

Requirements engineering
Assignment 6 - organisational analyse
Stichting Wireless Leiden

Rick van der Zwet
<hvdzwet@liacs.nl>

LIACS
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

January 5, 2008

Chapter 1

Introduction

The assignment given out by Luuk Groenewegen as part of the course Requirement Engineering to help learning applying the theory in a practical example. The assignment has been split into 6 parts, every chapter will handle one of them. The original questions could be found in appendix A. Instead of choosing a default industrial or governmental organisation, a non-profit organisation is chosen.

Chapter 2

Description

Stichting Wireless Leiden (hereafter called WL) has got a mission statement to build a wireless local network, technically comparable to the Internet, but standing alone and functioning independently. Making the connection easy and traffic free. This is done with the hands of sponsors and many volunteers.

Inside WL a number groups are working on ICT related services. Some highlighted ones:

Maintenance ('beheer') Keeps the network and supporting services keeps running.

R&D ('techniek') will build new software and hardware solutions.

Support ('gebruikers') is helping persons to get connected.

Construction ('nodebouw') will create new network points.

This paper will focus on the group maintenance as I am primary active inside this group and know most about it. The group mainly communicate by the means of an mailinglist, email send such list will received by all members of the list. All persons are unpaid volunteers with no agreements made with the foundations in terms of SLA or else. Every active member however did sign the 'volunteer agreement', which ensures people will no abuse the power they receive and will not mislead other volunteers or external persons in the name of WL. It is a very informal infrastructure, where apart from some technical setup nothing is documented. There is no hierarchy in terms of bosses and/or managers and people are free to contribute whenever they like.

Chapter 3

Activity diagram

Instead of making activity diagrams for the whole organisation, which are complicated and will not be clear in a short time-period. Only some parts of organisation will be highlighted. The activity diagrams will focus on maintenance and construction.

Figure 3.2 on page 5 will show the activity diagram of construction of a node¹. A site-survey will mean taking pictures of the possible location and do radio scanning to check for potential connections with other nodes and possible interference inside the neighborhood. Documenting will include putting the information in the version control system.

Figure 3.1 on page 4 will show the activity diagram of construction, the caption has quotes around 'group'. Those are intentional cause we cannot look at the group at one glance, but we have to look all persons individually. No person is assigned to a certain task or responsible for certain part of the system, which will cause all persons in the group to do the same activity diagram and not the group all together.

None of the activities are related of monitoring the processes or making sure problems are actually fixed or improved, which are some vital parts of embedded feedback loop pattern. Some parts however are covered. Environment input is set to be incoming problems from various sources, output will be the fixed problem. Production process is visualized inside the activity diagram. The Management information system will be the gathering of all problems, comments and their status. But standards, management, control, (intern) data, information and external data are currently lacking.

¹piece of hardware with radio antennas which provides a connection point for the people around the node and a number of antennas/hardware to provide connectivity between nodes

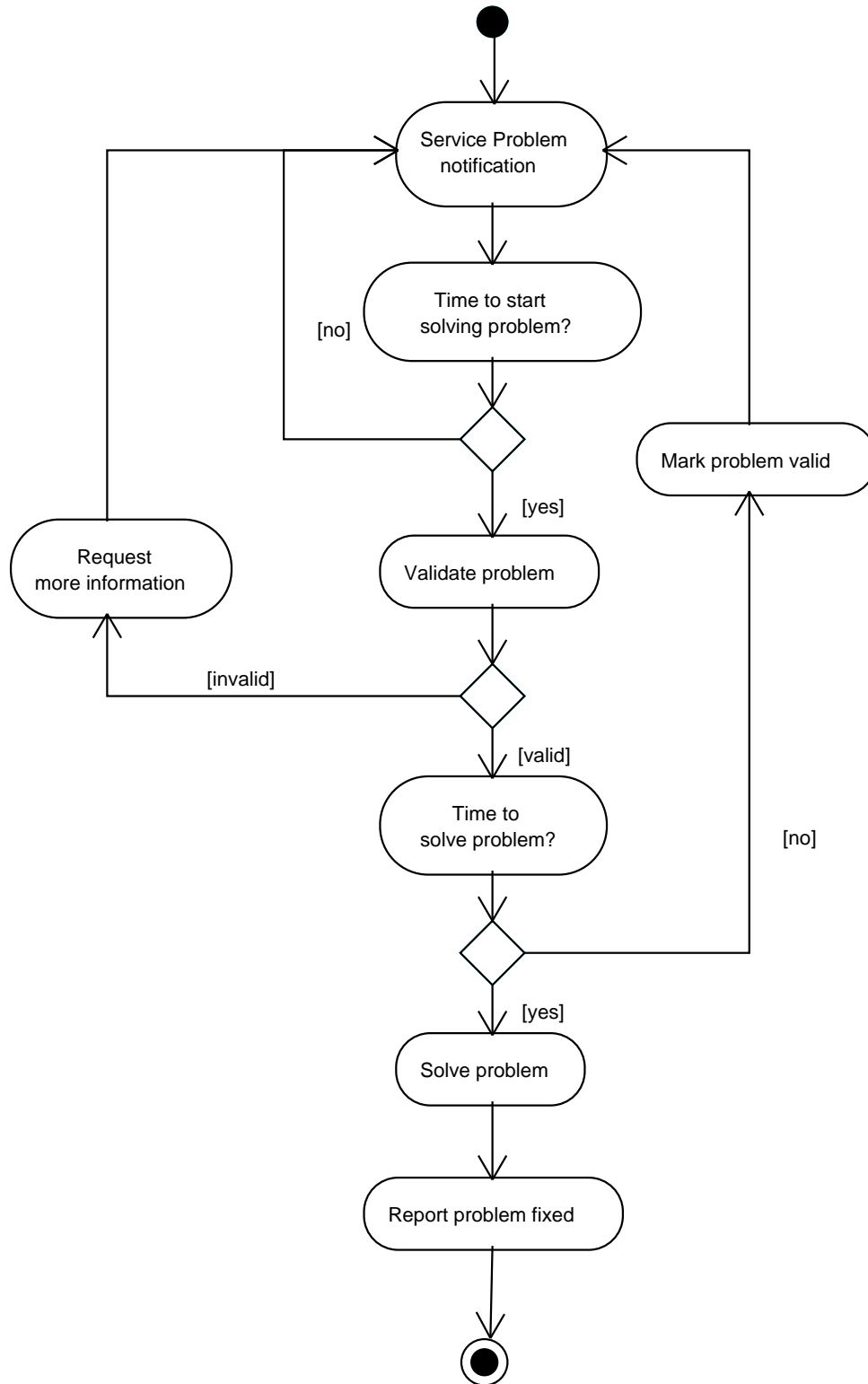


Figure 3.1: Activity diagram of maintenance 'group'

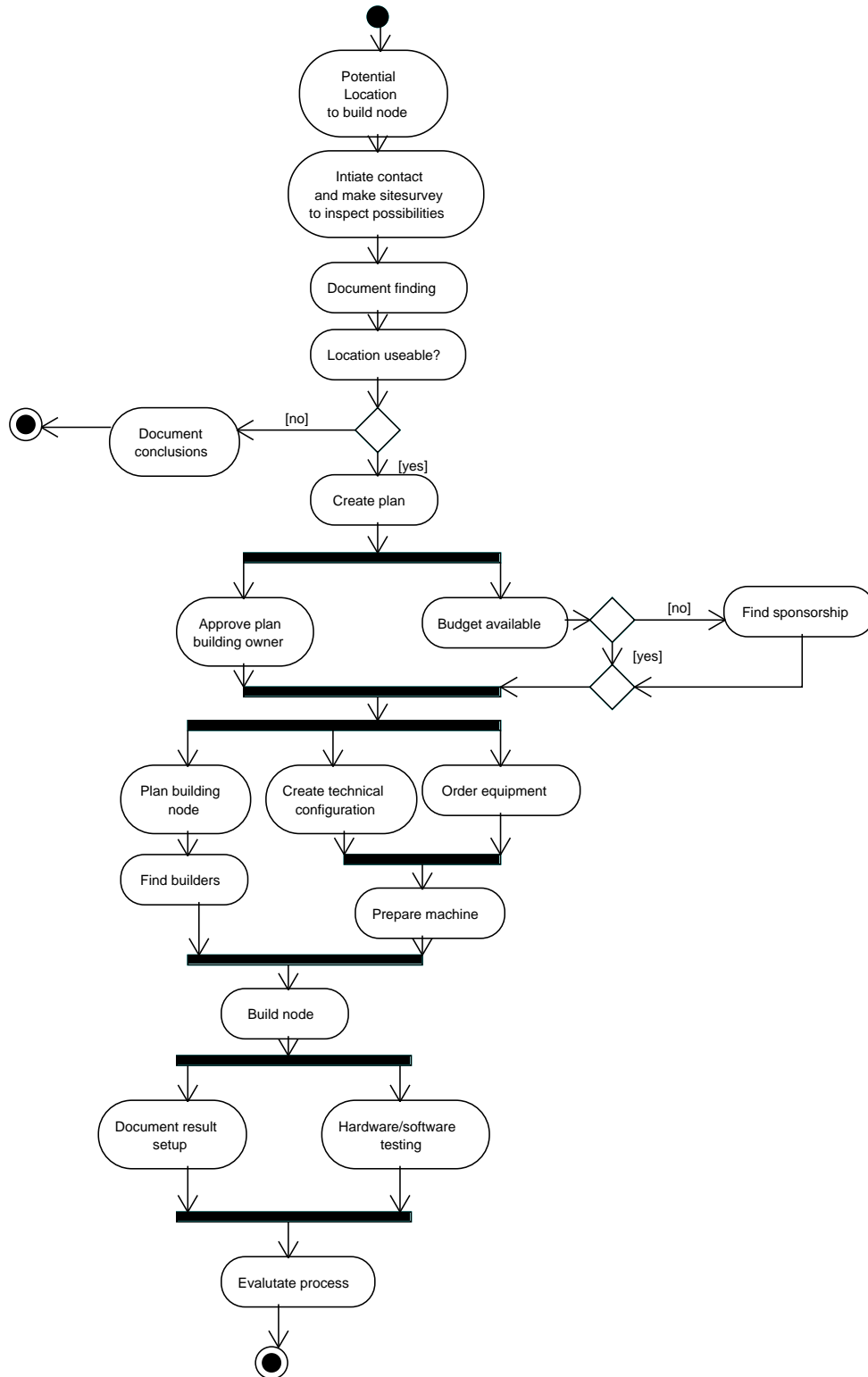


Figure 3.2: Activity diagram of construction group

Chapter 4

Class diagram

Figure 4.1 on page 7 has got a class diagram which reflect the execution of the activity diagram as found in figure 3.1. Depending whether someone has time is very subjective and has been put in a function `gotTime()`, but there is no solid definition for this case. Note that a problem is very subjective as well and there is not definition of a problem, it might as well could be an announcement for a birthday party.

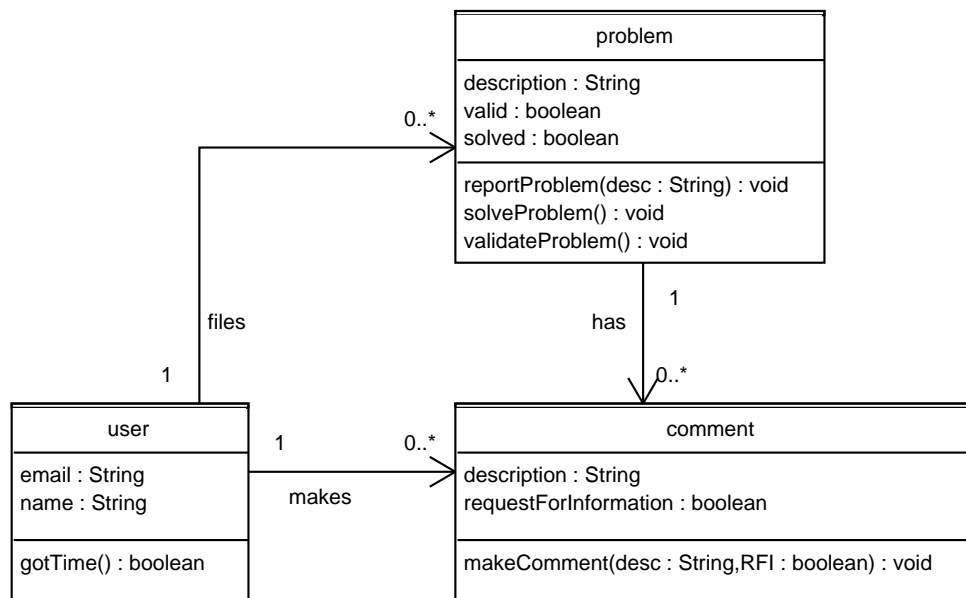


Figure 4.1: Class diagram of maintenance group

Chapter 5

Use Cases

Figure 5.1 on page 9 has got the use case of the open source mailinglist system called mailman ¹, this application is used in maintenance group to support communication and problem solving of activity and class diagram as described in chapter 3 and chapter 4. Functionality is relatively simple, both powerful at the same time. The system does not say anything about the possible usage of the system which makes it workable for multiple parts of the organisation, inside WL system is virtually used in all groups of WL to communicate and discuss various topics. The admin is in charge of the technical parts of the mailinglist system, most of the times he/she is responsible for all mailing-lists of the organisation. The moderator is set in place in order to prevent abuse of the system, by sending fake/unwanted messages like spam and to ensure to keep certain lists free of non-volunteers, like a list of all volunteers which did sign their 'volunteer agreement'. But also to make sure the admin could focus on his part and not to get distracted/flooded by the moderator requests.

¹<http://www.gnu.org/software/mailman/>

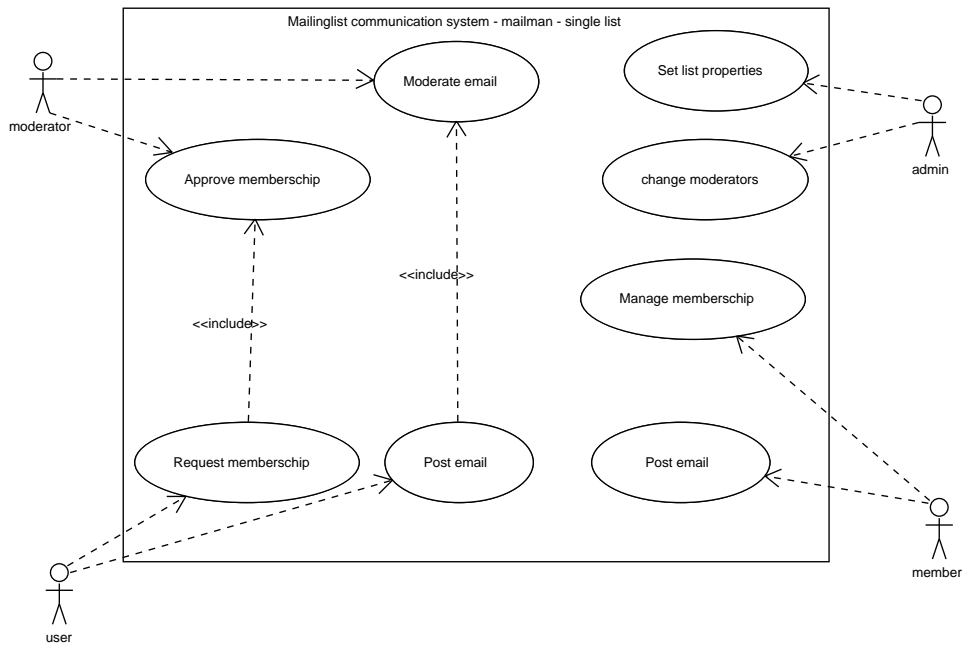


Figure 5.1: Uses Case of maintenance application, the mailinglist. In order to keep the picture readable the generalisations of the actors are left out. This is a hierarchical tree namely admin, moderator, member, user

Chapter 6

Change Proposal

The so powerful, easy to use, simple mailing system as described in chapter 5 has one major problem as it comes to supporting the maintenance group. This is no way to track a problem or to do something which related to reporting as there are no states defined. Which will cause the embedded feedback loop to be broken, the flow from the main process to the management and back is missing. Changing this will be almost impossible, cause it require a lot of rule setting -like predefined email formats- which will make it more difficult for persons to participate in the process. A better solution will be to start looking at a so called issue tracking system, which makes it possible to address a state to a problem.

An other problem relates to the moderators' task, email is frequently abused as spam making the moderator job cumbersome and time consuming. Every email of a non-member needs to verified and checked.

The last problem relates to the email communication in general;

- Everybody use their own email client which makes every message unique.
- There are social rules involved, where to type your text when replaying on a email message for example.
- The underlying email communication systems are fairly difficult and error prone, due the many technical factors -like DNS/network/programs/filters- involved.

Chapter 7

New Use Cases

Figure 7.1 at page 12 will propose the use cases involved into such issue tracking system. There exist almost an one-to-one mapping with the old use case at page 9. To get rid of email communication as primary source, a web-based application will be setup. To address the spam problem, users will first need to register before they are allowed to create a issue, which removes the 2 use cases 'Moderate email' and 'Post email' (linked to the actor user). 'Post email' has been split into 'Create issue' and alter 'Alter issue'. 'Set list properties will become 'Set project properties'

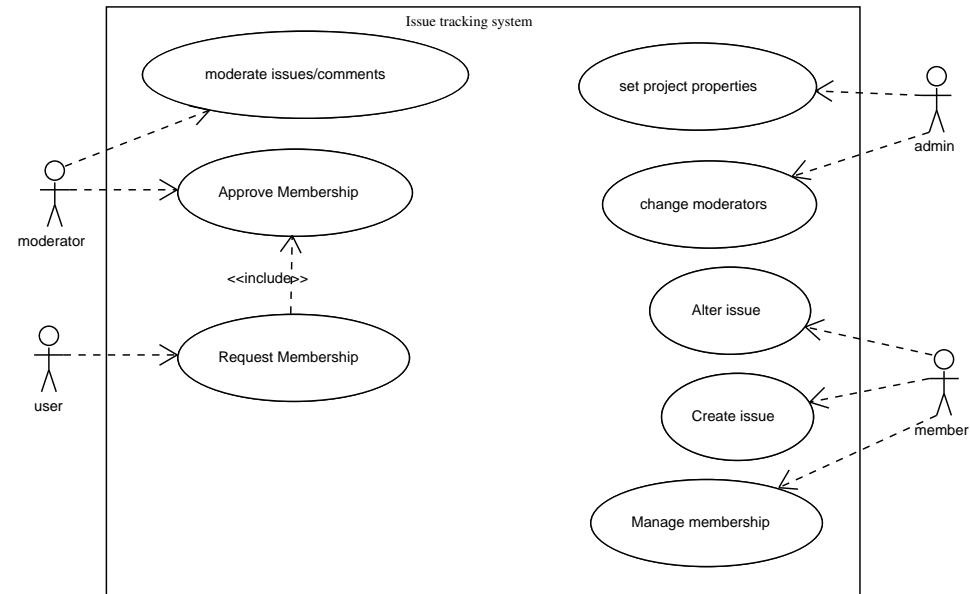


Figure 7.1: Uses Case of suggested maintenance application, issue tracker. In order to keep the picture readable the generalisations of the actors are left out. This is a hierarchical tree namely admin, moderator, member, user

Chapter 8

Check effectively

Effectively could be defined in many ways in a volunteer organisation, only checking the time needed to solve a problem might not be sufficient. More important is looking at the human touch. Volunteers needs to be more happy with the systems and willing to address more time to a project cause of the new system. The new system will need to assist them more than the old systems.

Next will be the user -the person who reported the problem-. Checking whether a user is more happy with the new system could be checked by simply asking him, but there could also be a more technical approach. Check the amount of problems coming in and whether a user asks multiple different questions. Asking new questions means that the user is willing to spend time to have a problems fixed and has trust the system and underlying structure will solve the problem.

Talking to volunteers and users will the only way to check whether the new system works more efficient, as the bits and bytes only will be far from sufficient.

Appendix A

Original questions

RE-assignment November 2005 my deadline proposal: Monday 2 January 2006

Select an existing organization - industrial or governmental - about which you can be informed quite easily and in which ICT-applications do play a role; e.g., the organization your mother is employed in or your father or yourself or someone else you are acquainted with.

a. Describe the organization verbally and not too long, including ICT's role(s) in it; use a half to one A4 for it.

b. Give an activity diagram of it (drawing!), which moreover is according to the embedded feedback loop pattern; apart from the figure, explain in not more than one A4.

c. Give a class diagram with their attributes and methods, such that these methods reflect the execution of the above activities or the manipulations by these activities and with relationships reflecting relevant connections between them. Explain in not more than one and a half A4.

d. Give one or more use case diagrams together with a description of the various use cases, covering the process support offered by the software / information system(s) as well as the main activities in and around the organization. Explain in not more than two A4 - and one might be enough.

e. Formulate a change proposal for the support as modelled in d. Keep the embedded feedback loop pattern (as clear) as it was; also, keep as many of the classes involved as you can. Explain in one A4.

f. Describe new use cases, where relevant in relation to already existing ones. explain the goal of the change proposed in not more than one A4.

g. How can one, after the change has been implemented, check the effectivity of the change in relation to the goal. Explain carefully, but in not more than one A4.