigation during the requirements analysis phase. You will "Get it right" the first time. The more complex the task, the more impact our process technology has in reducing development risk, costs and delivery time.

[Click here to understand how we achieve so much success.](#)

If it's not Axiomatic, it's not DFSS.

Industry Week Technology of the Year

Gold Medal of Honor

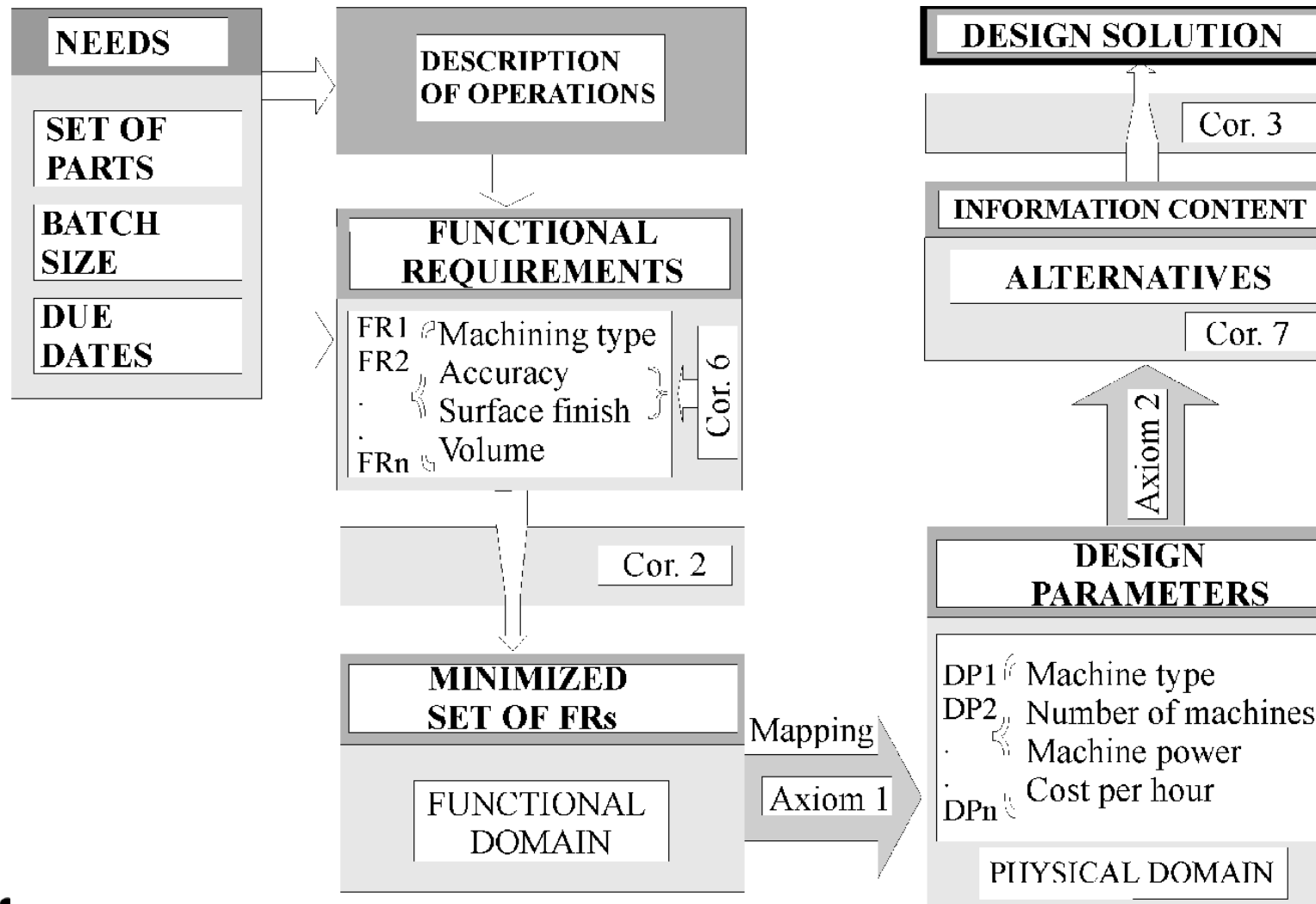
FW Taylor Medal

Talk with us...

Acclaro® is a registered trademark of Axiomatic Design Solutions, Inc. Copyright© 1998-2006 Axiomatic Design Solutions, Inc. All Rights Reserved. [Site Map](#)



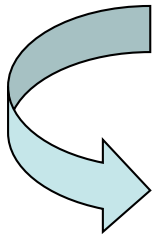
Uso industrial do AD





O artefato

Um sistema $S=(A,R)$ é um par ordenado composto de um conjunto de elementos (objetos) relevantes A e de um conjunto de relações entre estes objetos. Eventualmente, um elemento de A pode ser também um sistema, neste caso chamado de sub-sistema em relação a S



Sistema dinâmico



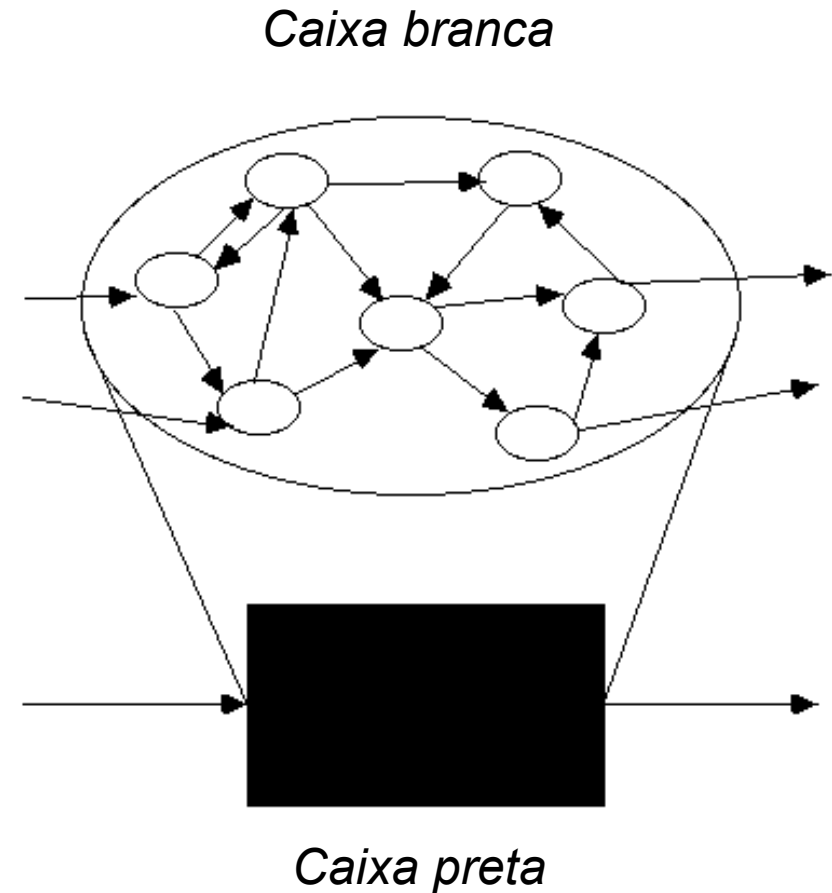
Mais formalmente...

Seja um conjunto de conjuntos $X_1, \dots, X_n$ e o produto cartesiano entre eles, $\mathfrak{K} = \prod_1^n X_i$. Um sistema S é um subconjunto de $\mathfrak{K}$, definido sobre os elementos dos conjuntos X_i .



Systems theory or systems science argues that however complex or diverse the world that we experience, we will always find different types of organization in it, and such organization can be described by concepts and principles which are independent from the specific domain at which we are looking.

<http://pespmc1.vub.ac.be/CYBSWHAT.HTML>





Definições Básicas

Um sistema $S=(A,R)$ é um par ordenado composto de um conjunto de elementos (objetos) relevantes A e de um conjunto de relações entre estes objetos. Eventualmente, um elemento de A pode ser também um sistema, neste caso chamado de sub-sistema em relação a S .

Portanto os sistemas podem ser classificados na prática de duas formas:

- sistemas restritos a certo tipo de elementos (objetos) de A
- sistemas restritos a certo tipo de relação entre seus elementos



Definições Básicas

$$S=(A,R)$$

O conjunto A reúne os **elementos** do sistema. Cada um deles possui um conjunto de **atributos**, qualitativos ou quantitativos. No segundo caso estes são definidos em um domínio de **valores**.

Existem **relações** entre os elementos de um sistema. Estas relações podem ser tais que alguns elementos têm atributos dependentes dos atributos de outros. Uma relação especial é a que se dá entre o sistema e o ambiente externo a este, chamado de **contexto** (environment). A relação entre o sistema e seu contexto é o que resulta no **objetivo do sistema**. O objetivo do sistema é a finalidade do seu hólons.



Definições Básicas

Decomposição

Seja um sistema $S = (A, R)$. Qualquer subconjunto próprio $\alpha \subset A$ ao qual se pode atribuir um conjunto de relações $\rho \subset R$ – sejam estas relações com outros elementos ou com o contexto – pode ser chamado de sub-sistema de S .

Um sub-sistema α é um sistema onde o contexto é definido como $C_S \cup (A - \alpha)$, onde C_S é o contexto do sistema S .

Trata-se portanto de uma definição recursiva.



Definições Básicas

Decomposição

Existem portanto várias maneiras de decompor um sistema. Em geral para um sistema com N elementos existem $2^N - 1$ escolhas possíveis de sub-sistema.

Entretanto o que queremos é de fato encontrar um conjunto de sub-sistemas $\mathcal{A} = \{\alpha_i\}$ de modo que,

$$\bigcup_i \alpha_i = A \quad \text{e} \quad \bigcap_i \alpha_i = \emptyset$$



Definições Básicas

Decomposição

Concluimos portanto que existem um número grande de possibilidades para a composição do conjunto A. Todas estas escolhas são igualmente válidas para decompor um sistema S (Teorema da modularidade).

Portanto, na prática uma decomposição ganha precedência sobre outras se a familiaridade da funcionalidade de suas partes ou a facilidade de interpretação e mapeamento destas funcionalidades em sistemas reais é maior.



Dinâmica dos Sistemas

Um sistema é dito ***dinâmico*** se os seus elementos, além de possuírem propriedades estáticas (atributos) têm a capacidade de realizar funções que modificam os valores dos seus atributos ou de outros elementos (do próprio sistema ou pertencentes ao contexto).

Chamaremos genericamente de ***ação*** à invocação e realização de um ou mais destas funções de transformação, pertinentes a um ou mais elementos.



Dinâmica dos Sistemas

Assim, podemos prever a evolução de um sistema no tempo pela realização de sucessivas **ações**. Vamos portanto caracterizar esta ocorrência.

Chamaremos de **estado** a situação (uma fotografia instantânea do sistema) caracterizada pelos valores de todos os atributos de todos os elementos. Assim, a ocorrência de uma **ação** (simples ou distribuída) faz com que o sistema **mude de estado**, configurando o que normalmente se conhece por **evento**.

A ocorrência de uma ação (ou de um conjunto de ações independentes) depende apenas do estado corrente e se este preenche os requisitos para a invocação das funções dos respectivos elementos, isto é, se os argumentos das respectivas funções são conhecidos.



O comportamento do sistema

Chamamos ainda **atividade** a um conjunto de ações distribuídas, e independentes, isto é, que podem ser realizadas em qualquer ordem ou simultaneamente.

Uma seqüência destas atividades, geradas por uma interação entre o sistema e o contexto é chamada de **processo**, cujo resultado (estado final) e soma das atividades realizadas no contexto, chamamos processo.

Se após a ocorrência de evento ou atividade, existirem várias possibilidades de escolha de ações (atividades), então se faz necessário ter um critério de escolha para chegar ao resultado desejado sobre o contexto. A elaboração destes critérios é chamado de **planejamento**.



Exemplo de sistema

Seja a cidade P, formada (no que interessa ao problema), por suas ruas e avenidas de acesso. Estas podem ser divididas em quatro tipos:

A1 = as ruas de mão dupla em pleno funcionamento

A2 = as ruas de mão única em pleno funcionamento

A3 = ruas de mão dupla em conserto, com trechos impedidos

A4 = ruas completamente interditadas

Esta cidade e seus pontos de localização constitui um contexto para o problema de transporte de cargas dentro do universo da cidade.



AD System Design

O Axiomatic Design aplicado a sistemas combina os conceitos básicos de sistemas vistos anteriormente com novos teoremas (e os axiomas básicos) que regulam o design de sistemas. O feature mais importante neste caso é a decomposição.

Teorema S1] O processo de decomposição de um sistema de grande porte não afeta o desempenho do design desde que os requisitos funcionais e as restrições de mais alto nível sejam satisfeitas e o conteúdo de informação inserido pelo processo de decomposição é nulo.



AD System Design

No design em engenharia o custo é fator decisivo e importante e é capaz de definir a viabilidade de vários projetos práticos. Assim, embora não seja de fato um requisito direto ou um fator “técnico” tem um lugar de destaque nos teoremas do AD.

Teorema S2] Sejam os designs $\mathcal{D}_1$ e $\mathcal{D}_2$, que têm os mesmos requisitos funcionais e restrições semelhantes (são funcionalmente equivalentes) e que possuem o mesmo conteúdo informacional. Apesar de serem equivalentes, $\mathcal{D}_1$ e $\mathcal{D}_2$ podem ter uma estrutura de custo completamente diferente.



AD System Design

Como seria de se esperar a fase inicial do processo, isto é, o estabelecimento de um conjunto de requisitos funcionais e de restrições é um fator decisivo para o desempenho e para a qualidade do processo de design, e para a sua estrutura.

Teorema S3] A qualidade do processo de design depende da escolha apropriada dos requisitos funcionais e das restrições e do seu mapeamento nos domínios físico e de processo, respectivamente. Uma escolha errada dos FRs feita em alto nível não poderá ser retificada ao longo do processo.



AD System Design

Um outro problema que certamente é amplificado pelo tamanho e complexidade dos sistemas é a base de critérios para a escolha de um design mais adequado, ou ótimo, para sistemas de grande porte e flexíveis.

Teorema S4] A melhor escolha para o design de um sistema de grande porte, integrado e flexível, que satisfaz a m requisitos funcionais deve repousar entre as soluções que satisfazem o Axioma da Independência se for conhecido a priori um conjunto completo de subconjuntos de FR que este sistema deve satisfazer.



AD System Design

Embora seja bastante desejável, é difícil ter um processo de otimização claro e transparente para qualquer tipo de design. Portanto, a obtenção de tal método é uma contribuição muito importante para o processo. Em geral se convive com o problema de saber se é necessário procurar um “design melhor”.

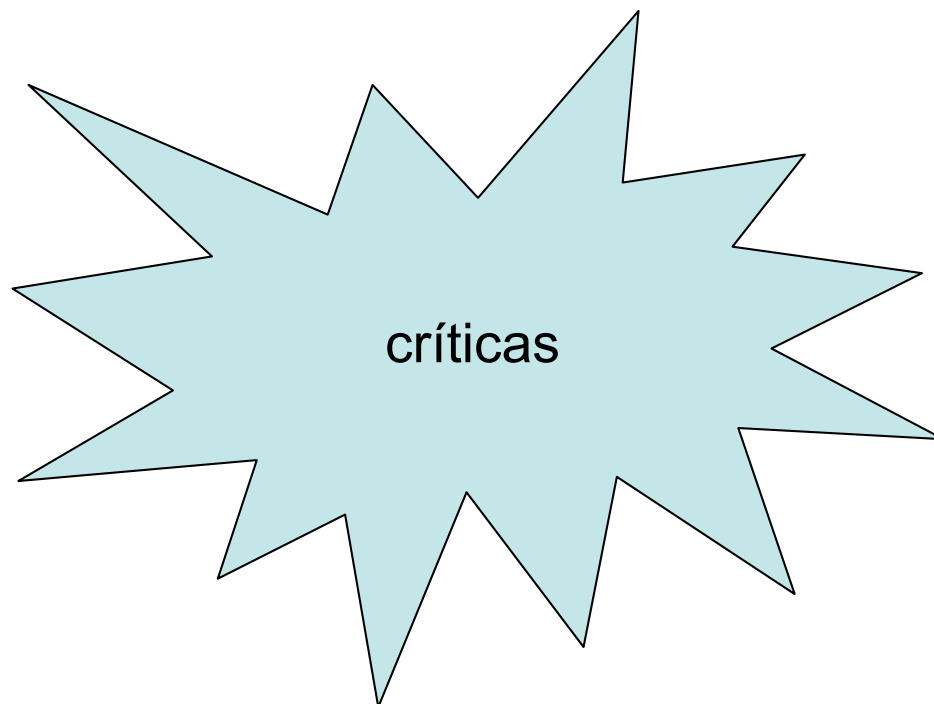
Teorema S5] Quando não se conhece a priori uma partição completa do conjunto dos FRs à qual o sistema deve satisfazer durante o seu ciclo de vida, não existe nenhuma garantia que um dado design minimizará o conteúdo de informação para todos os possíveis subconjuntos e portanto que um dado design é o melhor.



AD System Design

A qualidade é sempre um atributo perseguido e é imperioso, especialmente para os sistemas de sistemas (SoS) que um critério seja hegemônico para qualificar o design. Assim, introduzimos aqui o Teorema 9, seguindo a classificação de Nam Suh.

Teorema S9] Os fatores de qualidade do design de um sistema integrado e flexível de grande porte estão relacionados com a qualidade do banco de dados, com a escolha apropriada dos requisitos funcionais e com o mapeamento feito entre os domínios.





Exercício (um desafio)

Como leitura da semana teremos um artigo básico do Nam Piu Suh sobre Axiomatic Design aplicado a sistemas. Será também disponibilizado um texto com todos os teoremas enunciados pelo Nam Piu Suh no seu livro de 2001 sobre Axiomatic Desig. Vocês agora devem analisar este texto e discernir (por escrito) sobre a conveniência de usar um ou mais dos teoremas no seu trabalho final.



Fim