



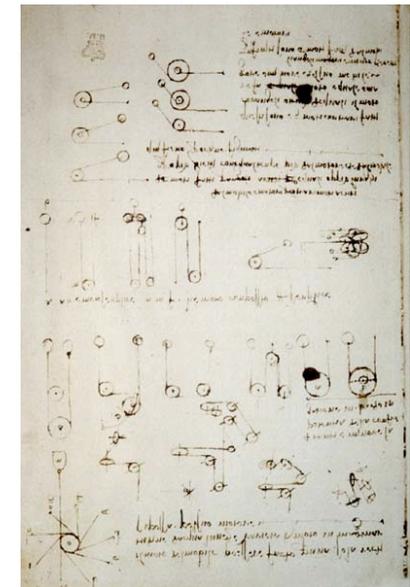
# PMR 5020

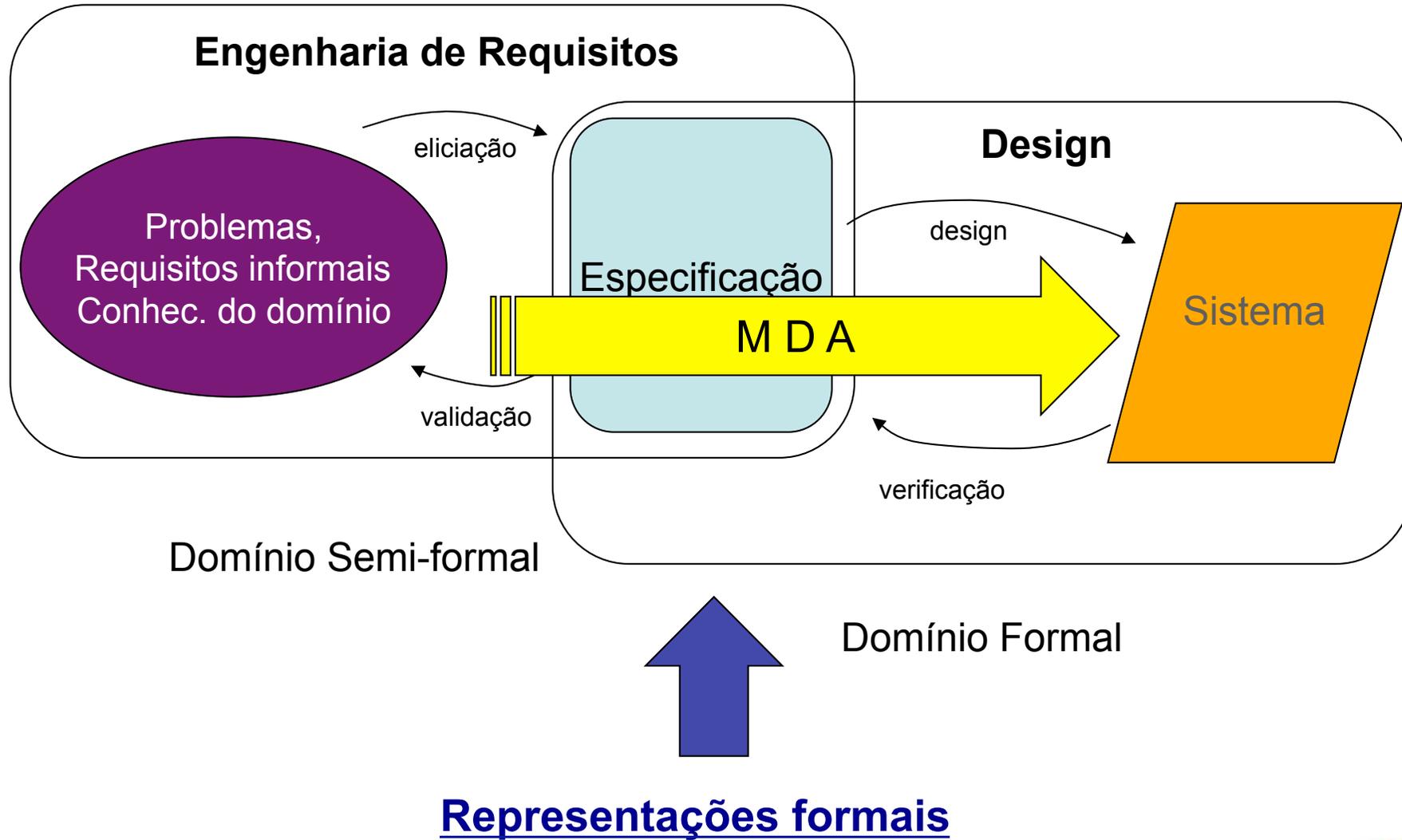
## Metodologia do *Projeto* de Sistemas

### Aula 10: Axiomatic Design



**Prof. Dr. José Reinaldo Silva**  
[reinaldo@poli.usp.br](mailto:reinaldo@poli.usp.br)







# Os nós da questão



Goal oriented requirements



Model Driven Engineering



# O problema dos paradigmas

Existem vários paradigmas “candidatos” para orientar o processo de design:

**Método estruturado**

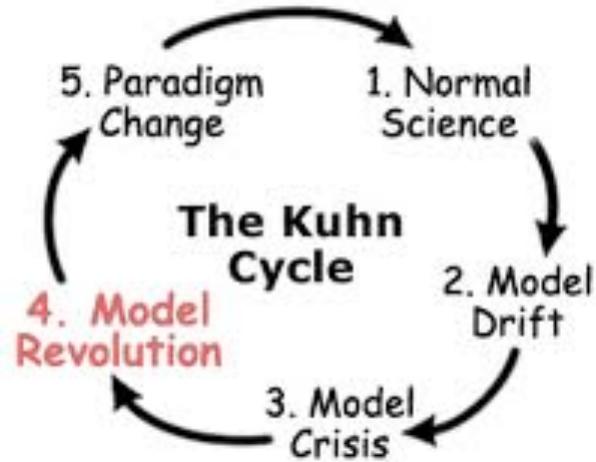
**Modularização**

**objetos**





# O Ciclo de mudanças



Flogístico

Método de Jackson

Física Clássica

Análise estruturada

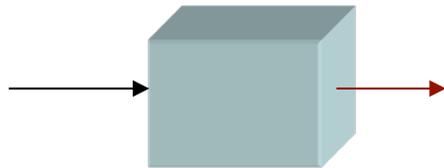
Física quântica

Objetos

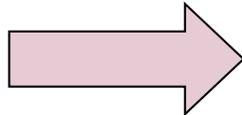




# Elementos de estruturação



Elementos próprios

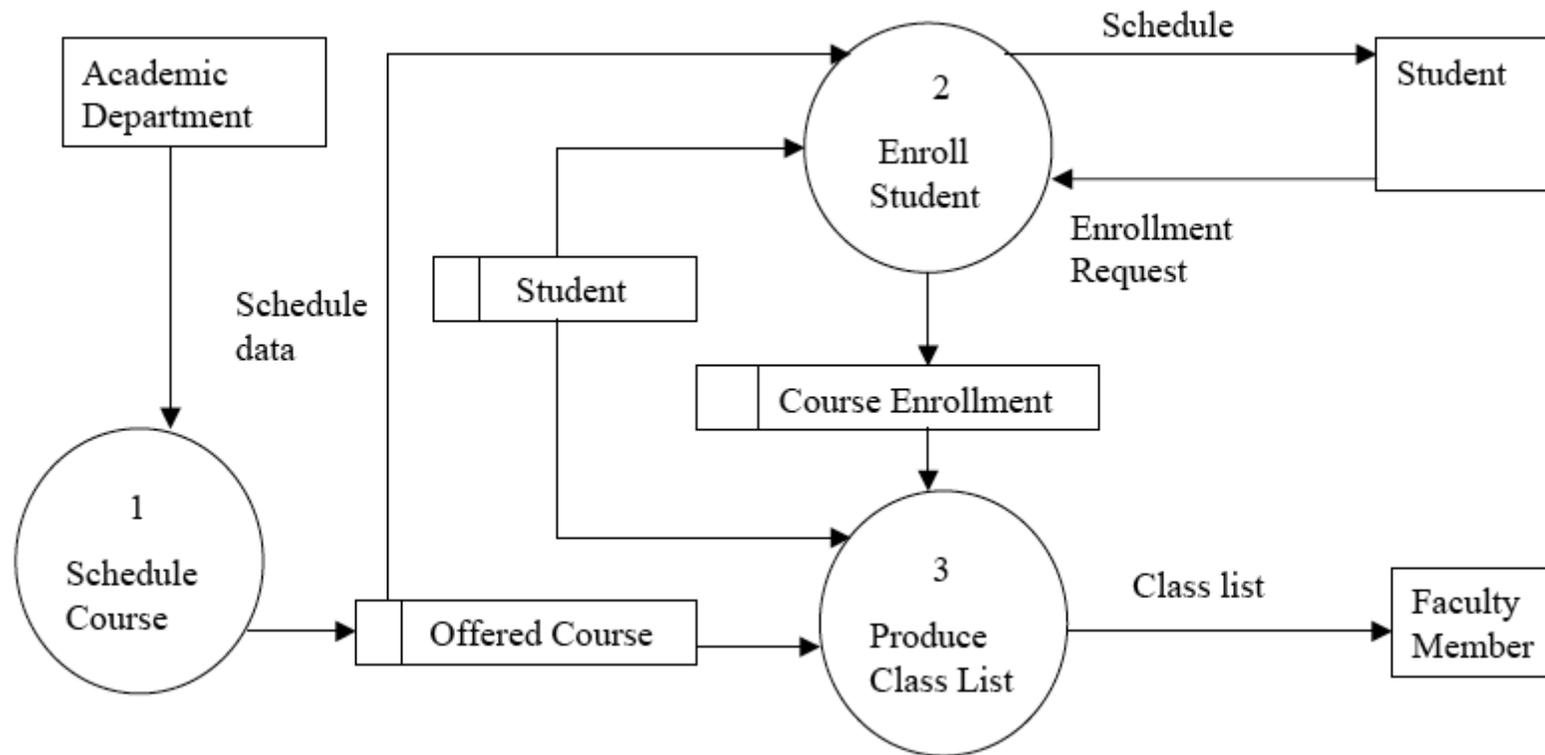


Base de elementos constituintes (componentes)

Elementos próprios indivisíveis são chamados primos. Um conjunto LI de elementos primos pode constituir uma base e portanto pode descrever qualquer programa.



# Example: semester student enrollment





## The Structured Method Agenda

- Structured programming in circa 1967 with Edsger Dijkstra - "Go To Statement Considered Harmful"
- Niklaus Wirth Stepwise design in 1971
- Nassi–Shneiderman diagram in 1972
- Warnier/Orr diagram in 1974 - "Logical Construction of Programs"
- HIPO in 1974 - IBM Hierarchy input-process-output (though this should really be output-input-process)
- Structured Design around 1975 with Larry Constantine, Ed Yourdon and Wayne Stevens.
- Jackson Structured Programming in circa 1975 developed by Michael A. Jackson
- Structured Analysis in circa 1978 with Tom DeMarco, Yourdon, Gane & Sarson, McMenamin & Palmer.
- Structured Analysis and Design Technique (SADT) developed by Douglas T. Ross
- Yourdon Structured Method developed by Edward Yourdon.
- Structured Analysis and System Specification published in 1979 by Tom DeMarco.
- Structured Systems Analysis and Design Method (SSADM) first presented in 1983 developed by the UK Office of Government Commerce.
- IDEF0 based on SADT, developed by Douglas T. Ross in 1985.[5]
- Information Engineering in circa 1990 with Finkelstein and popularized by James Martin.



www.idef.com

Welcome to IDEF.com

http://www.idef.com/

IDEF Integrated DEFinition Methods

IDEF0 IDEF1 IDEF1X IDEF3 IDEF4 IDEF5

Home » Welcome to IDEF.com

### IDEF Family of Methods

A Structured Approach to Enterprise Modeling & Analysis

IDEF.com was developed and is maintained by Knowledge Based Systems, Inc. (KBSI), the developers of the next generation IDEF methods: the IDEF3 Process Flow and Object State Description Capture Method, the IDEF4 Object-Oriented Design Method, and the IDEF5 Ontology Description Capture Method.

#### IDEF Method Reports

You can download all of the IDEF Method reports from this site. To place an order for a bound copy of any of the IDEF Method reports (\$20.00 per report), please e-mail requests to [IDEFinfo@kbsi.com](mailto:IDEFinfo@kbsi.com).

#### Automated IDEF Tools

KBSI has developed automated tools to support the IDEF0, IDEF1, IDEF1X, and IDEF3 methods. For more information on their complete suite of IDEF-based modeling and analysis software, see the [Tools](#) page on the KBSI website.

Select Language  
Powered by Google Translate

#### Downloads

- All IDEF Method Reports

#### KBSI Links

- KBSI Software
- Evaluate KBSI Software
- KBSI Sales

#### KBSI Community

- Offices, Phone, & Fax
- On the Web
- In the News
- Conferences & Events





# Características da modelagem e do sistema

**Sistema de  
Estados Finitos**



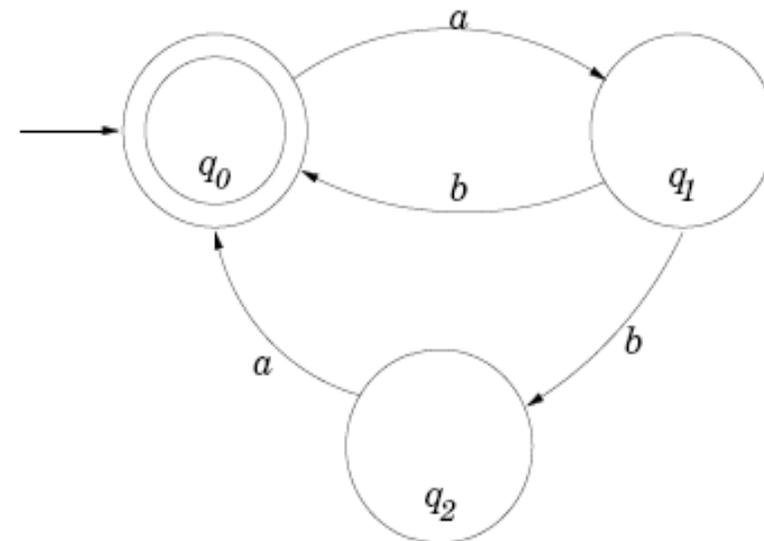
**Modelagem  
Estado-Transição**



# Modelagem com autômatos

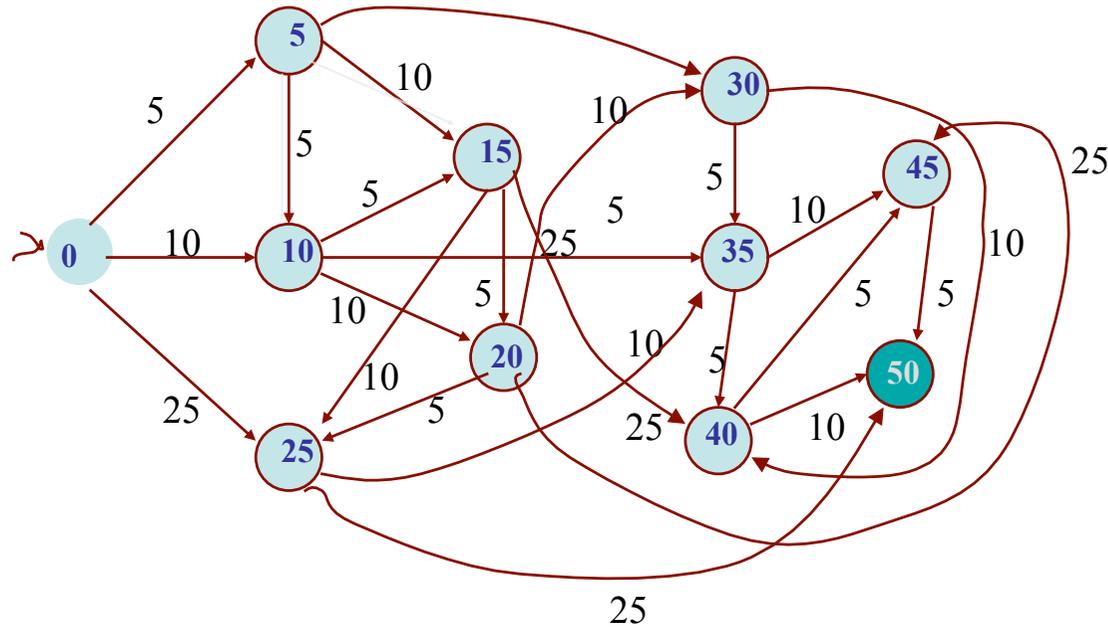
Example: Consider the finite automaton M in Figure 1. In this case,

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $q_1 = R(q_0, a)$ ,  $q_2 = R(q_1, b)$ ,  $q_0 = R(q_2, a)$ ,  $q_0 = R(q_1, b)$  (note that R is not a function)
- $\text{Init} = q_0$  (indicated by the straight arrow in Figure 1)
- $F = q_0$  (indicated by a double circle in Figure 1)
- $L(M) = \{\epsilon, ab, aba, abab, abaaba, \dots\} = ((ab)^n(aba)^m)^k \mu \Sigma^*$
- non-deterministic





# Um exemplo em Redes de Petri



Uma máquina de vender refrigerantes trabalha com moedas como estímulo (evento), onde estas podem ser de \$5, \$10 e \$25, e o refrigerante custa R\$0,50



# Definição de Rede de Petri

Podemos então introduzir os conceitos elementares para se definir uma rede, ou melhor o que passaremos a chamar de rede elementar.

Def.] Uma rede  $N$  é um grafo bipartido, não-nulo, direcionado, representado pela  $n$ -upla  $(S, T; F)$ , onde a relação de incidência  $F$ , aqui chamada de relação de fluxo é tal que  $F \subseteq (S \times T) \cup (T \times S)$ .

Se a rede  $N$  não possui laços, esta é dita pura. Se além disso a rede é simples, isto é, se não possui duas arestas distintas com os mesmos extremos – mesmo que estes não sejam coincidentes – então a rede é dita simples.

[www.informatik.uni-gamburg.de/TGI/PetriNets/](http://www.informatik.uni-gamburg.de/TGI/PetriNets/)



WELCOME TO NAKED OBJECTS

www.IDEF.com: Downloads Welcome to Naked Objects object oriented systems design ...

http://nakedobjects.net/home/index.shtml

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

GAME & APPS structured analysis exam Web Search Login 22°C

**NAKED OBJECTS GROUP**

- Home
- News
- Product
- Demo
- Licensing
- Downloads
- Resources
- About us

*"Perfection is finally attained not when there is no longer anything to add but when there is no longer anything to take away"*  
Antoine de Saint-Exupéry

**NAKED OBJECTS MVC**  
Turn a domain object model into a complete web application in minutes

Naked Objects MVC combines the power of the **naked objects** pattern with Microsoft's **ASP.NET MVC 3** framework.

Now you can take a POCO domain object model and turn it into a fully-functional web application in minutes, without writing a single line of user interface code.

You can then customise the generic user interface by adding custom style sheets, custom views and custom controllers, following standard ASP.NET patterns.

Read more **product details**, or better still ...

**Watch these Tutorial Videos:**

- Creating a new application**  
Creating a simple application from scratch using the 'Code First' approach
- Exploring a Naked Objects MVC**  
A closer look at the relationship between domain code and the user interface
- Customising the user interface**  
Customising using .css alone. Customising by adding new views

**Download**  
Evaluation version

**3 benefits from using Naked Objects MVC**

- A faster way to get started with MVC
- More productive development
- Easier maintenance



# Algumas respostas



Entender mais detalhadamente o processo de design

Antecipar a formalização

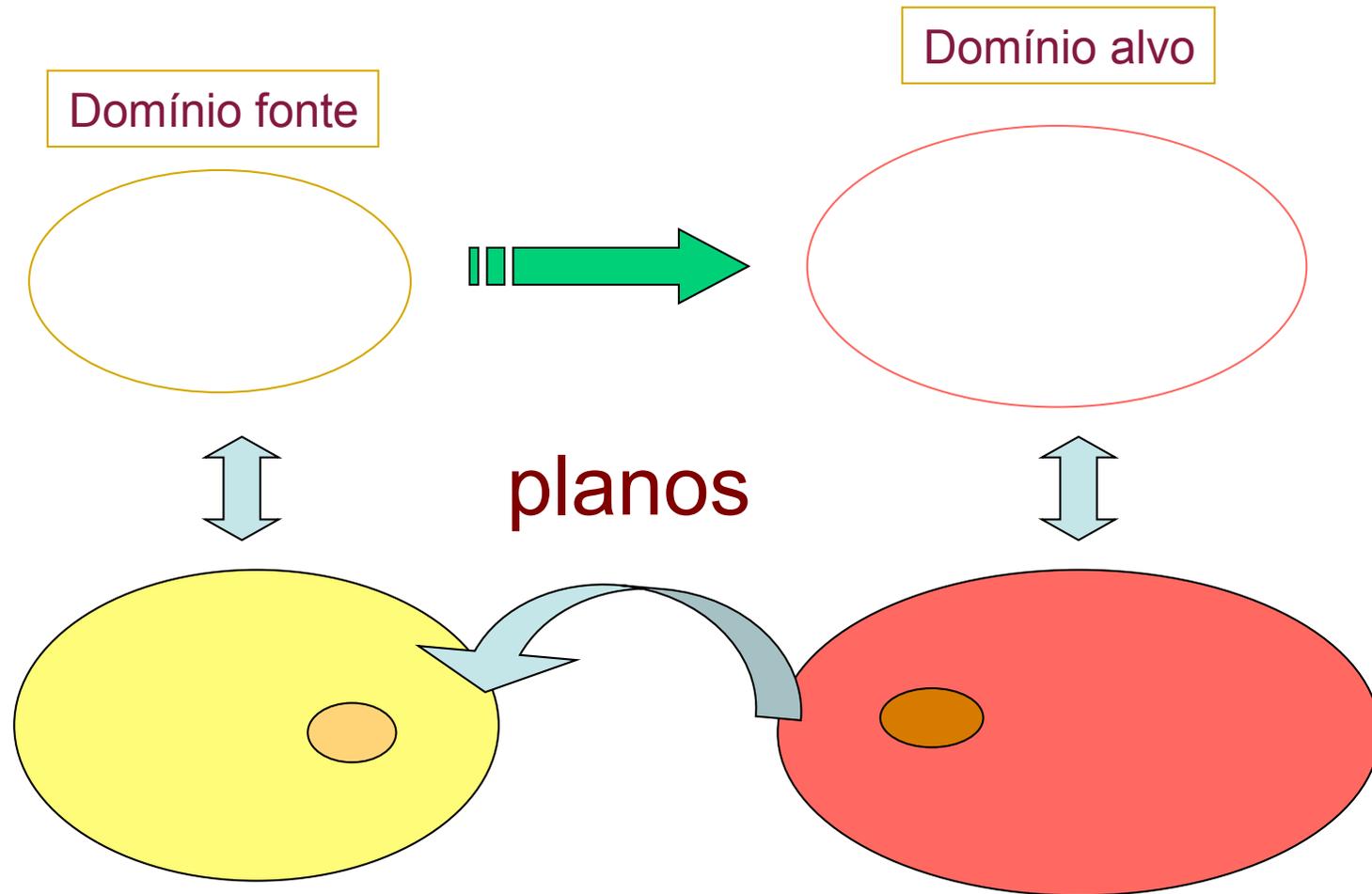
Basear a documentação em rationales

Reuso: privilegiando a experiência

Existe uma Teoria Geral do Design ?



# Abordagem em lógica de predicados de 1a. ordem





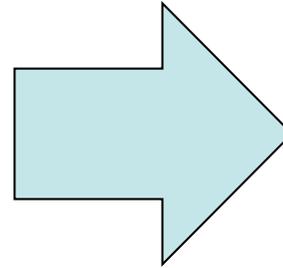
# Axiomatic Design

Criada no MIT pelos pesquisadores do Engineering Design Institute, por volta do início dos anos 90, tem hoje em Nam P. Suh seu principal e mais conhecido teórico.



# The Axiomatic Design Plan

Visão unificadora  
do design como  
atividade  
humana.



Síntese de um  
processo formal  
dividido em  
domínios.



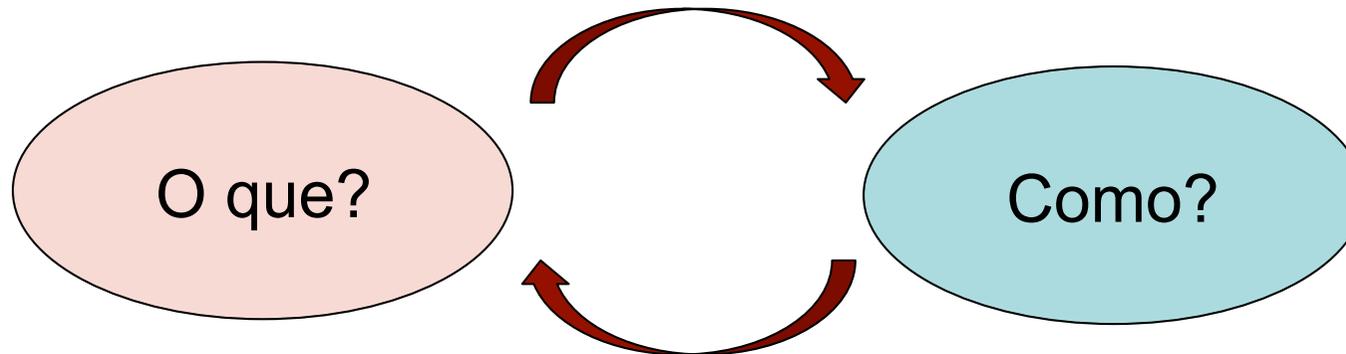
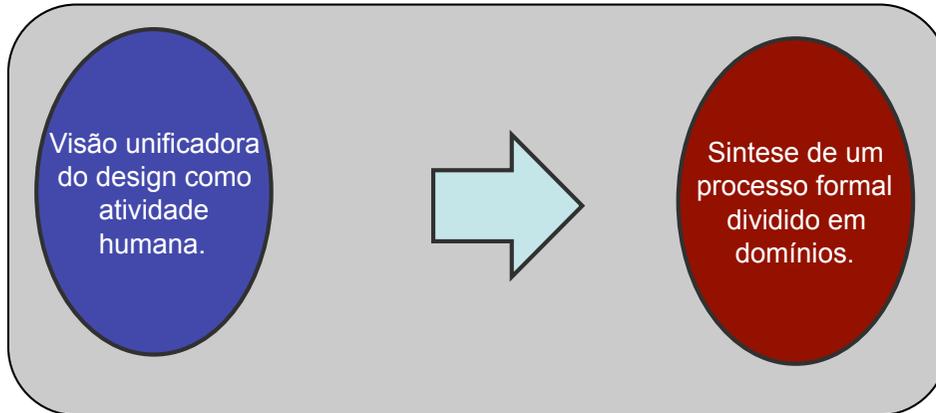
# Sintetizando o AD

Processo geral:

- i) Capturar as necessidades dos usuários (requisitos);
- ii) Definir o problema a ser resolvido para satisfazer estas necessidades;
- iii) Conceituar a solução por síntese;
- iv) Analisar e otimizar a solução proposta;
- v) Checar a solução resultante com respeito aos requisitos



# The Design Metaphor

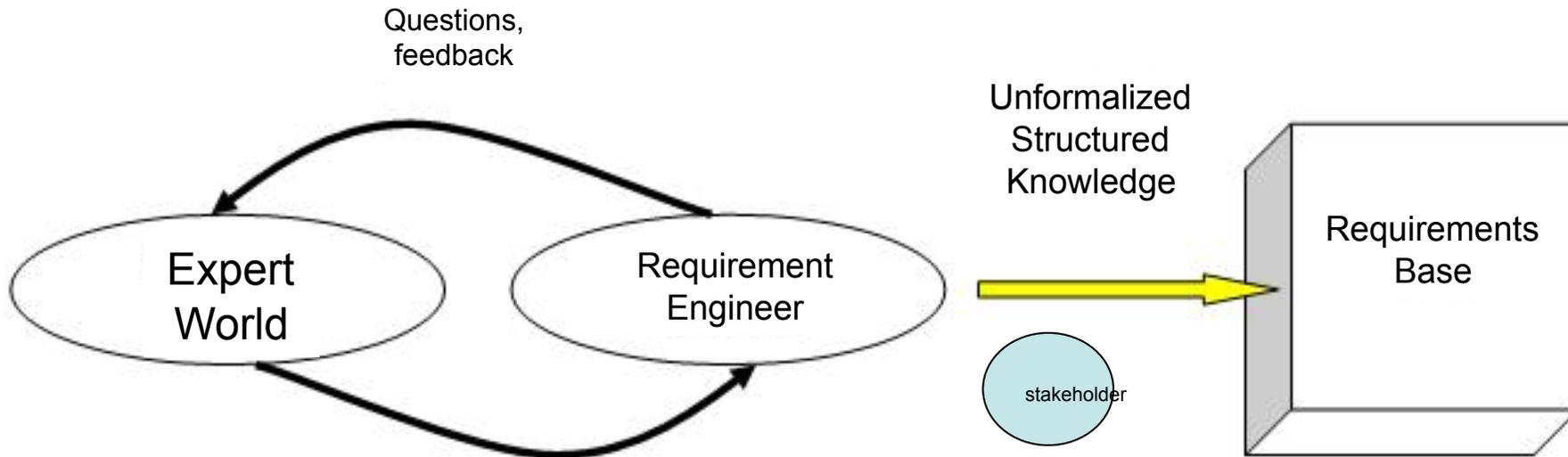
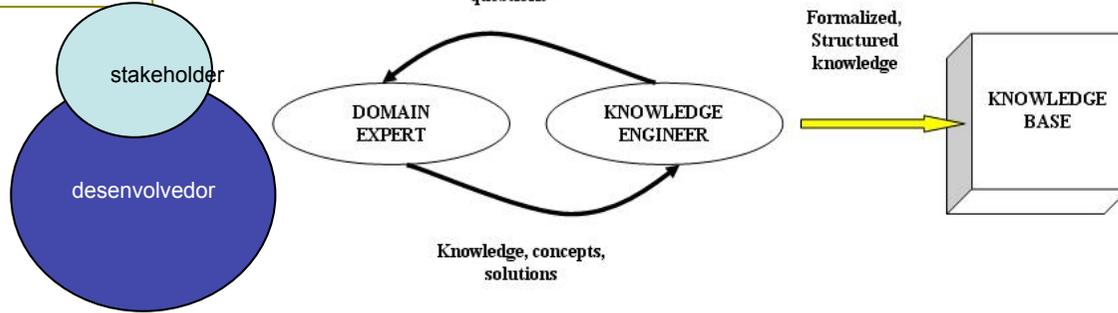


Convergencia?



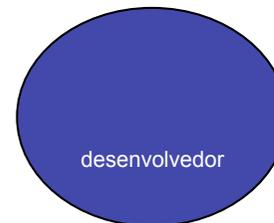
# O Processo de eliciação

IA,  
Modelagem formal



Knowledge, concepts,  
Application results

Heurísticas,  
Métodos esquemáticos





# Axiomatic Design

Os balizadores do Axiomatic Design são:

CA – Costumer Attributes

FR – Functional Requirements

DP – Design Parameters

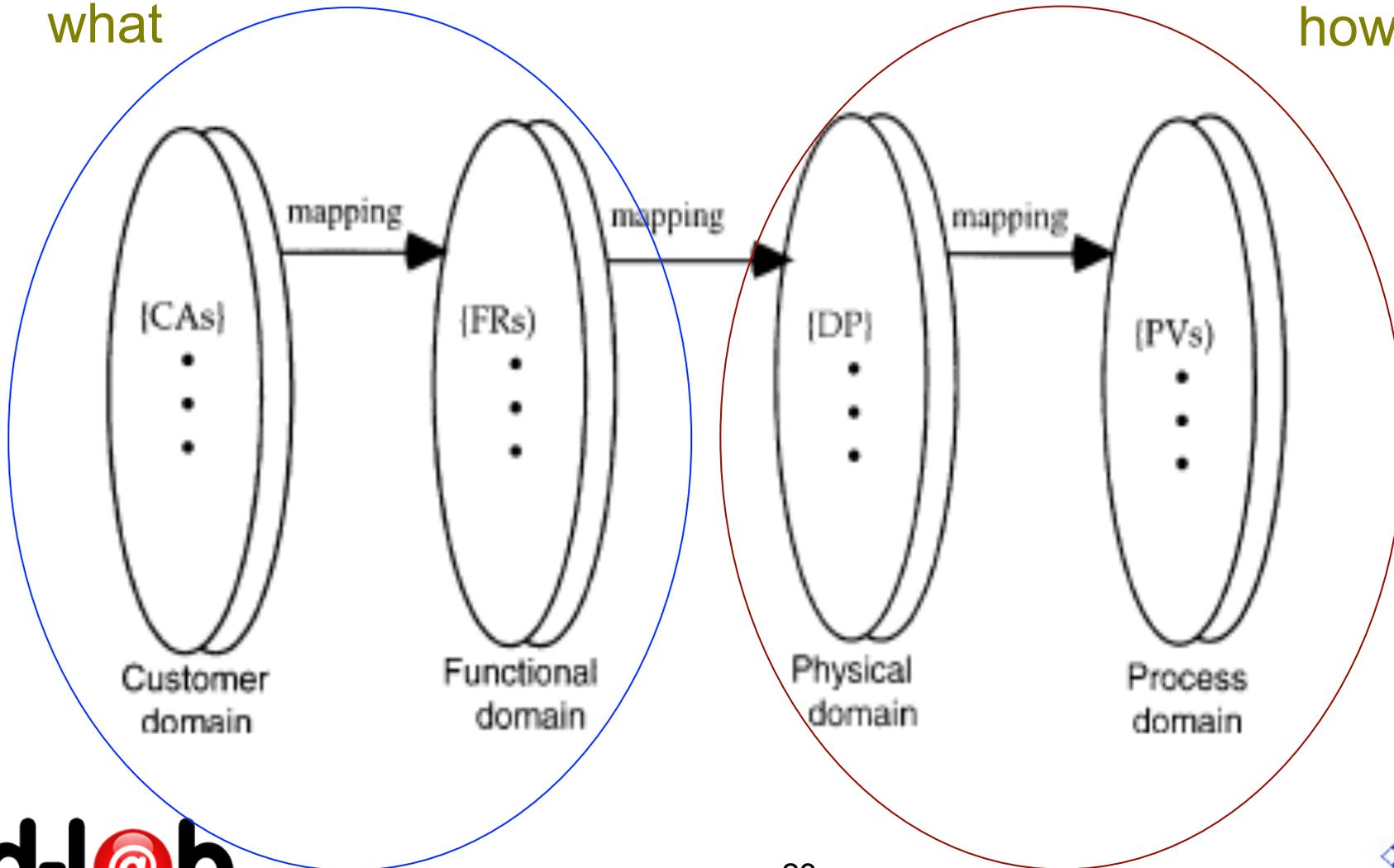
PV – Process Variables



# Domínios na ADT

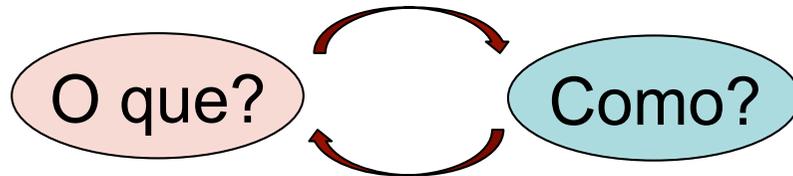
what

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.*



# Axiomatic Design: caracterização

Axiomatic design consists of:

- 1) domains in the design world,
- 2) mapping between these domains,
- 3) characterization of a design by a vector in each domain,
- 4) decomposition of the characteristic vectors into hierarchies through a process of zigzagging between the domains, and
- 5) the design axioms, Independence & Information Axioms.



## Characteristics of the four domains of the design world

Domains Character Vectors	Customer Domain {CAs}	Functional Domain {FRs}	Physical Domain {DPs}	Process Domain {PVs}
Manufacturing	Attributes which consumers desire	Functional requirements specified for the product	Physical variables which can satisfy the functional requirements	Process variables that can control design parameters (DPs)
Materials	Desired performance	Required Properties	Micro-structure	Processes
Software	Attributes desired in the software	Output Spec of Program codes	Input Variables or Algorithms Modules Program codes	Sub-routines machine codes compilers modules
Organization	Customer satisfaction	Functions of the organization	Programs or Offices or Activities	People and other resources that can support the programs
Systems	Attribute desired of the overall system	Functional requirements of the system	Machines or components, sub-components	Resources (human, financial, materials, etc.)
Business	ROI	Business goals	Business structure	Human and financial resource



# Customer attributes (CAs)

Em geral as necessidades do usuário serão reduzidas a uma frase ou mesmo a um conjunto de tokens ou palavras. Portanto a referência é em geral metafórica, especialmente na situação em que o domínio do stakeholder é próximo do desenvolvedor.

Não obstante, especialmente neste domínio é preciso mapear as necessidades do usuário com os requisitos funcionais (FR) de modo que haja um desacoplamento entre eles, na medida do possível.



# Quality Flow Deployment



Dr. Yoji Akao

"Time was when a man could order a pair of shoes directly from the cobbler. By measuring the foot himself and personally handling all aspects of manufacturing, the cobbler could assure the customer would be satisfied,"

*QFD foi criado no final dos anos 60 no Japão por Prof. Shigeru Mizuno e Prof. Yoji Akao. Os trabalhos pioneiros destes autores foram traduzidos por Glenn Mazur e levados de volta para o ocidente.*

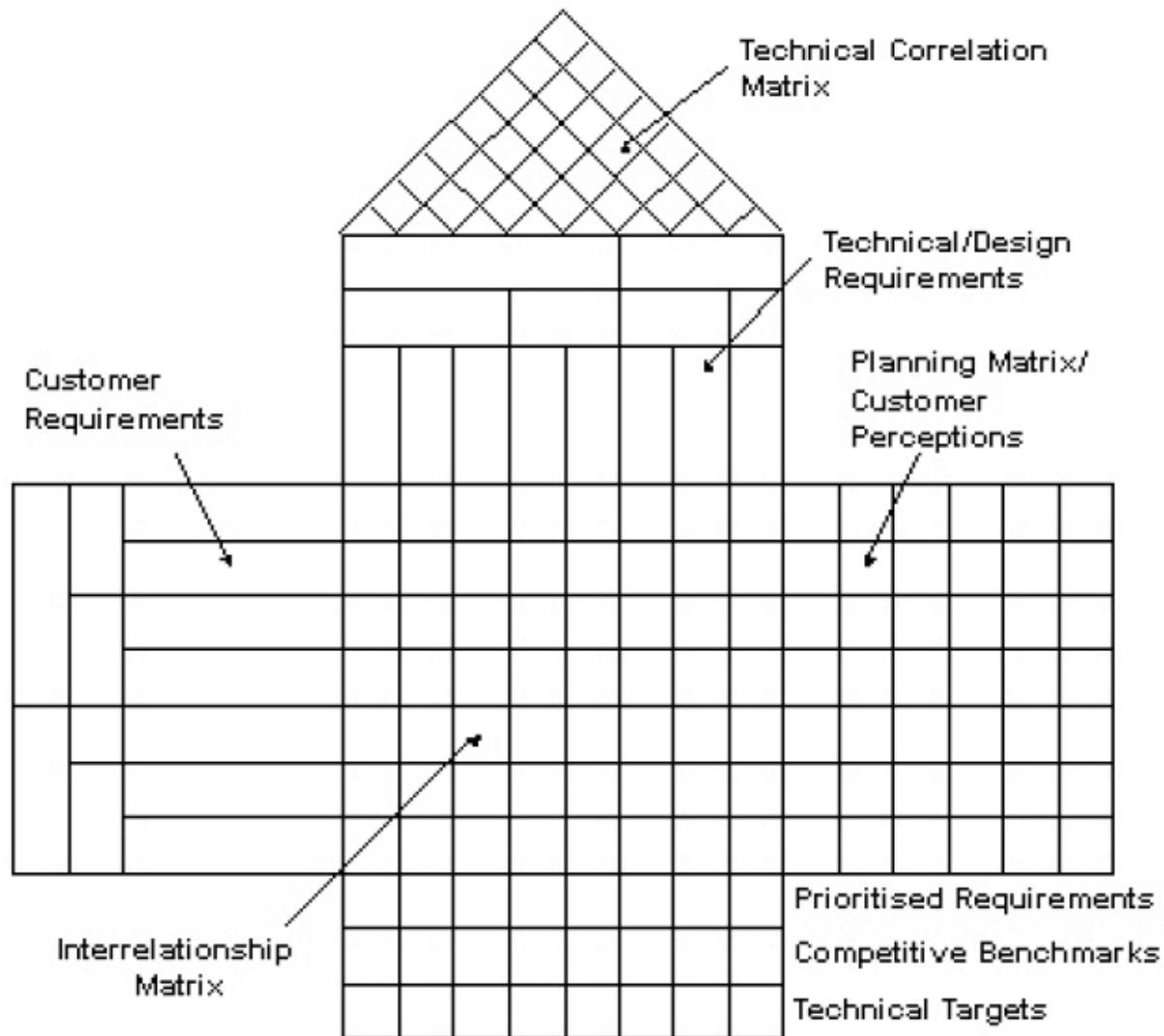
*Mais tarde, estes trabalhos inspiraram outros pesquisadores a propor o TQC e TQM.*

QFD: The Customer-Driven Approach to Quality Planning and Deployment (1994 Quality Resources: ISBN92-833-1122-1; written by Mizuno and Akao; translated by Glenn Mazur)

QUALITY FUNCTION DEPLOYMENT: Integrating Customer Requirements into Product Design (Productivity Press: ISBN 0-915299-41-0; written by Akao; translated by Glenn Mazur)



# House of Quality





# The House of Quality





QFD Institute. Home page of Quality Function Deployment Institute. The Official Source for QFD.

http://www.qfdi.org/

Getting Started Latest Headlines Apple Yahoo! Google Maps YouTube Wikipedia Notícias Popular

QFD Institute. Home page of Q... who is dr akao



# QFD INSTITUTE

The official source for QFD.

**Free QFD Newsletter**  
Email   
  
[Sample](#) | [Privacy](#)

---

**Home**

**Training**

- QFD Gold Belt®
- QFD Green Belt®
- QFD Black Belt®
- Quality Methods

**Calendar**

- Public Courses
- Symposium

**What is QFD**

- Case Studies
- Free Newsletter
- Reference Books
- QFD Software
- QFD Student Pack
- 

**About Us**

## QFD:Quality Function Deployment

is a comprehensive quality design method that:

1. Seeks out spoken *and* unspoken customer needs from fuzzy Voice of the Customer verbatim;
2. Uncovers "positive" quality that wows the customer;
3. Translates these into designs characteristics and deliverable actions; and
4. Builds and delivers a quality product or service by focusing the various business functions toward achieving a common goal—customer satisfaction.

**The QFD Institute** is a non-profit organization dedicated to dissemination and advancement of QFD through on-going R&D, current best practices and tools, and state-of-the-art QFD education programs and training methods.

Here you will find a comprehensive list of resources and opportunities to:

**Learn:**

- [Public QFD courses](#)
- [In-company training](#)
- [QFD certificate program](#)
- [Published QFD case studies & research](#)

**Upcoming Events**

**May 20**  
[Modern VOC and QFD Workshop](#)  
in UK

**QFD Green Belt®**

**June 23-24**  
[Public Course](#)  
Princeton, New Jersey USA

**September 23-24**  
[Public Course](#)  
Beijing, China

**October 22-23**  
[Public Course](#)  
Santa Fe, New Mexico USA

[In-company QFD Green Belt® Certificate Course](#)

Done



### **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.**



# No domínio Acadêmico...

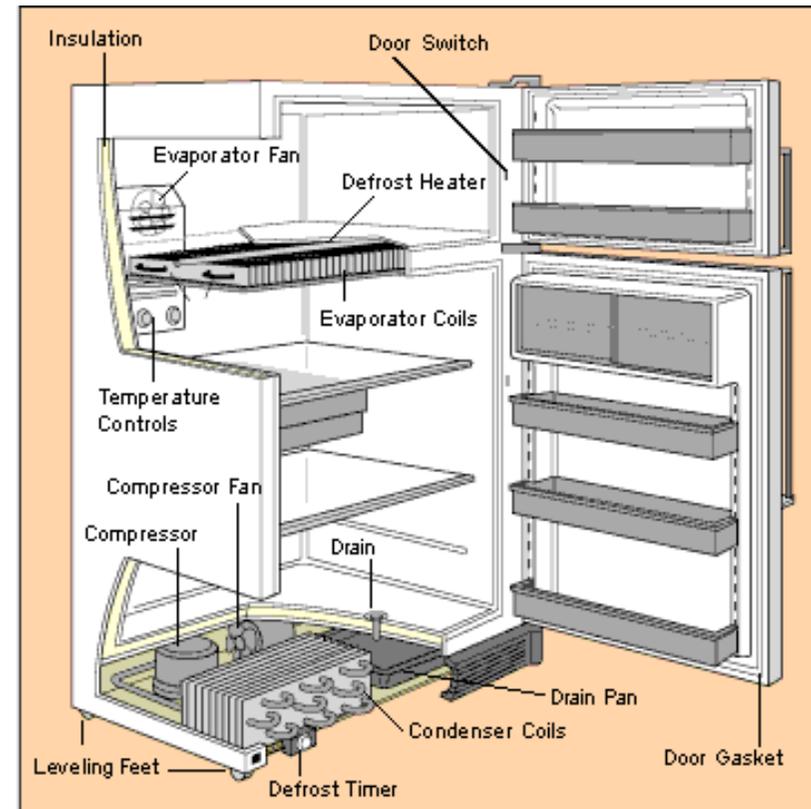
Customer Domain	Functional Domain	Physical Domain	Process Domain
CA1: Customer Satisfaction	FR1: Quality	DP1: Programs	PV1: Academic People
CA11: Undergrad students	FR11: Provide quality in undergrad education	DP11: Undergrad Program	PV11: Faculty involvement
CA12: Graduate students	FR12: Provide quality in in graduate programs	DP12: Graduate Program	PV12: Academically strong graduate programs
CA13: Research sponsors	FR13: Conduct quality research	DP13: Research Groups	PV13: Strong faculty
CA14: Public (society at large)	FR14: Show some commitment and participation in public activities	DP14: Service and Extension programs	PV14: Faculty involvement in extension activities.





## Um exemplo (clássico)

Como exemplo vamos tomar o design de um refrigerador, que tem certamente vários aspectos interessantes e fundamentais para o usuário. Vamos no entanto nos concentrar somente em um “feature” para poder avaliar mais profundamente o papel do AD na tomada de decisão de projeto (rationale).





# O Problema reduzido

Assim, o problema reduzido fica sendo o de decidir sobre o número e posição das portas do refrigerador para melhor atender aos “customer needs”.

**Prós e contras?**





# CAs para o refrigerador

- C11: um sistema para armazenar alimentos por muito tempo;
- C12: um sistema que também armazene alimentos por pouco tempo;
- C13: o manuseio do sistema é feito pelo usuário de pé;
- C14: o mecanismo de acesso deve combinar com a ergonomia usual;
- C15: a temperatura deve ser ajustável;
- C16: os demais utensílios domésticos usam força motriz elétrica;

C17: o sistema deve perder o mínimo de energia durante o manuseio;





# Síntese dos FRs para o refrigerador

- C11: um sistema para armazenar alimentos por muito tempo;
- C12: um sistema que também armazene alimentos por pouco tempo;
- C13: o manuseio do sistema é feito pelo usuário de pé;
- C14: o mecanismo de acesso deve combinar com a ergonomia usual;
- C15: a temperatura deve ser ajustável;
- C16: os demais utensílios domésticos usam força motriz elétrica;

**FR1: Congelador para conservação por longo período**

**FR2: Chiller para conservação por período curto**



# 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.



# A fase de design II

No caso mais simples e direto, a matriz de design **A** é tal que todos os seus elementos são constantes e cada **FR** é uma combinação linear dos **DPs**. Neste caso o design é dito *linear*. No caso geral os elementos da matriz **A** são funções dos DPs.



Além de ter uma noção mais precisa da complexidade do processo de design, considere também a diferença entre o que normalmente chamamos de “documentação” no caso do design axiomático.



# 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.



Tomando o caso onde  $A$  é quadrada  $m \times n$ :

- se  $A$  é uma matriz densa, denota um sistema fortemente acoplado, isto é, a alteração em um DP altera todos (ou quase todos) os FRs;
- se  $A$  é uma matriz triangular (superior ou inferior), denota um sistema parcialmente acoplado, onde existe pelo menos um mapeamento direto entre FR e DP;
- finalmente, se  $A$  é uma matriz **diagonal**, temos um sistema desacoplado e todos os FRs mapeiam cada um apenas um DP.



# Revendo o design do refrigerador

Vamos voltar ao problema do refrigerador, agora restrito ao problema do manuseio. Podemos portanto sintetizar dois FRs que dizem respeito ao acesso e à perda de energia no processo:

FR1 : Ter acesso direto ao conteúdo armazenado no refrigerador

FR2 : Minimizar a perda de energia



Cont.

FR1 : Ter acesso direto ao conteúdo armazenado no refrigerador

FR2 : Minimizar a perda de energia

Para estes FRs podemos deduzir pelo menos dois DPs:

DP1 : porta de acesso vertical (convencional)

DP2 : colocar isolante térmico na porta

Neste caso o design fica,

$$\begin{pmatrix} FR1 \\ FR2 \end{pmatrix} = \begin{bmatrix} X & 0 \\ X & X \end{bmatrix} \begin{pmatrix} DP1 \\ DP2 \end{pmatrix}$$



FR1 : Ter acesso direto ao conteúdo armazenado no refrigerador

FR2 : Minimizar a perda de energia

O Design pode ser completamente desacoplado mudando a solução, isto é, inserindo uma porta vertical que economiza energia na abertura e está diretamente associada ao acesso, nas outras situações (porta fechada) a única forma de economizar energia é pelo isolamento da porta (sem folgas). O design fica agora,

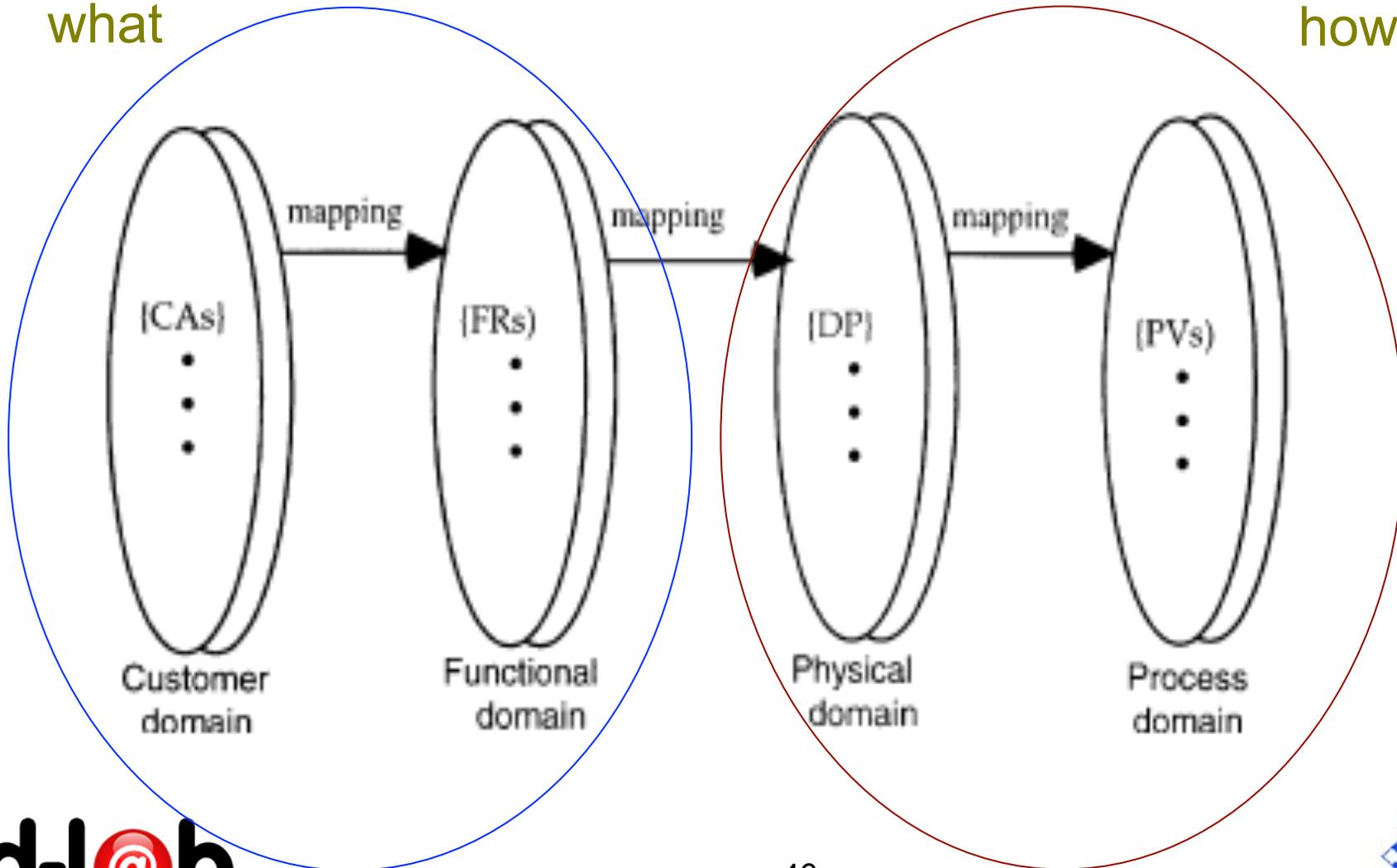
$$\begin{pmatrix} FR1 \\ FR2 \end{pmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{pmatrix} DP1 \\ DP2 \end{pmatrix}$$



# Domínios na ADT

what

how





# 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 .

$$DP = [B] PV$$



# Ambientes de design

Axiomatic Design Solutions, Inc.

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

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

**Leaner  
Smarter  
Design**

**Click to view the  
Acclaro DFSS Software  
Demonstration**



Pugh

**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



Talk with us...

**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.





FW Taylor Medal

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





Axiomatic Design Solutions, Inc.

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

http://www.axiomaticdesign.com/clients/university.asp

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

Solutions Services Software News/Events Clients Partners Technology Publications

**Clients**

- Overview
- Client Portal**
- University Portal
- Success Stories

**Acclaro DFSS for Universities**

Select your University:

University of Sao Paulo, Sao Paulo, Brazil

Continue

Your University not here? [Request Acclaro DFSS for your University students.](#)

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





Axiomatic Design Solutions, Inc.

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

http://www.axiomaticdesign.com/clients/university2.asp

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

Solutions Services Software News/Events Clients Partners Technology Publications

### Clients

Overview

Client Portal

- Lockheed Martin Space Divisions
- Northrop Grumman
- NASA-MSFC
- General Dynamics

University Portal

Success Stories

### University of Sao Paulo, Sao Paulo, Brazil

Request an Acclaro DFSS license key.

\* First Name:

\* Last Name:

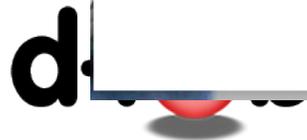
\* School  Your license will be sent to this address.

Email:  Your email must be from one of the following domains: usp.br,poli.usp.br

\* Password:

For more information, or to request a password, please contact Reinaldo Silva at reinaldo@usp.br.

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





# Exercício (um desafio)

Como leitura da semana teremos um artigo básico do Nam Piu Suh sobre Axiomatica Design e o exercício será o seguinte: vocês devem, baseados nos conceitos básicos do AD, ponderar se este método seria ou não adequado para o seu trabalho final e porque. Um texto deve ser feito sobre isso (veja que não estamos necessariamente falando do artigo, dado que a resposta para esta questão pode ser negativa). O que importa é trabalhar com os conceitos (básicos ainda) da AD.



*Fim*