# Modelling of Distributed Systems - Tutorial 1

19th April 2018

**General information about the exercise** In “lecture materials” we include the course script, slides, exercise sheets and cited literature (all available at `http://www4.in.tum.de/lehre/mvs/`). In the lecture, key concepts and practical motivation are explained. To work out the exercises, we recommend using the notations from the lecture notes and possibly cited literature or slides.

## Exercise 1 Answer the following questions based on the example of a bank cash machine (ATM)

Answer based on everyday knowledge –

- To which complete system does ATM belong to?
- Where must be the system boundary of the ATM sub-system?
- Which other sub-systems exist in the immediate environment of the ATM?
- How do the interfaces to these sub-systems look like?
- Which working states can the ATM be in?
- What can be the problems in the use or operation of the ATM?

Use knowledge from the introduction lecture.

## Exercise 2 Modelling of ATMs

As *basic condition*, we are given that – each EC-card has an access-number (PIN) and a maximum allowed transaction amount.

Model an ATM system that should, as *requirements*, allow the following *operations*:

- The user enters an EC-card at the start of the operation.
- The user is asked for a PIN.
- The ATM compares user-provided PIN with the PIN of the EC-card.
  - During the prompt for PIN, the user can either enter the PIN, or cancel the process. If process is cancelled, the EC-card is returned and the operation is finished.
  - If the PIN matches, the user can enter a sum of money.

- The ATM compares the entered amount with the maximum amount for the EC-card
  - If the amount is too high, it can be entered again or the process can be cancelled by the user. If the process is cancelled, the EC-card is returned and the operation is finished.
  - If the amount is less than or equal to the maximum amount, first the EC-card is returned, and then the amount is dispensed.

In the lecture, we discussed state machines with unlabelled, labelled (classified into input or output labels) and input/output-marked transitions. Use these concepts for the following exercises.

### 2.1: Modelling of a state machine with unlabelled transitions

(a) Define the set $\Sigma$ of all possible states and name the initial states, $\sigma_0 \in \Sigma$.

(b) Define the state transition function, $\Delta : \Sigma \to \mathcal{P}(\Sigma)$.

(c) Draw a state transition diagram.

### 2.2: Modelling of a state machine with labelled transitions

(a) Define the set $\Sigma$ of all possible states and name the initial states, $\sigma_0 \in \Sigma$.

Hint: Keep in mind that here the set of control states are represented as variable assignments. Take care that these sets are disjoint, otherwise the state transition relation will not correspond to the representation in the diagram.

(b) Define the set of all possible actions, $A$.

(c) Define the state transition function, $\Delta_a : \Sigma \to \mathcal{P}(\Sigma)$ for all specified $a \in A$.

(d) Draw a state transition diagram.

### 2.3: Modelling of a state machine with input and output

(a) Define the set $\Sigma$ of all possible states and name the initial states, $\sigma_0 \in \Sigma$.

Hint: Often, the control states are not clearly characterized by states of the variables. In such cases, we can attach special attributes to these control states to characterize them clearly.

(b) Define the set of all inputs, $I$ and outputs, $O$.

(c) Draw a sketch that clearly represents this interface.

(d) Define the state transition function, $\Delta : \Sigma \times I \to \mathcal{P}(\Sigma \times O)$.

(e) Draw the state transition diagram.

**2.4:** What are the advantages and disadvantages of these state machines?

## Exercise 3 [Homework] Modelling a train door control

For a mechanical door unit of a public transport system (S-Bahn), a secure, software-based central door control must be developed. We only consider S-Bahns with a very simplified door unit. The door control should do the following:

- Closing controlled by the Bahn-personnel.
- Should be opened by the passenger(/s) only when the train is stopped; the train will start only after the door is closed.
- Automatic release and manually opened by the passengers in case of emergency.

(a) Identify the components within the system boundary of the door control. Also identify the components outside this boundary. Which possible communication channels may exist between them?

(b) Specify the control behaviour for each component my means of a state machine. [1]

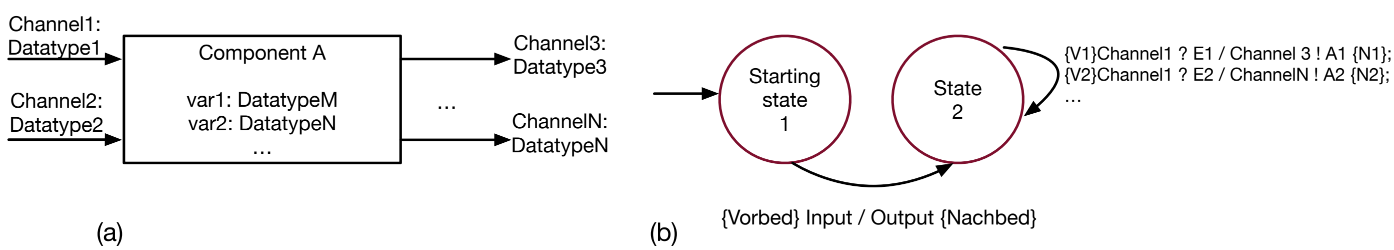


Abbildung 1: Notation for interface (a) and component behaviour (b)

[1] Please refer to the notation in fig. 1b. For transitions: *Vorbed* denotes the internal data states, and *Nachbed* denotes the assignments to these data states. *Vorbed* and *Nachbed* are optional. The terms *input* and *output* are as follows: *channel1* □ *Data1, ..., ChannelN* □ *DataN*, where □ to "?" specifies input - and to "!" specifies output. Empty inputs and outputs can be shown with "–". A *Data* element is understood by the current channel assignment (1, ..., N) and the data state of the corresponding component.