
Engenharia de Software

Modelos Computacionais

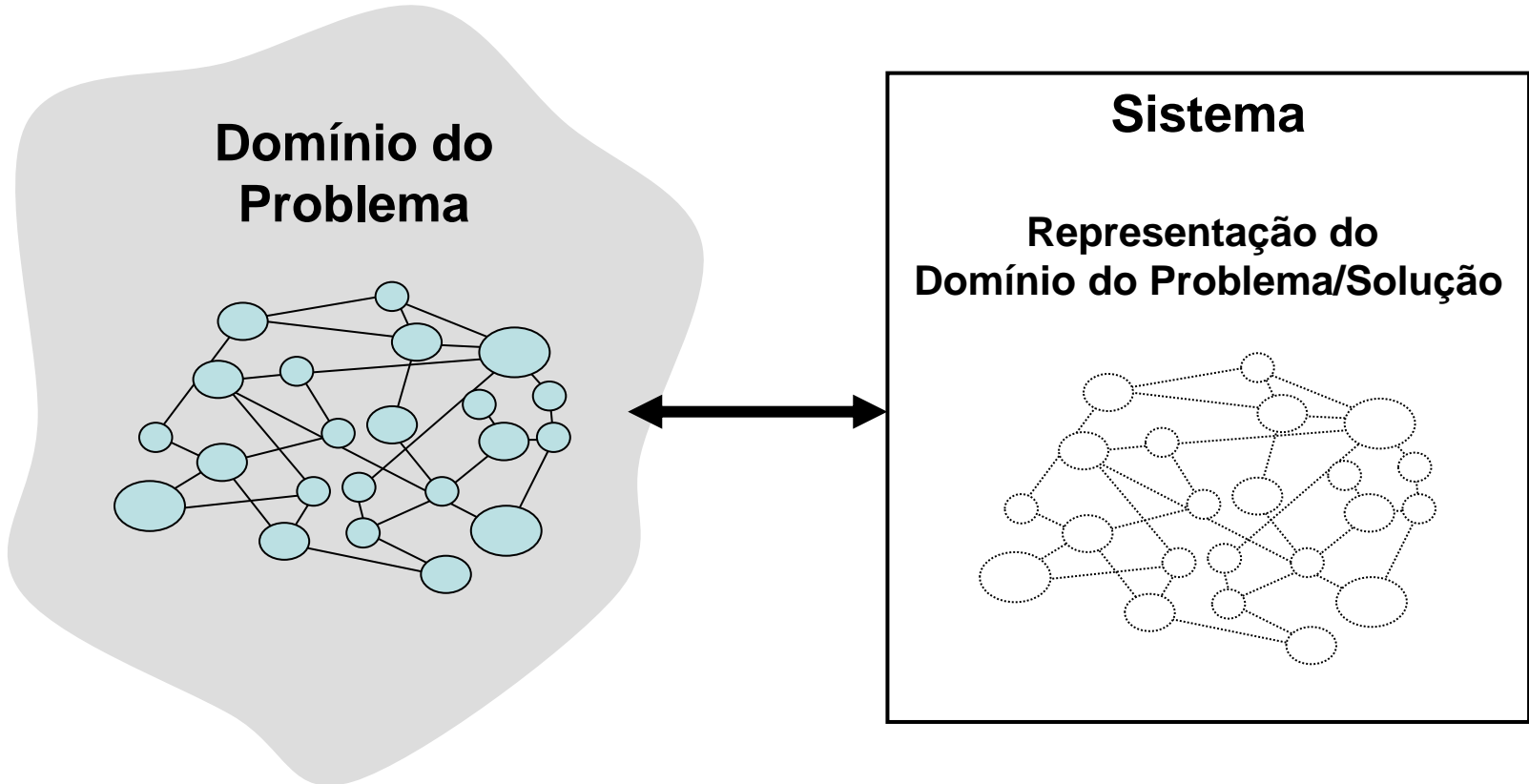
Luís Morgado

Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

ESPECIFICAÇÃO DE SOFTWARE

REPRESENTAÇÃO DO DOMÍNIO DO PROBLEMA/SOLUÇÃO



ESPECIFICAÇÃO DE SOFTWARE

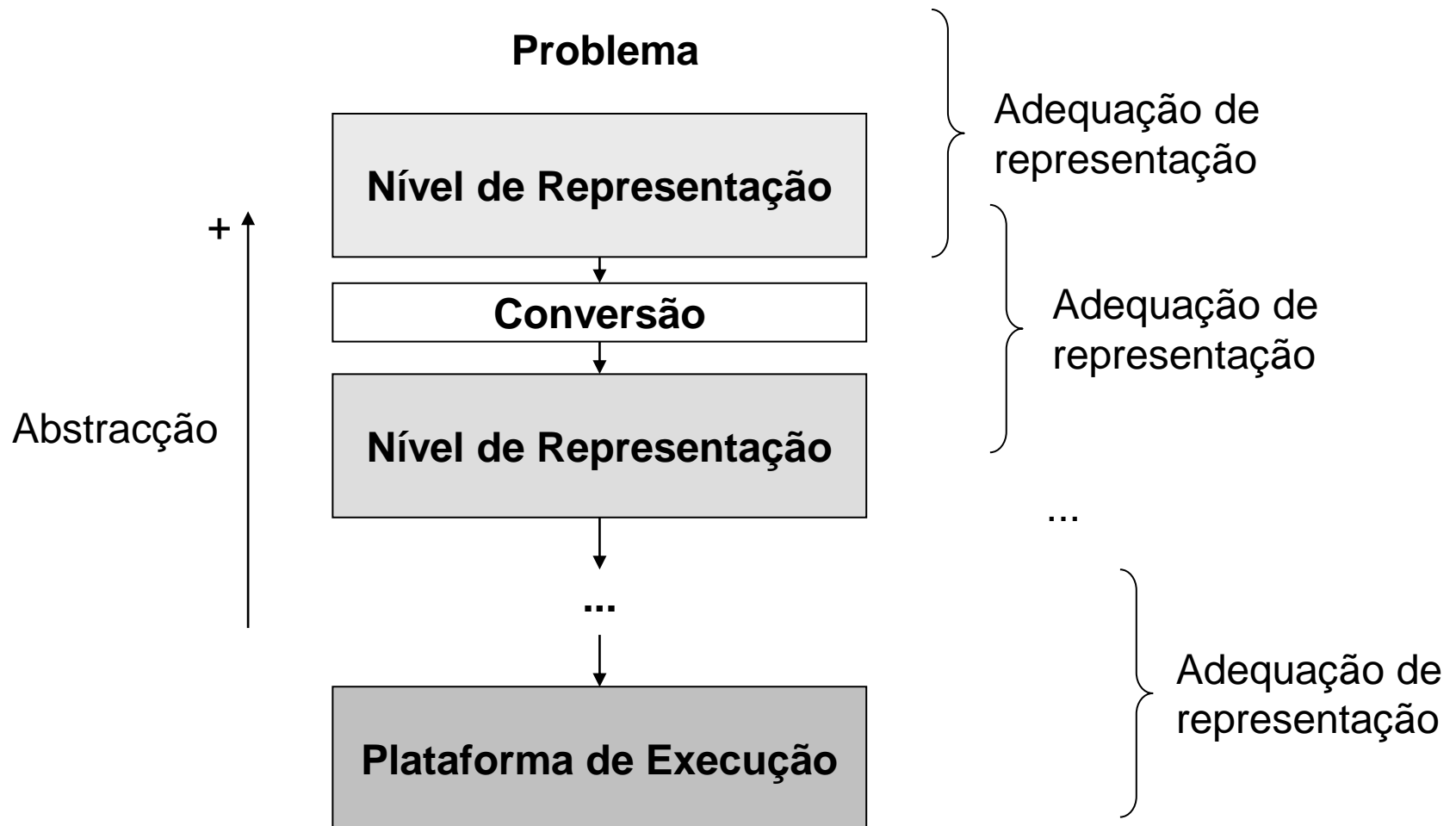
REPRESENTAÇÃO:

- A **notação** (*sintaxe*) a utilizar
- A **denotação** (*semântica*) das entidades representadas
- A forma de **manipulação** das entidades representadas

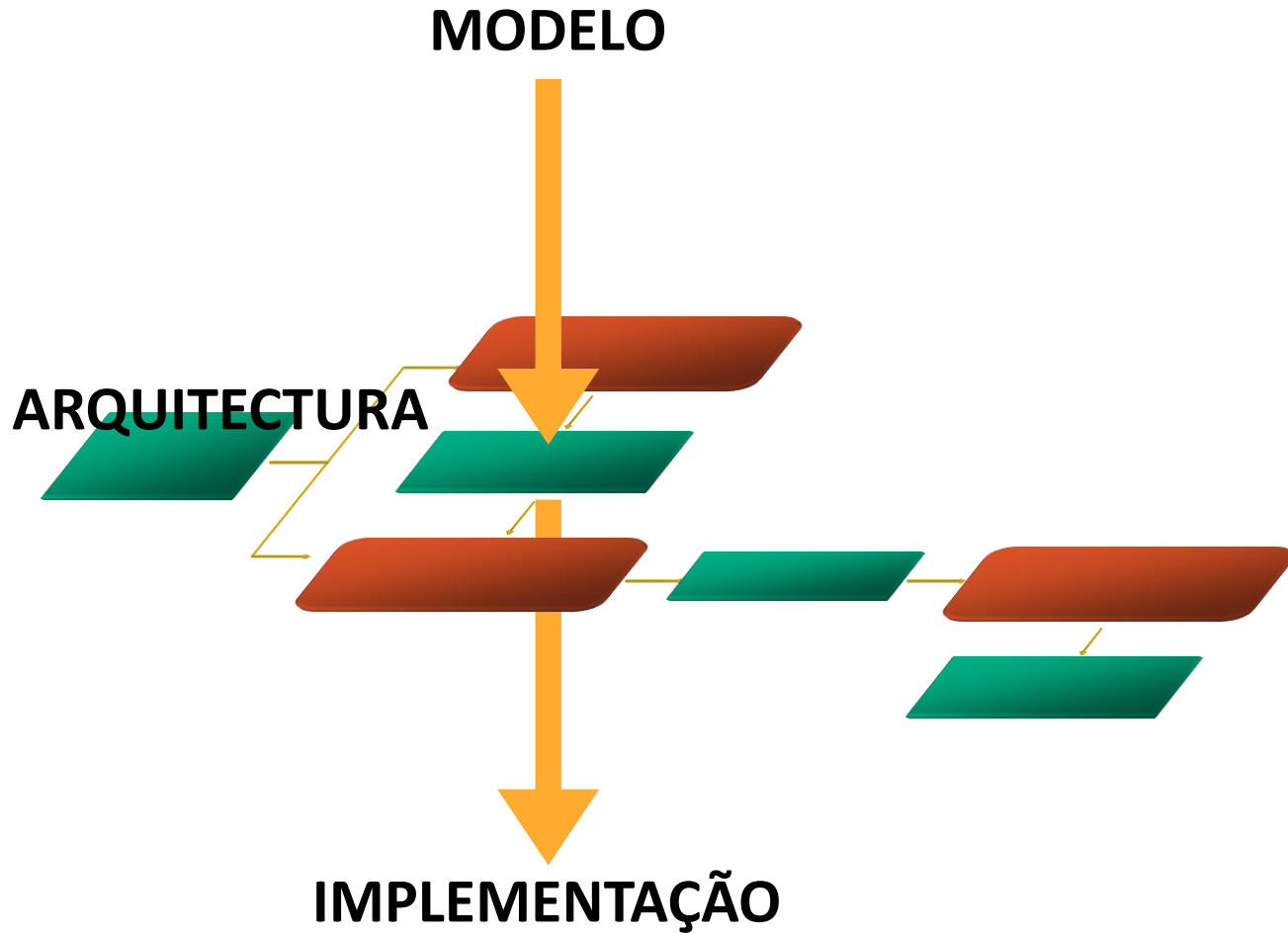
**REPRESENTAÇÃO = NOTAÇÃO +
DENOTAÇÃO +
MANIPULAÇÃO**

ESPECIFICAÇÃO DE SOFTWARE

O PROBLEMA DA REPRESENTAÇÃO (LINGUAGEM)

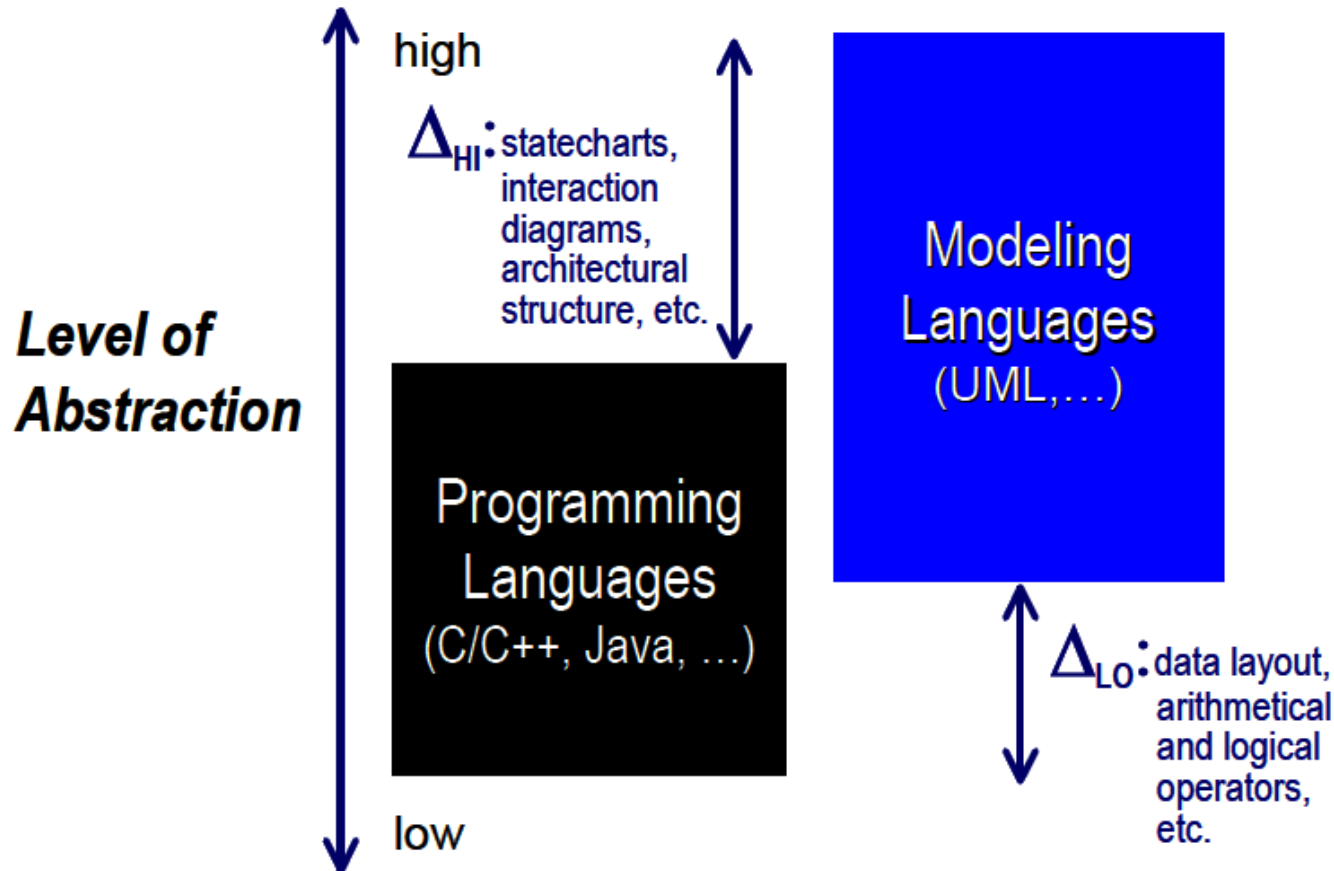


NÍVEIS DE REPRESENTAÇÃO

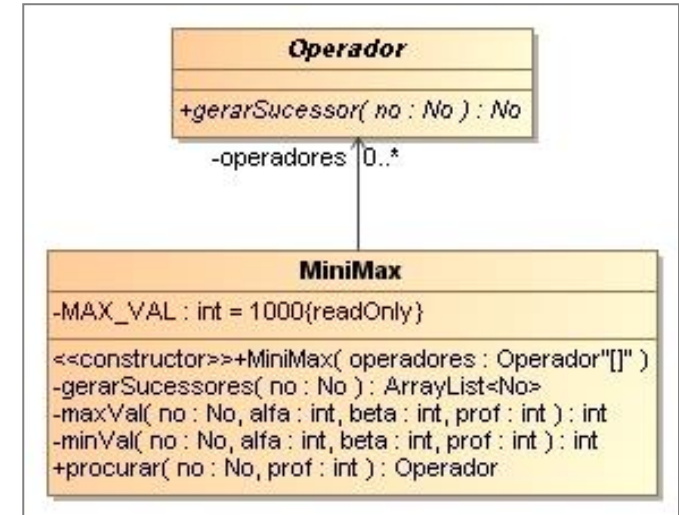
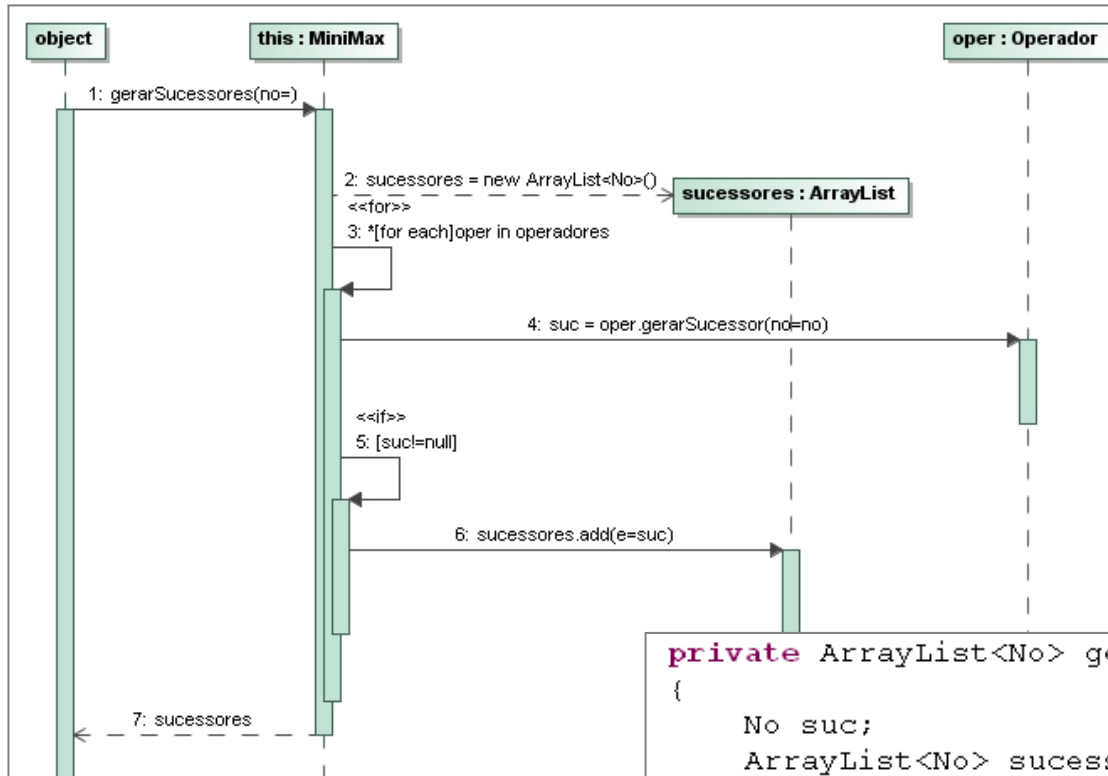


LINGUAGENS DE MODELAÇÃO

DESCRIÇÃO DO SISTEMA A DIFERENTES NÍVEIS DE ABSTRACÇÃO



MODELOS DE SOFTWARE



```

private ArrayList<No> gerarSucessores(No no)
{
    No suc;
    ArrayList<No> sucessores = new ArrayList<No>();

    // Para todos os operadores gerar sucessor do nó
    for(Operador oper : operadores) {
        suc = oper.gerarSucessor(no);
        if(suc != null)
            sucessores.add(suc);
    }

    return sucessores;
}
  
```

Descrições de um sistema em diferentes perspectivas e níveis de abstracção

MODELOS DE SOFTWARE

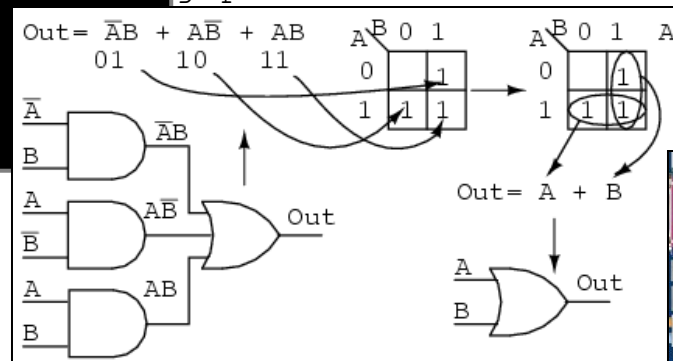
```
private ArrayList<No> gerarSucessores(No no)
{
    No suc;
    ArrayList<No> sucessores = new ArrayList<No>();

    // Para todos os operadores gerar sucessor do nó
    for(Operador oper : operadores) {
        suc = oper.gerarSucessor(no);
        if(suc != null)
            sucessores.add(suc);
    }
}
```

Descrições de um sistema em diferentes perspectivas e níveis de abstracção

```
004113CE C7 05 AC 71 41 00 00 00 00 00
004113D8 83 3D A0 71 41 00 00
004113DF 7E 1F
004113E1 A1 AC 71 41 00
004113E6 03 05 9C 71 41 00
004113EC A3 AC 71 41 00
004113F1 A1 A0 71 41 00
004113F6 83 E8 01
004113F9 A3 A0 71 41 00
004113FE EB D8
00411400 5F
```

```
mov     dword ptr [_res (4171ACh)],0
cmp     dword ptr [_x (4171A0h)],0
jle     411400h
mov     eax,dword ptr [_res (4171ACh)]
add     eax,dword ptr [_y (41719Ch)]
mov     dword ptr [_res (4171ACh)],eax
mov     eax,dword ptr [_x (4171A0h)]
sub     eax,1
mov     dword ptr [_x (4171A0h)],eax
jmp     4113D8h
```



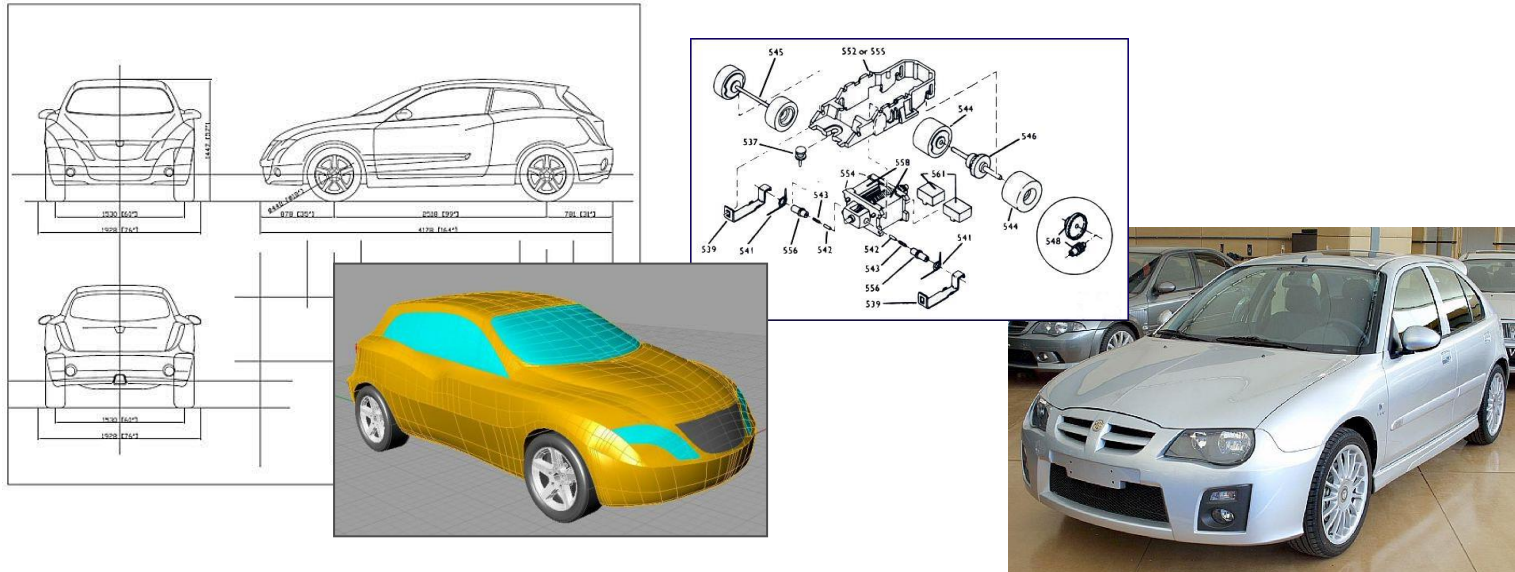
MODELO

- **Representação abstracta de um sistema**
 - Especificação com base em conceitos abstractos das características fundamentais de um sistema
- **Meio para lidar com a complexidade**
 - Compreensão de um sistema
 - Comunicação acerca do sistema
 - Especificação de referência para a realização do sistema
 - Documentação de um sistema

ABSTRACÇÃO

CONTROLO DA COMPLEXIDADE

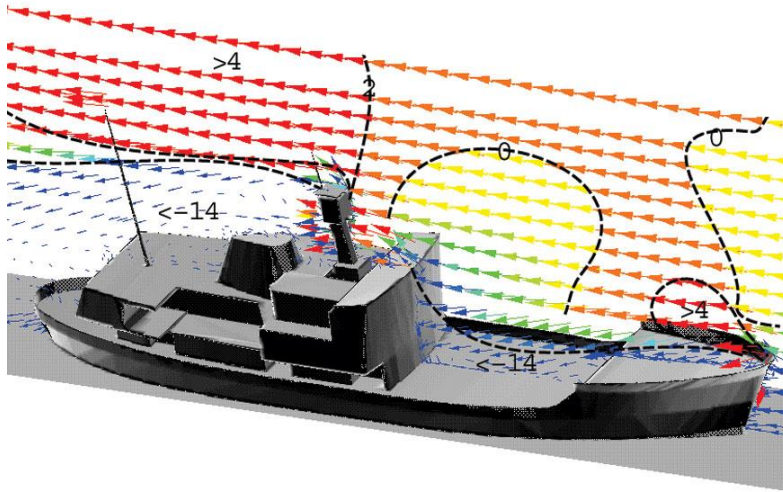
- Abstracção como ferramenta essencial para lidar com a complexidade
- Obtenção e sistematização progressiva de conhecimento
- **MODELO**



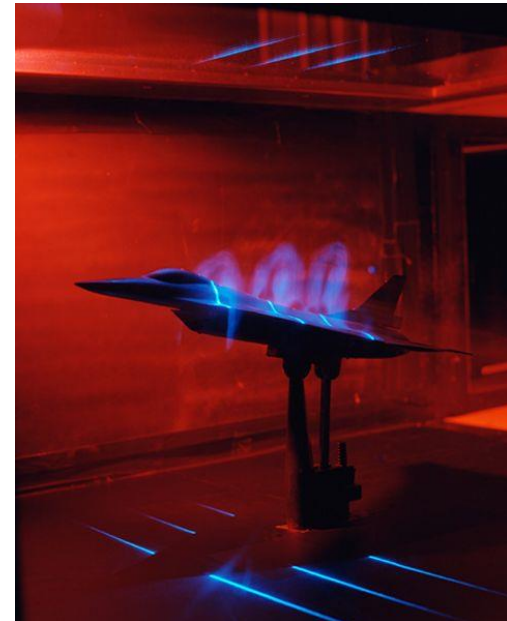
MODELOS EM ENGENHARIA

REDUÇÃO DE INCERTEZA / RISCO

- Elaboração de modelos para verificação de propriedades através de simulação
- Obtenção de conhecimento antes de construir o sistema concreto



[Yelland et al., 2002]

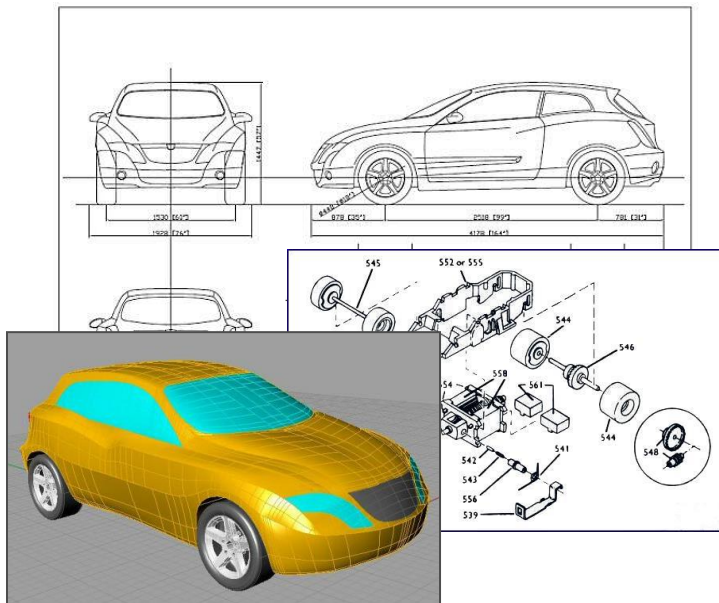


[aerospaceweb.org]

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**

MODELAÇÃO DE UM SISTEMA

DEFINIÇÃO DOS PADRÕES DE ORGANIZAÇÃO DO SISTEMA

CARACTERÍSTICAS IMPORTANTES DE UM MODELO

– ABSTRACÇÃO

- Foco nos aspectos importantes, remoção de aspectos não relevantes

– COMPREENSÃO

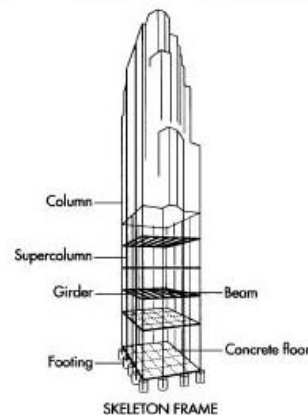
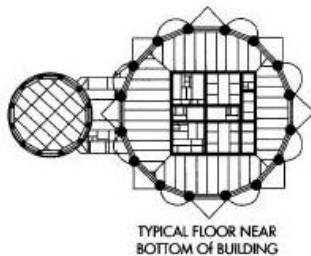
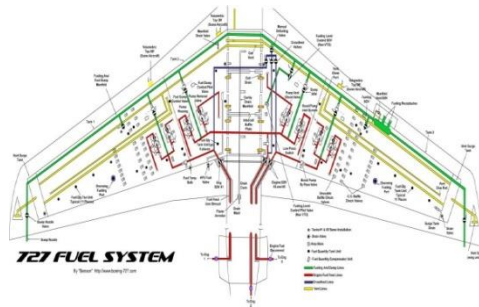
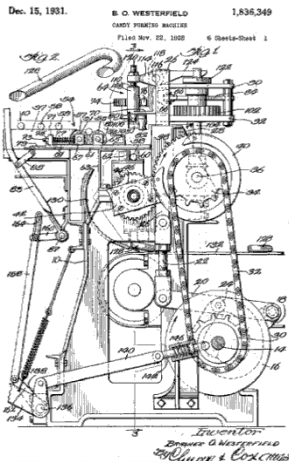
- Facilidade de transmissão e compreensão das ideias envolvidas

– PRECISÃO

- Representação correcta e rigorosa do sistema

– PREVISÃO

- Possibilidade de inferência de conhecimento correcto acerca do sistema descrito



TIPOS DE REPRESENTAÇÃO

DECLARATIVO

- Representação **declarativa** de *o que* se sabe acerca de um domínio
- Controlo não representado explicitamente
- “*Saber que ...*”

IMPERATIVO

- Representação **procedimental** de *como* obter um resultado específico
- Controlo representado explicitamente
- “*Saber como ...*”

PARADIGMAS DE PROGRAMAÇÃO

- **FUNCIONAL**
 - Computação definida com base em avaliação de funções (estruturas de dados imutáveis)
- **LÓGICO**
 - Computação definida com base em inferência lógica
- **ESTRUTURADO**
 - Computação definida com base em procedimentos e estruturas de dados modulares
- **ORIENTADO A OBJECTOS**
 - Computação definida com base em objectos encapsulados que interagem através de mensagens (simulação do mundo)
- **ORIENTADO A SERVIÇOS**
 - Computação definida com base na composição de serviços modulares
- **ORIENTADO A AGENTES**
 - Computação definida com base em agentes autónomos

ABORDAGEM ESTRUTURADA

- **Vantagens**

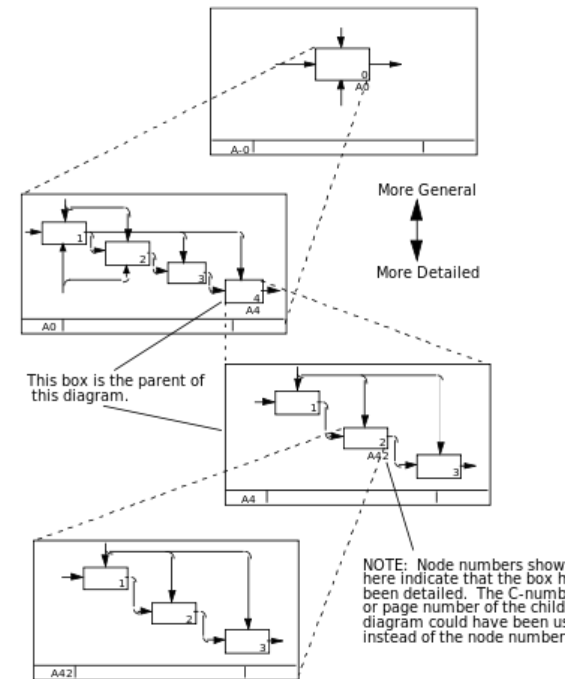
- Modularidade

- Dados
 - Função
 - Comportamento

- **Problema**

- Estado externo

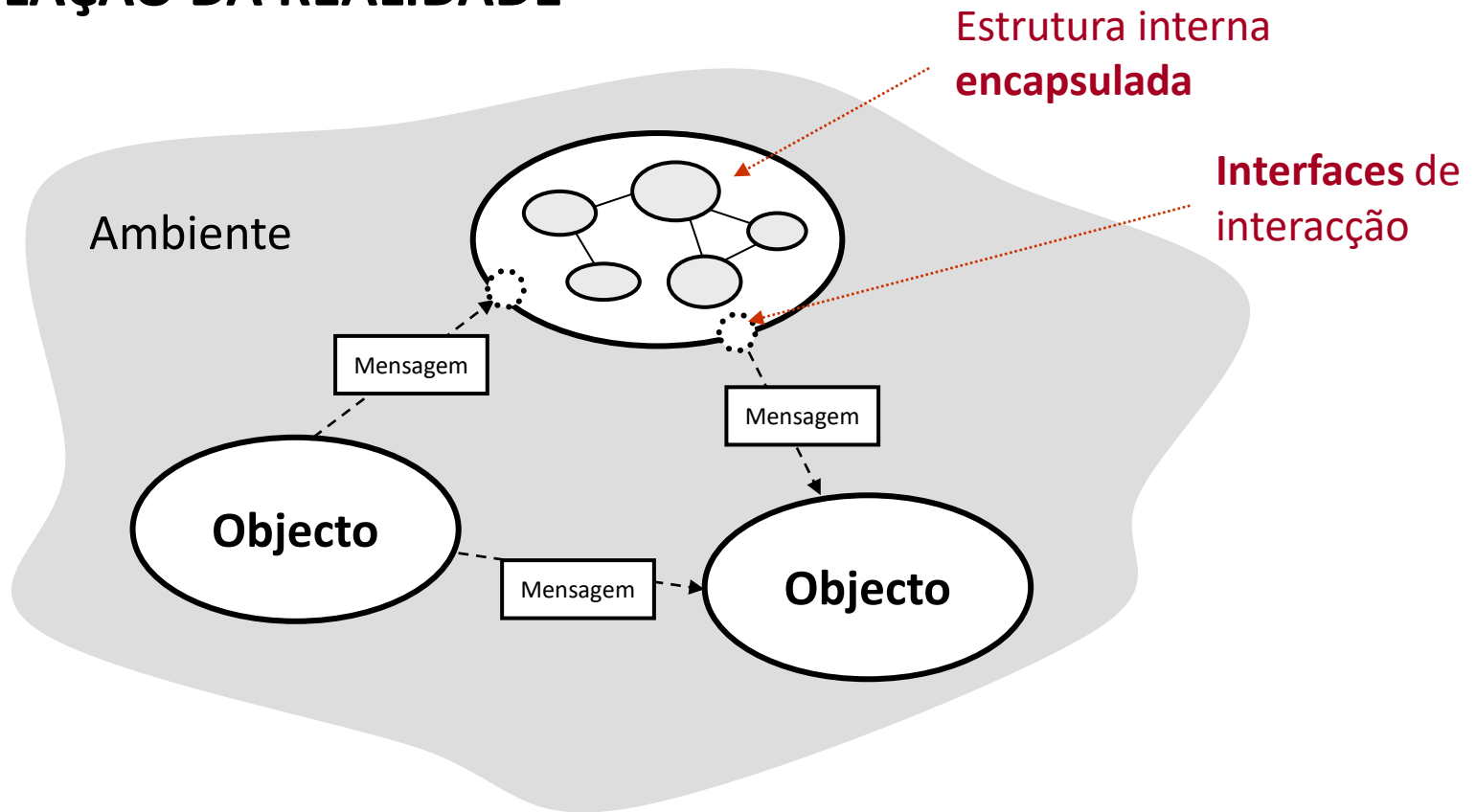
- **Complexidade combinatória** na relação entre dados, função e comportamento
 - Complexidade intra-modular
 - Complexidade inter-modular



ABORDAGEM ORIENTADA A OBJECTOS

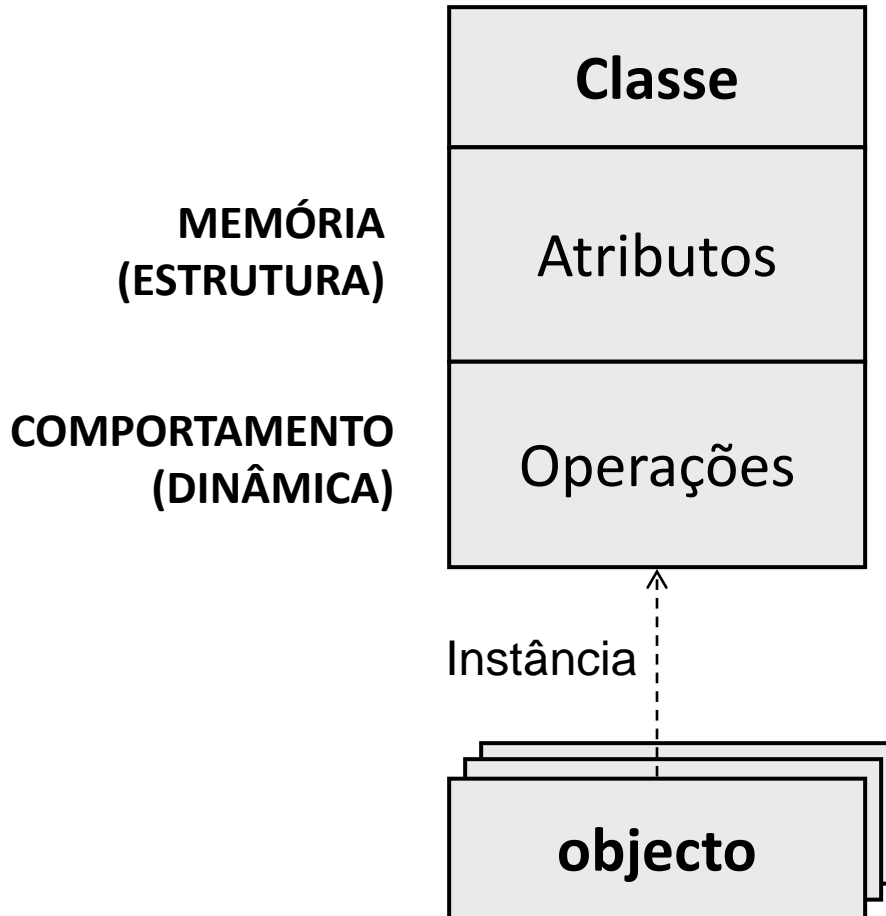
SISTEMA MODELADO COMO UM CONJUNTO DE OBJECTOS QUE INTERAGEM PARA PRODUIR O COMPORTAMENTO PRETENDIDO

SIMULAÇÃO DA REALIDADE



MODELOS ORIENTADOS A OBJECTOS

MODELO COMPUTACIONAL



HERANÇA

- Super-classes
- Sub-classes
 - Os objectos de uma sub-classe partilham todas as características dos objectos da respectiva super-classe

POLIMORFISMO

- Capacidade de assumir múltiplas formas
 - Classes
 - Operações

MODELOS ORIENTADOS A OBJECTOS

CONCEITO

OBJECTO

COMUNICAÇÃO
(entre objectos)

MENSAGEM

- Envio
- Recepção

REPRESENTAÇÃO

TIPO DE OBJECTO (CLASSE)

RELAÇÃO
(entre objectos)

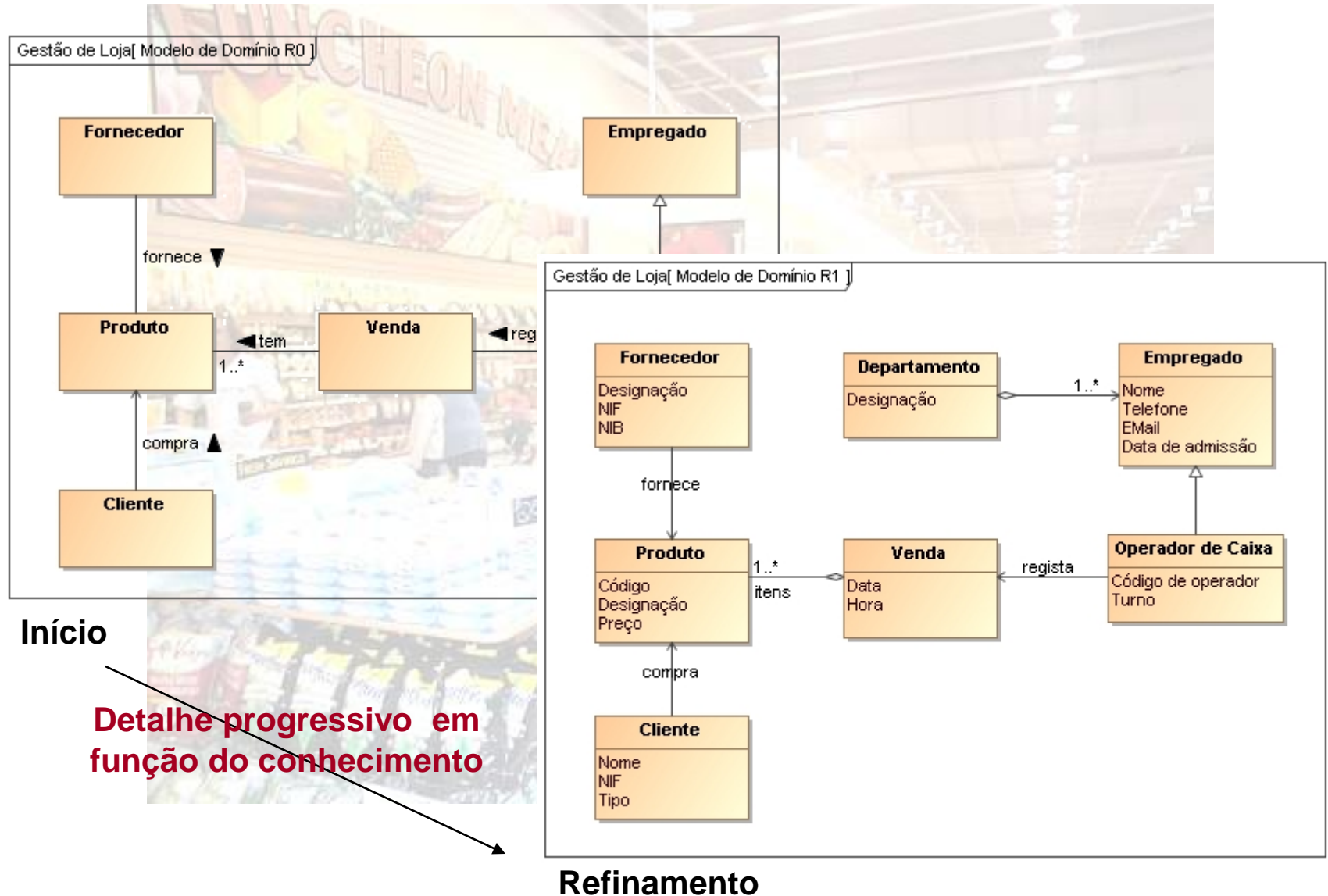
OPERAÇÃO (MÉTODO)

- Evocação
- Execução

MODELOS ORIENTADOS A OBJETOS



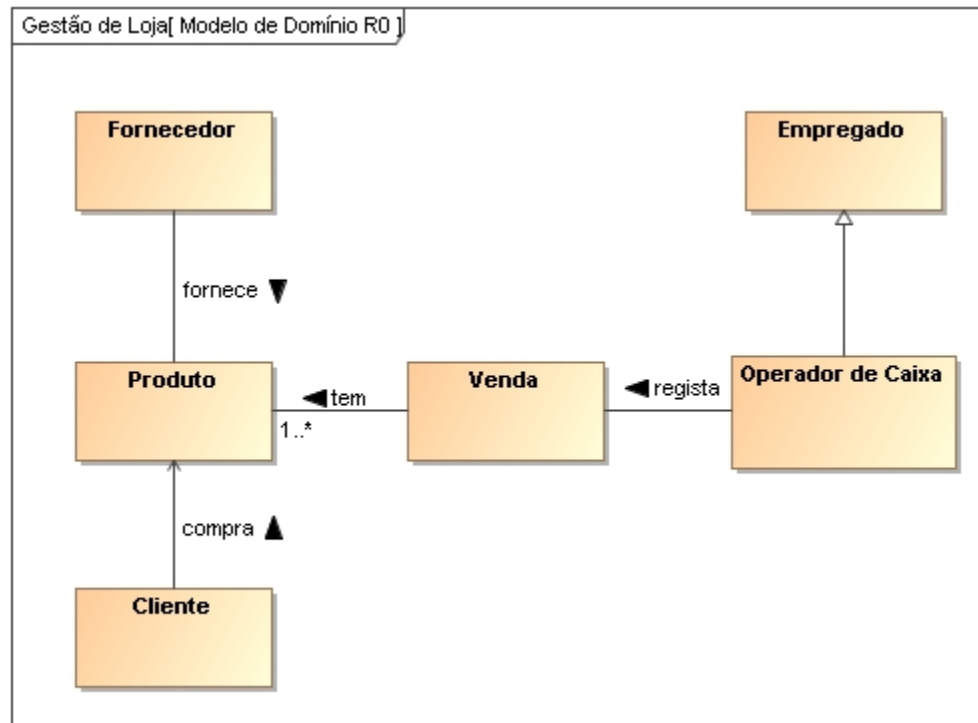
MODELOS ORIENTADOS A OBJECTOS



MODELOS ORIENTADOS A OBJECTOS

RELAÇÕES ENTRE OBJECTOS

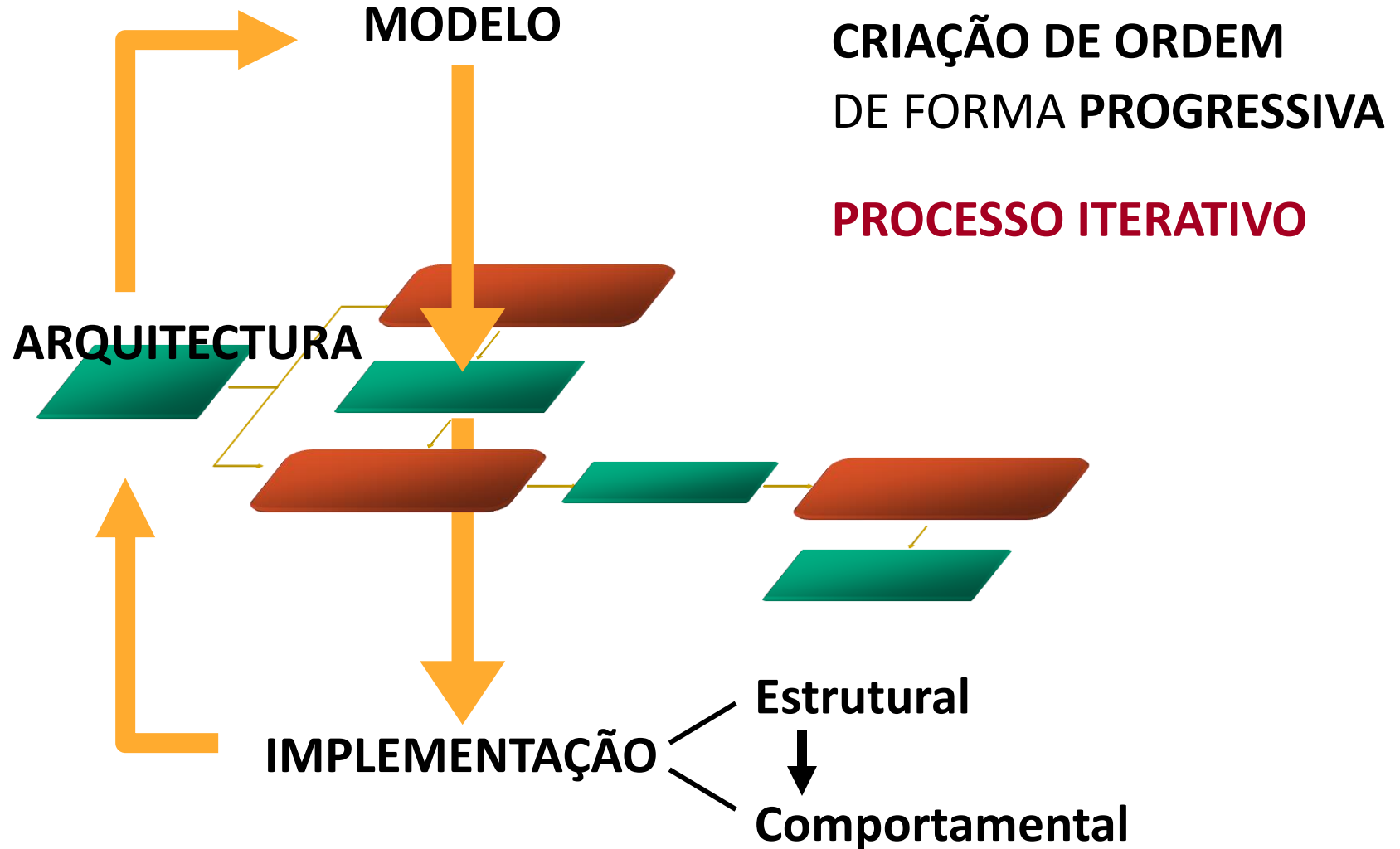
- Os objectos operam em conjunto para produzir a funcionalidade de um sistema
 - Estão relacionados / interagem



TIPOS DE REPRESENTAÇÃO

Características das unidades básicas	Programação Monolítica	Programação Estruturada	Programação Orientada por Objectos	Programação Baseada em Agentes
Comportamento	Não modular	Modular	Modular	Modular
Estado	Externo	Externo	Interno	Interno
Evocação	Externa	Externa (Chamada)	Externa (Mensagem)	Interna (Objectivos)

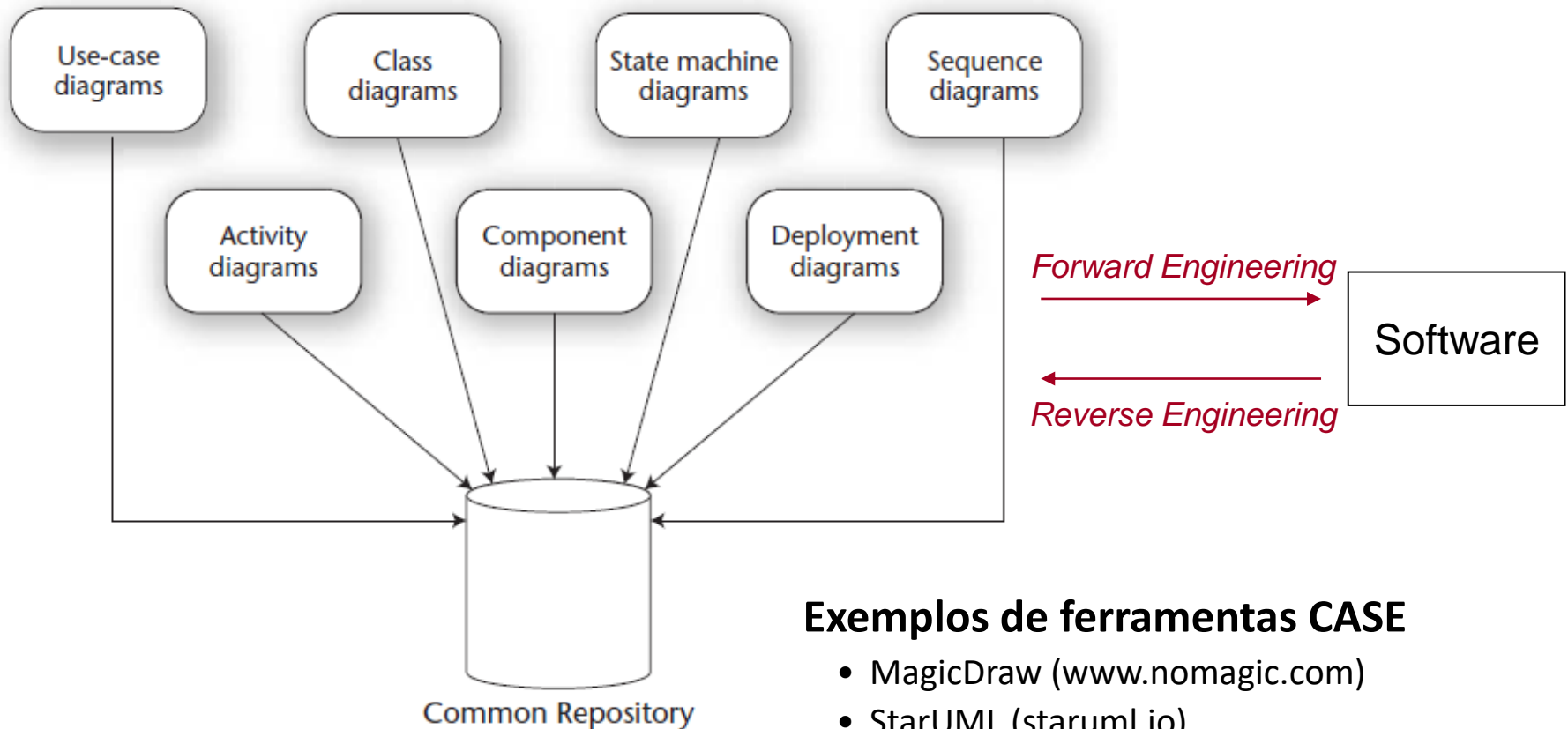
PROCESSO DE DESENVOLVIMENTO



PROCESSO DE DESENVOLVIMENTO

FERRAMENTAS DE MODELAÇÃO DE SOFTWARE

CASE (*Computer-Aided Software Engineering*)



[Eriksson et al., 2004]

Exemplos de ferramentas CASE

- MagicDraw (www.nomagic.com)
- StarUML (staruml.io)
- Enterprise Architect (www.sparxsystems.com)
- Rational Software Architect (www.ibm.com)

BIBLIOGRAFIA

[Pressman, 2003]

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

[Schach, 2010]

S. Schach, *Object-Oriented and Classical Software Engineering*, 8th Edition, McGraw-Hill, 2010.

[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.