

MODELOS DE DINÂMICA

Luís Morgado

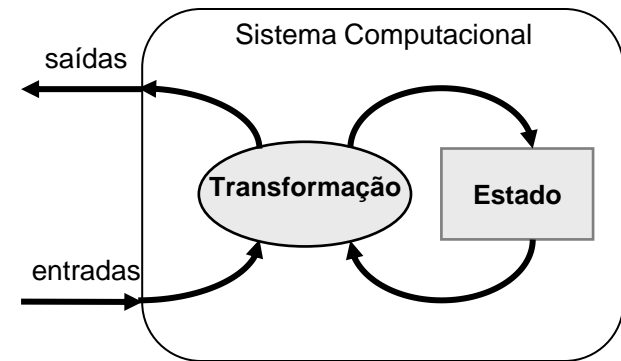
2022

MODELAÇÃO DE UM SISTEMA COMPUTACIONAL

Organização de um sistema

- **Estrutura**

- Organização no **espaço**
- Denota as partes e as relações entre partes de um sistema
- **Estado** (memória)
 - Configuração do sistema relevante para caracterizar a sua dinâmica



- **Dinâmica**

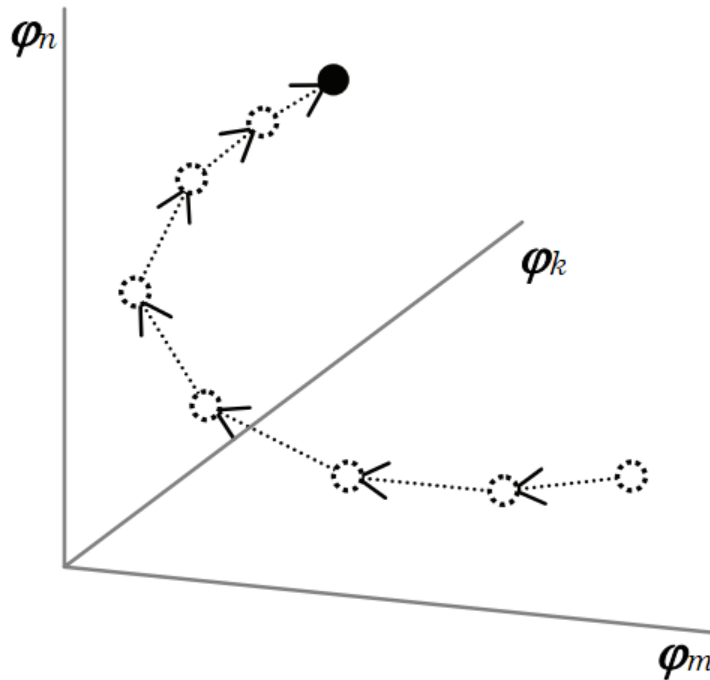
- Organização no **tempo**
- Denota a forma como as partes e as relações entre partes de um sistema evoluem no tempo

- **Comportamento**

- Denota a forma como o sistema **age ou reage** (interage) **perante os estímulos** do ambiente envolvente
- Expressa a estrutura e a dinâmica do sistema

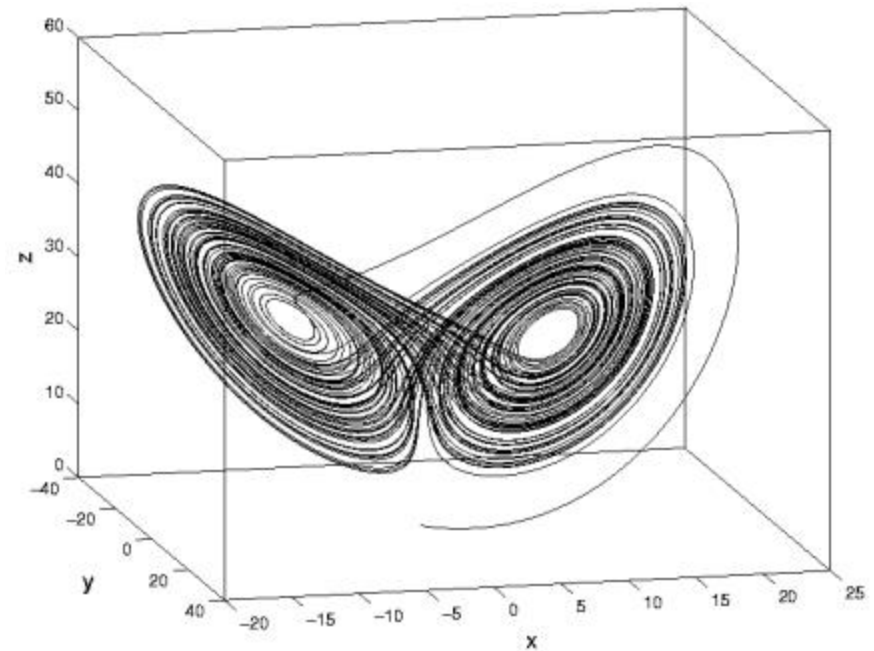
MODELO DE DINÂMICA DE UM SISTEMA

Domínios de valores discretos



Espaço de Estados

Domínios de valores contínuos

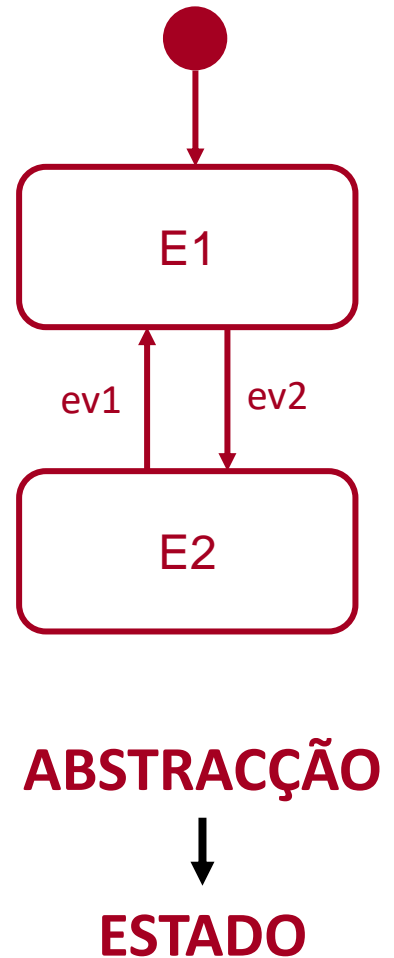
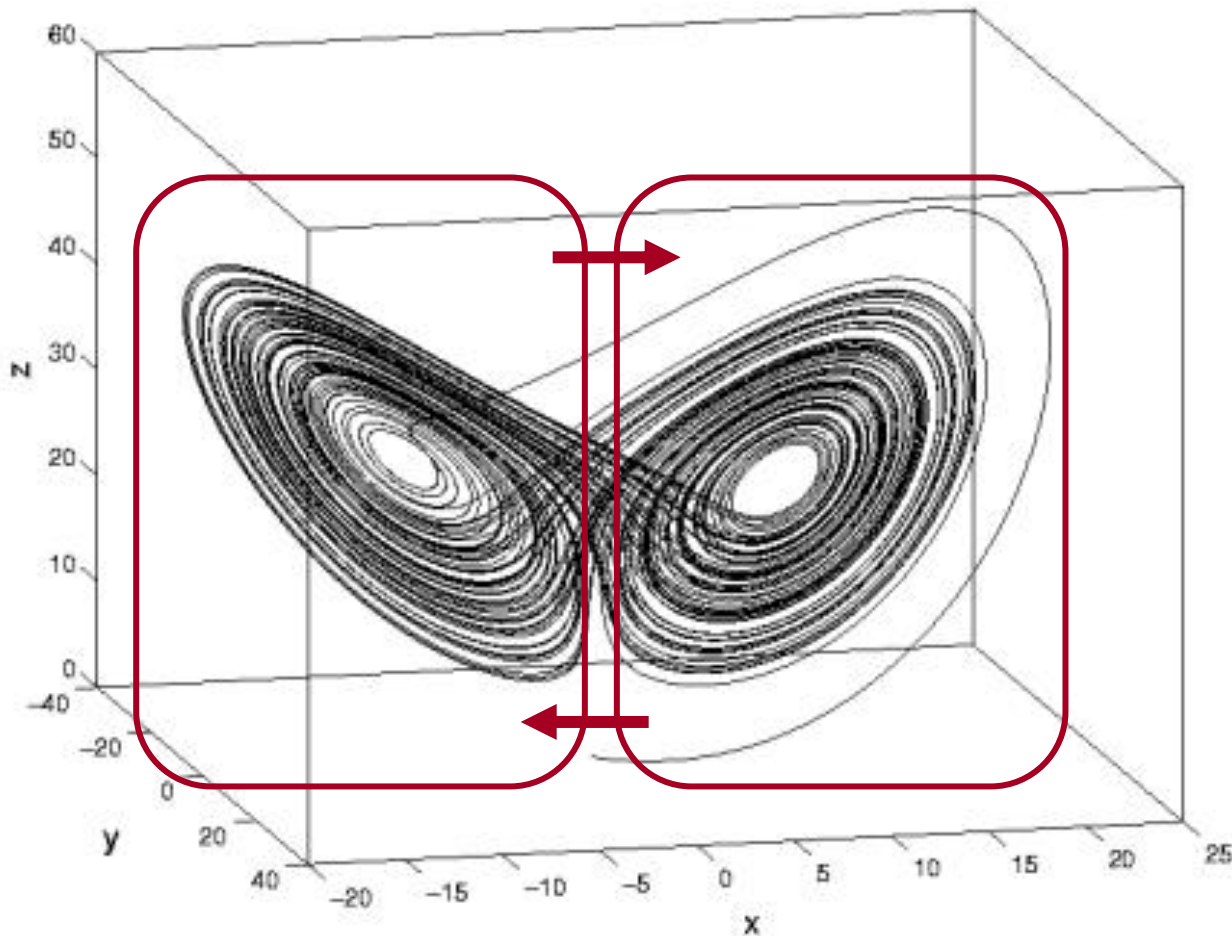


Espaço de Fase

Suporte de descrição do
comportamento de um sistema

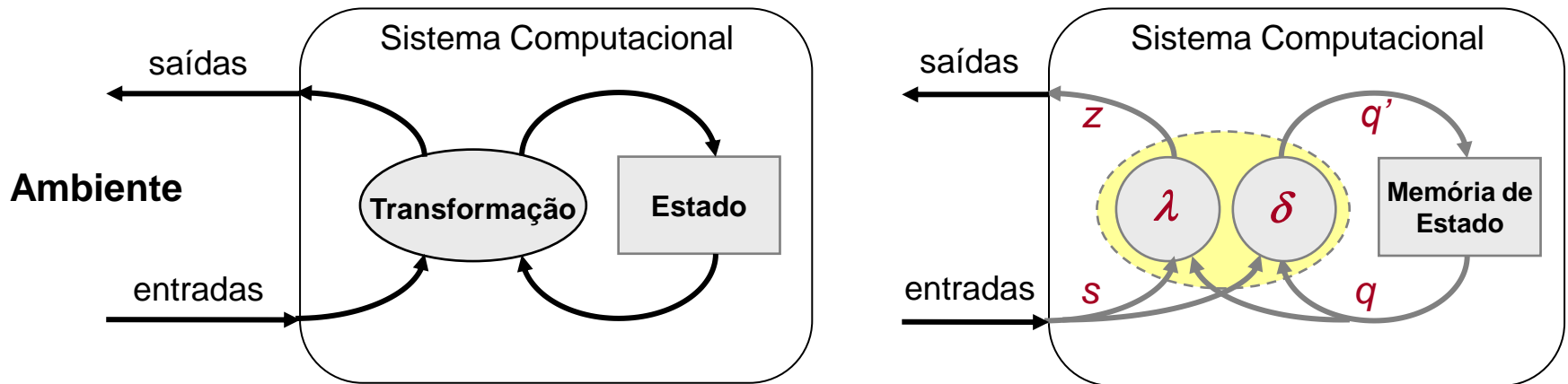
MODELO DE DINÂMICA DE UM SISTEMA

Representação Qualitativa



MODELO DE DINÂMICA DE UM SISTEMA

A **dinâmica** pode ser expressa como uma *função de transformação* que, perante o estado actual e as entradas actuais, produz o estado seguinte e as saídas seguintes.



Esta caracterização de um sistema computacional é **independente da forma concreta como este possa ser implementado em termos físicos**. O suporte físico pode ser, por exemplo, mecânico, electrónico, químico.

MODELO FORMAL DE COMPUTAÇÃO

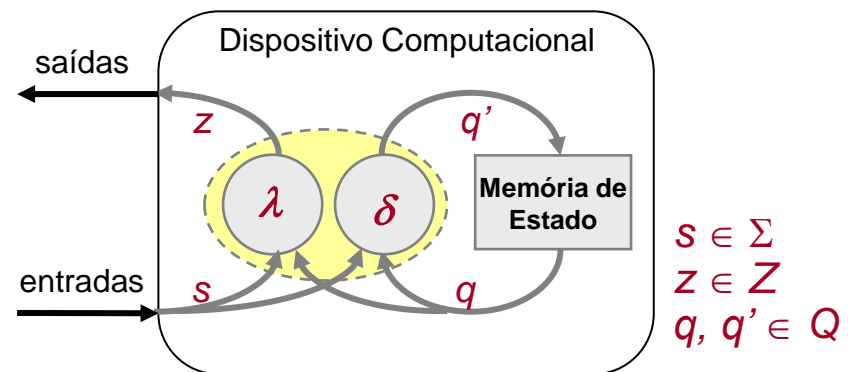
- Entradas e saídas abstraídas em termos dos conjuntos de símbolos que nelas podem ocorrer
 - Esses conjuntos de símbolos são designados *alfabetos*
 - Consideremos um *alfabeto de entrada* Σ e um *alfabeto de saída* Z
- **Estado interno** do sistema descrito em termos de um conjunto de estados possíveis
 - Q
- **Função de transformação** do sistema descrita com base em duas funções distintas δ e λ

- **Função de transição de estado**

- $\delta: Q \times \Sigma \rightarrow Q$

- **Função de saída**

- $\lambda: Q \times \Sigma \rightarrow Z$



MODELO FORMAL DE COMPUTAÇÃO

Este tipo de modelo descreve um mecanismo computacional designado ***Máquina de Estados***

- A sua implementação física implica que o número de estados possíveis seja finito
- ***Máquinas de Estados Finitos***

Duas formulações distintas da função de saída λ :

- ***Máquinas de Mealy***, nas quais a função de saída depende das entradas

$$\lambda : Q \times \Sigma \rightarrow Z$$

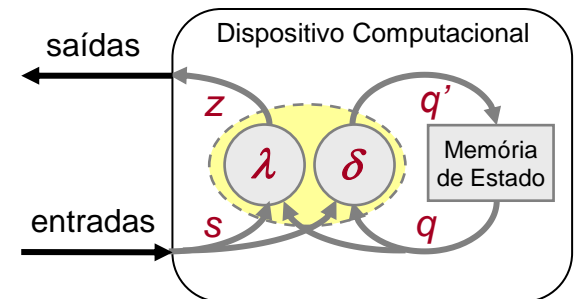
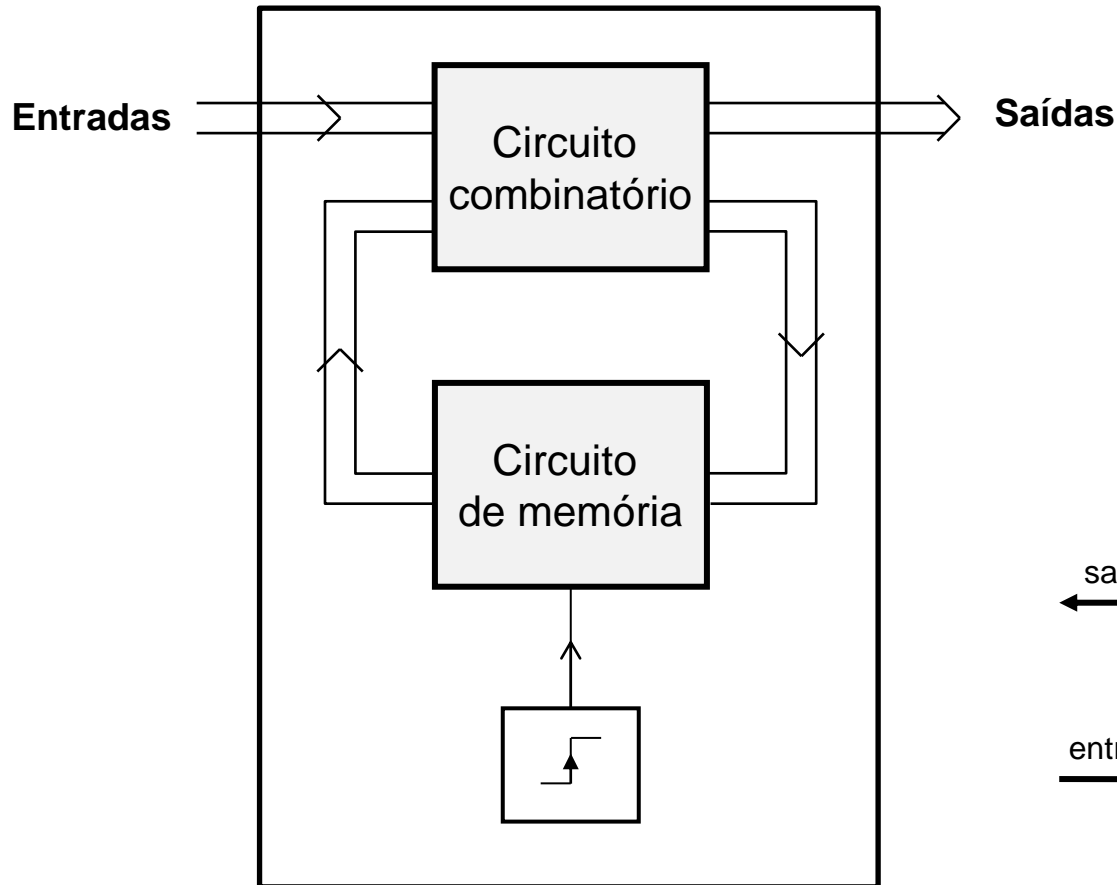
- ***Máquinas de Moore***, nas quais a função de saída não depende das entradas

$$\lambda : Q \rightarrow Z$$

EXEMPLO: SISTEMAS DIGITAIS

Modelo geral de um Sistema Digital

[Peatman, 1981]



EXEMPLO: CASO PRÁTICO

Personagem virtual



Eventos

- Silencio
- Ruido
- Animal
- Fuga
- Fotografia
- Terminar

Acções

- Procurar
- Aproximar
- Observar
- Fotografar

Estados

- Procura
- Inspeção
- Observação
- Registo

Conjunto de símbolos de entrada (o **alfabeto de entrada**):

- $\Sigma = \{ \text{Silencio, Ruido, Animal, Fuga, Fotografia, Terminar} \}$

Conjunto de símbolos de saída (o **alfabeto de saída**):

- $Z = \{ \text{Procurar, Aproximar, Observar, Fotografar} \}$

Conjunto de estados que caracterizam a personagem:

- $Q = \{ \text{Procura, Inspeção, Observação, Registo} \}$

DINÂMICA DA PERSONAGEM

Quando o jogo se inicia a personagem fica a procurar animais. Quando detecta algum ruído aproxima-se e fica a inspeccionar a zona, procurando a fonte do ruído. Quando volta a haver silêncio a personagem continua a procurar animais. Quando detecta um animal a personagem aproxima-se e fica a observar o animal, preparando-se para o registar. Na situação de registo, se o animal fugir a personagem fica a inspeccionar a zona, à procura de uma fonte de ruído. Caso o animal continue, a personagem faz o registo do animal fotografando-o. Na situação de registo, caso ocorra a fuga do animal ou a personagem tenha conseguido uma fotografia do animal, esta fica novamente à procura.

Função de transição de estado

$$\delta: Q \times \Sigma \rightarrow Q$$

Estado	Evento	Novo estado
Procura	Animal	Observação
Procura	Ruido	Inspeção
Procura	Silencio	Procura
Inspeção	Animal	Observação
Inspeção	Ruido	Inspeção
Inspeção	Silencio	Procura
Observação	Fuga	Inspeção
Observação	Animal	Registo
Registo	Animal	Registo
Registo	Fuga	Procura
Registo	Fotografia	Procura

Tabela de transição de estado

Função de saída

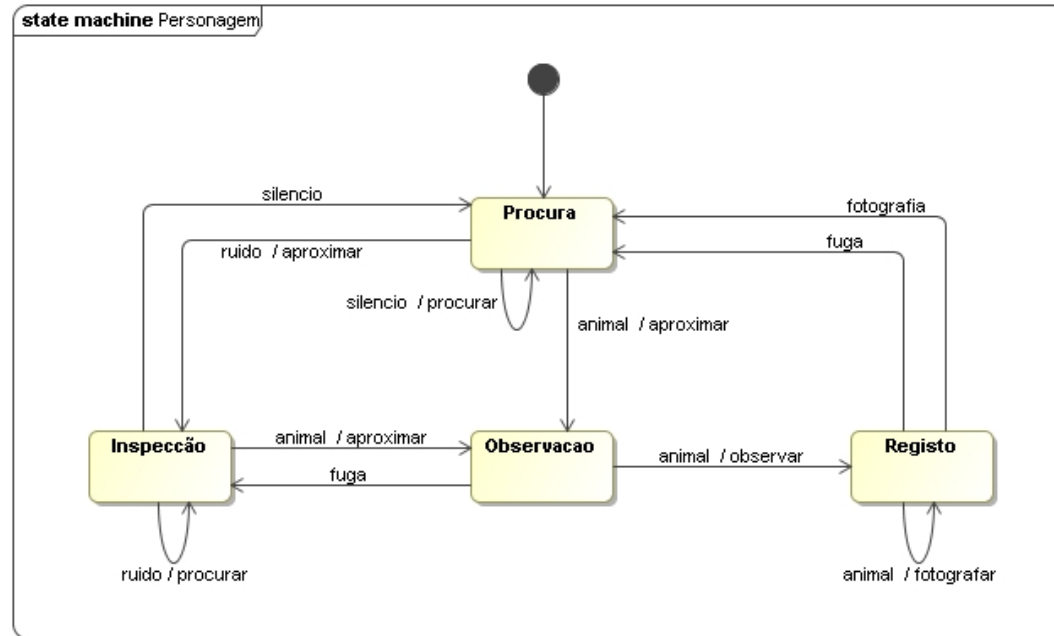
$$\lambda: Q \times \Sigma \rightarrow Z$$

Estado	Evento	Acção
Procura	Animal	Aproximar
Procura	Ruido	Aproximar
Procura	Silencio	Procurar
Inspeção	Animal	Aproximar
Inspeção	Ruido	Procurar
Inspeção	Silencio	
Observação	Fuga	
Observação	Animal	Observar
Registo	Animal	Fotografar
Registo	Fuga	
Registo	Fotografia	

Tabela de acção de estado

DIAGRAMAS DE TRANSIÇÃO DE ESTADO

Representação gráfica
facilita a descrição e
compreensão da
dinâmica de um sistema



Função de transição de estado

$$\delta: Q \times \Sigma \rightarrow Q$$

Estado	Evento	Novo estado
Procura	Animal	Observação
Procura	Ruido	Inspecção
Procura	Silencio	Procura
Inspecção	Animal	Observação
Inspecção	Ruido	Inspecção
Inspecção	Silencio	Procura
Observação	Fuga	Inspecção
Observação	Animal	Registo
Registo	Animal	Registo
Registo	Fuga	Procura
Registo	Fotografia	Procura

Função de saída

$$\delta: Q \times \Sigma \rightarrow Z$$

Estado	Evento	Ação
Procura	Animal	Aproximar
Procura	Ruido	Aproximar
Procura	Silencio	Procurar
Inspecção	Animal	Aproximar
Inspecção	Ruido	Procurar
Inspecção	Silencio	
Observação	Fuga	
Observação	Animal	Observar
Registo	Animal	Fotografar
Registo	Fuga	
Registo	Fotografia	

BIBLIOGRAFIA

[Pressman, 2003]

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

[Peatman, 1981]

J. Peatman, *The Design of Digital Systems*, McGraw-Hill, 1981.

[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.

[Martin, 2003]

J. Martin, *Introduction to Languages and the Theory of Computation*, McGraw-Hill, 2003.

[Sipser, 2005]

M. Sipser, *Introduction to the Theory of Computation*, Thomson, 2005.