

Requirements in Natural Language Specification Techniques

As a diabetes patient, I would like to register observations to memorize what happened.

As a diabetes patient, I would like to register biometrics to monitor myself.

As a diabetes patient, I would like to read about the experiences of peers to learn how to handle my condition.

As a diabetes patient, I would like to read treatment advice to learn how to handle my condition.

As a physician, I would like to see an in-depth, contextualized record of the patient's diabetes-related parameters to offer personalized treatment.

As a nurse, I would like to see biometrics to help the patient managing the day-to-day issues with disease management.

As a nurse, I would like to...



Communication of Requirements



Communication of Requirements

People differ in their implicit knowledge about culture, context, and events that have happened in the past. These differences imply hidden assumptions, wrong beliefs, omissions of apparently unnecessary information, and ambiguities in how statements may be understood.



Learning Objectives

Understand

- Causes for misunderstandings of requirements
- Role of glossary
- Role of decision tables

Be able to do the following activities

- Specify requirements with Simple Active Sentences
- Specify requirements with User Stories
- Specify requirements with Rupp's Shall-Templates

Example: Extracts from an Ambiguous Specification

The following are requirements extracted from a specification for a system called “Universal e-Commerce Platform (EP)”. The EP is a development framework with standard functionality for building online shops like Amazon. The EP is to be developed, not the shop.

- *The e-commerce platform (EP) gives customers rapid ramp-up their own web-based selling channel.*
- *The EP must integrate with established inventory and dispatch systems and must also communicate credit card information to the credit card processing company for validation before an order is accepted.*
- *Customers are presented with a summary of their order with a shipment address. If they click on “confirm” to confirm it, then the system asks them to log in if they have not already done so.*
- *Ideally, the checkout should recognize the customer in which case the log in is automatic.*
- *All new customers must fill out a form.*

Exercise

Goal: understand common causes for misunderstanding about requirements.

Assume you are a developer of the EP, have received these requirements, and are expected to deliver a system that satisfies exactly these requirements.

Tasks: for each requirement,

- Study the requirement.
- Reflect how you would implement the requirement. You do not need to come up with a design; the idea is to take the perspective of a developer and identify the problems that the ambiguity may generate.
- Explain what is wrong in the specified requirement that caused the potential misunderstanding.

Common Defects of Natural-Language Requirements

Nominalization

Nouns without reference index

Universal quantifiers

Incompletely specified conditions

Incompletely specified process verbs

Nominalization

By means of nominalization, a (sometimes long-lasting) process is converted into a (singular) event. Information necessary to accurately describe the process is thereby lost.

Examples

- *to ramp-up*:

 - deployment of an e-commerce site is a rather complex process

- *credit card validation*:

 - again a complex process with many alternatives of how this could be done

Nouns without Reference Index

As with process verbs, nouns are frequently incompletely specified or replaced with a placeholder. Linguists call this a missing or inadequate index of reference.

Examples

- *established existing inventory systems*:
which ones exactly?
- *must fill out a form*:
what form? Which information is to be collected?
- *the user, the controller, the system, the message, the data, the function, ...*:
which one exactly?
- *it, they, this, them, ...*:
which one?

Universal Quantifiers

Universal quantifiers specify amounts or frequencies. They group a set of objects and make a statement about the behavior of this set. When using frequencies, there is the risk that the specified behavior or property does not apply to all objects within the specified set.

Examples

- *All new customers must fill out a form:*

Does this apply for the customers the system does not know or the customers the vendor does not know?

- *never, always, no, none, every, all, nothing, ...:*

There will be exceptions, I promise you.

- *ideally, some, sometimes, ...:*

There will be exceptions, I promise you.

Incompletely Specified Conditions

Incompletely specified conditions are an indicator of a potential loss of information. Requirements that contain conditions specify the behavior that must occur when the condition is met. In addition, they must specify what behavior must occur if the condition is not met (the part that is often missing).

Examples

- *if they click on confirm...:*
what if they do not do so?
- *communicate credit card information...for validation before an order is accepted:*
what if validation fails?
- *If, in case, whether, depending on, before, after, ...:*
the *else* statement is a must

Incompletely Specified Process Verbs

Some process verbs require more than one object to be considered completely specified.

Examples

– *to check out*.

requires a definition of at least

- 1) who is checking out and paying,
- 2) what is checked out,
- 3) where the products should be shipped to, and
- 4) how the order is to be paid.

The problem is here that the required objects may only be known to people with experience in the domain, feature, or functionality.

Solutions to the Linguistic Problems

Nominalization

- Language templates that force good sentence structure

Nouns without reference index

- Glossary that defines the vocabulary used in the specification

Universal quantifiers

- Specify standard case and all exceptions

Incompletely specified conditions

- if ... then ... else ...
- Decision tables

Incompletely specified process verbs

- Language templates that force good sentence structure
- Domain knowledge that tells you what needs to be specified

Language Templates for Specification of Functional Requirements

To force good sentence structure, a language template is used for guiding the writing of a requirement. There are many templates; you have to choose the one that suits your style and situation best.

Simple Active Sentences

Beck's and Cohn's User Stories

Rupp's Shall Templates

Rolls-Royce/Intel Easy EARS

- Not part of this course (but you may read about it if you are interested)

Behaviour Driven Development (BDD)

- Approach to specify tests for Cohn's User Stories

Use Case Templates

- See UML course blocks

Exercise

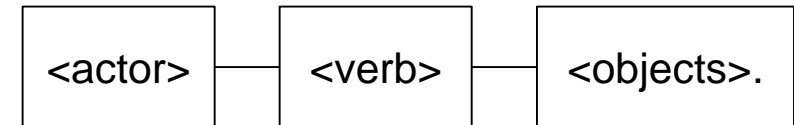
Goal: be able to specify functional requirements by using common language templates.

Tasks (work in your mini-project team):

1. Select your “killer” feature. Agree on the meaning of the feature by stating a) what it is, b) why it is needed, and c) the key idea of how it should be implemented.
2. Brainstorm to identify all functional requirements for that feature. Write them down in keywords. Give each requirement an identifier (R-001, R-002, etc.)
3. Train yourself in using the language templates by writing down one functional requirement after the other with the
 - Simple active sentence template
 - User Story template
 - Shall template
4. Select in your team the single template you want to use for your mini project. Justify your decision.

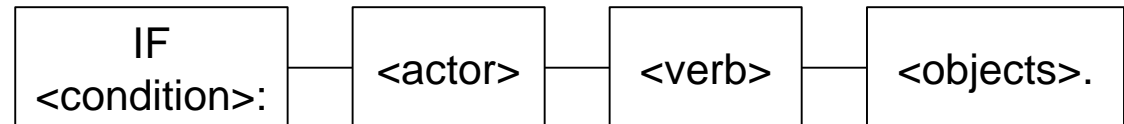
Simple Active Sentences

The sentences are structured as follows:



- Simple: only the main clause.
- Active: ensure the initiating actor is stated and enable completeness check.
- Avoid plural: use singular as far as possible.

If conditions are unavoidable:



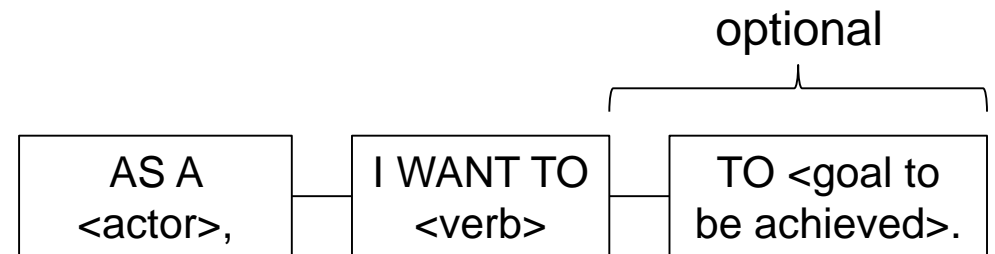
- Ensure that all alternatives are specified

Examples

- *The credit card processing company validates the credit card.*
- *EP displays an order summary to the customer.*
- *IF the customer is new: the customer fills out the customer registration form.*
IF the customer is NOT new: EP logs the customer in.

Becks' and Cohn's User Stories

The sentences are structured as follows:



–Written always from the perspective of the (human) user.

This template is widely used in Agile development.

–Complements: **Front** →
Clarifications,
Test cases

–Variations do exist

Back →

As a customer, I want to pay with a credit card to avoid lengthy and complicated bank transfers.

Note: do we accept Barclaycards?

Note for GUI: no field for card type necessary (can be derived from the two first credit card number digits).

Test with Via, MasterCard, and Amexco (pass)

Test with Diners Club (fail)

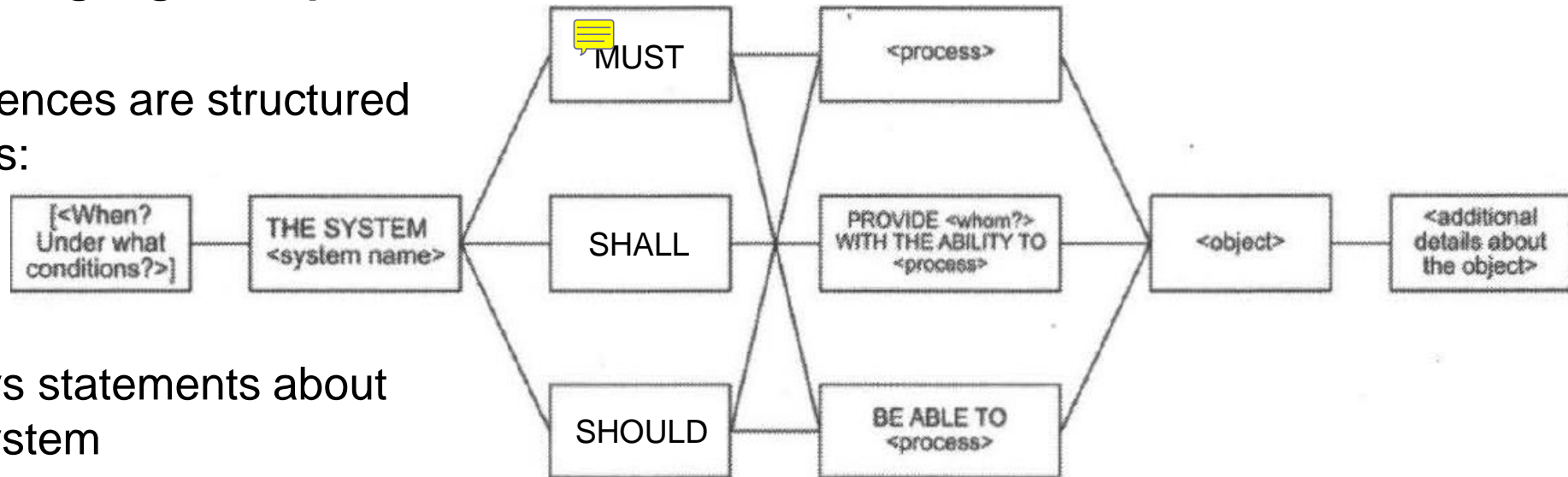
Test with correct, wrong, and missing CC number

Test with expired cards (fail)

...

Rupp's Language Template

The sentences are structured as follows:



- Always statements about the system

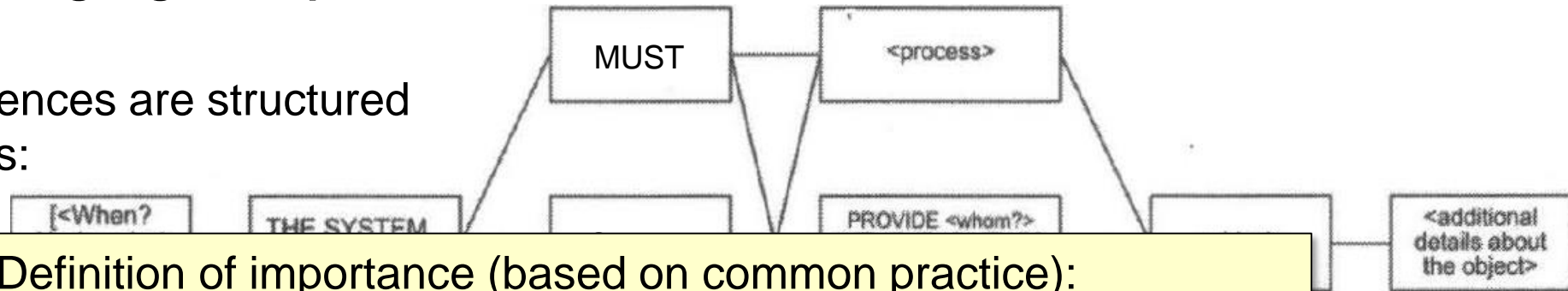
Example: *The system EP shall be able to transfer credit card information to the credit card processing company.*

This template is used in public tenders.

- Note that common practice and recommendations for requirements importance differs from the suggestions in Rupp, Pohl (2011): Requirements Engineering Fundamentals.

Rupp's Language Template

The sentences are structured as follows:



Definition of importance (based on common practice):

MUST: There is no way around this functionality

– Always the system
SHALL: A workaround exists if the functionality is missing, but the workaround is painful.

Example credit card
SHOULD: The functionality is purely optional; the stakeholders can live without it.

This term
– Note: definitions are consistent with Pohl and Rupp (2011), but the terminology is inconsistent.

differs from the suggestions in Pohl and Rupp (2011).

Examples

Simple active sentences

- *The credit card processing company validates the credit card.*
- *EP displays an order summary to the customer.*

Beck's and Cohn's User Stories

- *As a credit card processing company I want to receive credit card information to validate the credit card.*
- *As a customer I want to see an order summary to know what I order.*

Rupp's Language Templates

- *The system EP shall be able to transfer credit card information to the credit card processing company.*
- *The system EP must provide the customer with the ability to see an order summary.*