

# ***PMR 5020***

## **Metodologia do Projeto de Sistemas**

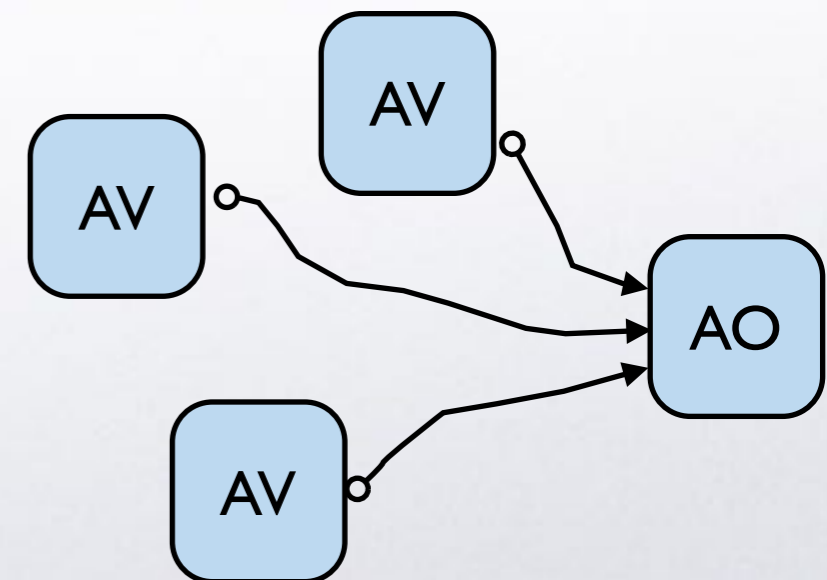
Aula 9: Uma ponte entre os métodos formais e os sistemas reais para o Projeto de Sistemas

# Alternativas para o Design de Sistemas

Até aqui vimos as seguintes alternativas para o design de sistemas:

- I. investir na fase inicial, na eliciação de requisitos, análise e especificação;
- II. a possibilidade de usar várias linguagens na fase de que vem após a especificação;
- III. a *esperança* de que a formalização inerente ao processo não seja o motivo de perda de informação ou desvio dos requisitos iniciais;
- IV. a possibilidade (especialmente para sistemas automatizados) de sofisticar o controle introduzindo sistemas inteligentes;

*Existe de fato um processo de design formal?*



## A busca por uma Teoria Geral do Design

A discussão sobre a formalização do processo de design começou em 1981 com a proposta de Hiroyuki Yoshikawa, e foi logo em seguida ampliada por Tetsuo Tomiyama, seu orientado. A polemica perdura até hoje e varia de alegações ao arcabouço teórico, à abordagem conceitual, até a perspectiva de aplicação.



Hiroyuki Yoshikawa



Tetsuo Tomiyama

**Axiom 1 (Axiom of Recognition)** Any entity can be recognized or described by the attributes.

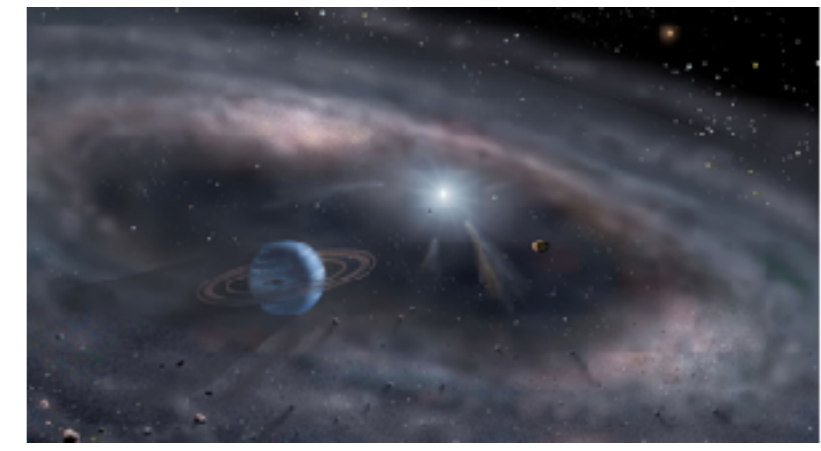
**Axiom 2 (Axiom of Correspondence)** The entity set  $S'$  and the set of concept of entity (ideal)  $S$  have one-to-one correspondence.

**Axiom 3 (Axiom of Operation)** The set of abstract concept is a topology of the set of entity concept.

Yoshikawa, H. (1981). General Design Theory and a CAD system, In: *Man-Machine Communication in CAD/CAM*, Sata, T. and Warman, E. (eds.), pp. 35-58, North-Holland, Amsterdam.

# The System of Systems Challenge

A practical obstacle to the formalization of design is the practical effectiveness of this approach, specially in this era of complexity. Generally, formal approaches do not fit the complexity of large systems (of systems).



# Designing large Service Information Systems

## Novo SIS

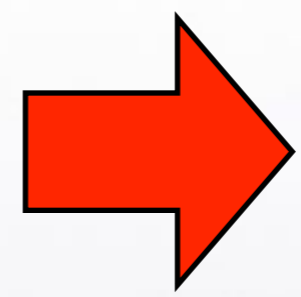
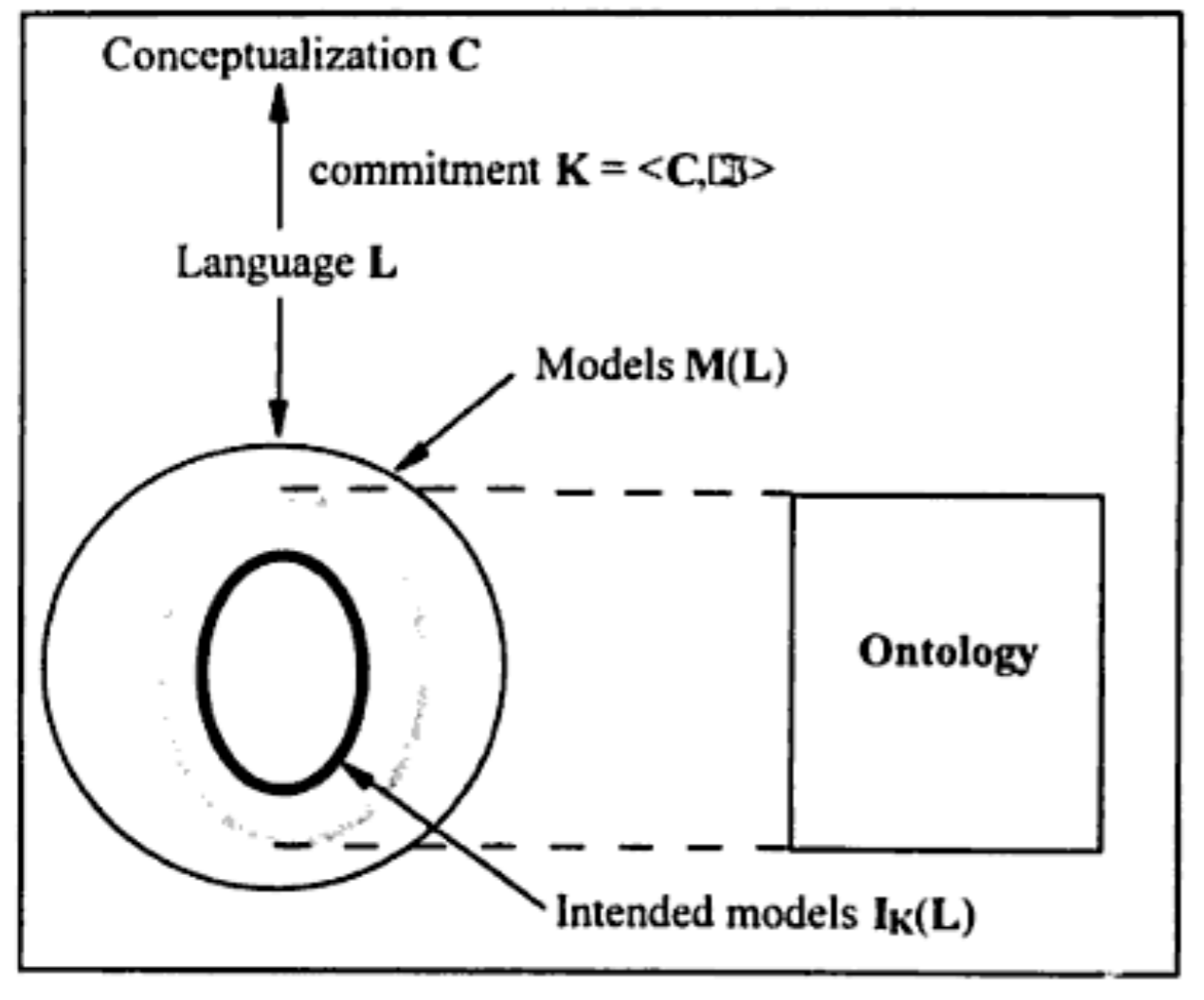


Sistemas de informação conjugam flexibilidade e capacidade de integração, fundamental para inovação e automação.[1]  
Convergência entre sistemas de serviço e sistemas de informação. [2]

- [1] Stair, R.; Reynolds, G. "Information Systems", 9th ed., Course Technology, 2010.  
[2] Bardhan, I. ; Demirkan, H.; Kannan, P.; Kauffman, R.; Sougstad, R. "An Interdisciplinary Perspective on IT Services Management and Service Science". *Journal of Management Information Systems*, v. 26, n. 4, p. 13-64, 2010.

... or designing really intelligent systems

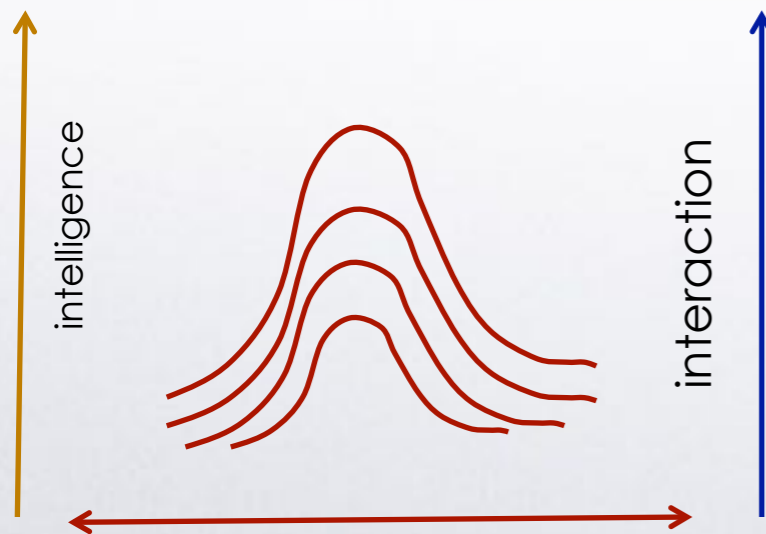
# Ontology driven IS



Artificial Intelligence

... with embedded intelligent control

From server to service and  
up to symbiosis



## A alternativa formal

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.



Nam Pyo Suh



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.

## Elementos do processo de projeto

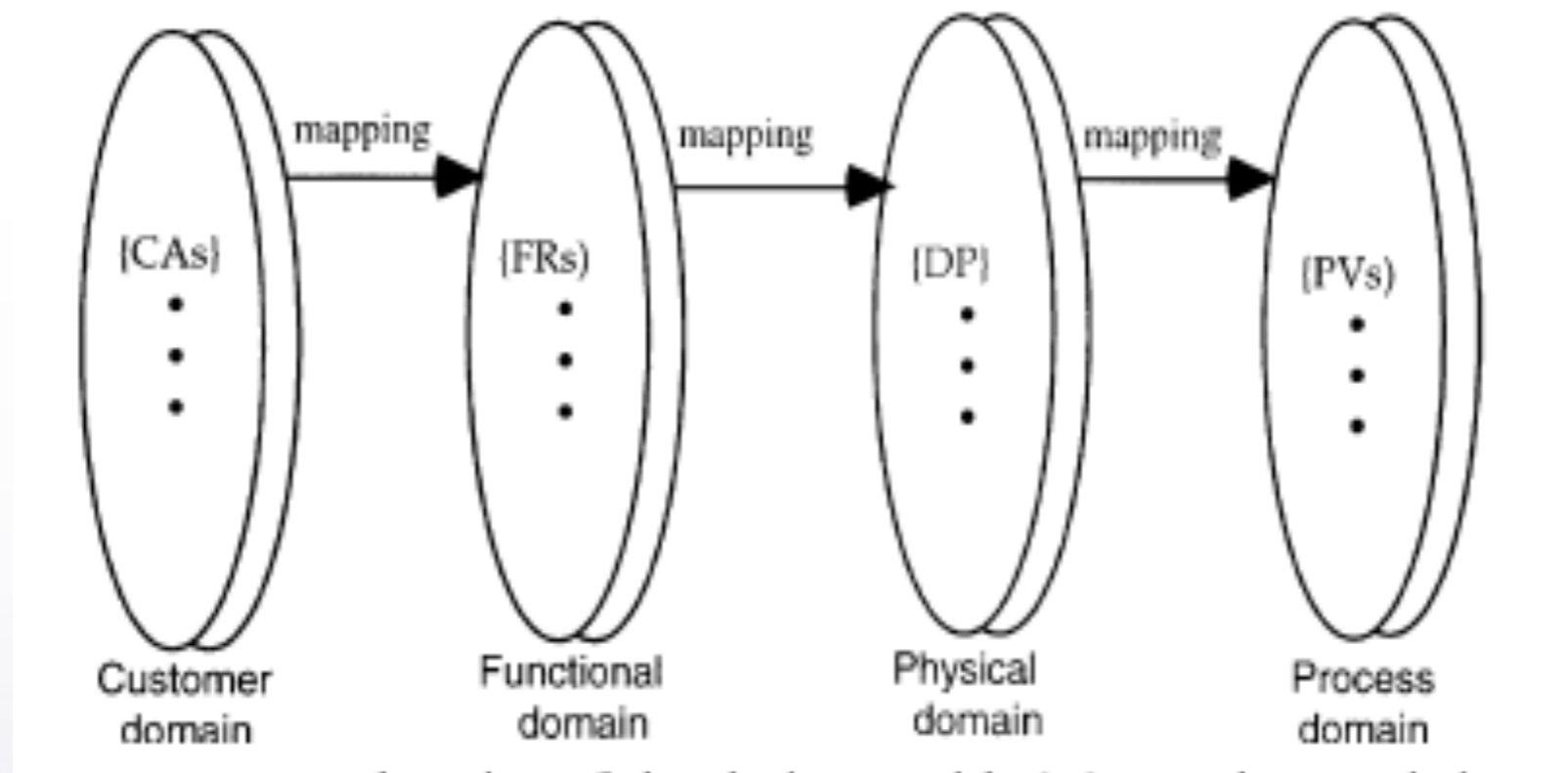
Os balizadores do Axiomatic Design são:

FR – Functional Requirements

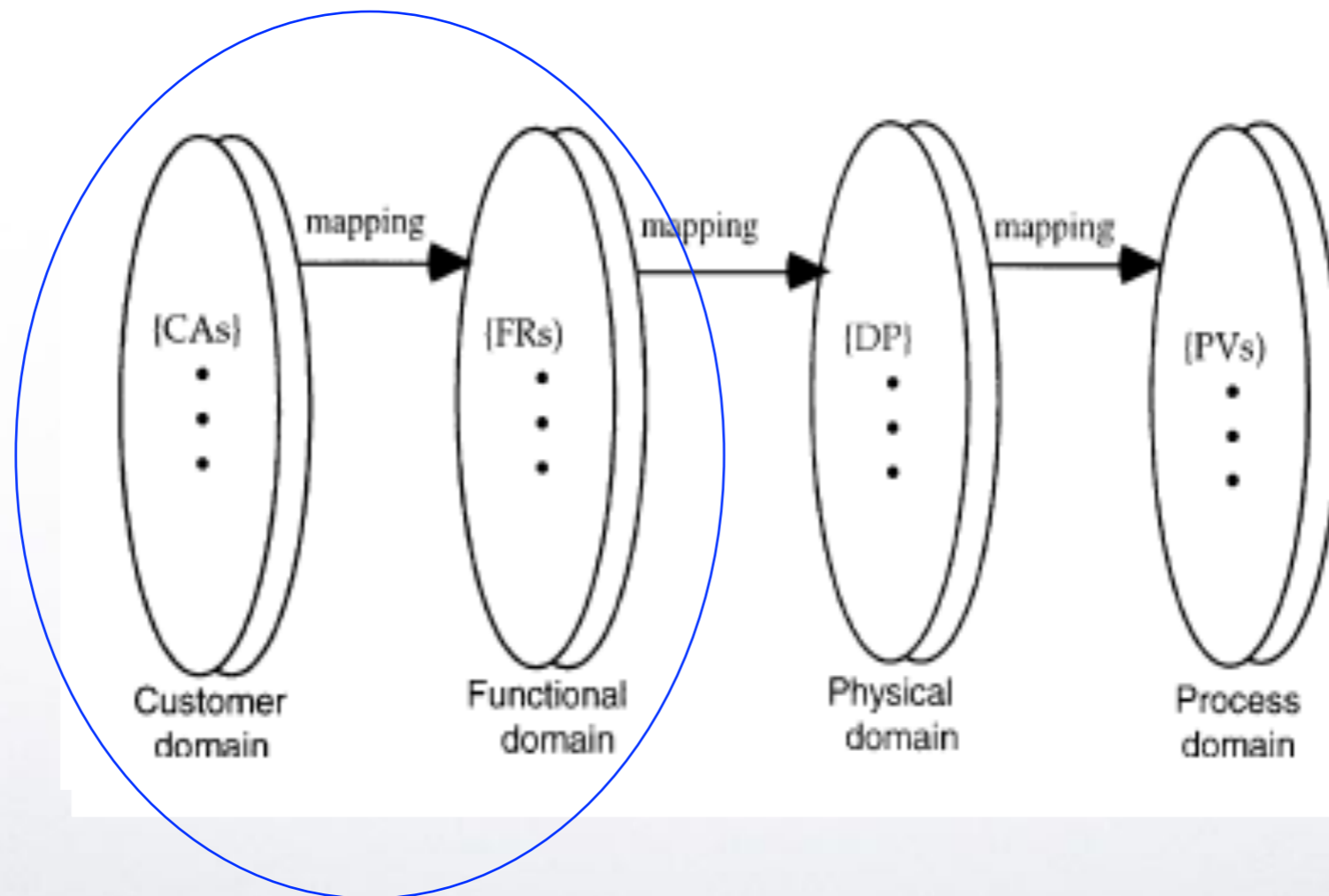
DP – Design Parameters

PV – Process Variables

# Dominios do processo de projeto

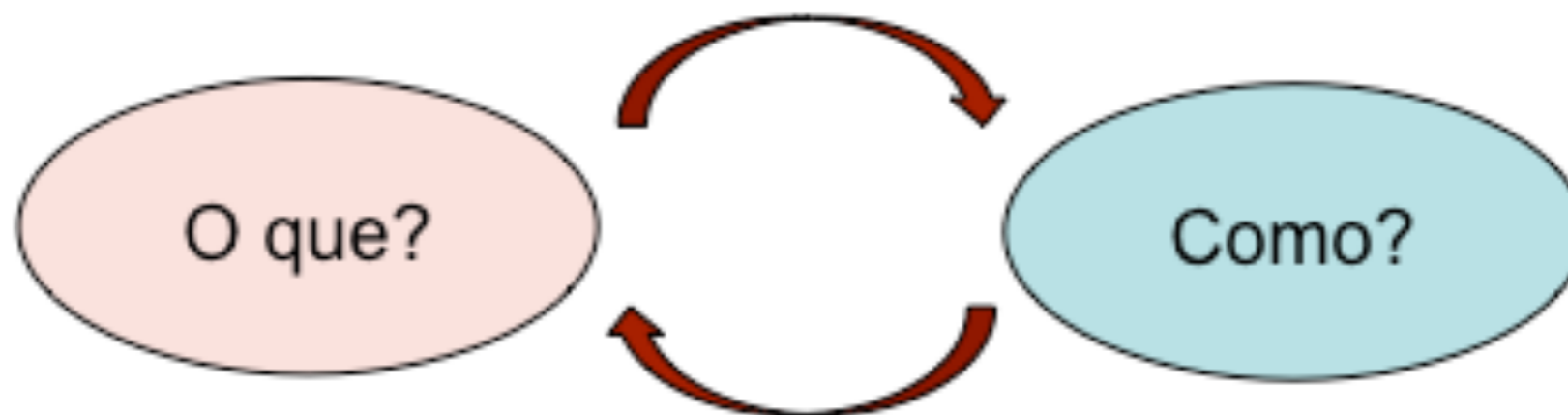
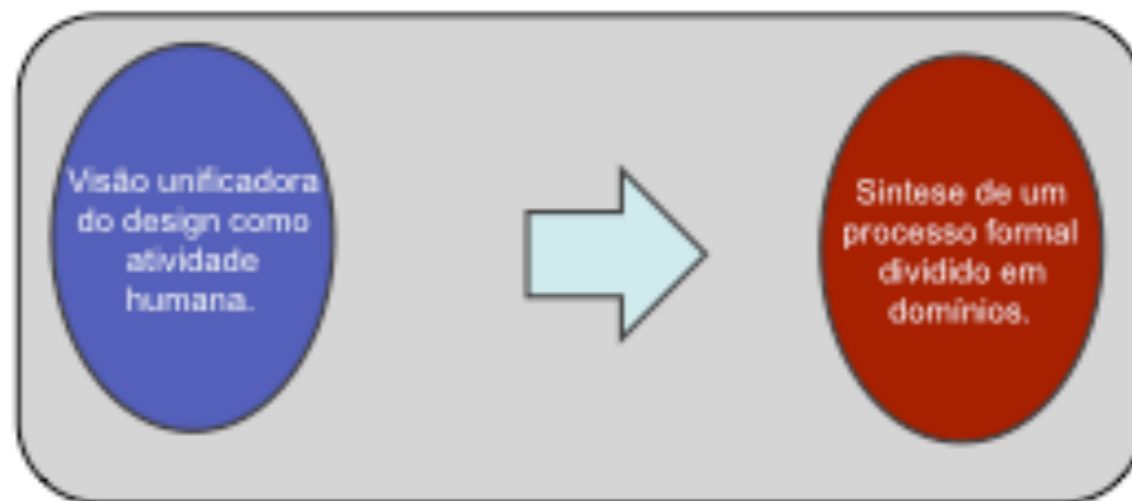


# The functional approach

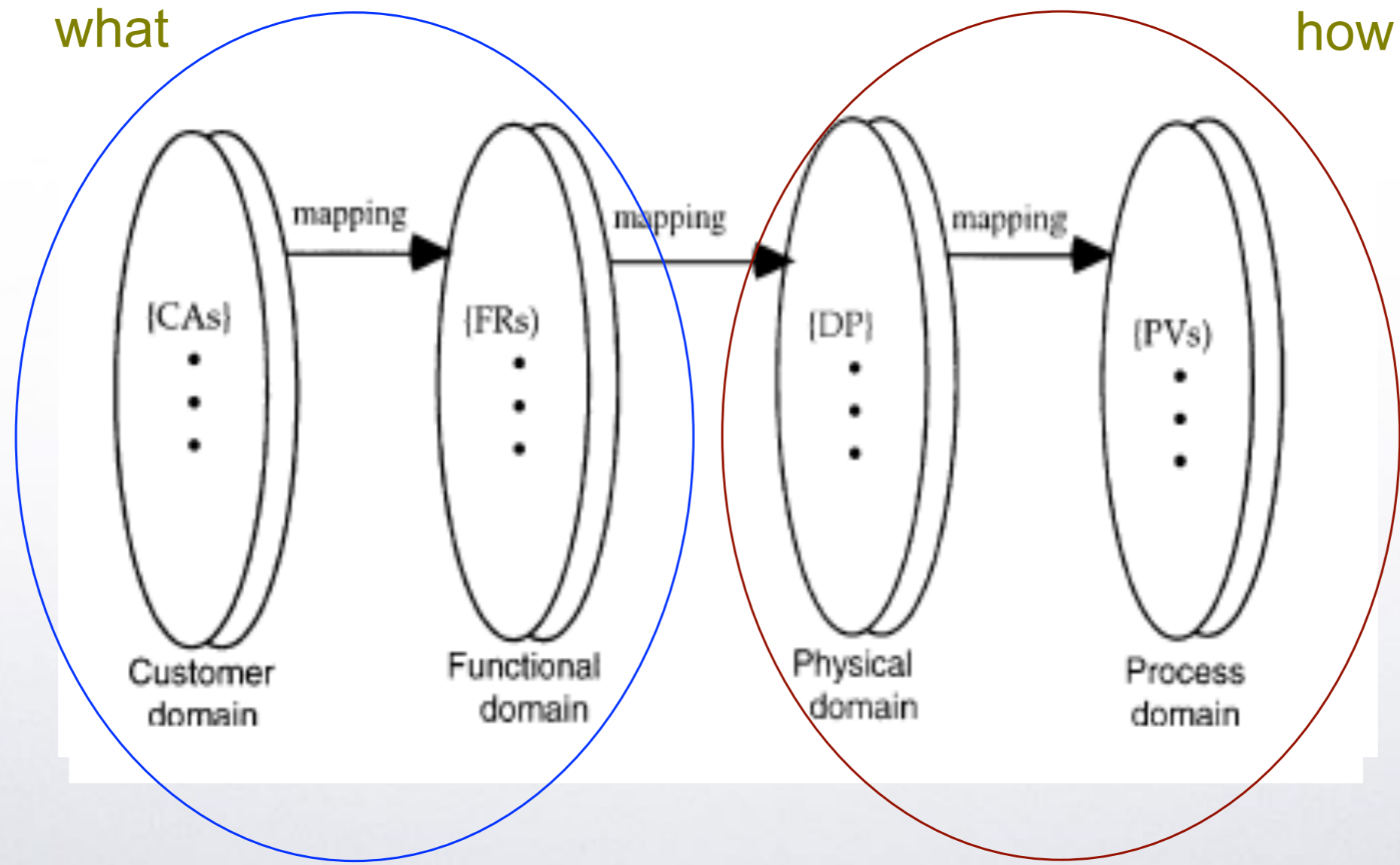


early phase

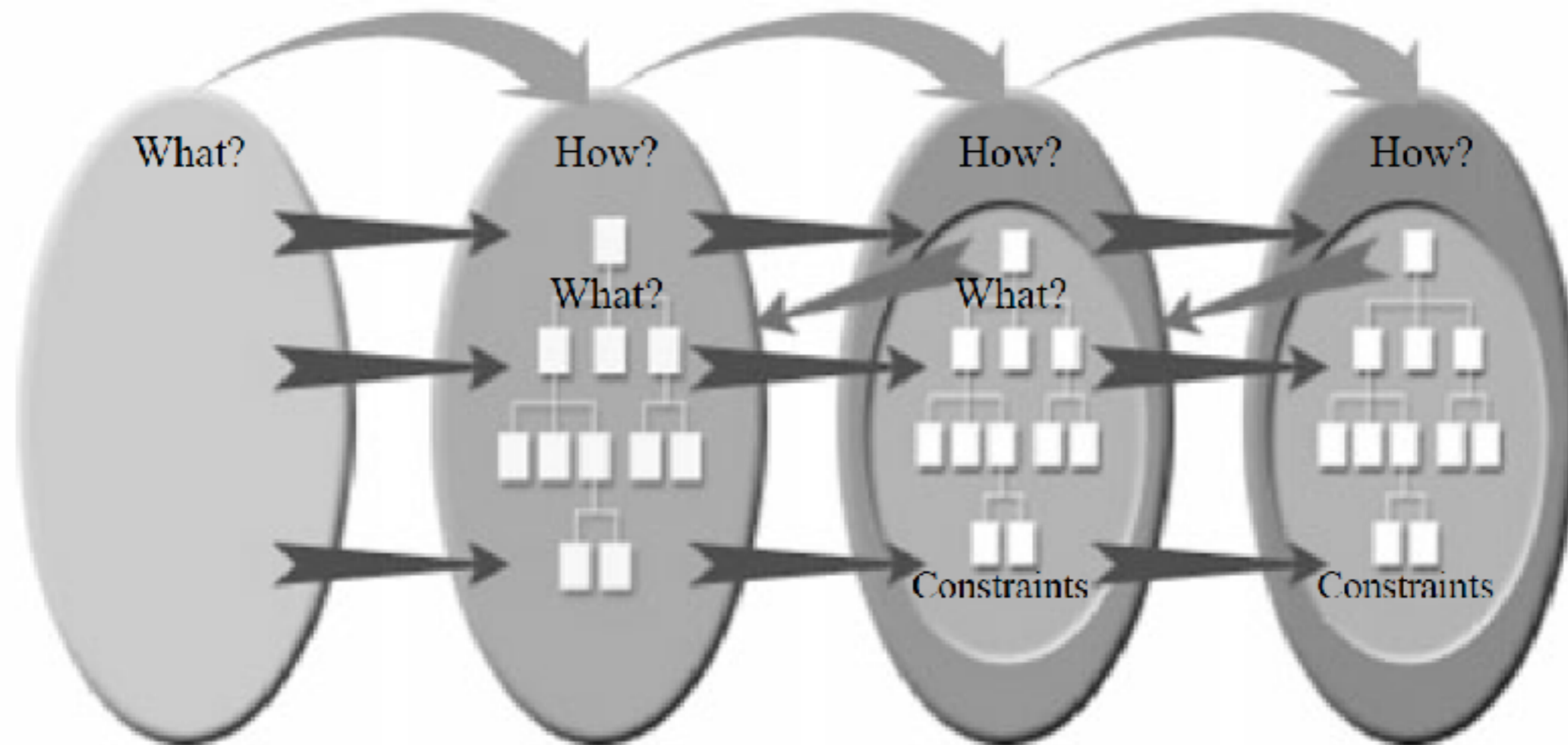
# Os domínios do design



# Um mapeamento "ingênuo" para o Axiomatic Design

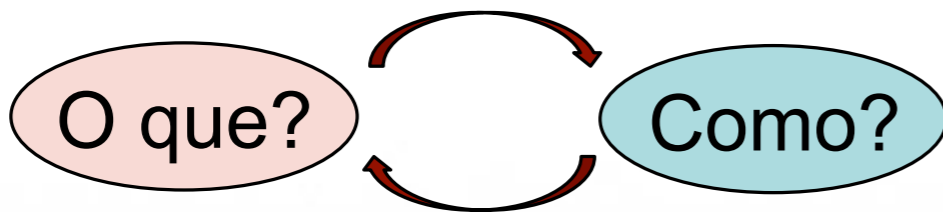


Customer domain      Functional domain      Physical domain      Process domain



Customer needs      Functional requirements      Design parameters      Process variables

## Os Axiomas

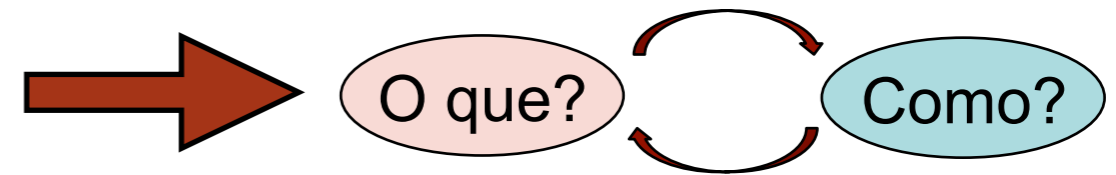


**Axioma 1:** *Axioma da Independência*  
*Mantenha a independência entre os FRs;*

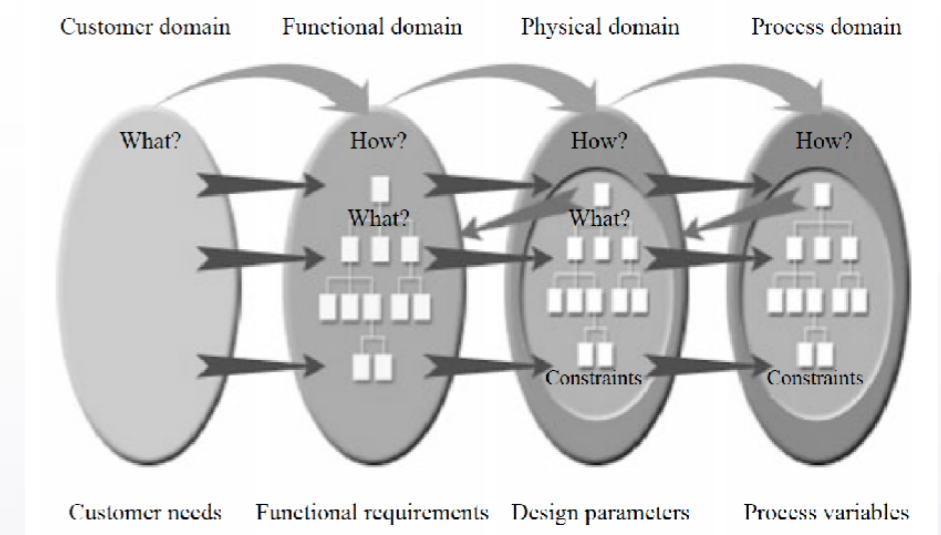
**Axioma 2:** *Axioma da Mínima Informação*  
*O Melhor design é aquele que minimiza a informação sobre o artefato.*



# A fase de "especificação"



A fase de eliciação e análise de requisitos (resultando na especificação) é aqui reduzida à identificação dos “Costumer Attributes” (CAs) que por sua vez induzem a identificação dos respectivos “Functional Requirements” (FRs). Entretanto não há a prescrição de nenhum processo formal (ou mapeamento) e nem a garantia de descoplamento se aplica a esta fase, isto é, não há garantias de que um conjunto de FRs atenda ou modele de forma independente um ou mais CAs.



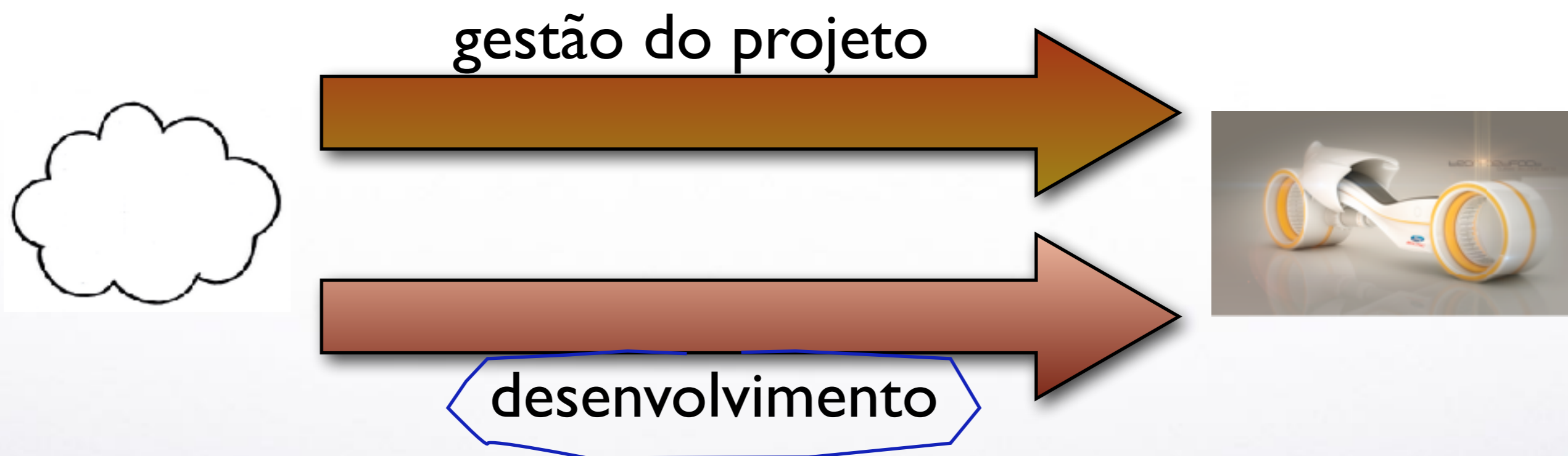
## A fase de design

A fase reconhecida como design 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.

## O processo de projeto



O axiomatic design não inclui a gestão no processo de desenvolvimento

## **Axiom 1: The Independence Axiom**

Maintain the independence of FRs.

*Alternate Statement 1:* An optimal design always maintains the independence of FRs.

*Alternate Statement 2:* In an acceptable design, DPs and FRs are related in such a way that a specific DP can be adjusted to satisfy its corresponding FR without affecting other functional requirements.

Em um design aceitável os *functional requirements* estão linearmente associados aos *design parameters*,

$$\mathbf{FR} = \mathbf{A} \mathbf{DP}$$

Em uma situação “ideal” a dimensão do vetor FR e DP é a mesma e a matriz de transformação - chamada **matriz de design** - é quadrada. Vamos supor que a  $|\mathbf{FR}|=|\mathbf{DP}|=3$ , somente para simplificar a notação. Assim,

$$\begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \end{bmatrix}$$

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.

Teorema 1] Seja um design 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.

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

Teorema 2] Seja um design  $D$  acoplado, onde o número de DPs é menor que o número de FRs. Se  $D$  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  $D$  acoplado deve ser dividido em partes, bem como a sua solução.

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



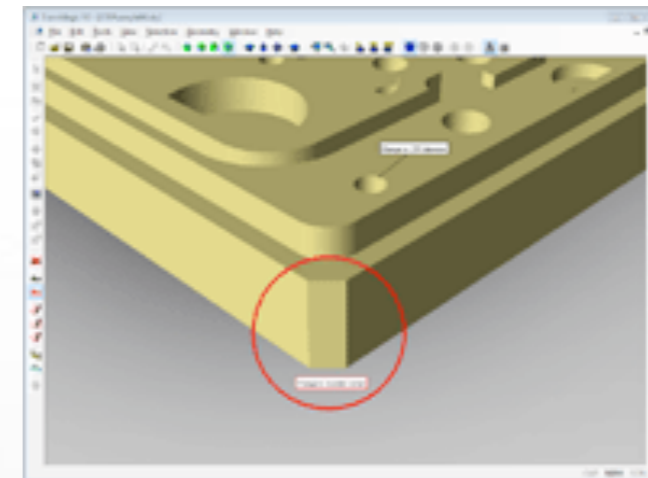
Teorema 3] Seja um design D acoplado, onde o número de DPs é maior que o número de FRs. Neste caso o design pode ser acoplado ou redundante.

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.

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

## Introduzindo "features"

**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 ...



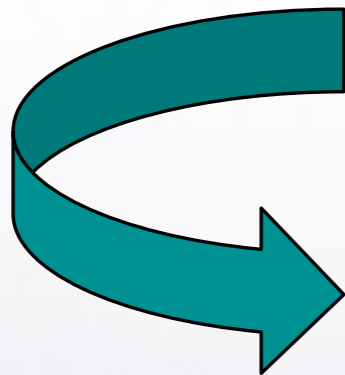
Corolário 3] É possível integrar *design features* em um único componente físico se os FRs puderem ser satisfeitos independentemente nesta nova solução.

Teorema 4] Um design  $D$  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

Teorema 5] Seja um design 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

	Design equation	Design process
Uncoupled design	$\begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \end{bmatrix}$	$FR_1 = A_{11} \times DP_1$ $FR_2 = A_{22} \times DP_2$ $FR_3 = A_{33} \times DP_3$
Decoupled design	$\begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \end{bmatrix}$	$FR_1 = A_{11} \times DP_1$ $FR_2 = A_{21} \times DP_1 + A_{22} \times DP_2$ $FR_3 = A_{31} \times DP_1 + A_{32} \times DP_2 + A_{33} \times DP_3$
Coupled design	$\begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \end{bmatrix}$	$FR_1 = A_{11} \times DP_1 + A_{12} \times DP_2 + A_{13} \times DP_3$ $FR_2 = A_{21} \times DP_1 + A_{22} \times DP_2 + A_{23} \times DP_3$ $FR_3 = A_{31} \times DP_1 + A_{32} \times DP_2 + A_{33} \times DP_3$

## Um exemplo de design (de produto)

O exemplo a seguir foi proposto por Nam Suh em 2000, sobre um design do NSF de 1998, e trata de um feature específico no projeto de um aparelho refrigerador: o posicionamento da porta.

$FR_1$  : Provide access to the items stored in the refrigerator.

$FR_2$  : Minimize energy loss.

### Solution

Design parameters for the vertically hung door in Figure 2.2a are as follows:

$DP_1$  : Vertically hung door

$DP_2$  : Thermal insulation material in the door

Os requisitos funcionais neste caso seriam o acesso aos ítems dentro do refrigerador e a perda de energia que pode ser causada pela abertura da porta (DP1) ou pela vedação (DP2). O acesso depende somente do ângulo de abertura da porta, portanto o acoplamento é dado pela equação abaixo, onde a matriz de projeto é triangular.

$$\begin{bmatrix} FR_1 \\ FR_2 \end{bmatrix} = \begin{bmatrix} X & 0 \\ X & X \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \end{bmatrix}$$



Uma outra possibilidade (note que neste caso temos que montar outra equação), é dada pelo uso de uma porta horizontal no topo do refrigerador. A vantagem é que o ar frio se distribui mais na parte de baixo e a perda pela porta é desprezível se for aberta um número pequeno de vezes. A perda pela vedação é mais importante. Assim, a equação anterior ficaria,

$$\begin{bmatrix} FR_1 \\ FR_2 \end{bmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \end{bmatrix}$$

$FR_1$  : Provide access to the items stored in the refrigerator.

$FR_2$  : Minimize energy loss.

$DP_1$  : Horizontally hung door

$DP_2$  : Thermal insulation material in the door

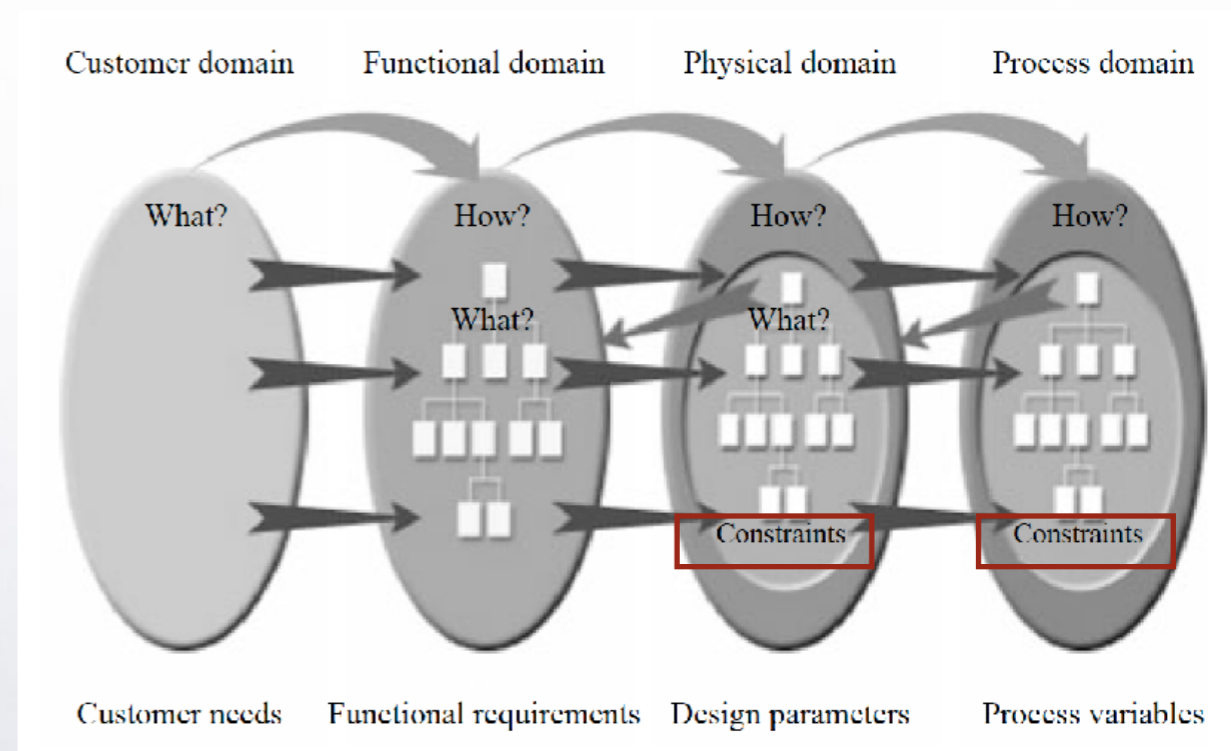




# Usando restrições (constraints).

Certamente a porta horizontal seria a solução ideal para os FRs considerados. Entretanto se inserirmos mais FRs ou um constraint dado pela necessidade do usuário de ver o conteúdo antes de escolher o que quer retirar do refrigerador, a solução horizontal seria proibitiva por organizar o conteúdo na forma de pilha impedindo a visão do que está em baixo da pilha.

Como as restrições devem ser consideradas primeiro, antes de determinar os DPs, a solução horizontal sequer seria aventada se tivéssemos esta restrição.



## Exemplo: torneira com misturador

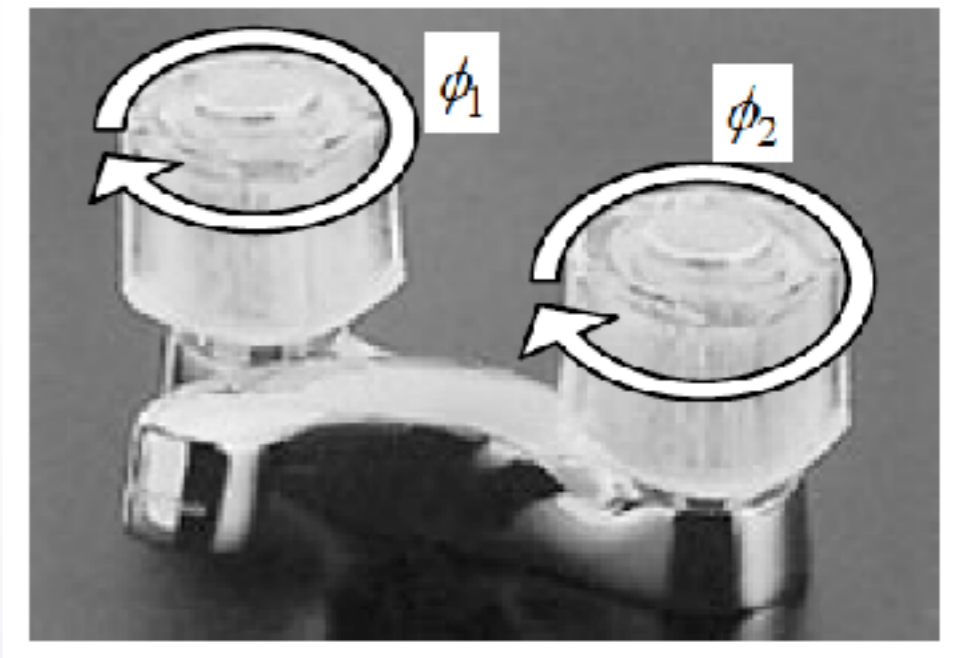
Um outro problema certamente emblemático para o Axiomatic Design é a torneira com misturador de água quente e fria.

$FR_1$  : Control the flow of water ( $Q$ ).

$FR_2$  : Control the temperature of water ( $T$ ).

$DP_1$  : Angle  $\phi_1$

$DP_2$  : Angle  $\phi_2$



A solução convencional pra este problema com duas torneiras implica que para o  $FR_1$ , isto é, para o controle do fluxo de água, a posição das duas válvulas contribui.

Similarmente, para o controle da temperatura os dois ângulos também contribuem e a solução convencional prevê um sistema acoplado, que segundo o padrão apresentado .

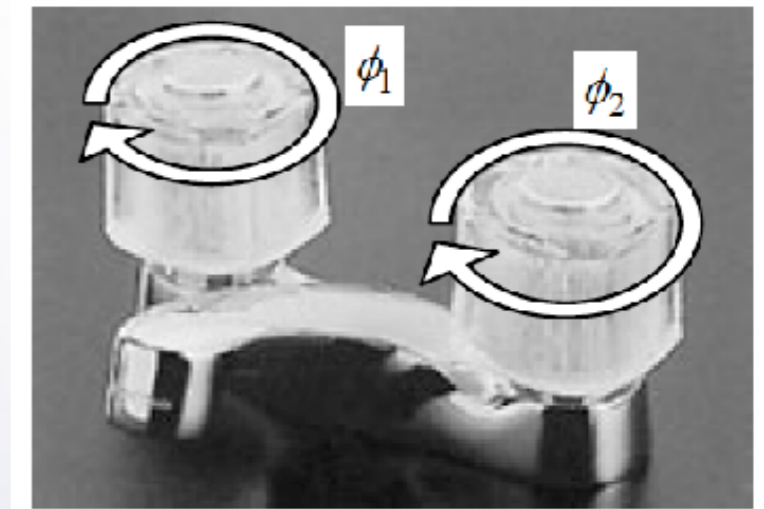
$FR_1$  : Control the flow of water ( $Q$ ).

$FR_2$  : Control the temperature of water ( $T$ ).

$DP_1$  : Angle  $\phi_1$

$DP_2$  : Angle  $\phi_2$

$$\begin{bmatrix} FR_1(Q) \\ FR_2(T) \end{bmatrix} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \begin{bmatrix} DP_1(\phi_1) \\ DP_2(\phi_2) \end{bmatrix}$$



Uma segunda solução pode ser encontrada para melhorar o acoplamento. Neste caso podemos encontrar uma outra seleção de DPs onde se lida com um ângulo para a válvula de admissão de água quente e fria. Um deslocamento vertical levanta a válvula e permite a entrada de água quente e fria em parte proporcionalmente iguais.

Neste caso o controle da vazão depende exclusivamente do deslocamento vertical, enquanto a temperatura depende unicamente do ângulo da válvula de vedação. Temos neste caso a equação de design fica,

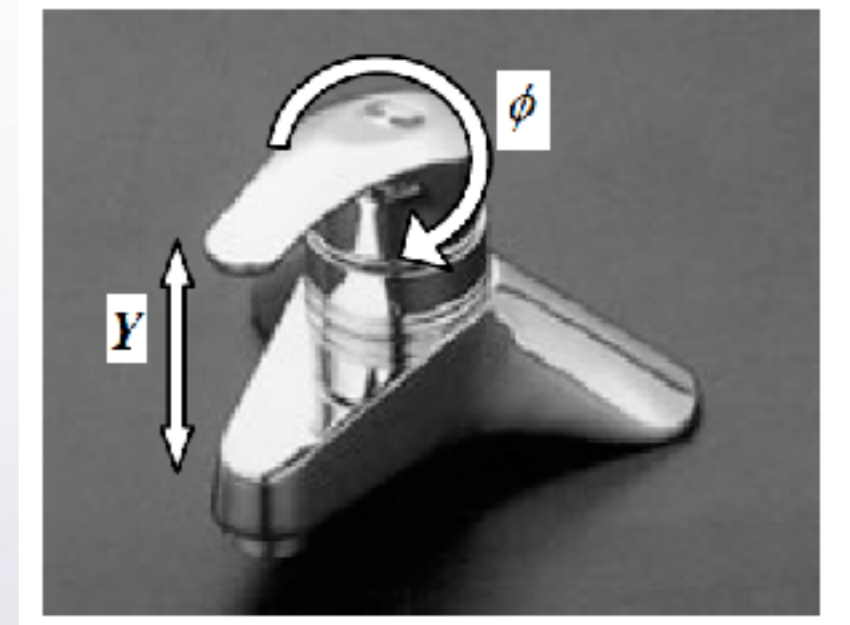
$$\begin{bmatrix} FR_1(Q) \\ FR_2(T) \end{bmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{bmatrix} DP_1(Y) \\ DP_2(\phi) \end{bmatrix}$$

$FR_1$  : Control the flow of water ( $Q$ ).

$FR_2$  : Control the temperature of water ( $T$ )

$DP_1$  : Displacement  $Y$

$DP_2$  : Angle  $\phi$



## AD-SD: Axiomatic Design - 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-SD: Axiomatic Design - 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  $D_1$  e  $D_2$ , que têm os mesmos requisitos funcionais e restrições semelhantes (são funcionalmente equivalentes) e que possuem o mesmo conteúdo informacional. Apesar de serem equivalentes,  $D_1$  e  $D_2$  podem ter uma estrutura de custo completamente diferente.

## AD-SD: Axiomatic Design - 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-SD: Axiomatic Design - 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-SD: Axiomatic Design - 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”.

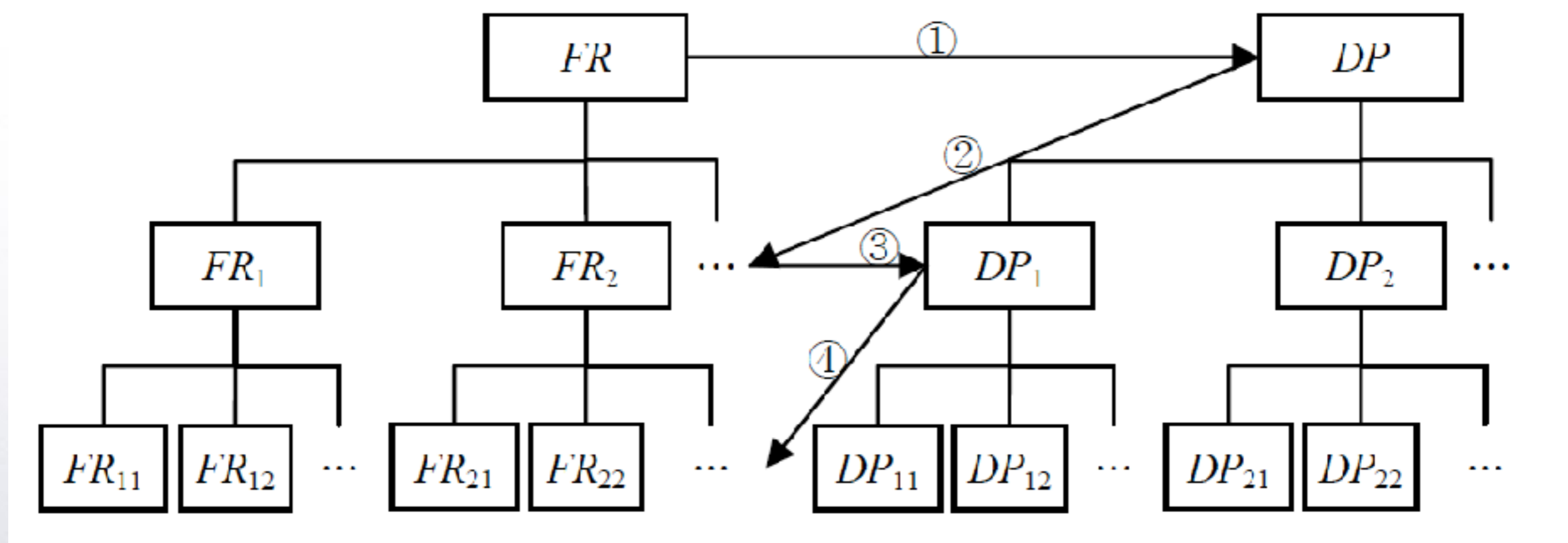
Teorema S5] Quando não se conhece a priori uma partição completa do conjunto dos FRs à qual o sistema deve satisfazer durante o seu ciclo de vida, não existe nenhuma garantia que um dado design minimizará o conteúdo de informação para todos os possíveis subconjuntos e portanto que um dado design é o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

## *AD-SD: Axiomatic Design - 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.

# Ziguezaguing



Axiomatic Design Solutions, Inc.

www.axiomaticdesign.com/technology/axiomatic.asp

Essential... Course M... Humanis... ScholarCI... media.wi... Academi... Edit Pro... Axiomat... artifact

**AXIOMATIC DESIGN SOLUTIONS, INC.** Support / WebEx / eTraining / Contact Us

Solutions Services Software News/Events Clients Partners Technology Publications

**Technology**

- Overview
- Introduction
- Value Proposition
- Axiomatic Design
- Benefits
- Process Benefits
- Technical Papers
- Links

**Axiomatic Design Technology**

Axiomatic design technology reduces product development risk, reduces cost, and speeds time to market by:

- Formalizing the conceptual design process into a continuous and measurable activity driven by requirements.
- Communicating the state of the design to all stakeholders at the earliest possible moment, well before traditional CAD documentation.
- Improving quality of design by analyzing and optimizing design architectures.
- Providing explicit traceability from Customer Needs to Requirements to Design Logic to Design.
- Clearly documenting and communicating the logical How and why of a design, not just the What of CAD documentation.
- Permitting design issues to be identified early and resolved without the cost of design-build-test-redesign cycles.
- Providing project management with the dependency structure of the design, enabling optimal scheduling and risk mitigation.

---

- General axiomatic design concepts
- Software specific axiomatic design concepts

**General Axiomatic Design Concepts**

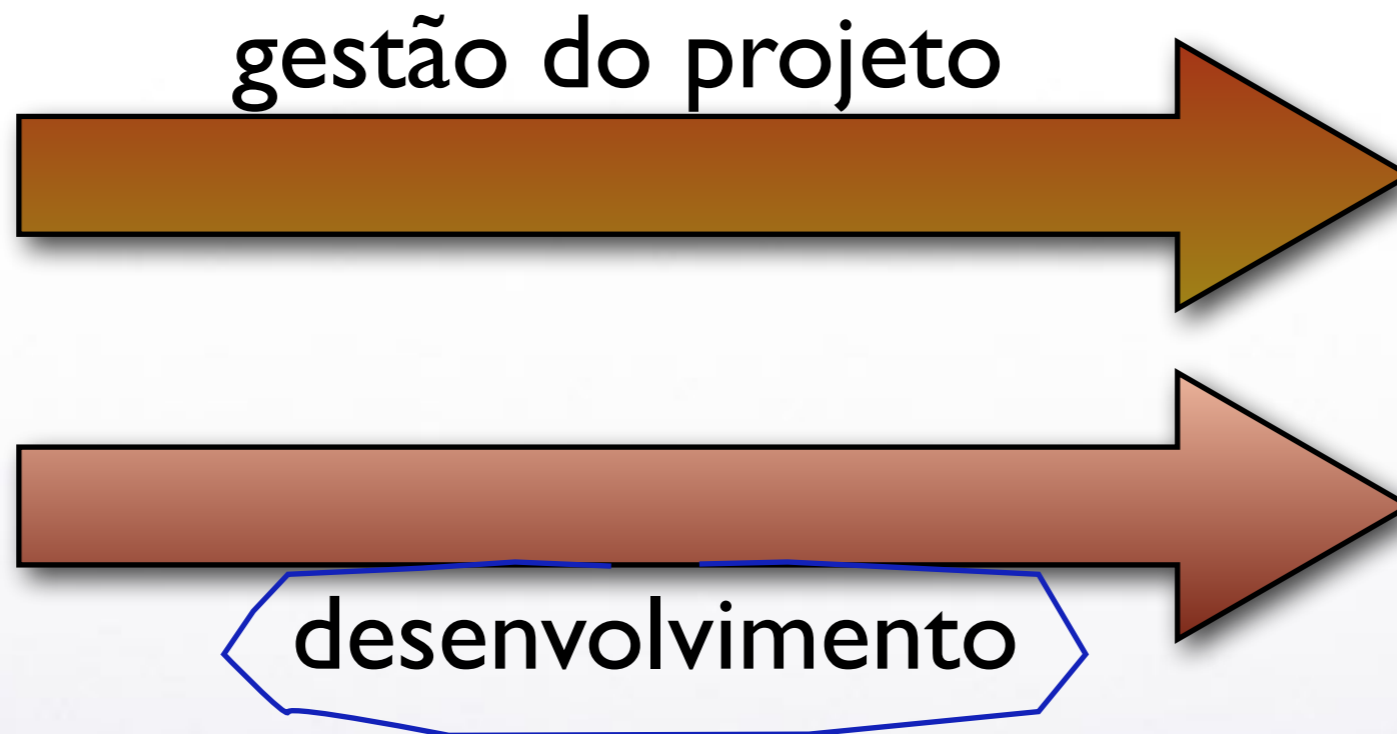
Whether the design solution is a tangible product, service, software, process, or something else, designers typically follow these steps:

1. Understand their customers' needs
2. Define the problem they must solve to satisfy those needs
3. Create and select a solution
4. Analyze and optimize the proposed solution
5. Check the resulting design against the customers' needs

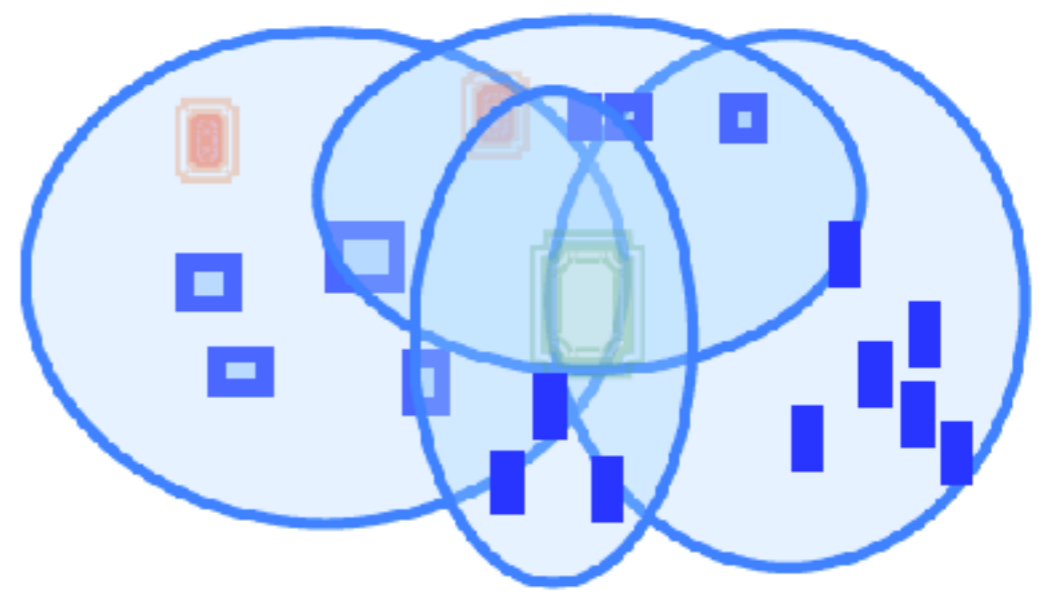
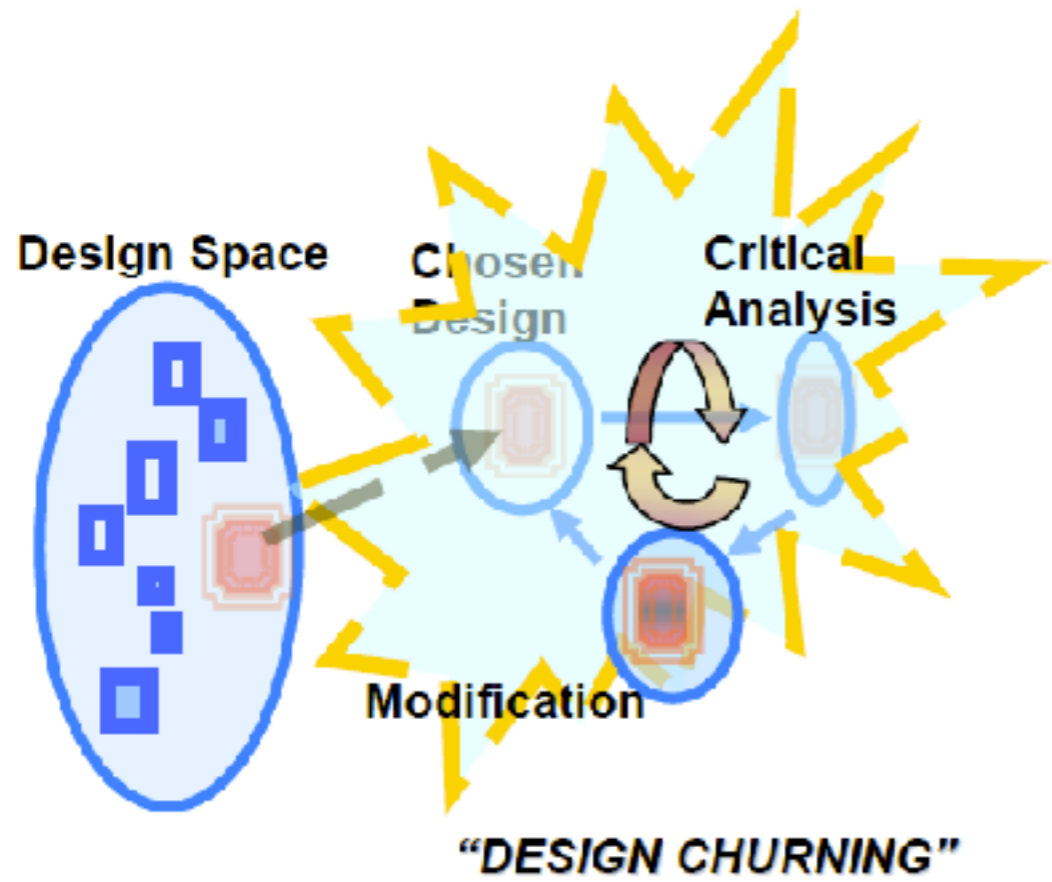
The axiomatic design process guides designers through these same steps. They can use all of their existing design tools and software and efficiently arrive at a successful new design, or diagnose and correct an existing design.

There are four main concepts in axiomatic design:

# Métodos de Design e o mercado



# Set-Based Concurrent Design



**"ELIMINATE WEAKEST ALTERNATIVES"**

## O problema da flexibilidade

O problema da flexibilidade é certamente um fator crucial na possibilidade de transferir métodos propostos pela academia para o mercado. O retrabalho e as repetidas idas e vindas entre fases, geralmente provocadas por uma abordagem “point based” são os principais fatores. O point based engineering infelizmente é muito praticado na academia e se propaga rapidamente para o mercado.

Os requisitos “single-point” são um parceiro inseparável do Point-Based Engineering, e do insucesso de tantos projetos.

Toyota's Principles of Set-Based Concurrent Engineering

sloanreview.mit.edu/the-magazine/1999-winter/4025/toyotas-principles-of-set-based-concurrent-engineering/

Essentia... Cursor... Human... Scholar... media.w... Academ... Edit Pro... Axiomat... Toyota!...

**Home**

**The Magazine**

**Big Ideas**

- Social Business
- Data & Analytics
- Sustainability
- Digital Transformation

**Blog: Improvisations**

**Archives**

**Browse by Topic**

Save 22% plus **MIT Sloan** FREE ACCESS [Click Here](#)

**IMPROVISATIONS**  
A Blog from MIT Sloan Management Review

- Crowdsourcing: Hanging Out with All the Right People
- It's All About the Platform: What Walmart and Google Have in Common
- "Want to Help Someone? Shut Up and Listen."
- The "Benefit Corporation" Movement
- Data Decisions: You Can't Be Held Responsible (Or Can You?)

[More »](#)

OPERATIONS MANAGEMENT AND RESEARCH

## Toyota's Principles of Set-Based Concurrent Engineering

By Durward K. Sobek II, Allen C. Ward and Jeffrey K. Liker  
January 15, 1999

How Toyota's product design and development process helps find the best solutions and develop successful products.

[Go premium](#) to get access to this and all other articles. [Buy](#)

SHARE THIS

0 0 0

Like Tweet +1

Share Email Buy

Toyota Motor Corporation is an industry leader in product development lead time while using fewer engineers than its U.S. competitors. It has also shown remarkable consistency in market share growth and profit per vehicle, which led to cash reserves of \$21 billion, exceeding those of the "Big Three" automakers combined.<sup>1</sup> The Toyota Production System (TPS), dubbed "lean manufacturing" has been critical in these accomplishments,<sup>2</sup> but we believe that Toyota's product design and development system is also an important contributor.<sup>3</sup>

While Taiichi Ohno and others have meticulously described the TPS, the Toyota development system has not been well documented.<sup>4</sup> Indeed, Toyota does not use many of the practices often considered critical to successful concurrent engineering and associated with Japanese manufacturers. Its development teams are not collocated. Personnel, with the exception of the chief engineer and his staff, are not dedicated to one vehicle program. Cross-functional job rotation is unusual for the first ten to twenty years of an engineer's career. Engineering and test functions rarely use quality function deployment (QFD) and Taguchi methods. Toyota excels at value engineering (VE) and value analysis (VA), yet Toyota engineers say they do not use any of the text-book tools and matrices for VE or VA. And there is nothing remarkable about Toyota's CAD or CAE systems. These practices, then, do not explain Toyota's effectiveness in developing new vehicles.

In a previous article, we called Toyota's product development system the "second Toyota paradox."<sup>5</sup> TPS was the first: its features seem wasteful but result in a more efficient overall system, such as changing over manufacturing processes more frequently (presumably inefficient) in order to create short manufacturing lead times. The second paradox can be summarized in this way: Toyota considers a broader range of possible designs and delays certain decisions longer than other automotive companies do, yet has what may be the fastest and most efficient vehicle development cycles in the industry.

**FROM THE MAGAZINE**

Fall 2012 Social Business

**The Key to Social Media Success Within Organizations**

One essential factor often overlooked: how employees feel about the organization.

[More from this issue »](#) [SUBSCRIBE](#)

**ADVANCED MANAGEMENT PROGRAM** [LEARN MORE](#)

**HARVARD BUSINESS SCHOOL**  
Executive Education

Save 22% PLUS FREE ACCESS [Click Here](#)

**THE MAGAZINE — PAST ISSUES**

- Summer 2012 Dethroning an Established Platform
- Spring 2012 Achieving Successful Strategic





# Os Comitês de Estudo para o Design de Sistemas

INCOSE

OMG

IEEE

## Uma Norma para o Design de Sistemas

A norma ISO/IEC 15.288 foi lançada em Novembro 2002, editada por Stuart Arnold e arquitetada por Harold Lawson;

Em 2004 foi adotada pelo IEEE e passou a ser uma norma ISO/IEC/IEEE;

A última revisão foi publicada em Maio de 2015.

## Uma Norma para o Design de Sistemas

Até aqui olhamos o Projeto de Sistemas do ponto de vista técnico, sempre privilegiando a formalização do processo após a fase de requisitos. Para chegar a um “projeto real” teremos que incluir a aspecto de negócios (business process) assim como os aspectos de gestão do próprio processo.

## Technical processes

Business or mission analysis process	Integration process
Stakeholder needs & requirements definition process	Verification process
System requirements definition process	Transition process
Architecture definition process	Validation process
Design definition process	Operation process
System analysis process	Maintenance process
Implementation process	Disposal process

## Technical management processes

Project planning process
Project assessment and control process
Decision management process
Risk management process
Configuration management process
Information management process
Measurement process
Quality assurance process

## Agreement processes

Acquisition process
Supply process

## Organizational project-enabling processes

Life cycle model management process
Infrastructure management process
Portfolio management process
Human resource management process
Quality management process
Knowledge management process

## Definition of Systems

real

"...are man made, created and utilized to provide products or services in defined environments for the benefit of users and other stakeholders"

model

"...an integrated set of elements, sub-systems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements."  
(INCOSE)

## *System and System of Systems*

A system-of-systems is an assemblage of components which individually may be regarded as systems, and which possess two additional properties:

1. **Operational Independence of the Components:** If the system-of-systems is disassembled into its component systems, the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.
2. **Managerial Independence of the Components:** The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems. (Maier 1998, 271)

Maier, M. W. 1998. "Architecting Principles for Systems-of-Systems". *Systems Engineering*, 1(4): 267-84.

# Classification of SoS: US Dept. of Defense

According to US DoD systems of systems could be classified into:

Virtual  
Collaborative  
Acknowledged  
Directed

DUS(AT). 2008. Systems Engineering Guide for Systems of Systems," version 1.0. Washington, DC, USA: Deputy Under Secretary of Defense for Acquisition and Technology (DUS(AT))/U.S. Department of Defense (DoD).

## *Other System Architectures*

### Federation of Systems

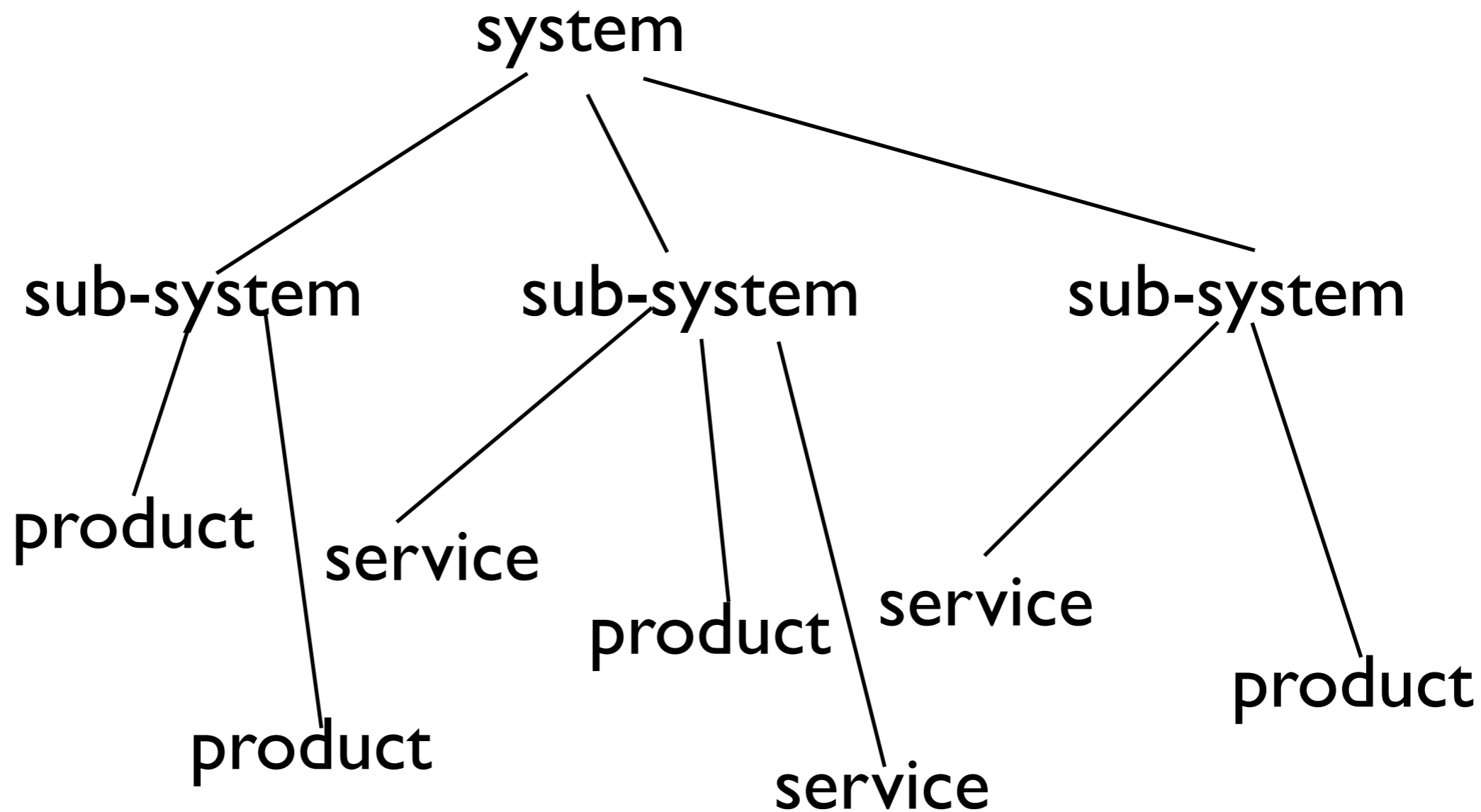
A Federation of Systems (FoS) is a loosely coupled set of collaborative and distinct institutions (systems) with a weak structuring control that “voluntarily” contributes to some social goal. This set could be closed or open.

### Family of Systems

A family of systems is set of systems that have some common characteristics and also a loosely coupled association but that could collaborate to achieve some social goals and share facilities.

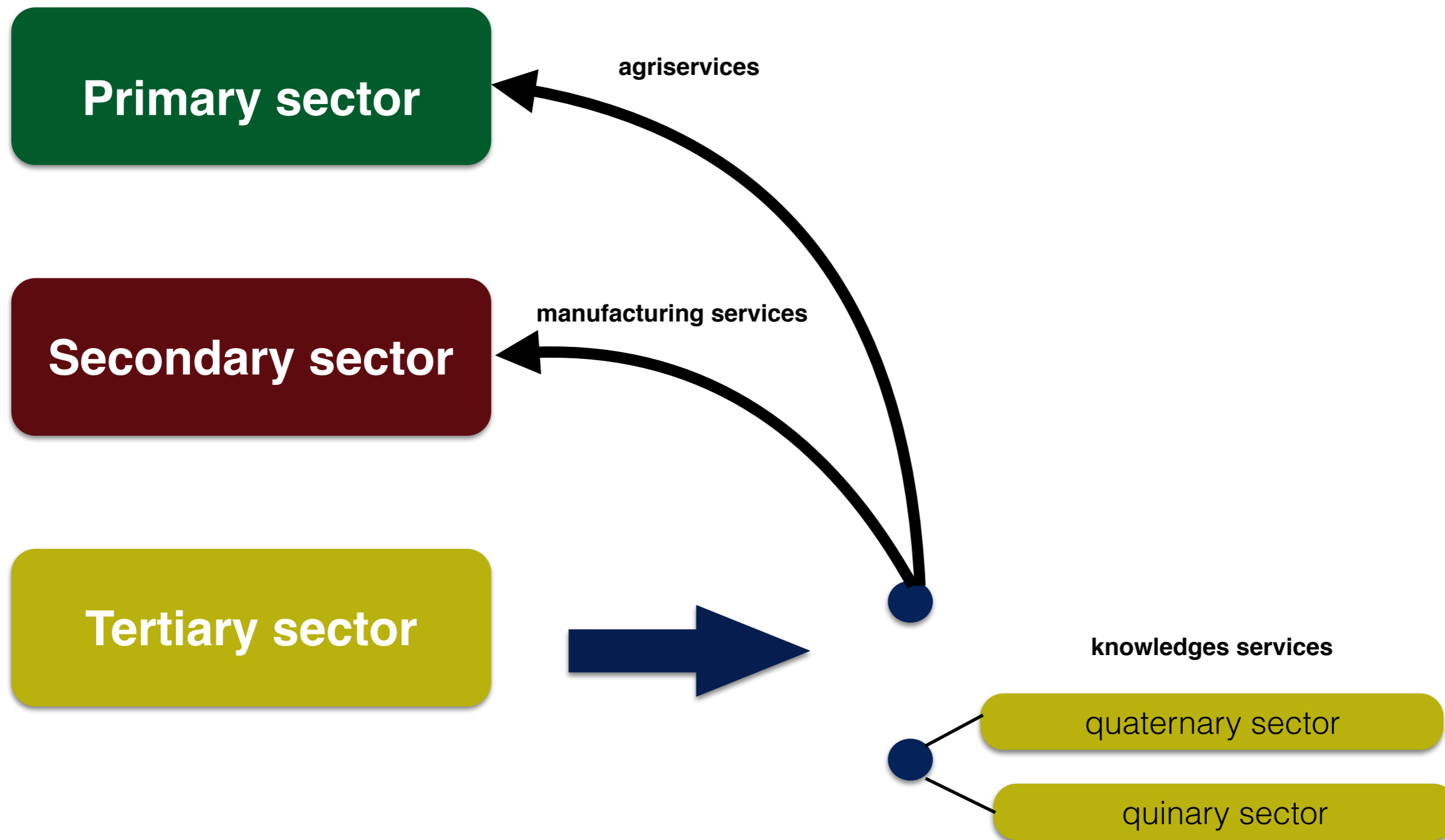


# Current tendency



# New tendencies in production





# Tendencies towards manufacturing services

## Amazon Project

Ford Motors Suppliers

resp.: Luc de Ferran

4 levels in supply chain:

"Systemistics"

Suppliers of components

Sporadic suppliers

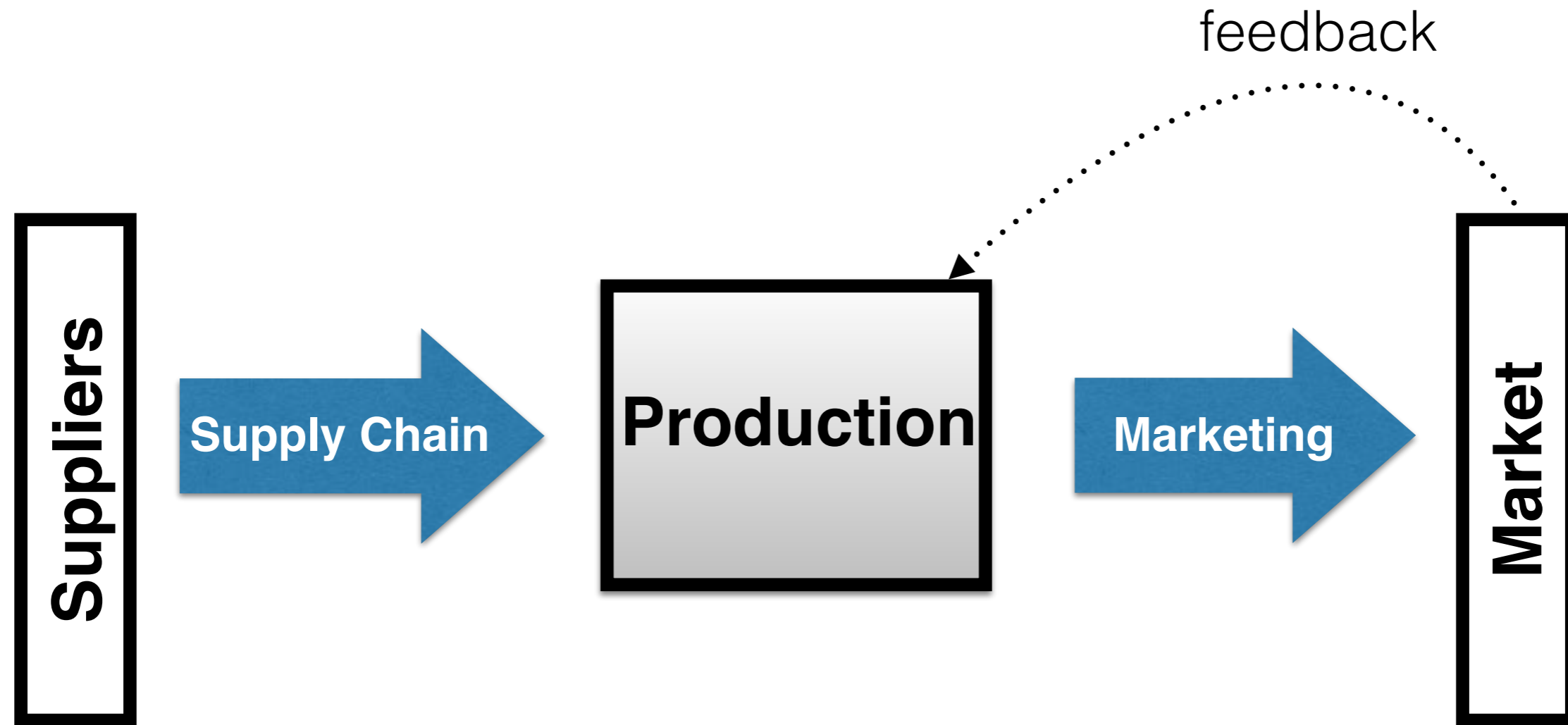
Accessories

**supply chain**

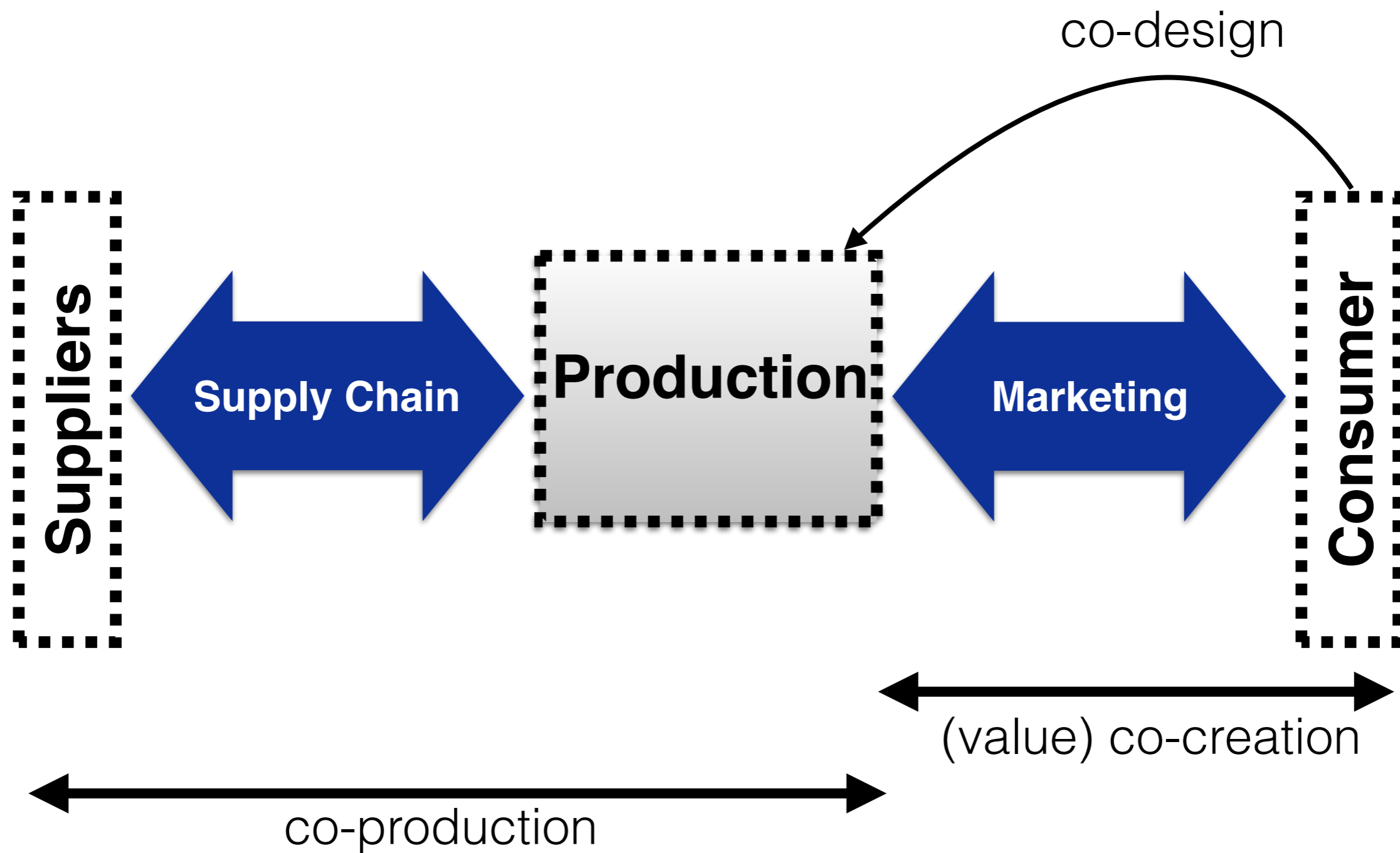


**production**

# Abstract Context of manufacturing

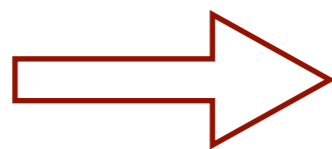


# Converging to service and intelligent systems



# Introducing manufacturing services

Even if manufacturing will be still devoted to products (good-oriented) we claim that the general production system could be based on services maintained by independent players with which we should co-create this product.



Strong focus on the early phases  
Model driven  
Strongly focused on viewpoints

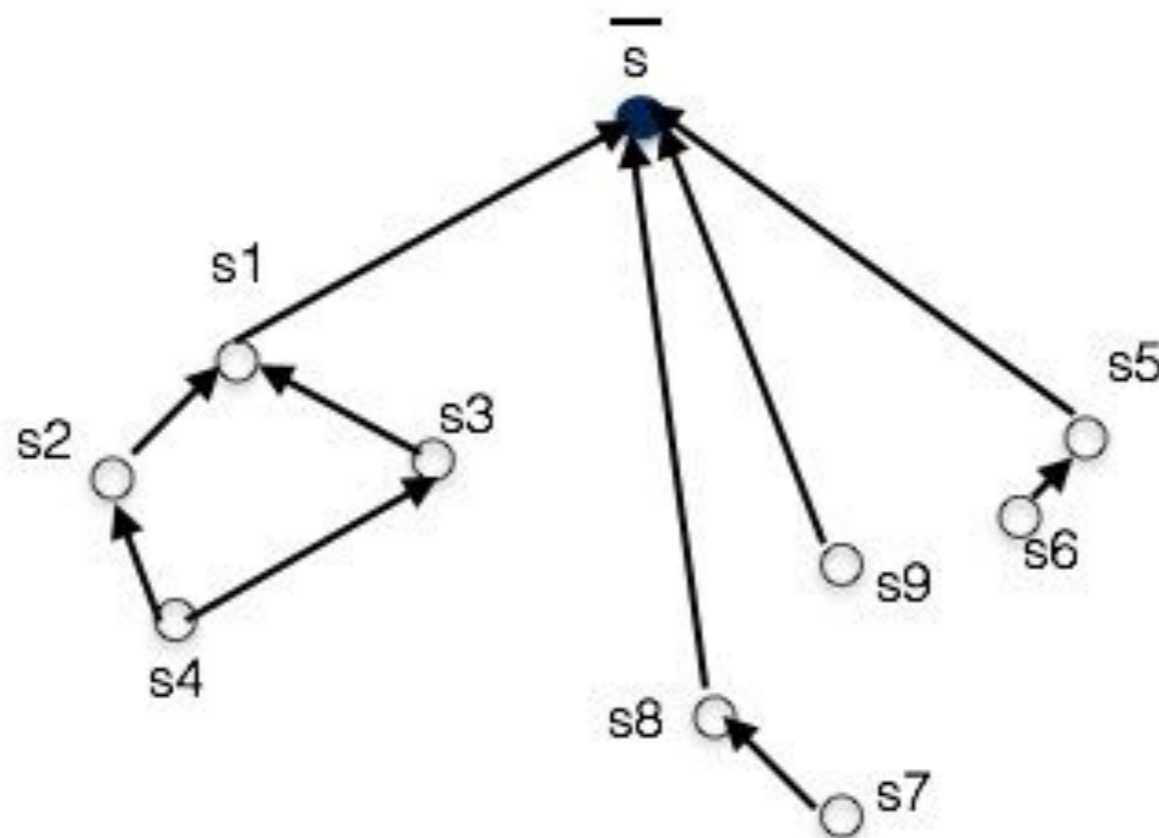
## PSA: Introducing a Service Manufacturing Architecture

Let  $S$  be a finite set of *system agents* as describe above. A manufacturing service  $s$  can be defined as  $(S, R)$ , where  $R$  is a set of relations among the service agents in. The relations could be the delivering of goods, service or a hybrid artifact composed of product and service.

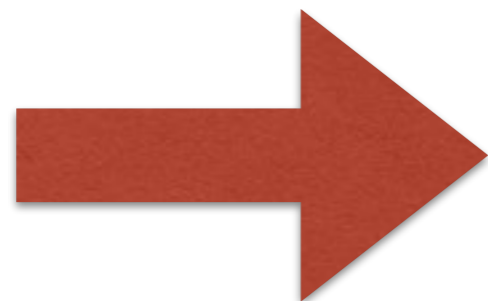
A PSA architecture is a connected arrangement of collaborative service agents that could be represented as a graph, where each service agent is a node and the possible delivers are represented by arcs. There is a special node denoted by  $s_0$  which represents the service agent responsible for the main business strategy and also to deliver the whole service to the costumer.



# PSA Example: From Factory to Service



A grande pergunta que permeia a aula de hoje é justamente como vamos resolver o problema de escalabilidade entre os métodos que vimos até aqui (e sem o que a discussão seria inviável) e o problema que temos que enfrentar de projetar grandes sistemas ou sistemas de sistemas.



Model Driven Engineering

## Milestone:

Agora o seu artigo final deve atingir uma fase de maturidade e o próximo milestone requer um artigo praticamente completo onde a aplicação de técnicas de modelagem e design já possa ser avaliada. Faremos a simulação de um processo de “submissão”, e agora os comentários serão aqueles típicos de um revisor.



Obrigado

*Reinaldo*