

ENGENHARIA DE SOFTWARE

INTRODUÇÃO

Luís Morgado

2016

ENGENHARIA DE SOFTWARE

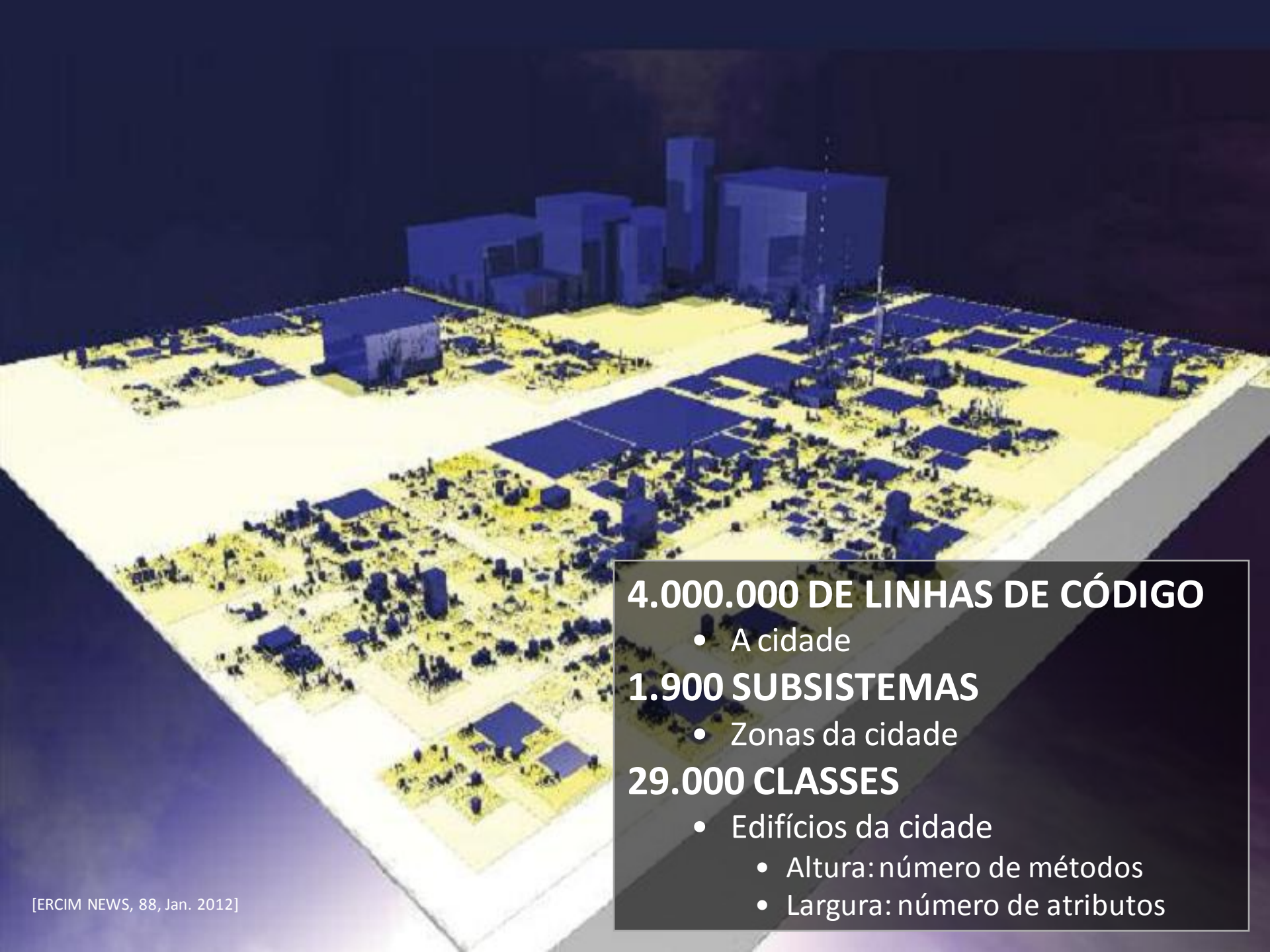
- Desenvolvimento de Software como uma **actividade de engenharia**
 - Sistemático
 - Quantificável
- Desenvolvimento, operação e manutenção de **software**
 - Complexidade
 - Mudança
 - Dependência crítica



```
AgenteReact1.java x
18     }
19
20     @Override
21     public void percepcionar()
22     {
23         imagem = getSensorImagem().detectar();
24     }
25
26     @Override
27     public void processar()
28     {
29         // Detectar estímulos
30         boolean alvo = (imagem[1] == Ambiente.IMG_ALVO);
31         boolean obst = (imagem[1] == Ambiente.IMG_OBST);
32
33         // Regras: Estímulo => Resposta
34         // Prioridade decrescente por ordem de declaração
35
36         if(alvo) {
37             // Recolher alvo
38             accao = new Accao(0, 1, Accao.Tipo.PEGAR);
39             return;
40         }
41
42         if(obst) {
43             // Evitar obstáculo (Rodar)
44             accao = new Accao(0.125, 0, Accao.Tipo.MOVER);
45             return;
46         }
47     }
```

Outline x

```
agentereact1
├── import declarations
└── AgenteReact1
    ├── imagem : byte[]
    ├── accao : Accao
    ├── iniciar() : void
    ├── percepcionar() : void
    ├── processar() : void
    └── agir() : void
```



4.000.000 DE LINHAS DE CÓDIGO

- A cidade

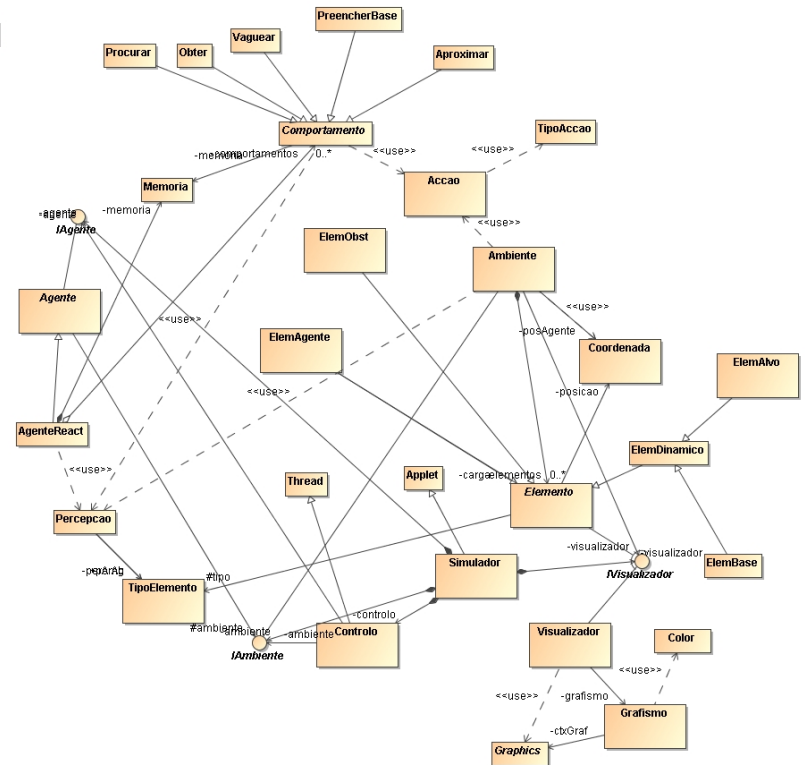
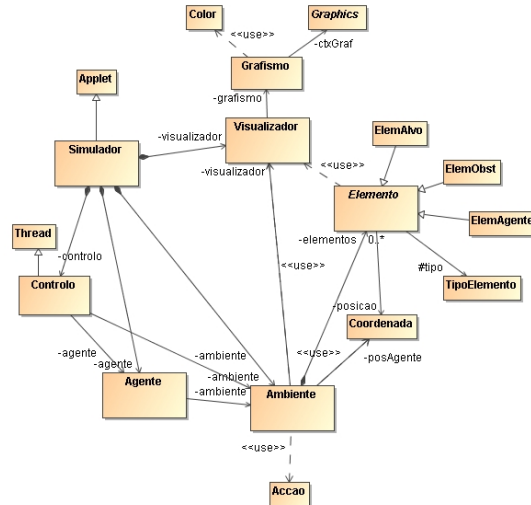
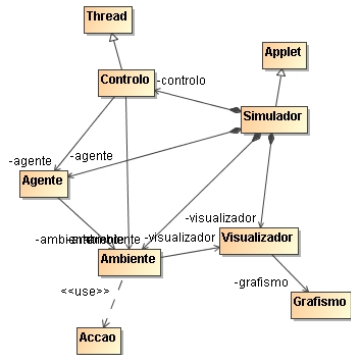
1.900 SUBSISTEMAS

- Zonas da cidade

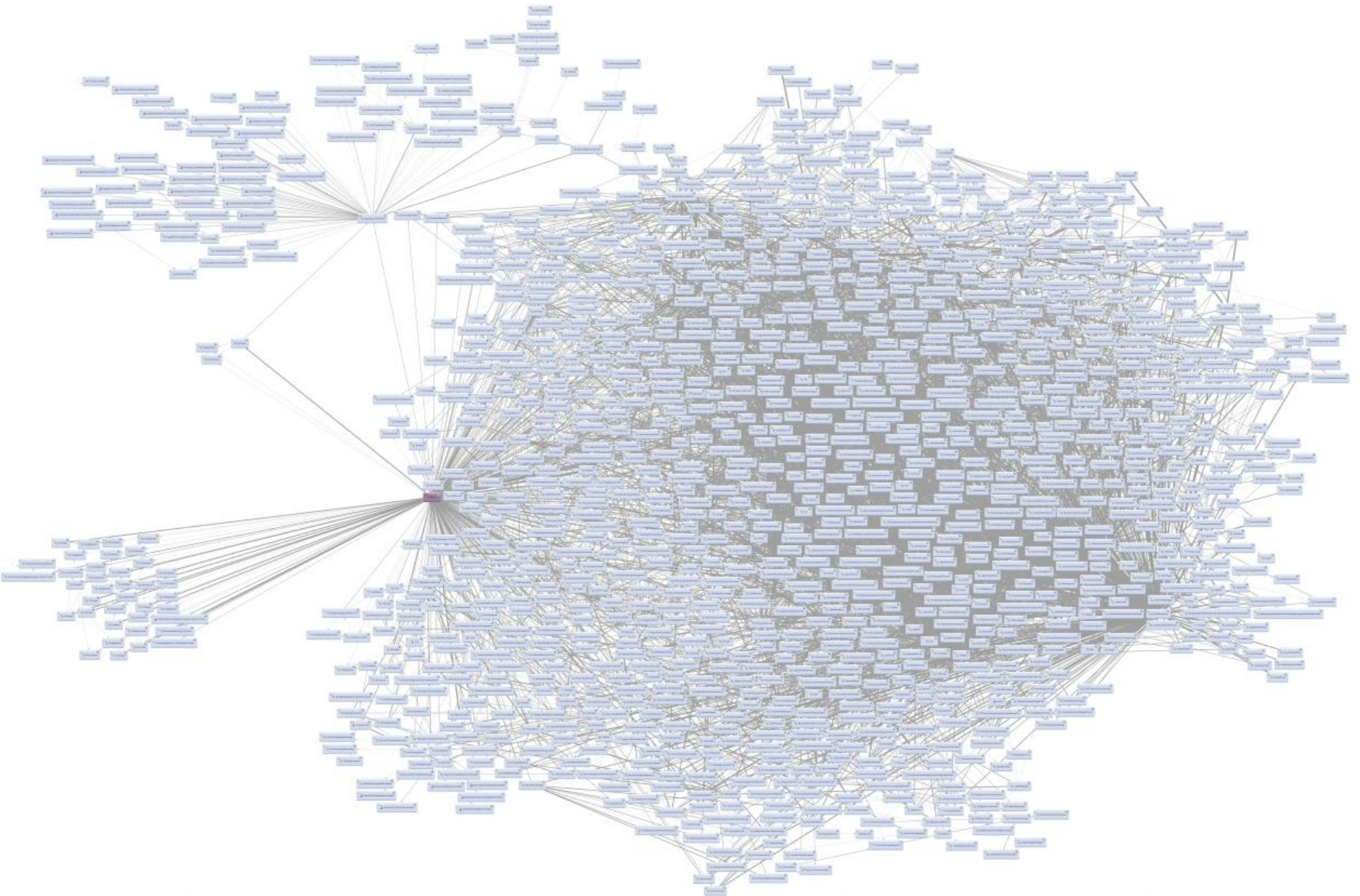
29.000 CLASSES

- Edifícios da cidade
 - Altura: número de métodos
 - Largura: número de atributos

SOFTWARE E COMPLEXIDADE



SOFTWARE E COMPLEXIDADE



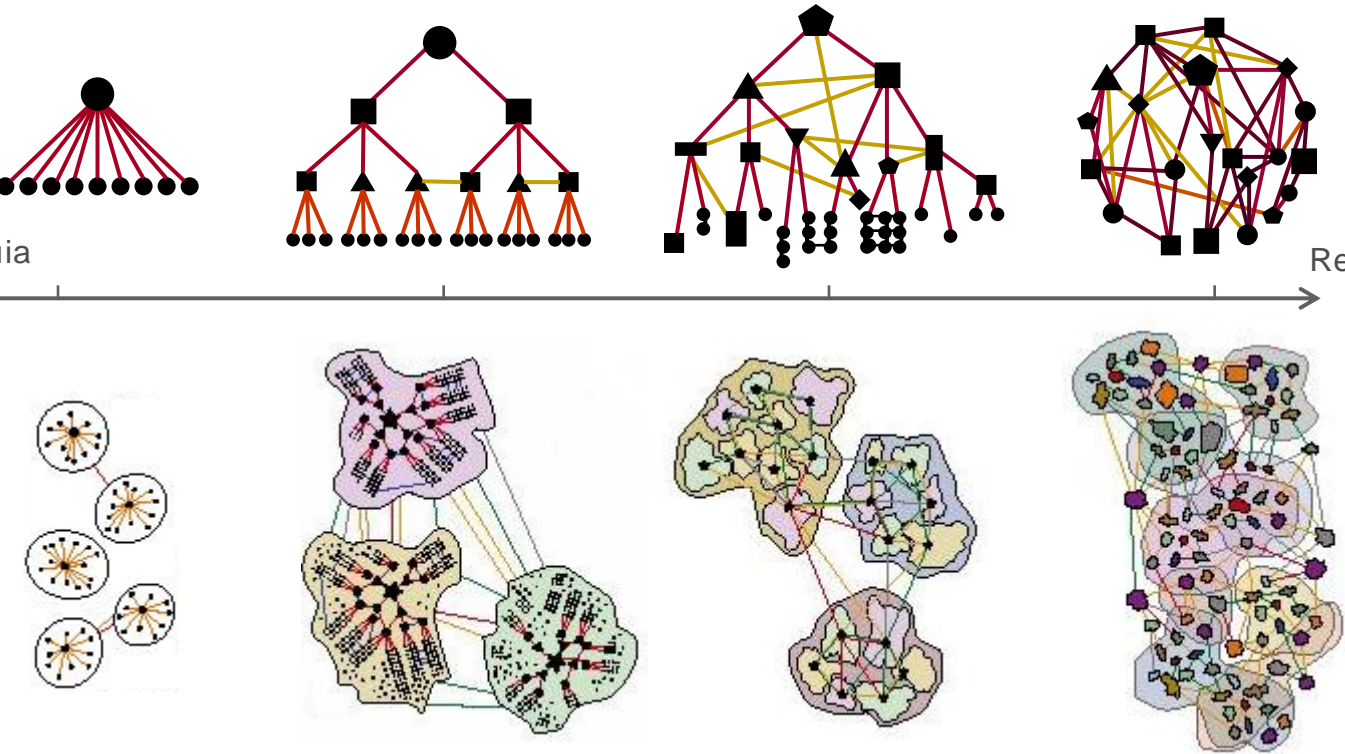
COMPLEXIDADE DE SOFTWARE

Tipo de Organização

Hierarquia

Rede

Topologia

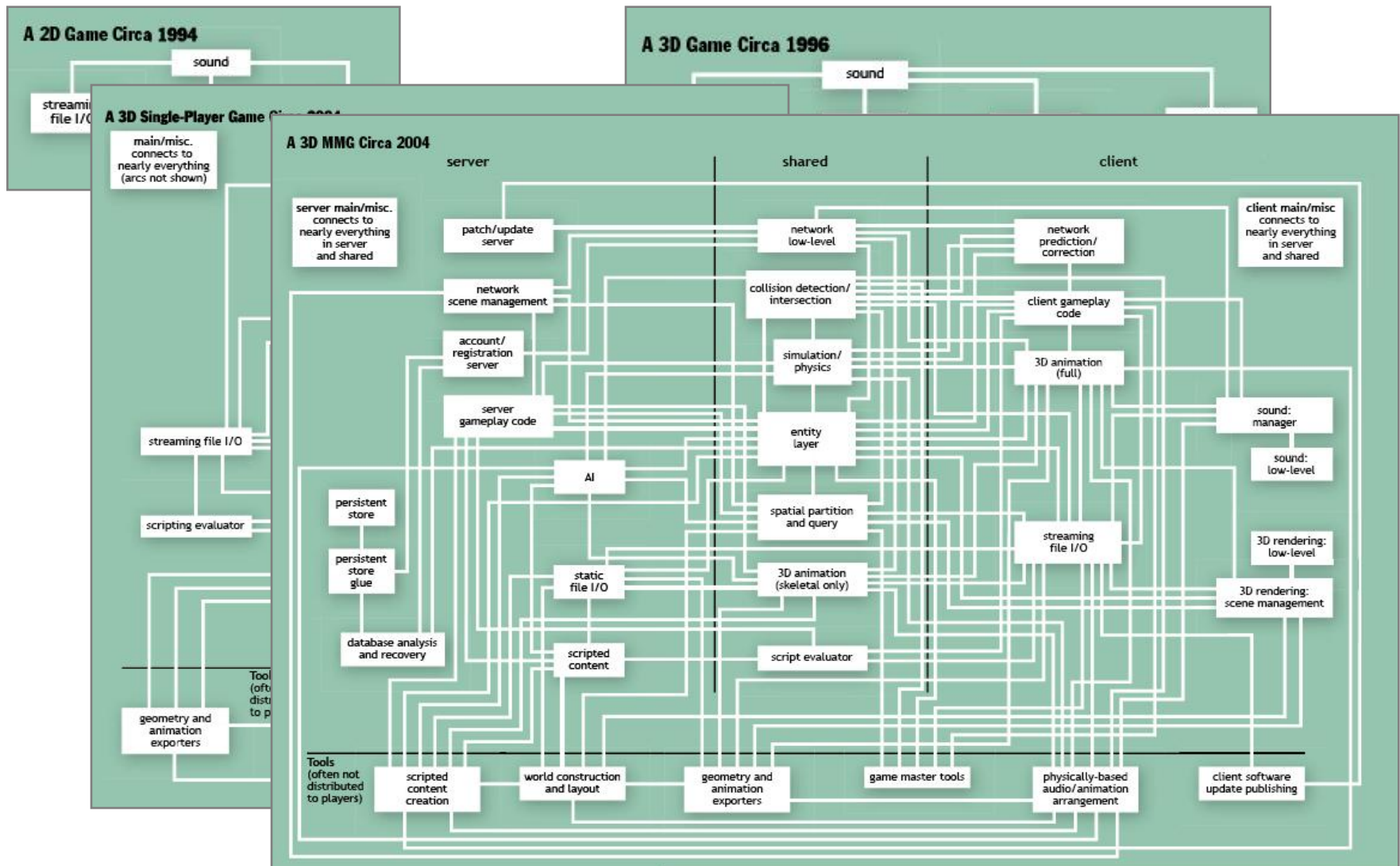


↑ **Complexidade**

Especialização

Diversidade

EXEMPLO: ARQUITECTURAS DE JOGOS



EVOLUÇÃO EXPONENCIAL



IBM Real-Time Computer Complex - NASA Manned Spacecraft Center
Década de 1960



Hoje

EXEMPLO: INDÚSTRIA AUTOMÓVEL

SPECTRUM

This Car Runs on Code

By Robert N. Charette

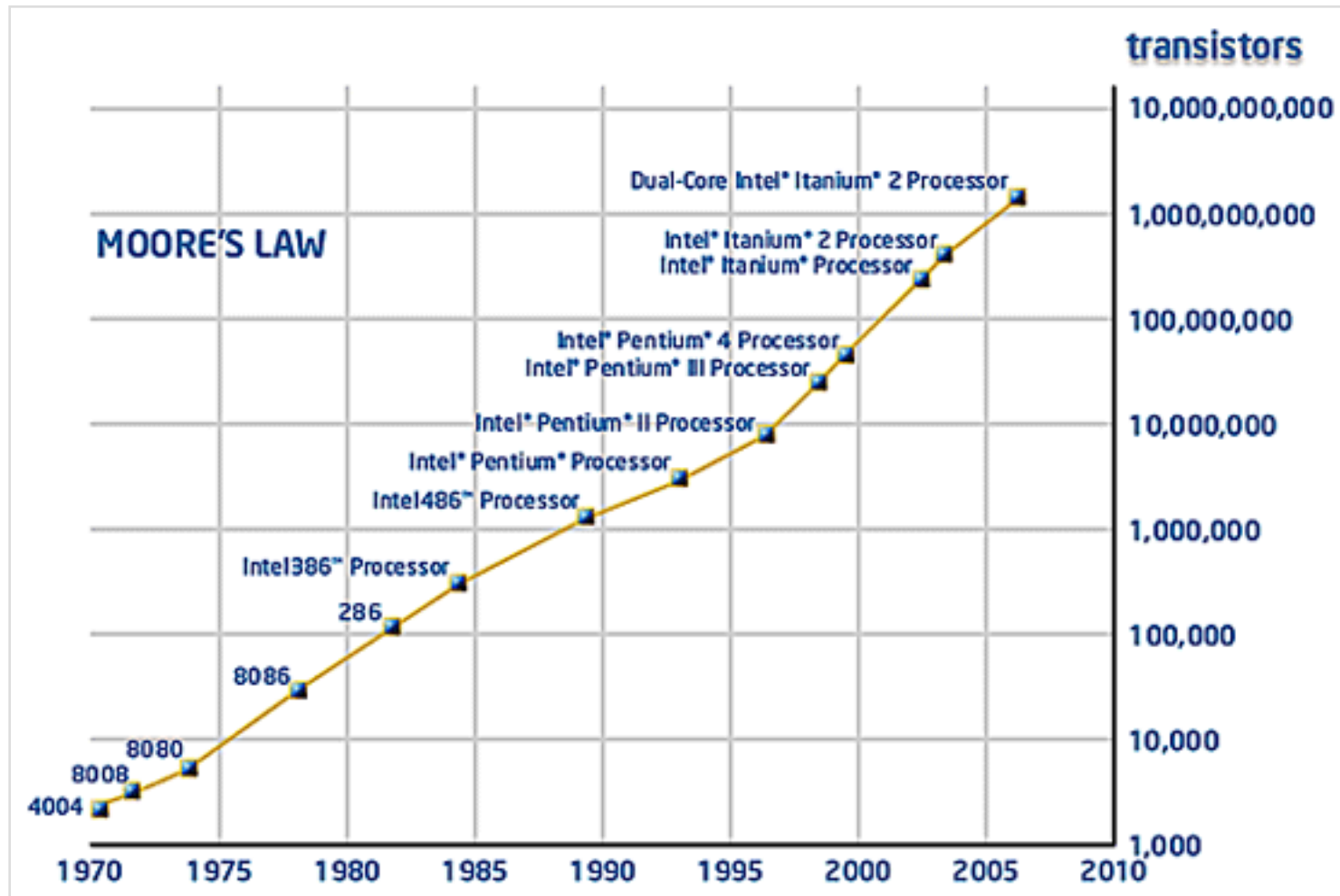
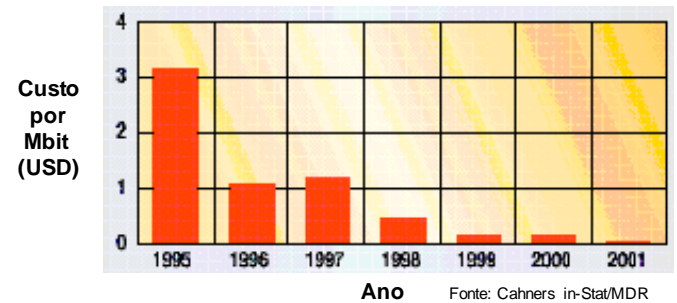


IMAGE: DAIMLER

The avionics system in the F-22 Raptor, the current U.S. Air Force frontline jet fighter, consists of about 1.7 million lines of software code. The F-35 Joint Strike Fighter, scheduled to become operational in 2010, will require about 5.7 million lines of code to operate its onboard systems. And Boeing's new 787 Dreamliner, scheduled to be delivered to customers in 2010, requires about 6.5 million lines of software code to operate its avionics and onboard support systems.

These are impressive amounts of software, yet if you bought a premium-class automobile recently, "it probably contains close to 100 million lines of software code," says Manfred Broy, a professor of informatics at Technical University, Munich, and a leading expert on software in cars. All that software executes on 70 to 100 microprocessor-based electronic control units (ECUs) networked throughout the body of your car.

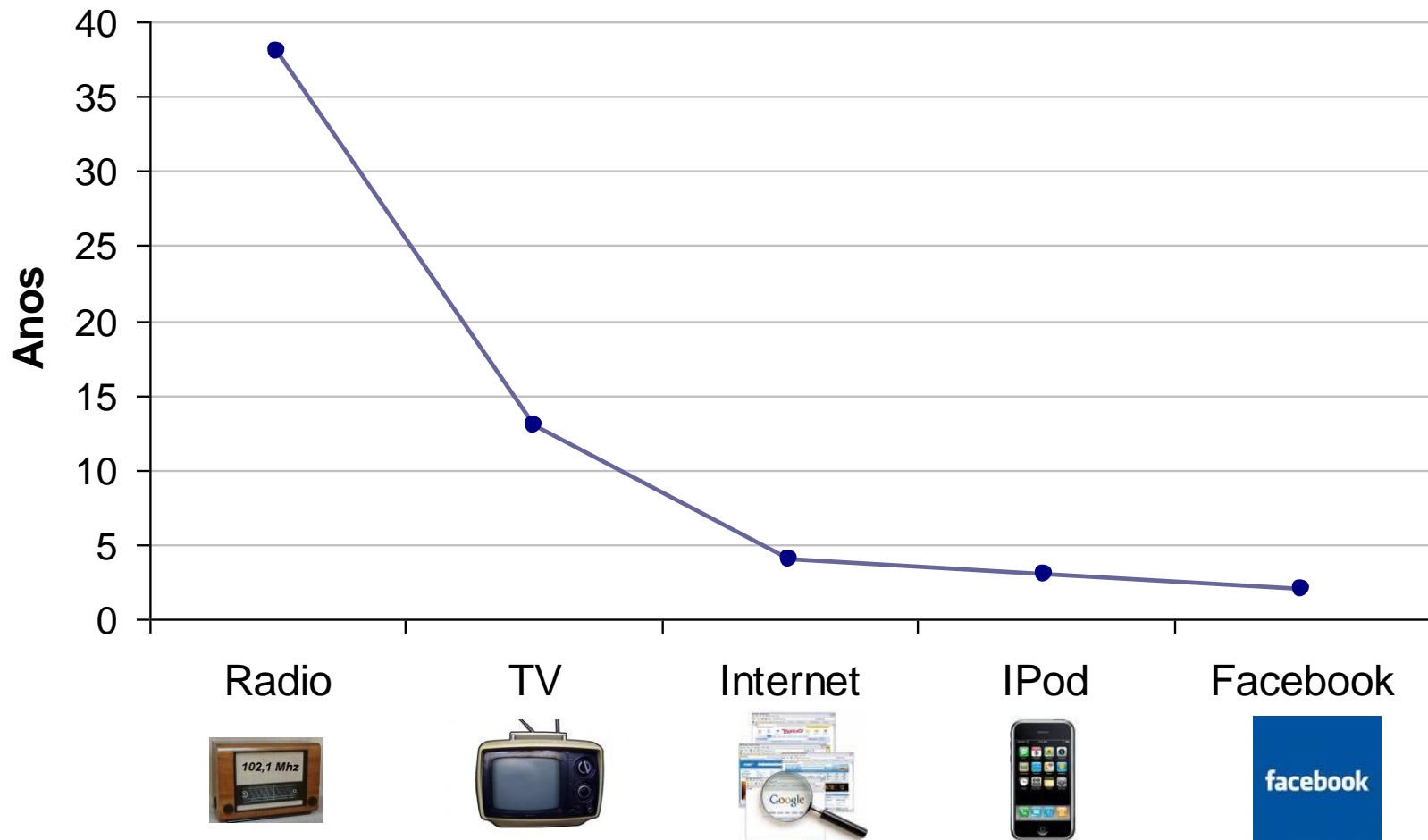
LEI DE MOORE



[Intel, 2007]

MUDANÇA RÁPIDA

Tempo de difusão da tecnologia
(50 milhões de utilizadores)



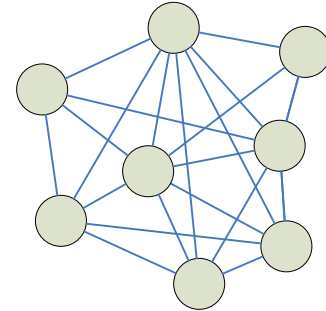
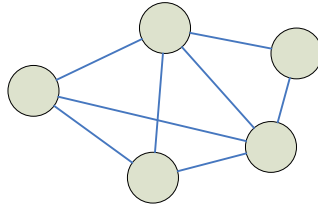
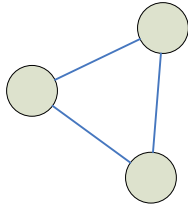
COMPLEXIDADE

Grau de **difículdade de previsão** das propriedades de um sistema dadas as propriedades das partes individuais [Weaver, 1948]

- Relacionada com a **informação** que é necessária para a caracterização de um sistema
- Um sistema é tanto mais **complexo** quanto mais **informação** for necessária para a sua **descrição**
- Reflete-se no **esforço** necessário para geração da **organização (ordem)** do sistema

O PROBLEMA DA COMPLEXIDADE

COMPLEXIDADE ESTRUTURAL

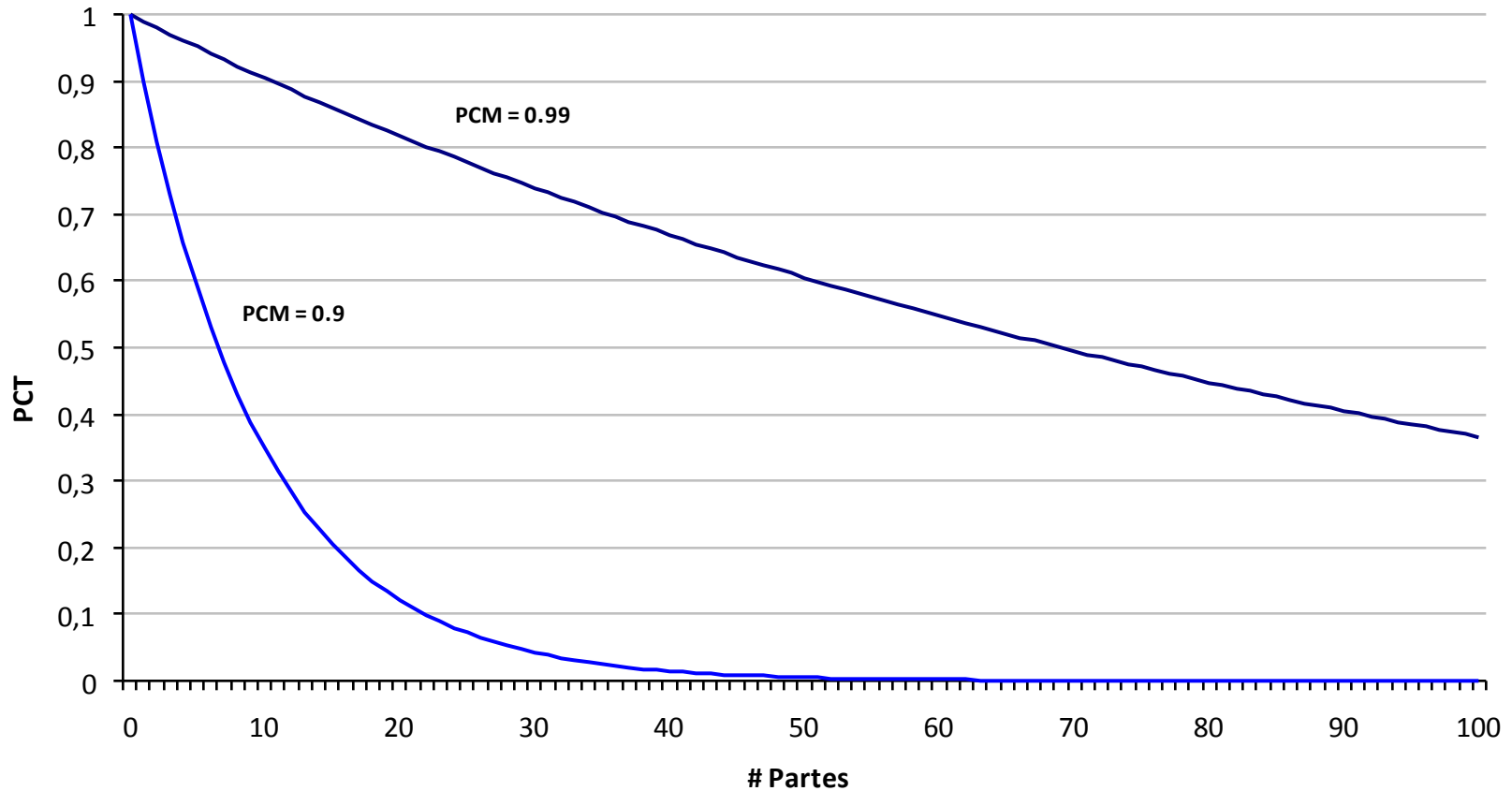


- **UM PROBLEMA DE INTERACÇÃO**
 - De partes do sistema
 - De elementos de informação
 - De elementos das equipas de desenvolvimento
- Um sistema com duas vezes mais partes é muito mais do que duas vezes mais complexo

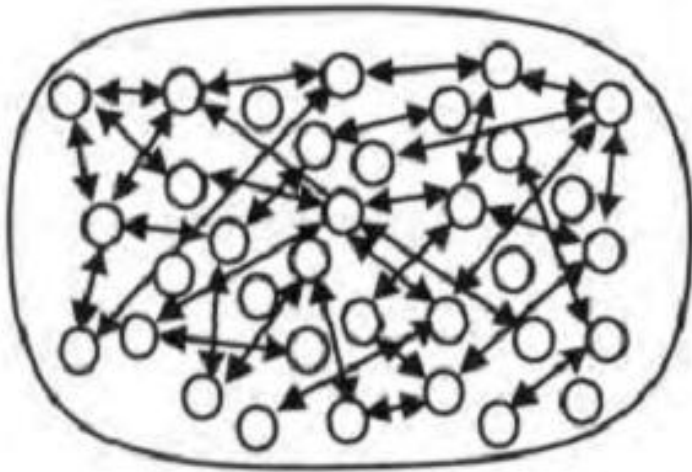
EXPLOSÃO COMBINATÓRIA

O PROBLEMA DA COMPLEXIDADE

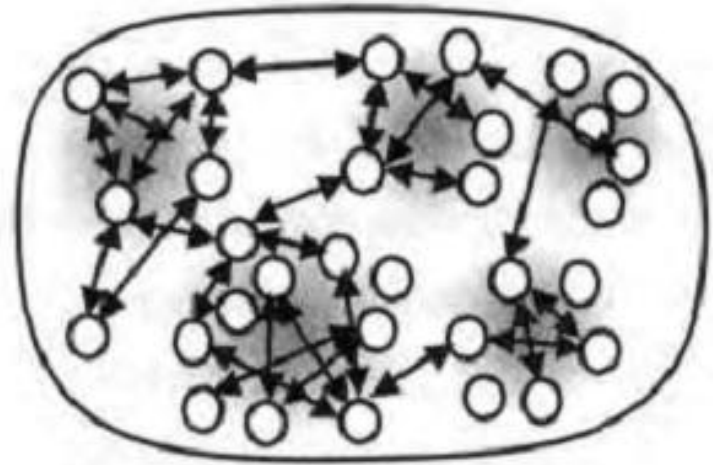
Variação da probabilidade de correcção total em função do número de partes



COMPLEXIDADE E ORGANIZAÇÃO



AN UNSTRUCTURED SYSTEM



AN “ORGANIZED” SYSTEM

TIPOS DE COMPLEXIDADE

COMPLEXIDADE DESORGANIZADA

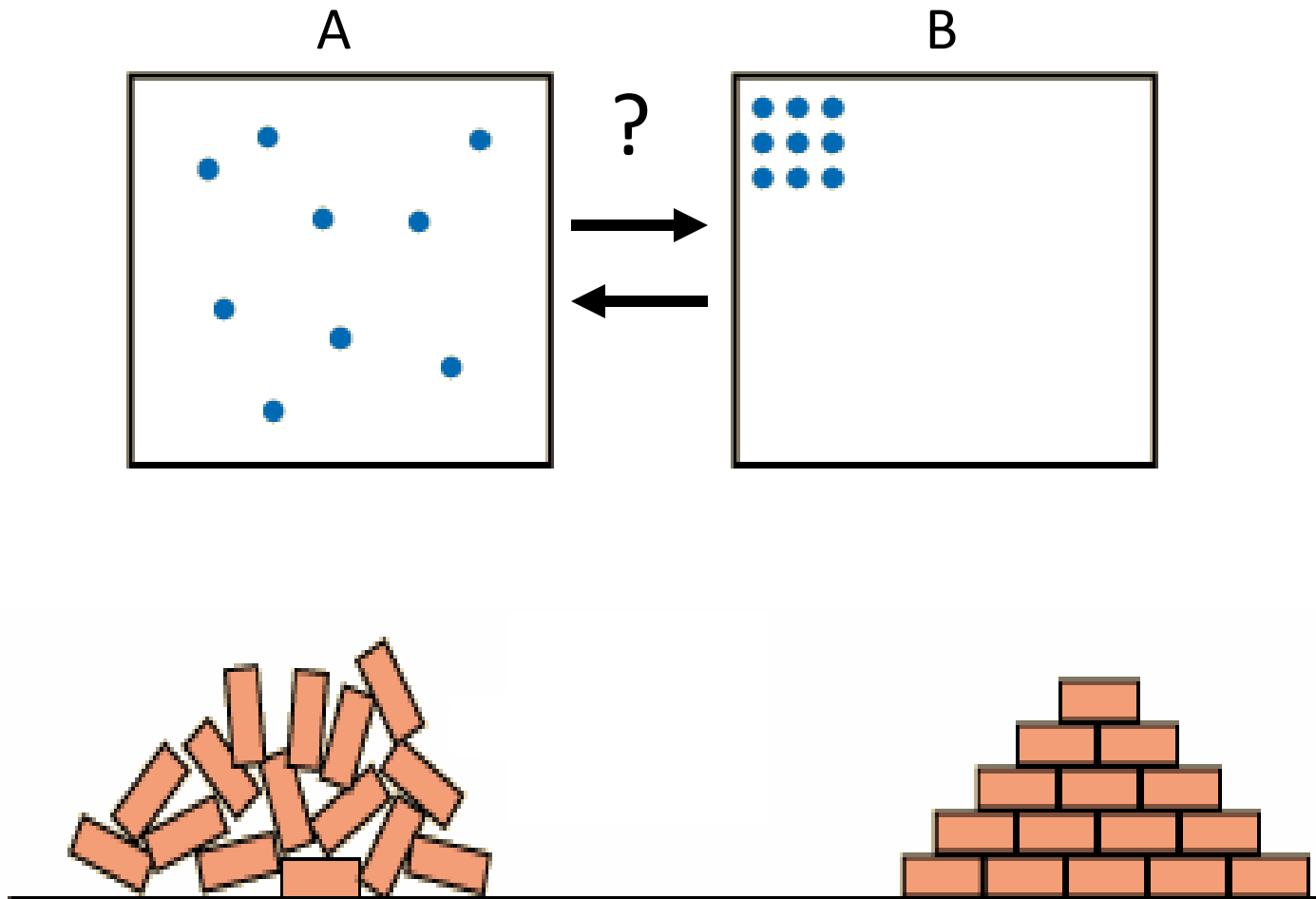
- resulta do **número e heterogeneidade** das partes de um sistema
- as partes podem interactuar entre si, mas a **interacção é irregular**
- as características globais do sistema podem ser inferidas com base em **métodos estatísticos**

COMPLEXIDADE ORGANIZADA

- resulta dos padrões de **inter-relacionamento** entre as partes
- as interacções entre partes obedecem a **padrões correlacionáveis** no espaço e no tempo
- **ORDEM, ORGANIZAÇÃO**
 - Com um propósito (finalidade)

COMPLEXIDADE E ENTROPIA

DINÂMICA DE EVOLUÇÃO



COMPLEXIDADE E ENTROPIA

- ENTROPIA

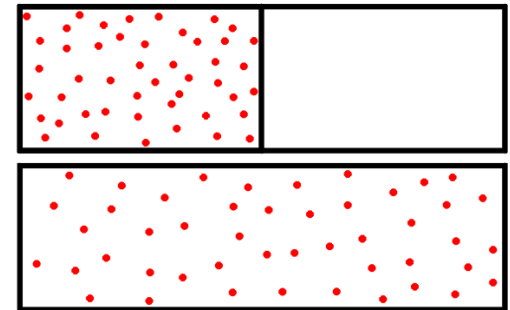
- Medida do grau de dispersão relativa que existe num sistema fechado num dado instante de tempo
- Diferentes formalizações consoante o contexto de aplicação (e.g. Termodinâmica, Teoria da Informação)
- Formalização proposta por Boltzmann:

$$S = k \log(W)$$

k – Constante de Boltzmann

W – Multiplicidade dos graus de liberdade de um sistema

MULTIPLICIDADE



COMPLEXIDADE E ENTROPIA

- **SISTEMA**

- Agregado de partes relacionadas entre si com uma função específica

- **ESTADO (MACRO)**

- Configuração global (macro) resultante das **partes** e **relações** entre partes de um sistema

- **MICRO-ESTADO**

- Configuração específica (micro) das partes de um sistema

- **MULTIPLICIDADE** de um estado

- Número de micro-estados admissíveis do estado

MULTIPLICIDADE E ENTROPIA


- Exemplo
 - Conjunto de dados



MULTIPLICIDADE E ENTROPIA

ESTADO (macro)	MICRO-ESTADOS (2 dados)
2	1+1
3	1+2, 2+1
4	1+3, 2+2, 3+1
5	1+4, 2+3, 3+2, 4+1
6	1+5, 2+4, 3+3, 4+2, 5+1
7	1+6, 2+5, 3+4, 4+3, 5+2, 6+1
8	2+6, 3+5, 4+4, 5+3, 6+2
9	3+6, 4+5, 5+4, 6+3
10	4+6, 5+5, 6+4
11	5+6, 6+5
12	6+6

MULTIPLICIDADE E ENTROPIA

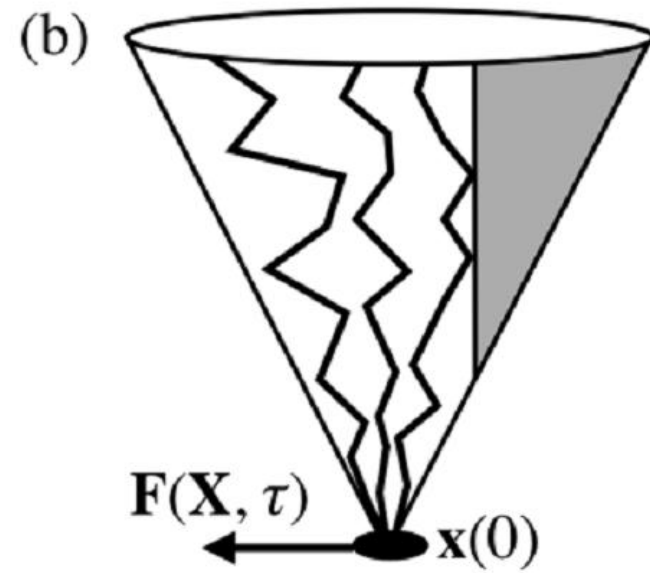
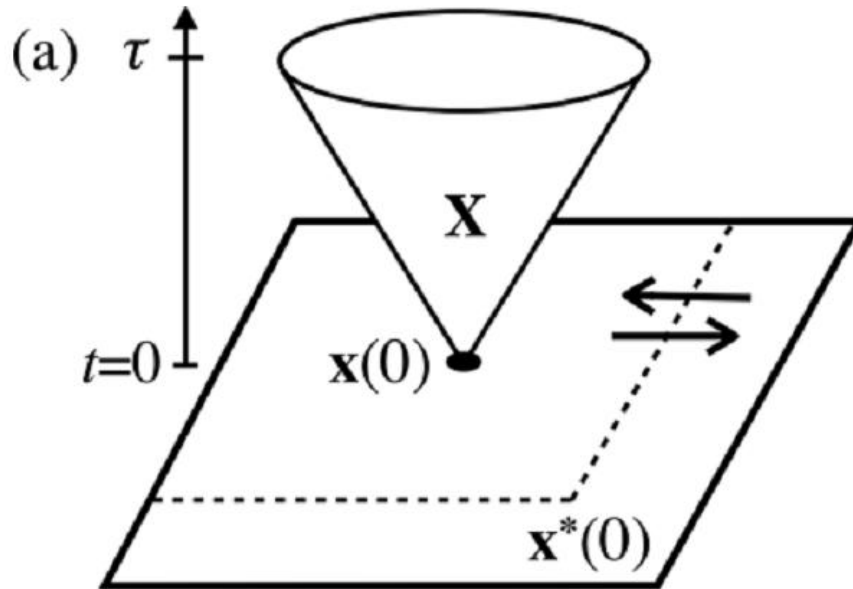
ESTADO (macro) s	MULTIPLICIDADE W	Prob. Ocorrência $P(s)$
2	1	2,8%
3	2	5,6%
4	3	8,3%
5	4	11,1%
6	5	13,9%
7	6	16,7% 
8	5	13,9%
9	4	11,1%
10	3	8,3%
11	2	5,6%
12	1	2,8%

ENTROPIA

Medida da multiplicidade dos estados de um sistema, indicando a tendência para o sistema assumir estados de máxima multiplicidade

MULTIPLICIDADE E ENTROPIA

FORÇAS CAUSAIS ENTRÓPICAS



2ª LEI DA TERMODINÂMICA

- Num processo espontâneo a alteração total de entropia num sistema e no seu ambiente é positiva:

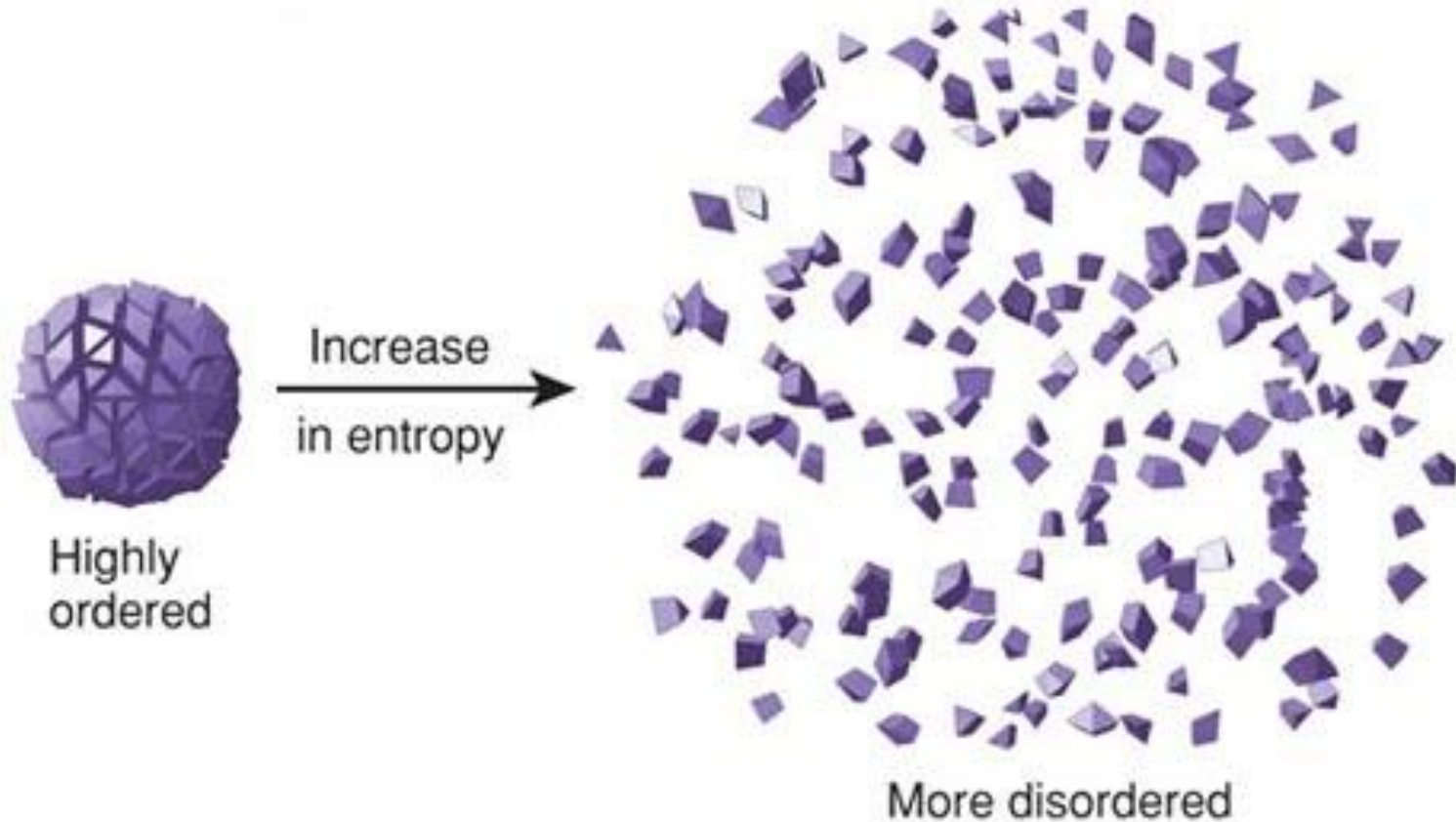
$$\Delta S_{\text{universo}} = \Delta S_{\text{sistema}} + \Delta S_{\text{ambiente}} \geq 0$$

- Num sistema fechado a entropia apenas pode manter-se ou aumentar
- A **acção** de um sistema é inerentemente **geradora de entropia**

COMPLEXIDADE E ENTROPIA



COMPLEXIDADE E ENTROPIA

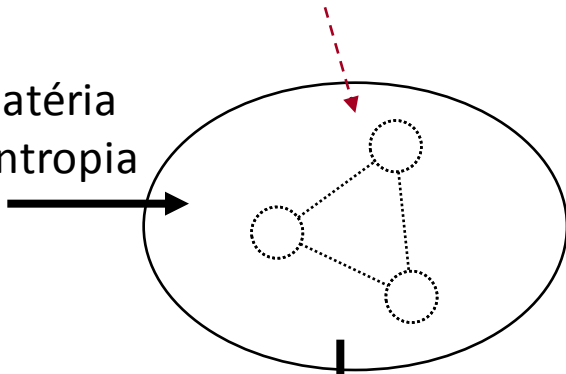


COMPLEXIDADE E ENTROPIA

SISTEMA FÍSICO

Formação de ordem interior
(matriz organizadora / auto-organização)

Fluxo de
energia/matéria
de baixa entropia

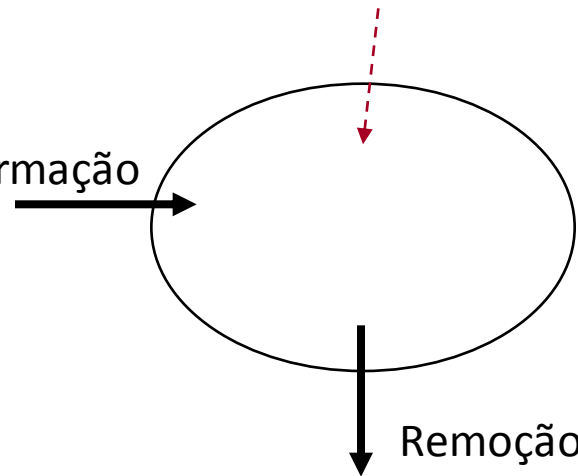


Fluxo de
energia/matéria
de **alta entropia**
(**resíduos**)

SISTEMA LÓGICO

Organização, estrutura,
conhecimento

Informação

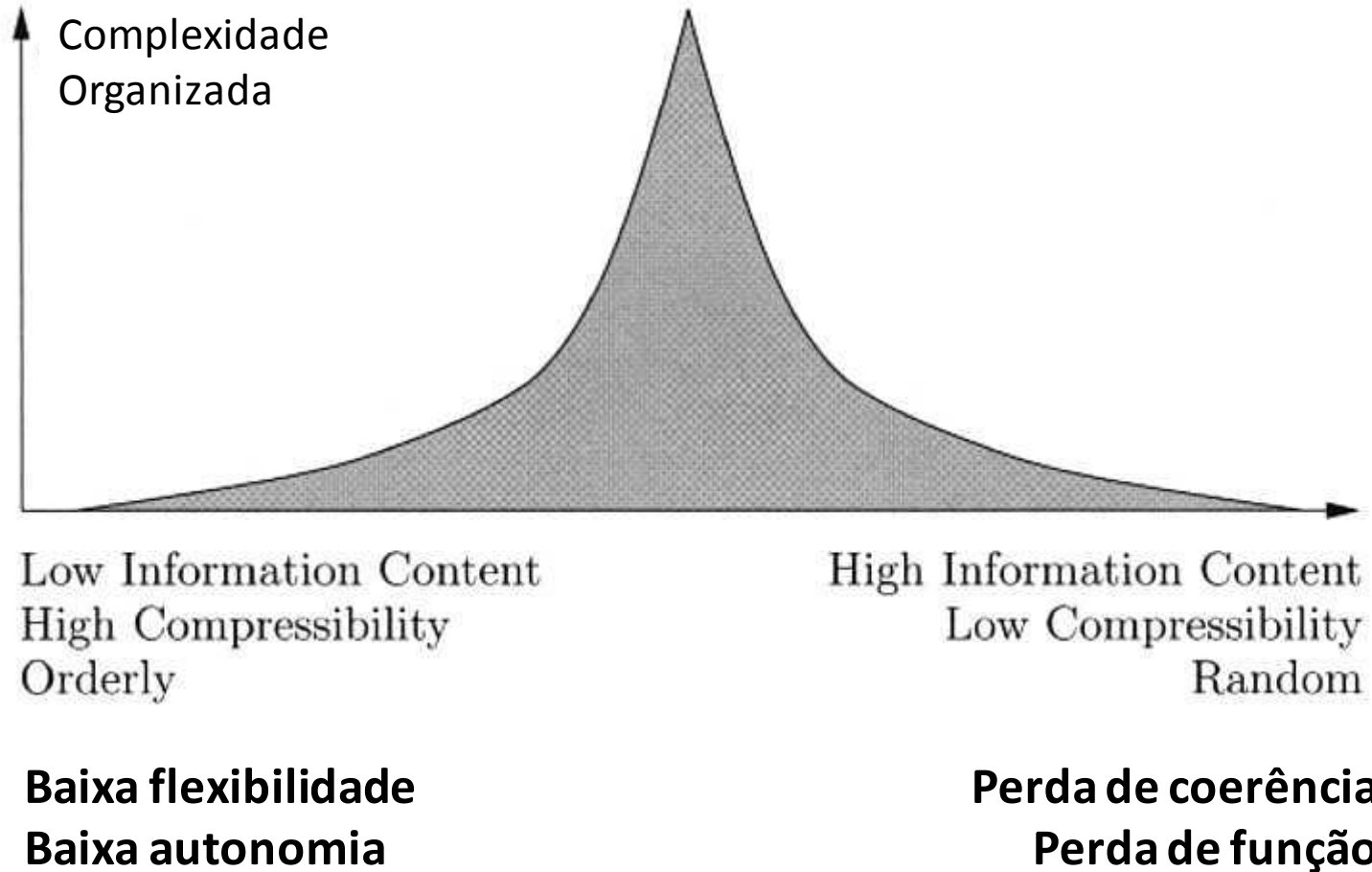


Remoção de
defeitos, ...

ESFORÇO

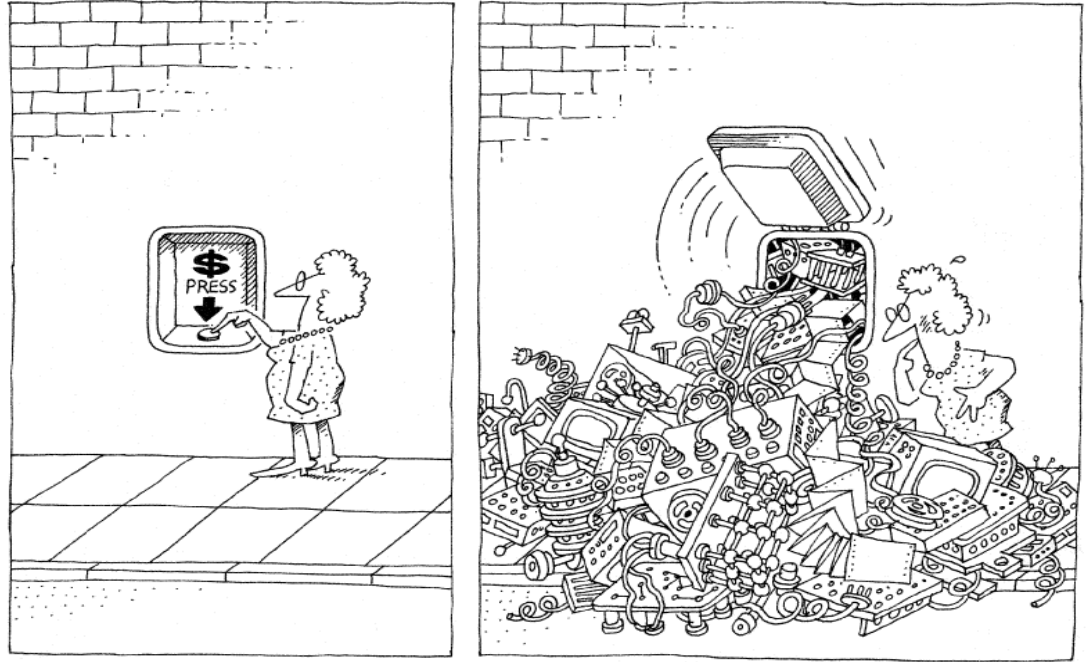
- Desenvolvimento
- Manutenção
- Evolução

COMPLEXIDADE E ENTROPIA



ARQUITECTURA DE SOFTWARE

- MÉTRICAS
- PRINCÍPIOS
- PADRÕES



[Booch, 2004]

COMPLEXIDADE

- Redução
- Controlo

MÉTRICAS DE ARQUITECTURA

- **COESÃO**

- Nível coerência funcional de um subsistema/módulo (até que ponto esse módulo realiza uma única função)

- **ACOPLAMENTO**

- Grau de interdependência entre subsistemas

- **SIMPLICIDADE**

- Nível de facilidade de compreensão/comunicação da arquitectura

- **ADAPTABILIDADE**

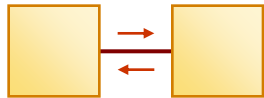
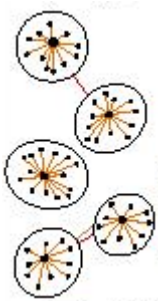
- Nível de facilidade de alteração da arquitectura para incorporação de novos requisitos ou de alterações nos requisitos previamente definidos

ACOPLAMENTO

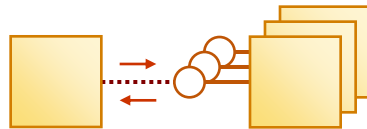
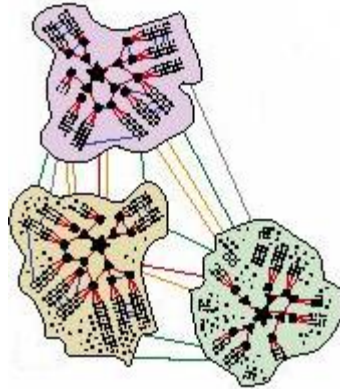
REDUÇÃO DO NÍVEL DE ACOPLAMENTO

- **Maior facilidade de desenvolvimento, instalação, manutenção e expansão**
- **Melhor escalabilidade**, devido à possibilidade de distribuição e replicação de módulos que prestem serviços, sem que isso tenha um impacto significativo nos clientes desses subsistemas/módulos
- **Maior tolerância a falhas**, logo maior robustez, uma vez que a falha de um subsistema/módulo tem um impacto restrito

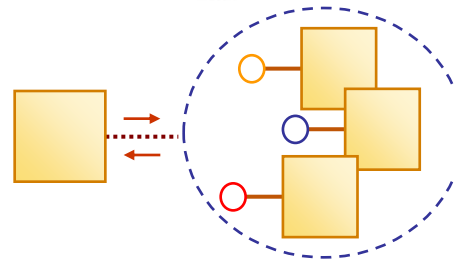
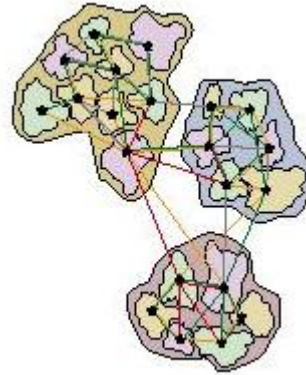
TIPOS DE ACOPLAMENTO



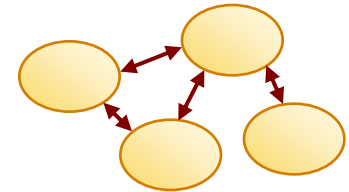
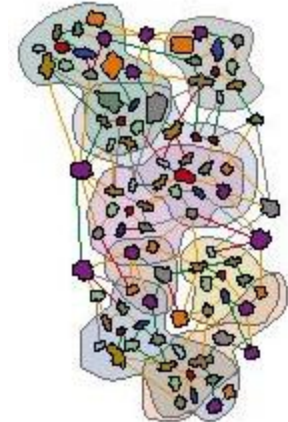
Acoplamento de
natureza estrutural



Acoplamento de
natureza funcional



Acoplamento de
natureza denotacional



Acoplamento de
natureza motivacional



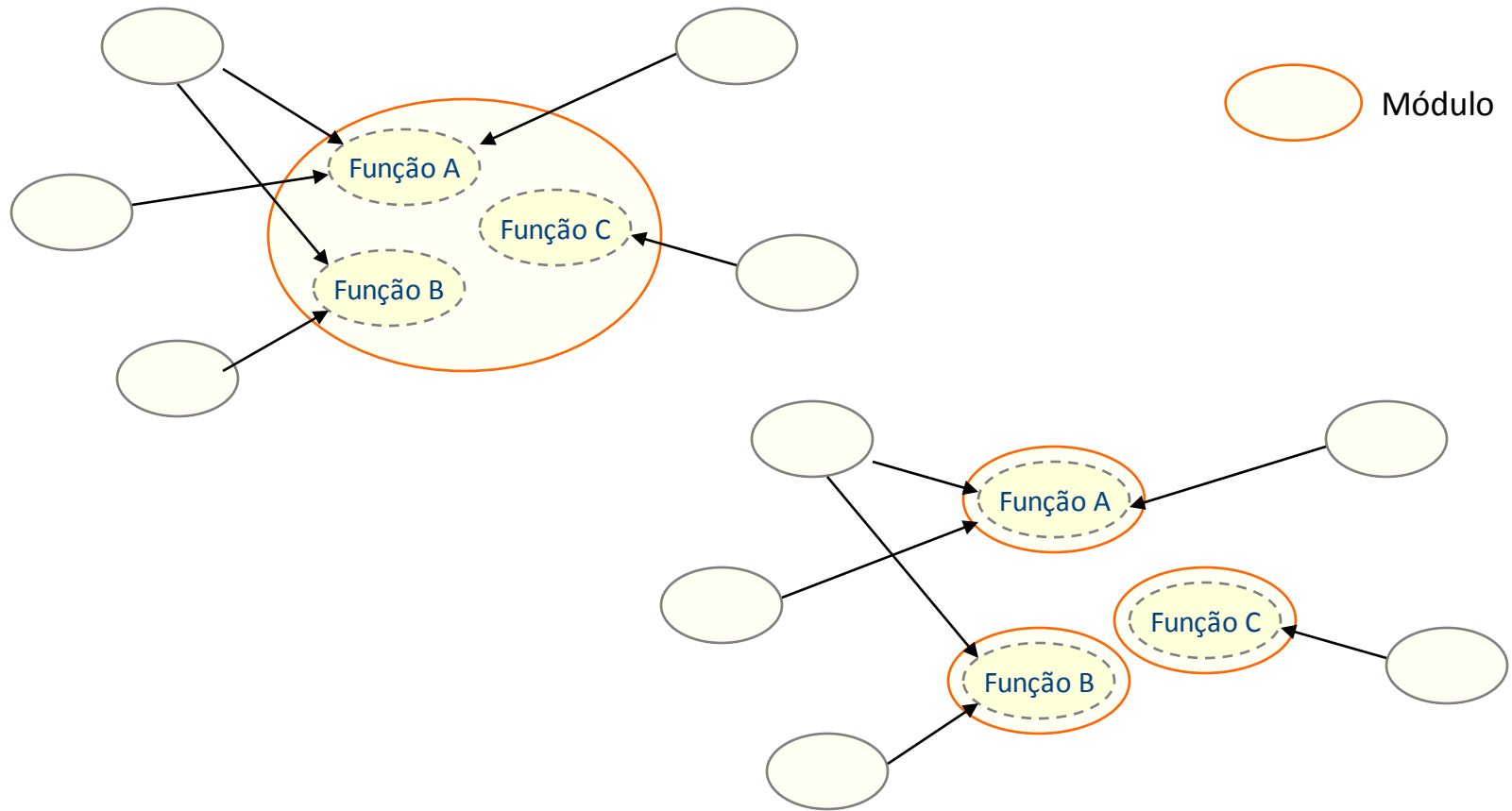
+ Organização estática

Nível de acoplamento

Organização dinâmica

-

COESÃO



Nível de coesão
dos módulos:

BAIXO

ALTO

COESÃO

NÍVEL COERÊNCIA FUNCIONAL DE UM SUBSISTEMA

(até que ponto esse subsistema realiza uma única função)

NÍVEL DE COESÃO

- Um nível de coesão baixo leva a que, em caso de necessidade de **alteração** de um subsistema, o **número de módulos afectados seja elevado**
- Se o **nível de coesão for elevado**, o número de módulos afectados será **minimizado**
- Um módulo com um **nível de coesão baixo** é mais **complexo**, logo mais difícil de conceber e de testar

PRINCÍPIOS DE ARQUITECTURA

- MODULARIDADE

- DECOMPOSIÇÃO
- ENCAPSULAMENTO



+ COESÃO

- ACOPLAMENTO

- ABSTRACÇÃO



COMPLEXIDADE

MODULARIDADE

- **DECOMPOSIÇÃO**

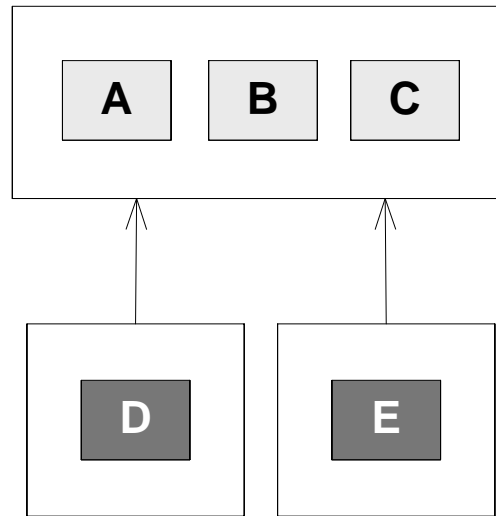
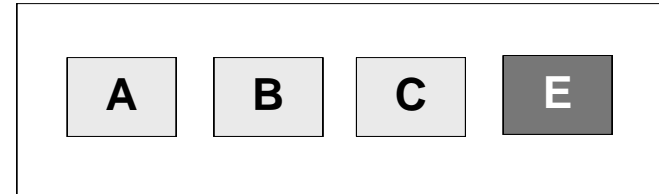
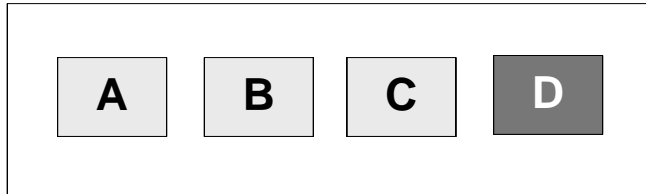
- De um sistema em partes coesas
 - Para sistematizar interacções
 - Para lidar com a explosão combinatória
- **FACTORIZAÇÃO**
 - Eliminação de redundância
 - Garantia de consistência

- **ENCAPSULAMENTO**

- **Isolamento dos detalhes internos** das partes de um sistema em relação ao exterior
 - Para reduzir dependências (interacções)
 - Relacionar estrutura e função no contexto de uma parte
 - Acesso exclusivo através das interfaces disponibilizadas
- **INTERFACES**
 - **Contractos funcionais** para interação com o exterior

FACTORIZAÇÃO

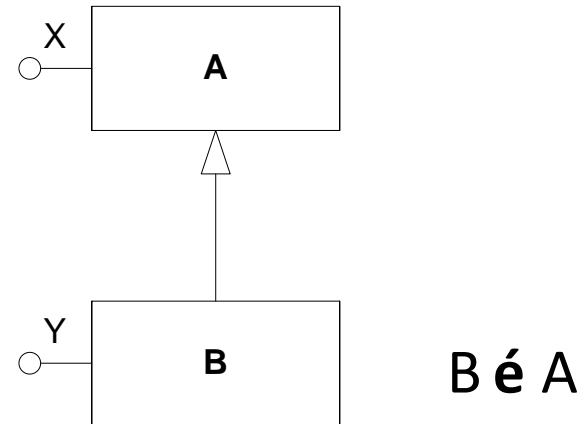
REDUÇÃO DE REDUNDÂNCIA



MECANISMOS DE FACTORIZAÇÃO

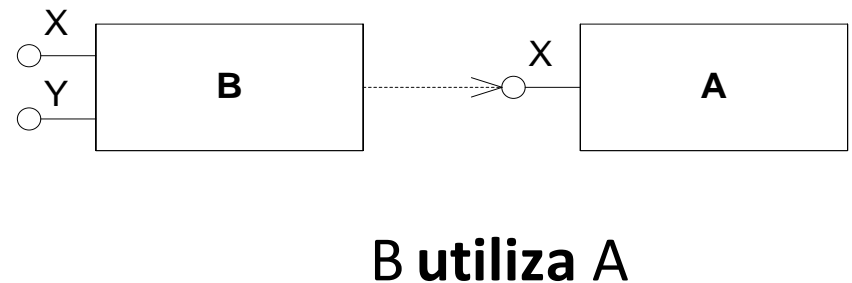
HERANÇA

- Nível de **acoplamento alto**
- B é A



DELEGAÇÃO

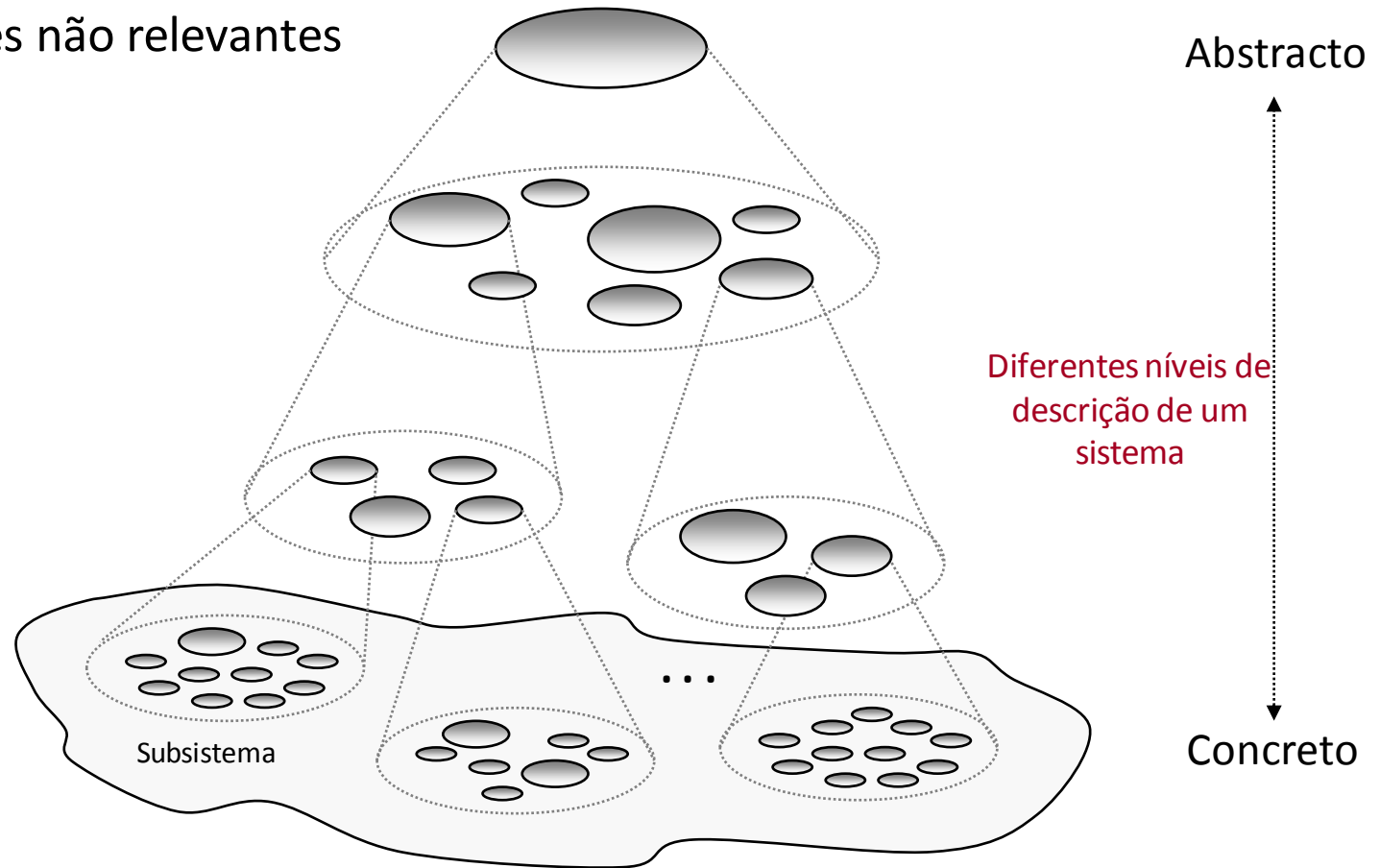
- Nível de **acoplamento baixo**
- B utiliza A
- Agregação de partes
- Acoplamento pode variar dinamicamente



ABSTRACÇÃO

FERRAMENTA BASE PARA LIDAR COM A COMPLEXIDADE

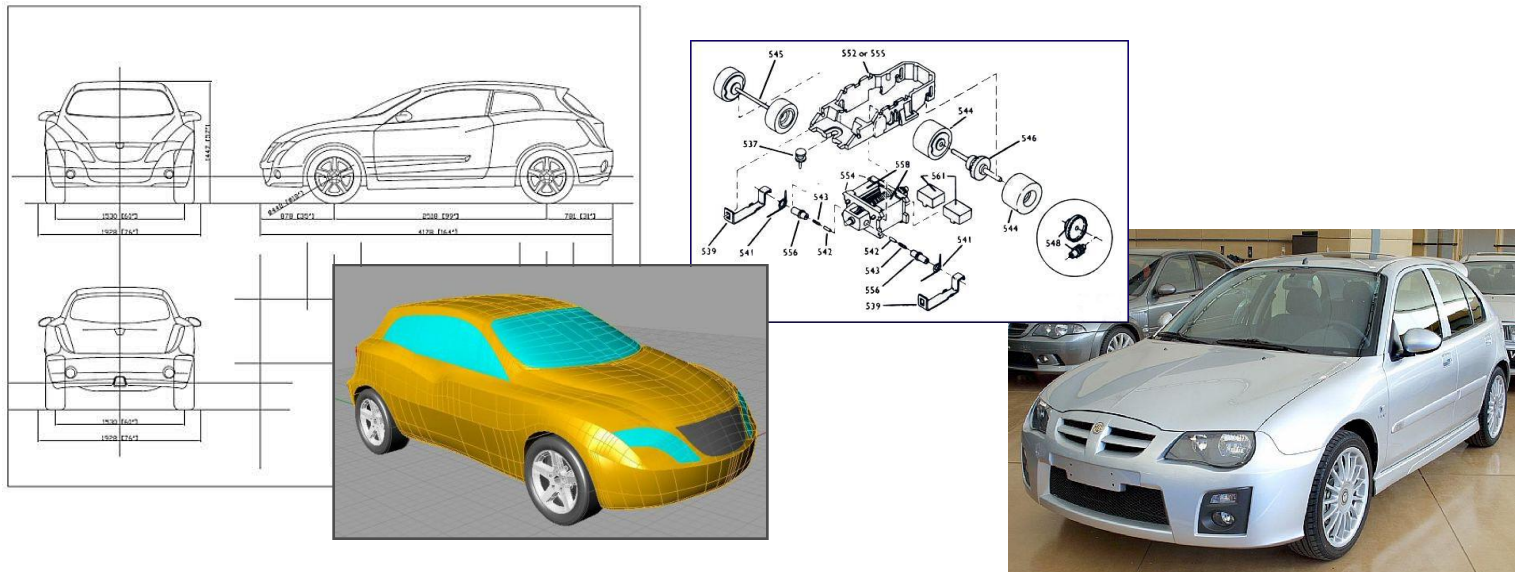
- **Realçar** o que é **essencial**, omitir detalhes não relevantes
- **Simplificação**
- **Focagem**
- **Modelos**



ABSTRACÇÃO

CONTROLO DA COMPLEXIDADE

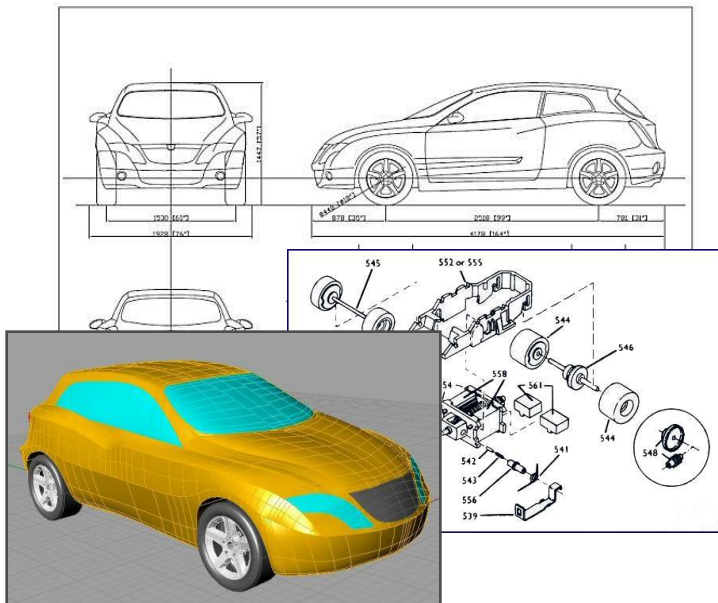
- Abstracção como ferramenta essencial para lidar com a complexidade
- Obtenção e sistematização progressiva de conhecimento
- **MODELO**



MODELOS EM ENGENHARIA

O PROBLEMA DOS MODELOS

- **A realidade é muito mais rica que qualquer abstracção!**
- “... the good thing about bubbles and arrows, as opposed to programs, is that they never crash.” [Meyer, 1997]



HIATO
SEMÂNTICO



- Especificidades dos materiais
- Efeitos de escala
- Métodos de construção
- Perícia dos construtores
- Falhas de comunicação

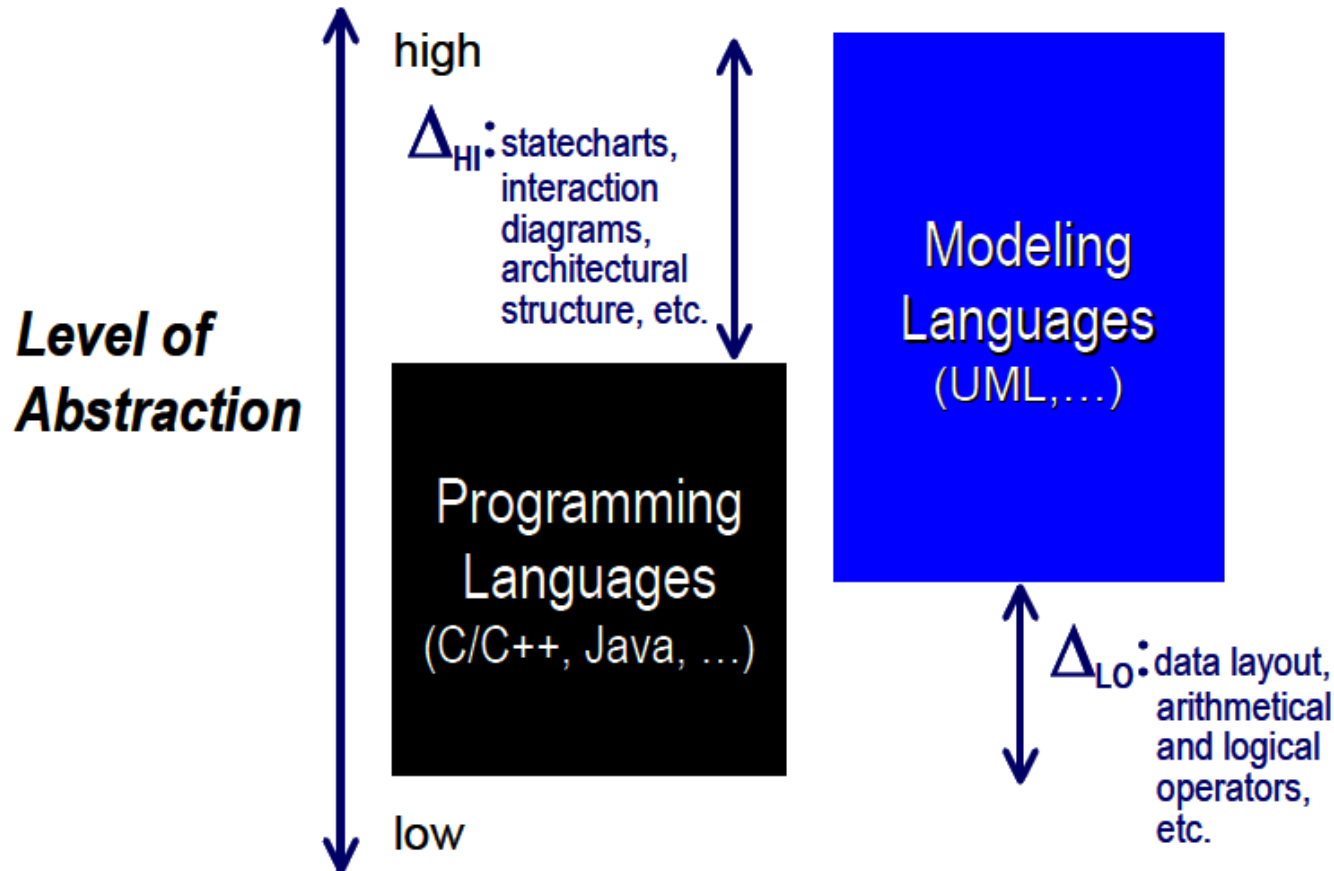


Discrepâncias entre modelo e realização
Falhas de operação

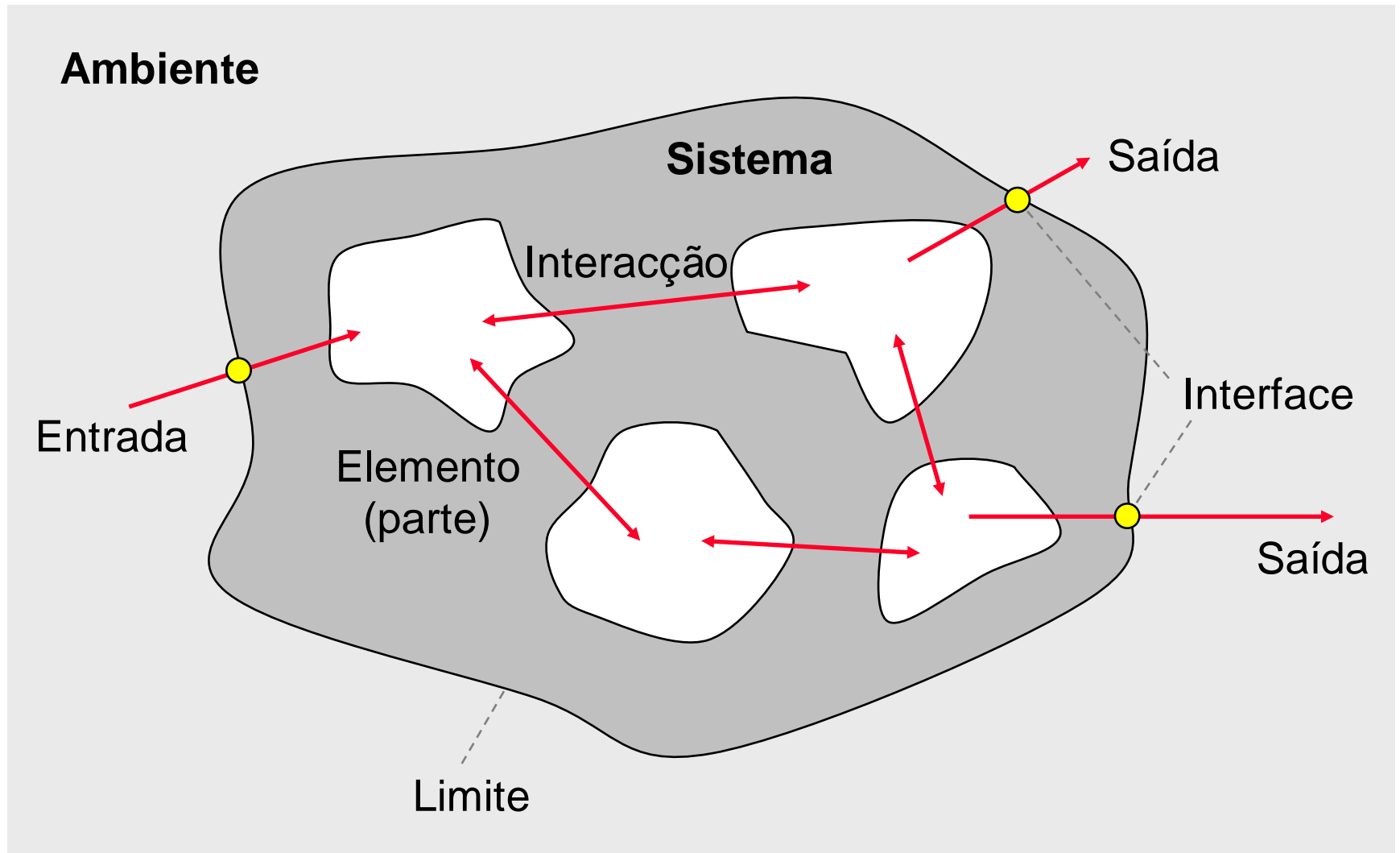
**NECESSIDADE DE LIGAÇÃO EFICAZ
ENTRE MODELOS E REALIZAÇÃO**

LINGUAGENS DE MODELAÇÃO

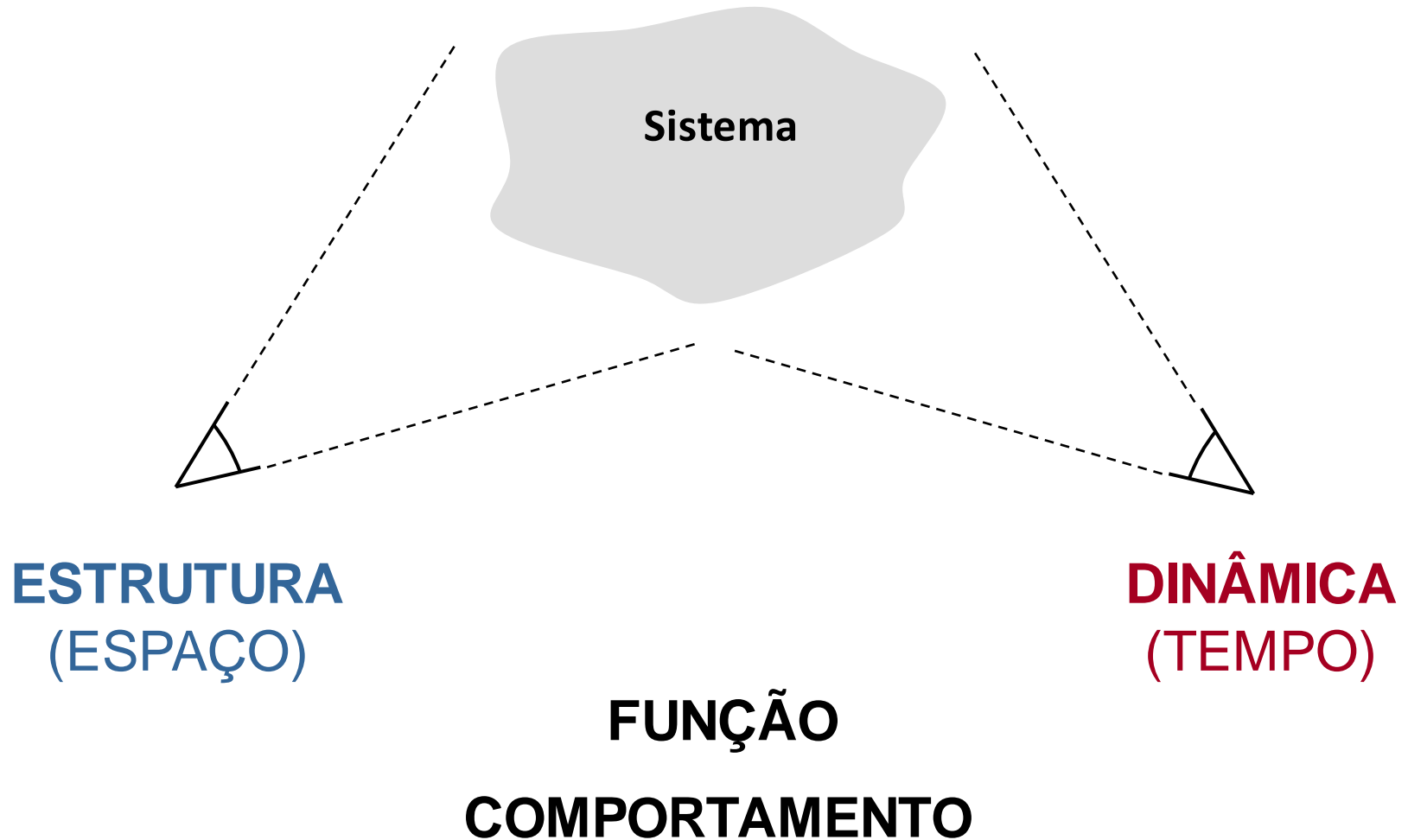
DESCRIÇÃO DO SISTEMA A DIFERENTES NÍVEIS DE ABSTRACÇÃO



MODELAÇÃO DE UM SISTEMA



PERSPECTIVAS DE MODELAÇÃO

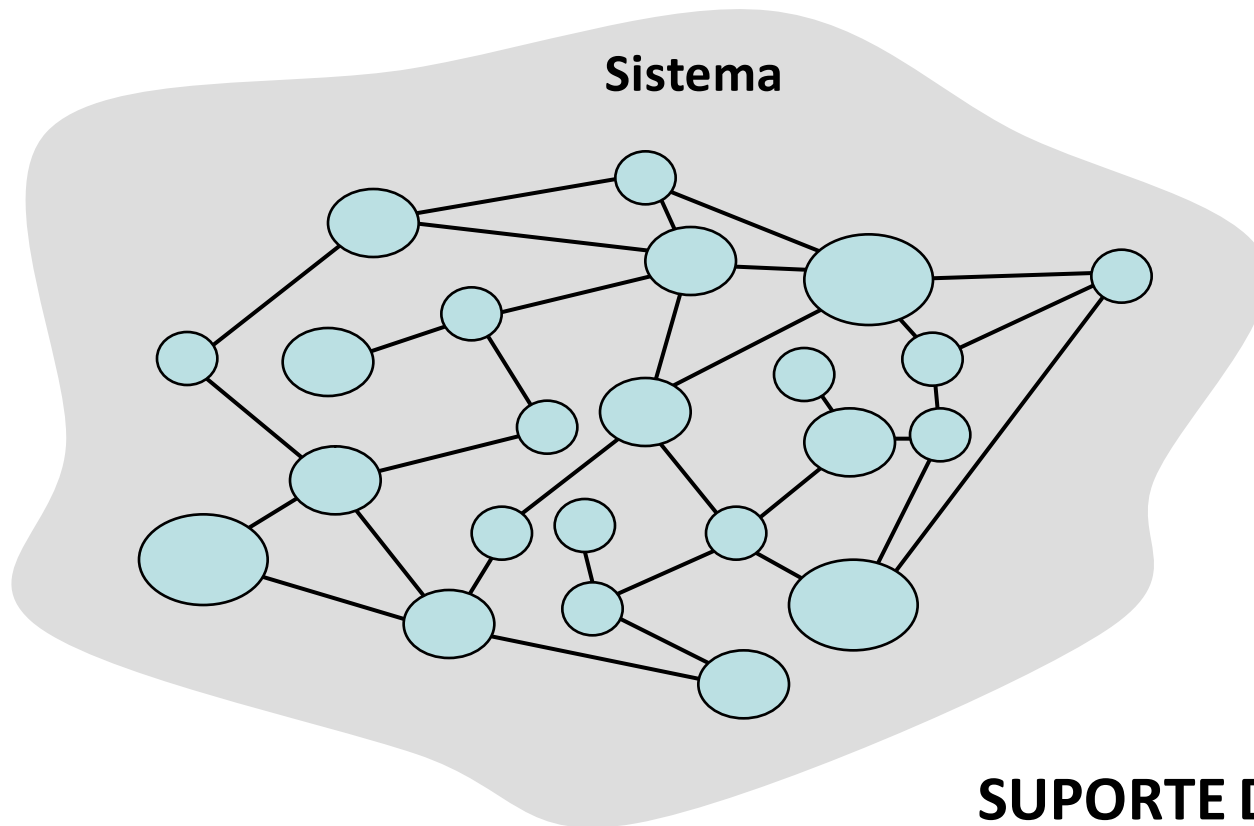


ESTRUTURA DE UM SISTEMA

ESPAÇO

PARTES DO SISTEMA

RELAÇÕES ENTRE PARTES

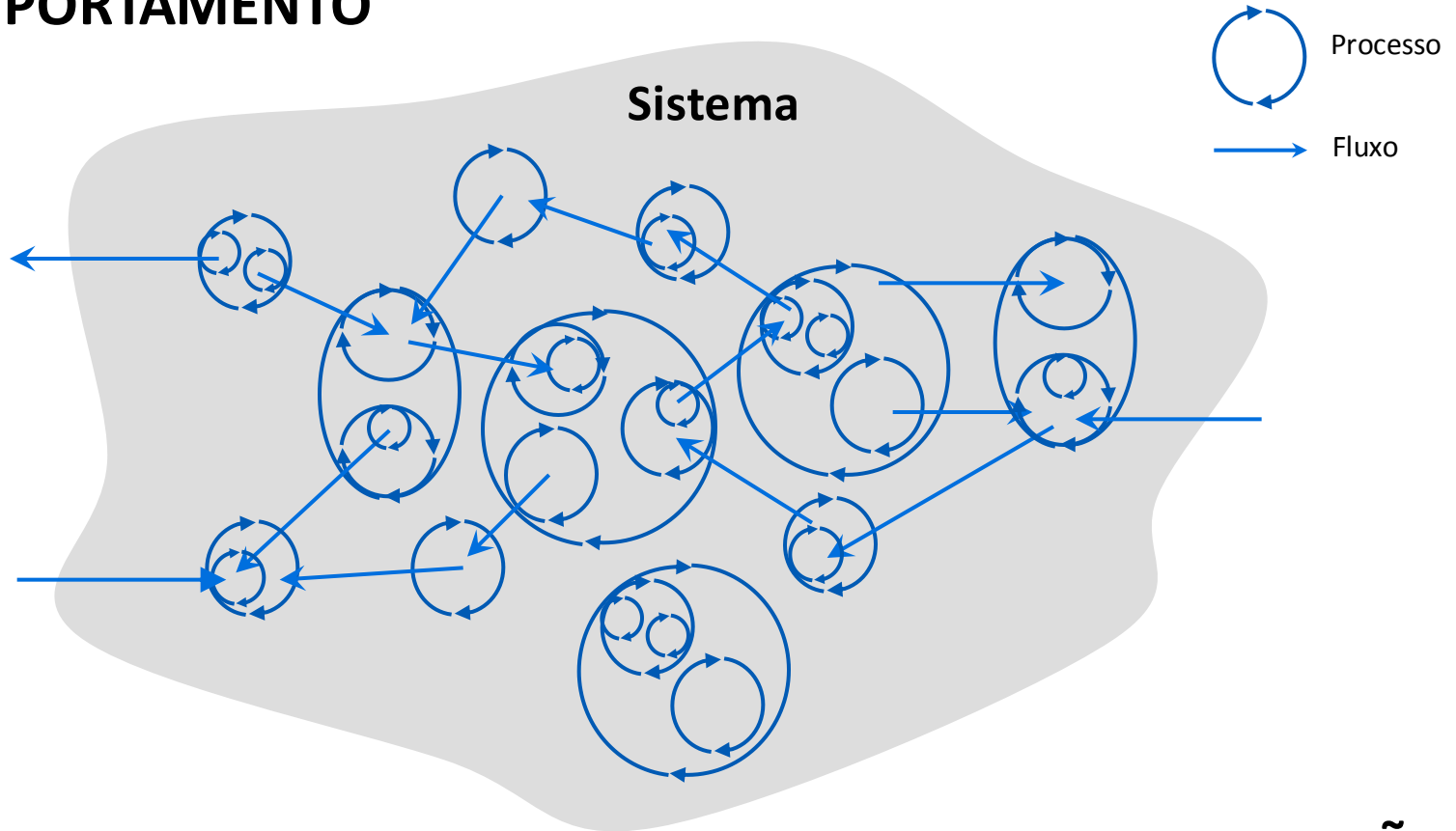


SUPOORTE DA FUNÇÃO

DINÂMICA DE UM SISTEMA

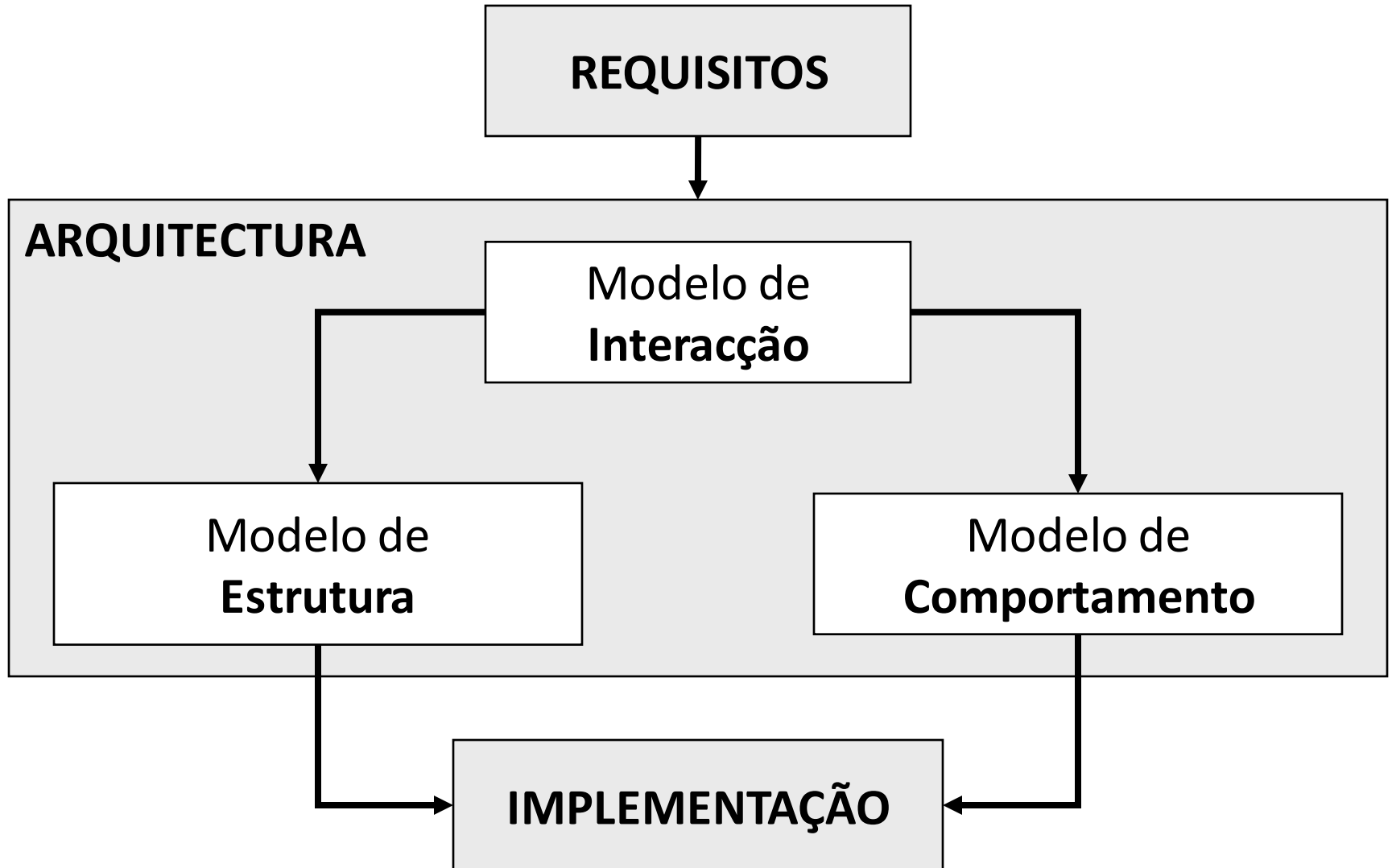
FUNÇÃO

COMPORTAMENTO

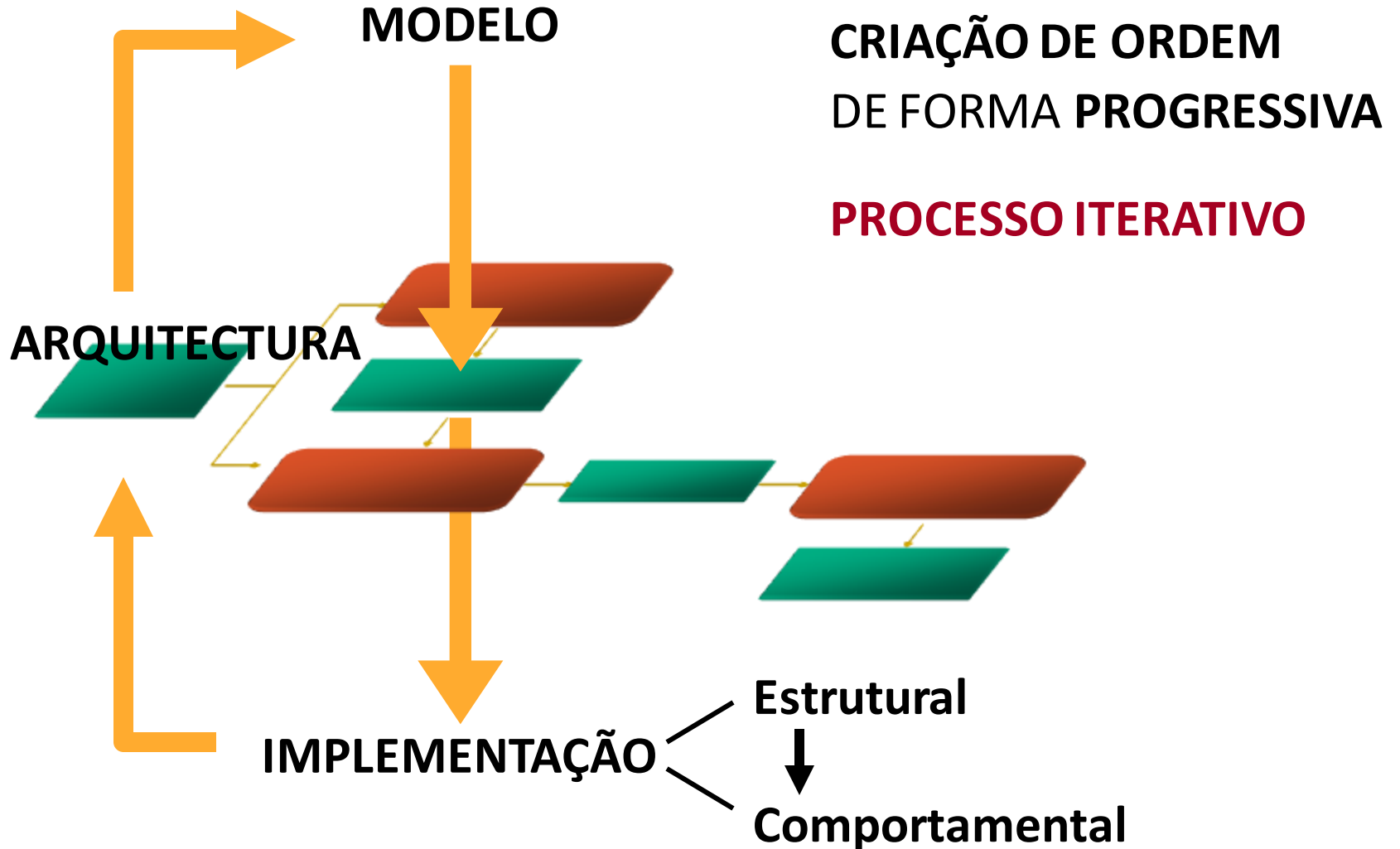


TRANSFORMAÇÃO
DINÂMICA

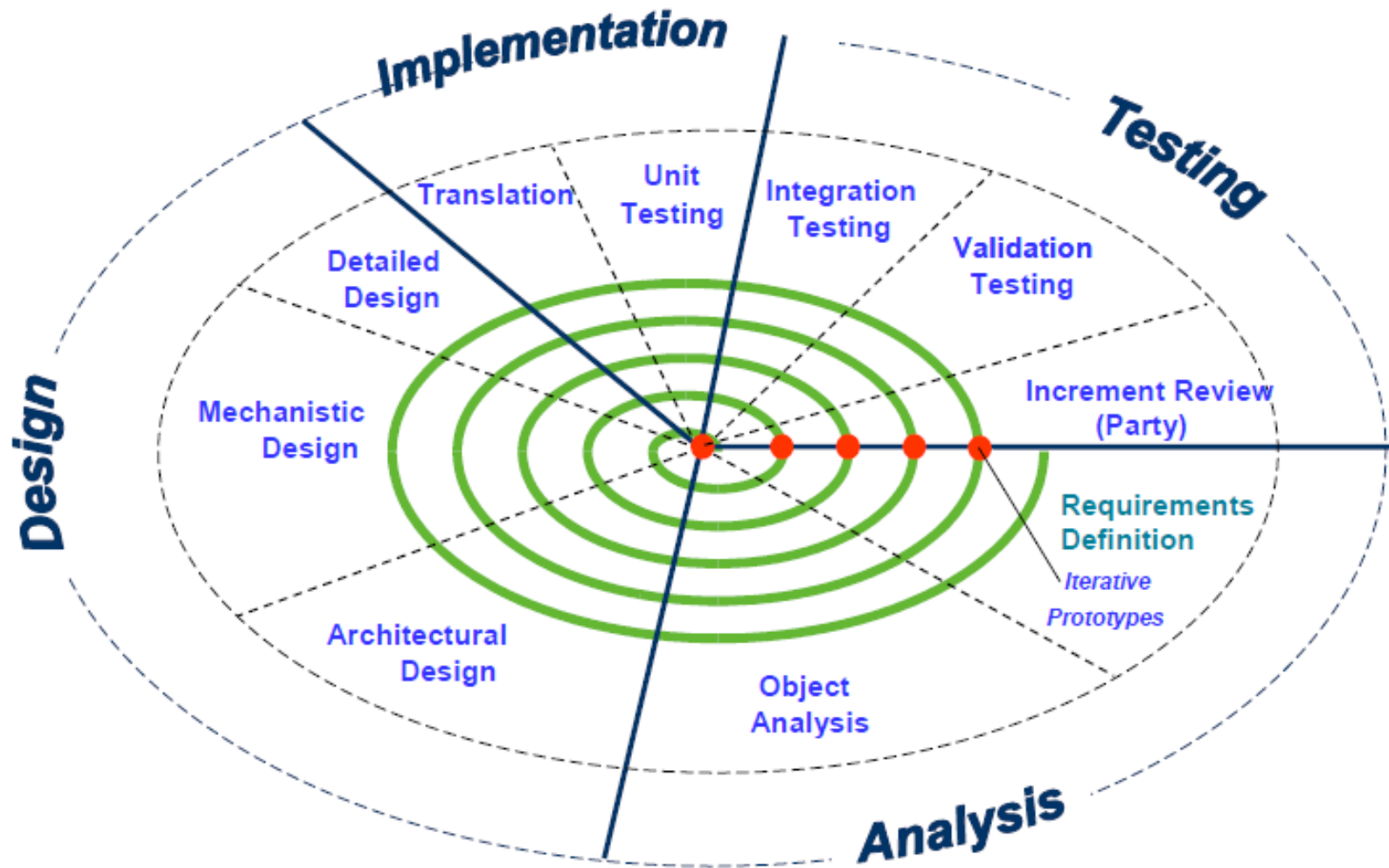
PROCESSO DE DESENVOLVIMENTO



PROCESSO DE DESENVOLVIMENTO



PROCESSO DE DESENVOLVIMENTO



[Douglass, 2006]

BIBLIOGRAFIA

[Pressman, 2003]

R. Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.

[Booch et al., 1998]

G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1998.

[Miles & Hamilton, 2006]

R. Miles, K. Hamilton, *Learning UML 2.0*, O'Reilly, 2006.

[Eriksson et al., 2004]

H. Eriksson, M. Penker, B. Lyons, D. Fado, *UML 2 Toolkit*, Wiley, 2004.

[Douglass, 2009]

B. Douglass, *Real-Time Agility: The Harmony/ESW Method for Real-Time and Embedded Systems Development*, Addison-Wesley, 2009.