Software Requirements

Xin Feng

Outline

- Requirements types
- Requirements specification
- What is a good specation

Requirements

- Describe system services (服务) and the constraints (约束)
- It can be either very abstract (抽象) or very detailed
- Functions in the requirements are used to
  - be shown to the contract bidders （投标人 ）
  - validate （确认）the final system

Service and Constraints

- Services are the functions that a system should implement,
  - e.g,
    - Show a menu
    - Adjust font size
    - Display a picture
- A constraint is some terms that a system should follow,
  - e.g.,
    - Implemented in C++ language
    - Run with Windows XP
    - Online

Abstract and Detailed Requirements

- Abstract
  - A file reader should be developed to read a file
- Detailed
  - A file reader should be able to open a file, read a file and close a file. Then fonts are supported. The size can be changed. The file size is under 64k. The file can have diagrams(图例), tables, bullets(项目符号). The reader will be run under Windows XP. The reader is implemented in C language.

Can you figure out a more detailed version???

Requirements Types

- Categories by
  - Readers
    - User requirements
    - System requirements
  - Contents
    - Functional requirements
    - Non-functional requirements
    - Domain requirements （基于特定知识背景的需求）

Requirements Types

- Categories by
  - Readers
    - User requirements
    - System requirements
  - Contents
    - Functional requirements
    - Non-functional requirements
    - Domain requirements （基于特定知识背景的需求）

User Requirements

- Usually are written in natural language (自然语言)
- Diagrams （图表） of the services may be provided
- Specify necessary operational constraints
- Abstract
- Written for customers by either customers or developers.

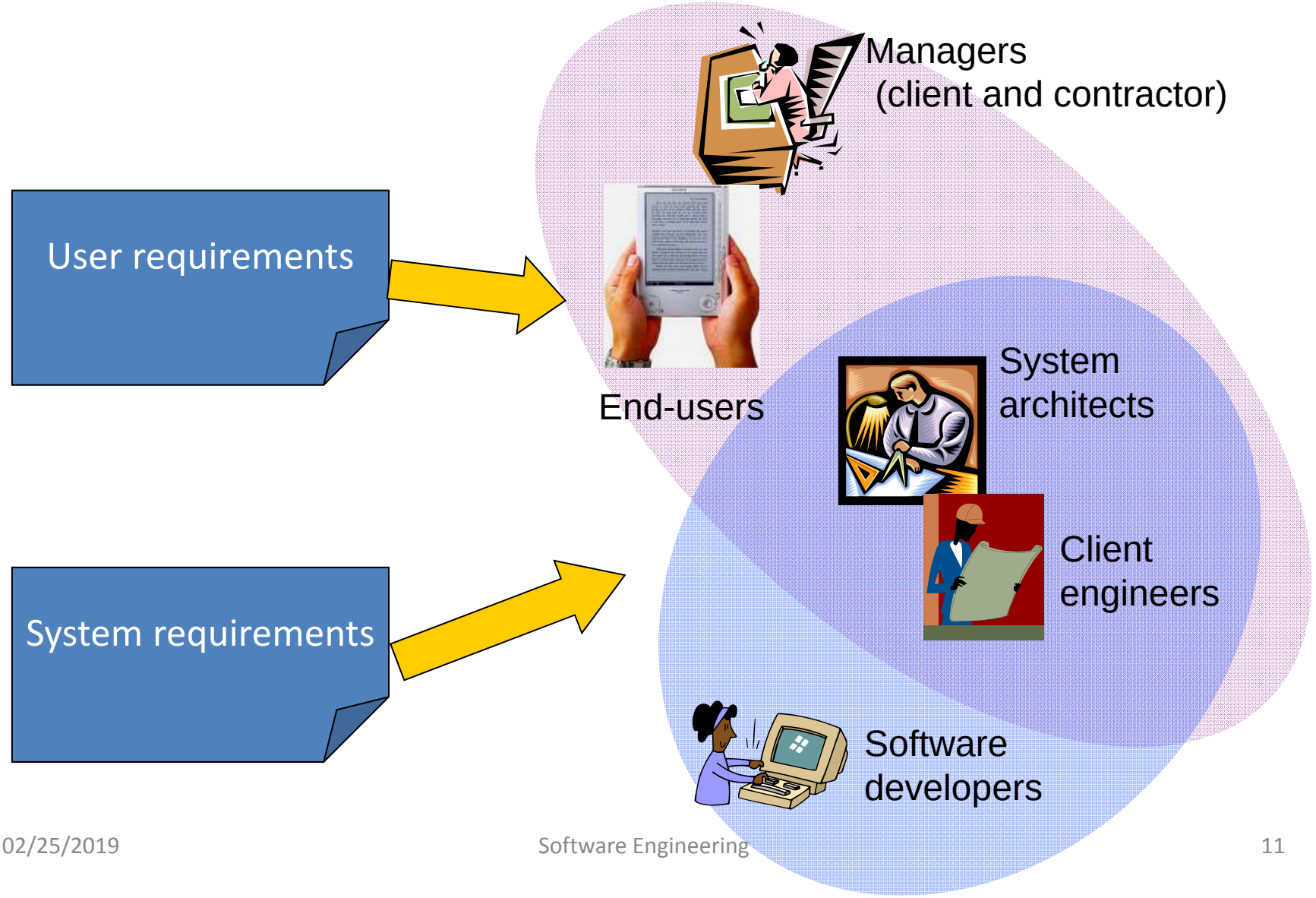
Readers: User Requirements

Who will read user requirements???

Readers: System Requirements

- Described in a structured document , in diagrams, tables...
- Detailed descriptions of the system's functions, services and operational constraints
- Defines what should be implemented
- Written for developers by developers
- System requirements specification （规格说明）.

Readers (in normal cases)



User Requirements and System Requirements

LIBSYS shall keep track of all data required by copyright licensing agencies (许可代理)

User

1.1 On making a request for a document from LIBSYS, the requestor shall be presented with a form that records details of the user and the request made
1.2 LIBSYS request forms shall be stored on the system for five years from the date of the request
1.3 All the LIBSYS request forms must be indexed by user, by the name of material requested and by the supplier of the request
1.4 LIBSYS shall maintain a log of all requests that have been made to the system
1.5 For material where authors lending rights apply, loan details shall be sent monthly to copyright licensing agencies(许可代理) that have registered with LIBSYS.

System

Requirements Types

- Categories by
  - Readers
    - User requirements
    - System requirements
  - Contents
    - Functional requirements
    - Non-functional requirements
    - Domain requirements （基于特定知识背景的需求）

Contents: Functional Requirements

- Describe functionality （功能） or system services
  - functional user requirements
    - may be high-level statements of what the system should do
  - functional system requirements
    - describe the system services in detail
- Input/Output
- Behaviours

Requirements Can Be Imprecise (不精确)

- Ambiguous （二义）
  - Interpreted in different ways by different persons
- Incomplete
  - Some information is lost
- Inconsistency （不一致）
  - Some descriptions conflict （冲突）

An Example

Write a program F with two inputs A and B
1. If A is greater than B, add A and B
2. If A is less than B, output A – B
3. If A equals 0, output 0

An Example

Write a program F with two inputs A and B
1.If A is greater than B, add A and B
2.If A is less than B, output A – B
3.If A equals 0, output 0

Ambiguous : F(int A, int B) or F(float a, float b) or…

Incomplete: What if A = B

Inconsistency: What is the output of F(0, 3)

Ambiguous Or Not?

1.1 On making a request for a document from LIBSYS, the requestor shall be presented with a form that records details of the user and the request made

1.2 LIBSYS request forms shall be stored on the system for five years from the date of the request

1.3 All the LIBSYS request forms must be indexed by user, by the name of material requested and by the supplier of the request

1.4 LIBSYS shall maintain a log of all requests that have been made to the system

1.5 For material where authors lending rights apply, loan details shall be sent monthly to copyright licensing agencies that have registered with LIBSYS.

Contents: Non-Functional Requirements

- Specify the system/system process properties （性质 ）and constraints
  - Product requirements
    - Specify product behaviours
    - E.g., performance （性能）, reliability （可靠）…
  - Organizational requirements
    - Policies （政策）and procedures in the customer's and developer's organisation.
    - E.g., process standards, implementation requirements …

Contents: Non-Functional Requirements

- External （外来的）requirements
  - All the requirements external to the system and its development process
  - E.g., legislative （立法的） requirements

Non-Functional: Product Requirements

- Usability
  - Example: keyboard/mouse
- Efficiency (有效性)
  - Performance (性能)
    - E.g., response time 0.001 sec.
  - Space
    - E.g., memory
- Reliability (可靠性)
  - E.g., mean time to failure: 2000 hrs
- Portability(可移植) requirements
  - E.g., run on both Mac OS and Windows

Non-Functional: Organizational Requirements

- Delivery （交付）
  - Example: delivery time, delivery documents
- Implementation
  - Example: C programming language
- Standards
  - Example: meet IEEE standard, ISO.

Non-Functional: External Requirements

- Interoperability （两不同程序共同工作交通的能力）
  - Example: the input/ouput date format dd/mm/yyyy
- Privacy （隐私）
  - Example: the copyright
- Security
  - Example: The system cannot show the bank staff the customer's password
- Ethical （伦理）
  - Example: Avoid the coarse （粗俗）language

Verify Non-Functional Requirements

- It is hard to specify and verify some non-functional requirements
  - Ease of use
- It is likely that some non-functional requirements conflict
  - Time and space
- Try to specify these requirements in a quantitative (量化) way (not qualitative(质的))
  - Fast, how fast?!

Metrics

Property	Measure
Speed	Response time, second
Size	Mbytes, LOC
Ease of use	Training time (hrs) Number of help frames
Reliability	Mean (平均) time to failure (MTTF) Rate of failure occurrence
Robustness	Percentage of failure Restart time
Portability	Number of target systems

An Example

Develop an ATM system for a bank ….

Response time to a user's request: *** micro-second
Programming language: ***language
MTTF: … days
Restart time on failure: …microsecond
Training time: … hrs

Contents: Domain Requirements

- The application domain of system
- The characteristics （特性）of that domain
- If domain (领域) requirements are not satisfied, the system may be unworkable
- Problems
  - No common background for both customers or developers
  - Confusion （混淆）with functional requirements
  - Difference in profession makes one feel worlds apart (隔行如隔山).

Requirements ***Specification***

- Specify the users' requirements in a document. The specification can use
  - natural language
  - structured natural language
    - Define standard forms or templates
  - diagrams
    - Use graphical notation （标志）
  - tables
    - Define table types
  - formal language
    - Specify the requirements using mathematics

An Example

Develop an ATM system so the customers of the bank can inquire, withdraw(提取). If the card is illegal, reject the card. If the card is legal and if the balance is less than the amount to withdraw, display warning information; otherwise, do the operations that a user requires.

Natural Language ***Specification***

Develop an ATM system so the customers of the bank can inquire, withdraw(提取). If the card is illegal, reject the card. If the card is legal and if the balance is less than the amount to withdraw, display warning information; otherwise, do the operations that a user requires.

Structural Natural Language

Specification

System: ATM
Description: the customers of the bank can inquire, withdraw. If the card is illegal, reject the card. If the card is legal, handle user's requirements.
References:
Inquire: F1
Withdraw: F2
Input: card, selection, amount
Output: information, deliver banknote

F1
Function: Inquire
Description: Display a customer's account balance.
References:
N/L
Input: card
Output: information

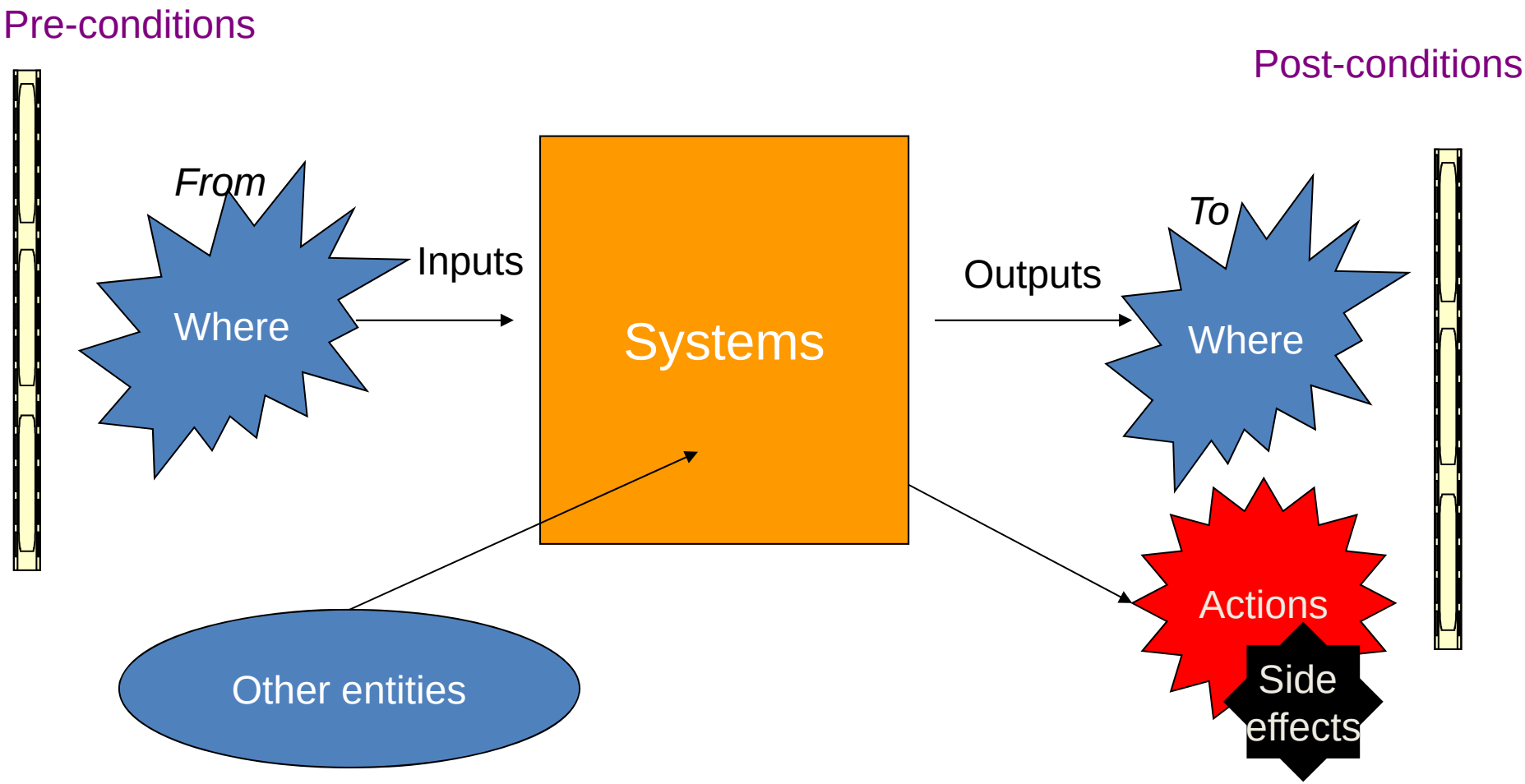
F2
Function: Withdraw
Description: Deliver the amount that a customer requests if the amount is less than the balance
References:
N/L
Input: card, amount
Output: deliver banknote, information

Structural Natural Language
Specification

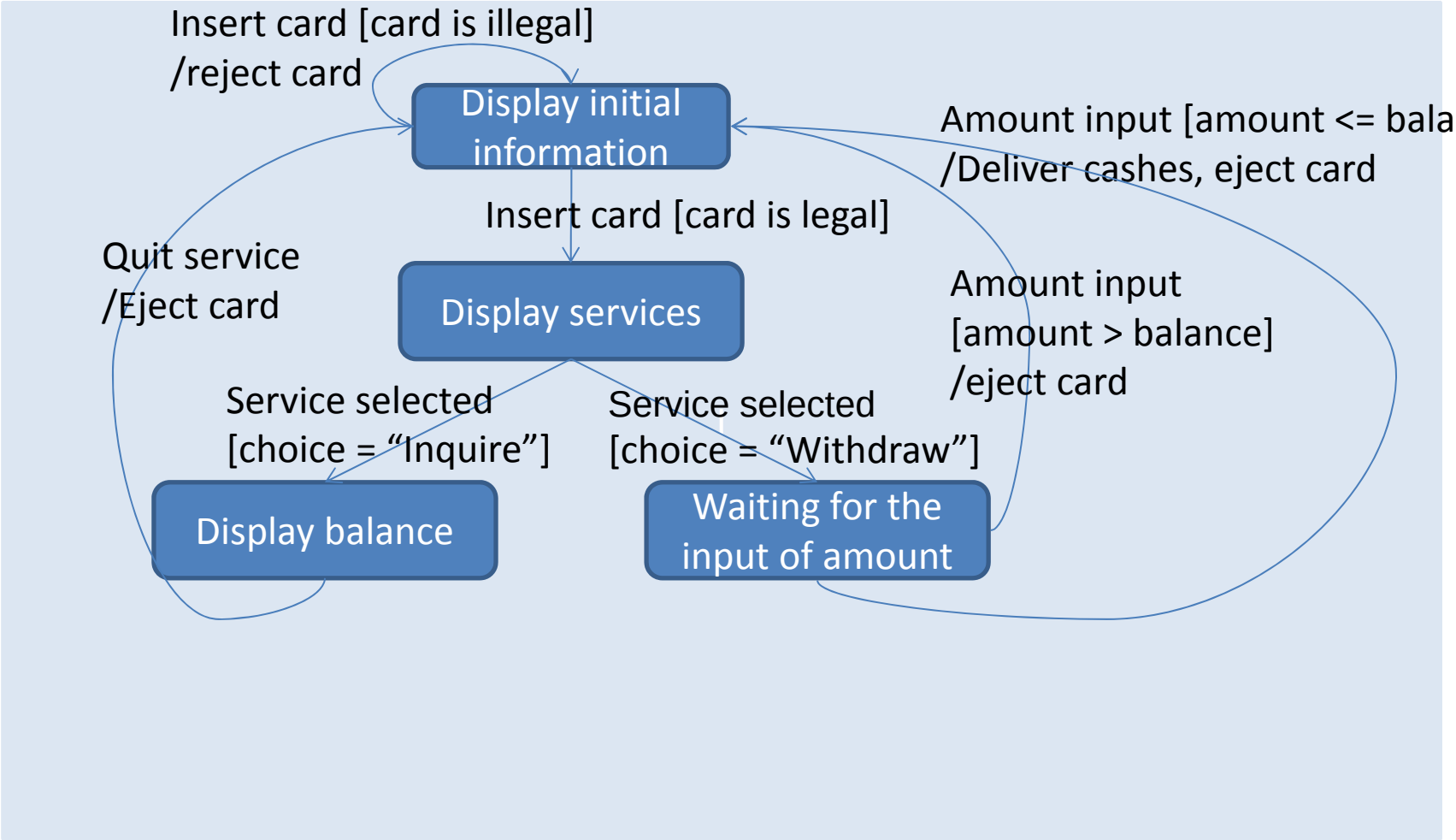
- Description of the function or entity （实体）being specified
- Description of its inputs and where they come from
- Description of its outputs and where they go to
- Indication （指示）of what other entities are used
- Description of the action to be taken
- Pre-and post- conditions （前置和后置条件）for the function to be used
- Side effects （副作用）

Structural Natural Language

Specification



Graphical Notation ***Specification***



Tabular （表） ***Specification***

<table>
<tr><th colspan="3">Card is legal</th><th>Card is illegal</th></tr>
<tr><td rowspan="2">Selection = “Inquire”</td><td colspan="2">Selection = “Withdraw”</td><td rowspan="2"></td></tr>
<tr><td>Amount > balance</td><td>Amount <= balance</td></tr>
<tr><td>Display balance</td><td>Display information</td><td>Deliver cashes</td><td>Display information</td></tr>
</table>

Formal language （形式语言）***Specification***

obj QUEUE is

Sort List

Sort Element

Import INTEGER

---

op Create: -> List

op Construct: List Element -> List

op Head: List -> Element

op Length: List -> Integer

op Remove: List -> List

---

var L: List

var v: Element

---

eq Head(Create) = Undefined exception (empty list)

eq Head(Construct(L, v))) = if L = Create then v else Head(L)

eq Length(Create) = 0

eq Length (Construct(L, v)) = Length (L) + 1

eq Remove (Create) = Create

eq Remove(Construct(L, v)) = if L = Create then Create else Construct(Remove(L), v))

Class Exercise

- Requirements:
  - Calculate postage(邮费), if the weight is not over 50g, the postage is 50c for a postcard or a letter, 95c for a large envelope. If the weight is between 50 and 100, the postage is 60c for a postcard or a letter, 125c for a large envelope. If the weight is between 100g and 150g, the postage is 80c for a postcard, 155c for a large envelope. No postcard or a letter whose weight is over 150g is accepted. If the weight of a large envelope is over 150g, the postage is the weight × 2c

Questions:

1. Are there any problems with the requirements?
2. Is there other way to re-describe the requirements to convenience checking of the ambiguity, consistency or redundancy?

Guideline on the Writing of **Requirements Specification**

- Consistency （一致）
  - Format
    - Work out the standard style in documenting （编写文档） the requirements
  - Language
    - Same language
  - Content
    - As precise （精确） and possible
- Readable
  - Avoid use of jargon （行话）
  - Understand the differences in backgrounds
  - Highlight the key parts
  - References （引用）
  - Employ tables and diagrams
- No design information

No Project Requirements in the SRS

- The following information should NOT be included in an SRS (Software Requirement Specification)
  - Cost
  - Delivery schedules
  - Reporting procedures
  - Software development methods
  - Quality assurance
  - Validation and verification criteria
  - Acceptance procedures.

Parts in SRS

Table of Contents

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, acronyms, and abbreviations
   - 1.4 References
   - 1.5 Overview
2. Overall description
   - 2.1 Product perspective
   - 2.2 Product functions
   - 2.3 User characteristics
   - 2.4 Constraints
   - 2.5 Assumptions and dependencies
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)

Appendixes

Index

A Template on Section 3

3. Specific requirements
- 3.1 External interface requirements
  - 3.1.1 User interfaces
  - 3.1.2 Hardware interfaces
  - 3.1.3 Software interfaces
  - 3.1.4 Communications interfaces
- 3.2 Functional requirements
  - 3.2.1 Mode 1
    - 3.2.1.1 Functional requirement 1.1
    - .
    - .
    - .
    - 3.2.1.*n* Functional requirement 1.*n*
  - 3.2.2 Mode 2
  - .
  - .
  - .
  - 3.2.*m* Mode *m*
    - 3.2.*m*.1 Functional requirement *m*.1
    - .
    - .
    - .
    - 3.2.*m*.*n* Functional requirement *m*.*n*
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes
- 3.6 Other requirements

Assignment 2

- Can you give examples to explain software requirement ambiguity, inconsistency, incompleteness respectively (the example should not be from the lecture)?
- Use a table to describe the types of software requirements.
- Deadline: 11:55 pm 6 Mar 2019

Summary

- Requirements can be described in different forms.
- There are different types of requirements.
- Specification should contain proper contents.