HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Juha Päivärinta

Customer Care Tool for WWW Services

Master's Thesis submitted for approval XX XXth, 2003 for the degree of Master of Science

Supervisor Prof. Martti Mäntylä.

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author and title of the thesis:

Juha Päivärinta: Customer Care Tool for WWW Services

Date: XX.XX.2003 Number of Pages: 71

Faculty: Dep. of Computer Science and Engineering

Professorship: Computer Networks (T-110)

Supervisor: Prof. Martti Mäntylä

Instructor: Prof. Martti Mäntylä

Providing good service quality is a necessity in today's competitive business environment. E-business and the WWW are fairly new concepts for service providers, which means that the operations model of E-business is still evolving. Hence, there exist a need for tools that help service providers to improve service quality in this new context.

The WWW creates new challenges for the customer care personnel. Problems may occur either in the service being used or in the network. Unfortunately, powerful tools that could help the customer care personnel to perform their tasks more efficiently are not widely used. There already exist some technologies solving the problems of the customer care personnel. However, they may be too complex to use, because not everyone working in the customer care unit has a technical background. In addition, the tools of this kind are not primarily developed so that the information they provide would allow the customer care function to operate fast enough.

In this thesis, the requirements for the customer care tool were formulated by interviewing some Finnish companies and organizations that provide services in the WWW. The architecture of the customer care tool based on these requirements is also presented in this report.

The customer care tool designed in this thesis has the following properties: The customer care personnel can receive information about the services used by the customers by monitoring the state of the WWW-application and the network, by testing the WWW-application, and by examining the information about past problem situations. All this can be done efficiently, because all the operations are easy to launch and the results can be understood without any technical background

Keywords: customer care, testing, monitoring, WWW-application, service quality

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä ja työn nimi:

Juha Päivärinta: Customer Care Tool for WWW Services

Päivämäärä: XX.XX.2003 Sivumäärä: 71

Osasto:
Tietotekniikan osasto

Professuuri:
Tietokoneverkot ja tietoturvallisuus (T-110)

Valvoja: Prof. Martti Mäntylä

Ohjaaja: Prof. Martti Mäntylä

Hyvä palvelu on olennainen osa liiketoimintaa. E-kauppa ja WWW ovat palveluntarjoajille suhteellisen uusia käsitteitä, minkä vuoksi E-kaupan toimintamallit ovat yhä kehittymässä. Tämän takia on olemassa myös tarve työkaluille, joiden avulla palveluntarjoajat voivat kehittää palvelun laatua uudessa ympäristössä.

Asiakaspalveluhenkilökunta kohtaa uusia haasteita WWW:stä johtuen. Ongelmia saattaa esiintyä sekä palvelussa, että verkossa. Valitettavasti toimivia työkaluja, jotka auttaisivat henkilökuntaa suorittamaan tehtävänsä tehokkaammin, ei ole laajalti käytössä. On olemassa tekniikkaa, joka auttaisi ratkaisemaan asiakaspalveluhenkilöstön ongelmat. Se on kuitenkin liian monimutkaisia käyttää asiakaspalvelussa, koska jokaisella asiakaspalveluhenkilöllä ei ole teknistä taustaa. Lisäksi työkaluja ei ole kehitetty ensisijaisesti tavalla, joka sopisi nopeaan toimintaan, mikä on asiakaspalvelulle tärkeää.

Tässä diplomityössä asiakaspalvelutyökalun vaatimukset on kerätty haastattelemalla suomalaisia yrityksiä, joilla on WWW-palveluita. Asiakaspalvelutyökalun vaatimuksiin perustuva arkkitehtuuri on myös esitetty.

Suunitellun asiakaspalvelutyökalun ominaisuudet ovat seuraavat: Asiakaspalveluhenkilöstö voi saa tietoa asiakkaiden käyttämästä palvelusta seuraamalla WWW-sovelluksen tilaa, testaamalla sitä ja tutkimalla tietoa vanhoista ongelmatilanteista. Tämä kaikki voidaan tehdä tehokkaasti, sillä toiminnot on helppo käynnistää ja niiden tulokset voidaan ymmärtää ilman teknistä taustaa.

Avainsanat: asiakaspalvelu, testaus, monitorointi, WWW-sovellus, palvelun laatu

TABLE OF CONTENTS

1 INTRODUCTION

People expect that they receive good service when they for example go shopping, eat in restaurants or travel by using public transport vehicles. Feeling that everything goes easy and that customers are treated well is such an essential part of any services we use, that it will not be noticed unless something is not happening as we expect. Customer care is a place that can be contacted, if problem situations cannot be solved on the spot or if the customer feels that she has something to complain about. It can be thought that customer care is the last attempt to make customer feel that her problems are taken into account and that she finally receives good service.

Computers and Internet can be utilized so that many of our everyday tasks could be performed more comfortably. Earlier these tasks could be carried out only by having business with employees of service providers or by using equipment located in the premises of the service provider. This changed situation may create new kinds of problems when customers use the services. Customer care personnel have to be ready to answer these questions that arise because of computers and Internet.

It is difficult to give a customer good advice when it is not completely known what she tries to do. This is the case in the situation when the customer cannot successfully complete operations she tries to by using a computer. The customer can tell what she tries to do, but also the computer and the network may cause problems and thus the result of actions is not the one which was expected, although nothing that customer does causes trouble.

In this kind of a situation customer care personnel have a hard time so that they can answer properly the questions that are presented. Well designed tools may help them to complete their jobs. However, the problem is that tools of this kind are not yet used in customer care. This is partly because services have not been available long via Internet and the business culture is still developing.

This thesis presents functional requirements and an architecture of a customer care tool. The idea was to design a tool that can be used efficiently and that does not require much from its users. However, this kind of functionality must not affect to the quality of information that can be received by using it. The goal still is that customer care personnel can make customers satisfied and thus fulfil the idea of delivering

good service.

The structure of thesis is the following: The chapter “Problem Statement” defines terms and concepts which are used in this thesis. In addition the domain of a customer care tool, reasons why a customer care tool needs to be designed, and the scope and objective of thesis are presented. Criteria, which are used for evaluating the customer care tool that is designed, are presented in the chapter “Criteria”. The chapter “Generic Model” describes the position of a customer care tool when it is compared to branches that have something common with it. A general solution for a customer tool is also introduced as a form of requirement specification. The chapter “Previous Work” shows what useful relating to a customer care tool has been done earlier. The chapter “Own Solution” tells about the process of gathering requirements and presents an architecture of a customer care tool that fits in the general model. The chapter “Analysis” evaluates the architecture against the criteria so that it can be seen how well the designed architecture actually satisfies needs that customer care personnel have. This chapter tells also about simplifications and problems that were not solved. In addition it describes further work on this topic. The chapter “Conclusions” summarizes what was done in this thesis.

2 PROBLEM STATEMENT

2.1 Definition of terms

2.1.1 Concepts of Internet

If any two parties are going to interact, they have to have a common way to communicate. A protocol in the Internet context refers to a set of rules used to control data transmission between the communicating parties [15].

Transfer Control Protocol/Internet Protocol, or TCP/IP is a set of protocols used for data transmission between the machines connected in a network. [24]

The Internet is a network that consists of other networks. It links computers to other computers by using the TCP/IP-protocol [5].

Hypermedia is a combination of several different media, for example text, sound, pictures and video [42].

The World Wide Web or WWW is an information service that is used for browsing information. It offers a hypermedia system and stores information as text, graphics and audio, for example. [9]

Uniform Resource Identifier or Uniform Resource Locator (URI or URL) identifies a source. For example information can be located in the WWW by using URL.[10]

A distributed system is a collection of independent computers that co-operate and appear to users as a single computer [23]. For example, applications that run in the Internet (or the WWW) have a distributed nature.

2.1.2 Services and content

It is possible to carry out everyday tasks like shopping in the Internet. Also information like timetables or news can be found in the WWW. These kinds of Internet applications have their own vocabulary for defining terms, depending on the

nature of media, where these applications are used.

Service (in this context) is an application that is used via WWW. Service is something that a service provider offers to a customer. A part of this research concentrates on observing the state of WWW-services. In this study, a service can be defined as a commodity that can be reached by using the WWW-application of a company or an organization.

Content can be seen as the value that service offers to customer. It may not only be a simple action but also, for example, information. Content is not just a technical function of the application, although functionality can affect to the quality of the content. It should be noted that the content is a part of a service and not a totally distinct concept.

Service quality is some metric that is used for determining the goodness of a service. Customers may use different metrics than service providers. It should also be noted that service quality may not be objective, when WWW-services are concerned, because also the psychological issues may affect the results. Service quality should not be mixed up with the concept of 'Quality of Service' or 'QoS', referring to consistent and predictable data delivery in networks and that can be measured by using the metrics, such as packet loss, delay and bandwidth [30].

Critical services are services, which can have a major impact on the life of a customer. Services that handle money, for example, are critical, because faults in the system providing these services can cause severe consequences to the customer. These can be financial losses, for example.

A Web site is a collection of Web files on a particular subject that includes a beginning file called a home page [43]. Customers can use these services via web sites. For example, they can use a web site of a bank or some commercial company.

A service provider is an organization that offers services to customers. This study focuses on the services that are used via the Internet and the WWW, which sets limitations also for the concept 'service provider'.

E-business denotes the transformation of key business processes by using the Internet technology [3].

2.1.3 Customer service

Customers have to have a possibility to contact the service provider. Usually companies have some kind of customer service function that takes care of their customers.

The customer care function is typically a distinct part of an organization. Its task is to take care of the complaints and requests for help that come from the customers. The customer care function also handles the feedback that is received from the customers. This feedback usually considers the quality of service from the customer's point of view. The customer care function covers all the business domains of organization. This study treats the customer care function as if it was only a support service for the WWW-applications.

A help desk is a part of an organization that gives customers help and guidance in problem situations. The help desk concentrates on giving help to customers by using the WWW- or mobile services and receiving feedback from the same group of users. This makes the help desk different from the traditional customer care function that takes care of the other business interfaces, too. However, in this study, the terms "customer care function" and "help desk" are used interchangeably.

2.1.4 Management and development of software

Software engineering refers to the management of software and software development to minimize the costs. It contains all the phases from gathering customer requirements to the testing and maintenance of system [6]. This study applies the concepts of software engineering to the design of the customer care tool.

Requirement specifications are collections of requirements that satisfy the needs of the customer. Requirements are not presented technically. This means that requirement specifications are not descriptions of an application. Applications can be designed based on the information that the requirement specification contains. A single application is just one solution that satisfies these requirements.

An architecture design is a part of software development process. It divides the application to be developed into independent components [14]. Application architectures can also be constructed by using the physical location of software

components as a division criterion.

A module design divides architectural components to smaller parts that have a certain functionality. Also this functionality is designed. [14]

Testing is an action that tries to find existing faults in programs. Testing is considered to be an action that starts with the conditions that are known beforehand. Functions and procedures used are also known and they have predictable results. The goal is to define whether these results meet the requirements or not. Testing should be designed beforehand. Its purpose is to be rigid, constrained and systematic. [4]

A bug is an error in an application that causes incorrect function or results.

A fault can be seen as an equivalent to a bug. It is an error in application that causes unwanted results or actions.

Debugging is an action identifying errors in the functionality of a system. The identification of errors is followed by the operations fixing the bugs. Debugging is by no means a systematic operation, but it is performed by using methods that may reveal errors.

When a system is observed as a 'black box', knowledge on the internal structure of a system is not used. The functionality of a system is observed by using interfaces and functional requirements.

When a system is observed as a 'white/glass box', information about its inner structure is utilized. Program code and used algorithms, for example, may be utilized in the white/glass box approach.

Monitoring of a service is a process of evaluating the state of the application offering services to customers. This means collecting and analyzing information on several different parameters. For instance, these parameters may include the current number of users and the number of transactions under evaluation.

Network/application management concentrates on controlling the resources and taking care of the faults, configurations and security of an application. Its tasks include also collecting information about the state and performance of the system, as well as information for accounting. Its goal is to improve services, reduce downtimes, control costs and response to the requirements of customers. [34]

2.2 Problem domain

2.2.1 Case: a busy day in customer care

It is a busy day in the customer care function of a WWW-service of a public library. The library offers its customers a possibility to use the Internet to check the availability of various books, cd-records and multimedia so that customers can reserve and borrow them. In addition, it is possible for customers to renew existing loans by using this service, which makes their lives a bit easier. They do not have to visit the library every time their loan expires and they have not finished their book yet.

Today this service has a lot of users. However, they have difficulties to get connected to the WWW-application. In addition, it takes a lot of time to find out what material can be found in the library and to complete the reservation of the material. The customer care center receives many phone calls from customers wanting to know what is wrong with the service. Without any tools indicating the state of the application and the network it would be difficult to solve the problems experienced by the customers using the WWW service.

Luckily, the personnel have an application that helps them to identify problems in the service. They can easily find out that the system has had a lot of users all day long. However, they cannot be sure, if this huge load can cause false functionality. The same application can be used to determine how the system really works. For instance, the customer care personnel can run a series of tests showing that the WWW-application works correctly, unfortunately it is only slower than usual.

This slowdown of the service was not known before, because the service has not been available very long. It is important that this kind of problem can be recognized, if it occurs again. In addition, it is important to report this kind of situations so that the maintenance personnel can find the bottlenecks of the system and get rid of them. This is why the personnel use their support tool to automatically create a report about the technical details of the current situation in the WWW-application. They also add the customer complaints to the database of the customer care tool, as soon as the situation in the customer care center cools down.

2.2.2 Environment

E-business is quite a new business area. It was only in 1995, when online marketing began in simple sites [35]. Nowadays it seems to be a trend that companies have Internet sites that sell or market products.

In addition, there exist also diversified services that do not provide any physical commodities but only content or the ability to evaluate tasks utilizing the Internet.

Services can be divided to critical and non-critical services. Critical services perform operations that do not exert a concrete impact on customers' lives.

Customers using services and willing to pay for them are naturally expecting good quality content and properly working services. However, performing tasks in the Internet causes problems that do not exist in more traditional environments. Some problems occur because of the distributed nature of the Internet. For instance, there may occur breaks in Internet connections at the same time when some tasks are performed. Other problems include the malfunctions of hardware or software in the customer's or service provider's applications. In such situations, the customers cannot be sure whether the actions performed by them were really completed. This is always annoying, but it can also cause more severe consequences. Let's consider a situation, where a customer is paying goods from an Internet store and is paying his or her purchases, when a fault situation occurs. Under such circumstances, the customer cannot be sure, if he or she really paid the bill and if the service provider knows the contents of the customer's virtual shopping cart so that the order can be properly delivered. In the worst case, the customer performs the paying, but never receives the goods, because the service provider lost the information about the purchases due to malfunction.

Properly developed applications and networks should eliminate problems of this kind. However, the customer may feel insecure about what really happened and may want to make sure that nothing harmful happened. On the other hand, if the customer could not use the service, he or she may want to report the fault and find out, when the service works properly again.

Support services for the service provider's applications are needed so that these goals can be reached. This means that there exists a need for efficiently operating customer care units and help desks. These units have to have relevant information about the

state of the services and possible fault situations. This information is delivered to the customers in problem situations. In addition, customers need to be advised relative to how they should act, if there is a need for customers to perform some tasks because of the fault. However, the customer care function cannot rely on the other departments of the organization to provide them with information on problem situations fast enough. The information flow inside the organization of the service provider can be slow or it may not reach the customer care function at all. In the worst case, the reasons for or the solutions to the problem may be delivered too late to satisfy needs of the customer. Hence, there has to be some tools in the customer care center, taking care of collecting information about the system. This reduces the delay of getting information to minimum.

Service quality is an important factor. As Internet technologies improve and people get used to the Internet, this factor will be far more important than it is today. Customer care centers and help desks can improve service quality. This can be accomplished easier, if the personnel of the customer care function or help desk has tools that can monitor the state of the service and produce enough information on the problems of the customer in a timely manner.

In addition, the profits of E-business are not big. The operations that have to be performed for solving problem situations can easily consume them. This is why efficient functioning of customer care is of outmost importance.

The library use case that was discussed in previous section is not an everyday situation in a customer care function. In real life, it would not be possible to use the tools exactly in the same way as described in this example.

2.2.3 Customers

The typical client of the E-business support services cannot use the E-business service or he or she reports about the fault of the system. The Internet, WWW-applications and mobile services are still relatively new inventions, which means that the persons, who use computers a lot, use also the Internet and related services. This is why the customer of today knows a bit more about the underlying technology than the customer of tomorrow, although this situation is already changing. When the services become more popular, a wider crowd will start using them. This means that

the average person using the service will not differ much from the ordinary people who do not know much about the computer and network technologies. Therefore, the customers of tomorrow may experience the problems related to using Internet services even more confusing than the customers of today. This may cause more stress to customer care centers and help desks. First, the number of contacts to customer care centers will increase. Second, the customers with a limited understanding of the technologies used for building networks and applications will feel more insecure and thus have more questions to the personnel. This increases the amount of time spent per customer.

It should be noted that customers want non-complicated services on the user interface level. However, the tasks should be easy to perform. In addition, this means that there should be no need for customers to study the Internet or software technologies, so that they could use these services. If problems occur, there should be some instance that could rapidly deliver information and advice that is needed. People who can do this work in the customer care function. They may complete their tasks more efficiently, if they have proper tools in hand, when they communicate with customers. The system that was utilized in the library-use case comes handy in many situations.

2.2.4 Customer care personnel

Customer care units in companies have two types of employees. There are persons that have education for serving and helping customers in problem situations. The other type has a technological background and can help customers in the application specific tasks. The former type has skills for managing customer relations, but this kind of customer care personnel does not have a deep technological understanding. The latter type knows about the technology, but persons belonging in this group may have difficulties in formulating their knowledge so that even a layman could understand them.

The customer care function and help desks are usually located close to the persons who take care of application maintenance. However, the customer care function may be separated from maintenance, when it is not possible to get information fast or find rapidly persons that could locate and fix the problem.

It is also seen that the customer care personnel does not need a strong technological background, but it should be trained by giving brief courses about the application and other relevant technologies.

When a customer care person needs to give detailed information about the problem at hand, he or she needs to be able to access to the right kind of data or he or she needs to have some kind of technical education for this purpose. The inability of the customer care center to answer the questions of a customer may be perceived as bad service quality. On the other hand, if a customer care person contacts directly the technical unit, it also increases the time that he or she spends with one customer. In addition, the customer also has to wait. This cannot be seen as good service quality either, although the customer will finally get information he or she wanted.

Customer care tools need to be efficient enough so that the customer care function does not get overloaded and that the customer can get fast what he or she needs. This also means that the requirements for customer care people do not have to change substantially.

2.2.5 Objective of study

Because of the reasons stated above, there is a need for a software application for the customer care function of the WWW services. The specification for this study was the following short description:

"Consider an example where a customer calls the operator at the customer care center reporting that the service she is trying to use is not working. Right there the person in the customer care center will be able to tell the customer if the service is really down and when it is going to get fixed again with the help of an easy-to-use customer care tool."

Observing the state of an application and even performing some testing in the customer care function is a new concept. This study was carried out to add to our knowledge on the business domain and the needs of customer care function. The original idea was to identify critical properties of the customer care tool. Another goal was to find out whether this kind of application is needed at all. In addition, the architecture of this kind of a system was seen as an important part of the study.

The objective of this study was to identify important and essential properties that a customer care tool of the future should have. This was done by interviewing some companies that have services in the WWW and by analyzing the results of these interviews. A requirement specification and an architecture specification of a customer care tool will be presented in this thesis.

2.2.6 Scope of the thesis

The thesis is limited to considering the properties of a supporting application used by a customer care center or a help desk for Internet services. These properties can be used for formulating the requirement specification and the architecture specification. This study does not present a module level specification of the system nor does it implement the system that is designed.

Very complex systems, such as the account systems of banks are not discussed in this study. Components of these systems monitor state of other components and these systems must have extremely high fault tolerance. Customer care-based monitoring and testing was not seen as a necessity in these cases. In addition, this kind of a system is difficult to design and implement. The solution is suitable for one system and cannot be used in some other context without big modifications.

3 CRITERIA

Information about the important properties of the customer care tool of the future was gathered by interviewing various parties with relatively large-scale Internet services. These interviews also resulted in information about the operational environment of the application and the markets of the customer care tool. The criteria and requirements for the customer care tool were formed based on the analysis of the interview data.

All the parties that can make use of the customer care application or have to maintain it naturally have some requirements relative to how it should work. The two most important groups that use the application (directly and indirectly) are the customer care personnel and customers.

The third party that sets the requirements for the system is the maintenance personnel. The WWW-services are modified relatively often, which means that there may be a need to update the configurations of the support tool every now and then.

The criteria, which can be later on used for analyzing the architecture and properties of the customer care tool are presented below. These criteria are divided into two groups. The first group includes interviews, whereas the second one involves the customer care tool.

3.1 Criterion for interviews

Interviews are a very important part of this work. They define the needs of the customer. If these interviews are carried out so that the most important aspects of the customer needs are not understood, this study cannot be used to forming a system that works in a way that the customer requirements would be satisfied [14]. This criterion does not emerge from the interviews. Instead, it aims at determining how well the interviews were carried through.

Criterion 1: Success of interviews

This criterion considers how the interview situations can be arranged and carried through, how interviewees may understand the questions in interview situations, and

how the requirements of the customer care tool can be derived from these results.

3.2 Criteria for customer care tool

The performance of the customer care tool can be analyzed in several levels. Different points of view are, for example, how the customer care tool should work with its users and how the customer care tool should work with the WWW-application that it tests and monitors.

Criterion 2: Universal applicability

The customer care tool must not work with only one particular application, for example, only with the library service mentioned in the chapter "Problem statement". It has to be possible to use the customer care tool with all kinds of WWW-applications.

Criterion 3: Capability to examine the state of the WWW-application

The customer care tool has to be able to tell how the WWW-application is operating and track down any problems in its functionality. This criterion makes it possible to tell the customers whether there is something wrong with the application.

Criterion 4: Transparency of the customer care tool

The customer care tool should not exert an impact on the system that it tests and monitors. This means that the performance of the WWW-application should not become weaker if the customer care tool is plugged into the system.

Criterion 5: Low maintenance costs

There is no way of producing a product that could be used by everybody without any modifications. However, the work of the maintenance personnel should be made as easy as possible, because applications may undergo changes on a fairly regular basis.

Criterion 6: Capability to utilize several data sources in the same application

The interviews revealed that there are distinct tools in use for delivering information about the different aspects of the system to be monitored. However, the data produced by one application cannot always be utilized by another application, which

can be seen as a defect of the current tools.

Criterion 7: Possibility to use the customer care tool without technical background

The users of the customer care tool do not always have a technical background. This kind of people must also be able to use this tool and understand the information that it delivers. However, it is acceptable that a short training period can take place before people start utilizing the customer care tool in their work.

Criterion 8: Fast performance

Operations that are performed by the customer care tool must be carried out while still in contact with the customer. This means that operations that the tool performs have to be rapid. In addition, the results must be delivered in an easily understandable format.

4 GENERIC MODEL

4.1 Customer care tool and existing application management software

The customer care tool needs to be able to compile information about the state of the WWW-application. There already exist methods for data collection. These operations are monitoring the parameters of the network and the WWW-application and testing if the application produces the right results.

Testing and monitoring of software and applications are not new concepts. However, the context of this study implies some features and constraints distinguishing these actions from their traditional counterparts. An interesting aspect is that the customer care personnel do not always have enough knowledge to make the decisions themselves. They have to trust that the information and the explanations given by the customer care tool are correct.

4.1.1 Software testing

Testing is an essential part of a software developing process. It is performed by people who have a technical background and who know about the structure and the implementation of the application to be tested. Testing in software development does not take place in any particular phase of the process. Rather, it is performed throughout the development process, since software does not always function quite as was designed in the earlier phases. Because people are incapable of programming perfect code or correctly functioning applications that meet all the requirements, the costs of debugging and testing account for a very large part of the developing costs. It is calculated that testing and debugging costs range from 50% to 80% of the total costs of developing an application [4]. Tests can be done by using different approaches and by emphasizing different aspects in different phases of the software testing process.

At the moment, online testing according definition is not performed with the finished software product in the most of the companies that were interviewed. Only

companies that have critical services use testers. The customer care unit does not perform any testing operations. Sometimes fault situations are tried to repeat by reproducing operations that customer has made before.

The faults are identified and fixed by technical personnel. The primary task of the customer care function is to offer information in situations where customers do not know how the system works or when the system is malfunctioning.

If the customer care unit is going to carry out the testing, the test types must be chosen carefully. For example, stress tests, in which high background loads are used for bringing the system to its limits, cannot be used on systems serving customers at the same time. The black box type system level testing can be used for checking that the components work properly. If the system to be tested can be modified, glass box testing can also be used. This reveals more exact information on the location of the fault. In any case, the customer care personnel are passive testers. They rely on the information and interpretation that the customer care tool provides. In addition, they do not try to fix the faults.

4.1.2 Application and network management

Application management concentrates on maintaining the software. Network management is used for monitoring and configuring both the software and the hardware of some particular network so that it can operate efficiently. These tasks are evaluated by experts. The customer care function takes a passive approach to network management. In other words, it is the customer care tool that monitors the state of the network and tracks down the problems. The personnel are not allowed to configure the network or the applications.

4.2 Requirement specification

The ultimate purpose of the customer care supporting tool is to help the personnel to complete their tasks effectively and thus provide the customer good service quality. In other words, the customer care personnel should be able to do their jobs more easily, faster and better than without the supporting tool. Customers, in turn, should receive relevant information about the problem that bothers them.

To satisfy the requirements mentioned above, an application needs to manage the tasks presented below and shown in Figure1:

- It has to monitor service performance.
- It has to be able to test network connections and test the functionality of the WWW-service.
- It has to store data about the results in databases and fetch information from database
- It has to be able to create reports about the problems and the data stored in the databases.

Figure 1 Functionality areas of customer care tool

4.2.1 Monitoring the performance of services

Monitoring the performance of a service can be divided in two groups, including monitoring in the technical level and monitoring in the content level. Content level monitoring is very hard to automate. This is why this kind of monitoring is not included in the software. For example, the software cannot tell, if there are any spelling mistakes in the names of the books in the database records of a public library. Monitoring on the technical level involves checking the state of the system so that it is possible to say whether it is up and running or if it has crashed. In addition,

if the system can be modified, it is possible to check the parameters about the transactions of the service. Naturally, a WWW application may already collect data about its state and this data can then be utilized in the customer care tool without the modifying the application.

The service monitoring section of a customer care tool has the following requirements:

- It is possible to check if a service is reachable and that it is up and running.
- It is possible to collect data about the network and the service and show it.

4.2.2 Testing the services

Testing helps find out whether the WWW-application produces the right results to the requests or not. However, the customer care personnel do not have enough time for designing the tests themselves. In addition, they may not have enough technical knowledge to be able to do that. The same problem arises, if the customer care personnel need to analyze the results of the tests. Hence, the tests that are used by the customer care unit need to be designed beforehand. In addition, the customer care tool should contain the right results that can be compared to the test results. As a result, the customer care personnel do not have to perform any analysis by themselves.

The black box tests fit well to the tasks of the customer care unit. The personnel working for the customer care function can run the tests indicating whether the devices in the system are up or have they gone down. If the devices are up, it is possible to test whether the services are working properly. One approach is to use combinations of individual tests that focus on distinct parts of the system. A skilful grouping of the tests makes it easy to monitor the functioning of a system under operation.

Tests can be grouped based on the physical architecture of the WWW-application. The glass box testing can also be used. If the division of the system to modules and components is known, it is also possible to check the parts of the system that work properly. However, this may require modifications in the system that is currently being tested.

One efficient feature would be a capability to track the spheres of influence. This feature comes up naturally, if the glass box testing is used. If testing reveals faults in the system, it is important to know, what services will be malfunctioning because of this fault. This will reduce the amount of unnecessary testing.

If several people in the customer care unit run testing operations in problem situations, this will cause unnecessary pressure on the system. However, this load can be reduced, if the system can be accessed by using different user rights. The tool may also reduce the number of tests by using different rules based on the state of the WWW-application to be tested.

The testing section of the customer care tool has the following requirements:

- It is possible to create tests and result analysis beforehand.
- It is possible to black/glass –box test the WWW-application
- It is possible to run existing tests, when needed.
- It is possible to analyze the results of the tests automatically.

4.2.3 Database management

The testing and monitoring function of a system can also save information on the performance of the WWW-application. The customer care tool should have a database that contains information on the following topics:

- Information about the previous faults of the system
- Information about the previous problems that customers have had
- Information about the state of the system

Information on the previous faults and the problems of the customers should be available in text format. The personnel should have a possibility to enter information into a database. Naturally, the system itself can also do it in cases where the testing reveals some problems. It is important that the text material can be investigated on different levels. The customer care function may want to know about the faults and their antecedents, but the technical personnel usually needs more detailed information

on these topics.

Although information needs to come in an easily understandable format, it is also important to store the unprocessed information that is collected from the system. Naturally, text cannot always indicate the reasons for the faults. Also the maintenance personnel will be able to use the collected data. However, clear descriptions of the faults are essential for the efficient performance of the customer care function.

Data types that will be stored in the database were described earlier. In addition, the database management of the customer care tool has the following requirements for its functionality:

- It is possible to add fault information about the WWW-application to the database and fetch it from there. This information covers the previous faults and the previous problems of the customers.
- The customer care tool adds automatically the data produced by the monitoring system to the database and it is possible to fetch this information from there.

4.2.4 Report management

Reports can be divided into two groups. The first one contains information about the faults and the reasons of the faults. These reports are saved in a database by the personnel. It is seen that the previous fault situations may occur again and the old information can be utilized in these cases. At the moment, customer care units do not perform reporting, because everybody involved in running supporting services knows the previous faults fairly well. However, for new employees databases are a good information source. In addition, if the customer care personnel do not have technical knowledge, the databases may be the only way of identifying the causes of malfunctions.

The second group of reports contains the data that the system collects itself. This data is a result of the monitoring of the network and the WWW-application. Also the graphs and the reports created by using this information can reveal reasons to the faults.

The report management section of the customer care tool has the following

requirements:

- It is possible to create reports focusing on the fault information that is collected in a database.
- It is possible to create reports about the data that is collected from the system and stored in a database.
- The system has to automatically create a report in cases where the problem is revealed by testing

The customer care tool does not totally cover all the areas from which the requirements are gathered. This is because much more can be done in every area, but these actions are not relevant for the customer care tool. It can be seen that the coverage is, by utilizing Figure 1, the one shown in Figure 2:

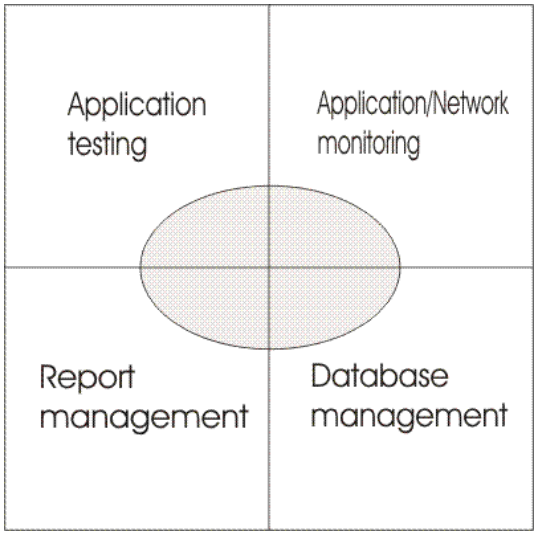

Figure 2 Customer care tool's coverage of its functionality area

5 PREVIOUS WORK

The interviews revealed that the use of the customer care and help desk tools for testing or monitoring services is not very common in the companies that offer services in the WWW. However, this does not mean that there exist no tools that could be utilized or that nothing is done so that the services could be maintained efficiently.

There also exist commercial applications that can be used for testing and maintenance. Internet protocols also offer tools that usually form a part of the maintenance and monitoring system.

5.1.1 Equipment and methods used in the case companies

All the companies that were interviewed for this thesis use some kind of monitoring system. The functionality and significance of these applications is dependent on the importance of system that is monitored and tested. A testing and monitoring system is usually not a single application, but it consists of different tools that have been developed rapidly when they were needed.

At the moment, there are plans to improve and replace the existing service management systems. The reason is, as was mentioned above, that most of the current applications are constructed on ad hoc basis and it is seen that there will be a need for more efficient systems in the future.

Different kinds of screen capture applications are utilized. Systems write log files about the user transactions. By investigating these logs, it is possible to find out what has gone wrong while these transactions occurred. Some applications use simple graphs that represent some factors of the system, such as the number of users. Changes in these graphs indicate that faults have occurred.

An important task is to define the transaction flows performed by the customer when he or she uses the application. This may also reveal the reasons of the faults. A more complex task falling into the same category involves defining the spheres of influence that a single action has.

The intranet maintenance units, however, have commercial tools that belong to one product family. This is because the devices constituting the intranet as well as related software are usually similar. These products perform their tasks well in the current conditions. However, these systems are not open and there is no easy way of updating them to work efficiently on some other tasks or with different machines.

Some products are also designed for the management purposes. They have databases for the reports and for personnel information. However, these applications unfortunately have problems with database consistency and they may be somewhat too heavy to be used efficiently.

5.1.2 Defects of current systems

In general, there does not exist any single application that tests or monitors services. Applications are created on ad hoc basis. It is seen that one centralized application could serve the personnel better than the current group of applications. This is because different tasks can utilize the results of the other tasks. Also, the databases would remain consistent, because all the tasks use similar tables when only single application is used in the data storing operations.

Most service providers do not run the testing function, according to the definition, when the system is online. They simulate the actions of the customer to identify the faults in the system. If testing is planned in advance, it can be automated to some degree. This makes the operation of the customer care function faster, because the only thing that the personnel have to do is to choose the tests to be performed. Automated tests will also help the customer care unit to perform their tasks better, because instead of creating the tests themselves, but they can utilize the existing ones.

Lack of information may also be a defect in a system. There exist systems that monitor the running application, but information may be difficult to get in a decent format or it is not available for every one that might need it.

It is difficult to find faults, if they are located outside of the service provider's network or services. It was seen that any information that could be gathered from the connections or configurations of the customer's equipment would help finding the antecedents of the problems.

Traceroute uses also the ICMP echo request and echo reply. Traceroute works a bit like a Ping-application. It sends echo requests to the destination machine trying to solve a route between the sender and the destination. This is done by utilizing several echo requests and the TTL (time to live) field of Internet datagram. The Traceroute application sets the TTL-field of Internet datagram to 1 and sends this request to the first router. The router decrements the TTL by one, setting the value of the datagram field to zero. Zero means that the datagram expires. However, the router processes an ICMP time exceeded message and sends it to the host running Traceroute. This reveals the first node on the route to the destination. Traceroute repeats this procedure a couple of times before it increments the TTL by one. When several echo requests are sent using this new TTL-value, the second node of the route can be solved. By continuing this procedure of incrementing the TTL and sending echo requests, the whole route can finally be solved [22].

It should be noted that firewalls may sometimes block the echo-request messages, since they can be misused in an attempt to create a denial of a service situation. In this context, this means that the host receiving messages uses its resources by replying to them and cannot thus provide other services. In addition, large echo request packets with big optional data fields may even cause the death of a kernel [17].

Ping and Traceroute can also be used by the customer care testing tool. If the tool and the WWW-application, which cannot be reached or used properly, are located in the same part of the network, it is possible to test whether there are problems in the route between the customer and the WWW-application. Ping can also be used for determining if the application machine is up and running. The task of Ping and Traceroute is part of the network and application management.

5.1.4 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is a collection of specifications that defines the protocol, database and associated concepts [34].

A network management system consists of managers and the machines that need to be managed. The managed entities run software that can report the occurrence of problems. In addition, they store information about their state in a local database. The

managers can configure the settings of the managed machines and monitor the stored information. [7]

The structure of SNMP can be divided into the operations and the ways of storing the management information of the managed machines.

In order for an SNMP protocol to work, there has to be a database containing data about the relevant issues in network management. Each node working under the network management does have such a database. It is called a management information base (MIB). MIB is a structured set of variables that can be fetched and stored. [9]

The namespace containing the names of MIB-variables is administrated by International Organization for Standardization (ISO) and International Telecommunication Union (ITU). Each object stored in the MIB is associated with an object identifier. The defined objects have identifiers that form a treelike structure. The location of an identifier in a tree can be presented by integers and dots. Dots are nodes in the namespace tree and integers define the leaf of a previous dot. Every integer also has a name. For example, an MIB-group can be defined as 1.3.6.1.2.1 or "iso.org.dod.internet.mgmt.mib". The number-presentation makes the system easy to use with computers. [9]

An SNMP message consists of the version number, community name and protocol data unit. The version number ensures that the same SNMP version is used. The community name is used so that the SNMP software knows that the unit to be managed belongs to the same unit as the manager [40].

The protocol data unit has five fields. It specifies the type of protocol data unit the message contains. It contains an ID-field connecting the responses to the requests. Error status and error index fields specify the error type and associate it with a particular object. The last field is the variable binding field, which associates the values with the objects [28].

SNMP uses five different protocol data units (PDUs) to communicate. The get-request –message is used for requesting values of one or several variables that are stored in the MIB. The get-next-request performs the same operation, but it is used for tables. After the first get-request is completed, the get-next-request returns the next values in the table, when it is sent with the object identifier of the previous

request. The set-request updates the values in MIB. The get-response returns the value that was requested by using the get-request or get-next-request. Finally, the trap-message is used for warning about network events such as shut-downs and start-ups. [9][19]

Properties that are monitored in network management can also be monitored in the context of WWW-applications. Naturally, the state of the computer that runs the service can be monitored. In addition, it is possible for enterprises to register special MIB-modules [32]. This means that the database can be customized to store application specific information in addition to network specific statistics. The data stored in customized MIBs can show, for example, the number of users and the amount of application specific traffic. In addition, task specific information can be stored. This data can be, for example, the number of performed tasks, or the number of the tasks succeeded and failed.

The SNMP and ICMP protocols cover the following part of the customer care tool's functionality area [Figure 4]:

Figure 4 Protocols' coverage of customer care tool's functionality area.

5.1.5 Maintenance of the Software

Network and application maintenance is not restricted to using ICMP and SNMP. There are software tools developed by the communities and commercial companies that make the use of these protocols easier and, in addition, also add other important functionality into these tools.

Tasks that belong to the maintenance personnel are very diversified, which means that there also exist tasks and software that do not rely on using the network management protocols. Electronic mail is an important communication form today. These mail systems need to be maintained and configured so that system remains usable. The Cyrus IMAP mail server [2] and the Mailman [13] mailing list manager are examples of this kind of software. Operations that differ from the operational environment of network management, and what administrators of this software have to do, include, for example, adding and removing users and mailing lists and authenticating users. This kind of administration happens by using the scripts that come with the software.

Another example of different kind of software management tools are the Security Administrator Tool for Analyzing Networks (SATAN) [1] and its improved version Security Administrator's Integrated Network Tool (SAINT) [8]. These tools help administrators find the vulnerable parts of the system. Their mission is to gather information about the network by testing services that are provided. This information is then used to analyze the potential security risks.

The existing body of network management software is versatile. These applications contain a lot of properties that can be used also in customer care tools. However, they also cover areas that are not useful. In addition, these solutions, used as such, would be too heavy in the context of the customer care function. Examples of the network management software are LoriotPro by Luteus [20] and Fidelia by Fidelia Technologies Inc [12]. It can be seen that the tasks of this kind of software can be divided into monitoring, management, and storing data as well as creating reports about the collected data.

LoriotPro allows the administrator to monitor and control the resources of the information system efficiently. These resources contain workstations, network resources, such as switches and routers, applications and data. LoriotPro uses SNMP

as a base for the management operations. A network management system collects data, which can be accessed by using LoriotPro. This software is also capable of determining the structure of the system it is managing. It is also possible to remotely configure the parameters and performance of the system's resources. Configurable parameters are changed by using the set-request of SNMP. Script functions help to accomplish the repetitive tasks. When the need for configuring or locating problems occurs, the target is easy to find by using the search engine accepting both technical and organizational search terms. It is, for example, possible to find computers that are located in a particular room or a particular department of a company, or a device that has a certain performance. Also alarms and reports can be viewed by configuring their order of importance. Management operations can be done remotely by using the Internet browser.

The graphical user interface makes it easier to piece the environment together. Different viewing points are offered to the system. It is possible to form several hierarchical trees for monitoring both the events and the physical system.

Storing the data in a database supports the creation of reports. The tool offers also pre-formatted reports for different situations.

Fidelia is a more versatile management tool. The administration of the system is divided so that users can have different privileges. There may be several administrators that manage different parts of the system. Some users may not have any rights to perform the tasks, but just to monitor the state of the system.

Fidelia can monitor events real time on a network, system or application level. The data is stored to a database and it can be used for real time reporting or for creating long-term statistics. The tool uses the SNMP and MIBs, but it can also use any existing MIB or custom enterprise MIBs. However, Fidelia does not limit itself only to the SNMP-functionality. It provides also monitoring of the other parts of the system. For example, monitoring the HTTP-protocol for browsing information via the WWW and SMTP-protocol for electronic mail as well as the FTP protocol for file transfer is supported. Reports can be created for different purposes, for example, for finding bottlenecks in a system or analyzing the trends of the system use.

Data handling is implemented so that the system will not get overloaded because of the monitoring and testing activities. There exists a separate data gathering unit in the

system. Data is then requested from this unit. This can be done anywhere in the Internet.

An interesting detail is the possibility to monitor URL-level transactions. This kind of functionality is essential for the customer care tool as well. Fidelia performs testing also on the system and network level, as was mentioned before. This makes it possible to monitor critical services such as paying goods by using a WWW-application.

Fidelia is relatively easy to install to work in a standalone mode. This software can be also extended for specific purposes, because it offers an application programming interface for the developers. This interface also allows plugging the tool into other products used in network and application management.

It is also possible to construct a management system by using the components of product families as a point of departure. These kinds of products include OpenView [16] by Hewlett-Packard and Netcool [21] by Micromuse. These products offer tools for people working at various levels of organizations. The idea here is that a solution fulfilling the requirements can be constructed individually by collecting the right modules from a product family and leave away the properties that are not needed.

The network management software covers the following areas of customer care tool's requirements [Figure 5]:

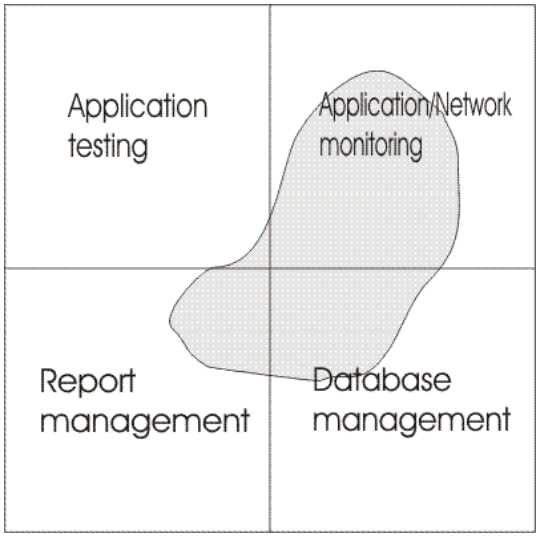

Figure 5 Network management software and customer care tool's functionality

areas.

5.1.6 Testing the software

Software testing is a diversified area. The goal of this function remains the same regardless of what kind of testing is being performed. The phase of the software development, however, determines the kinds of tools to be used in different testing methods. Some testing types are presented below.

In the functional testing, the functionality of the tested system is compared to the specifications. The use of the certain inputs should produce the outputs that are known beforehand. When the testing relies on the knowledge about the implementation details, the type of testing is structural. Code structure, database design and hardware all exert an impact on the functionality of the tool. Information on these parameters can be used to locate the bugs more efficiently than, for instance, when only using the black box tests. [4]

The path testing is one of the most important testing techniques and it should be included in every testing plan. A path can be defined as any executable sequence of instructions that a program has. [4]

The path testing is suitable for low level testing, such as testing of the units. When the system units are integrated and the system is running, the emphasis of testing should be on the transactions. The paths can be constructed of transactions. In this case, a path is any possible sequence of transactions.

Invalid data entered into the system can cause unpredictable results, the consequences of which can be catastrophic from the perspective of the running system. Besides testing whether the system accepts legal inputs or not, it is equally important to test what happens if the application is exposed to illegal inputs.

The regression tests confirm that programming changes in the software do not affect its functionality [41].

The running system can also be exposed to stress, it can be tested how the system recovers from serious errors, and how the system co-operates with different hardware and software configurations. [4]

Software applications for these tests already exist. For example, tools for defect tracking, performance testing and testing management can be found [29].

The examples of WWW-application testing software include e-Test suite by Empirix [11] and WebKing by Parasoft [26]. Both applications offer diversified tools for verifying the functionality of a web application.

Both applications enable the testing of the functionality of a WWW document. They check, for example, that the images, links and scripts located in the document are valid. In addition, they naturally offer tools for testing the functionality of applications. These tests can be divided into load tests and functional tests.

The overall functionality of the application can naturally be checked by using these tools. The paths to be tested can be chosen manually and recorded so that they can be used again if needed. Executing these paths automatically reveal runtime errors, because all the functionality in the path will be tested. The most common paths along which the users navigate can be tested efficiently by using this method.

Applications generate loads that simulate the real usage of the WWW application for the load tests. These tests help observe application scalability and find the possible bottlenecks of the system. Load tests can be extended to cover stress tests, which show how the WWW-application manages to handle large amounts of users. Load tests may be also generated by using the path tests that were previously recorded.

Both systems collect the data about the WWW-application under observation. This data can be later used for improving the performance of the service.

Both applications take care of the same tasks. They both allow the use of scripting languages so that the tests can be customized for maximizing the benefits of these tools. WebKing also tries to offer a possibility to use the system more efficiently so that script programming would not be needed at all. This would make testing faster. However, every application is different, which means that there is no single solution that would take care of every possible task without modifications. WebKing uses the user transaction data and log-files of servers so that realistic load tests can be created automatically. Also the sites that are exposed to load tests may be chosen one by one. The paths to be tested can be created automatically. The application also helps static analysis by applying the rules to the code that is under static testing. This helps to prevent errors that occur often in WWW-applications.

Figure 6 shows how testing the WWW-application software relates to the functionality of customer care tool:

Figure 6 Testing the WWW-application software and customer care tool's functionality areas.

5.1.7 Properties of customer care tool

The tools presented above have many suitable properties for the customer care tool. As a matter of fact, if the techniques and properties of the individual applications presented in this chapter would be combined, all the requirements for the customer care tool would be satisfied at the technical level. However, the use of this kind of a system would be too heavy for the customer care function. There would be a lot of functionality that may not be needed by a person without a deep technical understanding. In addition, developing and maintaining such an application would require a lot of time and money.

The difference between testing the software that was presented earlier and the customer care tool is that the latter works with a system that has real users at the same time as testing is performed. The test sets, their correct results as well as their interpretations have to be developed beforehand. This improves service quality and reduces a need for hiring customer care personnel with technical backgrounds. The testing methods offered by the previously presented software are more extensive that can be used in the customer care function.

There has to be also some kind of network management functionality in the customer care tool so that its performance can be monitored well.

After all, this means that the customer care tool must combine suitable parts of the network management systems and WWW-application testing tools.

6 OWN SOLUTION

6.1 Interviews

The interviews were carried out so a holistic view of the business domain could be formed and the requirements for the customer care tool could be formulated. In this study, a total of twelve companies were interviewed. These companies offer a wide array of services ranging from banking and stock services to library services and electronic newspapers. Information on the persons interviewed is presented in Table 1. If the person interviewed did not have much to do with WWW services, his or her work description is described by using the word "other".

TYPE OF WWW SERVICE	WORK DESCRIPTION OF PERSON INTERVIEWED	NUMBER OF COMPANIES
Electric newspaper	Editor/maintenance person	3
Library	Librarian/maintenance person	2
Intranet of a company	Maintenance person	2
Information service of a bank	Customer care person/other	3
Investment service	Maintenance person/other	2

Table 1 Information about the interviewed companies

The questions on the business domain and customer care tool were handed out to the interviewees in advance so that they could prepare themselves for the interview. The interviews took place in the premises of the company to be interviewed. An average interview lasted about one hour. Despite our list of questions, a serious attempt was made to carry out the interview more like a conversation between the parties rather than just a process of collecting answers.

There was only one interview session with each company. The interviewees were

asked to contact the interviewer later, if additional information emerged after the interview. It was a bit difficult to arrange meetings and that is why additional interview rounds clarifying the requirements for the customer care tool were not arranged.

Questions were divided into the following groups:

- Questions about the business domain
- Questions about the equipment currently used in online testing
- Questions about the personnel
- Questions about the customers
- Questions about testing/monitoring and their goals
- Questions about the weaknesses of current system
- Questions about future trends
- Questions about reporting

In addition, there was a possibility to present ideas not covered by the interview questions but seen as important properties of the customer care tool. These ideas also produced useful information.

6.2 Implementation

The architecture of the customer care tool is designed by utilizing the requirement specification presented in the chapter "Generic Model". The customer care tool is able to test the WWW-application with which it operates. In addition, it monitors this application by collecting data about the defined parameters. This data, as well as problem descriptions collected by the customer care personnel, are entered into the database. This data helps identify the causes for the emerging problems. The tool can also create reports that can be sent to the maintenance personnel so that they may fix the faults revealed by the testing function. This data may also be used for improving the performance of system.

6.2.1 Essential design principles

There are several different WWW-applications that could use the customer care tool. One example is the library service -use case that was described in the chapter "Problem statement". However, libraries are not the only places, which may offer services via the WWW. In addition, various services implementing the same functionality may differ quite a bit, when the inner structure of service is concerned. This means that the customer care tool has to be able to operate with different kinds of services. The functionality provided by services cannot be predicted at the low level. This is why the customer care tool is designed in a way that some functionality and configurations must be created by the maintenance personnel and not necessarily by the developers of the customer care tool. In addition, monitored properties cannot be defined too tightly beforehand. The customer care tool provides a framework that can be easily configured to operate with a new application.

It should be noted, however, that in different applications there exist components that take care of the same tasks. For example, the users have to log in to the system in many applications. This means that it is possible to create instructions for the developers of the WWW-application on the level of these common tasks. This makes the integration of the customer care tool and the WWW-application an easier task. However, formulating these instructions is left outside the scope of this thesis. These instructions may deal with the relevant parameters to be observed and the possible test cases, just to mention a few examples.

The customer care tool should be implemented so that it can be used in several platforms. Hence, the application code should be portable.

The customer care tool should work efficiently and reveal faults, if they exist. The black box testing may show that there exist some kinds of problems in the system. However, it does not tell what went wrong and where. Although the customer may be pleased that the fault is recognized, it is also good that these faults can be fixed as soon as possible. This is why the customer care tool is able to test the system more thoroughly than the black box testing. This decision requires modifications to the the WWW-application that is going to be tested and monitored. However, these extensions are designed in a way that they do not pose any requirements for the language used in the implementation of WWW-application. In addition, implementing these extensions does not require a lot of work.

The architecture of the customer care tool is distributed and there are several reasons for this decision. Table 2 presents the elements of the distributed architecture in general level without the connections between the elements.

SOFTWARE COMPONENT	TASKS OF SOFTWARE COMPONENT
Customer care unit	• user interface for the customer care tool • storage of tests and correct results
Database	• stores information about the state of the WWW-application and reports about the problem situations
Data collecting unit	• collects information about the state of the WWW-application
Data processing unit	• processes test results by using information that the WWW-application delivers during testing
WWW-application	• the WWW-application that is monitored and tested • contains extensions that belong to the customer care tool
Maintenance unit	• delivers tests and user rights to customer care units • configures the WWW-application by delivering information about legal testers

Table 2: Architectural elements of the customer care tool

Testing and monitoring should not cause stress to the WWW-application. Hence, the extensions that are implemented to the existing WWW-application do not process data that they handle. Instead, they just transfer it to a distinct processing component that does not slow down the WWW-application.

For the same reason, there is only one unit collecting and monitoring data about the WWW-application via MIB that contains information about the network and the application. If the MIB of the WWW-application does not exist, it has to be designed also. The exact design of this MIB is not presented in this thesis. The data-collecting unit delivers data to the database of the customer care tool. This causes delays in the availability of the monitoring information, but these delays are not significant. In some situations, it is possible to reduce unnecessary traffic causing an overload to the system. This kind of situation occurred in the library –use case described in the Problem statement section.

There may be several customer care units. This is why the databases containing information about the faults and monitored parameters are separated from the individual customer care units. Every customer care unit utilizes the same databases. The maintenance component of the customer care tool is also distributed. This enables the configuration of different components from one location.

6.2.2 Architecture and component bindings of customer care tool

Figure 7 shows how architectural elements communicate with each other. The arrows in the figure show the direction of the information flows.

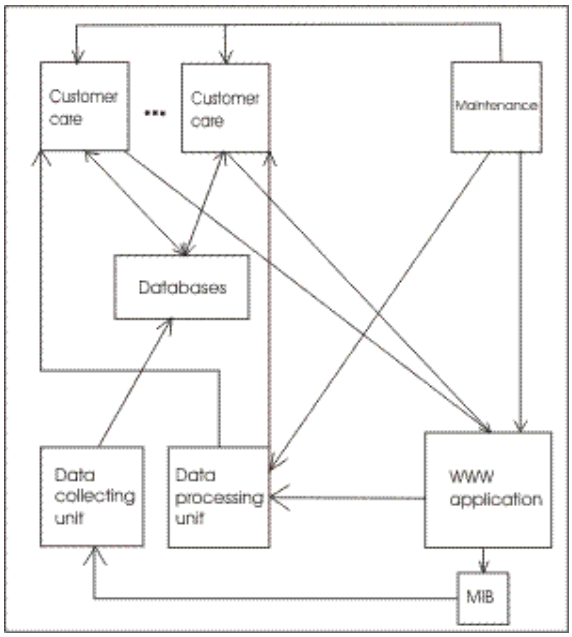

Figure 7 Architectural elements and their interaction

The customer care component makes it possible for the personnel to monitor the state of the WWW-application. This WWW-application can also be tested. Monitoring happens through databases. It is also possible for the customer care personnel to enter information to the database. This information is in textual form and it is about the fault situations experienced by the customers.

The data collection unit compiles information about the WWW-application and the underlying network. This happens via MIB that is customized for the purposes of the application. The MIB has standard network management properties in it, but it also stores information, such as the number of transactions and the response times of the application. The data collection unit gathers relevant information on the MIB on a regular basis by using the SNMP's get-request. Then it adds this information in the database of the customer care tool. The customer care personnel then can use the database for retrieving information. The data collection unit may operate at the same location as the databases.

The WWW-application is tested through its public interface. The application sends relevant information about its state to the data processing unit. The data processing unit processes information to the final form and sends results back to the customer care component. The testing procedure is described more in detail later in this chapter.

The maintenance unit takes care of modifying the system. It takes care of creating the tests and delivering them to the customer care components. Its task is also to inform the WWW-application and the data processing unit about who can test the system by setting the user rights. It should be noted that the maintenance unit does not create any extensions to the WWW-application. All the modifications to the system are installed before it is launched.

6.2.3 Used Technologies

The customer care tool is designed to be implemented with the Java-technology everywhere. The only exceptions are the databases and possibly the extensions of the WWW-application.

The extensions of the tested WWW-application will be implemented by using the same programming language as with the WWW-application itself. This is because it

is typically not possible to add components to the existing software by using a different programming language. Software has to be compiled before it can be launched. This phase requires that the software is developed by using the same language. Without this restriction, customer care tool would work only with software that was implemented in Java.

According to its developers [37], Java can be defined as follows:

"Java(TM) technology is a simple, robust, object-oriented, platform-independent multi-threaded, dynamic general-purpose programming environment. It is best for creating applets and applications for the Internet, intranets and any other complex, distributed network."

It is relatively easy to implement the application logic, network operations, graphical user interfaces and connections to the database system with Java. Platform independence provides that the functionality of the application and the look of graphical user interfaces do not change, if the customer care tool runs in different platforms.

A database system will be added to the customer care tool as a ready-to-use product. It will not be developed for this purpose, because there already exist efficient and reliable solutions. Only the structure of information will be designed and the scripts for initializing the database will be done so that database product can store the data of the customer care tool. These activities are not presented in this thesis. The database language used will be Structured Query Language or SQL, because it is the most used language for queries and modifying databases [38]. In addition, Java provides a data base connectivity kit for accessing databases by using SQL [18].

There will also be software modules in the customer care tool that will work as extensions of the WWW-application that is tested and monitored. In order to work in the WWW-application, these modules have to be written in the same program language as the application. These extensions have to send the results of testing to the data processing unit. In the worst case, the system has to transform the objects so that they can be sent over the network to a data processing unit, where they are transformed again to a suitable form and then processed to the result. In addition, the code for these operations may have to be updated, when the system is updated to evaluate new tests. Naturally, all this cannot be avoided, because the system is

distributed. However, there already exists a solution, which works in this situation and reduces the work that has to be done.

The Common Object Request Broker Architecture (CORBA) developed by OMG is developed for distributed applications [36]. The idea behind CORBA is that software objects can be distributed, which means that the same software in principle operates in several computers. Java has also a system that uses the same idea. However, this system called "Remote Method Invocation" or RMI works only in the applications that are written completely in Java. This is why CORBA fits better in the customer care tool.

CORBA works by using the following principle: Objects can communicate with other objects that are located in a remote computer. This is done by defining the interface of the remote object in the software that is running in another computer. The definition of these objects is made by using the Interface Definition Language or IDL. With the help of IDL, the objects will be able to know which other objects can be called on a remote computer and how this can be done. The Object Request Broker or ORB takes care of the concrete implementation of calling remote objects and transferring data. This procedure takes care of locating the remote object, sending a request to it, waiting for response and delivering a response to the object that made the call.

On the remote computer, there exist another ORB. It receives a request and processes it to a form understood by the object receiving the call. There is also an IDL defined object on this side. This object performs the real request and gets the response from the object. This response is then transferred back the same way as the request was made.

The ORB is the reason why the CORBA is independent from the programming languages. It explains why the objects in different computers can be implemented using different languages. It should be noted that the CORBA hides object transformations from the software developer, because ORBs do not have to be developed. Although the CORBA makes modifying existing software faster, there still exist data transformations, when the data is transferred to a new location.

6.2.4 Monitoring

Application monitoring is done by using a database that contains information collected by the data collection unit. Also the Ping and Traceroute applications of ICMP are used. If the WWW-application or the server, where application is running has no own MIB, it has to be added. This database may also be an SQL-database. In this case, its SQL-requests are conducted from simple SNMP-MIB statements, if there is a need to use an SNMP message format. In addition, if there are no sources from which the data collection unit can poll information, the WWW-application has to be modified so that this monitoring can take place.

The parameters that can be monitored include, for instance, the number of users, amount of requests, the amount of total traffic and response times of a WWW-application. These parameters are general enough so that they are relevant in the context of several WWW-applications. In addition, it is possible to see how they affect the performance of system. The faults caused by stress or concurrent operations may be revealed by using this information in addition to the tests.

MIB collecting information about the parameters can be implemented even if there is no need for it in case that the WWW-application already collects this information. This is because the amount of initial work decreases, if there is a possibility to use a database that already stores the values that are needed. This backup database can be implemented together with the data collection unit, which reduces the amount of traffic needed to request new values.

A data collection unit requests new values regularly from MIB and adds them to the database of the customer care tool. Old values are not erased and they can be used for observing the history of the WWW-application's performance.

Monitoring information that comes from the database is not and it does not have to be real time information. It takes some time for the customer to contact the customer care center and explain the problem situation. For example, if the data collection unit asks new values from the database twice in a minute, there will be information that is accurate enough for the needs of the customer care function, when the customer has explained his or her problem.

The customer care software does not show just numeric values that are retrieved from the database. It is reasonable to form graphs from these values. For example, if the

amount of traffic is monitored, only the total number of requests is not a good metrics for seeing, if there is a peak in the utilization of the WWW-application. The customer care software can create almost real time reports on the history and the current moment, which can be shown by the customer care tool.

Ping and Traceroute can be used for indicating if the WWW-application is reachable and running. If the customer care function is in the same part of network as the WWW-application, this is easy. However, if the customer care unit is located somewhere else, Ping and Traceroute may not be able to access the network where the WWW-application is located. This is because of the firewalls, which may block the requests of these applications.

6.2.5 Testing

Testing must be planned beforehand. The tests that are performed and the analysis of results have to be ready when the customer care personnel need to make sure that everything is working, as it should. The customer care personnel are responsible only for deciding when they want to test the WWW-application. They just launch the testing procedure. Even in this case, the customer care tool may decide which kind of tests, if any, it performs.

Testing does not take place first. Ping and Traceroute mentioned in the previous section take care of checking whether the WWW-application is up and running. If there is already a huge load stressing the WWW-application, it may not feasible to test the system and hence stress it even more. The rules stipulating whether to test or not to test can also be set beforehand.

The tests, correct results and analysis of the test results are located in the customer care tool. These tests aim at capturing the behavior of the WWW-application as well as possible. This is why they cover the different aspects of the WWW-application's normal usage. The tests are path tests, which work as normal requests that come from the users. However, the tests cover only the functionality that does not require real changes in the data that the application handles. If the library case is used as an example, the tests do not really renew loans or reserve books. They just check that everything works correctly to the point where the transactions of this kind are executed. It should be noted that the testing of these kinds of transactions would

require modifying the code of the database, or, for example, the paying system of a bank so that it would be possible to say where the problem resides. Naturally this cannot be done easily.

A general description of the testing procedure follows next. There is also a more concrete use case about how it can be tested, whether the WWW service of the library can correctly check if the library has a certain book in its collections. This use case is explained together with the general description so that a clearer picture of testing procedure can be achieved.

The test is launched by using the URL that has parameters containing information relevant for the WWW application. In addition, two additional parameters are needed. The first one indicates that this request is a test, while the second one is the serial number of this test.

Figure 8 shows how the WWW-application is extended so that it is possible to test its functionality. The squares inside the WWW-application are original software components and the circles are software extensions that have to be implemented. The arrows show the direction of information flows.

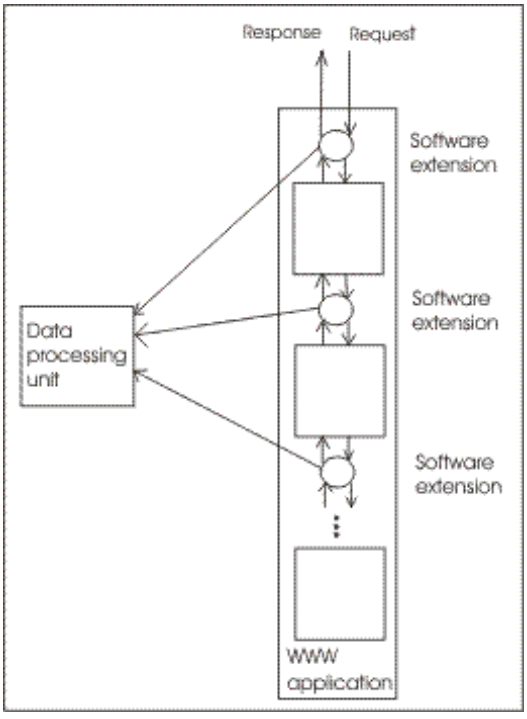

Figure 8 WWW-application and its extensions

Figure 8 is a general model about the WWW-application in a test situation. A simple example about the testing of a library service is presented below. Figure 9 shows the testing of the use case that was presented above. This figure will be referred to later on in the text.

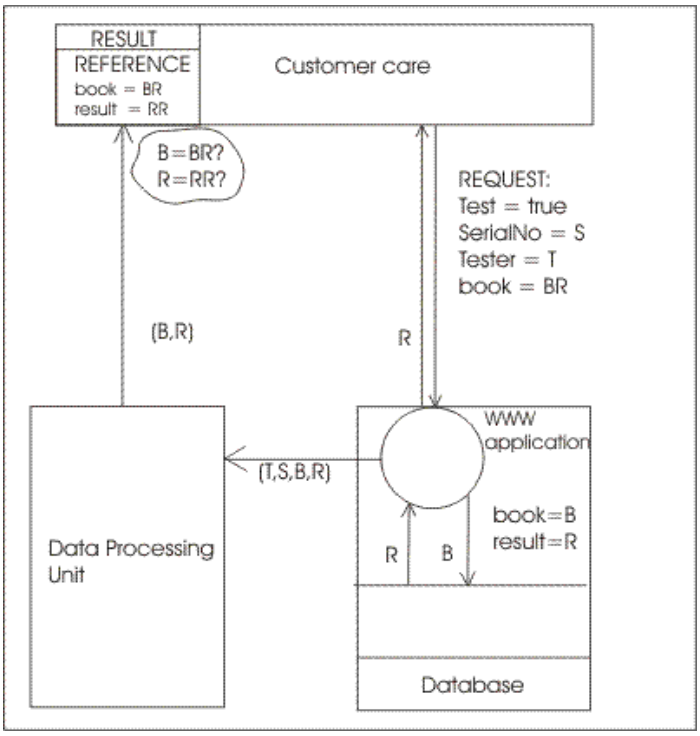

Figure 9 Testing whether library has a certain book in its collection

Software extensions have to be placed in such points in the execution paths of the WWW-application that they can get information about the success of some well-defined subtask that has to be completed when the request is processed. This information indicates if the WWW-application had the right parameters before this subtask was completed and if the result of this subtask were correct. In the library use case, there may be a need to test whether the WWW-application can correctly check if the library has some book in its collection. The software extension can be located in the execution path prior to the component performing database queries. This software extension receives information that makes it possible to check if the identifier of the book is correct, before it is entered to the component that contacts the database. In addition, it gets information that tells whether the result of the database query was correct.

It can be seen that the locations of software extensions are dependent on the WWW-

application. However, general instructions can be made so that these extensions can be added in positions where they reveal as much as possible about the functionality of the software to be tested. These instructions are not discussed thoroughly in this thesis. Instead, they are seen as a topic for a future study. In the testing use case, there is actually only one important software extension, which checks that the arriving request has correct information and that the database query produces the right result.

If the customer care tool was not going to be used, the components of the WWW-application could call each other's methods without extensions. The extensions, which are implemented by using the CORBA are placed in the points where the functionality of the software is going to be tested. The number of these points is not limited. The methods of the original software located in the paths that are going to be tested also have to be extended. They have to distribute information about the test that is executed. Instead of calling the software component, which should handle the request, the caller component calls the extension, if the extension lies next in the execution path. The caller object adds also information about the tests to the parameters. The extension that receives the request collects information about the parameters of the request. It then calls the component that should handle the request with the same parameters. When the response is received, information about it is also gathered. Then the same response is returned to the caller. A test request goes through the same path in the application than the request that is not a test. The only difference is that in the test case, the extension module transfers the parameters of the request and the response, as well as the information of this particular test to the data processing unit that waits for the test results.

The first extension module receives all the requests that the WWW application gets. It checks them so that it can decide whether a request is a test or a real request of the WWW-application. In addition, the address, from which the request came, is checked. If this address is identical to the address of some particular customer care unit, the test can be made. Otherwise the fake test is discarded. If the request is not a test, it calls the next original software unit in test path without any information about tests. Otherwise the information about the tester and the serial number of the test are added to the request. This information makes the test identifiable. When the response is received, it is delivered to the caller object. Although the first extension to the path delivers the results to the tester, this result will not be analyzed. The test result containing all the information is delivered by the data processing unit. If new test

paths are going to be developed, it should be noted that adding new extensions requires also the updating the WWW-application.

In the testing use case, one parameter of the request shows that the request is a test. In Figure 9, the parameter “Test” has a value “true”. The sender’s address reveals that the party which made request is the customer care center of the library. In Figure 9, there is a parameter “Tester” for this purpose. However, this information is received from the sender’s address. The serial number of the test is also collected so that the test can be identified. In Figure 9, the parameter “SerialNo” has a value “S”. Information about the book that is being searched can be achieved in a similar manner as with normal requests (BR in Figure 9). This does not require any changes in the message format. All this information is collected and then the database component is called with all the information that is needed so that it can be checked whether the library has a certain book. It should be noted that there might occur transfer problems in the network. Hence, it cannot be said that the WWW-application receives the information called “BR”. This is why the information about the book is marked as B when it is received. When the results (R in Figure 9) are received from the database component, they are also collected. After that all the information about the test, the state of the parameters prior to the database request and the results of the database query are transferred to the data processing unit. In Figure 9 this is marked as “(T,S,B,R)”.

The extensions cause an update need also in the data processing unit, so that it knows what kind of calls will be executed by the WWW-application.

The data processing unit gets information from all the extension points that are in the test path, if the WWW-application is tested. Hence, the data processing unit has to know how many extension points there are in the application and which points lie in the execution path. This information is received from the original request, which was sent to the WWW-application. By checking the URL, the path can be defined. The number of extension points is defined, when the data processing unit is implemented. If new tests containing new paths are going to be developed, the data processing unit has to be updated. In the testing case, there is only one point, which can be checked from the URL. In this case, there will be only one call to the data processing unit, which transfers all the information.

The data processing unit gets the information about the test, the state of the calling

parameters and the responses. Test information reports who ordered this test and what is the serial number of the test. This makes each test unique. This makes it also possible to construct a test result message containing information about a particular test and not to mix different tests. When all the information composing a particular message is received, the data processing unit transforms the information to the form so that it can be compared to the results located in the customer care unit. The final test result message is transferred to the customer care unit who made the request. In the library testing case, the data processing unit is called only once. After the tester is identified, the results are processed. In this case, all the information can be presented in text format. The identifier of a book and all the information that can be achieved from the database by using this identifier are added to the test result. Then the data processing unit sends it back to the customer care unit that launched the test ("(B, R)" in Figure 9).

The tool carrying out the test in some particular customer care unit receives the result and compares the result message and the predefined information indicating what result should be. If the actual test result differs from the predefined result, there is some kind of a fault in the system. The customer care personnel cannot tell what is wrong, if there are no descriptions of the potential faults in each exception point. If the customer care tool notices that there exists a fault in the system, it shows this description to the person who ran the test. It also automatically creates a report of the test by adding the test request and the results to the test description. This report is stored in the database, but it is also sent to the maintenance personnel.

In the library testing case, the book identifier parsed from the request in the software extension of the WWW-application is compared to the identifier that can be found in the reference test result. In Figure 9, these are marked as "B" and "BR". Also the real result of the database query is compared to the corresponding reference ("R" and "RR" in Figure 9). If the book identifiers do not match, it can be seen that information is not transferred correctly over the network. If the database queries do not match, there is a bug in the database component executing the query, or in the database itself.

This kind of testing reveals also how widely the fault affects the functioning of the system. After the point where a problem occurred in the execution path, every possible subpath in execution causes presumably wrong responses, unless there are some other bugs in the software that accidentally fix the error.

6.2.6 Reporting

Reports are made automatically when testing reveals problems in the WWW-application. In addition, the customer care personnel can ask the customer care tool to create reports when they are needed. If the system is tested, reports contain information about the test result and the correct result. In addition, reports contain information about the state of the system at the moment of testing, or at the moment when a report was requested. The report creation process relies on the data retrieved from the database. It is not necessary to make any requests to the WWW-application or monitor the network. The database makes it possible to create a history of how the state of the WWW-application has changed. The customer care tool makes it possible to send reports to other components, store them in a database as problem descriptions and to print them.

Reports can visualize information by using graphs about different parameters of the system. In addition, it is possible to add problem descriptions to the report if required. Graphical reports can be shown also in the customer care component of the tool. The report can be updated so that the state of the system can be observed almost real time. This improves service quality, because the requests to the database do not have to be made, while the customer is waiting.

The customer care personnel can also create reports based on the problems of the customers. These descriptions are entered to the database, in a hope that they could help in situations, where similar problems occur.

6.2.7 Maintenance

The maintenance unit takes care of creating the tests and updating information about the customer care units so that the WWW application and the data processing unit have the information relative to who has the right to test the system and to whom the results have to be sent.

In addition, the maintenance tool takes care of delivering the rights of the users dealing with the customer care tool. Not everybody has the permission to use the customer care tool and not everybody who is allowed to use this tool has the rights to maintain it. These issues are discussed below.

The creation of tests can be carried out anywhere. However, the tests are entered to the system by using the maintenance tool. The maintenance tool cannot distinguish between the legal tests and the illegal ones. In addition, it does not know how the results should look like. Finally, it cannot create descriptions about what went wrong. This is why the tests and the descriptions should be designed by the persons who know how the WWW-application works, what are the important transactions performed by the customer, and what are the important points in the execution paths of these transactions. It should be noted that the tests have to be designed in a way that they utilize existing extensions of a running WWW-application. Tests and their results are naturally dependent on the WWW-application to be tested. However, different WWW-applications have components that perform similar tasks. Thereby, it is possible to develop instructions to the persons creating the tests. These instructions indicate the critical parts of different components. This thesis does not cover the design of instructions.

The maintenance tool aims at delivering test information by contacting every unit that has something to do with the updates. If these units cannot be reached, they are probably down. This is why the components that can receive updates from the maintenance component, request new updates every time they come into operation.

6.2.8 Security

There are some issues in the customer care tool that deal with data security. Only the customer care personnel and maintenance people are allowed to use this software. This reduces the number of users, which means less stress to the WWW-application. In addition, only the people with some knowledge of the application are allowed design the tests.

Another security issue involves the possibility that the testing data may reveal some information when transferred over the network. In addition, the test sets and information about the customer care function must be transferred over the network so that this information is not changed during the transfer.

Everybody who has the right to use the customer care tool has a user name and a password that are entered to the tool when the tool is going to be used. The user rights may be defined independently for every user. There are two categories for user

rights. First, there is the customer care personnel who may use the tool but not maintain it. Second, there is the maintenance personnel who can do anything with the tool.

If communication securing is needed, information is transferred by using the public key cryptography. The idea is that the information to be sent can be encrypted by using the public key of the receiver. This information can be decrypted only by using the private key of the receiver. Private and public keys are delivered when the software components are installed. If all components of the customer care tool are located in a safe network, it is not necessary to encrypt the data. However, if the test results reveal something critical about the WWW-application, and the customer care units or the data processing unit are not in a safe network, communication needs to be secured.

The integrity of the transferred data can also be checked by hash functions and public key cryptography. Hash is calculated based on the information to be transferred, encrypted and added to the data to be transferred. This hash is also calculated in the receiving unit. If these calculations equal, the data has not changed during the transfer.

7 ANALYSIS

The architecture of the customer care tool specifies one solution seeking to satisfy the criteria by implementing the requirements defined in the chapter "Generic Model". However, the customer care tool is used in special environments, which makes it difficult to completely satisfy every criterion that is set for the general solution. It is a challenging task to design such a system.

7.1 Customer care tool and evaluation criteria

7.1.1 Success of interviews

This criterion analyzes how the interviewing process could be carried through and how the results of the process could be utilized.

The interviews were carried out during the time period between mid-July 2000 and mid-August 2000. This time is a very popular holiday season in Finland. This is why it was quite difficult to find enough participants for the interviews. In addition, the interviewees had only a limited amount of time to spend for an interview. Finally, there were not as many interviewees taking part in this study as was expected in the beginning of the study. In addition, people in companies that were going to be interviewed were quite busy and it was also difficult to arrange interviews later during the fall. The time limitations set for this project made it impossible to postpone the interviews any further.

It was pretty difficult to find people with knowledge both on the technology and the tasks of the customer care personnel. The technical staff and the customer care personnel rarely overlap. In addition, they do not usually have knowledge about the business areas other than their own. It was usually either a customer care person or a technical expert taking part in the interviews, not both of them. This made it somewhat difficult to form a holistic picture of the requirements of testing and monitoring system.

The concept of utilizing a testing system in the customer care function was proved to be a new idea. This sometimes caused problems, because it was understood that the

goal was to gather information about the test system that is used only by the technical personnel. It was also understood that the testing system was planned to be an application used in the phase when the system providing services was set in action for the first time.

Interviewees could not formulate their needs very clearly, because the concept of the customer care tool was quite new to them. Another problem associated with the interviews was that the interviewer could not impose his idea of the optimal characteristics of the customer care tool on the interviewees. This is because the essential goal of the study was to get first-hand information on the needs of the customers.

The first couple of interviews represented a completely new situation for the interviewer also. This created some awkwardness in interview situations, because it was quite difficult to clarify the concepts to the interviewees for the first time. However, this problem did not last very long. In addition, it cannot be said that the results of these first interviews were any weaker than the results of the remaining interviews. They gave as much information about the customer care tool as others.

The overall impression of the interview situations was that they were carried out decently. In every single interview, it was possible to get information about the properties of the customer care tool. In addition, it was possible to clarify the concepts that were ambiguous to the interviewee so that he or she could answer the questions.

There was only one interview round meaning that all the information was captured by the first take. If there had been more meetings, the requirements may have become more precise. However, this may have caused problems in the analysis, because not every company has the same needs.

Important requirements and properties were chosen by prioritizing the most frequently presented ideas. One problem was that there was not just one company, which was interviewed about a support tool for the customer care function. Services that are offered to customers vary from one organization to another, which naturally introduces some variety in the requirements for the customer care tool.

It was possible to form a requirement specification addressing most of the questions presented by the interviewees. However, it was not possible to create such a

requirement specification that would cover every issue that emerged from the interviews. This is still acceptable, given that the goal of the study is to define the general properties of the future's customer care tool, not only one solution for a particular WWW-application. Had this not been the case, the solution would have been contradictory to the criterion of universal applicability.

7.1.2 Universal applicability

This criterion aims at solving whether the customer care tool can be used with various WWW-applications. It is hard to satisfy this criterion, because different services provide different functionality and, in addition, they are not constructed completely by using the same design principles. Reality sets limitations and this criterion can be fulfilled only after some initial work is done, before the customer care tool is up and running. It can also be seen that by satisfying this criterion, the customer care tool cannot have special properties relevant only in some applications. For example, the customer care tool cannot offer already implemented functionality by testing whether the library service mentioned in the "Problem statement" -chapter works properly, when the customer wants to check what material is available in the library. This kind of functionality must be added later in the tool by using general mechanisms, which the architecture of the customer care tool provides.

By choosing the requirements of the customer care tool in a way that they satisfy the most relevant needs of the customer care tool, it was possible to implement a system that provides the functionality that is needed in several cases. In addition, generality makes it possible to configure the customer care tool in a way that it works well in several environments. The criterion of universal applicability and the criterion of small maintenance need are contradictory. This is because universal applicability is achieved by increasing the amount of configuration work, which is usually done by the maintenance personnel.

The only testing form that is used is path testing. This requires extensions to the application to be tested, but it does not set hard limits relative to what parts of the system can be tested. The generality of the system is not lost in the sense of universal applicability, because it can still operate with any WWW-application.

Monitoring of the network connections is necessary in every environment. Ping and

Traceroute do not require any special action so they can be used in several situations. In addition, the monitored parameters of the WWW-application are chosen in a way that they are useful in all environments and do not tell anything that is important only for some types of WWW-applications. Suggested parameters are the number of users, the number of transactions made, response times and the total amount of traffic.

Database operations are not specific to any type of WWW-application. The values that are received by monitoring are general as was explained above. In addition, problem descriptions that are stored in the database by the customer care personnel require nothing from the system that is monitored and tested.

Universal applicability can be reached, but as it was seen, it is contradictory to some other criteria. This criterion has as a consequence that this tool cannot be used as a ready-to-use solution. This creates a need for work to be done before the customer care tool is ready to be used with some particular WWW-application. This can be seen as a drawback.

7.1.3 Capability to examine the state of the WWW-application

The purpose of the customer care tool is to make it easier for the customer care personnel to tell if there is something wrong with the WWW-application. This can be done only if there is information available about the current state of the WWW-application and its environment. The designed solution should fulfill these criteria so that this tool could be useful for the customer care function.

The monitoring and testing capabilities of the customer care tool are parts of the design trying to satisfy the needs that this criterion highlights. Badly designed components do not bring enough data about the WWW-application so that the customer care personnel could deliver relevant information to the customer.

The network used for data transmission and the WWW-application itself are parts that affect the performance of the WWW-application. The state of these parts can be examined by the customer care tool. The functionality of network is checked by using the existing solutions based on ICMP. The WWW-application is exposed to path testing that can be launched by the customer care personnel. In addition, the customer care tool collects information about the usage of the WWW-application.

Network monitoring can be done well, if the customer care unit launching the operation of Ping or Traceroute is located in the same network than WWW-application. Then it is possible to check if there exists a connection between customer care unit and WWW-application. This assures that the network of the WWW-application's service provider is working properly. However, there may be problems in the network that located between the customer and the service. If firewalls do not allow these applications to enter to the other network, it is impossible to check whether the customer can be reached by customer care tool. Naturally, this will not indicate the route of the customer's requests. This means that the network-monitoring problem is solved only partially in a sense that it is not possible to locate the problems in the connections between the customer and the WWW application in a satisfactory manner. This is the problem of the customer care units today. It is also understood that the problems of this kind are hard to solve.

Testing is also a problematic case. A WWW-application is running and having users while its state is being examined. A system that is running cannot be tested so thoroughly as the application that is just implemented and not in use. However, the preliminary testing prior to the system is taken in real use should remove all the faults. This is taken into account when the customer care architecture was designed. In addition, there was no attempt to try every possible testing technique for running the systems.

Path testing can produce a lot of information about the WWW-application. However, this depends on how the paths and extension points in these paths are chosen. The design does not affect the way how this should be done. As a result, it is possible to make bad decisions relative to what to test.

The information on the WWW-application is not real time information. There exist some delays, because the application is not polled all the time. However, as it was explained in the "Implementation" section, these delays are not significant, because the customer care unit has the right information at the moment when the customer contacts the customer care center.

The customer care tool is able to collect information about the state of the WWW-application. This information does not cover everything that it is possible to check in the system. However, it satisfies the criterion well enough so that sufficient amount of information can be received about the actual state of the system.

7.1.4 Transparency of the customer care tool

Good service quality delivered by the customer care unit means that the customer can get advice and information from the customer care when he or she needs it. The customer care tool has to operate in a way that helps it to perform this task. Still, it must not make the performance of the WWW-application worse and thus weaken overall service quality. In the worst case, the use of the customer care tool would cause more work for the customer care personnel, because of the low level of the WWW-application performance. The criteria of the transparency of the customer care tool states that the WWW-application should function as well with the customer care installed in the system as without it.

The customer care tool affects the network and the WWW-application by increasing the amount of traffic in the network. This is because of the Ping and Traceroute operations, testing requests, database operations and WWW-application monitoring. In addition, the extensions that are implemented in the WWW-application cause more work. There exist more function calls in the application. Also the parameter values must be transferred to the data processing unit.

There may be several customer care units in the customer care tool. However, they do not produce a lot of traffic. Testing requests are not big, because they are just normal URL's. In addition, the number of them remains reasonable, because it is not supposed that there would be dozens of customer care units and they are probably not testing all the time. Finally, there exist a use policy in the customer care tool. If there already exist a lot of traffic in the network, the WWW-application will not be tested, although the customer care unit would try it. It is only possible to check that the WWW-application is up and running. Database operations are frequent, because the state of the WWW-application is monitored. However, there exist a need for requests for new values only when the data collection unit polls the WWW-application. The customer care units alone do not operate against the transparency criterion.

The data collection unit is added to the design because it reduces the stress that falls on the WWW-application. There exists only one unit that collects data and inserts it to the database. The situation would be the opposite, if every single customer care unit would request new values from the WWW-application. This solution is scalable, because the number of monitoring requests sent to the MIB of the customer care tool remains small and transparency stays the same.

The amount of Ping and Traceroute traffic may be quite large. However, the customer care tool does not need to use big packets and it does not have to send them on a regular basis, which means that it does not block or slow down other traffic.

The software extensions of the WWW-application affect the functionality of the system all the time. If the request comes from the customer, software extensions pass the arguments forward and return the results to the caller. In this case, the extensions do nothing else and the delay caused by this kind of operation is not significant.

When testing is performed, information about the parameters of the function calls is collected and transferred to the data processing unit. This is reasonable, because the process of combining test results and sending them to the customer care unit happens in one place. This increases the transparency of the customer care tool, because this task is performed in a distributed unit and hence it does not slow down the WWW-application. It should be noted that the remote calls made by CORBA to transfer the information to the data processing unit slow down the WWW-application. Still, without CORBA, one option is that the WWW-application and the customer care tool should be written in Java, which is contradictory to the 'universal applicability' criterion. In this case, RMI would be used instead of CORBA. If RMI was not used, the extensions could be programmed in the same language as the WWW-application. In this case, the programmers implementing the software extensions should also take care of transforming the data in the format that it could be transferred over the client-server connection to the data processing unit. This would be contradictory to the 'small maintenance need'-criteria. Naturally, there would be an option to do data processing in the WWW-application, but this is against the transparency criterion, because building tests results and sending them to the customer care unit takes some time.

The transparency criteria can be fulfilled with the design of the customer care tool. Obviously, the customer care tool cannot be implemented in a way that it would not affect the WWW-application at all. However, the design is done so that it seeks to minimize the stress to the system being monitored and tested, while still obtaining enough information about it.

7.1.5 Small maintenance need

Every application has to be installed first so that it can be used. A need for updating the software later may emerge when there are faults in the application that are fixed with software patches. In addition, there may be new versions of the application or new properties published, which may cause the need for updating the software. If installing the software and especially updating it becomes a time demanding task, it will increase the workload of the maintenance personnel. In addition it takes more time to take new versions in use. The criteria of small maintenance need demands that installing and updating do not become troublesome for the personnel seeing that the right version of customer care tool is running.

There exist several tasks that are directed to the customer care tool maintenance personnel. First, the maintenance personnel have to install software components. In addition, they have to create extensions in the WWW-application so that it can be tested properly. Tests and their results have to be created so that the software extensions can be utilized. Finally, also user rights for the customer care and maintenance personnel have to be created. However, this task is not demanding and it does not increase the workload of the maintenance personnel.

It can be seen that this criterion can not be satisfied completely because the maintenance personnel has to create software code so that the customer care tool can be used in testing. In addition, coding may need to be done again, if the WWW-application being tested is updated when its structure changes. This is because also the paths being tested may change, when the WWW-application is updated.

Although several tasks are directed to the maintenance function, there exist reasons why this design principle was chosen. Universal applicability could not be achieved, if tests were designed before the customer care tool is installed. If software extensions were not be coded at all, the state of the WWW-application could not be monitored very closely. This is against criterion which requires that the state of the WWW-application must be checked by using the customer care tool. The tool would not satisfy the criterion of fast performance, if tests were created only then when they were needed. In this case, the customer care personnel would have to create the tests, which means that they should have a technical background and knowledge of how the WWW-application is implemented. Hence, this situation would not satisfy the criterion of the possibility to use the customer care tool without technical

background. It can be seen that the criterion of small maintenance need is contradictory to some other criteria.

However, the customer care tool is tried to be designed in a way so that the tasks of the maintenance unit personnel require as little work as possible. Path testing requires software extensions in the WWW-application and the data processing unit. All the extensions in the WWW-application are similar kind and pretty simple. In addition, CORBA makes it possible to leave the task of information coding to the underlying ORB.

ORB takes care of transforming the parameters to the right format also in the data processing unit. It is a bit more complicated to update the data processing unit, because its task is to build test results by using the information from the WWW-application. In addition different data types transferred to the data processing unit require different type of transforming so that the test result can be built. These results have to be comparable to the correct test results located in customer care units.

The creation of tests can be done by using the maintenance part of the customer care tool. However, this requires that the creator of the tests knows about the structure of the WWW-application. Also, the right results that will be compared to test results need to be constructed, as well as the descriptions of potential problem situations.

If it is known already at the moment when the WWW-application is being implemented that the customer care tool is going to be used, it is easier to add extensions to the software than later when the system is already in use. In addition, when the customer care tool is installed, it is reasonable to add all extensions that may be needed later at the same time. This reduces work to be done in the future. In this case, there is no need to update the WWW-application and the data processing unit later, unless the WWW-application itself is updated. New tests can thus be added only by using the maintenance component of customer care tool.

The criterion of small maintenance need can be solved only sufficiently, because it is contradictory to the other criteria. However, it can be seen that good work in the installing phase reduces the amount of maintenance work later, when the customer care tool is in use.

7.1.6 Capability to utilize several data sources in the same application

As described in the 'Previous work' –chapter, there already exist applications that provide information about the state of the network and applications being monitored and tested. In addition, there exist information about the problems that have been identified earlier. However, not all this information is provided by the same application, which causes some problems. This is because the information provided by some specific tool, such as a testing application, cannot be immediately used together with the information provided by some other tool, such as a monitoring application. This is seen as a deficiency. The customer care tool should take care of offering information in a way that the data produced by different components can be shown together.

The reporting system utilizes all the information that the customer care tool stores. Test results reveal more, when the information about the state of system is added along. The automatic creation of reports eases the work of the customer care tool by eliminating the need to collect information manually from different sources. This also results in faster performance. In addition, it is possible to get information about the same kind of problems that have occurred earlier.

The customer care tool has to satisfy also the criterion of universal applicability. That is why it cannot provide information about all the special parameters that would be worth monitoring. There can be useful information that cannot be reached by the customer care tool. Testing reports can avoid this problem by offering detailed descriptions about every particular test. However, general reports that do not have anything to do with any particular test case that is entered to the system cannot utilize an approach of this kind. It may be useful to request only general reports about the state of the system in situations where the customer reports about the problem and the problem description are entered to the database. A monitoring report may be useful when it is added to problem description.

It can be seen that information can be collected in a way that the customer care tool gets versatile information. This also makes it possible to utilize different data sources usefully. A slight drawback is that parameters being monitored are predefined. It would be nice if it were possible to define the monitored parameters, which would make it possible to provide more useful information.

7.1.7 Possibility to use customer care tool without a technical background

The customer care tool is used by people with varying levels of technical knowledge. Part of the customer care personnel does not have any technical background. However, they should also be able to use the customer care tool.

The customer care tool is designed in a way that its functionality is semi-automatic. Some of its tasks are performed automatically. The data collection is performed this way. Also the report creation is automatic in testing situations. Actions required from the users are quite minimal. They have to give a command if they want to test the WWW-application or create a report on its state. However, these operations are very easy and they do not require more than just pressing a button.

The greatest amount of work is required when the problem descriptions are searched from database. However, this task can also be performed easily, because it only requires that the keywords are entered to the tool.

It can be seen that the tasks can be performed easily and this satisfies criterion according to which a person does not need to have a technical background to be able to use the customer care tool. However, the results of the actions performed by the tool have to come in an understandable format. Graphs give clear information about the state of the system. When the system is tested, the customer care personnel do not have to interpret results themselves. The interpretations are made while the tests are being created. This makes the work of customer care unit easier, because, in principle, they do not need to understand anything about the findings of the testing process. They can give customers a clear explanation. In addition, when the problems occur, reports are automatically sent to the maintenance unit so that the customer care personnel does not have to explain what was tested and what were the results. It should be noted that the customer care tool provides also technical information, which can be delivered to customer if he or she needs to know about it.

The customer care personnel do not need have a technical background. However, before starting their jobs, they should take part in a short course teaching them what a WWW-application does, how it works and how the customer care tool can be used. This means that that the customer care personnel should always know at least the basics about the domain where they work.

It can be seen that the information delivered by the customer can easily be created in

an easily understandable format. The work of the maintenance personnel affects the results. However, the criterion is satisfied well.

7.1.8 Fast performance

The fast performance of the customer care tool means that customer care persons can get all the information they need while they are communicating with the customer. There are several aspects that together affect how well this goal can be reached. Some of them were already discussed, when the customer care tool was evaluated against the previous criteria.

Information about the state of the WWW-application can be fetched from the database. This is done regularly, which means that there always exists almost real time information to be observed. The delay caused by the fact that system is not polled all the time is not significant as mentioned earlier. This means that the customer care tool satisfies the criterion of 'fast performance' when the monitoring part of the tool is evaluated.

The testing case is more complex. This is because also the performance of the WWW-application affects how quickly results can be produced. If there exist a huge load that falls on the WWW-application, performance suffers. However, the customer care tool does not automatically test the system when the network or the WWW-application is congested, which improves the performance of WWW-application and reduces waiting times in the customer care unit. However, the monitoring capabilities of the customer care tool can still tell if problems occur because of stress.

The customer care tool does not leave the process of interpreting the results to the customer care personnel. This contributes to fast performance. In addition, the fact that the maintenance personnel create interpretations about the test result both at the technical and general level makes it easier to tell the customer what is happening in the system.

Finding existing problem descriptions from the database requires most from the customer care personnel, because they have to listen to the customers and type suitable keywords to the customer care tool so that relevant information can be found. In addition to the database operations, the customer care personnel have to analyze which problem descriptions received from the database are relevant for the current

situation. This also takes some time. However, this operation cannot be changed because database queries cannot be made in other way. Waiting times should not be long and they do not slow down the process very much.

Java and CORBA may not provide the fastest possible implementation. However, they make it possible to satisfy the criterion of 'universal applicability'. In addition, they try to reduce the implementation work falling on the maintenance personnel, thus fulfilling the criterion of 'small maintenance need'.

The criterion of 'fast performance' can be fulfilled by the customer care tool. However, there still exist tasks that must be performed completely by the customer care personnel. In addition, some parts of operability customer care tool are dependent on the WWW-application that is being monitored and tested. The performance of these tasks cannot be affected by the customer care tool.

7.1.9 Overall fulfillment of the criteria

The architecture of the customer care tool is designed in a way that it satisfies quite well the criteria presented in the chapter "Criteria". However, there is one criterion that could not be satisfied as well as the others. This criterion is somewhat contradictory to others, which caused some problems.

The criterion causing problems is "small maintenance need". However, by giving more responsibility to the maintenance personnel, it was possible to implement a system that performs its tasks efficiently when installed and running with the WWW-application. In addition, the maintenance work can be done in a way that the most tedious tasks are performed in the phase when the customer care tool is installed.

7.2 Simplifications

The designed architecture is not perfect, although it describes the functionality and the interaction between the components of the customer care tool. It is possible to use the architecture that is designed and implement the customer care tool based on it. However, there will be some work left to be done, before the design of the technical specification for inner functionality can be started.

The process of designing the architecture by first interviewing the companies, analyzing the results and then designing the architecture by utilizing the requirements derived from the results was simplified. If it had been possible to make another interview round after constructing the requirements and designing a simplified architecture model, changes to the architecture could have been made, if needed. The architecture could have been constructed on a more detailed level by designing the interesting components so that their interfaces would have been ready. This means that the requests and data to be transferred while the components interact would have been defined exactly.

Although it is mentioned that the data collection unit component may have include the MIB that is polled so that monitoring information can be collected. However, the details of the MIB are not described. It would be necessary to define the structure for custom MIB so that the monitored values can be fetched from it by using the get-request of SNMP.

7.3 Remaining problems

The criterion of 'small maintenance need' could not be solved. The architecture was designed in a way that the other criteria evaluating the customer care tool as a system in use were satisfied better. This is because the key idea behind designing this architecture was to create a tool that would facilitate the work in a customer care unit.

The interviews revealed that every service provider had had problem related to monitoring the network connections. It is possible to check if there exists a connection between the WWW-application and the customer care unit by using Ping and Traceroute, if there exists no firewalls between the customer care unit and the WWW-application that would block any traffic they create. It is impossible to say anything about how the route between the customer and the WWW-application is constructed and where the problems occur. This problem is not solved in this work.

Architecture design leaves tests and their results to be designed and implemented by the maintenance unit of the customer care tool. Customer care tool provides a possibility to implement an efficient and useful system. However, if caution is not exercised, when the tests and their results are created, the tool may not work optimally. This kind of design is acceptable so that universal applicability can be

achieved. In addition, it would be good, if some kind of instructions could be included with the customer care tool so that harms caused by this approach could be reduced.

7.4 Future work

The architecture of the customer care tool is presented in this study. The next logical step would be to design the inner structure of the components and the database structure. After this phase, the customer care tool is ready to be implemented and the solution to be tested.

There was only one interview round during this study. Hence, there was no possibility to get feedback from the interviewees during the design phase. However, the future work can fix this shortcoming in the system-testing phase. If there is a possibility to use several different kinds of WWW-applications, it is possible to identify some weaknesses of the customer care tool and try to improve them. This may require co-operation with the service providers of the WWW-applications.

Universal applicability is important factor in the customer care tool. If the tool proves to be useful with many types of WWW-applications, the components of the customer care tool can be designed to be more like a product family. This makes the tool more flexible and hence the amount of configuration work can be reduced. It should be noted that constructing this kind of framework is not easy. It would require a marketing study so that information about the special functionality of the existing solutions can be gathered. Results need to be analyzed so that it can be decided which solutions will be supported by the customer care tool. A risk analysis should also take place prior to the implementation of the product family, since the construction of this kind of frame work requires a lot of resources.

It is mentioned in this study that there should be instructions available for the maintenance personnel for creating software extensions and tests. These instructions are not developed in this thesis, and they are left as a part of a future work. Different WWW-applications have similar software parts. For example, many applications require that the customer logs into the system before he or she can use the service. Instructions can be developed by collecting information about the components that can be found in several applications and by analyzing the most important operations

of these components. Tests and software extensions can then be created by utilizing this information. This kind of model makes it easier to focus on the critical parts of functionality.

8 CONCLUSIONS

The goal of this thesis was to find out what are the critical properties that a customer care tool supporting a WWW-applications should have. The customer care unit assisting customers with problems with Internet services is a relatively new concept. Thus, it is not quite clear yet what can be done in the customer care unit to help the customer. In addition, the customer care personnel have different kinds of skills so it will be important that they have tools that help deliver relevant information to the customer regardless of their level of technical understanding. The application to be designed should help personnel so that the questions presented above could be solved.

The first objective of this work was to interview people in companies with WWW services so that it would be possible to determine how important it would be to design a customer care tool. In addition, as the customer care tool was seen as an important addition to the customer care unit, the requirements for this tool were derived by analyzing the results of the interviews. The architecture presents a solution that can be utilized in several customer care units so it is not limited to a particular WWW-application. This was the second objective of the work.

The customer care personnel are contacted when the customers cannot use system or when there are some troubles either in the network or the WWW-application itself. The customer care tool tries to ease performing tasks by giving advice on the problems of this kind.

The state of the WWW-application can be observed by using the information collected from the running application. If it seems that there may be errors in the system, the WWW-application can also be tested. It is also possible to check if the network connections are working by using the customer care tool.

The tool has also a database, which contains information about the previous problems experienced by the customers. By using this information, it is possible to find answers to the questions that were asked before and that the customer at the moment has.

It was interesting to notice how the customer care environment affects the properties

of the customer care tool. There already exist technologies that satisfy the different parts of the requirements that were given to the tool. However, the customer care personnel need to use these existing properties more lightly than they are usually implemented in the already existing software. In addition, if the solutions were designed by directly combing the applications that together satisfy the requirements, the system would be too difficult and heavy for the customer care use. The idea behind the functionality of the customer care tool is that the customer care tool gets information about the system, but it cannot affect the WWW-application or the network. If there exist problems, the tool informs the parties that know what needs to be done. The customer care personnel use the tool passively. Almost everything is done automatically by the tool, which reduces the amount of knowledge required from the users.

The results show that it is possible to implement the customer care tool that can be used in different environments and with different WWW-applications. This kind of universally applicable solution leaves some parts to be implemented to the phase, when the tool is taken in use. The amount of work depends on how detailed information the customer care personnel want to receive about the state of the WWW-application.

Service quality is important in all the situations, where the customer contacts the service provider. This means that the most desirable situation is the case where the customer never needs to contact the customer care unit because of some problems in the WWW-application. It can be seen that if the customer care unit is needed, this goal has not been reached. However, it is important to have good tools for dealing with difficulties that cannot be solved by the customer alone. The customer care tool is a valuable solution for such cases.

9 REFERENCES

[1] Anon, SATAN, 2.7.2003 [Referred 2.7.2003], <http://WWW.fish.com/satan/>

[2] Andrew Systems Group, Project Cyrus, 25.1.2003 [Referred 2.7.2003], <http://asg.web.cmu.edu/cyrus/>

[3] Bennett, B. A., E-business definition, 17.12.2000 [Referred 30.6.2003], <http://WWW.bretabennett.com/babrwc/e-business_definition.htm>

[4] Beizer B., Software System Testing and Quality Assurance, International Thomson Computer Press, 1996, 358p.

[5] Barker, J., What is the Internet, the WWW, and Netscape? An Introduction, 27.9.2000 [Referred 4.1.2000], <http://WWW.lib.berkeley.edu/TeachingLib/Guides/Internet/WhatIs.html>

[6] Birdwell, J. D., What is software engineering?, 4.1.2001 [Referred 4.1.2001], http://hickory.engr.utk.edu/~birdwell/ee205/lecture4/sld004.htm

[7] Cisco Systems, Inc, Network Management Basics, 20.2.2002 [Referred 1.7.2003], <http://WWW.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/nmbasics.htm>

[8] Clarke Computer Company, SAINT – Network Security Scanning, 2.7.2003 [Referred 2.7.2003], <http://WWW.clarkecomputer.com/services/saint.html>

[9] Comer, D. E., Internetworking with TCP/IP, Volume I, Principles, Protocols and Architecture, 3rd edition, Prentice-Hall Inc., 1995, 613p.

[10] Connolly, C., Web Naming and Addressing Overview (URIs, URLs,…), 9.7.2002 [Referred 2.7.2003], http://WWW.w3.org/Addressing/

[11] Empirix, Web Testing: Improved Web Application Testing: e-Test suite,

3.7.2003 [Referred 3.7.2003],
<http://WWW.empirix.com/Empirix/Web+Test+Monitoring/testing+solutions/integrated+web+testing.html>

[12] Fidelia Technology Inc., Fidelia Network Management Software, <http://WWW.fidelia.com/products/technical_overview.php>

[13] Free Software Foundation, Mailman the GNU Mailing List Manager, 2.7.2003 [Referred 2.7.2003], <http://WWW.list.org/>

[14] Haikala I., Märijärvi J., Ohjelmistotuotanto, 5th edition, Gummerus Kirjapaino Oy, 1998, 385p.

[15] Halsall, F., Data Communications, Computer Networks and Open Systems, 4th edition, Addison-Wesley, 1996, 907p.

[16] Hewlett-Packard Company, Hewlett-Packard OpenView, 2.7.2003 [Referred 2.7.2003] <http://WWW.openview.hp.com/>

[17] HKUST Cyberspace Center, Planning the network security, ICMP, 1997 [Referred 30.6.2003], <http://WWW.cyber.ust.hk/handbook4/03c_hb4.html>

[18] Horstmann C.,S., Cornell G., Core Java 2, Volume II-Advanced Features, The Sun Microsystems Press, 2000, 920p.

[19] Institute of Electronics And Telecommunications, SNMP protocol : Message Format, 1.7.2003 [Referred 1.7.2003], <http://WWW.et.put.poznan.pl/snmp/protocol/pformat2.html>

[20] Luteus, NMS LoriotPro DataSheet, 2.7.2003 [Referred 2.7.2003], http://WWW.loriotpro.com/Products/LoriotPro/LoriotPro_EN.htm

[21] Micromuse Inc., Netcool Solutions – Products – Home, 2.7.2003 [Referred 2.7.2003] <http://WWW.micromuse.com/products/>

[22] MOREnet, Ping and Traceroute | Technical Support | MOREnet, 14.2.2002

[Referred 1.7.2003],
<http://WWW.more.net/technical/pingtrace.html>

[23] Niehaus D., What is a distributed system?, 22.7.2003 [Referred 22.7.2003] <http://WWW.ittc.ukans.edu/~niehaus/classes/s99-750/notes/HTML/CH-1/ppframe.htm>

[24] Network Solutions, Inc, What Is TCP/IP?, 2.3.1998, [Referred 4.1.2001], <http://public-library.calgary.ab.ca/train/overview/tcpip.htm>

[25] Network Sorcery, Inc., ICMP type 8, Echo request message, 30.6.2003 [Referred 30.6.2003], <http://WWW.networksorcery.com/enp/protocol/icmp/msg8.htm>

[26] Parasoft, WebKing, 3.7.2003 [Referred 3.7.2003], <http://WWW.parasoft.com/jsp/products/home.jsp?product=WebKing&products/webking/quick/about.htm>

[27] Postel J., Rfc 792, Internet Control Message Protocol, September 1981 [Referred 30.6.2003], <http://WWW.networksorcery.com/enp/RFC/Rfc792.txt>

[28] Reddy M. S., messageformat, 1.7.2003 [Referred 1.7.2003], http://alpha.fdu.edu/~mogulla/messfor.html

[29] Quality Assurance & Software Testing Portal, Software Testing & Quality Assurance Links: Tools, 13.5.2003 [Referred 3.7.2003], <http://WWW.qalinks.com/Tools/>

[30] Research.Ihets.Org, What is a Quality of Service (QoS)?, 25.6.2003, [Referred 25.6.2003], <http://research.ihets.org/research/ipvc/qos/articles/qos_intro.jsp>

[31] SearchNetworking.com, ping – a searchNetworking definition, 1.1.2002 [Referred 30.6.2003] http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci214297,00.html

[32] Simple Web, The Simpleweb – Enterprise specific MIB modules, 11.7.2002

[Referred 1.7.2003]
<http://WWW.simpleweb.org/ietf/enterprise/>

[33] SolarWinds.net, An Overview Of ICMP and its use in Network Management, 30.6.2003 [Referred 30.6.2003], <http://support.solarwinds.net/Help/Reference/ICMP.htm>

[34] Stallings, W., SNMP, SNMPv2, and CMIP, The practical guide to network standards, Addison Wesley Publishing Company, 1994, 625p.

[35] Stanford Graduate School Of Business, Stanford GSB: CEBC: Hanson Book Review, 12.12.2000 [Referred 12.12.2000], http://WWW.gsb.stanford.edu/CEBC/book_rpt/Hanson.html

[36] Sun Microsystems, Introduction to CORBA, 22.3.2003 [Referred 9.7.2003], <http://developer.java.sun.com/developer/onlineTraining/corba/corba.html>

[37] Sun Microsystems, The Java(TM) Platform, 9.7.2003 [Referred 9.7.2003], <http://java.sun.com/allabout.html>

[38] Ullman, J.,F., Widom J., A First Course in Database Systems, Prentice Hall, 1997, 470p.

[39] Whatis.com, trace route – a whatis definition – see also: traceroute, 30.8.2002 [Referred 30.6.2003], <http://whatis.techtarget.com/definition/0,,sid9_gci213209,00.html>

[40] University of Western Ontario, Departement of Computer science, SNMP Message Format, 1.7.2003 [Referred 1.7.2003], http://WWW.csd.uwo.ca/courses/cs457/reports/handin/kywu/A2/snmp_msg.htm

[41] Whatis.com, regression testing. A whatis definition, 3.12.1999 [Referred 3.7.2003], <http://whatis.techtarget.com/definition/0,,sid9_gci212884,00.htm>

[42] World Wide Web Consortium, What is Hypertext?, 22.7.2003, [Referred 22.7.2003], <http://WWW.w3.org/WhatIs.html>

[43] Whatis.com, whatis.com: Website, 23.11.1999 [Referred 4.1.2001], <http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,213353,00.html >