

# Identifying Stakeholders and Their Preferences about NFR by Comparing Use Case Diagrams of Several Existing Systems

September 9, 2004

**Haruhiko Kaiya**

Shinshu University, Japan

# Outline

- Background and Purpose of this Research
  - purpose is to develop a req. elicitation method.
- Inputs and Outputs of our Method
- How to use the Outputs
- Assumption and Basic Idea
  - Goal Oriented Req. Analysis
  - Goal Question Metrics (GQM)
- Procedure
- A Case Study
- Conclusions and Future Works

# Background

- Existing similar systems help us to develop new system.
- Of course, differences between the new system and the existing systems are needed so as
  - to hold a dominant position with the new system,
  - not to infringe others copyright, and
  - to fit the new system into a business context where it will be used.
- **We have to clarify above BEFORE defining and writing requirements specification.**

# Purpose of this Research

## Building a method

- to clarify the goals of new system, and
- to clarify the context of the new system in contract with existing systems.

- Existing Systems

- ⇒ Requirements Specification written in Use Case Diagrams (UCD)

- Goals

- ⇒ Non-Functional Requirements (NFR)

- Context

- ⇒ Stakeholders

# Inputs/Outputs of our Method

- Inputs
  - Customers of the new System.  
through interview and/or questionnaires.
  - Domain (or Kind) of the new system.  
‘What kind of system do you want?’
  - Repository for Use Case Diagrams in a same domain.
- Outputs
  - Goals representing Non-functional requirements (NFR)
    - NFR mainly become modifier of functional requirements (Use Cases).
  - Stakeholders and their Preferences that are not identified before applying this method.

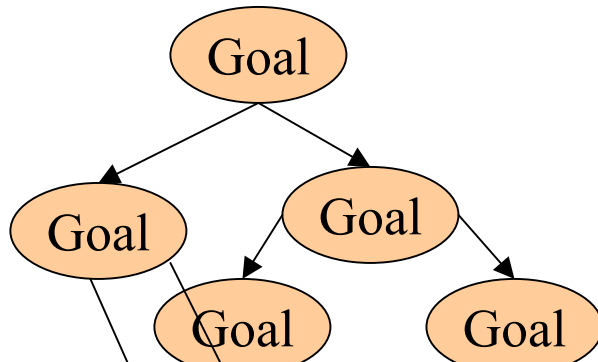
# How to use the Outputs

- Resources for writing software requirements specification (SRS).
- The outputs allow us
  - to clarify the advantages and characteristics of the system to be developed.
    - For each function, we can clearly state what its advantages are and why.
  - to find and contact person who will give additional resources for the system.
    - If we cannot, we may assume their demands.
- Reuse requirements, design and implementation of existing systems in the repository.
  - The repository is not an output but we can reuse it.

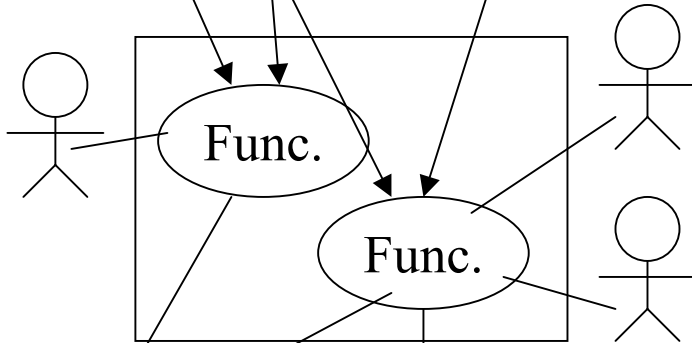
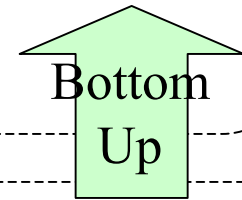
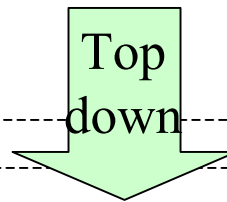
# Assumption for building the method

- It is difficult to identify the advantages of new system from business goals in the top down manner.
  - Rather, business goals are derived from information about existing systems.
- We are sensitive to the changes and differences.
  - By changing and/or comparing requirements,
    - Someone becomes aware that he/she gains/loses something by the system.
    - Some characteristics can make the system to be more valuable than before or others.
- It is easy to compare existing systems because they are concrete and they exist.
  - On the other hand, it is often vague when we discuss abstract things; e.g. goals, objectives, policies.

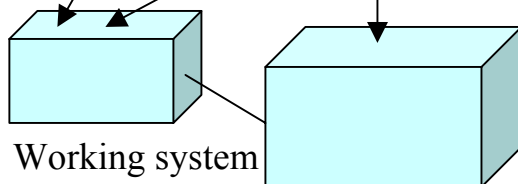
# Basic Framework for Building the Method



Representing *why* the system is required and/or valuable.  
NFR are written directly in this level.



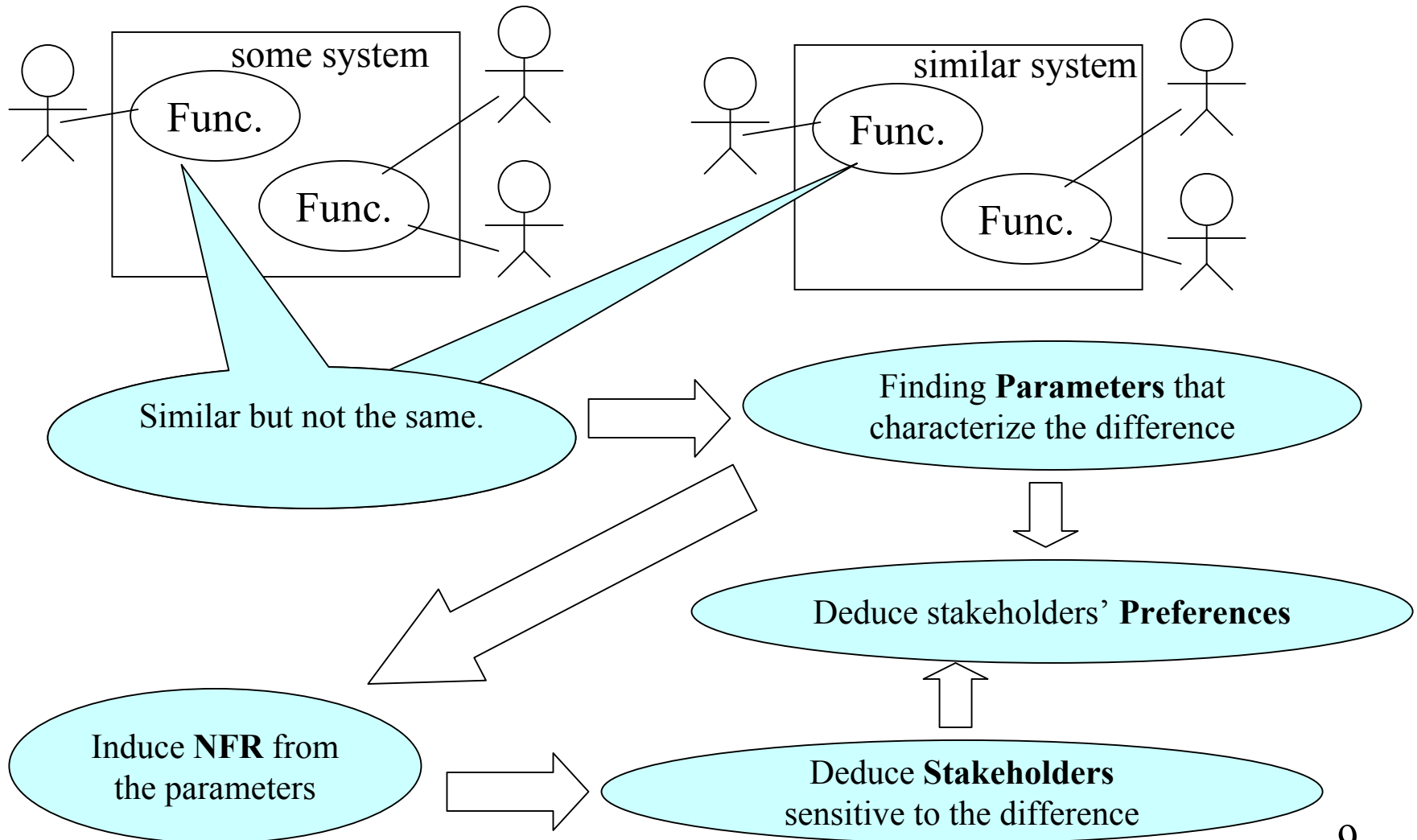
Representing *what* the system do.  
NFR are implemented as several function(s).



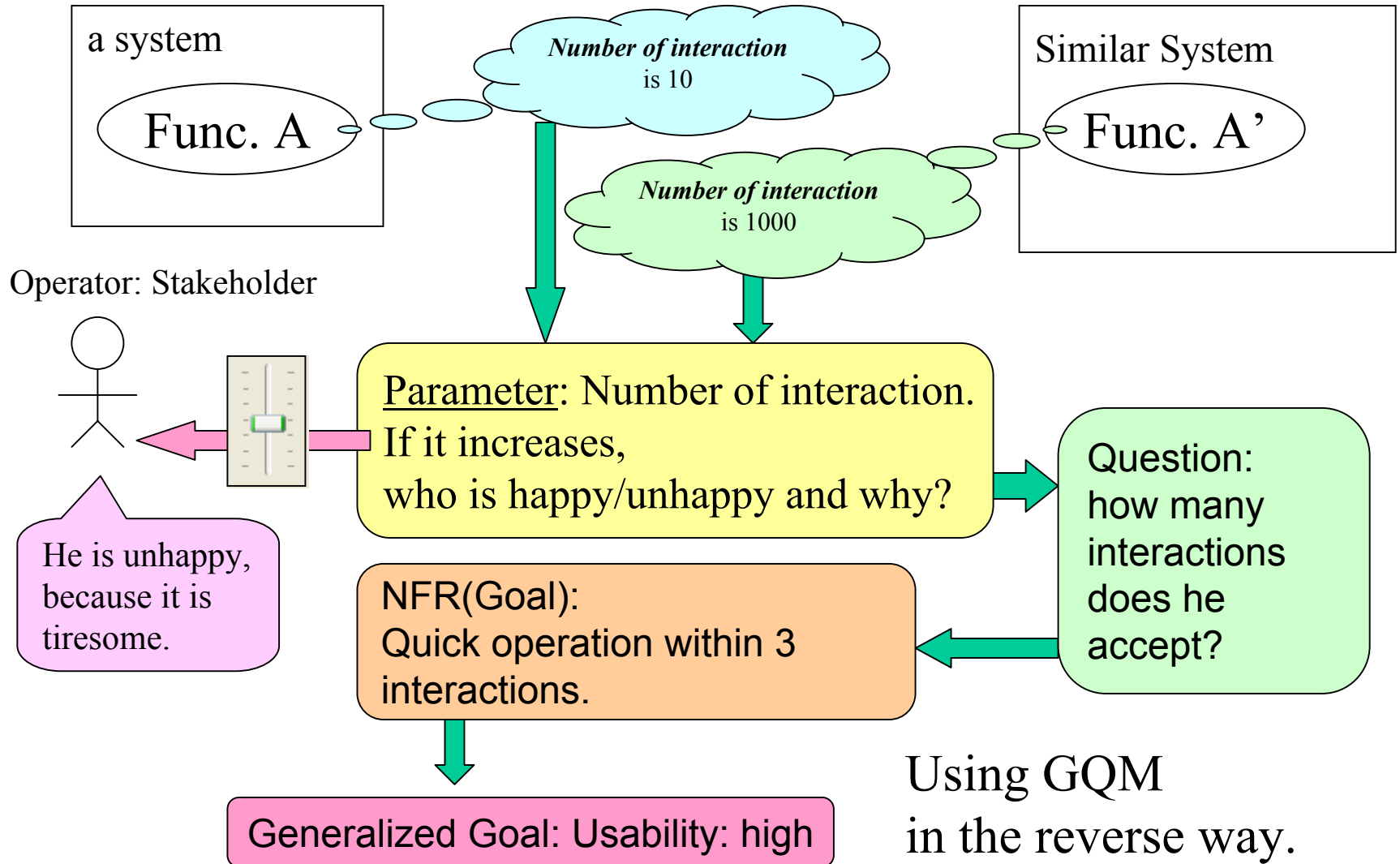
Representing *how* the system works.  
*out of scope in this research.*



# How to find Goals and Stakeholders?



# Simple Example



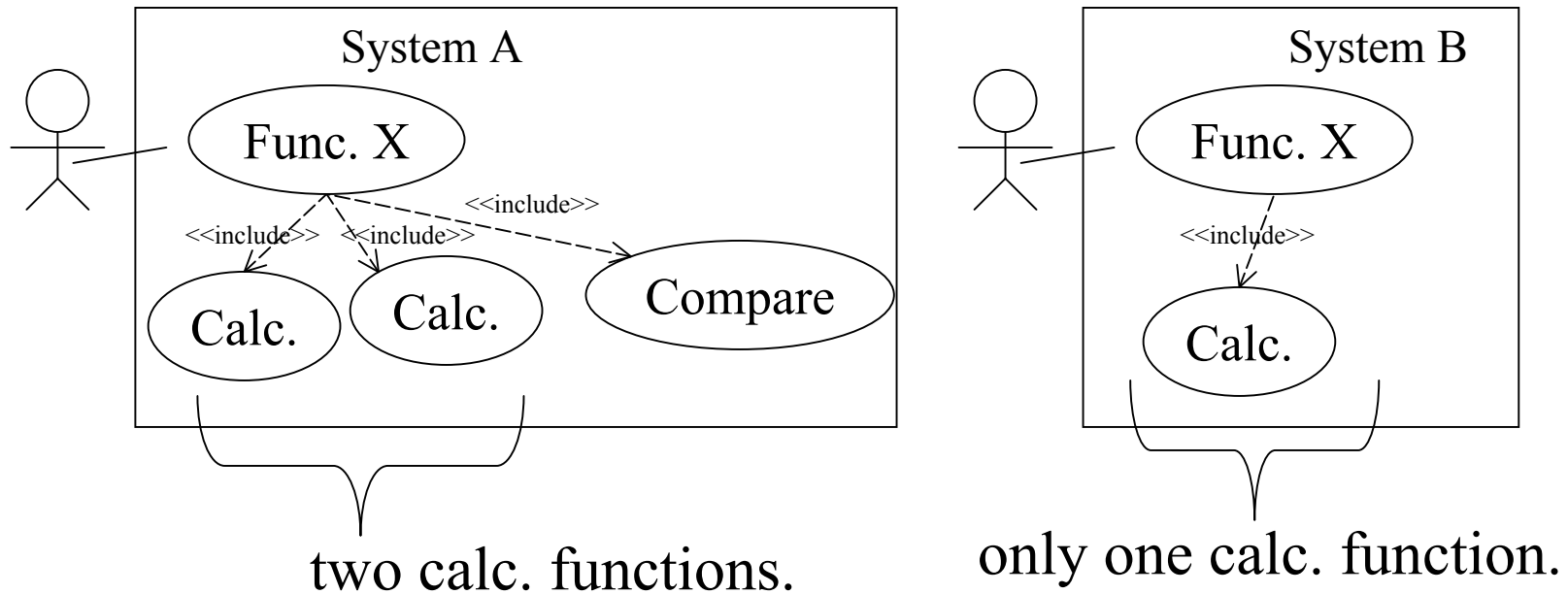
# How to identify similar use cases?

- In advance, use case diagrams (UCD) have to be collected.
- We should *subjectively* decide whether a UCD belongs to a domain.
  - e.g. a domain for mail client applications.
  - The similarity among use cases and actors helps us to decide above.
- We should also *subjectively* decide whether use cases or actors are similar or not.
  - by using words in each use cases' name.
  - We don't use internal description (scenario) of use cases now.

# How to identify differences among use cases?

- We focus on the structure among use case diagrams (UCD), mainly the *surrounding* of each use case.
- **Surrounding of a use case:**
  - set of use cases and actors that are directly connected to the use case.
- We identify differences by comparing surroundings of two (or more) use cases.

# Simple Example



Variable (Parameter): 'the number of calculation'.

Type of Variable: natural number

NFR (Goal): Reliable Calculation for Function X

Stakeholders and their Preference: When the value increases,  
customer: Stakeholder is happy because the result is more correct, but  
developer: Stakeholder is unhappy because it's hard to implement.

# Inputs/Outputs of our Method

- Inputs
  - Customers of the new System.  
through interview and/or questionnaires.
  - Domain (or Kind) of the new system.  
‘What kind of system do you want?’
  - Repository for Use Case Diagrams in a same domain.
- Outputs
  - Goals representing NFR
    - NFR mainly become modifier of functional requirements (Use Cases).
  - Stakeholders and their Preferences that are not identified before applying this method.

# The outline of procedure

1. Collect several use case diagrams (UCD) in the same or similar domain.
  - Write UCD by referring manuals and/or helps.
2. Select similar use cases from more than two UCD.
3. Identify variables based on their surrounding.
4. Identify stakeholders and their preferences by changing the values of variables.
5. Identify Non-Functional Requirements (NFR)
  - by generalizing the change of values, and
  - by using existing categorization of NFR for this generalization.

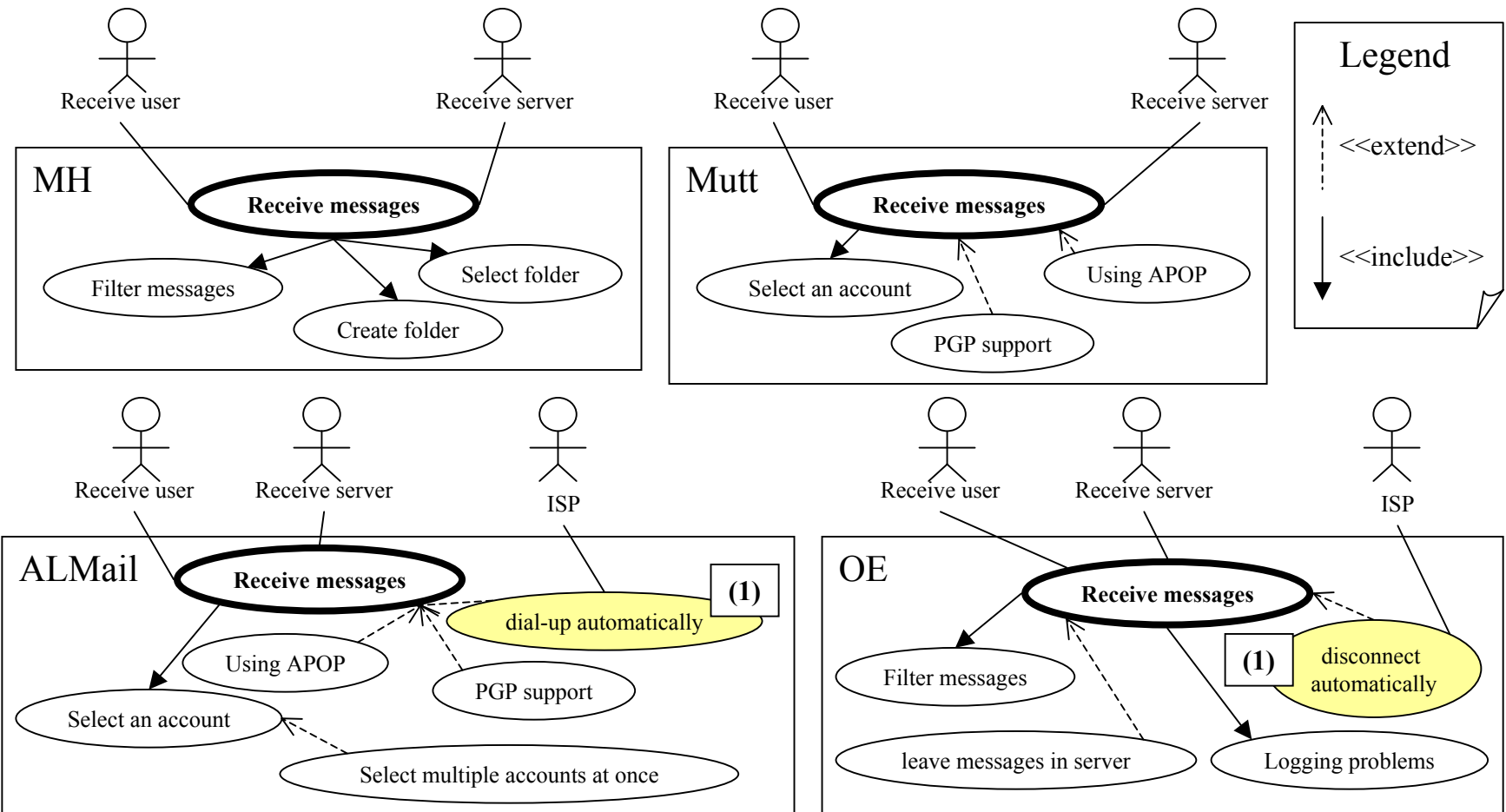
e.g. ISO/IEC9126, NFR framework[6].

# A Case Study

- Domain: Mail Clients (Mail User Agent, MUA)
- Existing MUA
  1. Outlook Express (OE): typical MUA in Windows
  2. Message Handler (MH): Good old days MUA running on UNIX. Set of commands.  
I still use this, I love this.
  3. AL-Mail: running on Windows, made in Japan.
  4. Mutt: Interactive MUA mainly running on UNIX.  
This works on the simple textual terminal (e.g. VT100).
- Most functions are similar, but not the same.



# Example of Differences: Receive messages

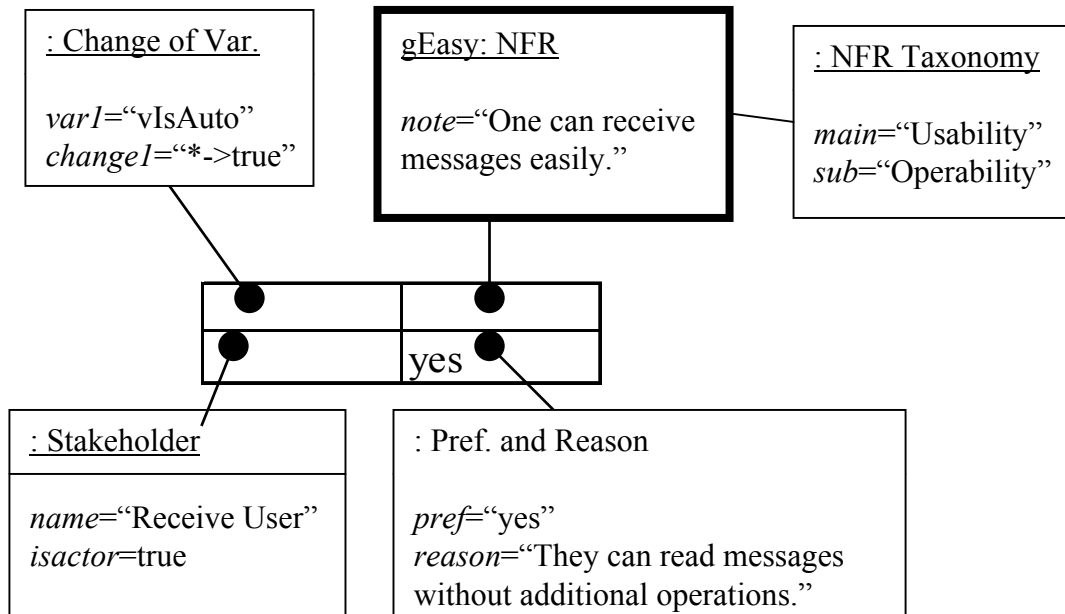


Differences are about functions for managing connection of communication. 17

# Variable and Related Preference

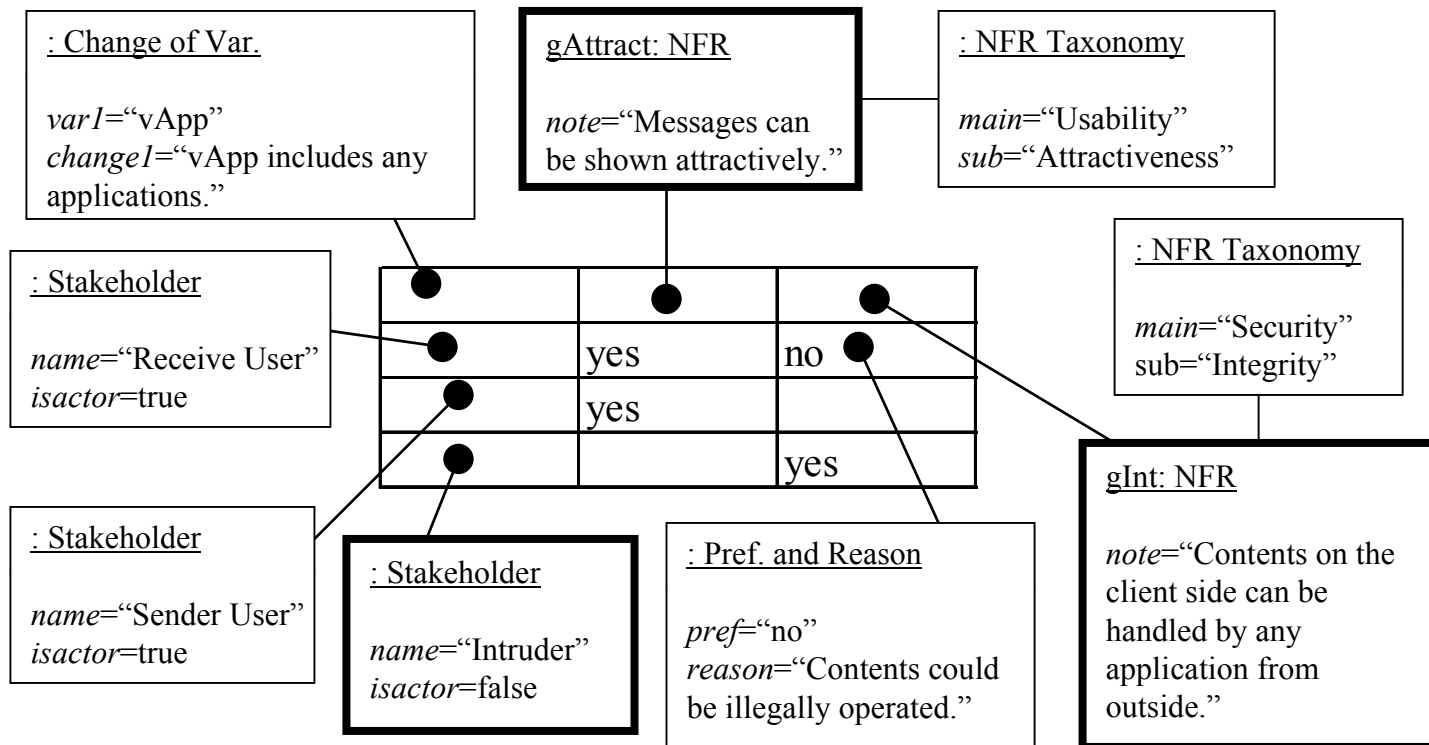
Variable = vIsAuto: boolean

whether the connection is automatically  
connected/disconnected or not.



# Another Example

For one change,  
many stakeholders and many NFR.



Variable = vApp: pset , Set of application to be executed.

# Conclusion

- Propose a method to identify NFR, stakeholders and their preferences before writing requirements specification.
  - Preparing materials for specifying requirements.
- Support identification tasks, that are intrinsically subjective, by referring structural differences among use case diagrams.
- Apply this method to a domain: An email client.

# Future Works

- Develop Supporting Tool
  - Currently, only functions for recording and managing use case diagrams, their differences, stakeholders and their preferences are planned.
  - Even a tool with such functions can help analysts.
- Systematic Support for Identification
  - by using ontology and/or thesaurus,
  - by using more precise description in a use case,
  - by enriching NFR patterns.

Thank you.