# CH8.Verifying Models

School of Computer Science
Prof. Euijong Lee

# Introduction

- **Functional model**
  - **to represent how the business system will behave**
  - **Use case diagram, Activity diagram**
- **Structural model**
  - **representing the objects that are created and used by a business system**
  - **Class diagram, object diagram, package diagram**
- **Structural model is evolving over time**
  - **In analysis phase (conceptual model or doamin model)**
    - shows the logical organization of the objects
  - **In design phase (design model)**
    - reflects how the objects will be organized in database and files.
- **Behavioral model**
  - **to represent the internal behaviors of an object**
  - **to represent the interactions between objects**
  - **Sequence diagram, State machine diagram**

# Objectives

- **Verifying the analysis models**
- **Consistency**
  - between functional model and structural model
  - between functional model and behavioral model
  - between behavioral model and structural model
- **Documenting the information in development life-cycle**

# Before Design Phase

- **Inspect Analysis and Design**

[17] The material in this section has been adapted from E. Yourdon, *Modern Structured Analysis*

**"There is nothing in the programming field more despicable than an undocumented program."**

**"If you haven't spent at least a month working on the same program — working 16 hours a day, dreaming about it during the remaining 8 hours of restless sleep, working several nights straight through trying to eliminate that 'one last bug' from the program — then you haven't really written a complicated computer program."**


에드워드 유르동 (Edward Yourdon)

소프트웨어 엔지니어

영어에서 번역됨 - Edward Nash Yourdon은 미국 소프트웨어 엔지니어, 컴퓨터 컨설턴트, 저자 및 강사, 소프트웨어 엔지니어링 방법론 개척자입니다.
위키백과(영어)

원래 설명 보기

**출생:** 1944년 4월 30일

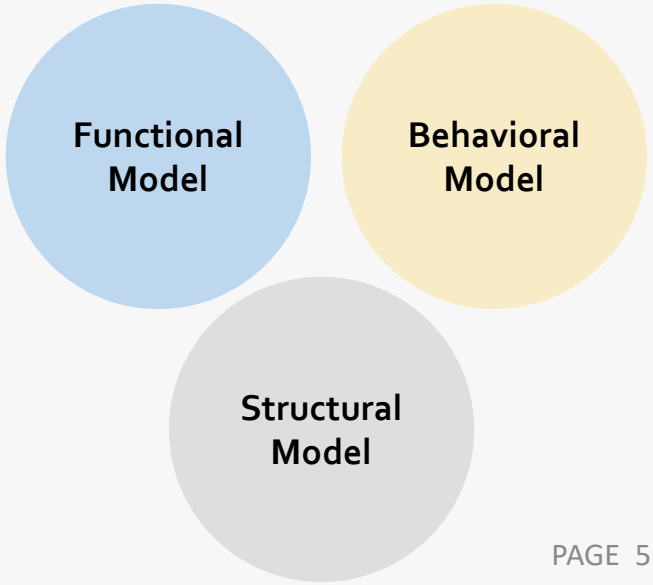
**사망 정보:** 2016년 1월 20일

**학력:** 매사추세츠 공과대학교

**알려진 분야:** 구조적 프로그래밍, SSADM

# Balancing the models

- **The process of ensuring the consistency among analysis models, functional, structural, and behavioral models**
- **To ensure that the current set of analysis models faithfully represent the problem domain under consideration**
- **Intention**
  - Ensuring the consistency among different models
  - Verifying and validating the intersections of the analysis models using a set of rules



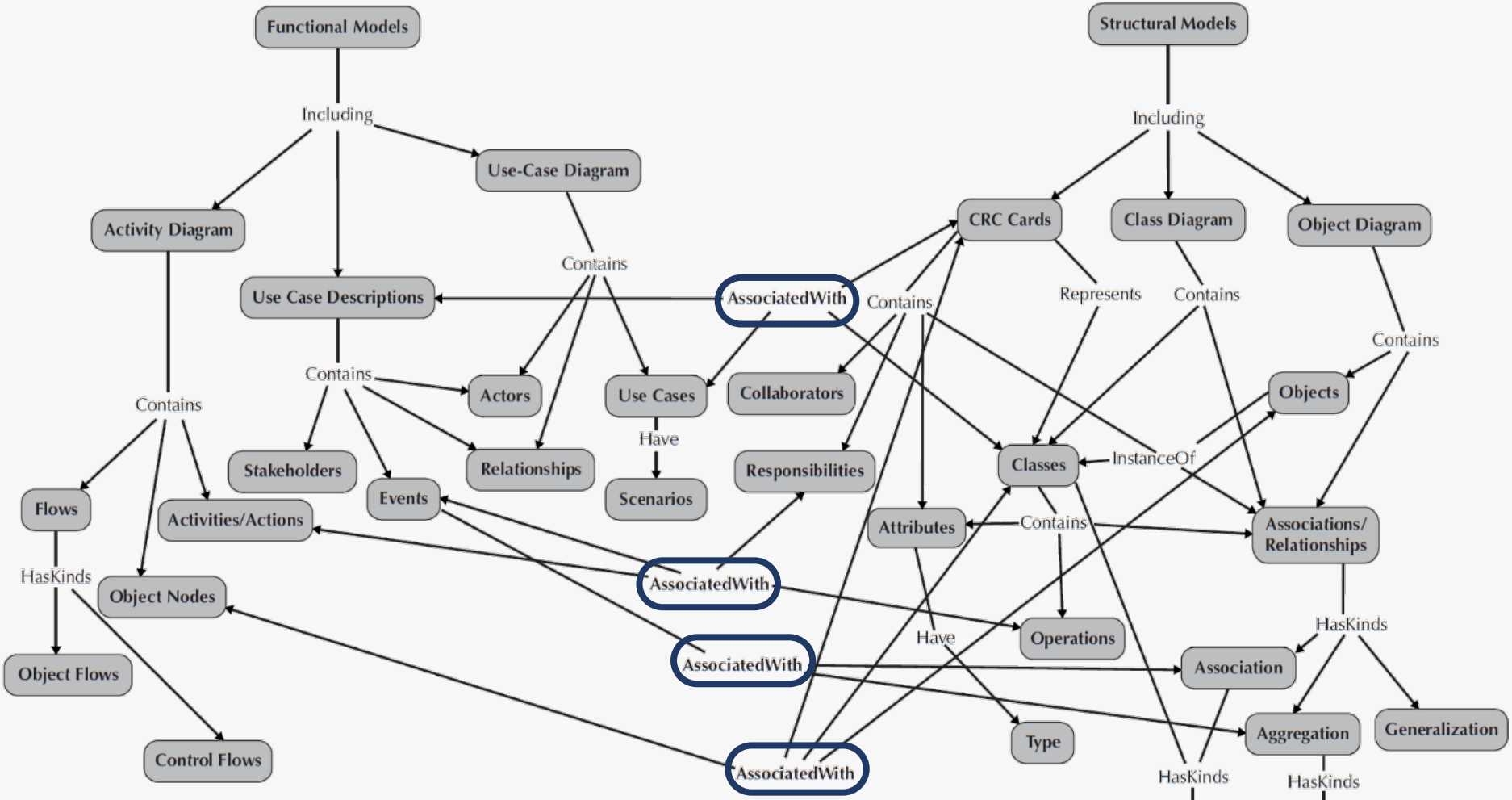
# Balancing Functional and Structural models

- **Ensure that two sets of models are consistent**
  - The activity diagrams, use-case descriptions, and use-case diagrams must agree with the CRC cards and class diagrams that represent the evolving model of the problem domain.

- **4 sets of associations between models**
  - Use Case Description vs. CRC cards, Class
  - Activities / Actions vs. Class responsibility, Class operations
  - Events vs. Association, Aggregation relationship
  - Object node vs. CRC cards, Class, Object

# Balancing Functional and Structural models

- Relationship between Functional and Structural models






Source: Alan Dennis, et al., Systems Analysis and Design with UML, 4th ed., 2013

# Balancing Functional and Structural models

- **Rules to verify the models : Check the association points**

1. Every class on a diagram and every CRC card must be associated with at least one use case, and vice versa.

**: Example, Medical Appointment System**

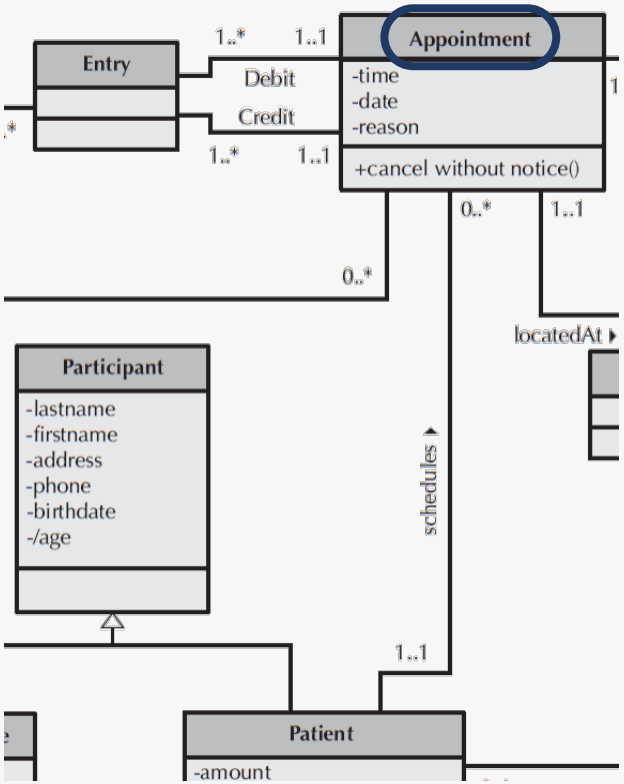
**Relationships:**

**Generalization (a-kind-of):** Participant

**Aggregation (has-parts):**

**Other Associations:** Appointment, Medical History

| | | |
|---|---|---|
| Use-Case Name: Make Old Patient Appt | ID: 2 | Impor |
| Primary Actor: Old Patient | Use Case Type: Detail, Essential | |
| Stakeholders and Interests:<br>Old patient - wants to make, change, or cancel an appointment<br>Doctor - wants to ensure patient's needs are met in a timely manner | | |
| Brief Description: This use case describes how we make an appointment as well as changing or canceling an appointment for a previously seen patient. | | |
| Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment | | |

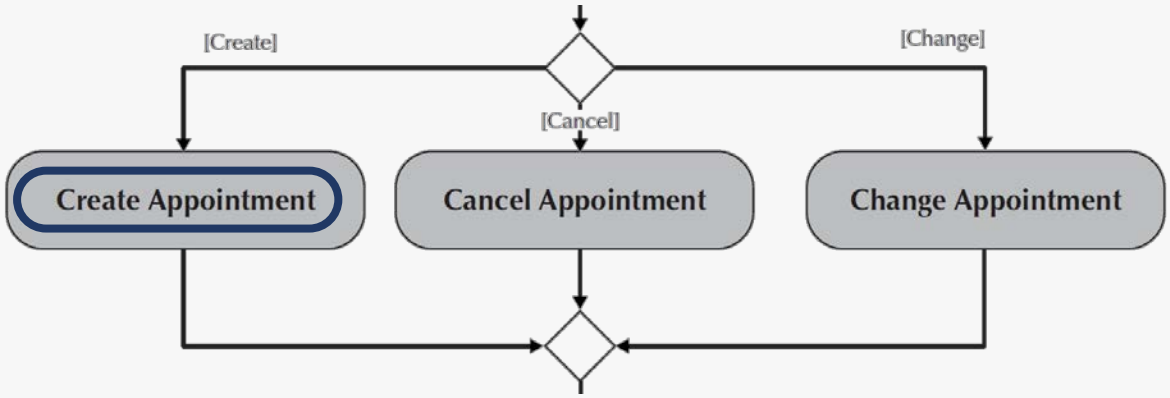


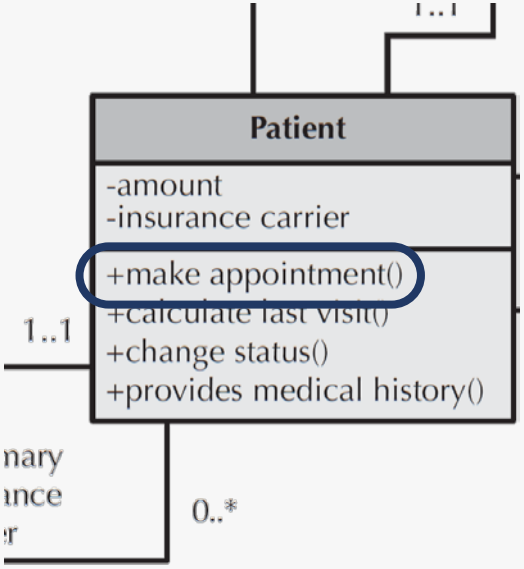
# Balancing Functional and Structural models

- **Rules to verify the models : Check the association points**

  2. Every activity or action contained in an activity diagram and every event contained in a use-case description should be related to one or more responsibilities on a CRC card and one or more operations in a class on a class diagram, and vice versa.

  **: Example, Medical Appointment System**



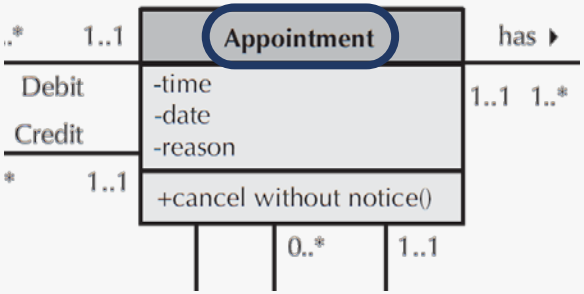



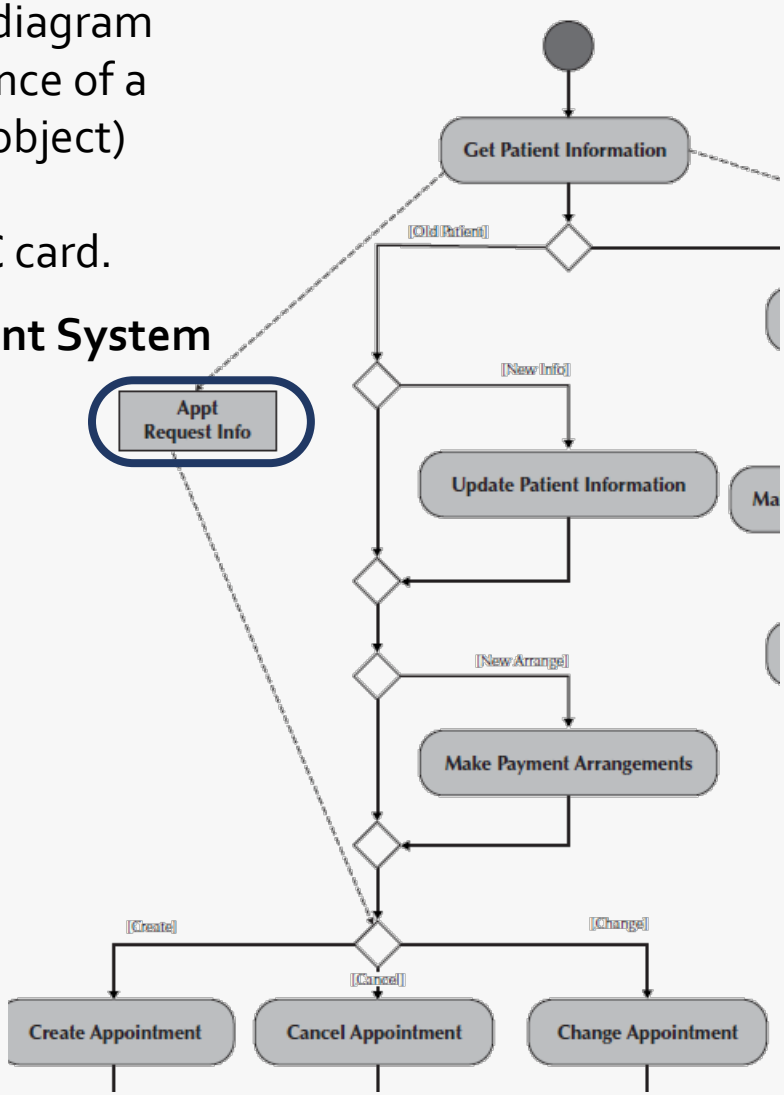
# Balancing Functional and Structural models

- **Rules to verify the models : Check the association points**

3. Every object node on an activity diagram must be associated with an instance of a class on a class diagram,(i.e., an object) and a CRC card, or an attribute contained in a class and on a CRC card.

**: Example, Medical Appointment System**



# Balancing Functional and Structural models

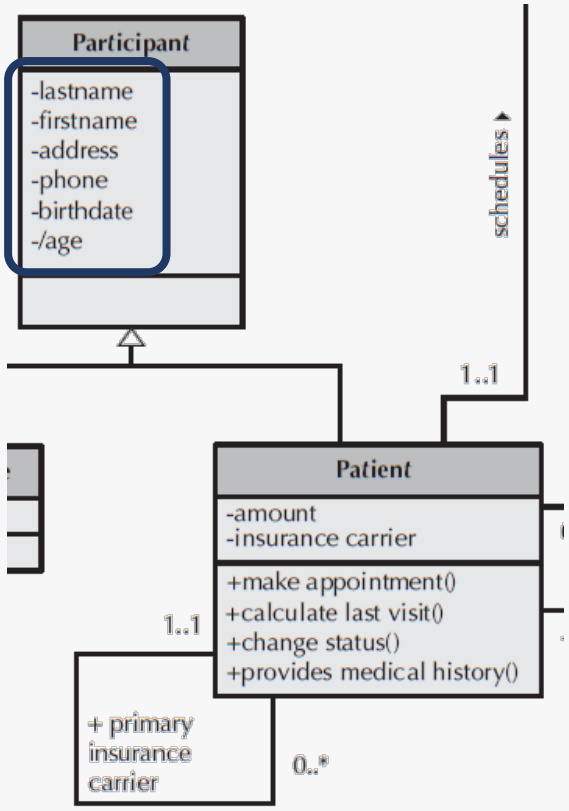
- **Rules to verify the models : Check the association points**

  4. Every attribute and association/aggregation relationship contained on a CRC card(and connected to a class on a class diagram) should be related to the subject or object of an event in a use-case description

  **: Example, Medical Appointment System**

Normal Flow of Events:

1. The Patient contacts the office regarding an appointment.
2. The Patient provides the Receptionist with his or her name and address.
3. If the Patient's information has changed
   Execute the Update Patient Information use case.
4. If the Patient's payment arrangements has changed
   Execute the Make Payments Arrangements use case.



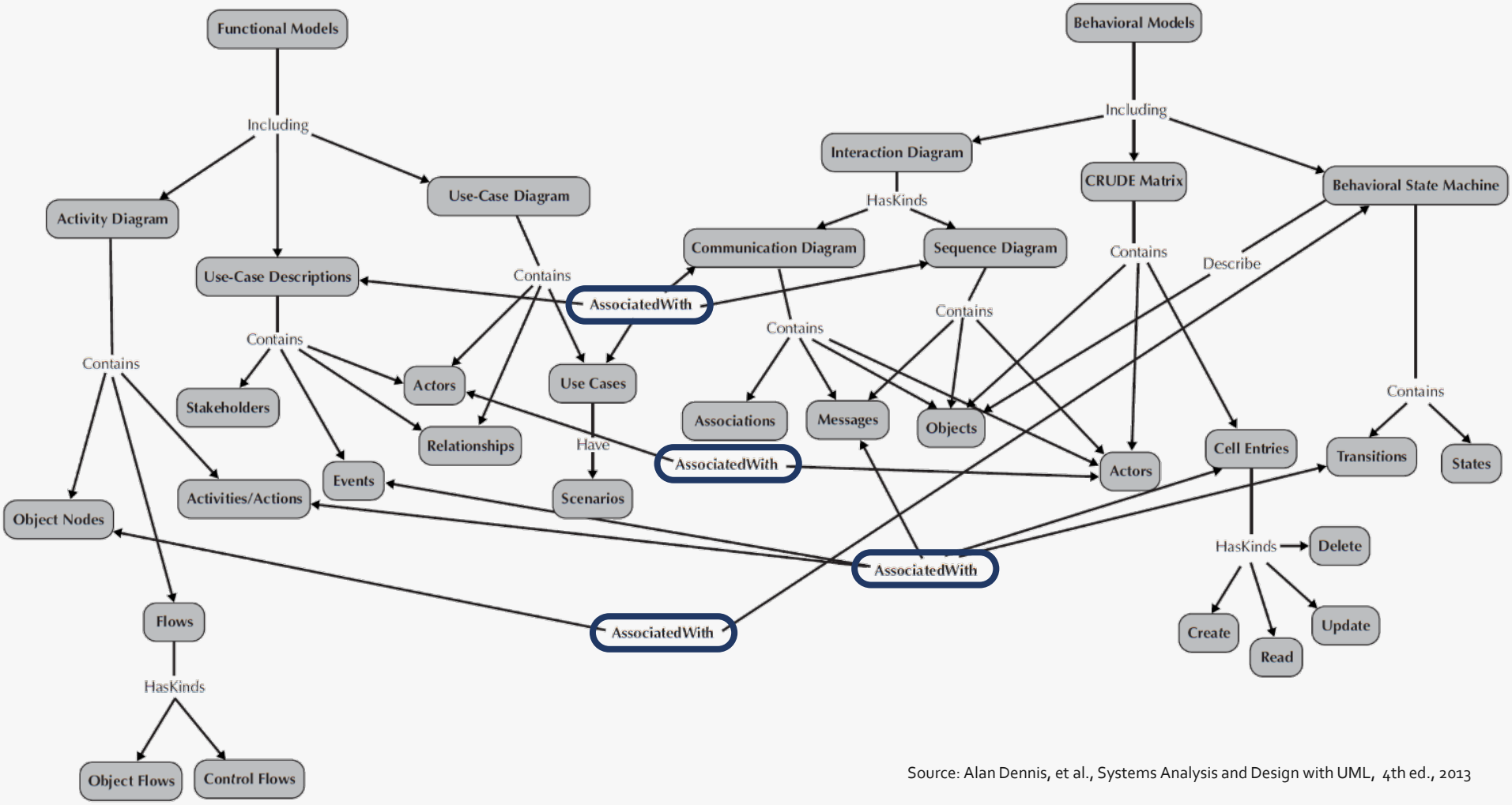
# Balancing Functional and Behavioral models

- **Ensure that two sets of models are consistent**
  - The activity diagrams, use-case descriptions, and use-case diagrams must agree with the sequence diagrams, communication diagrams, behavioral state machines, and CRUD matrix.

- **4 sets of associations between models**
  - Use Case Description vs. Sequence diagram
  - Actors vs. Actors in Seq. diagram, Comm. diagram
  - Events, Activities vs. Messages, Transitions
  - Object node vs. State machine

# Balancing Functional and Behavioral models

- Relationship between Functional and Behavioral models



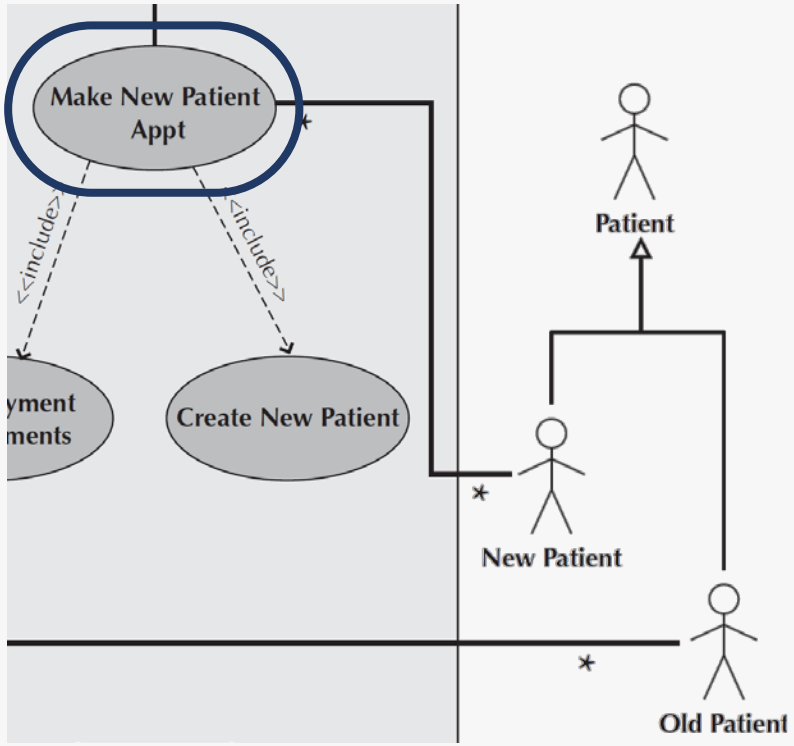

Source: Alan Dennis, et al., Systems Analysis and Design with UML, 4th ed., 2013

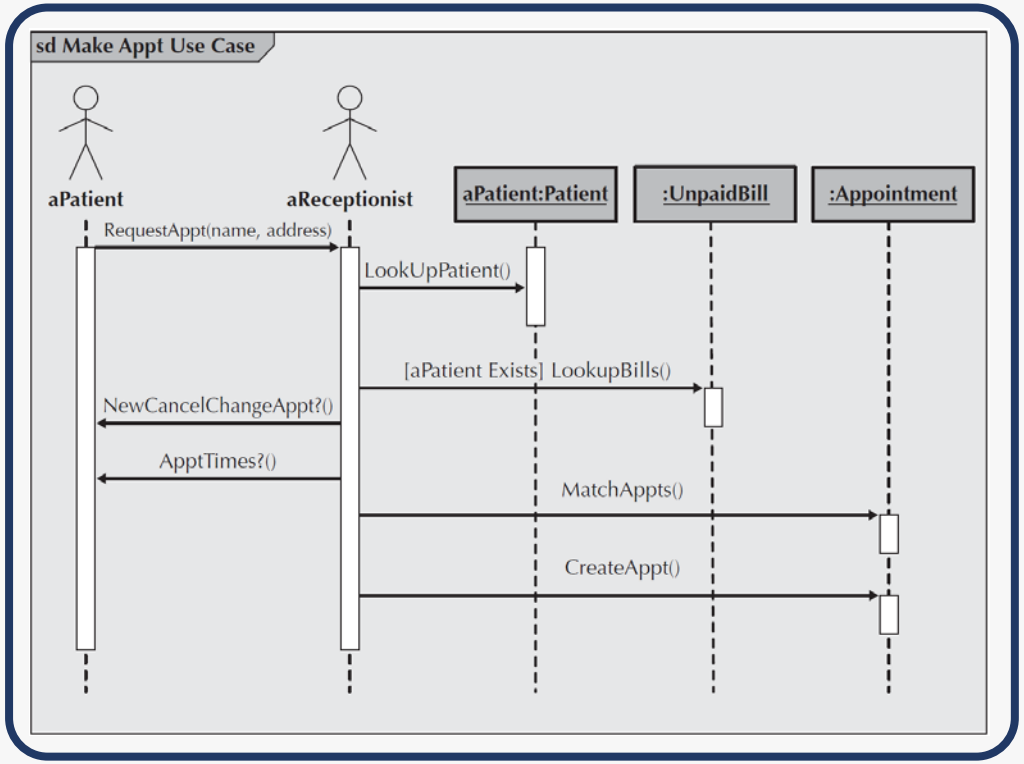
# Balancing Functional and Behavioral models

- **Rules to verify the models : Check the association points**

1. The seq. and comm. diagrams must be associated with a use case on the use-case diagram and a use-case description.

**: Example, Medical Appointment System**



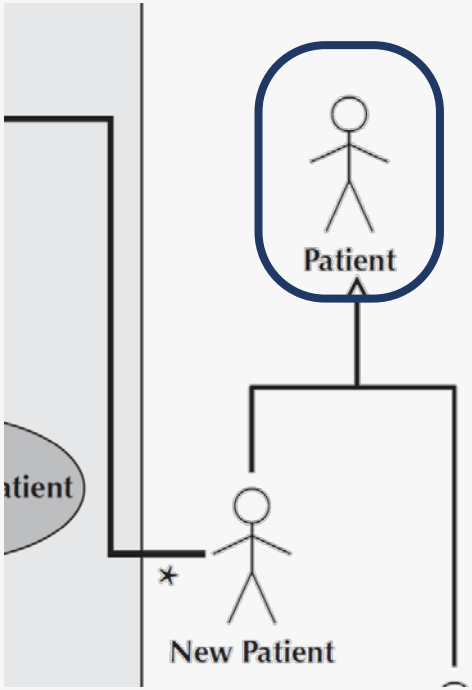



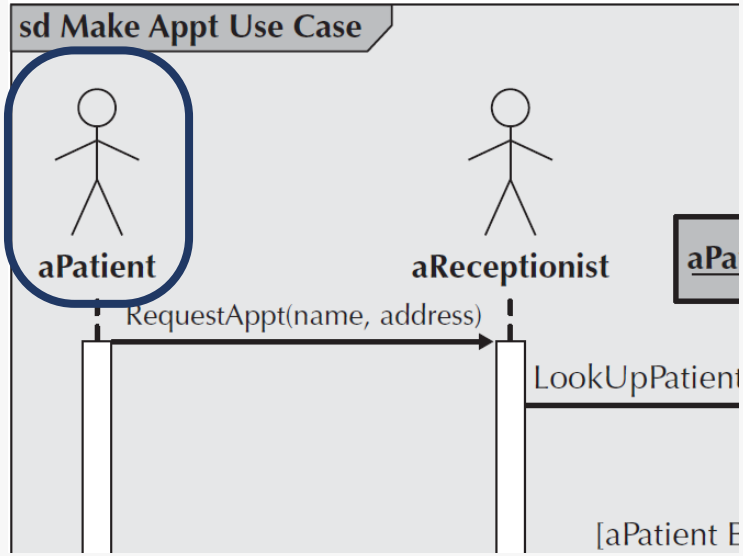
# Balancing Functional and Behavioral models

- **Rules to verify the models : Check the association points**

2. The actors on seq. diagrams, comm. diagrams must be associated with actors on the use-case diagram or referenced in the use-case description, and vice versa.

**: Example, Medical Appointment System**



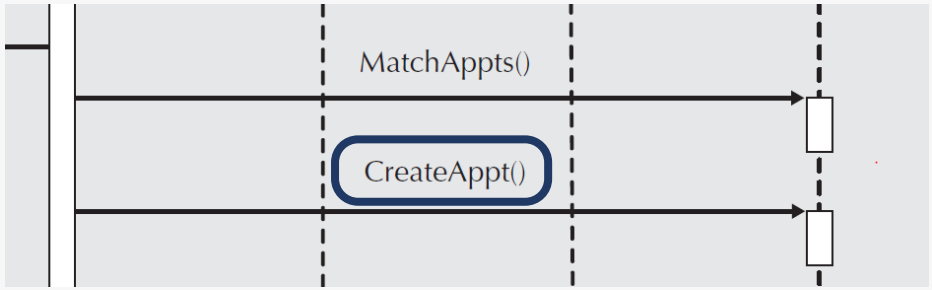



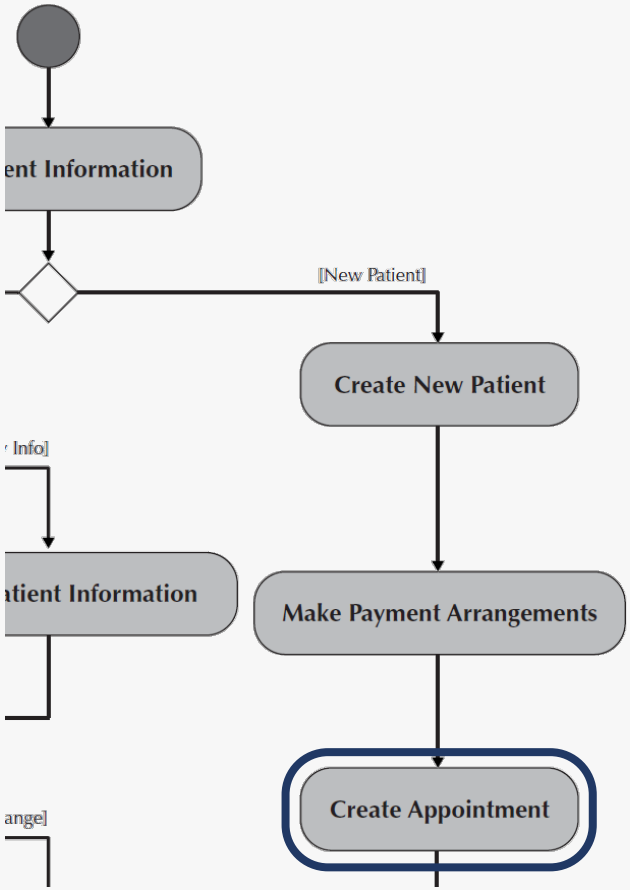
# Balancing Functional and Behavioral models

- **Rules to verify the models : Check the association points**

3. The messages on seq. and comm. diagrams, transitions on behavioral state machines must be related to activities and actions on an activity diagram and events listed in a use-case description, and vice versa

   **: Example, Medical Appointment System**



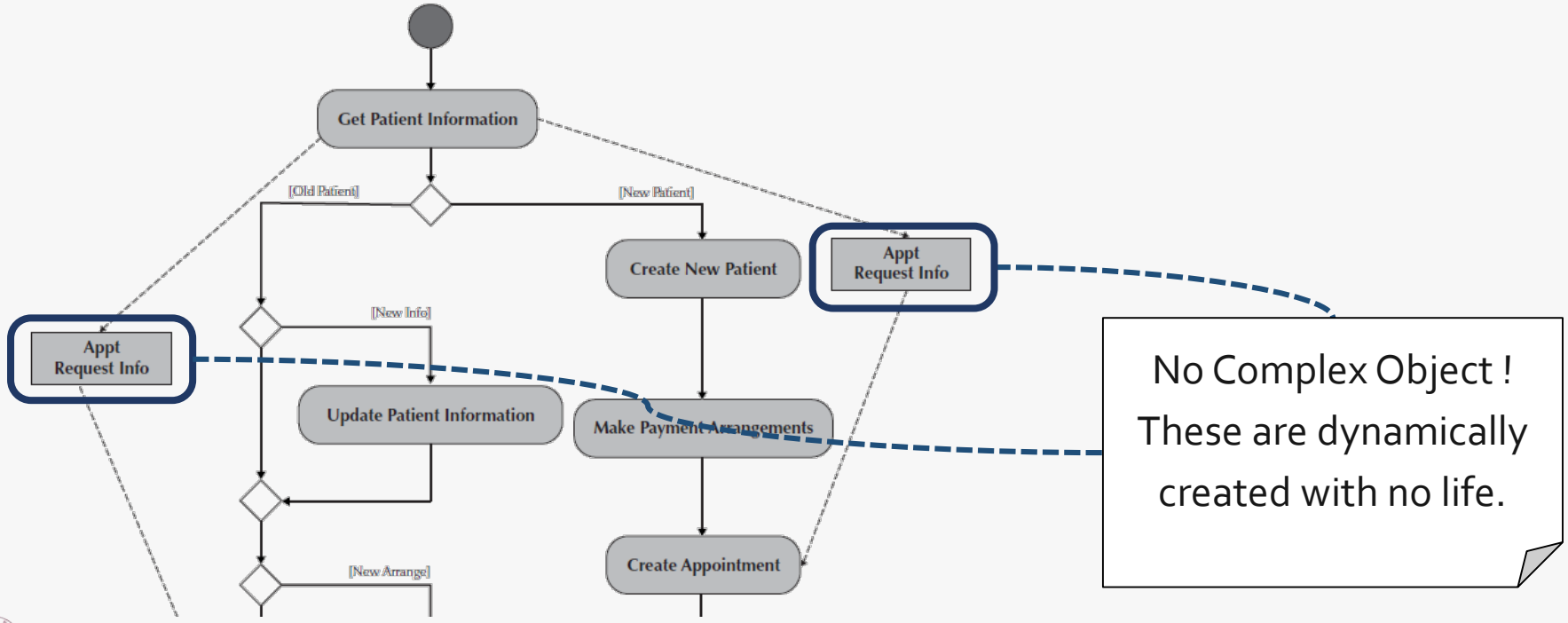



# Balancing Functional and Behavioral models

- **Rules to verify the models : Check the association points**

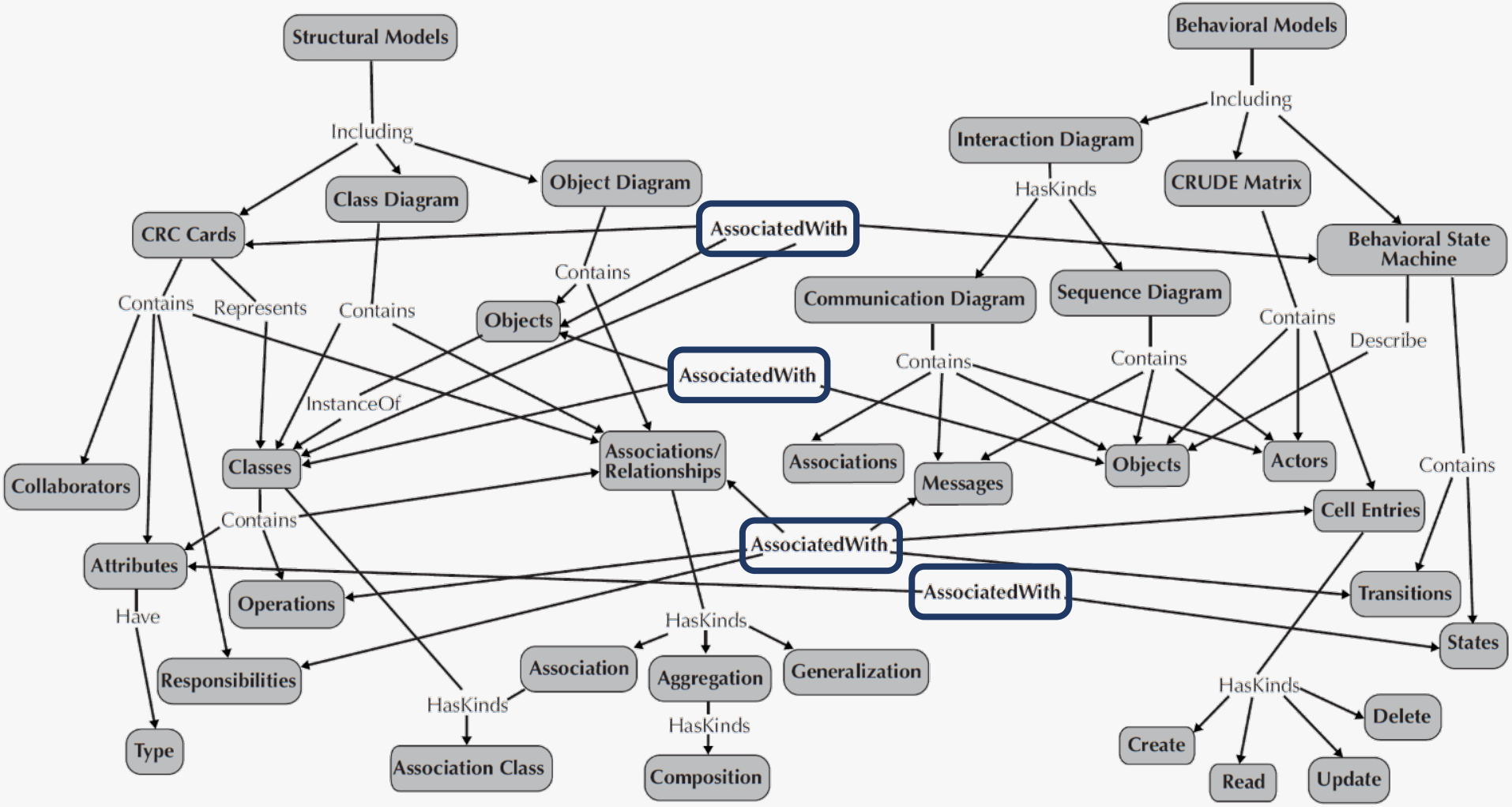
4. All complex objects represented by an object node in an activity diagram must have a behavioral state machine that represents the object's lifecycle, and vice versa.

**: Example, Medical Appointment System**



# Balancing Structural and Behavioral models

- Relationship between Structural and Behavioral models



# Balancing Structural and Behavioral models

- **Rules to verify the models : Check the association points**

1. The **state machines** must be associated with **instances (objects) of classes** on a class diagram and with a CRC card that represent the class of the instance.

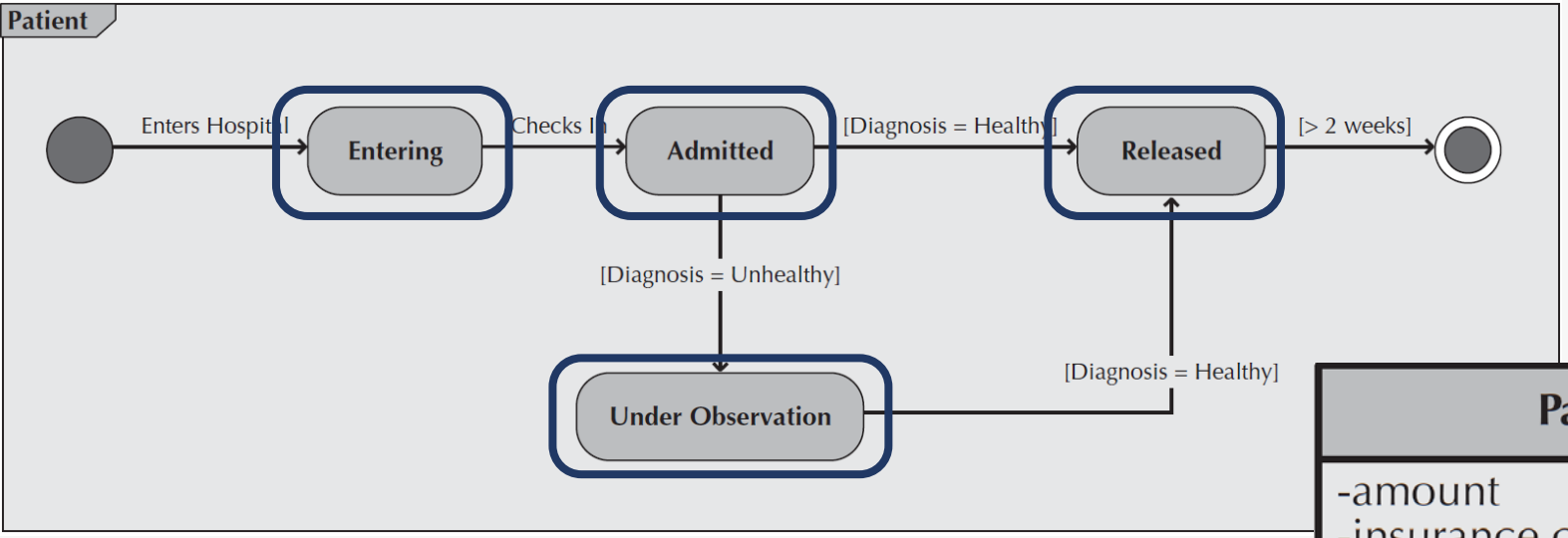
2. **Comm. and seq. diagrams contain objects** that must be an **instantiation of a class** that is represented by a CRC card and is located on a class diagram.

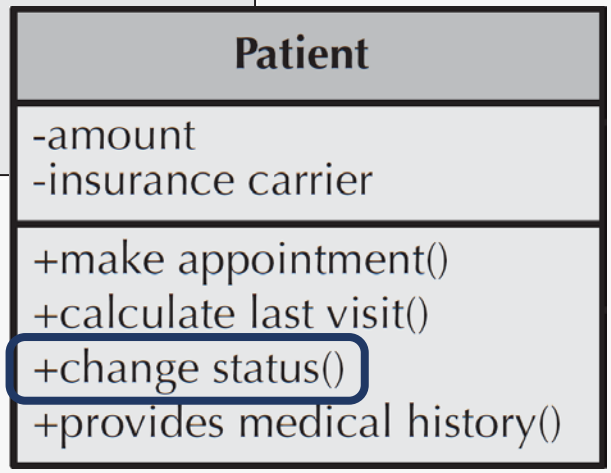
3. **Messages** contained on the seq. and comm. diagrams, transitions on state machines must be associated with responsibilities and associations on CRC cards and **operations in classes** and associations connected to the classes on class diagrams.

# Balancing Structural and Behavioral models

- **Rules to verify the models : Check the association points**

  4. The **states** in a state machine must be associated with **different values of an attribute** or set of attributes that describe an object.



# Summary and Discussion

- **When evolving analysis into design models,**
  - it is important to review the analysis models , then add system environment information.
- **Consistency in 4 association rules**
  - derived from the concept map for functional, structural, and behavioral models
  - between functional model and structural model
  - between functional model and behavioral model
  - between behavioral model and structural model
- **What are the relationships among the functional, structural, and behavioral models in modeling process?**