Chapter 1

Reactive Systems

Questions for rehearsal

1. *(Basic concepts)* Define the following concepts:

   - Reactive system
   - Transformational system

   Give examples of these systems from your environment.

Questions for discussion

2. *(Reactivity)* Which characteristics of reactive systems (figure **??**) are present in the teabag boxing controller?

3. *(Reactivity)* Which characteristics of reactive systems (figure **??**) are present in the bug fixing database?

Chapter 2

Questions for rehearsal

1. *(Basic concepts)* Define the following concepts:
   - Subject domain
   - Lexical item
   - Connection domain
2. *(Subject domain)* Can the subject domain of a single interaction be different from the subject domain of the entire system?
3. *(Connection domain and subject domain)* Can a connection domain of a system be part of the subject domain of that same system?
4. *(Overlapping subject domains)* Can the subject domains of the messages sent and received by a system overlap? If yes, give an example; if no, argue why.
5. *(Subject domain part of the system)* Explain why a system can contain part of its subject domain.
6. *(Classification of functionality)* Reactive systems may offer a mix of two or three of the basic kinds of functions information provision, direction and manipulation. Describe the mix for each of the three example systems:
   (a) The tea bag boxing system,
   (b) The bug fixing support system,
   (c) The chemical tracking system.

Questions for discussion

7. *(Subject domain and system functions)* The subject domain of a reactive system changes when we add or delete functions of the system. Explain this and give an example.
8. *(Subject and connection domains)* Identify the subject domains of the following systems, as well as the communication channel by which messages travel between the system and the subject domains. There are usually several possible valid answers, so motivate the answer you have chosen.

(a) An elevator control system.

(b) A cruise control system in a car.

(c) A personnel information system.

(d) A word processing system.

(e) The bank where you have your current account.

(f) The account database of your bank.

(g) An on-line bookstore.

(h) A supermarket.

(i) An EDI system that connects a supermarket with suppliers of diary products.

(j) A web search engine.

9. *(Subject domain of system and of part of system)* Can the subject domain of company be the same as that of an information system owned by the company?

10. *(Subject domain of system and of part of system)* A workflow management system (WFMS) monitors and controls the flow of work through an organization. It allocates work to people, makes the relevant applications available and monitors the progress of a task through the organization. A WFMS is typically supported by a database system (DBS). Consider the WFMS of an insurance company that handles damage claims and the underlying DBS of insurances. What is the subject domain of

    (a) the DBS,

    (b) the WFMS, and

    (c) the insurance company?

11. *(Reflexive subject domains)* Give an example of a system that can be part of its own subject domain.

12. *(Existence "in" an SuD)* If we open up a computer, we will see physical components such as a printed circuit board and a ventilator. We will not see the lexical items stored in the computer. So what does it mean to say that a lexical item exists "in" a computer?

Chapter 3

Questions for rehearsal

1. *(Basic concepts)* Define the following concepts:
   - Event and action.
   - External event and temporal event.
   - Named external event and condition change event.
   - Stimulus and response.
   - Observer and actor.
2. *(Event recognition a response computation)* Explain why we need event recognition and response computation in a reactive system.
3. *(Temporal events)* Are temporal events observable? Explain.

Questions for discussion

4. *(Connecting a reactive system to its subject domain)* A reactive system is linked to its subject domain by a connection domain. It is part of a loop which runs from events through the connection domain to the system and then back again from the system through the connection domain to the desired action. In each of the following examples, identify
   (i) The subject domain event being responded to,
   (ii) the subject domain entity (or entities) in whose life the event occurs,
   (iii) the observer used to pass information about the event to the system,
   (iv) the desired subject domain action,
   (v) the subject domain entity (or entities) in whose life the action will occur, and
   (vi) the actor used to pass the information about the desired action to the subject domain.
   (vii) Write down any assumptions that you are making about the subject and connection domains, if you think these are relevant.

   Only mention observable events and realizable actions.

(a) System: An elevator control system. Stimulus: Reception of a up_request signal from floor 1. Response: Send motor_down signal to motor 3.

(b) System: A word processor. Stimulus: Reception of delete signal. Response: The cursor moves left, erasing the character at the position that it moves to.

(c) System: A bank account database. Stimulus: Reception of a the command update account 345 with 567 Euro from the line connected to terminal 999. Response: Add 567 to the balance field of the record that holds account 345 and send a confirmation to a line connected to terminal 999.

(d) System: An EDI system connecting the information system of a supermarket with the production information system of a supplier. Stimulus: The EDI system receives a signal that the number of milk packages in store has fallen below 100. Response: The EDI system issues an order to deliver 300 milk packages.

5. *(Observability)* The following systems respond to events in their subject domain. For each system, mention an event that you would like the system to respond to, but that is nevertheless unobservable by the system; and mention an event that the system is able to respond to instead, plus an observer of the event.

   (a) A car cruise control.

   (b) A library document circulation information system.

   (c) An elevator control.

   (d) A workflow system.

   (e) An email system.

6. *(Realizability)* The following systems cause actions in their subject domain. For each system, mention an action that you would like the system to cause, but that is nevertheless unrealizable by the system; and mention an event that the system can cause instead, plus an actor that realizes this.

   (a) A patient monitoring system used in an intensive care unit, showing vital data of patients to a nurse.

   (b) A library document circulation information system.

   (c) The information system of a supermarket that maintains current stock from minute to minute and that is used to take care that there are always sufficient items in stock.

   (d) A web shop, i.e. a web page with product catalog and transaction facilities.

   (e) An email system.

7. *(Actors)* "An actor (or observer) cannot be a subject domain entity." Give an argument why this is true (in one sense) and an argument why it is false (in another sense).

8. *(Events and stimuli)* Is a stimulus always caused by an external event?

Chapter 4

Questions for rehearsal

1. *(Basic concepts)* Define the following concepts:
   - Requirement
   - Constraint
   - Function
   - Service
   - Feature
2. *(Assumption-requirement specifications)* What is the structure of an assumption-requirement specification? Give an example.
3. *(Operational specifications)* What is the structure of an operational specification?
4. *(Stimulus-response specifications)* Explain how stimulus-response specifications can be used in an operational specification.

Questions for discussion

5. *(Functional aspects)* Consider a coffee machine: We can put in ground coffee and water and switch it on. When switched on, it will heat the water and drip coffee in a pot. We can take out the pot at any moment to pour coffee from it. Separate the functional properties of the coffee machine into three: functions, behavior and communication.
6. *(Assumption-requirement specifications)* Suppose that in a particular situation the assumptions of an assumption-requirements specification are not satisfied. The system manufacturer argues as follows:

   > "Your specification has the form
   >
   > (*) If assumption then requirement.
   >
   > Now, because your usage environment does not satisfy the assumption, (*) is vacuously true. Therefore, my system satisfies your specification (*)."

   How would you counter this argument? Explain your answer.

7. *(Operationalizability)* State for each of the following requirements whether or not it can be put in operational form. Explain your answer.

   (i) The system shall register borrow, extend and return events.

   (ii) The system shall respond to any update request within 2 seconds.

   (iii) The system shall be easy to learn.

   (iv) The system shall be interoperable.

   (v) The system shall only use well-understood concepts at its user-interface.

8. *(Operationalization and test)* Is an operationalized requirement the same thing as a test for the finished product? Explain your answer.

9. *(Operationalization and design decisions)* Suppose we have refined a set R of non-operationalized requirements into a set R' of operationalized requirements. Does R' give the designer sufficient information to design a system decomposition such that the system will satisfy R'? If so, explain why. If not, explain why we had to go to the trouble of refining R into R'.

Chapter 5

Questions for rehearsal

1. *(Responsibilities)*
   (a) What is the difference between a function and a responsibility of a product?
   (b) What is the relationship between product responsibilities and environment goals?

Exercises

2. *(Levels of problem solving)* Consider the Chemical Tracking System described in chapter 1 (page **??**).
   (a) Draw the business goal tree of the company.
   (b) Identify the activities to be supported by the Chemical Tracking System.
   (c) What is the purpose of the Chemical Tracking System?
   (d) What are its responsibilities?
   (e) Are there any exclusions you can identify rightaway?
3. *(Mission statement)* Write a mission statement for the teabag boxing controller. (See page **??** for the case description.)
4. *(Mission statement)* Write a mission statement for the bug fixing database system.

Questions for discussion

5. *(Mission statement)* Write a mission statement of the following products.
   (a) A sedan car.
   (b) A public coffee machine that you regularly use (a machine in which you insert money and receive a cup of coffee).
   (c) The word processing software system that you regularly use.
   (d) The organization unit where you currently work.

Chapter 6

Questions for rehearsal

1. *(Function refinement tree)* What does it mean to refine a function in subfunctions?

Exercises

2. *(Making a function refinement tree)* Make a list of functions of the tea bag boxing controller.

3. *(Making a function refinement tree)* Make a list of functions of the bug fixing database system.

4. *(Making a function refinement tree)* Make a list of functions of the chemical tracking system.

Questions for discussion

5. *(Making a function refinement tree)* Make a function refinement tree of the following products. (See exercise 5 of chapter 5 for the mission statements.)

   (a) A sedan car.

   (b) A public coffee machine that you regularly use (a machine in which you insert money and receive a cup of coffee).

   (c) The word processing software system that you regularly use.

   (d) The organization unit where you currently work.

6. *(Function refinement and decomposition)* "There is a relationship between a function refinement tree and the internal decomposition of the system". Give an argument in favor and an argument against this claim.

Chapter 7

Questions for rehearsal

1. *(Delivered value)* Explain why it is important to describe the value delivered by a function before giving other details.

2. *(Assumptions)* What is the role of assumptions with respect to function descriptions?

Exercises

3. *(Service descriptions)* Consider the following function of the teabag boxing controller: Place teabag in box or in waste container according to weight. Describe this function according to the template at the beginning of this chapter, adding any information from the case description (chapter 1) as necessary.

4. *(Service descriptions)* Consider the following function of the bug fixing database system: Register test report and inform programmer and bug manager. Describe this function according to the template at the beginning of this chapter, adding any information from the case description as necessary.

Questions for discussion

5. *(Behavior-oriented versus value-oriented descriptions)* In the following pairs of descriptions, one is a description of behavior as it can be observed, and the other indicates the value of the activity for the environment. Which one is behavioral and which one is value-oriented?

   (a) Sweeping versus cleaning the floor.

   (b) Lighting the room versus switching on the light.

   (c) Setting your preferences versus clicking on the OK button.

   (d) Stopping at a floor with a floor request versus allowing a passenger to get in.

6. *(Value-orientation and design freedom)* A description of delivered value of a service gives you more design freedom than a description of required behavior of the service. Explain.

7. *(Overfull descriptions can hide the fact that information is missing)* The following description of a service of the chemical tracking system includes details that should be excluded (**?**, page 133). (Refer to page **??** for a description of the system.)

> The Requester specifies the chemical to request, either by entering its chemical IS number or by importing its structure from a chemical drawing tool. The system can satisfy the request either by offering the requester a new or used container of the chemical from the chemical stockroom or by letting the Requester place an order to an outside vendor.

(a) Simplify the description to a minimal form that still tells the customer why this is a valuable function.

(b) What kind of information did you drop?

(c) The description omits some important information needed to understand why the service is valuable. Which information is missing?

Chapter 8

Questions for rehearsal

1. *(Basic concepts)* Define the following concepts:
   - Entity,
   - Attribute,
   - Relationship,
   - Association entity.
2. *(Classification concepts)* Define the following concepts:
   - Type,
   - Instance,
   - Extension of a type,
   - Extent of a type,
   - Intension of a type.
3. *(Counting concepts)* Define the following concepts:
   - Absolute cardinality property,
   - Relative cardinality property.
4. *(Taxonomic structures)* Define the following concepts:
   - Generalization of a type,
   - Static generalization,
   - Dynamic generalization,
   - Disjointness of a set of specializations of a type,
   - Covering of a set of specializations of a type.

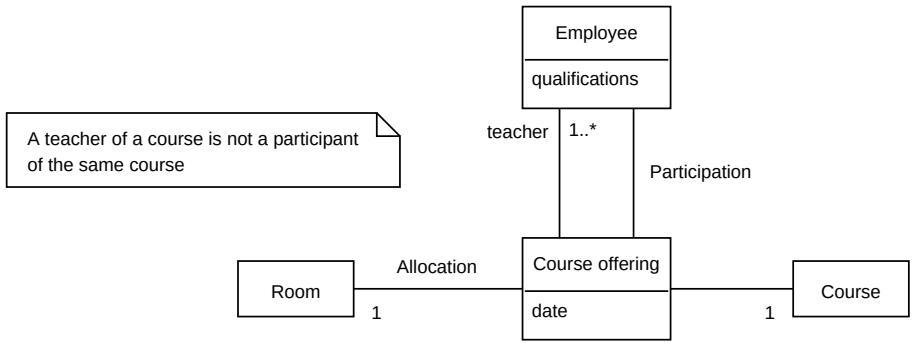
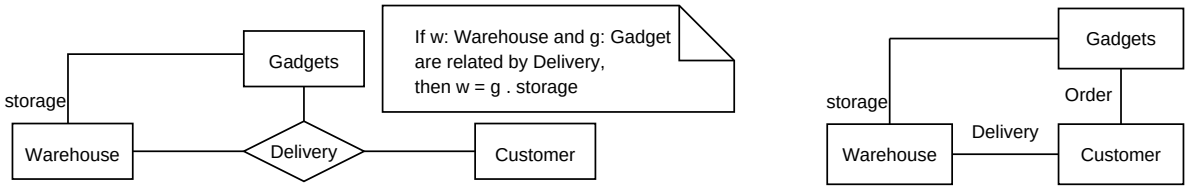


Figure 8.1: A model of courses, participants and teachers.



(a) (b)

Figure 8.2: Choosing between a relationship and an entity.

Exercises

5. *(Entities and relationships)* A company gives in-house courses to its employees. The teachers of each course are employees as well. Figure 8.1 gives an ERD. Suppose that one course offering may be given in several rooms simultaneously, with one teacher per room. Change the model to represent this. Be sure to make your model represent the fact that one teacher cannot teach in two rooms at the same time.

6. *(Ternary cardinalities)* Rephrase the cardinalities of figure **??** in understandable English.

7. *(Connection trap)* Figure 8.2 shows two ways of modeling the situation that a customer is delivered a gadget from a particular warehouse where this gadget is stored. One of them is wrong. Which one? Explain your answer.

8. *(Historical cardinality properties)* Suppose we change the meaning of a cardinality property, so that it expresses *the number of entities that exist or have existed in the past*. Change the cardinalities of figure **??** to reflect this new interpretation of cardinality property.

Questions for discussion

9. *(Intension)*

   - Suppose that currently, all heating tanks in our factory are grey. Does this imply that this property is part of the intension of the type Heating tank?
   - If extension(E1) $\subseteq$ extension(E2) then intension(E2) $\subseteq$ intension(E1) and vice versa. Explain.

10. *(Taxonomy)* Russian army tank number 1967 was the first one to enter Prague in 1945 and is now a monument. Make a model of the generalization relationships (if any) between Car, Vehicle, Monument and Tank.

Chapter 9

Questions for rehearsal

1. *(Basic concepts)*

   - What is the difference between an entity type and an attribute?
   - What is the difference between an entity type and a relationship?
   - What is the difference between a dynamic subtype, a static subtype, and a role?

2. *(Classification)* When do we need a recognition and a counting criterion for a type? Why do we need them?

3. *(Classification and counting)* Explain why the identification criterion of a subtype is the same as that of a supertype, but the recognition criterion of the subtype is different.

Exercises

4. *(Making an ERD)* Consider the following functions of the tea bag boxing controller. The list of functions is basically the list of controller responsibilities:

   - Allow operator to set required tea bag weight.
   - Allow operator to set maximum tea bag count.
   - Place tea bag in box or in waste container according to weight.
   - Replace full box by empty box.

   What is the subject domain of these functions? Make an entity model.

5. *(Making an ERD)* Consider the following functions of the bug fixing support system:

   - Register assignment of bug to programmer and tester, and inform them by email.
   - Register bug fix and inform tester by email.
   - Register test report and inform programmer and bug manager by email.
   - Register bug release.

   What is the subject domain of these functions? Make an entity model.

6. *(Making an ERD)* Consider the functions of the chemical tracking system as described in chapter 1, page **??**. What is the subject domain of these functions? Make an entity model.

7. *(Historical accumulation of information)* Often, an entity accumulates a number of attribute values over time. Consider the following case.

   > An agricultural field has a location, size and has a certain kind of soil. At a certain point in time, it lies fallow, and then it starts accumulating attributes: First, it is decided what crop to plant this year, which seed to use for this crop, and which seed manufacturer to use. The crop is then sown. Now the field has a crop. Next, the crop is harvested. Now, the field has a yield too. And independently from this, fertilizer may be distributed over the field, and every time this happens, the field has a fertilizer on it.

   There is a temptation to model this in an ERD with attributes that have initial value null and that will get a value in the future. Examples are attributes crop and yield. This is bad practice, because null values have many different meanings. Some of the frequently occurring meanings are "unknown", "non-existent" and "not applicable". There are no sound manipulation rules for null values. The accumulation of values over time can be modeled without using null values, by the standard process of creating entities and links between entities.

   Make an ERD of the field subject domain in which you avoid accumulation of attributes.

8. *(Type-instance distinction)* Consider the following subject domain.

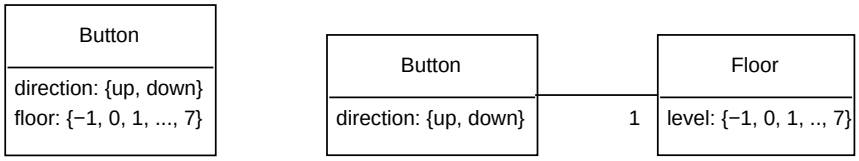
   > Passengers can book flights on an airplane. Each flight consists of a number of hops, and each hop is a trip from an airport to a destination airport. Each airport has a unique abbreviation and a name. Each hop has an estimated duration. Flights have a unique number and an estimated duration. Each flight is flown periodically, at different dates and times. Accordingly, hops are flown periodically at different dates and times. All hops of one flight are flown by one airplane. Passengers actually book flight instances, not flights in general. Each booking has a booking number. After a booking has been made, a ticket can be created, with its own ticket number. And when a ticket exists, seat assignments for the different hops of the flight can be made. The bill for a ticket may be sent to another person than the passenger itself. This other "person" may be a company or in general any legal or natural person.

   Make an ERD of this subject domain. Write down all domain properties that you cannot express in the diagram.

Questions for discussion

9. *(Subject domain boundary)* Explain why Passenger and Conductor are not entities in the ETS subject domain.

10. *(Subject domain boundary)* Consider an information system that contains the train schedule of the Netherlands, plus information about deviations from the schedule, that is made available through the Web to travelers. Using this system, travelers can



(a) (b)

Figure 9.1: Two models of buttons and floors.

find the shortest route from point of departure to a destination, and they can find out about current delays. Which of these following entities are in its subject domain and which are outside? Motivate your answer with respect to the observability of the entity and the relevance of the entity for the information system.

(a) Railway station

(b) Rail track

(c) Traveler

(d) Railway connection

(e) Rail car

11. *(Entities and attributes)* An ERD of the subject domain of an elevator system control must represent buttons. Discuss the reasons to model this as in figure 9.1(a) and the reasons to model this as in figure 9.1(b).

12. *(Entities and relationships)*

    (a) An ERD of the subject domain of a course management system must contain information about students and the exams they do. Discuss the reasons to model this as in figure 9.2(a) and the reasons to model this as in figure 9.2(b).

    (b) Two alternative models are shown in figure 9.3. Discuss the merits of figure 9.3(a) with respect to figure 9.2(a), and the merits of figure 9.3(b) with respect to figure 9.2(b).

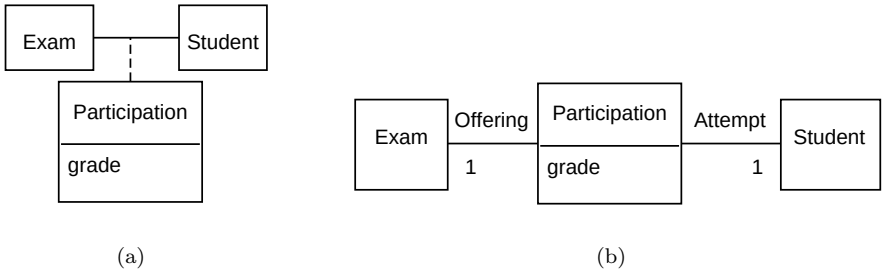
13. *(Taxonomies)* Improve the document taxonomy of figure **??**, taking the principles of classification into account.

14. *(Roles)* Figure 9.4 shows a model of car ownership.

    (a) At least one entity type in this model is really a role that some entity is playing with respect to some other entity. Which entity or entities are roles?

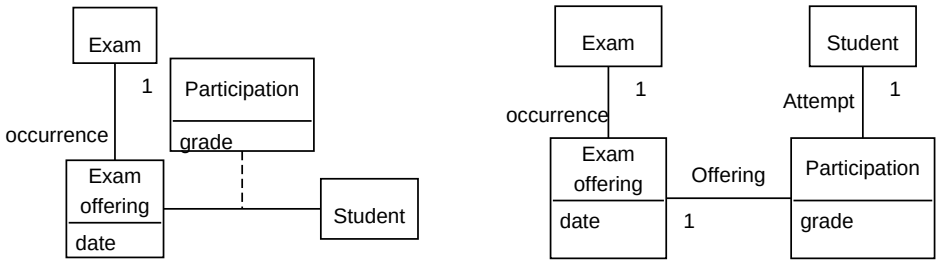
    (b) Explain why the model is wrong.

    (c) Try to provide an improved model that deals with roles more appropriately, and explain why you think it is better.



(a) (b)

Figure 9.2: Two models of students and exams.



(a) (b)

Figure 9.3: Two more models of students and exams.

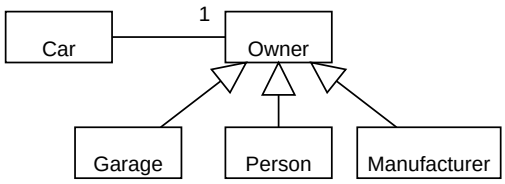


Figure 9.4: A model of car ownership.

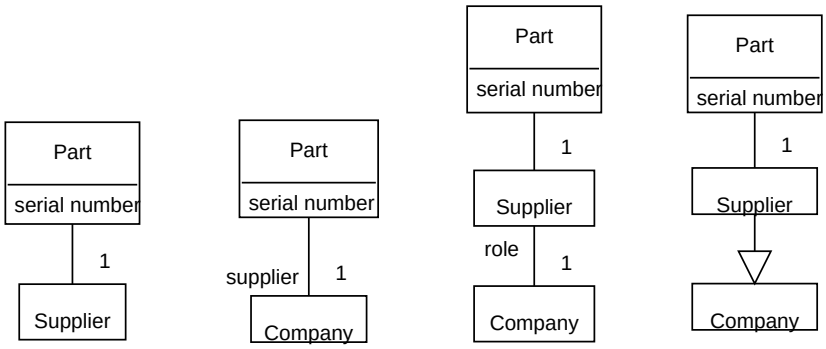


Figure 9.5: Four models of suppliers.

15. *(Roles)* Figure 9.5 shows four models of parts and suppliers. Ask the following questions for each of these models.

    (a) Is a supplier identical to a company?

    (b) Can a supplier have additional attributes in addition to those of companies?

    (c) Can a supplier be identified independently from identifying a company?

    Discuss in which situation you would want to use which model.

16. *(Taxonomy)* Is a wrecked car still a vehicle? Which criteria do you use to answer this question?

Chapter 10

Questions for rehearsal

1. *(Identification)* What are the identifier requirements?
2. *(Definition)*
   - Explain the difference between extensional and intensional definitions.
   - When is it useful to define a term extensionally?
   - Do we list the complete intension of a concept in an intensional definition?

Questions for discussion

3. *(Identifiers)* Which of the following attributes is an identifier? If it is, what does it identify?
   (a) A key of a database table.
   (b) A passport number.
   (c) A personnel number in a company.
   (d) A social security number.
   (e) A Unix process number.
   (f) An Ethernet address.
   (g) An Internet domain name.
4. *(Definitions)* Some of the following terms are open-textured, others have a crisp meaning. Of those that have open texture, explain why you think this is so, and give an extensional definition and indicate who would decide in particular cases whether the term is applicable or not. Of those that have a crisp meaning, give a definition by genus and difference of what you think it means.
   (a) Chair
   (b) Assembler
   (c) Employee
   (d) Car

Chapter 11

Questions for rehearsal

1. *(Basic concepts)*
   - What is the difference between an event list entry in transactional form and in scenario form?
   - Why can an event list entry in scenario form be transformed into a list of entries in transactional form?
2. *(Nondeterminism)* Explain how a nondeterministic event list entry can be disambiguated.
3. *(State transition table)*
   - Are the entries in a state transition table in transactional form?
   - What is the form of the entries in a next-state table?
   - What is the form of the entries in a stateless transformation table?
4. *(Decision table)* What is the relationship between a column in a decision table and a row in a stateless transformation table?

Exercises

5. *(Subject domain behavior)* Make a state transition table of the different states and events in the life of a bug as monitored by the bug fix support system.
6. *(Stimulus-response table)* Consider the following function description of the tea bag boxing controller.
   - **Name:** Remove teabag from balance.
   - **Triggering event:** Tea bag drops on balance.
   - **Delivered service:** The controller ensures that the teabag on the balance is placed in the box if the weight is OK and in the waste container if the weight is not OK. If the box becomes full, it is replaced by an empty one.
   - **Assumptions:**
     - There is a teabag on the balance.

- The box and waste container can still accept teabags.
- Teabag removal is fast enough for the robot arm to be able to remove the next teabag.

This is sufficient as a function description but does not give all scenario details. Expand the service description into a state transition table. Hints: Assume that the controller has three state variables: required, max and current, that represent the required tea bag weight, the maximum tea bag count in a box, and the current number of tea bags in a box, respectively. Assume also that the triggering event is tea bag arrives(w), where w represents the weight of the tea bag. Take the simplistic view that w should be equal to required.

Chapter 12

Questions for rehearsal

1. *(Event triggers)* Which kinds of events can trigger a transition in an STD? How are these represented in an STD?

2. *(Temporal events)* There are two kinds of temporal event. Describe these.

3. *(Transition without named event trigger)* There are two events that can trigger a transition that has no named event trigger. Which events are these?

4. *(Decision states)* A transition without named event trigger can be used to model decisions. Describe in which situations this is unavoidable.

5. *(Variables)* What is the difference between an identifier variable and other variables in an STD?

6. *(State reactions)* What are the two differences between a state reaction and a state transition?

7. *(State hierarchy)* Define the following concepts:

   (a) Default state

   (b) Basic state

   (c) Configuration

   (d) Basic configuration

8. *(Parallelism)*

   (a) Describe the meaning and use of the in(State) predicate.

   (b) What is event broadcasting?

Exercises

9. *(Effect of higher-level interrupt)* Explain why it would be wrong to include switch on as exit action of Not heating in figure **??**.

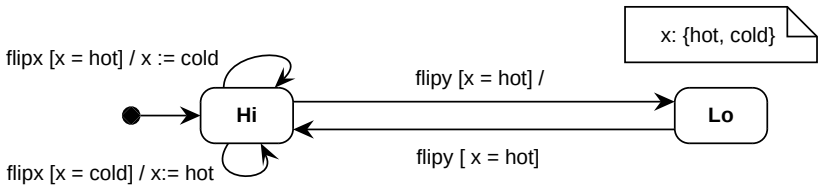


Figure 12.1: An STD with a named and an unnamed state variable.

10. *(Event broadcasting)* Make a table with a trace of the state transitions of the traffic light system in figure **??**, starting from the arrival of event start NS in configuration Red, Red. Follow the trace until you leave the NS.Red state for the second time. Each entry of the table must contain the event being responded to, the current configuration, the next configuration, and the action performed.

11. *(Multistep responses)* Section **??** shows a trace of the diagram in figure **??**. Give the trace that would happen if e5 occurred after e2 occurred but before e3 occurred.

12. *(Representing a state by a node or by a variable value)* There are two ways to represent a state: As a node in an STD or as a value of a variable. Figure 12.1 shows an STD with two variables, one explicitly represented by variable x with possible values hot and cold, and the other nameless, represented by the two nodes of the STD and with possible values Hi and Lo.

    (a) Make a statechart with two parallel components that represents each variable by a parallel component of the statechart. One component is called x and has states Hot and Cold, the other is called y and has states Hi and Lo. Assume that x is initialized to Hot. Does this statechart represent the same behavior as the Mealy diagram in figure 12.1?

    (b) Make an STD with one state and four state transitions, each of which represents one update of a variable. Does this diagram represent the same behavior as the Mealy diagram in figure 12.1?

13. *(Statechart constructs)* Figure 12.2 shows a statechart for a book-lending process. A book is either in or out, and when it is out, the library can remind the borrower to return or extend the loan. If a borrower does not respond to the second reminder, the book is written off. (No doubt you can think of numerous improvements to this simplistic process. But we will use this simple process for the exercise.)

    (a) Use state hierarchy to reduce the number of return and extend arrows in the diagram.

    (b) Use a local variable to count the number of reminders. Take care that this variable has a correct value at all times.

    (c) When a book is not checked in, it can be reserved. When it is reserved, its loan period cannot be extended. Add a parallel Reservation process to the diagram in which this is expressed.

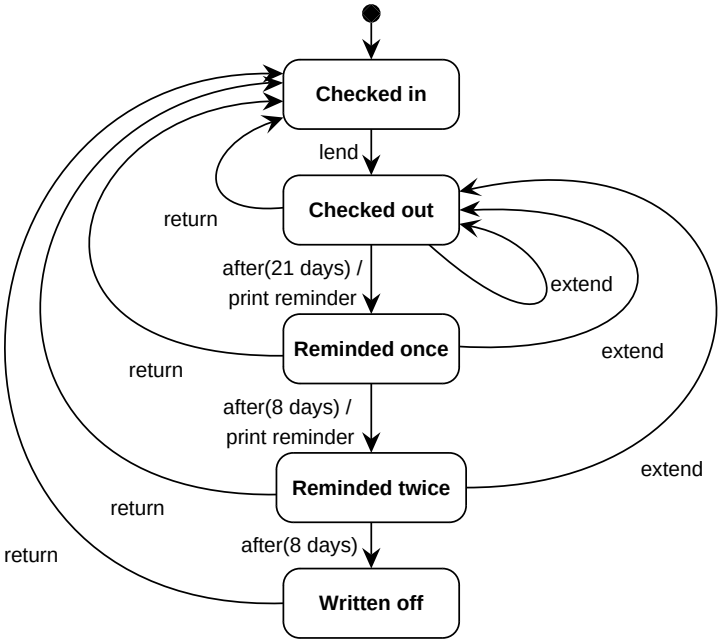


Figure 12.2: A Mealy diagram.

Chapter 13

Questions for rehearsal

1. *(Derivation rules and the frame rule)* Explain how derivation rules and the frame rule save us writing when we describe state transitions.

2. *(Step semantics)*

   (a) Describe what happens when one event triggers two transitions (1) in the step semantics and (2) in the single-transition semantics.

   (b) Describe what happens when several events each trigger a transition (1) in the step semantics and (2) in the sequential-event semantics.

3. *(Multistep semantics)* Explain the difference between a superstep and a run-to-completion semantics.

4. *(Clock-synchrony)* Explain the difference between a clock-synchronous and a clock-asynchronous semantics.

5. *(Perfect technology assumption)* The perfect technology assumption can never be satisfied. Explain what the assumption means in practice.

6. *(Supersteps)* Explain why clock-asynchrony and perfect technology imply a superstep semantics.

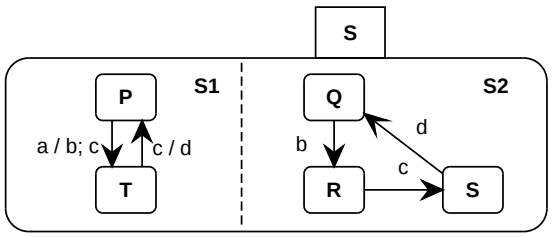


Figure 13.1: A statechart.

Exercises

7. *(Derivation rules)* Write down the derivation rules represented by the diagram in figure 13.1.

8. *(Step and superstep semantics)* Figure 13.2 shows a statechart with broadcasting that we have seen before. Show what happens when in configuration S11, S21, S31 first e1 and immediately after that, e5 occurs, according to the step semantics and according to the superstep semantics.

9. *(Nondeterminism)* Figure 13.3 shows two Mealy diagrams with the same state names.

   (a) Two diagrams are called trace-equivalent if the same sequence of inputs generates the same sequence of outputs. Are the two diagrams in figure 13.3 trace-equivalent?

   (b) Does the node labeled S3 in both diagrams represent the same state? Why (not)?

10. *(Superstep and run-to-completion semantics)* Suppose that in figure 13.1 the current configuration contains P, Q and that event a occurs. Show what happens according to the superstep semantics and what happens according to the nested step semantics. What is the final basic configuration at the end of the superstep and at the end of the nested step?

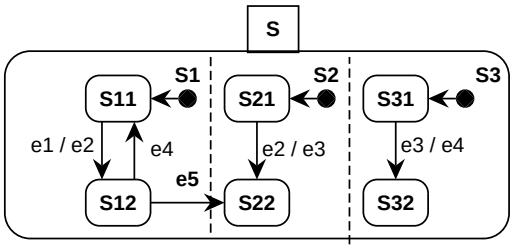


Figure 13.2: A statechart with a multi-step response.

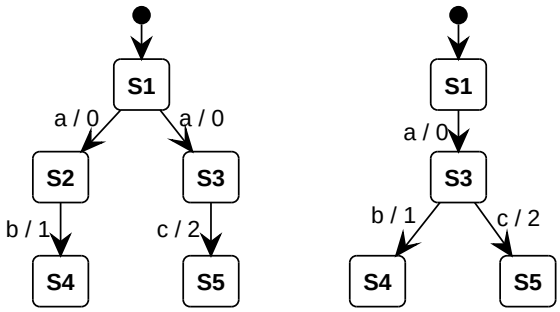


Figure 13.3: A nondeterministic and a deterministic STD.

Chapter 14

Exercises

1. *(Wait states and activity states)* Which states are wait states and which activity states in figure **??**?

2. *(Parallelism)* Which events can and which cannot occur at the same time in the event-action list of the heating controller (figure **??**)?

3. *(Hierarchy)* Make an STD of the behavior in figure **??**. Use hierarchy to deal with the possibility that two events with inconsistent responses can occur at the same time.

4. *(State hierarchy)* This exercise is about the bug fixing database introduced in chapter 1 (page **??**). Figure 14.1 shows a state transition table for bugs. Extend this to allow for the fact that a manager can, in any state of a bug but Fixed, to close the bug, causing it to transition to a state NFBC (Not fixed but closed). Make a hierarchical statechart of the resulting behavior.

5. *(Parallelism)* This exercise is about the tea bag boxing controller introduced in chapter 1 (page **??**). Figure 14.2 shows a state transition table for tea bag removal. required is the required tea bag weight, max is the maximum number of tea bags in a box and

Initially		Bug identified
Event	**Current bug state**	**Next bug state**
assign programmers	Bug identified	Fixing bug
propose fix	Fixing bug	Fix proposed
start testing	Fix proposed	Testing
report problem	Testing	Fixing bug
report fix	Testing	Fixed
release	Fixed	Relased

Figure 14.1: State transition table of bugs.

required: Rational
current, max: Natural

Initially		current := 0	Ready to receive tea bag.
Stimulus	**Current controller state**	**Controller response**	**Next controller state**
tea bag arrives(w)	Ready to receive tea bag, w = required	put in box, current := current +1	Waiting for tea bag to be put in box
	Ready to receive tea bag, w $\neq$ required	remove teabag	Waiting for tea bag to be put in container
teabag removed	Waiting for tea bag to be put in box, current $\geq$ max	stop belt, replace box	Waiting for box to be replaced
	Waiting for tea bag to be put in box, current $<$ max		Ready to receive tea bag
	Waiting for tea bag to be put in container		Ready to receive tea bag
box replaced	Waiting for box to be replaced	start belt, current := 0	Ready to receive tea bag

Figure 14.2: STT for removal of tea bags.

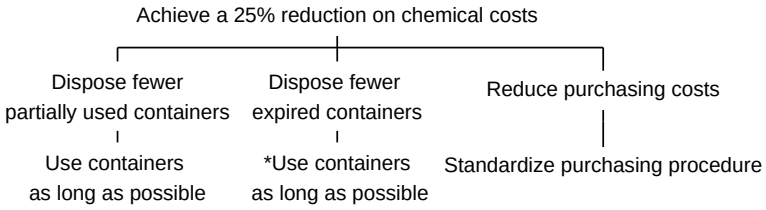


Figure 14.3: Business goals to be contributed to by the chemical tracking system.

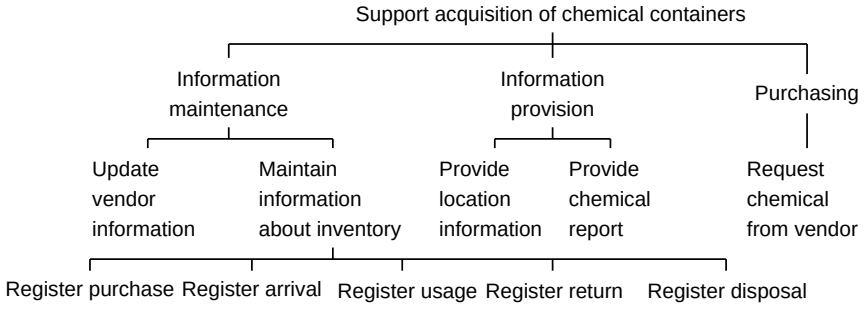


Figure 14.4: Function refinement tree of the chemical tracking system.

current is the current number of bags in a box. For simplicity, we do a straightforward comparison of the weight w with the required value.

(a) Transform this into a Mealy diagram with local variables. Start with listing the states of the diagram.

(b) Although there is a definite sequence among the events, we nevertheless for clarity decide to split the STD into two, one for tea bag removal and one for box replacement. Transform the Mealy diagram into a statechart with a parallel component that deals with tea bag removal and a parallel component that deals with box replacement. Is your statechart equivalent to the Mealy diagram?

6. *(Subject domain behavior modeling)* Consider the following events in the subject domain of the Chemical Tracking System: A chemical container is purchased, a chemical container arrives, a chemical container is used, it is returned, and it is disposed of.

   (a) Draw a statechart for the life cycle of a container and one for the life cycle of a chemical, as they have to be tracked by the CTS. Use state hierarchy where necessary.

   (b) Why is it not possible to combine the two statecharts into one?

7. *(System behavior modeling)* Consider again the Chemical Tracking System. Figure 14.3 shows the relevant business goals and figure 14.4 shows a function refinement tree.

   (a) Which business activities are supported by the system?

(b) What are the transactions to be performed by the system?

(c) Show, by following the system engineering argument, that a system with this functionality will contribute to the solution goals. Note any assumptions about the environment of CTS that you need for this, and any desirable properties of CTS unstated in figure 14.4.

Chapter 15

Questions for rehearsal

1. *(Flows)* Define the following concepts:
   (a) Data flow
   (b) Event flow
   (c) Time-discrete flow
   (d) Time-continuous flow
2. *(Values versus time)*
   (a) Can an event flow be time-continuous? Why (not)?
   (b) Can a time-discrete flow contain continuous values? Why (not)?
3. *(Processes)* Define the following concepts:
   (a) Data process
   (b) Control process
   (c) Stateless process
   (d) Stateful process
4. *(Parameterization)* What is the difference between an index and a process identifier?

Exercises

5. *(Using an STD with variables)* Figure 15.1 shows a DFD for a controller of the tea bag packaging cell. (See chapter 1 for the case description.) The process Control teabag removal is specified by an STD with variables, shown in figure 15.2.
   (a) Make a list of the local variable(s) used by the STD and of the updates and tests performed on them.
   (b) Eliminate all variables, updates and tests from the STD by defining data stores for variables and defining a data process for each update and test, and redirecting flows in the DFD accordingly. Draw the resulting DFD and STD.

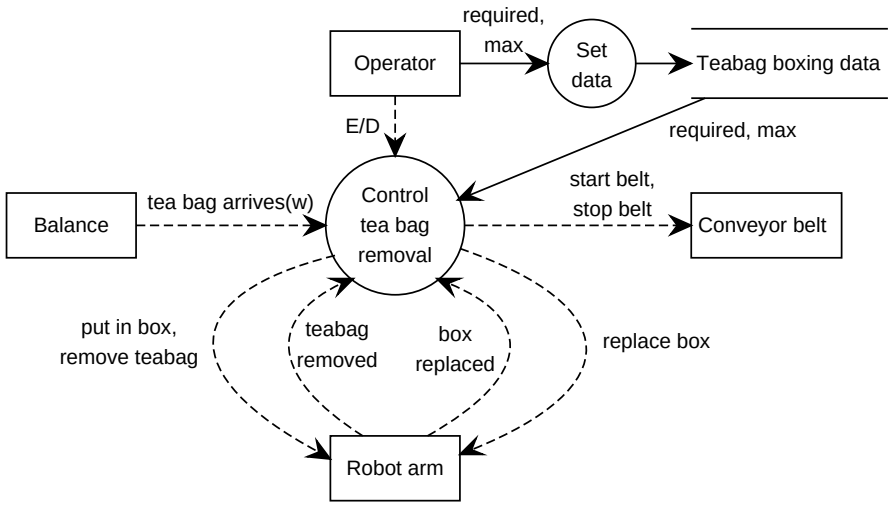


Figure 15.1: DFD of tea bag boxing. "Control teabag removal" is specified by figure 15.2.

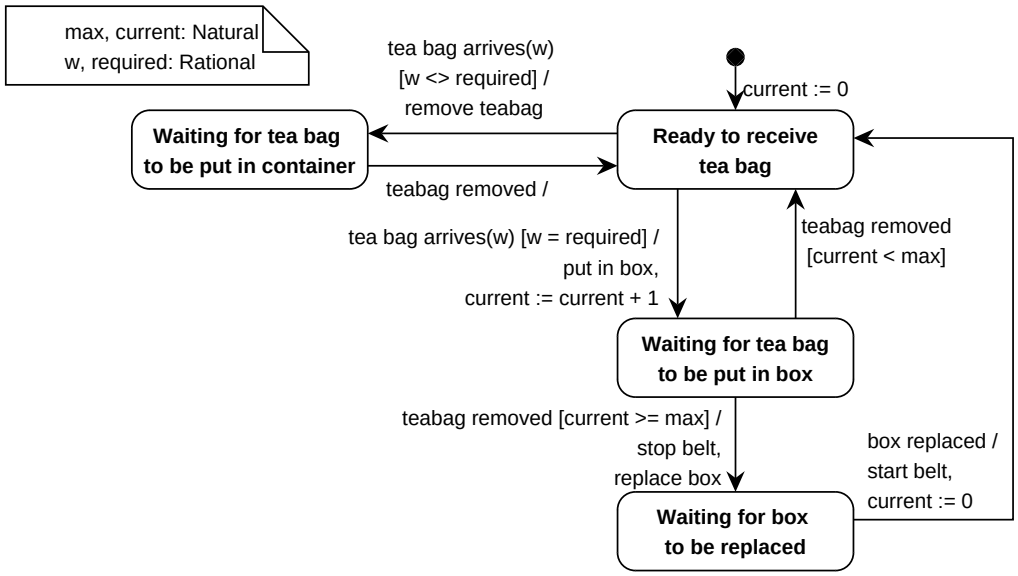


Figure 15.2: STD for removal of tea bag, with local variables. It contains a simplistic test of the tea bag weight.

6. *(Making a DFD)* Exercise 5 of chapter 9 (page 19) lists the functions of the bug fixing support system and asks you to make an ERD of the subject domain.

   (a) Use the function list and ERD to make a DFD of the bug fixing support system, in which the data stores contain data about the subject domain.

   (b) Extend the DFD with a bug monitoring function as follows. Exercise 4 of chapter 14 gives a state transition table for the life cycle of bugs. Include a data process in your DFD that monitors the state of each bug and only updates the bug data according to this life cycle.

Questions for discussion

7. *(Processes and activity states)* Is there a difference between a process and an activity state?

8. *(Composite processes)* A composite process may or may not be stateful. State exactly when it is stateless.

9. *(Enabling a stateless process)* Would it be meaningful to enable or disable a stateless process? If so, explain the meaning and if not, explain why.

10. *(Material flow and store)* Some people like to include material flows and material stores in a DFD, that transport and hold material items, defined as items that have a physical mass, size, and location in space. For example, a material flow from a process to a material store would represent the production of a material item by the process, and the placement of this item in store. An arrow from the store to a process would represent the usage of a material item from the store by the process.

    (a) What is the differences between material and data?

    (b) What are the differences between a material flow and a data flow?

    (c) What are the differences between a material store and a data store?

    (d) Suppose we represent material flows by fat arrows and material stores by fat parallel lines. Make a DFD in which you represent the processes, material and data flows in the following situation:

    > A bank customer writes an order to transfer money from bank account A to bank account B. He posts this order by ordinary mail. When it arrives at the bank, it is put into the in-box of a desk clerk, who then enters the transaction in the bank's information system, that executes the order. The bank sends a confirmation of the execution back by mail.

Chapter 16

Questions for rehearsal

1. *(Basic concepts)* Define the following concepts:
   (a) Component
   (b) Communication channel
   (c) Event channel
   (d) Data channel
2. *(Traceability)*
   (a) What is the difference between allocation and flowdown?
   (b) How does a traceability table help with giving the system engineering argument?

Exercises

3. *(Allocation and flowdown)* The functions of the tea bag boxing controller have earlier been identified as follows: The list of functions is basically the list of controller responsibilities:

   - Allow operator to set required tea bag weight.
   - Allow operator to set maximum tea bag count.
   - Place tea bag in box or in waste container according to weight.
   - Replace full box by empty box.

   Exercise 5 of chapter 15 (page 41) gives a DFD for the tea bag boxing controller. Transform this into a communication diagram and make an allocation table.

4. *(Allocation and flowdown)* The functions of the bug fixing support system have earlier been identified as follows:

   - Register assignment of bug to programmer and tester, and inform them by email.
   - Register bug fix and inform tester by email.
   - Register test report and inform programmer and bug manager by email.
   - Register bug release.

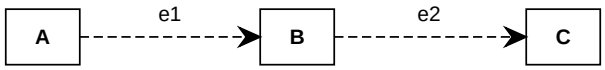


Figure 16.1: A chain of cause and effect

We take registration to include monitoring, i.e. the system will not register a state change in a bug if this is not possible according to the life cycle of the bug. Now suppose you are given the following components:

- **Manage assignments.** Register and store assignments of programmers and testers to bugs and inform them.
- **Programmer data.** Store data about programmers and testers.
- **Bug monitor.** Store data about bugs and check that the current action is permissible in the bug life cycle.
- **Release.** Release a bug.
- **Manage fixes.** Register and store fixes, and inform tester of fix.
- **Manage test reports.** Manage and store test reports, and inform programmer and manager about this.

Assemble a bug fixing support system from this. Draw a communication diagram to show the communication channels, and an allocation table to show that all functions have been implemented.

Questions for discussion

5. *(Global channels)* Enable and disable input prompts of a component C that itself has lower-level components, are really global input channels that are available to all components of C. A trigger input is however local like any other channel, and must be connected to particular components of C. Explain.

6. *(Causation)* Figure 16.1 shows a chain of cause and effect involving three components and two event channels. A causes e1, which causes B to cause e2. In which cases can we take out B and let A cause e2 directly?

Chapter 17

Questions for rehearsal

1. *(Input buffer)*

   (a) Does the system, as a whole have one input buffer or does each component have an input buffer?

   (b) Describe the semantic options for input buffers.

2. *(Reading and writing)* What is the difference between lazy read/eager write and eager read/lazy write?

3. *(Synchronous communication)* Consider the following two incompatible definitions of synchronous communication:

   (a) A sender cannot proceed until the receiver has received the message.

   (b) A sender cannot proceed until the receiver has processed the message.

   Which one is equivalent to the definition given in this chapter?

4. *(Blocking)* If there is overwrite semantics, we do not need to choose a blocking semantics. Explain.

5. *(Network behavior)* Explain why both the network step and the network multistep semantics can be executed in a breadth-first or depth-first way.

Exercises

6. *(Data store semantics)* Explain what a data store would do in an eager read/lazy write semantics and what it would do in a lazy read/eager write semantics.

7. *(Network semantics)*

   (a) Execute the network in figure **??** according to the breadth-first semantics until you enter an infinite loop. Assume that on all output channels, a component sends some message every time it receives one on any input channel.

   (b) Execute the network according to a depth-first semantics, choosing a shortest path from `e` to some output. Explain why this leads to different output responses of the SuD.

Questions for discussion

8. *(Data store semantics)* The communication between processes and data stores should be as follows: At the start of a step, the process performs all its read operations from a data store, and at the end of a step, it performs all its updates to the data store. Describe this semantics from the point of view of a data store and explain what the difference is with the eager read/lazy write semantics discussed in this chapter.

9. *(Input buffers)* When can we dispense with input buffers?

10. *(Zero delay and time-continuous channels)* "A time-continuous value stays in a channel until it is replaced, and therefore the channel introduces delay in the transmission." Show why this argument is false.

11. *(Nested step and depth-first semantics)* What is the relationship between nested step semantics of component behavior and depth-first semantics of network behavior?

12. *(Ordering of input events)* What are the possible causes for loss of ordering of input events? Give for each possible cause a way to avoid this loss.

Chapter 18

Exercises

1. *(Context structuring)* Identify the information-provision, directive and manipulative context structures in the following examples:

   (a) The tea bag boxing system,

   (b) The bug fixing support system.

2. *(Context modeling)* Make a context model of the Chemical Tracking System (CTS). Ask the following questions to find the context boundary:

   (a) What is the composite system of which CTS is a part?

   (b) About which entities does CTS need information? How will this information get to CTS?

   (c) Where should the desired effects of CTS take place? Through which channel will the system cause this effect?

   (d) Who are the users?

   Which structure(s) does this context have?

Questions for Discussion

3. *(System boundary)* Figure 15.1 (page 42) shows a DFD of the tea bag boxing system, including external entities. The functions of the system are the following: The list of functions is basically the list of controller responsibilities:

   - Allow operator to set required tea bag weight.
   - Allow operator to set maximum tea bag count.
   - Place tea bag in box or in waste container according to weight.
   - Replace full box by empty box.

   (a) Discuss ways in which functions could have been re-allocated from the tea bag boxing system to the operator.

   (b) Could the pieces of the functionality of the system have been re-allocated to the balance, robot arm and conveyor belt?

4. *(Context boundary)* The context of TIS contains communications not modeled by figure **??**, such as information that the registration desk and coordinator get from the printer.

   (a) Give an argument why these communications should be included.

   (b) Give an argument why these communications should *not* be included.

   Which argument has your preference? Why?

5. *(The subject domain of a manipulative function)* Figure **??** (page **??**) shows part of the subject domain of the ETS.

   (a) How would you decide that some of these entities are to be stored and manipulated by the system itself?

   (b) A ticket represents a right to passage. Why aren't these rights represented by the context diagram?

Chapter 19

Questions for rehearsal

1. *(Architectural terminology)* What is the difference and relationship between the concept of an architecture and that of an architectural style?

2. *(Encapsulation versus layering)* What is the difference and relationship between encapsulation and layering?

3. *(Architectural styles)*

   (a) There are really two basic architectural styles for reactive systems at the requirements level, that each have some variants. Describe these styles and variants.

   (b) Relate these styles to the three basic reactive system functions of information provision, direction and manipulation of lexical items.

4. *(Decomposition guidelines)* Decomposition guidelines are related to the different functional aspects of a system: Functions, behavior, and communication, and the subject domain. Give these guidelines.

5. *(requirements-level and implementation-level architectures)* Explain the difference between requirements-level architecture and implementation-level architecture. What is the use of this distinction?

Exercises

6. *(Making a DFD)* Figure 20.1 shows a function refinement tree of the chemical tracking system (CTS). Design a functional decomposition of the system.

7. *(Decomposition guidelines)* Exercise 5 (page 41) gives a DFD for the tea bag boxing controller. Transform this into a communication diagram using a hybrid style. Explain which decomposition guidelines have been used in the resulting decomposition.

8. *(Decomposition guidelines)* Figure 20.2 shows a decomposition of the elevator controller. Classify each component according to the guidelines given in section **??**.

9. *(Data transformations and objects)* Figure 20.3 shows a decomposition of the bug fixing support system, using a hybrid style. The Bug component monitors the life cycle of bugs. (Exercise 4, page 37 gives an STT for this.) Assignment, fixing,

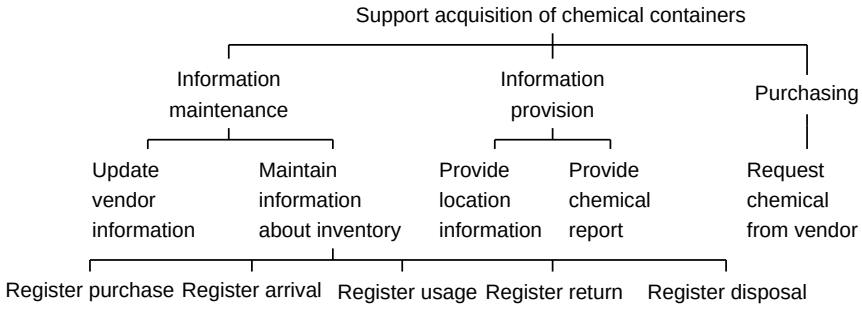


Figure 19.1: Function refinement tree of the chemical tracking system.

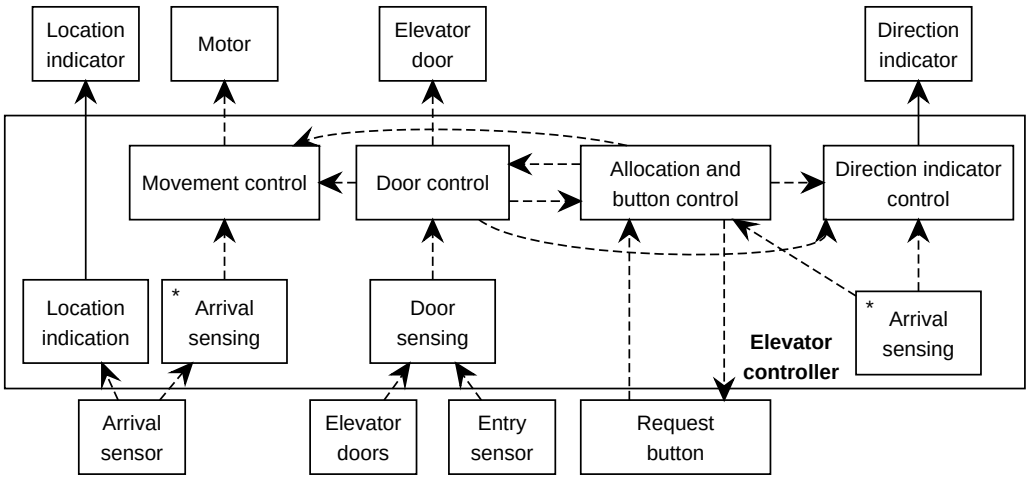


Figure 19.2: Decomposition of the elevator controller. For ease of reading, arrow labels have been omitted.

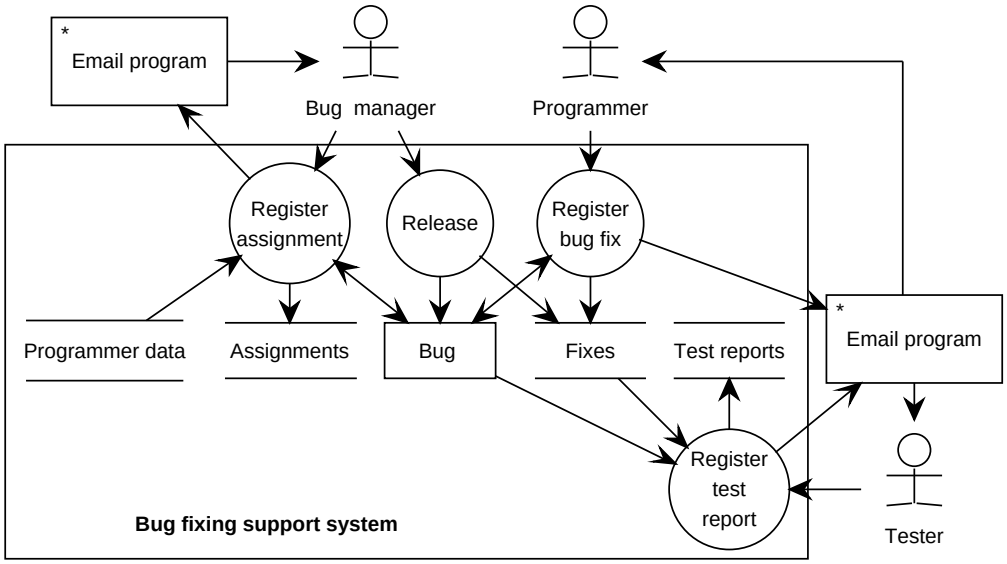


Figure 19.3: Yet another decomposition of the bug fixing support system, using a hybrid style.

testing and releasing a bug are all represented in figure 20.3 as data transformation, i.e. stateless components that produce their output in one step. Explain why it is a design mistake to model these events as data transformations and give two possible solutions.

10. *(Hybrid style)* Design a decomposition of the Electronic Ticket System that encapsulates the ticket life cycle in an object but also retains the necessary databases. Draw an STD for the ticket life cycle.

11. *(Changing style)* The architecture of the heating controller shown in figure **??** uses a Von Neumann style. Change it into one with a pure object-oriented style.

Chapter 20

Questions for rehearsal

1. *(Architectural terminology)* What is the difference and relationship between the concept of an architecture and that of an architectural style?

2. *(Encapsulation versus layering)* What is the difference and relationship between encapsulation and layering?

3. *(Architectural styles)*

   (a) There are really two basic architectural styles for reactive systems at the requirements level, that each have some variants. Describe these styles and variants.

   (b) Relate these styles to the three basic reactive system functions of information provision, direction and manipulation of lexical items.

4. *(Decomposition guidelines)* Decomposition guidelines are related to the different functional aspects of a system: Functions, behavior, and communication, and the subject domain. Give these guidelines.

5. *(requirements-level and implementation-level architectures)* Explain the difference between requirements-level architecture and implementation-level architecture. What is the use of this distinction?

Exercises

6. *(Making a DFD)* Figure 20.1 shows a function refinement tree of the chemical tracking system (CTS). Design a functional decomposition of the system.

7. *(Decomposition guidelines)* Exercise 5 (page 41) gives a DFD for the tea bag boxing controller. Transform this into a communication diagram using a hybrid style. Explain which decomposition guidelines have been used in the resulting decomposition.

8. *(Decomposition guidelines)* Figure 20.2 shows a decomposition of the elevator controller. Classify each component according to the guidelines given in section **??**.

9. *(Data transformations and objects)* Figure 20.3 shows a decomposition of the bug fixing support system, using a hybrid style. The Bug component monitors the life cycle of bugs. (Exercise 4, page 37 gives an STT for this.) Assignment, fixing,

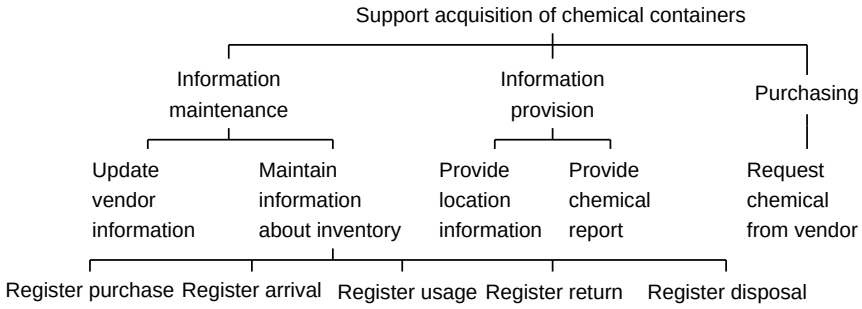


Figure 20.1: Function refinement tree of the chemical tracking system.

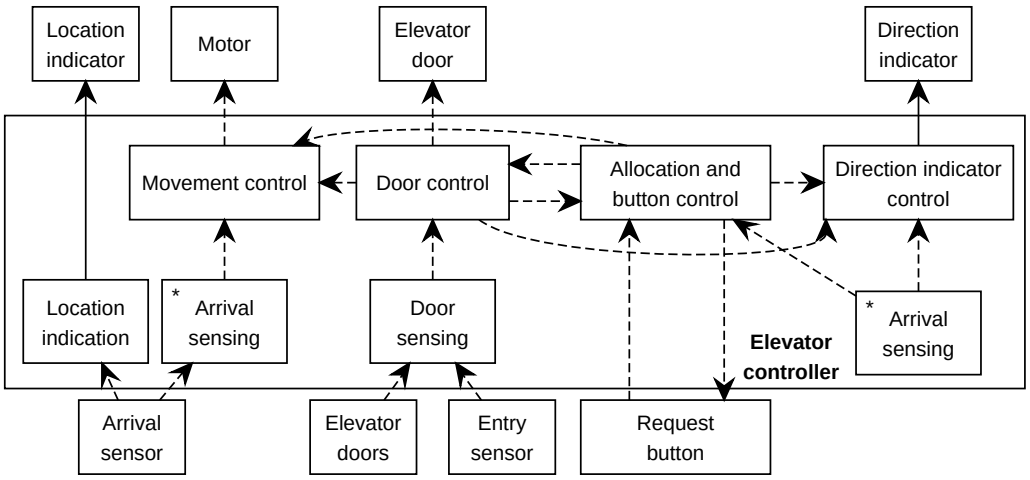


Figure 20.2: Decomposition of the elevator controller. For ease of reading, arrow labels have been omitted.

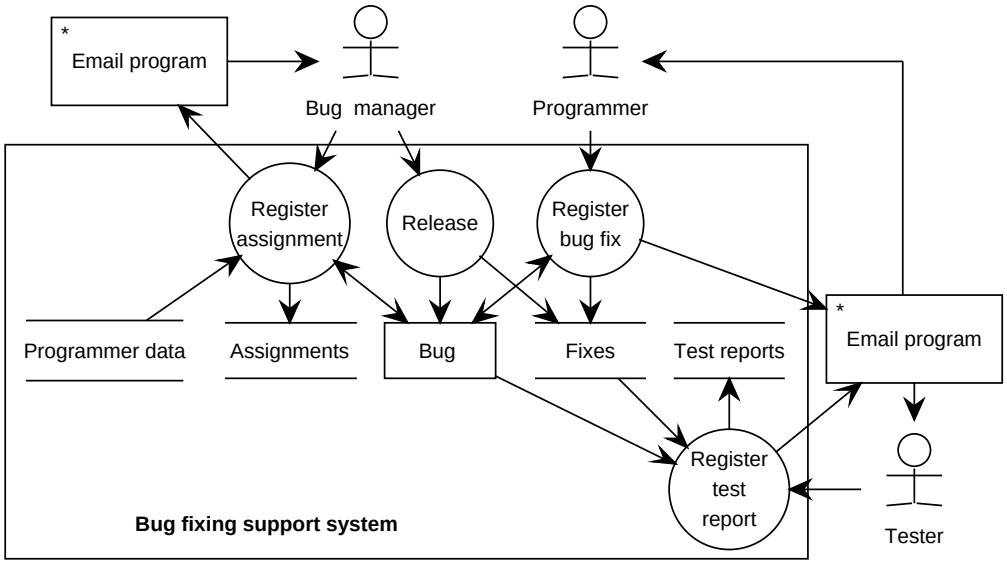


Figure 20.3: Yet another decomposition of the bug fixing support system, using a hybrid style.

testing and releasing a bug are all represented in figure 20.3 as data transformation, i.e. stateless components that produce their output in one step. Explain why it is a design mistake to model these events as data transformations and give two possible solutions.

10. *(Hybrid style)* Design a decomposition of the Electronic Ticket System that encapsulates the ticket life cycle in an object but also retains the necessary databases. Draw an STD for the ticket life cycle.

11. *(Changing style)* The architecture of the heating controller shown in figure **??** uses a Von Neumann style. Change it into one with a pure object-oriented style.

Chapter 21

Exercises

1. *(Factoring out activities in a state)* When the statechart in figure **??** is started, it enters the initial configuration {S1, S11} in one step and then waits for an event. Describe the step(s) that occur(s) when activity A in figure **??** is started, using the step execution algorithm.

2. *(Factoring out activities in a state)* Give a pair of communicating statecharts that represent the same behavior as that of figure **??**.

3. *(Execution semantics)* Figures 21.1 and 21.2 shows an activity chart and a statechart for the tea bag boxing controller. Suppose the statechart is in configuration {Ready to receive teabag, Ready to replace box} and current = max − 1. Describe the sequence of steps and states triggered by the following sequence of events:

   E1 tea bag arrives(w) with w = required

   E2 tea bag removed

   E3 box replaced

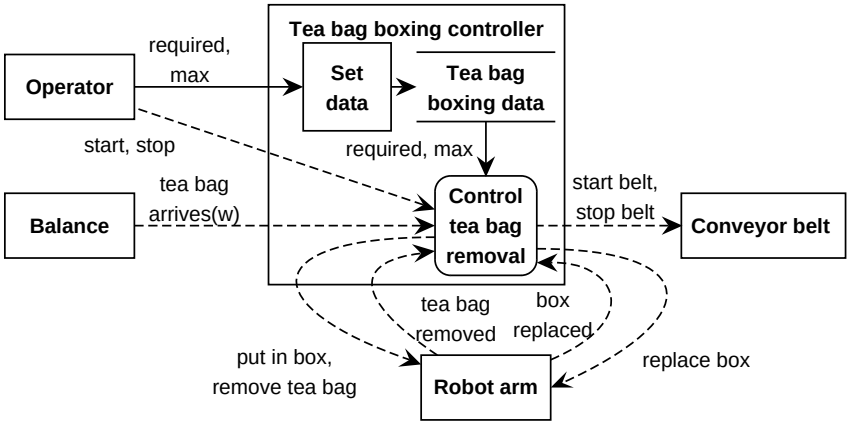


Figure 21.1: Activity chart of the tea bag boxing controller.

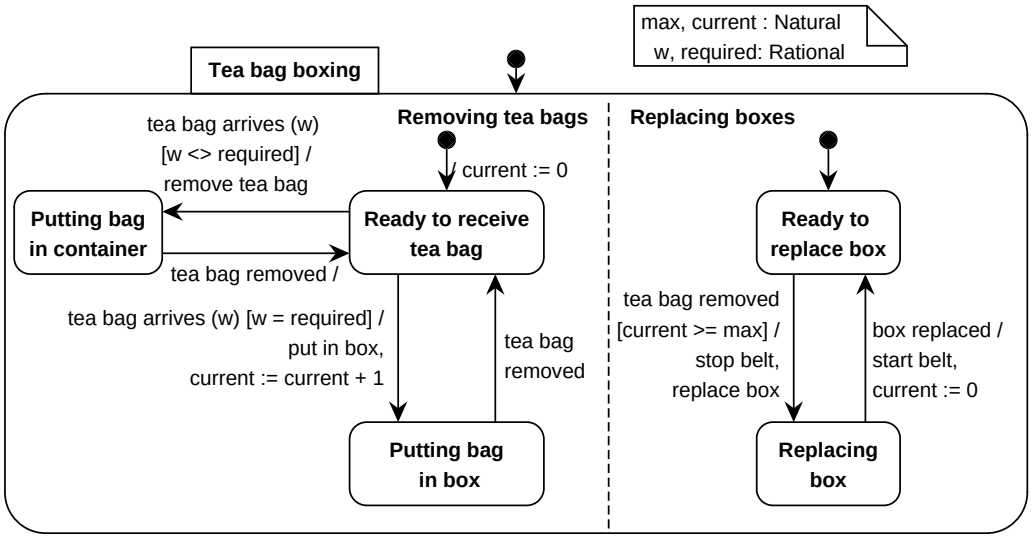


Figure 21.2: Statechart for the "control tea bag removal" activity.

Chapter 22

Questions for rehearsal

1. *(Activity diagrams)*
   (a) What is the difference between a wait state and an activity state?
   (b) Can a wait state be decomposed into a lower-level activity diagram?
   (c) Can an activity state be decomposed into a lower-level activity diagram?
   (d) What is a completion event?
2. *(Static structure diagrams)* Define the following concepts.
   (a) Object
   (b) Class
   (c) Class extension
   (d) Class extent
   (e) Class intension
   (f) What is the difference between an object and an entity?
3. *(Services)* Define the following concepts.
   (a) Operation
   (b) Method
   (c) Signal
   (d) Signal reception
4. *(Service specification)*
   (a) How can an operation be defined?
   (b) How can a signal be defined?
   (c) What are the differences between an operation and a signal?
5. *(Messages)*
   (a) List the five type of messages that can be sent between objects.
   (b) Are there actions that do not lead to a message?
   (b) Are there events that cannot be caused by a message?

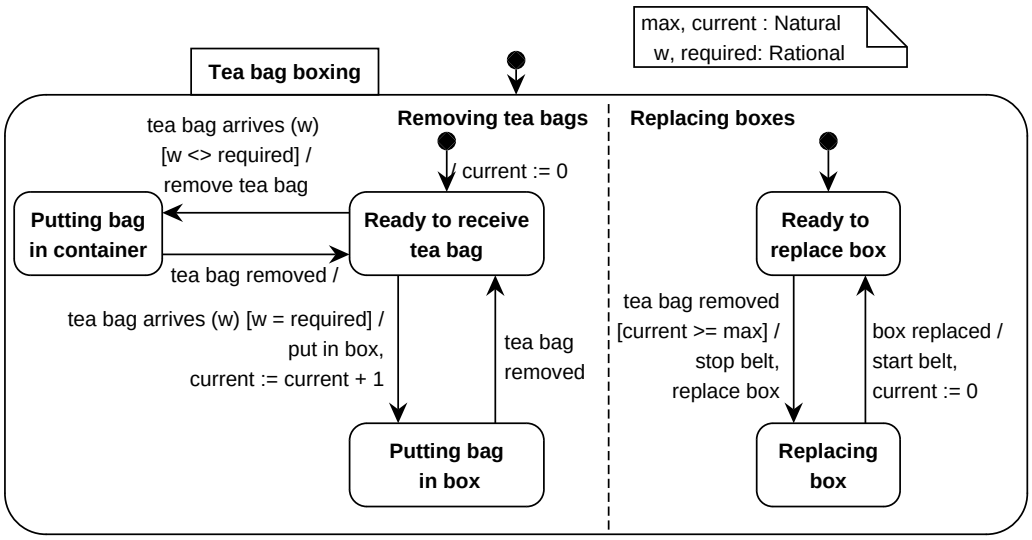


Figure 22.1: Statechart for removal of tea bag.

Exercises

6. *(Communication and coherence)* Figure 22.1 shows a statechart for tea bag boxing.

   (a) Give a communication diagram for the tea bag boxing controller with two components: A controller of the robot arm, whose behavior is specified in figure 22.1, and a data container, that contains the required tea bag weight and maximum number of bags in a box. The data container is initialized by the operator and read by the control object. Assume that the tea bag removal controller is created and destroyed by the operator.

   (b) Make an SSD of the system and its context.

   Take care that the communication diagram and SSD are consistent with the statechart.

7. *(Building an SSD)*

   (a) Figure **??** (page **??**) lists the specifications of the transformations Arrive stimulus rec, Door stimulus rec and Motor action comp of the elevator controller architecture. Figure **??** shows an SSD for the relevant access paths needed by these components and by Movement control. Check that the SSD defines all access paths needed by these components.

   (b) Define the transformations Entry stimulus rec and Door action comp of the elevator controller, using the dictionary of section **??** (page **??**).

   (c) Draw an SSD that defines the access paths needed by Entry stimulus rec, Door action comp and Door control.

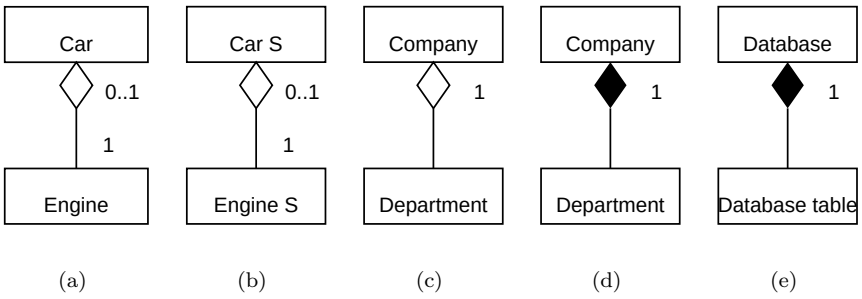


Figure 22.2: Examples of aggregation and composition.

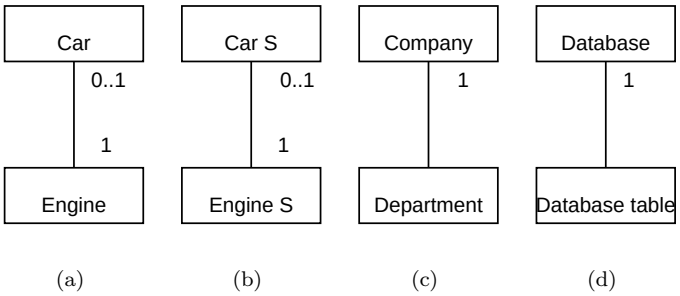


Figure 22.3: Examples of association.

Questions for discussion

8. *(Aggregate and composite objects)*

   The UML allows the representation of aggregation and composition relationships. Figure 22.2 gives four examples. Car and Engine are physical entities in the subject domain, Car S and Engine S are software objects in a software system.

   - The hollow diamond represents a whole-part relationship: An engine is a part of a car.
   - The black diamond represents strong composition, which means that the part cannot exist outside the whole. So the figure says that a database table cannot exist outside a database.

   Discuss the differences in meaning between the models in figure 22.2 and figure 22.3 in the following respects:

   - Creation, update or deletion of the part or whole.
   - Any meaning attached to the diamond that is not already conveyed by the multiplicity constraints.

Questions for rehearsal

1. *(Design strategies)*

   (a) Describe the situations in which you can use a linear strategy, an incremental strategy, an evolutionary strategy, and a concurrent strategy.

   (b) Describe the role that can be played by the Yourdon, Statemate and UML approaches in these strategies.

2. *(Role of techniques in design)* Describe the role that the Yourdon, Statemate, and UML approaches can play in the design cycle.

3. *(Architectural ontologies)*

   (a) Describe the ontology of DFDs and how the different kinds of entities in this ontology are represented by a DFD.

   (b) Describe the ontology of SSDs and how the different kinds of entities in this ontology are represented by an SSD.

4. *(Hybrid notations)*

   (a) Explain how we can augment a DFD with extended Mealy diagrams.

   (b) Explain how we can augment an SSD with elements from the ontology of DFDs.

5. *(Transforming an SSD into a DFD)* There are two ways to transform an SSD into a DFD. Explain these two ways.

Bibliography

Goldsmith, S. (1993), *Real-Time Systems Development*, Prentice-Hall.