Scenarios: The Idea

Use Cases

Use Case Diagrams

User Stories

Sequence Diagrams

A Brief History

Live Sequence Charts

LSC Body

Syntax:

Well-Formedness

Towards Semantics:

Informatik III (Automata Theory)

Cuts, Firedsets

Automaton Construction

Excursion: Symbolic Büchi Automata
The idea of scenarios (sometimes without negative or anti-scenarios) (re-)occurs in many process models or software development approaches.

In the following, we will discuss two-and-a-half notations:

- **Use Cases** and **Use Case Diagrams** ([OOSE](https://en.wikipedia.org/wiki/Use_case)
- **User Stories** (part of **Extreme Programming**)
- **Sequence Diagrams** (here: **Live Sequence Charts** ([Damm and Harel](https://www.springer.com/de/book/9783540421631), 2001))

### Use Case: Definition

**Use Case** — A sequence of interactions between an actor (or actors) and a system triggered by a specific actor, which produces a result for an actor. ([Jacobson](https://en.wikipedia.org/wiki/Ivar_Jacobson), 1992)

### Use Case: More Precisely

- A use case has **participants**: the system and at least one actor.
  - **Actor**: an actor represents what interacts with the system.
  - An actor is a role, which a user or an external system may assume when interacting with the system under design.
  - Actors are not part of the system, thus they are not described in detail.
  - Actions of actors are non-deterministic (possibly constrained by domain model).
- A use case is triggered by a **stimulus** as input by the main actor.
- A use case is **goal oriented**, i.e. the main actor wants to reach a particular goal.
- A use case describes all interactions between the system and the participating actors that are needed to achieve the goal (or fail to achieve the goal for reasons).
- A use case **ends** when the desired goal is achieved, or when it is clear that the desired goal cannot be achieved.

### Use Case Example: ATM Authentication

```plaintext
name: Authentication
goal: the client wants access to the ATM
pre-condition: the ATM is operational, the welcome screen is displayed, card and PIN of client are available
post-condition: client accepted, services of ATM are offered
post-cond. in exceptional case: access denied, card returned or withheld, welcome screen displayed
actors: client (main actor), bank system
open questions: none
normal case:
1. client inserts card
2. ATM reads card, sends data to bank system
3. bank system checks validity
4. ATM shows PIN screen
5. client enters PIN
6. ATM reads PIN, sends to bank system
7. bank system checks PIN
8. ATM accepts and shows main menu
exception case 2a:
2a.1 ATM displays "card not readable"
2a.2 ATM returns card
2a.3 ATM shows welcome screen
exception case 2b:
card readable, but not ATM card
exception case 2c:
nom connection to bank system
exception case 3a:
card not valid or disabled
exception case 5a:
client cancels
exception case 5b:
client doesn't react within 5 s
exception case 6a:
nom connection to bank system
exception case 7a:
first or second PIN wrong
exception case 7b:
third PIN wrong
```


### Once Again: Use Case Definition

**Use Case** — A sequence of interactions between an actor (or actors) and a system triggered by a specific actor, which produces a result for an actor. ([Jacobson](https://en.wikipedia.org/wiki/Ivar_Jacobson), 1992)
Use Case Diagrams: Basic Building Blocks

Example: Use Case Diagram of the ATM Use Case

Use Case Example: ATM Authentication

- **name**: Authentication
- **goal**: the client wants access to the ATM
- **pre-condition**: the ATM is operational, the welcome screen is displayed, card and PIN of client are available
- **post-condition**: client accepted, services of ATM are offered
- **post-cond. in exceptional case**: access denied, card returned or withheld, welcome screen displayed
- **actors**: client (main actor), bank system
- **open questions**: none

**Normal case**

1. client inserts card
2. ATM reads card, sends data to bank system
3. bank system checks validity
4. ATM shows PIN screen
5. client enters PIN
6. ATM reads PIN, sends to bank system
7. bank system checks PIN
8. ATM accepts and shows main menu

**Exception cases**

- **2a**: card not readable
  1. ATM displays "card not readable"
  2. ATM returns card
  3. ATM shows welcome screen
- **2b**: card readable, but not ATM card
- **2c**: no connection to bank system
- **3a**: card not valid or disabled
- **5a**: client cancels
- **5b**: client doesn't react within 5 s
- **6a**: no connection to bank system
- **7a**: first or second PIN wrong
- **7b**: third PIN wrong

(Ludewig and Lichter, 2013)
A User Story is a concise, written description of a piece of functionality that will be valuable to a user.

Use Case Example: ATM Authentication

1. The client wants access to the ATM
2. The ATM reads PIN
3. The bank system checks validity
4. The ATM shows PIN screen
5. The client enters PIN
6. The ATM reads PIN, sends data to the bank system
7. The bank system checks PIN
8. The ATM returns card

post-condition in normal case, client accepted, services of ATM are offered
post-condition in exceptional case, access denied, card returned or withheld, welcome screen displayed

estimation of effort may be difficult
agile spirit: strong dependency on competent developers
maybe best suited for changes/extension (after first iteration)
not designed to cover non-functional requirements and restrictions
objective/testable: by fixing test cases early
close contact to customer
easy to create, small units
maybe best suited for changes/extension (after first iteration)

As a [role] I want [something] so that [benefit].
A Brief History of Sequence Diagrams

- Message Sequence Charts, ITU standardized in different versions (ITU Z.120, 1st edition: 1993); often accused of lacking a formal semantics.
- Sequence Diagrams of UML 1.x (one of three main authors: I. Jacobson)
- SDs of UML 2.x address some issues, yet the standard exhibits unclarities and even contradictions (Harel and Maoz, 2007; Störrle, 2003).
- For the lecture, we consider Live Sequence Charts (LSCs) (Damm and Harel, 2001; Klose, 2003; Harel and Marelly, 2003), LSCs have a common fragment with UML 2.x SDs (Harel and Maoz, 2007).

\[
\text{LSC: buy water}
\]
\[
\text{AC: true}
\]
\[
\text{AM: invariant}
\]
\[
\text{I: strict}
\]
\[
\text{User}
\]
\[
\text{CoinValidator}
\]
\[
\text{ChoicePanel}
\]
\[
\text{Dispenser}
\]
\[
\text{C}_{50}\ \text{WATER}
\]
\[
\neg(\text{C}_{50} \lor \text{E}_{1} \lor \text{pS} \lor \text{pTEA} \lor \text{pFILL} \lor \text{water}_{\text{in}}_{\text{stock}})
\]
\[
\text{dWATER}_{\text{OK}}
\]
\[
\neg(\text{dSoft} \lor \text{dTEA})
\]
\[
\text{DurationConstraint}
\]
\[
\text{TimeObservation}
\]
\[
\text{TimeConstraint}
\]
\[
\text{DurationObservation}
\]
\[
\text{OMG}, 2007
\]
\[
\text{ITU-T}, 2011
\]
Informatik III

Comprehension of Abstract Syntax

From Concrete to Abstract Syntax

Cond ∈ L, φ

From Concrete to Abstract Syntax
The partial order \( \preceq \) and the simultaneity relation \( \sim \) of locations induce a direct successor relation on cuts of an LSC body as follows:

**Definition.** Let \( C \subseteq L \) be a cut of an LSC body \((L, \preceq, \sim, I, Msg, Cond, LocInv, \Theta)\).

A set \( \emptyset \neq F \subseteq L \) of locations is called fired-set \( F \) of cut \( C \) if and only if:

1. \( C \cap F = \emptyset \) and \( C \cup F \) is a cut, i.e., \( F \) is closed under simultaneity,
2. all locations in \( F \) are direct \( \preceq \)-successors of the front of \( C \), i.e., \( \forall l \in F \exists l' \in C \cdot l' \preceq l \land (\nexists l'' \in L \cdot l' \preceq l'' \preceq l) \),
3. locations in \( F \) that lie on the same instance line are pairwise unordered, i.e., \( \forall l \neq l' \in F \cdot (\exists I \in I \cdot \{l, l'\} \subseteq I) = \Rightarrow l \npreceq l' \land l' \npreceq l \),
4. for each asynchronous message reception in \( F \), the corresponding sending is already in \( C \), i.e., \( \forall (l, E, l') \in Msg \cdot l' \in F = \Rightarrow l \in C \).

The cut \( C' = C \cup F \) is called direct successor of \( C \) via \( F \), denoted by \( C \xrightarrow{\_} F \).
• Use-Cases:
  • interactions between system and actors,
  • be sure to elaborate exceptions and corner cases,
  • in particular effective with customers lacking technical background.

• Use-Case Diagrams:
  • visualise which participants are relevant for which use-case,
  • pretty useless without the underlying use-case.

• User Stories:
  • simple example of scenarios
  • strong point: naming tests is necessary,
  • weak point: hard to keep overview; global restrictions.

• Sequence Diagrams:
  • a visual formalism for interactions, i.e.,
    • precisely defined syntax,
    • precisely defined semantics (construct automaton from abstract syntax)
  • Can be used to precisely describe the interactions of a use-case.

References


