

MASTER

The influence of user expertise on the usability experience interfaces for different users at Vodafone call centers

van Lümig, C.J.L.

Award date:
2009

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

The influence of user expertise on the usability experience

Interfaces for different users at Vodafone call centers

C.J.L. van Lumig

July 2009

The influence of user expertise on the usability experience

Interfaces for different users at Vodafone call centers

C.J.L. van Lumig

0576893

Master Thesis

July 2009

Supervisors:

dr. J.R.C. Ham

TU/e, Department of Technology Management

dr. W.A. IJsselsteijn

TU/e, Department of Technology Management

drs. T. F. van Buitenen

Vodafone NL, Manager CM Process and Project Management

Eindhoven University of Technology - Department of Technology Management
Master's Program Human Technology Interaction

Preface

This Master Thesis is the final product of my graduation project, which is part of the Master program Human Technology Interaction at the Eindhoven University of Technology. This graduation project was conducted at the Customer Experience Management department of Vodafone Libertel N. V. During this project I investigated the usability of the software programs used by Vodafone call center agents. Furthermore, I investigated the influence of user expertise on the usability experience. I gained experience in conducting and reporting scientific research, and I learned how to use research methods relating to software usability and user expertise.

During my graduation project I received excellent supervision and valuable contributions from my supervisors at the Eindhoven University of Technology and from my supervisor at Vodafone. I would like to thank Jaap Ham and Wijnand IJsselsteijn, and my supervisor at Vodafone, Tom van Buitenen. Furthermore, I would like to thank all my Vodafone colleagues, and all the Vodafone agents that contributed to this project, for helping me during my graduation, and also for having a great time. Finally, I would like to thank my family and my friends for their support and personal advice during this project.

Chris van Lumig
Eindhoven, July 4th 2009

Summary

Since the existence of software programs there have been issues and concerns about their usability. Is the used screen size big enough, are the buttons positioned at a logical place, are there any key shortcuts that can help users at better performing their work? These are all usability issues one should keep in mind when designing a new software program. Basically, usability consists of three components: efficiency, effectiveness, and user satisfaction.

Based on concerns about usability, one could argue that both novice and expert users prefer interfaces with a higher usability, but especially novice users will benefit and appreciate usability improvements because expert users are already used to the “old” system. This question will be studied in the current research. More specifically, we will study usability issues related to the two main programs (Ciboodle and Gemini) that are used by Vodafone call center agents.

Vodafone has several call centers which all use software programs. Because of the large-scale use of these programs, it is important to establish a good view of the current usability. This insight is necessary in order to be able to make improvements.

First, we studied how the usability of both programs could be improved. In order to do this, we first had to get an in-depth view of the programs. We choose heuristic evaluations (Nielsen et al., 199) to do this. Via a checklist both software programs were evaluated from a usability viewpoint. In order to investigate whether the results of the evaluation are a reflection of the issues that actual users run into when using the software programs, another method called unstructured observation (Wixon et al., 1990) was used. During these observations, several usability issues were identified. The most important ones (existing shortcuts and user profiles) were used in the next phase of this study.

Second, we studied whether these improvements actually led to higher usability. Also, we studied whether there was an influence of user expertise on usability experience that was caused by the interface improvements. These questions were studied by running an experimental user testing (Nielsen, 1994). In the experiment we first assessed usability measures of the initial interface (session 1). Next, we provided users with the improved interface and they could use this for about a week, after which we assessed usability measures of the improved interface (session 2). Confirming our expectations, results indicated that the improved interface led to higher efficiency and more satisfied users. Furthermore, results indicated that, for novices, there were improvements in usability (they became faster) when using the improved interface as compared to using the original interface. These improvements were not there for experts.

Third, users were asked for their opinions about the majority of all identified usability issues. This was done using a usability questionnaire. Based on this feedback, we proposed recommendations for further improvement of the software package interfaces.

In general, the current study indicates that experts do not necessarily benefit from interfaces with a higher usability. Novices do however show an increasing performance due to the usability improvements. The outcome of this study allows to design interfaces which lead to an increased usability experience for users of different expertise levels.

Table of Contents

Preface	v
Summary	vii
Table of Contents	ix
1 Introduction	1
1.1 Vodafone	1
1.2 Research question	1
1.3 Conceptual overview.....	3
2 Literature review.....	5
2.1 Technology in general.....	5
2.1.1 Theory of reasoned action	5
2.1.2 Technology Acceptance Model	5
2.2 Technology and the user	6
2.2.1 Usability	6
2.2.2 User Expertise.....	8
3 Heuristic Evaluation	11
3.1 Introduction	11
3.2 Method	11
3.3 Results	11
3.4 Discussion	25
4 Unstructured observations.....	29
4.1 Introduction	29
4.2 Method	29
4.3 Results	30
4.4 Discussion	42
5 User Testing	45
5.1 Introduction	45
5.2 Method	49
5.3 Results	52
5.4 Discussion	56
6 Usability questionnaire	59
6.1 Introduction	59
6.2 Method	59
6.3 Results	61
6.4 Discussion	64
7 Discussion & Recommendations.....	67
7.1 Discussion	67
7.2 Recommendations.....	69
8 References	71
9 Appendix	73
9.1 Ciboodle: user testing tasks.....	73
9.2 Gemini: usability questionnaire.....	77

1 Introduction

Since the introduction of the mobile phone, many developments have taken place. Adding several functionalities is surely enjoyable for customers, but they also cause for the need for extra support. Vodafone offers such support via call centers.

This Master Thesis will investigate the effect that improvements of the user interfaces of the software packages used by these call centers will have on the usability experience of Vodafone call center agents. Will the usability experience actually increase when the user interface is improved? Also, we would like to know how these improvements translate into cost-reductions that Vodafone can achieve because of the improved systems' usability. The next paragraph will introduce you to Vodafone.

1.1 Vodafone

Vodafone Libertel N.V. is one of the biggest telecom providers in the world. The company offers a broad variety of services and products such as mobile Internet, email on your mobile phone, but also PDA's and other technical devices. Having over 4 million customers in the Netherlands alone, there is an increasing need for efficient customer service. Whenever a customer has questions about Vodafone services or products, they can contact the Vodafone call center and talk to one of the agents that work for Vodafone. It is vital to help customers as quickly and accurately as possible. Unsatisfied customers will eventually stop using the services provided, and might consider changing provider.

One Vodafone department that focuses on the customers' needs in particular is CEM (Customer Experience Management). CEM exists of several sub departments, each with their own focus points and goals. Two important departments are B&P (Billing and Provisioning) and Projects. CEM B&P continuously drives internal business improvements in order to exceed customers' expectations and to improve efficiency regarding subjects such as account set-up and provisioning of services, number portability, and migration. CEM Projects wants to introduce new services and manage and improve Customer Service project management skills and procedures. CEM has close contact with the Vodafone call centers, which gives them the ability to receive direct input from Vodafone customers. Being able to "hear" the voice of the customer, CEM gets to play the role of the customers' advocate.

The agents that work at the call centers use different custom-made software packages to assist them when helping a customer on the phone. The two main programs that agents use are Ciboodle and Gemini. Ciboodle serves as a portal to live network information. Gemini contains a large database that holds all the customers' information such as subscription type, services in use, but also billing information.

1.2 Research question

Vodafone has several call center departments, each with their own specialties. The departments all use different functionality subsets of the main packages, Ciboodle and Gemini. The remainder of this research will focus on three call center departments in

particular: Postpaid Consumer, Prepaid, and Distribution. This selection is based on the diversity with which the software is used by each particular department. Postpaid Consumer handles all issues and questions related to the largest customer group of Vodafone clients; the private consumers. The prepaid department answers questions/complaints via the phone, handles the administration of all customer contracts and registers new prepaid clients. The last department, Distribution, handles the support for sale points and performs retentions or sets up new connections for Vodafone customers.

From a usability point of view, it is important to know whether these software packages are easy to use, but also whether different user levels can manage to work with them effectively and efficiently. If a certain type of user experiences problems with the software, one should consider adjusting a part of the software interface such that this user becomes able to perform the tasks at hand properly. Obviously, this has to be done for every user group. Danino (2001) nicely described this as the golden rule of usability: "There's no such thing as a "user error"!"

This research is focused on investigating the impact of the users' level of expertise on the usability experience of the two main software packages used by Vodafone call centers. It is important to investigate differences in problem solving strategies, learnability methods, and overall work efficiency because they can all influence usability. A higher 'usability score' leads to more satisfied users, who perform better when enjoying their work. The main research question is:

What is the influence of user expertise on the usability experience?

This question still is very general, and in order to give a more detailed answer to the question, we decided to split up the main research question into smaller parts. First, we needed to investigate possible usability issues that were present in the current software packages. Also, we needed to know exactly what the differences are between the different expert levels of Vodafone agents. The following two questions were used to investigate these matters:

1. What are the user interface design flaws of Ciboodle and Gemini?
2. What are the differences between novice and expert agents in the evaluation of software when the usability is improved?

Based on these answers, we will propose usability improvements that will be put to the test by conducting an experiment. This experiment will give us an answer to the next question:

3. How can the usability of both software packages be improved?

Besides visualizing the impact that improvements of a user interface can have on the experience of the user, we also like to translate what these improvements mean for Vodafone. Therefore we will answer another question:

4. What impact can the improvements have on cost reduction?

Next, you can find an overview of methods that were used during this research, and also in which Chapter each particular method will be addressed.

1.3 Conceptual overview

Chapter 2 gives an overview of previously conducted research to introduce you to the field of technology use, the technology acceptance model, and more specifically, usability. The Chapter will also discuss the most important definitions of usability.

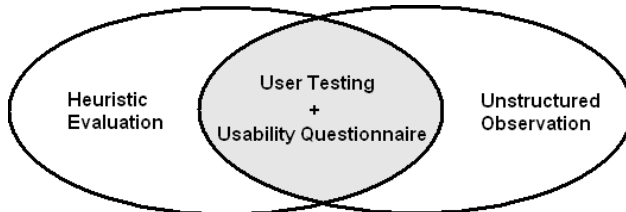


Figure 1. Four methods were used to research usability.

After introducing you to the field of our study, we will describe the four methods that were used during this research (figure 1.). Chapter 3 discusses the performed heuristic evaluations (Nielsen et. al, 1990). This particular method helped us to quickly get an overview of the used software packages, but also to detect usability issues. Next, we had to investigate whether the found issues are of concern for the actual users (Vodafone agents). Chapter 4 describes the unstructured observations (Wixon et al., 1990) we used to observe agents in their natural working environment. This way we could detect errors and came up with ideas for improvements. One other important reason why we choose this additional method is because a heuristic evaluation analytically considers the characteristics of a system, but it fails to encompass important aspects such as safety and comfort (Wickens et al., 2004). These aspects are very important to the user. To test whether the improvements would really lead to increased usability, we ran an experimental user testing (Nielsen, 1994) in which we investigated the differences between the agents' expertise levels and their usability experience over time. Also, we measured whether the usability improvements actually influenced performance. Chapter 5 describes this experiment in detail. The last method, a usability questionnaire, will be described in Chapter 6. Possible improvements that could not be tested during the experiment were presented to the user via a questionnaire. This user feedback can also be very useful when evaluating the programs' usability. We will conclude by discussing the results of this research and give recommendations for improving the investigated software packages. This will be done in Chapter 7.

2 Literature review

This Chapter will introduce you to several theories and concepts that serve as a foundation for this research. We will start by briefly describing the theory of reasoned action. This theory contains the basic components that are relevant for a more important theory used in this research, namely the technology acceptance model. Next, we will introduce you to the concept usability, which stands central throughout this whole study. Also, user expertise will be explained to you.

2.1 Technology in general

In this paragraph we will describe the theory of reasoned action and the technology acceptance model.

2.1.1 Theory of reasoned action

The theory of reasoned action (TRA) was first introduced by Fishbein et al. (1975). This theory describes the link between several components that are needed for a person to act. These components are attitude, behavioral intention, and subjective norm (figure 2). Fishbein et al. (1975) suggest that a person's behavioral intention is dependent of the attitude towards that behavior, and the subjective norm (social context). The behavioral intention can be defined as a person's relative strength of intention to behave a certain way. Attitude entails the belief a person has about the consequences of performing that behavior. The subjective norm is dependent on the context in which the behavior takes place. These three components combined will determine whether the actual behavior will take place.

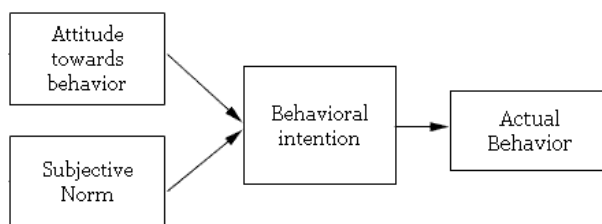


Figure 2. TRA components that lead to predicting certain behavior.

2.1.2 Technology Acceptance Model

In 1989, Fred Davis and Richard Bagozzi developed an extension for the already existing Theory of Reasoned Action (Fishbein et al., 1975). This model is known as the technology acceptance model (TAM), and it contains some basic components of the TRA. The focus of this model is on the acceptance to new technologies. Two components are added to the original theory: perceived usefulness and perceived ease of use (figure 3). The former relates to the degree to which an individual believes that using the system will actually improve his performance. The latter relates to the degree to which an individual believes that using the system will go effortless.

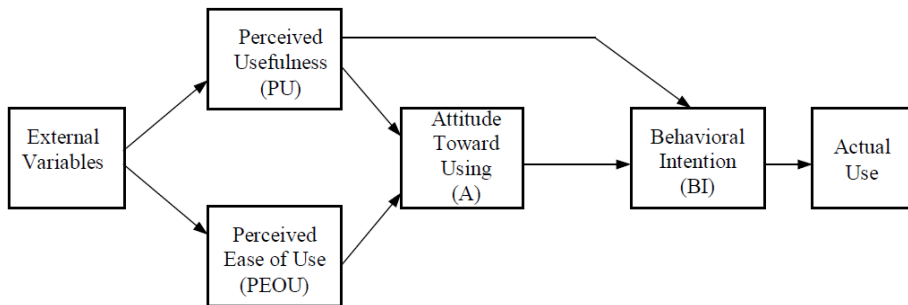


Figure 3. Technology Acceptance Model (TAM) as presented by Davis et al (1989).

Even before people start using a new technology, they begin to form attitudes and intentions toward trying to learn how to use the system (Bagozzi et al., 1992). This is especially of concern for designers, because the user creates his own opinion about, and attitude towards that technology inside his mind, disregarding how good the system actually might be. Davis (1989) points out another important fact, namely that perceived ease of use can contribute to an improvement in performance. If the system is easy to use, the user has to put less effort into the system, thereby sparing more effort to execute the actual task.

2.2 Technology and the user

In section 2.1 we discussed the relation between attitude and intention on behavior. Besides this relation, there is another relation to be discussed, namely the one between the product and the end-user, or in this study, software programs and Vodafone agents. McNamara et al (2006) proposed three primary elements that should be considered: the product itself (functionality), the interaction between the user and the product (usability), and the user (experience). This research will focus on the last two elements in particular. However, user experience will be replaced by user expertise. This will not cause a problem in this study because expert users have more experience than novice users.

The first element, functionality, refers to the product itself. What can the product actually do? What functions does it have? These questions are of less concern during this research because we like to put the user and the products' usability central.

2.2.1 Usability

There exist many definitions of the concept 'usability'. All researchers agree on the basic components of usability, but they use somewhat different components to describe usability. We will now present the most important descriptions of the concept usability.

In 1993, Jacob Nielsen described usability making use of the concepts efficiency, learnability, memorability, errors/safety, and satisfaction. All these concepts can influence system usability and therefore should be properly taken care of during, but also after the design process.

Nielsen
Efficiency
Learnability
Memorability
Errors/Safety
Satisfaction

Table 1. Usability concepts as described by Nielsen (1993).

Another important definition of usability is described in the ISO 9241-11 standard (Bevan, 1994) as

“..the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”

Effectiveness describes the accuracy and completeness with which users achieve specified tasks. The second component, efficiency, can be expressed as the relation between the resources expended and the accuracy and completeness with which users achieve their goals. Another important usability factor, satisfaction, is the comfort and acceptability of use by end users (van Welie et al, 1999).

ISO 9241-11
Efficiency
Effectiveness
Satisfaction

Table 2. The ISO definition of usability

Yet another description of usability was presented by Ben Shneiderman (1998). It consists of five measures of human factors that can help to evaluate human factors goals. Figure 4 shows these five measures. It shows that usability combined with utility defines the usefulness of a product or service. In this figure (4), utility refers to a list of technical specifications.

The first human factor that is important for a usable product or service is that it should be easy to learn. The product or service also needs to be efficient to use, meaning that only necessary actions should be expected from the user. The third factor states that the system should be easy to remember such that the next time the users know how to get certain things done. Obviously, the system should also help prevent users from making errors and the overall usage should be pleasing and comfortable.

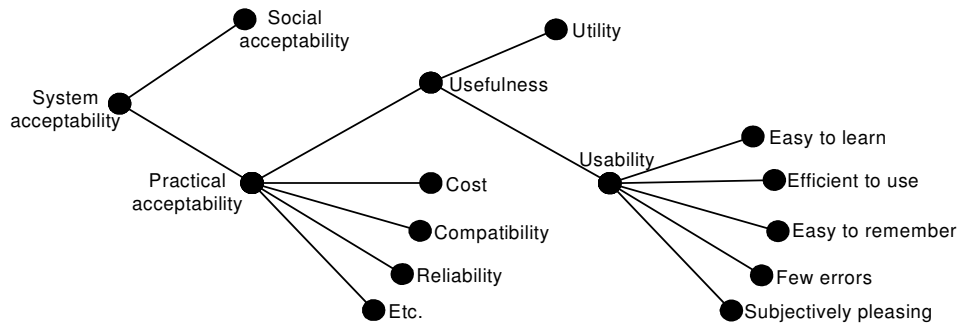


Figure 4. Human factors relevant for system usability.

Shneiderman
Speed of performance
Time to learn
Retention over time
Rate of errors by users
Subjective satisfaction

Table 3. Shneiderman's components of usability (1998).

Improving the usability of a system should be on the main agenda of any designer, but it might be a hard thing to accomplish. Sometimes it is inevitable to improve one of these usability factors without worsening another one. Improving efficiency, for instance, can have negative effects on learnability (eg. shortcuts). This last concept is especially of concern for novice users. The next paragraph will go more into detail about users of different expertise levels.

2.2.2 User Expertise

As stated by Farrington et al. (2006), expertise can be described as skills, knowledge or abilities, in tasks, activities, jobs, sport, and games. It can refer to a process such as decision-making or it can refer to an output such as a decision (Farrington et al., 2006). Expertise consists of several components such as social elements, but also memory capacity and perception of patterns as well as characteristics of processes underlying cognitive activities such as problem solving (Glaser and Chi, 1988). Although one might think that experts are indeed special because of their knowledge, Farrington et al. (2006) says that research suggests experts score poorly for reliability, accuracy, calibration and coherence in judgment and decision making and that they appear to be unaware of these shortcomings.

There are various sorts of users and they all have different levels of expertise. Shneiderman (1992) makes a distinction between novice, knowledgeable intermittent, and expert users. This research focuses mainly on novice and expert users.

Novice users know the task they have to perform, but they have little or no knowledge of the system (Sheiderman, 1992). As stated by Wickens et al. (2004), a designer should focus on ease of learning and low reliance on memory when designing software for novice users. The learnability of the system should be one of their main priorities.

Expert users have deep knowledge of the task, the goal, and the actions required for reaching this goal (Shneiderman, 1992). They are mainly concerned with an easy to remember system. This helps them to use the system more rapidly. Wickens et al. (2004) furthermore notes that a designer should strive to develop an interface that provides information and input mechanisms that map onto the task. This means that if the task is very complex, it will probably take a while to learn how to use the interface.

As a user interacts more with a system, their experience is likely to increase gradually. The time it takes a user to shift from novice to expert depends on several factors like previous experience, system usability and how often the user interacts with the system. It may be clear that this process occurs earlier for some users than for others. Training and personal guidance can speed up this process. Especially in the field of information retrieval, novice users perform poorly with the existing search tools (Allen, 1994). Research by Mendoza et al. (2005) proved that not only a users' experience increases after a certain time of interacting with the system, but also the level of frustration drops. More interestingly, Mendoza et al. (2005) suggests that the sorts of errors that are most prominently featured in conventional usability testing are likely of little consequence over longer periods of time.

Other researchers have proposed the assumption that there are actually differences in the types of knowledge that exist in the minds of novices and experts. LaFrance (1989) states that experts not only just possess more knowledge, but also that this knowledge is represented in a different way. These differences express themselves in aspects such as underlying schema, goal-orientation, practical focus, categorical chunking, cognitive complexity, automaticity of expert problem solving, and the episodic nature of expert memory (LaFrance, 1989). This would mean that practice alone will not make a novice turn into an expert. Only if the basic qualities are fit, a certain type of user can become an expert (LaFrance, 1989). We will try to investigate whether these assumptions are also applicable to the agents that work at the Vodafone call center. Chapter 3 will start this investigation by describing the heuristic evaluations that were performed.

3 Heuristic Evaluation

3.1 Introduction

In this Chapter we will describe the use of two very effective checklists to evaluate the usability of Ciboodle and Gemini. One of the most important researchers in the field of usability is Jakob Nielsen. Nielsen et al. (1990) describes heuristic evaluation as an informal method of usability analysis where a number of evaluators are presented with an interface design and asked to comment on it. The evaluators simply look at the interface and then pass judgment according to what they think is right. It usually takes over 30 evaluators to find nearly all usability problems. Because it is almost impossible to find so many evaluators, normally 3 to 5 evaluators will do (Nielsen et al., 1990). These evaluators will find about two thirds of the usability problems, which is quite good for this inexpensive evaluation technique (Nielsen et al., 1990).

Although we did not have enough resources to use more evaluators, we were able to critically investigate the software's usability by using a very detailed guideline for designing user interface software (Mosier et al., 1984). Following Nielsen's advice, we also tried to apply his 10 principles for user interface design to both interfaces (Nielsen et al., 1990).

3.2 Method

Materials and procedures

Because most Vodafone agents use two programs during their daily work, this evaluation will consist of two parts. The programs to be evaluated are Ciboodle and Gemini.

We started the evaluation by logging in to the software, using an account that was created specially for us. After a couple of turns stepping through all the menus, we continued by selecting which parts of both checklists were appropriate for our software packages. After this selection, we returned to Ciboodle and began our evaluation. We compared each step mentioned in the checklist against the actual design. Where differences were existent, we made notes for later discussion. This process was repeated until both checklists were used to evaluate both programs.

3.3 Results

First, we will describe the evaluation of the software package Ciboodle. Next, we will describe the evaluation of Gemini. Mind that all design principles and rules used in this results section are copied from Mosier et al. (1984) and, were explicitly noted, from Nielsen et al. (1990).

Ciboodle evaluation

Data entry

While the customer verification process should be accomplished using the information presented in Ciboodle, agents often switch to Gemini to retrieve this information because

the information presented in Ciboodle is not always up-to-date or correct (eg. the date of birth is sometimes not correctly inserted).

The used data entry fields are clearly visible because of a special border that is printed around these fields. However, some data fields, for instance those in which network information is presented, also uses this border. It is therefore not completely clear which fields the user can actually edit, or can only be updated by the system. In some cases, this updating by the system is a consequence of a user action. Live information is acquired from the network. This process sometimes took more than the 0,2 seconds, which is the standard time it should take for the system to give feedback upon a user action. In some cases WE timed a delay of 60 seconds or more. This is obviously too long, and especially since the user cannot retrieve the system status at this moment, this problem should be taken care of. This is also a violation of Nielsen's' first usability principle (Nielsen et al., 1990), which states that the system should always update the user about the current status of the system, especially when the system takes control from the user in order to perform a task.

When users have to enter long numbers, one should use separators in order to reduce the chance of making an error. The SIM (Subscriber Identity Module) card number consists of 19 numbers, which have to be entered manually when the old SIM card is stolen, or lost. Because the numbers are displayed as one string it is very easy to miss- read/type a digit.

Using abbreviations can help to speed up the work, and is especially useful for advanced users. Creating a call logging takes less time because the user does not have to type out all words, but can just summaries the conversation while using these abbreviations. Besides abbreviations, a user can also use keyboard shortcuts (eg. ctrl + c for copying a piece of text) to work faster. Ciboodle presents users the option of creating a case logging via 16 blue shortcut buttons or via 3 logging trees (figure 5.). The shortcut buttons instantly select a frequent occurring case topic. This is exactly what Nielsen's seventh design principle is about. Selecting a button is quicker than using the logging trees, because the button only requires one click, while the logging tree requires at least three clicks. In order to quickly find an option in the logging tree you can use a filter, which is displayed below the logging trees. This filter does however not work very well, since it only searches the first letters of a sentence. When you enter 'sms' into the filter, all options without 'sms' in the beginning (for instance 'incoming sms') are removed from the list. The filter should actually search for any particular word in that whole sentence.

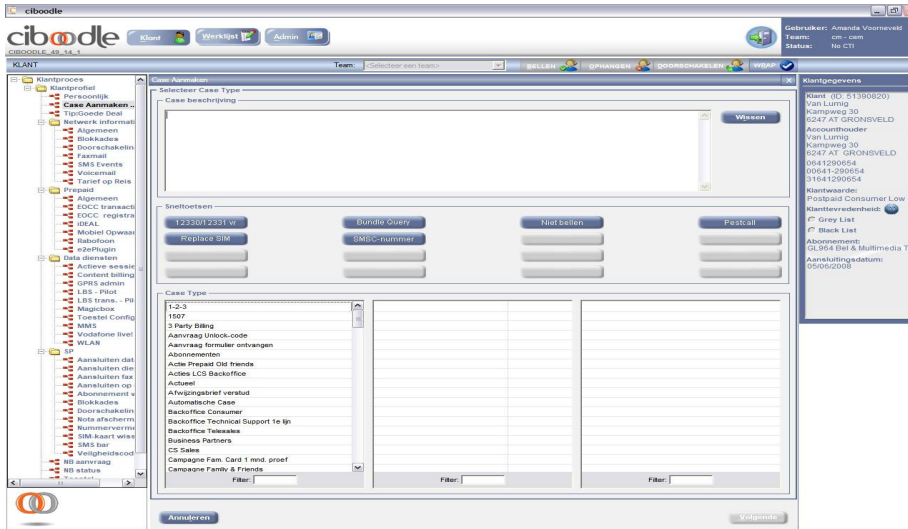


Figure 5. Main window in Ciboodle: the user can select the topic of the conversation.

To continue, there are some issues with the used keyboard shortcuts. For instance, the TAB function does not work properly. Normally, when pressing the TAB button, the next editable text field is selected. However, when pressing TAB in Ciboodle, it is not quite clear which field is selected because there is no visible color change of that particular field. The only thing that changes is the border of that field; it becomes dotted instead of a continuous line. It would be better to highlight such a field using a distinctive color, thereby making the current field more visible to the user. When you move the mouse over a button, the buttons' color changes from blue to red. This is a very clear and distinctive method for displaying which button/field is selected. Another issue that occurs at some points in the menu, is that double keyboard shortcuts are used. It is unclear which action will result when pressing ctrl + b (figure 6.)

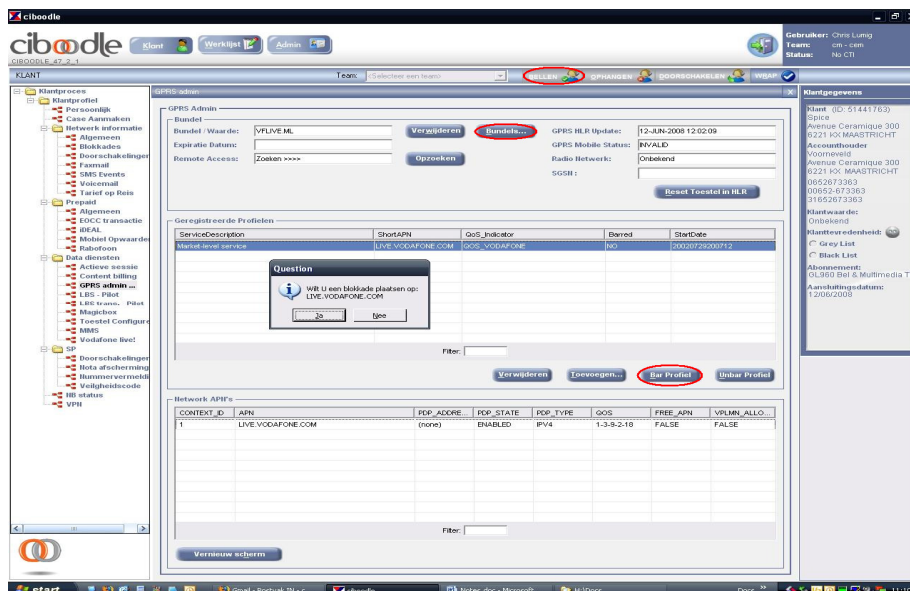


Figure 6. The same keyboard shortcut CTRL+B is used 3 times for different functions.

A software program should prevent users from having to enter data several times. The shortcut buttons enable users to quickly select a case topic without having to use the logging trees every time. Also, the customers' telephone number can be copied from the account bar on the right side of the window. This number sometimes has to be entered several times, so it is better to use the copy-paste option. However, not all users might be aware of this functionality, because there is no help menu that informs them about these workarounds.

Lastly, the used labels in Ciboodle are clear and distinctive. They are presented on the (top-) left of the information fields, which is a very common place.

Data display

One of the most important usability aspects of a software package is the presentation of information. The used language and word phrasing should be consistent with its' intended users (second design principle, Nielsen et al. (1990)). For users to be effective at performing their task, the information that is displayed must be relevant to the users' needs. Basically, this means that you should not display more information than the user needs for accomplishing his task. Another important aspect is that the information displays need to be flexible so they can be tailored online to meet the current needs. In Ciboodle, this could be accomplished by letting the agents choose their own shortcut buttons.

The screen layout of Ciboodle is very simple, yet straightforward and consistent (Nielsen's fourth design principle). The left side presents the user with different menu's, and the right side shows customer details. The middle of the screen is used for the presentation of information and for the actual case logging. Agents can describe the conversation with the customer in the description field, and they can select the appropriate case topic. Because the logging trees are displayed next to each other, the user can see which choices he made so far. These findings confirm Nielsen's sixth and eighth design principles: recognition is easier than recall, and use a minimalistic design. For example, the customers' information is always visible, so the user does not have to remember specific details. There are also some issues with the screen layout. For instance, the column on the left side is a bit too small to be able to read the menu descriptions at once. A user has to scroll sideways in order to see the full description. On the right side of the screen, 3 versions of the customers' telephone number (CTN) are displayed, but without a label. One should never assume that a user recognizes such data when presented without label. Three different versions of the CTN are presented because the different programs that are used by Vodafone agents use a different representation of this number. Gemini, for instance, adds an extra zero in front of the standard 10 digit telephone number.

When a customer has more questions, each question should be logged separately. This would however mean that a user has to close the current account, reopen it, and make a second logging. This takes too much time, especially because the customer is still on the phone while making the logging. There should be an option that enables the user to quickly create a second logging. Also, when the customer has a question about a topic which cannot be found in the logging tree, the user should be able to report this. If more customers start to ask related questions, new topics have to be added to the current logging tree. One should however try to avoid clutter by making lots of case topics. From

time to time, you should determine whether some topics might have become irrelevant, and therefore should be removed from the logging tree. By regularly updating the logging trees you avoid the possibility of users selecting irrelevant topics. As said before, users can select frequent occurring topics from the 16 buttons that are presented on the top of the screen. The buttons are grouped in alphabetical order, which makes it very easy to find one particular button.

The user should always know which data field is currently selected. Ciboodle displays a small dotted line around such a field. This is a good representation of a selected field. This selection can be done using the mouse pointer, or by using keyboard shortcuts. As Ciboodle runs on a Windows© Platform, you should try to use the same shortcuts as users would do in any other program on that same platform. As mentioned before, the TAB function usually causes for the cursor to jump to the next (editable) field. When the TAB key is pressed in Ciboodle, another field is selected, however, the order of the jumping between fields is not logical (not from left to right, top to bottom).

Sequence control

Users usually execute tasks in several steps. This way the tasks become a bit easier to solve. For this reason, it is also important that a user can return to a previous point in order to make changes. This is what sequence control is about. A system should permit the user to step forward, but also backward during a decision-making. If the interface only permits the user to make steps forward, this system is inefficient (Mosier et al, 1986). Ciboodle uses such sequence control. It is easier for the user because he does not have to remember previous made answers, since they can always be looked up. Also, when creating a case, the user can see which choices he previously made in the logging trees, and he can gather information from another menu of Ciboodle, and then return to the initial case. During this choice making, irrelevant options are discarded from the logging trees. The user is thus in control over the system, which confirms Nielsen's' third design principle.

Because the user always has to select a topic before he can close the case, it might happen that the user just selects any topic, which is irrelevant to the customers' question, but he is forced to do so. This kind of scenarios should be prevented, because it leads to inaccurate loggings and frustrated users. Nielsen's fifth design principle (design the system in such a way that error prevention is taken well care of) is not fulfilled because of this.

When two users try to logon to the same account, they do not receive a warning that the account is already opened. However, there is also no need for that because each user can create its own case logging, independent of the other user.

Ciboodle does involve a lot of mouse movement. During the 'wrap up' of a case (basically the same as saving and closing the case) the user has to move the mouse across the screen several times. For this kind of menu selection you should instead consider using selection by keyed entry.

A user should only be presented with options that are relevant for performing those actions that are necessary at that moment. In Ciboodle, users can have different accounts with different rights. The team coach can send a request to one of the administrators of Ciboodle and ask them to change the accounts of his agents. Note however, that only the shortcut buttons can be changed. Menu structures are the same for

every user account. These menus are presented with descriptive labels, which makes it easy for the user to distinguish between them. They are ordered according to their importance level, that is, the most frequently needed menus are presented at the top. At any time, all the menus are visible on the left side of the main window in Ciboodle.

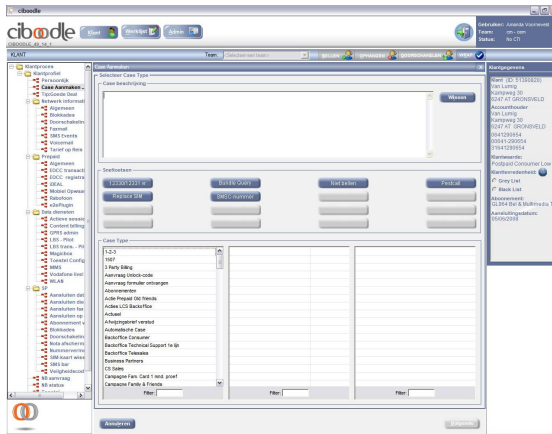


Figure 7. The menus are presented on the left side of the screen in Ciboodle.

Although the used menus and menu structures (figure 7) are very clear, in general purpose systems whose use is varied, it is desirable to permit users to tailor menu design to their needs. Some menus are irrelevant to certain users, and making it possible to completely hide such menus would improve the interface because only needed information is visible. Also, because of the varying use of Ciboodle, it is wise to provide users with a general list of the available commands and shortcuts.

User guidance

A user should always be able to retrieve the system status. When the system is currently processing a task, the user should be aware of this, so he does not start another process. While Ciboodle is busy processing a task, an hourglass is displayed when the mouse pointer is placed above the menu bar. However, when you move the mouse to another location, you do not see this hourglass anymore, and thus you will not know whether the system is idle or busy. This becomes a problem when the user selects certain menu options that gather live information from the mobile network (eg. whether the mobile phone is turned on). It takes a little while to retrieve this information. If the user is unaware of this delay, he might think that he did not select the option properly, and tries to select it again. This can lead to a system crash. It is thus advisable to show the system status at any time, regardless of the currently selected screen or mouse pointer position.

Another important aspect you should keep in mind while designing a software interface is the fact that the human memory is unreliable (Mosier et al, 1986). User guidance is necessary to avoid errors. Creating a case logging in Ciboodle is quite straightforward. Previous choices can be reviewed and changed. When stepping through the logging process, the user gets instructions on how to proceed. However, users sometimes reach a point at which they do not have the knowledge of how to proceed. Normally, users can turn to the help menu in order to be pointed in the right direction. Ciboodle does not have such a help option, thereby neglecting to fulfill Nielsen's tenth

design principle. Instead, agents are advised to look at the intranet site Coach. Coach contains all kinds of topics, such as currently active promotions, network maintenance, and also a list of FAQ's. What you can also find on Coach is a step-by-step description of how to create different cases. Users can follow these steps using the images that are displayed on this same page. This helps users during the logging process, and makes their job easier. However, they still have to switch between Ciboodle and Coach, which is somewhat redundant.

Data transmission

As mentioned in the previous paragraph, users have to switch between Ciboodle and Coach in order to be able to create case loggings. However, the more an agent gets used to his work, the less he needs to use Coach because he now knows how to do the job properly.

This takes us to another issue. Vodafone agents use two main software programs: Ciboodle and Gemini. Since different companies developed both programs, they look nothing like the same. Mosier et al. (1986) states that there is a serious disadvantage in trying to combine separately designed software packages. Users can get confused about the way information needs to be inserted. Opening an account in Ciboodle happens via a different method than in Gemini. In Ciboodle, you can just enter the customer's telephone number in order to open his account. In Gemini, you have to insert an extra zero (0) in front of the regular telephone number. This makes it harder to copy data from one program to the other because a different representation of basically the same data is used.

Although both programs use a somewhat different logging method, they both offer the user to create a message in which they can summarize the conversation they had with a Vodafone customer. This way, other agents can check whether the customer has already contacted the call center before about the question or issue at hand. Before these loggings are stored into the database, the user has to save them. Exiting the program without saving will lead to a loss of all inserted data.

Data protection

As a designer you should always try to provide automatic measures in order to minimize data loss. This can for instance be accomplished by making an auto save function; after a certain time interval, data is automatically saved. In case the system crashes, this data can then be recovered. During the evaluation it turned out that selecting certain menu options sometimes caused the system to crash. Unfortunately, Ciboodle does not have an auto save function yet, which means that if the user inserted information, this information would be lost. This dissatisfies Nielsen's ninth design principle, which states that the system should help a user recover from errors.

In order to protect a user account from misconduct, each account is locked with a password. After three times falsely entering this password, the account is locked. A user can set his own password in the administration menu of Ciboodle. These security measures are thus needed to protect both users and Vodafone.

Gemini evaluation

This section will describe the evaluation of Gemini. Again, mind that all design principles and rules used in this results section are copied from Mosier et al. (1984) and, were explicitly noted, from Nielsen et al. (1990).

Data entry

Depending on the question of a customer, the Vodafone agent creates a logging of the conversation, either in Ciboodle or in Gemini. In the previous section we described how this logging takes place in Ciboodle. We will now explain how this process works in Gemini.

During a conversation with a Vodafone customer, the agent opens the customers' account. Each account has its own number which is called the **billing account number**, or BAN. When the appropriate BAN is opened, the agent can create a memo that summarizes what the conversation was about, and what actions the agent has performed or will be performing. These memos are stored inside the customers' account so that other agents can view them at a later moment. To create a memo, the user first has to select the memo type. This type represents the topic of the conversation, and can be inserted manually or selected from a very long list of memo types. When entering a wrong memo type, Gemini returns an error, stating that a non-existent code was entered.

If you want to select a memo type from the list, you always have to scroll sideways in order to read the full memo description because the window is too small. For users it can be difficult to memorize codes because usually there is no representational link between the code and its' underlying function (Nielsen's second design principle). However, the memo codes used in Gemini do have some meaning. The first character indicates whether the memo is about a request, a complaint, or a question. The second character indicates the category level. The last two characters indicate the more specific description of the memo in question.

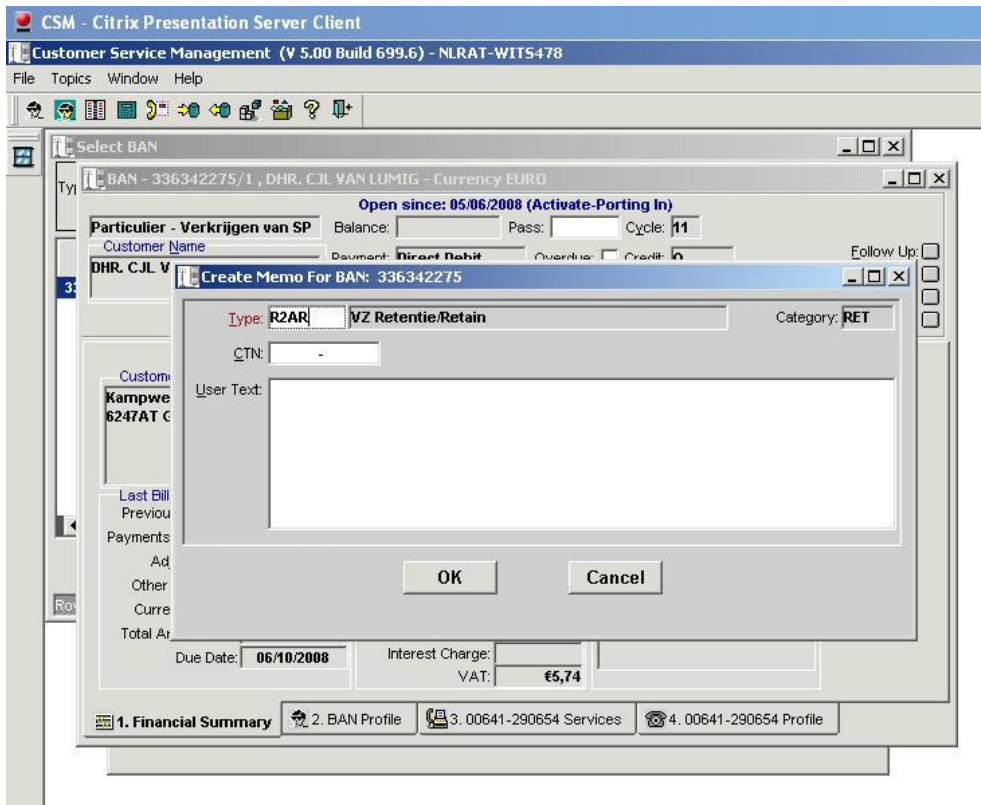


Figure 8. The memo field is used to summarize the conversation with a Vodafone customer.

In figure 8 you can see the ‘type’ field in the upper left corner of the memo window. After selecting the memo type, the user has to enter the appropriate customer telephone number (CTN). Selecting the right number is necessary because one customer can have more telephone numbers. A list of telephone numbers that are currently connected to the BAN can be acquired by double-clicking the ‘CTN’ field. Next, the user can enter a description in the ‘user text’ field. It is very common that abbreviations are used in this text field. Especially for novice users it can take some time to get used to these abbreviations, because there are no guidelines on how to use them. After filling out the text field, the user can close the memo.

The used representation for (non-) editable fields is handled properly. Editable fields have a white background, as opposed to the grey background used for non-editable fields. One particular field uses a red color to display the by a customer created password, which makes it easy to detect (this password is used during the verification process). Also, editable fields are highlighted in a blue color when they are selected. However, the problem is that information inside non-editable fields can be copied, but not selected. This copying is done via a popup menu that appears when clicking the right mouse button. Information in these non-editable fields can for instance be an address, or a telephone number. This kind of information is many times used in the memo text field. If a user is unaware of the copy option, he has to manually insert this information every time.

In figure 9 you can see that there is no space between the text inside a field and the border that surrounds it. It is better to keep the text separated from the border by at least one space. Also the outlining of these fields differs from one window to the next.

The same applies for the presentation of labels; sometimes labels are placed further away from the data field than most of the labels. This kind of inconsistency should be avoided.

Customer Address	
Avenue Ceramique 300 6221KX MAASTRICHT	
Last Bill Summary	
Previous Balance:	
Payments Received:	
Adjustments:	
Other Activities:	
Current Invoice:	€62,58
Total Amount Due:	€62,58
Due Date:	29/10/2008

Figure 9. No spacing between text and border, and a different outlining in different fields.

Data display

Mosier et al. (1986) states that for effective task performance, displayed data must be relevant to users' needs. To avoid informational overload, the BAN screen is split up into 4 different tabs. Each tab shows a different sort of information. Depending on the selected tab, different menu options become available. As stated in Nielsen's eight design principle, a design should be minimalistic and no more nor less conspicuous than is required. However, the account management screen, which is a special menu, looks exactly like the first BAN screen, although some labels and fields are positioned differently. It also has more menu options available. For users it is very confusing to have two screens that look the same but still have different options (Nielsen's fourth principle). Figure 10 shows both screens.

BAN - 330857582/1, MEVR. PKA VOORNEVELD - Currency EURO

Open since: 06.06/2008 (Restore S-bar opheffen)

Particulier - Verkrijgen van SP
Customer Name: MEVR. PKA VOORNEVELD

Payment: Direct Debit Overdue: Credit: 0

Team: #BENs: 1 Corporate: BER: Sp. Instruct: Discourt: Marketing ID:

Last Activity: Agentcode CTN gewijz. (2223-EAI API System, 12/11/2008)

Customer Address: Avenue Ceramique 300, 6221KX MAASTRICHT

Current Invoice Details: 06 Mobilie: €0.21, Vast in NL: €0.63

Last Bill Information: Invoice Number: 000100646031, Held Bill: 22/10/2008

Doc. Type: Inv. Freq: Monthly

Bill sent to: MEVR. PKA VOORNEVELD, Avenue Ceramique 300, 6221 KX MAASTRICHT

Last Bill Summary: Previous Balance, Payments Received, Adjustments, Other Activities, Current Invoice: €62,58, Total Amount Due: €62,58, Due Date: 29/10/2008

Recurring Charges: €57,04, One Time Charges, Equipment Charges, Credit Amount: €5,29, Discounts, Interest Charge, VAT: €9,99

1. Financial Summary | 2. BAN Profile | 3. 00652-673363 Services | 4. 00652-673363 Profile

Account Management - 330857582/1, MEVR. PKA VOORNEVELD - Currency EURO

Open since: 06.06/2008 (Restore S-bar opheffen)

Particulier - Verkrijgen van SI Balance

Customer Name: MEVR. PKA VOORNEVELD

Payment: Direct Debit Overdue: Credit: 0

Team: #BENs: 1 Corporate: BER: Sp. Instruct: Discourt:

Change Currency Info:

Last Activity: CTN SOC activity, (104389-Kroese, David, 13/11/2008)

Customer Address: Avenue Ceramique 300, 6221KX MAASTRICHT

Current Invoice Details: 06 Mobilie: €0.21, Vast in NL: €0.63

Last Bill Information: Invoice Number: 000100646031, Held Bill: 22/10/2008

Doc. Type: Inv. Freq: Monthly

Bill sent to: MEVR. PKA VOORNEVELD, Avenue Ceramique 300, 6221 KX MAASTRICHT

Last Bill Summary: Previous Balance, Payments Received, Adjustments, Other Activities, Current Invoice: €62,58, Total Amount Due: €62,58, Due Date: 29/10/2008

Recurring Charges: €57,04, One Time Charges, Equipment Charges, Credit Amount: €5,29, Discounts, Interest Charge, VAT: €9,99

Figure 10. The 1st BAN screen (left) and the account management screen (right) show only minor differences in layout, but have completely different menu options.

Before a user is allowed to give out private information during a call, he has to verify the customer. During this verification process, the user has to collect information from different tabs of the BAN screen. It is better to present this information in one screen. This is both quicker and easier. Also, some information is not directly visible because the used window is too small. Clicking the maximize button (upper right corner) results in a totally unexpected outcome (figure 11).

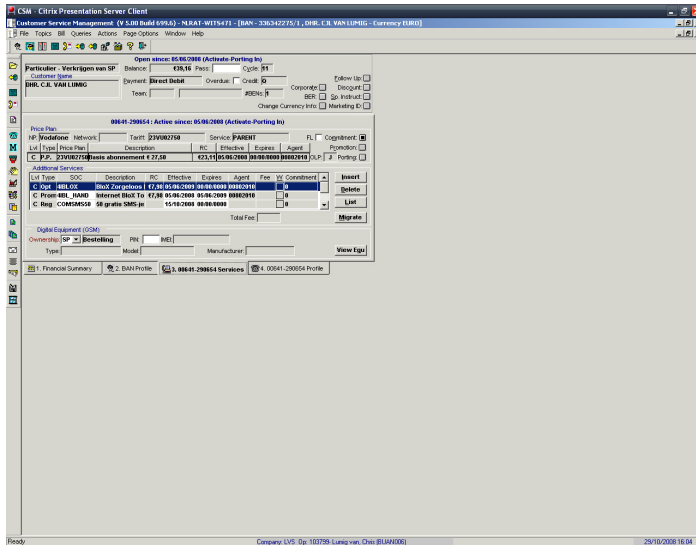


Figure 11. Maximizing the BAN screen does not result in more information becoming visible.

Instead of maximizing the whole BAN screen, only the window border is widened. Especially because many fields require the user to scroll sideways in order to view all information, you would expect that a screen filling view of the BAN would make all information visible.

On the left and top left of the main Gemini window there are several icons displayed that can be used for quickly selecting a function or option. When you move the mouse pointer over these icons, a description of the function appears. This description can also standard be displayed below each icon by selecting this in the menu that appears when you press the right mouse button. As Rogers (1989) stated, it is better to always present icons with labels.

..it is less the iconic representation than the associated use of text labels on icons that aids users in GUIs (Rogers, 1989)..

There is however no option to save this setting (presenting icons with a text label) for the next time you login. Also, in most cases just a single word is used to describe the function, but it is better to give a one-sentence description because some functions are ambiguous and therefore need a more extended explanation. The icons that are visible at a given moment depend on the selected window. If one window has more options than another, the appropriate icons are added to the list of already visible icons. They are however not added to the bottom of the list, but are, without a label, grouped according their functionality. This means that one particular icon does not have a static place on the

menu bar, but can move up and down the bar, depending on the available options for that window. This makes it harder to find one particular function, and thus decreases the learning capabilities. Another issue is that some icons are used on both the left and top menu bar, but they have different functionalities (figure 12). Nielsen's sixth design principle states that it is easier for a user to recognize than to recall something, but because the same icons and shortcuts are used for more function, this is not possible in the current interface.

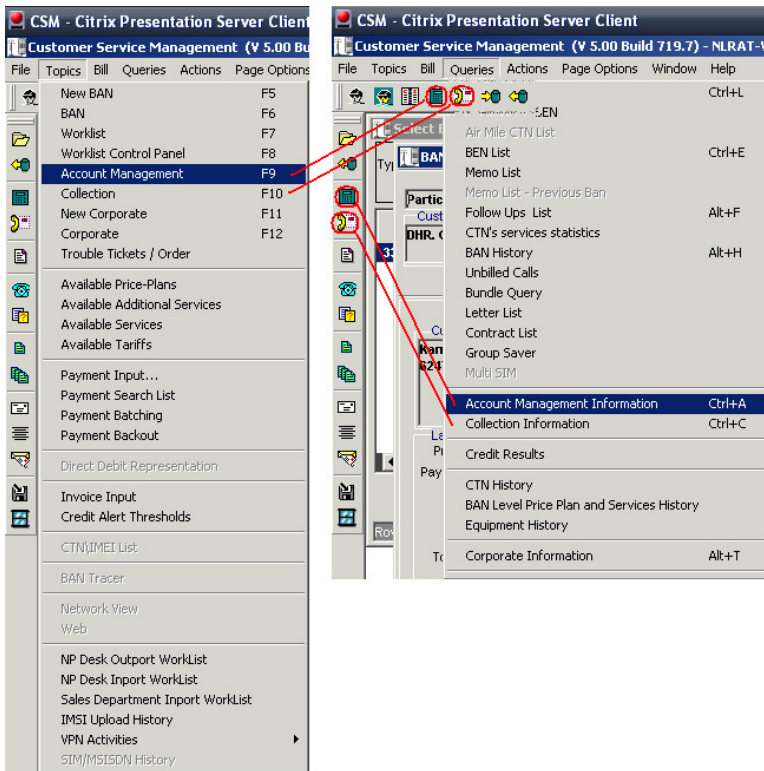


Figure 12. The same icons are used for different functionalities.

As described before, memos are used for logging the conversation with a Vodafone customer. When the user is creating a memo he sometimes has to open the BAN screen to retrieve information such as the customers' telephone number. Returning to the memo field by using the appropriate menu option results in the warning message displayed below (figure 13)

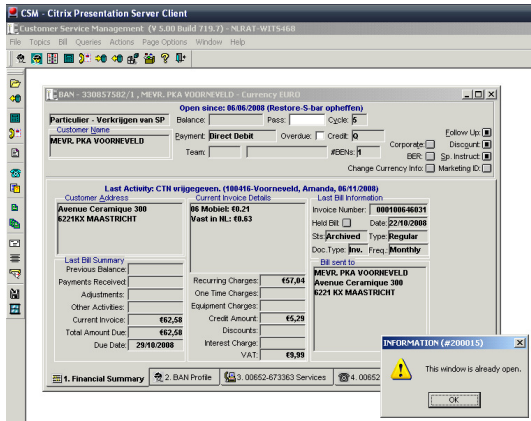


Figure 13. A warning is displayed when trying to create a memo, while the memo window is already opened.

The message warns the user that the memo window is already opened. However, the user cannot see this memo window because the BAN screen is overlapping it. The only way to get back to the memo is by moving the BAN screen aside.

Another problem with the memo window is that the mouse scroll option does not work properly. If the memo is too long to be read in full, the user has to use the scroll bars to scroll down the memo field. When the mouse is used to scroll, the whole memo field disappears (figure 14)

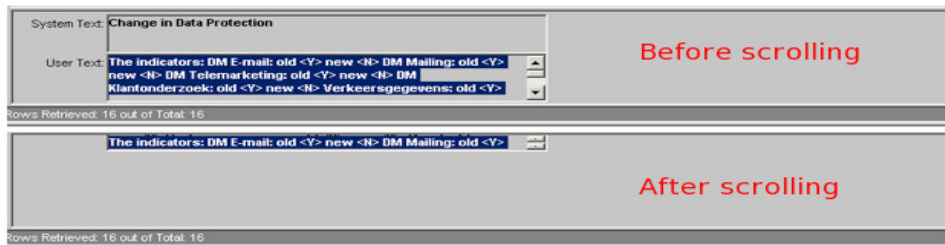


Figure 14. The whole memo field, rather than the data alone, is scrolled up.

Besides the icons and menus, users can also use many keyboard shortcuts to perform certain actions. The shortcut is printed behind the according option in the menu. Each time a user opens the menu, he can see that there is also alternative way to select the option. Especially for novice users this is a good way to learn this faster method.

Sequence control

When a user often has to select menu options, you should consider offering keyed selection. This reduces the need for mouse movement, which usually takes longer than pressing a key. Gemini offers lots of keyboard shortcuts. They do however depend on the selected window. This makes it harder for a user to remember, because he does not only have to memorize the shortcut itself, but also in which window it is applicable. Also, you should avoid assigning different functions to generally accepted shortcuts. Normally the shortcut ctrl+c means copying the selected text; in Gemini, the collection information screen is opened when this shortcut is used. Providing a list of all possible shortcuts and commands would be a way of making the problem less occurring and help users memorize them. If an error might occur, the system should hint the user about how to

recover from this error (Nielsen's ninth design principle). When a user tries to select services that cannot be combined, the system gives a warning stating that this combination is not allowed. There are some errors in Gemini that just display an error code, and no further information about why the error occurred, or how it can be prevented. This is an example of bad design (Nielsen's third and fifth design principles), because the user reached a dead end from which he cannot recover. Sometimes, the only way to actually be able to continue working is by restarting the system. A lot of time is involved in this process, and therefore should be avoided by offering better user assistance by the system.

Gemini offers different user accounts with different privileges. These privileged users can perform actions that are normally prohibited, like closing a customers' account, or crediting large amounts of money. Regular users can view the same information as privileged users, but they cannot make changes to the account. There is one button visible ("all rows") in all Gemini accounts, but even privileged users cannot use this function. When a customers' calling history is opened, only 100 rows are displayed at a time. Each time the button "more" is selected, 100 extra rows are added to the currently shown rows. The button "all rows" instantly opens the whole history, thus removing the need to press the "more" button several times. Strange enough, the "all rows" button does work in the training environment of Gemini.

A user should be able to retrieve the system status at any time (Nielsen's first design principle). In the bottom left corner of Gemini it is displayed whether the system is currently performing some task (busy) or whether the user can start a new task (idle). Also, when the system is busy, the mouse pointer changes into an hourglass. These are good indicators that show the system status. Nielsen et al. (1990) does however advise to use a more informative way of presenting the system status, for instance by using a progress bar.

User guidance

The system should give routine feedback, thereby informing the user about its actions and requests. The before mentioned warnings and errors should inform the user about what went wrong, and how the problem can be fixed. Some of the used warnings in Gemini indicate that the user made a false entry (eg. entering a wrong memo code) but others just display an error code. Usually, these error codes are explained in the help menu. In the main window of Gemini, there is a help option visible in the menu bar (figure 15), and there are even sub-categories visible when clicking the help button, but nothing happens after selecting one of the options. This is a major design error, because creating the assumption that there is a help function while there is actually none might affect the users' intention to use the system in the future (Nielsen's tenth design principle). You should always provide good help.

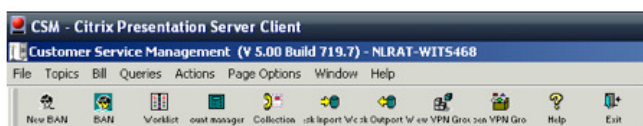


Figure 15. The help menu is disabled.

It may be necessary to test proposed menus to determine which structure seems logical to its intended users (Mosier et al., 1986). As you by now know, Vodafone has several call centers, and they all have different tasks to perform. These tasks require different menu options. At the moment, all Gemini users have the same menu structure. They can make small changes like enlarging particular screens, but these changes only last until the user logs off. It is therefore wise to investigate the possibility to permanently change the menu structure and window layout, based on a users' account. This would add both flexibility and efficiency to the interface (Nielsen's seventh design principle) because each user could tailor its' own layout.

Data transmission

Besides Gemini, a lot of users use the other software package Ciboodle on a daily basis. Also the intranet site Coach is used for data gathering. This means that call center agents have to learn how to use these two completely different looking interfaces. If it is unavoidable to use just one single software program you should at least try to keep the format of frequent occurring information the same. The format for the customers' telephone number differs between Gemini and Ciboodle. Also, only partial information can be copied from a customers' Gemini account to the according Ciboodle account (eg. the date of birth is not copied properly from Gemini to Ciboodle).

Data protection

Gemini does not provide users with an auto save function. Especially when creating a new customer account this could have great impact, because there are many fields that have to be filled with data. During a system crash, all this information would have to be inserted again. However, during the evaluation no system crashes occurred. This is mainly due to the way that information is stored and retrieved. Unlike Ciboodle, which gathers some information from the mobile network, Gemini gathers information from a large database.

Multiple users can open the same account at the same time. Also, they can all make changes (eg. changes the customers' address) in that account. However, only the first user that logged on into the account can actually save the changes that he made. Other users that try to save the changes receive a warning that the account is already being edited, and have to wait for the first user to log out.

As with Ciboodle, users have to enter a username and password before they can log on into Gemini. After three false attempts, the account is locked, and the user has to contact the administrator in order to reset the password. Also, after 30 minutes of system inactivity, the user is automatically logged out. These are all safety measures that prevent misuse of the program.

3.4 Discussion

Ciboodle

In order to get a general overview of the software package Ciboodle, we used a heuristic evaluation. Two checklists were followed step by step in order to determine the systems' usability issues.

First of all, Ciboodle did not satisfy 4 principles of Nielsen's 10 steps guideline for designing user interfaces (1990). The missing visibility of the systems' status (first principle) is a major issue because Ciboodle tends to crash from time to time. Being unable to view the status, a user might be waiting longer than needed. At the moment, only the standard Microsoft Windows hourglass is displayed when the system is busy. However, there is no progress indicator. Also error prevention (fifth principle) is not handled properly. At some points in the program, a user is forced to make a decision. Sometimes, the right answer is not available, thereby forcing the user to choose an irrelevant option. Besides preventing errors from occurring, it is also important to help a user recover from possible errors (ninth principle). As Wickens et al. (2004) states, these issues will affect novice users more than it will affect expert users. During the whole case logging process, there is no option to save the currently entered information. Furthermore, there is no auto-save option. This means that whenever the system crashes, all information will be lost, and has to be inserted again. This negatively affects users because call centers use call handling times as an indication of the users' efficiency (Brown et al., 2002). Lastly, users should be able to acquire help when needed (tenth principle). Ciboodle itself does not contain any help function whatsoever. Agents have to visit an intranet site (Coach) in order to find out how to perform certain actions in Ciboodle. This issue will be less severe for expert users because they have more experience with the task and therefore may find a solution on their own (Wickens et al., 2004).

By using another guideline (Mosier et al., 1986), more usability issues were detected. Some of these issues are related to the use and availability of shortcut buttons and shortcuts (eg. ctrl+c). Ciboodle presents users with 16 of such shortcut buttons, but depending on the user profile, more than half of these buttons are not programmed properly. Also, a lot of irrelevant options are programmed under these buttons in some user accounts. These buttons are a nice way of making the task easier for any user, but especially experts will benefit from them (Nielsen, 1993). However, they should be programmed properly. Besides the buttons, there are several shortcuts available. However, a couple of these shortcuts are used for more functions. This affects the learnability of the system, because users have to be aware of the currently selected window in order to know which shortcut results in a particular outcome. But even in the same window, some shortcuts are used double. This is an example of bad design, and it greatly affects the learnability of the system for novice users (Nielsen, 1993). Ciboodle also requires a lot of mouse movements. At the moment, the buttons for closing a case are placed on the opposite side of the screen. Simply placing them next to each other would solve this issue.

Gemini

The same two checklists as during the evaluation of Ciboodle were used for the evaluation of Gemini. The most important findings will now be discussed.

Where Ciboodle managed to satisfy 6 of Niensens' design guidelines (Nielsen et al., 1990), Gemini only satisfied 2 of these principles; first principle (visibility of system status) and ninth principle (help users recover from errors). Presenting users with codes without a detailed description is a terrible thing to do (Mosier et al., 1986; Nielsen, 1993). Gemini sometimes presents such codes, without informing the user about why the

warning/error occurred. Errors should be clearly defined, such that a user can recognize them, and try to avoid them in the future. When it is inevitable to only present a code, you should point the user to the help menu in which that code can be found, with a proper description (Mosier et al., 1986). The help menu of Gemini is however not working. Especially novice users need help during their first interaction with a new system (Wickens et al., 2004).

Designing a system that is consistent and only presents users with necessary information is a good thing to do (Mosier et al., 1986). Mosier et al. (1986) also states that users get comfortable with the program, and learn how to use it faster when a consistent design is used. These are some points at which Gemini needs to be improved. During the verification process that takes place at the beginning of each call, the agent has to collect information from several screens. It is better to present this information in one single window, thereby requiring less action from the user. As mentioned earlier, consistency is a good thing. However, Gemini contains some windows that look exactly alike, but different options are available in the menu. Besides that, several icons are used to represent different functionalities. Rogers (1989) states that this should definitely be avoided because it degrades the learnability of the system and confuses users.

The abovementioned findings indicate that the current usability of Ciboodle and Gemini is not perfect. There are some serious issues that need to be looked upon. One thing that can be done at the moment without changing the actual software packages is improving the provided help options. Improving user assistance can easily be done by regularly updating Coach. This way you avoid irrelevant information from being presented on the intranet site, but more importantly, you help users with currently relevant issues.

Based on the findings of the heuristic evaluations we performed in this Chapter, we decided to check whether the found usability issues are not only of a theoretical nature, but also whether they affect the actual users (Vodafone agents). This was done by using unstructured observations (Wixon et al., 1990), during which agents were observed while performing their normal work. These observations will be discussed in Chapter 4.

4 Unstructured observations

4.1 Introduction

The previous Chapter described the process of investigating the usability of a software package, using general checklists that have proven to be useful for this purpose. In order to get more specific information about the usability of the software packages that are used by Vodafone agents, another method called unstructured observation was used. This method involves unobtrusively observing participants while they perform actions in their natural work environment (Wixon et al., 1990).

We chose this method because it enabled us to get a quick insight in the software packages, whilst also getting to know the work environment and work conditions of the users. Together with several team coaches, three departments were selected that made use of the initially to be investigated software package, Gemini. This selection had to be made because Vodafone has too many departments to incorporate in this research, and the choice was based on the diversity with which the software was used by each particular department.

As mentioned before, the departments that were selected for the unstructured observations are Postpaid Consumer, Prepaid, and Distribution. The remainder of this Chapter will describe the performed observations in detail.

4.2 Method

Participants

After the departments were selected, a meeting was arranged with the team coaches of the particular departments to choose agents that we could observe. Since this study mainly focuses on the difference between novice and expert users, 10 agents per department were selected, based on their level of experience and the amount of hours they worked per week. 5 of these agents had just started working in the call center. The other 5 had already worked for over a year at Vodafone. In total, 30 agents were observed.

Material

Initially we intended to evaluate only the software package Gemini. However, after having observed a couple of agents, it turned out that there is another software package that is used interchangeably with Gemini. This program is called Ciboodle. It serves, among other things, as a portal to live network information. Since both programs were used in parallel, my Vodafone supervisor Tom van Buitenen and I decided to evaluate both Ciboodle and Gemini.



During the observations, a Plantronics headset was used. This headset gave us the advantage of being able to eavesdrop the conversation that agents were having with Vodafone customers, whilst not interfering the conversation itself. Figure 16 displays such a headset.

Figure 16. Plantronics headset, used by Vodafone agents.

Procedure

Together with the team coach we set a time and date for the observations to take place in the agents' natural working environment. At the beginning of each observation I briefly introduced myself, and explained the purpose of this research. After that, I sat next to the agent and observed him while he continued working.

Using my headset I could follow the conversation agents were having with Vodafone customers. Also, I could observe the interaction between the agent and the software packages. I was allowed to ask questions whenever something was not clear to me, and I made notes during each conversation. This process was repeated until all 30 agents were observed.

4.3 Results

During the observations, agents made many remarks about the usability of both software packages. These remarks, together with the observed issues, will be reported in this results section. We sorted similar remarks into similar topics and tried to use the same format as the topics used during the heuristic evaluations of Chapter 3. Remarks are presented in the order in which participants mentioned them, meaning that each of the sections below contain both remarks about Ciboodle, as well as Gemini. We will explicitly indicate whether the remark is about Ciboodle or Gemini.

Data entry

Redundant actions

Before a Vodafone call center agent can assist a customer, the customer has to be verified as being the owner of the telephone number regarding the question. This happens via a verification process that begins right after the conversation is started. As soon as a connection is established, a system called IVR (Interactive Voice Response) displays the customers' CTN (Customer Telephone Number) in a popup window in Ciboodle, and the customers' account automatically gets opened in Ciboodle. However, on some occasions it happens that the IVR cannot retrieve the CTN properly and the customer is asked to manually insert his telephone number, using his telephones' keypad. This number is then

visible in the same popup in Ciboodle, but since the number was not recognized automatically, the agent has to manually insert the telephone number. As noted by a novice agent, the problem is that you cannot copy-paste the number from the popup window into the search field and manually open the account from the database. This means that the agent has to remember the 10-digit number. As it happens, agents sometimes forget the number, and have to ask the customer again for their telephone number. This frustrates both agents and Vodafone customers.

A similar problem exists when opening an account in Gemini. Agents have to insert the 10-digit telephone number of the customer, but also have to add an extra 0 at the beginning of that number (eg. if the telephone number is 0612345678, an agent would have to insert 00612345678). Fortunately, Ciboodle presents users with a list of three versions of the customers' number, one of which has the extra zero added at the beginning. This way, agents can easily copy the number from Ciboodle to Gemini. All in all, it takes about 12 steps before an agent can assist the customer.

After the customers' account is opened, the agent can start answering questions. Using both Ciboodle and Gemini, information is collected from several tabs and windows. The agent fills out a small description field which summarizes the questions and requests of the customer. This information is then saved in a logging. Gemini uses a semi-automated logging (eg. activating a particular service automatically generates a memo that is saved in a list of memos). In Ciboodle, the agent has to fill in each field manually.

When the conversation is ended, the agent has to close the customers' account. However, when the agent wants to close the customers' account in Ciboodle, he first has to indicate how satisfied the customer was with the help the agent offered. This is done via a satisfaction smiley, which is displayed below (figure 21).

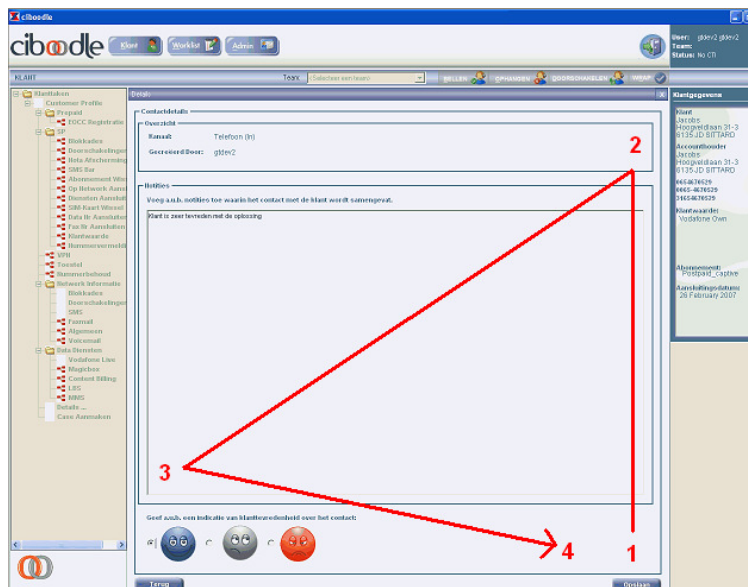


Figure 21. Actions necessary to save and close a case logging.

However, agents are not keen on using the smiley. In order to indicate how satisfied the customer was, you have to click the small selection box on the left side of the smiley.

This is in contrast to what agents would naturally expect. Several agents namely reported that it is more logical when you could click the picture, rather than the tiny selection box.

As described above, a lot of complaints that are made during the observations relate to the verification process that has to be completed before personal information may be shared with the customer. On average, agents have to perform 12 steps before they can assist the customer. Also, on-screen buttons are located in such a way that during every conversation, an agent has to move the mouse across the whole screen at least 4 times. Obviously, you should be careful when handing out such information, but placing the necessary information in a single window would simplify the process, and minimize the workload for the agent.

One other thing that occasionally happens is that a colleague agent tries to hand over work to the agent, while that colleague should actually do the work himself. It seems that novice agents are more prone to agree to such requests, probably because they have not yet figured out that the work should actually be done by the other person. Other actions could be performed automatically rather than having the agent do it. For example, replacing a SIM card involves entering a number of 15 digits of which the first 10 are always the same. This should be turned into an automated process.

When a customer does not pay his bill on time, his telephone number will be disconnected after a while. To get reconnected again, the customer has to pay, and then contact Vodafone. The problem here is that normally, when a customer contacts Vodafone, he automatically gets transferred to the right department. However, since the number is disconnected, he has to use another telephone, and then usually gets connected to the wrong department. Thus, the customer has to be forwarded to the right department. This does however involve unnecessary work for the first agent. Using a menu option from which the customer can select the right department would avoid this problem.

One last remark about redundant actions relates to filling out questionnaires. Some agents, that never have direct contact with customers still have to fill out a questionnaire that is about the customer contact. This problem is caused by the user profiles in Ciboodle which make use of a standard layout. Both novice as expert agents note to find it irritating to perform these irrelevant actions.

Text

Because agents try to keep conversations as short as possible, they use abbreviations to create memos. These memos consist of a summary of the questions that were asked by customers, and the actions performed or information provided by agents. In the beginning, some novice agents did not understand what some abbreviations meant. This happened because there is no list of used abbreviations; it is whatever a particular agent thinks is a logical and common choice. Also, the topic list in the memo field is very long, and it is sometimes hard to choose between several options.

Some problems cannot be solved directly by an agent and have to be forwarded to technical support. Before an agent can forward such a case, he has to fill in a questionnaire that provides technical support with sufficient information (eg. make and model of the customers' mobile phone) to handle the case. This causes for a problem: agents have to minimize their call time, so they like to work quickly. It therefore happens that these questionnaires are not filled in properly. On the other hand, for technical

support to perform their work properly, they need as much information as possible. This is a conflict of interest and should be look upon very seriously.

Data display

Menu structure

A couple of agents indicated that the verification process, which takes place at the beginning of each conversation in order to establish the customers identity, takes too long in Gemini. Not all the required information can be found in one screen, and thus the agent has to switch between screens. It is better to present this necessary information in one screen.

Another remark is about the creation of memos. It oftentimes happens that an agent has to create similar memos. At the moment, this had to be done manually each time. Several agents indicated that it would be desirable to be able to select certain memos from a shortcut menu.

A similar shortcut menu is already integrated in Ciboodle. However, here agents sometimes experience informational overload due to the large amount of possible conversation topics (Yang, 2003). Non-relevant topics (eg. expired promotions) should be regularly removed from this list in order to avoid clutter.

Popups

After opening a customer account in Gemini, a popup appears on the screen presenting the user with information about future requests (eg. cancel certain services). However, nearly all agents simply close the popup without actually reading it. Some of them seemed to do this unconsciously. After asking agents about this popup, it turned out that the information presented can always be retrieved through the menus. One agent remarked that she only found out about the meaning of the popup after 1,5 years when she was working at another department. Therefore, there does not seem to be a need to remember this information at the time. Besides that, an agent has to verify the customer before he is allowed to give out any information. The popup is thus redundant at this point.

Window size

Several agents reported to find the size of the memo screen in Gemini too small. In order to read the whole memo, the agent has to scroll down using the scrollbars. However, these scrollbars only become visible when you click in the memo field. Therefore the agent might be missing out information without knowing it. In figure 17 the system text field contains more than 3 lines of information, but this does not show initially. As you can see below, there is plenty of screen space available to enlarge the memo window.

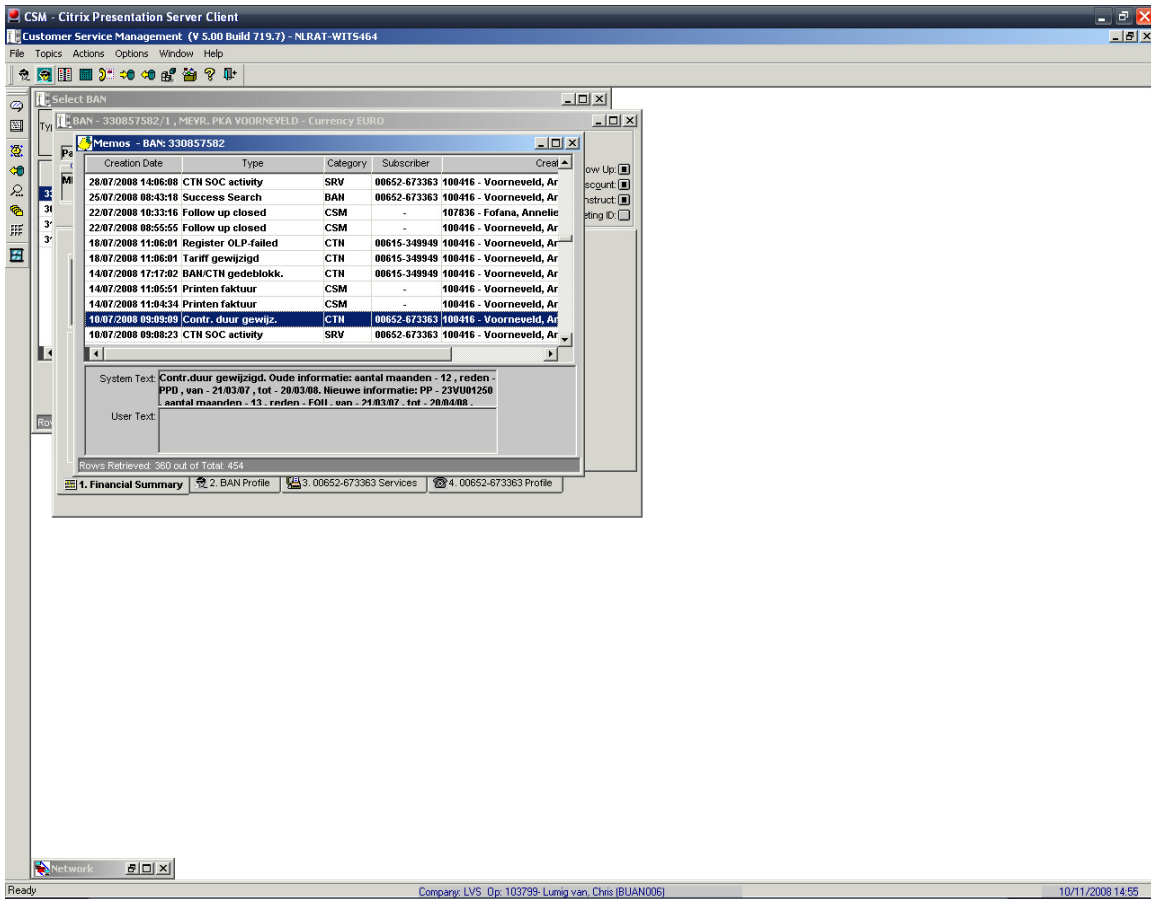


Figure 17. Memo screen in Gemini.

Another situation in which the used screen size is too small is when the agent wants to perform a letter request (figure 18). Such a request is normally made when a customer asks for a confirmation of enabling/disabling a service. However, since there are many sorts of letters, and since the description window is too small, agents indicate that they sometimes send the wrong letter to a customer. Also, the letters for requesting or canceling a particular service are presented in consecutive order. It is therefore easy to make a mistake.

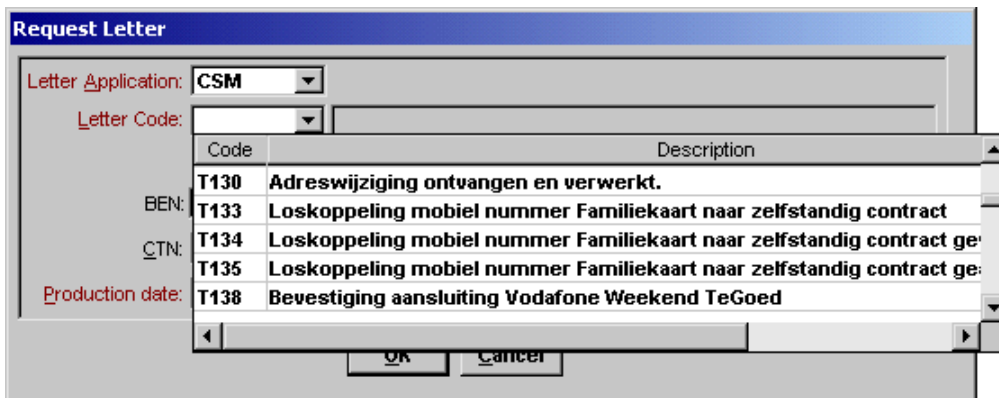


Figure 18. Letter request window. At the request of a customer the agent can send a letter regarding a particular topic.

The used letter codes have proven to be useful because, as agents get more similar letter requests, they start to memorize letter codes and can work more quickly.

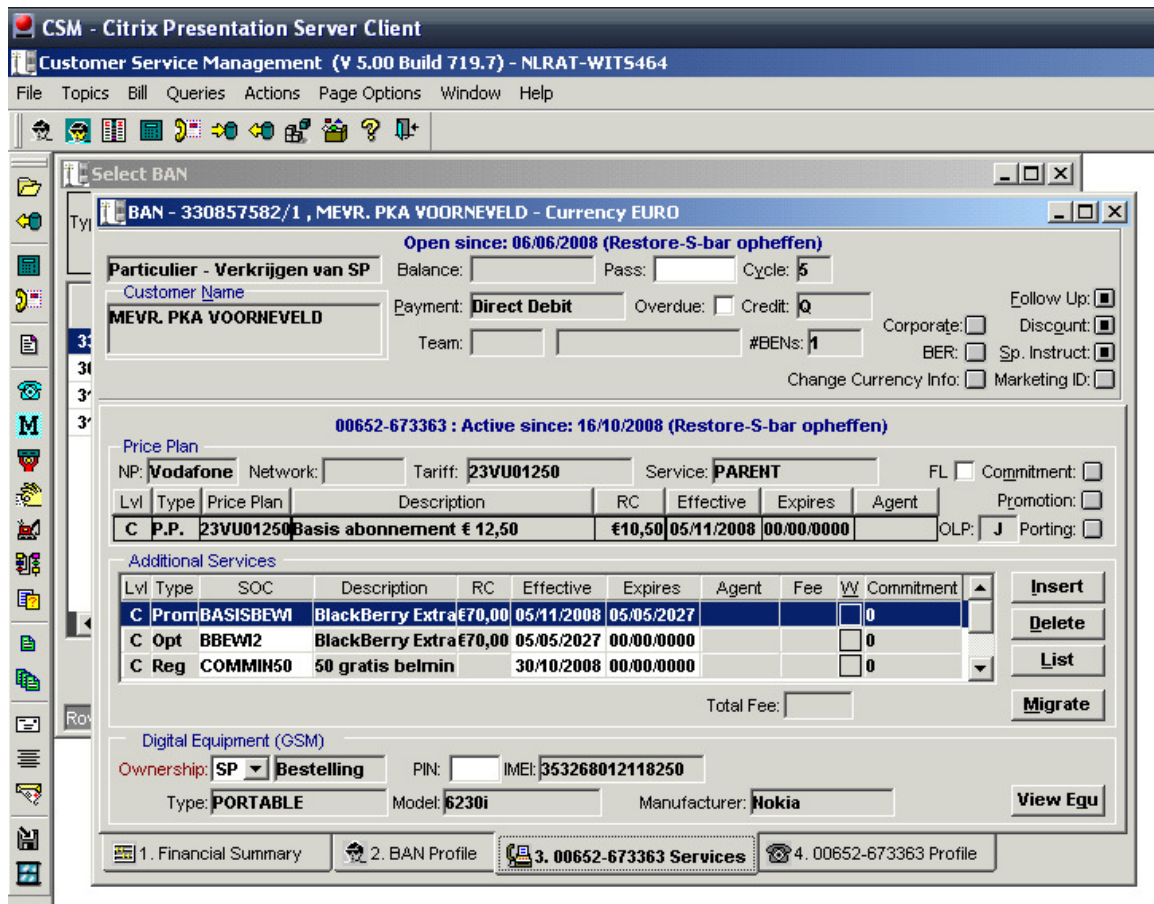


Figure 19. BAN screen. This screen contains various sorts of information like payments, price plan, and additional services.

The additional services description field (figure 19) is too small to read in full. Agents widen the text field to see which particular SOC (Service Option Customer) is activated. Also, when more SOC's are activated, the agent has to scroll down. As could be seen in figure 17, the BAN screen only covers about one fourth of the screen. There is plenty of space to make the whole BAN screen larger, thereby avoiding these unnecessary user actions.

The abovementioned problem is especially relevant for novice users because they do not remember the SOC code yet and therefore have to use the mouse every time to scroll sideways. As agents become expert users over time, they start to recognize the description codes and do not have to use the mouse anymore.

Layout

During the verification process the agent has to collect information from 2 different screens. The layout of these tab screens should be changed, and all required information should be placed in one screen. This saves time and requires less user action.

One thing we noted during the observation is that a couple of expert agents opened 3 different screens (BAN, account manager, pending calls) in Gemini and tiled them over the screen. This enabled them to retrieve the most relevant information in just one glance across the screen. Other agents had a bit more trouble locating the information the customer asked for. Some screens even do not have the possibility to be resized properly; maximizing the particular screen results in a big grey area surrounding the initial text area (figure 22).

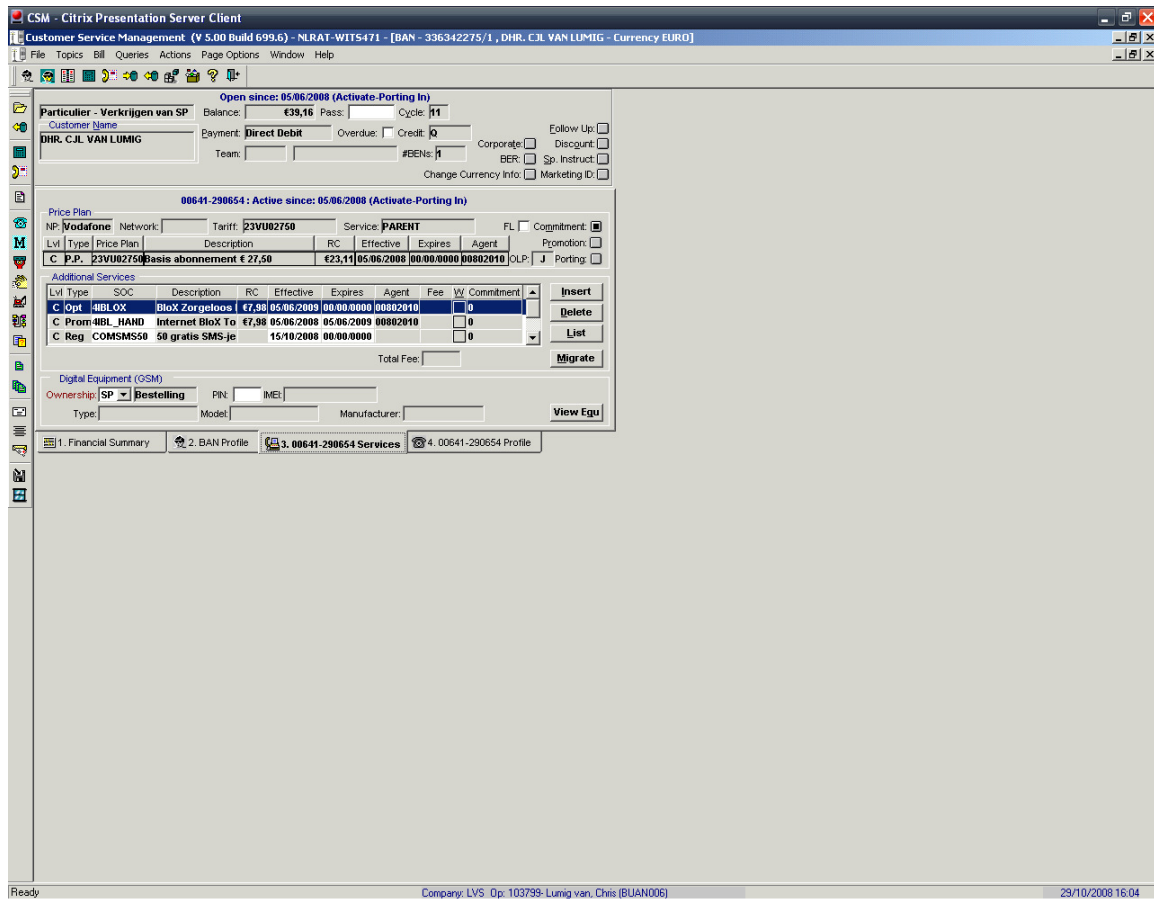


Figure 22. BAN screen maximised.

Visibility

When an agent tries to open a window twice, a warning is displayed (figure 24) indicating that the window is already open. This can be a bit tricky because when a larger window is placed over a smaller one, the agent is not able to see this; the window does not jump to the front. Minimizing or closing a window is the only option.

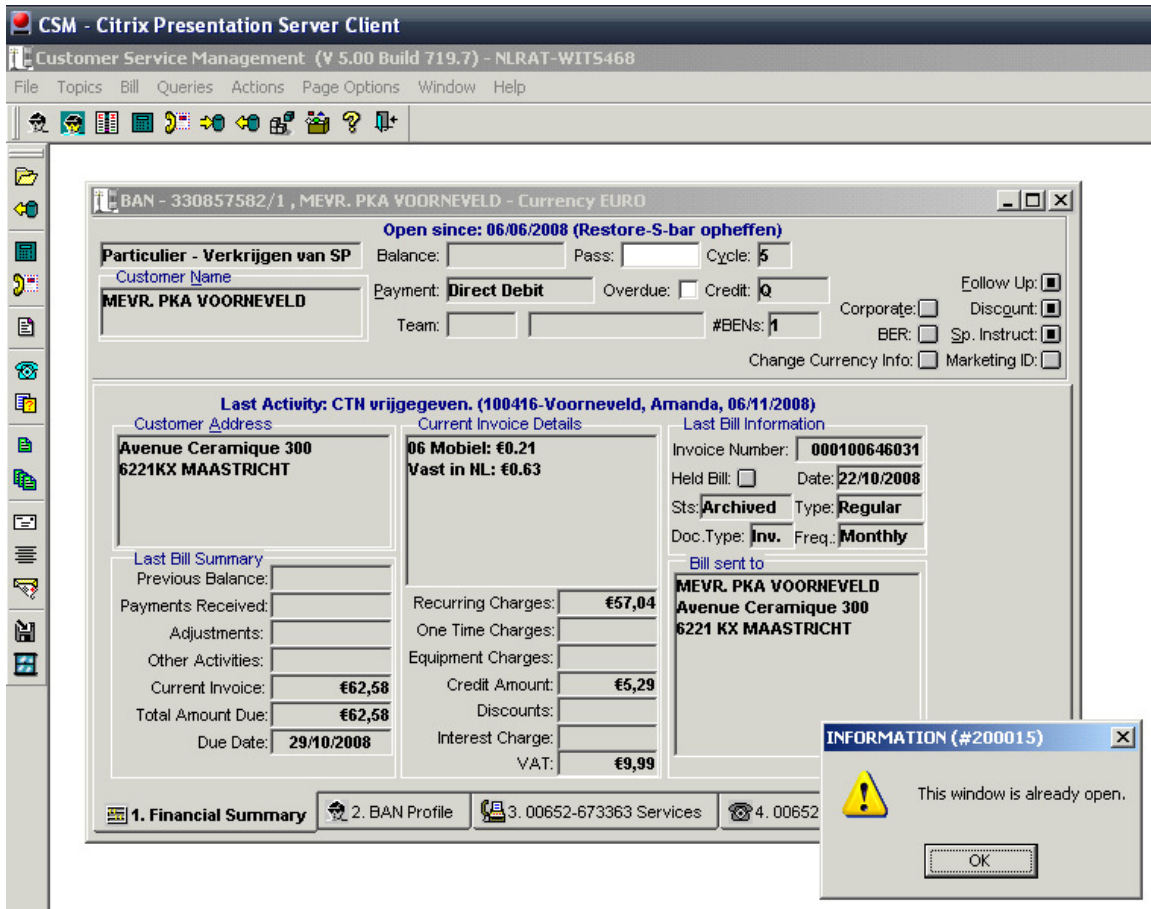


Figure 24. Re-opening the memo window.

Billing

Many customer questions are billing related. They want to know how costs are calculated or why they have to pay for a particular service. Other questions are about the recurring charges: a customer has to pay usage costs of the previous month and subscription costs for the next month. This is sometimes found confusing because, for example, the bill of August shows usage costs, even when that month has just started. Also the used double sided billing format often leads to questions. Customers think parts of their bill is missing, while the bills are actually complete, however half of the information is printed on the backside of a page. Customers often do not note this. Other billing issues like a delayed bundle query update (ie. a list of usage costs) were mainly caused by problems during a Gemini update and were fixed around September 2008.

Sequence control

Buttons

Customers often acquire information about their calling history. In order to get a complete list of the billed calls an agent has to press a button which is labeled 'more'. Every time the button is pressed only 100 rows are added to the already visible rows. In the 'train' environment of Gemini there was a special button assigned to this task (ie. pressing the 'all rows' button would show the entire history of a customer). The

remarkable thing is that this button is visible in the 'live' environment of Gemini, but it is also grayed out, which means that it cannot be used.

In Ciboodle there are also several buttons that do not work. This is caused by the fact that Ciboodle uses two types of profiles: user accounts and button profiles. Each button profile is linked to a specific skill. Each department has several button profiles (ie. skills) and each user account is linked to one of these profiles. It is possible that a user account is linked to the right department, but still has the wrong skill. This agent then has options available that will not result in an action because the underlying link is missing. A solution for the abovementioned problems would be to regularly update these button profiles, but also to check whether the agent still has to appropriate skill.

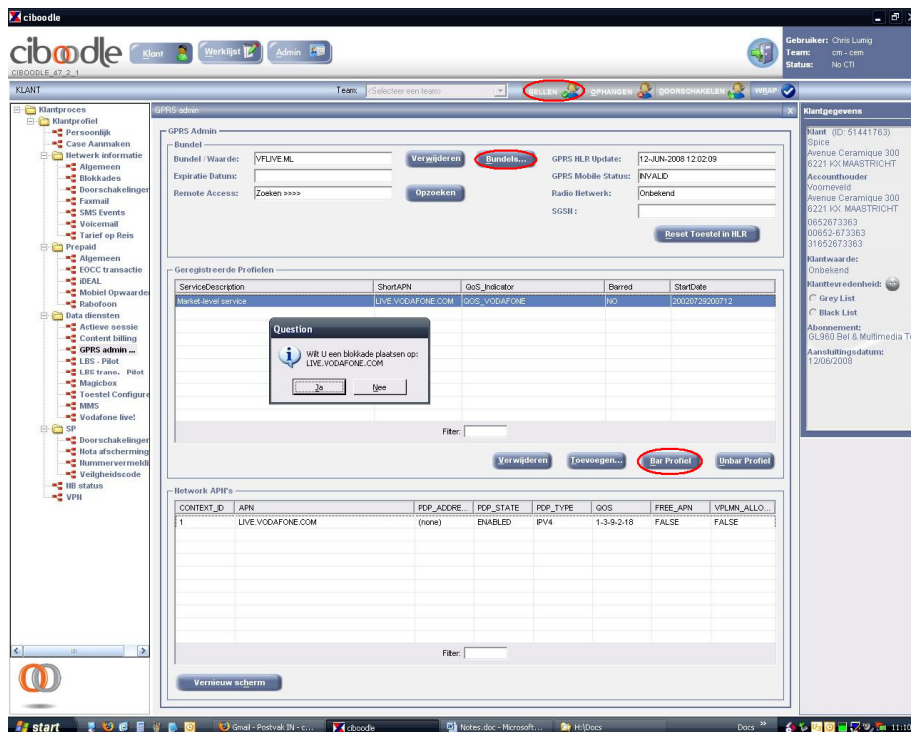


Figure 20. The same keyboard shortcut CTRL+B is used 3 times for different functions

Besides buttons, there are also shortcuts available in both programs. However, none of the agents WE spoke to knew that Ciboodle uses such keys, even though the same presentation style like other Windows© applications is used (e.g. Admin). One major flaw in Ciboodle is that at some points in the menu shortcuts have up to 3 different resulting outcomes (figure 20). It is very unclear which action would result from using that shortcut. Also, the Tab button does not work properly, since you can only switch to other text fields very limited, and you cannot select fields by pressing enter.

Logging

Creating a detailed logging memo costs time. Since agents are judged based on average conversation time, they like to wrap up the conversation as quickly as possible. It therefore happens that the created memos are incomplete, or that a created case consists of more customer questions, while these questions actually should be reported in separate cases.

Procedures

After a customer has signed up for a mobile subscription at Vodafone, he is automatically subscribed for online billing. Many customers are not aware of this and contact the call center requesting their bills.

Another thing customers are not aware of is that some subscriptions prevent them from being able to use particular services.

A corporate client asks with the request to receive 40 pukcodes. Unfortunately, only 5 codes per conversation are allowed, which means that client has to call 8 times in order to receive all codes.

User guidance

Trace

Since Vodafone has so many customers, it would be convenient to be able to track which agent helped which customer. The next time this customer would call, any agent could see who helped this person before, and then could forward the call to that particular agent.

Gemini automatically saves all actions that are performed by any agent, and it is easy to view the history of previous actions by an agent. However, there is no proper tracking of the cases that are created in Ciboodle. Whenever an agent forwards a case to another department, he cannot track the status of this case because there is no logging. This can cause problems whenever a customer calls for an update. Since a trace is missing, the agent has to contact the other department and ask for a status update.

Errors

There are two situations in which the IVR (Interactive Voice Response) opens the wrong customer account: when an employee of a Vodafone service point contacts the call center, the employees' account is opened, rather than the account of the customer visiting the service point. Also, when the customer uses another Vodafone number to contact the call center, the wrong account is opened. Novice agents sometimes did not immediately detect this, and only found out after retrieving more detailed customer information. In the worst case, they did not notice at all, and thus created a logging under the wrong account.

When a customer has a payment that is overdue, Ciboodle automatically places a blockade (also called a 'bar'). It happens occasionally that an agent tries to remove such a bar but that the bar is not removed properly by the system. The result is that the agent has to place the bar, and then remove it again. This system error should be corrected, for example by creating a function that automates the bar/unbar process.

Another error occurs when an agent is working on a case, and simultaneously gets a call. The number of that customer is not displayed in Ciboodle. There is also no alternative way to retrieve this number. This sometimes causes frustration for the customer: when a customer uses a non-Vodafone number to contact the call center, he is asked to type in his Vodafone number, using the keypad. Since the agent cannot see this number (due to an already opened account in Ciboodle), the customer has to repeat the number again.

When an agent, as a result of a request by the customer, sends a bill to this customer, it is not specified in the automatically generated Gemini memo which type of bill was send (which month and whether or not it was a specified bill, one which customers have to pay extra for). Another problem with the bill-menu occurred when a novice agent could not find the right menu option to send a detailed overview of the costs a customer made during the last month. Instead, she made a screenshot of Gemini and mailed it to the customer.

As mentioned earlier, Ciboodle gathers live information from the cell phone network and therefore it sometimes happens that the program locks up, or even crashes. When this happens while the agent was creating a case, all information inserted will be lost. There should be an auto-save function to prevent this kind of data loss.

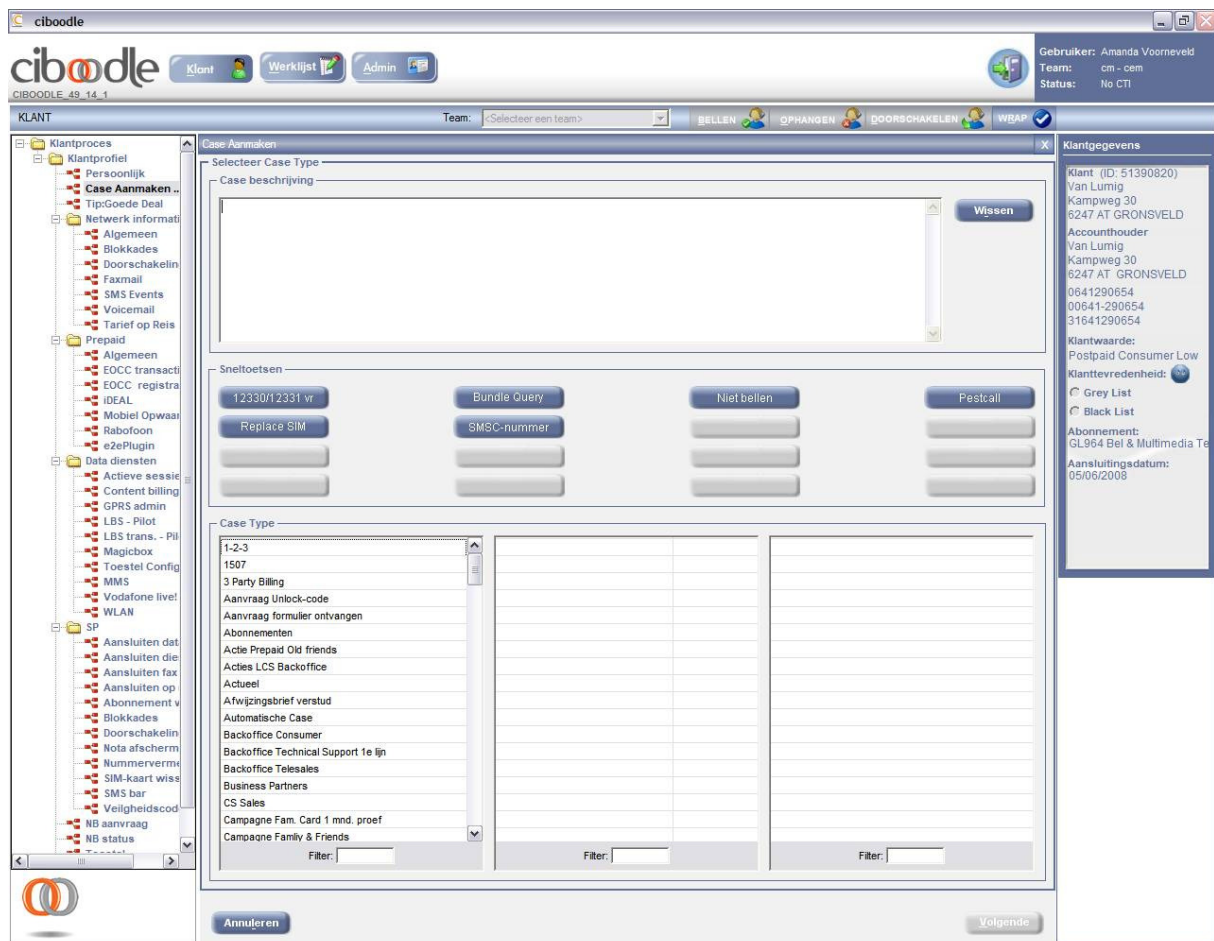


Figure 23. Ciboodle window with short buttons and logging trees.

Case loggings in Ciboodle are created via logging trees and shortcuts (figure 23). There are however some problems. Sometimes a logging tree is displayed wrong; the second tree (out of three) displays information which is unrelated to the first one. Also, some shortcuts lead to non existing options in the logging trees. This last error occurs because Ciboodle uses shortcut profiles and user accounts. Agents of the postpaid department have other options available than, for instance, prepaid agents have. Even if the user

account is set to the right department, the shortcut profile can be wrong, hence show false options.

After creating a case, the agent basically has two options: save and close the case, or forward it. In both situations, the agent cannot retrieve that case if he did not write down the case number. Normally this is not a problem, but sometimes a customer calls back to get a status update. The agent has to contact the other department to get this information.

As opposed to Gemini, where the customers' date of birth is automatically copied from the contract, Ciboodle requires the agent to enter this date of birth manually. Since this information is not always available, the agent does not fill in the particular field and the system inserts the current date as being the date of birth. The result of this is that during the verification process, the agent has to open Gemini in order to get actual date of birth.

Some customers have more than one cell phone number. It is however not possible in Ciboodle to access all these numbers from one account, meaning that if the customer has more questions, each time the current account has to be closed, and the account in question has to be opened. This does however take quite some time.

Examples

After entering only 4 characters (instead of 6) in an address field in Ciboodle, an example is displayed indicating how data should be entered. The user can then easily adjust the information.

Furthermore, Ciboodle has an implemented spell check functionality. This function is however not activated. It is wise to use such spell check in fields where users can freely insert information. This can help correct typos and keep things clear for other users.

Another positive remark made by agents was about the worklist in Ciboodle. This list gives a nice overview of the work that needs to be done, and when a task is due. The more an agent is aware of the available shortcuts, the more he uses it. It is however remarkable how few agents are aware that there actually are shortcuts. Therefore, it is advisable to explicitly point this out during a training session, because it can make the agents' task easier and quicker.

Data transmission

System speed

It is important to keep the user updated about whether the system is still busy processing information (Wood, 2004). Probably for this reason, the biggest irritation during the observation was slow system response/system crashes because these situations keep the user waiting while getting no feedback. Especially Ciboodle crashed many times during the observation period. Sometimes this was because the agent selected a special menu option which gathered live network information, or just because the program in overall had a low response time. Also, it happened occasionally that the agent thought the system crashed, while it actually took longer to retrieve the information. Just before he wanted to restart the program, the information was presented on the display. It might be clear that in these situations it is essential to be able to view the system status.

If the system did indeed crash, the agent had to kill and restart the program. Note that during this time the customer is still waiting on the phone. Novice agents mentioned to feel uncomfortable when this happened because they felt like they kept the customer waiting for a very long time (while it actually only took about 1 minute). Expert agents tried to help the customer during this time, for instance by using another program (like Gemini) to acquire the needed information, or just by talking to the customer, thereby better assisting the customer during the whole conversation.

Especially agents of the prepaid call center have to use menus that sometimes cause Ciboodle to crash. This is because these menus gather live information from the network and therefore sometimes take a little longer to load. Strange enough, these menus are also available in the postpaid profile while these options are not relevant for their work, and still can cause a system to crash when accidentally selected.

Switching between programs

In order to get all the information an agent needs during a call, he often has to switch between both programs, because they each contain different information. However, when one program crashes, the other program can still be used to get some, yet not all information.

Data protection

Illegal issues

Only contract owners are allowed to request specified bills. These bills contain all call history, including the numbers that were dialed. It is therefore important that the verification process is taken seriously. In a couple of cases a non-contract person made a specified bill request. If the agent is not alert and aware of this, it can lead to serious problems. Some agents create a memo describing these requests by non-contract persons, which increases alertness of other agents. Other information (private and business) is sometimes only visible to an agent (eg. KVK number) and this information may not be communicated to a customer because of legal issues (due to confidentiality). Still some customers try to acquire such information.

External issues

Content sms is a service provided by non-Vodafone companies. These companies send wallpapers and ringtones to your mobile phone and charge you for every item. Once a customer has activated the service, they receive several messages per week.

Often times, these customers contact Vodafone, because the costs for these services are charged on Vodafone bills. However, it is not Vodafone, but the external company that can inform customers about their services, and also tell them how to deactivate a subscription. Vodafone agents transfer these customers to the particular company.

4.4 Discussion

This method of observing participants while they were performing their normal work was done in order to check whether the detected usability issues found during the heuristic

evaluation were also present during actual use of the programs. This would then mean that the issues are not only of a theoretical nature, but that they actually have an influence on end-users.

Most of the remarks made by agents confirmed the detected usability issues of the heuristic evaluations (Chapter 3). Obviously, some issues were not detected, but this might be because of the time span during which the observations took place, since not every issue occurs on a daily basis.

Before we start describing the most important issues, we would like to point out that one issue we detected during the evaluations and observations has been fixed. The particular issue concerned a button that was functional in the training environment of Gemini, but not in the live environment. When a customer acquires information about his calling history, the agent had to press the 'more' button several times. Each time that button was pressed, 100 rows were added to the visible field. The training environment has a specially assigned 'all rows' button for this task (showing all rows, instead of just a part). This button has been activated in the live environment of Gemini. Whether this change was made because of the inquiry we did, or whether this was a planned change still remains unclear. Nevertheless, we are satisfied with the result, as are the Vodafone agents.

One important issue we detected during the observation relates to Chapter 2 where we described that users of different expertise levels have different areas of concern when using a software package (eg. learnability, efficiency (Nielsen, 1993)). This difference was vividly notable during the observations. Where novice agents reported to find it sometimes difficult to locate certain information, experts were more concerned with the system speed. Especially Ciboodle crashed many times during the observation. Expert agents cleverly switched to Gemini when this happened, thereby still being able to assist the customer (some information can be found in both software packages). This could very well be related to the aspect LaFrance (1989) refers to as the automaticity of expert problem solving. Whenever method A does not work, try method B.

Another important issue relates to the verification process, during which agents have to collect information from several windows. Expert agents found a way to quickly retrieve this information by placing 2 screens next to each other (BAN and account management screen). Normally you would have to select one of the tabs in the BAN screen to get this extra information. After the verification is completed, the agent can start assisting the customer and create a logging of the question/request. In some cases it is unclear in which software package this logging should be created. This causes for negative consequences because the user gets confused about how to use the programs properly (Wickens et al., 2004). This is another drawback of using two software programs, rather than one. Not only do users have to learn how to use both programs, but also for Vodafone itself it complicates things. There is no good overview of the currently existing requests or complaints, because the logging happens at two places. It is thus advisable to instruct agents were to create particular loggings, preferably in one program.

The final issue that will be discussed relates to the used window size and layout in Gemini. The standard account window (BAN) in Gemini only covers one fourth of the whole screen. This makes some information in data fields not completely visible, thus requiring action from the user. Some of these data fields are presented with a code. Expert agents recognize these codes, and do not see the need to make these data fields

wider. Novices however acknowledge the problem. Maximizing the window also does not result in the expected outcome. As stated by Mosier et al. (1986), these are examples of design flaws, and unfortunately we could not ask the designer why he chose this kind of window layout (Gemini is partially designed by an external company). It is especially important to know why this type of window design was chosen, because novice users regularly use the programs, and therefore encounter this problem on a daily basis.

As discussed above, it turned out that usability issues detected during the heuristic evaluations indeed caused problems for the users. It is therefore advisable to handle the mentioned issues seriously and, where possible, to try to improve the interface of both programs. Based on the findings of the two methods (heuristic evaluations and unstructured observations) used so far, we wanted to show the effect that improving the interface would have on the usability. In order to do this, we had to measure the possible performance increase that would occur as a result of presenting users with an improved user interface. For this reason, we set up an experimental user testing which will be discussed in Chapter 5.

5 User Testing

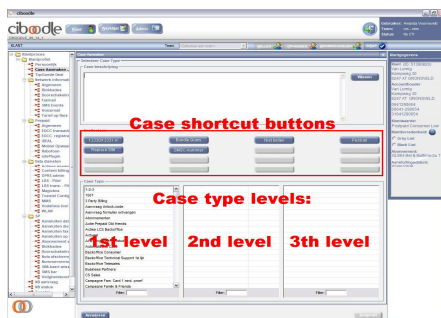
5.1 Introduction

Because we would like to investigate whether improvements (based on our heuristic evaluation and unstructured observation) of the user interface would actually lead to improved usability, we decided to set up an experiment that involved user-testing (Nielsen, 1994). The experiment will consist of two sessions; during the first session, the old interface will be used, and during the second session, an improved interface will be used. We will focus on expertise in relation to efficiency (case logging times) and effectivity/learnability (case creation method). Shorter logging times increase efficiency, while using the easiest creation method will increase effectivity (Nielsen, 1993). Before users can become effective they first have to learn these creation methods.

The main questions involved in this experiment are: what are the differences between expert and novice users when focusing on logging times and logging methods? How does their way of working improve from session 1 to session 2 when presented with the new user interface?

The first session will consist of a series of tasks participants normally would have to execute on the work floor. During this session, case creation times and case creation methods will be recorded. After this session participants will be introduced to a new Ciboodle profile which consists of better, so called, shortcut buttons. These buttons enable a user to create a case logging at once, rather than having to select the logging from 3 case type levels (figure 25).

Session1: the blue buttons are shortcuts



Session2: new buttons are available

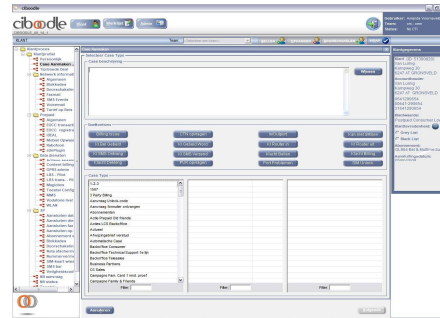


Figure 25. The old and the new interface of Ciboodle.

Participants get about a week time to get familiar with the new Ciboodle profile. After this period, they return for session 2. Again, they have to perform a series of regular tasks. Case creation times and case creation methods are recorded once again.

Afterwards, they have to fill out a usability questionnaire. This questionnaire will be discussed in the next Chapter (6).

In order to answer the main questions involved in this part of the research, several Hypotheses were set up.

First Hypothesis

Normally, experts perform specific tasks quicker than novices, just because they know the system better, and therefore do not have to think first but can act right away (Shneiderman, 1992). Hypothesis 1 tests the assumption that experts are indeed quicker than novices during both sessions:

H1: Experts are quicker than novices at completing the given tasks.

Second Hypothesis

Another interesting point to investigate is whether the session has influence on logging times. During the second session, participants can use the new Ciboodle profile which was presented to them upon completion of session 1. Using buttons for creating a case logging is faster than having to select 3 case type levels in the topic filter. Therefore, participants most likely need less time to complete the set of tasks. Another factor that might influence performance during the first session is arousal (Yerkes et al., 1908). Participants might experience such arousal because they are not sure what is expected of them. We will however not further investigate the effect of arousal on performance in this research.

Hypothesis 2 is formulated as follows:

H2: All participants are quicker at completing the given tasks during the second session.

Third Hypothesis

Although novices still have to learn the system, this also means that they can improve to a higher degree than experts can, because experts are already working at that higher level (figure 26). Thus, to investigate whether novices not only have improved, but also to see whether they, relatively speaking, show lower case creation times than experts when comparing session 1 with session 2, a third Hypothesis was created:

H3: Novices show a relatively higher improvement of case creation times than experts when comparing both sessions.

Hypotheses 1 till 3 use time as a performance indicator. Besides time, we used other measures to evaluate the efficiency and effectivity of users, namely case creation method and case creation manner. Basically, there are two methods for creating a case logging: via the buttons or via the topic filter. Each method can be used in a direct or indirect manner.

	Creation method	
Creation manner	Button direct	Filter direct
	Button indirect	Filter indirect

Table 4. Cells are mutually exclusive.

As the table 4 above shows, there are 4 possible logging options. Cells are mutually exclusive, meaning that each case logging can be appointed to just one cell. Mind that both creation methods are also mutually exclusive (eg. if 4 out of 12 cases are logged via the buttons, the 8 remaining cases are automatically logged via the filter).

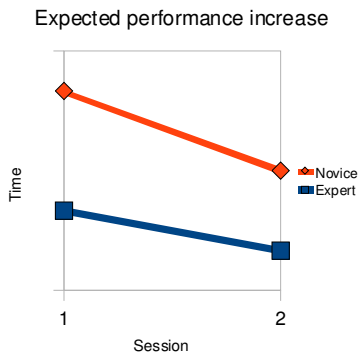


Figure 26. The time needed to which perform all tasks will likely increase decrease in the second session.

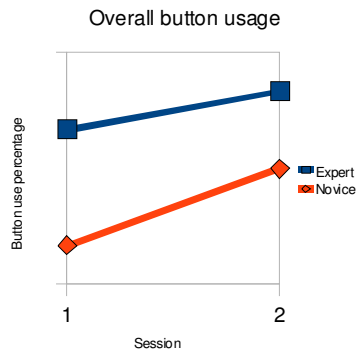


Figure 27. The overall button use is likely to increase in the second session.

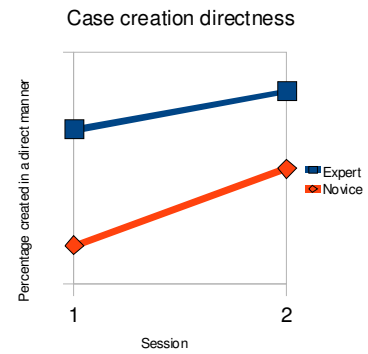


Figure 28. The directness in cases are created is likely to increase in the second session.

Normally, users become experts after using a particular software program for a certain while. They get to know how things work and learn to use shortcuts or short keys when available in the program (figure 27 and 28). They might even start to develop their own methods for performing certain tasks quicker (eg. by using mnemonics). Novices on the other hand have to learn the program first before they can start to use it more efficiently.

The following three Hypotheses investigate preferences for creation methods between the two expert levels.

Fourth Hypothesis

Hypothesis 4 tests the assumption that experts have a higher preference than novices for using buttons rather than the filter to create a case logging:

H4: Experts use buttons for creating a case logging more often than novices do.

Fifth Hypothesis

As mentioned before, using the button method is faster than the filter method. It is therefore expected that button use of all participants will be increased in the second session:

H5: Button usage will be higher in the second session than in the first session

Sixth Hypothesis

Because novices are new to Ciboodle, and therefore might not use the case buttons yet, it is plausible to say that novices will show a higher increase of button use in the second session when compared to the expert group. This is because participants are encouraged to use these buttons. Hypothesis 6 is stated as follows:

H6: Novices show a relative higher increase of button use than experts when comparing session 1 with session 2.

Besides the creation method, the manner of creating a case (direct/indirect) most likely differs between experts and novices. Experts know where each case logging can be found, whereas novices have to learn by doing. This takes time. The following 3 Hypotheses investigate preferences for a particular creation manner between the two expert levels.

Seventh Hypothesis

Hypothesis 7 states that experts use a more direct manner than novices to create cases:

H7: Experts create cases in a more direct manner than novices do.

Eight Hypothesis

Again, both users groups are likely to show an increased use of a direct creation manner during the second session, because they have learned since the first session. Hypothesis eight tests this:

H8: Cases will be created more direct in the second session than in the first session.

Ninth Hypothesis

As briefly discussed above, the notion that novices have to learn to work more direct also means that they have the ability to improve more than experts, because experts already use a more direct manner. To test this, the following Hypothesis has been formulated:

H9: Novices show a relative higher increase of using the direct creation manner than experts when comparing session 1 with session 2.

Interaction effect

Because there are more logging methods that each have a different logging manner, we expect to find an interaction effect. More specifically, we expect the logging manner is influenced by the logging method.

5.2 Method

Design

The two main variables that were investigated during this research are expertise and session. The expertise level consisted of two groups: novice and expert users. The session variable was split up into the first and second session. The experiment thus consisted of a 2 (expertise: expert vs. novice) x 2 (session: first vs. second) design that was used to investigate usability differences between expertise levels of Vodafone agents. These differences relate to working methods, accuracy, and other performance indicators. Time was used as such a performance indicator to investigate the learning process of both expertise levels over two sessions. In order to analyse logging times per session, based on the used creation method (filter vs. button) and level of expertise, a 1 (logging time) by 3 (creation method vs. session vs. expertise) mixed design was used.

Participants

For the experiment we created two participant groups: experts and novices. Each group contained 5 agents of the departments Postpaid Consumer, Prepaid, and Distribution. In total, 30 agents participated (12 men and 18 women). Table 3 gives an overview of the different departments and expertise levels.

Department	Expertise	
	Expert (n)	Novice (n)
Postpaid Consumer	5	5
Prepaid	5	5
Distribution	5	5

Table 5. 30 agents participated in the experiment.

To select expert users we searched for agents that already worked for over a year at the call center. To select novice users we searched for agents that worked for less than 4 months at the call center.

Because our experiment consisted of two sessions, and agents had to attend both of them, availability was another precondition. This meant that each agent had to attend both sessions, and they should work on a regular basis between both sessions. This was necessary because they had to get used to the new Ciboodle profile.

Materials

The software program that was evaluated during the user testing is Ciboodle. This particular software package allowed us to make changes, such as assigning new shortcuts and creating user profiles. We created such profiles using the top 16 case loggings of the last months¹ (an example of such a login would be: customer has a question about a particular service, like SMS or MMS). Each profile thus consisted of 16 case-buttons that enabled the users to quickly create a high frequently occurring case logging. The idea is that using a button is quicker than using the regular filter. Figure 29 shows 6 of these buttons (in total there are 16 of such buttons, but not all of them were initially programmed, at least at some departments).

¹ This top 16 case loggings was created by the department 'reports'

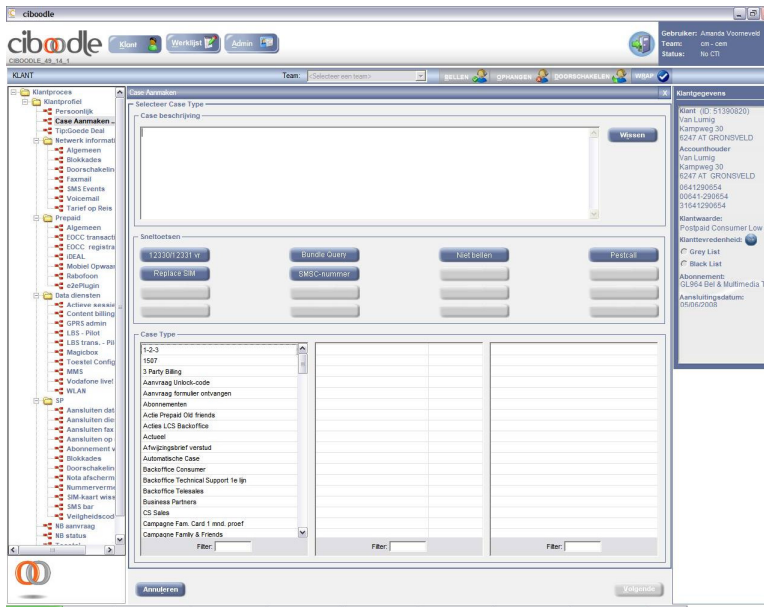


Figure 29. Ciboodle main screen.

The sessions took place in a closed office. This way the chance that participants got distracted was minimized. Note however that this is not the normal work environment, meaning there could be some influence (eg. higher concentration), but because both sessions would be held in similar offices, this effect would be cancelled out.

During the experiment we used a Dell laptop that was connected to an Eizo TFT screen. We used a laptop because the offices in which the experiment took place could only be reserved for a certain amount of time. After each session, the room has to be cleared.

The resolution of the external screen was set to 1280x1024 pixels because this was the minimum resolution that was needed to view Ciboodle full screen. For this same reason we could not use the laptops' internal screen because the resolution was too low. We also connected an external keyboard and mouse to the laptop. This way participants could move both peripherals in a desired position. The complete setup is displayed in figure 30.



Figure 30. Setup position of the computer and camera.

In order to be able to measure case creation times and to count case creation methods we used a Sony DCR-HC17E Handycam to record the experiment. The handycam was placed on a tripod which stood behind the participant. This way we could record the participants' usage of the peripherals, capture the screen, and avoid possible arousal of the participants because the camera was not directed at their face.

Procedure

After we found 30 agents who were willing to participate, we informed them via email about the time and date at which they were expected in our office. Also, we briefly explained the setup of our research without informing them about the underlying purpose of the experiment.

Next, we created user pairs: one novice and one expert user of the same department. Also, three different sets of tasks were created, one for each department. Within participant pairs, the task order was the same; between pairs, the order was randomized. This randomizing was done to prevent agents from discussing the tasks they had to perform during the sessions. Since there were two sessions, 2 sets of similar tasks were created (eg. customer cannot receive SMS vs. customer cannot send SMS). Keeping the questions similar between the sessions enabled us to ascribe possible differences to other factors than the case topic. The full task list can be found in the appendix (9.1).

Session 1

During the first session participants were asked to create 12 case loggings in Ciboodle. They were instructed to create each case like they would during a real conversation with a customer (ie. fill out every required field in the logging, choose appropriate topics). After they finished all tasks, the new Ciboodle profile was introduced to them. Mind that every department uses its own user profiles because they all handle different customer problems. The first session took participants about 15 minutes to complete.

After the first session, participants went back to their normal work environment but continued using the new profile. This way they could get used to the new shortcut buttons. About a week later, the second session took place.

Session 2

The second session was similar to the first session, except during this session they could use the new user profile. Afterwards, participants had to fill out the two questionnaires. This session took participants about 30 minutes to complete.

After both sessions were completed, we reviewed all video tapes and created a large database which consisted of variables like the participants' number, department, expertise, case logging times, shortcut usage, accurateness etc. Once we completed the database, we started the analysis.

5.3 Results

In order to test Hypotheses 1 to 3, case creation times were submitted to a 2 (expertise: expert vs. novice) x 2 (session: first vs. second) MANOVA, with the last variable manipulated within participants.

Hypothesis 1 and 2

Hypothesis 1 (experts are quicker than novices) was not confirmed, $F < 1$. Also, we found no confirmation of Hypothesis 2 (participants complete the set of tasks quicker in the second session), $F < 1$.

Hypothesis 3

In confirmation of Hypothesis 3 (novices improve relatively more than experts in the second session), we found the expected interaction between expertise level and session, $F(1, 17) = 9.04$, $p < .01$. More specifically, novices improved their case creation time from session 1 ($M = 319$, $SD = 116$) to session 2 ($M = 197$, $SD = 190$), $F(1, 18) = 6.24$, $p < .05$. However, experts showed no improvement in their case creation time from session 1 ($M = 240$, $SD = 55$) to session 2 ($M = 319$, $SD = 154$), $F(1, 17) = 3.01$, $p = .10$. The mean case creation times are displayed in figure 31.

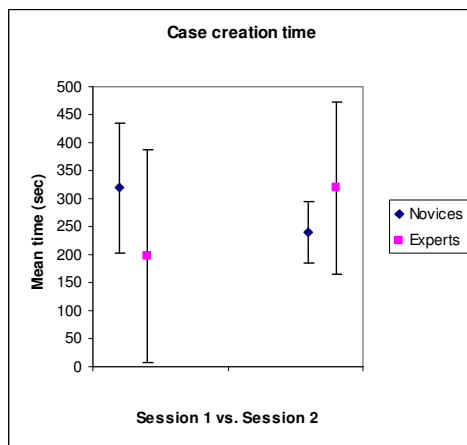


Figure 31. Mean time needed to create a case during the first session and the second session.

Besides the time required to perform all tasks, there were other variables that could be used to investigate differences in performance between experts and novices. A logging can be created using a button, or via a selection in the logging trees. We will refer to these options as the creation method. While participants were creating case loggings we noted that some of them first selected a button, but then switched to the logging trees in order to choose a similar yet different option. Thus, besides the used method, cases were also created in a direct or indirect manner. Whenever the participant directly chose the appropriate topic, we reported this as 'direct'. Accordingly, creating a case otherwise is called 'indirect'. These direct and indirect ways of creating a case logging will be referred to as the creation manner.

To summarize, a case can be created in a (in-)direct manner using either the button or the filter method (table 5). This means that the button use percentage equals 100% minus the filter use percentage. The same applies for direct vs. indirect creation manner. For this reason, we will only discuss one of both creation methods, and we will also only discuss one of both creation manners.

In order to test the Hypotheses 4 to 8, case creation methods were submitted to a 2 (expertise: expert vs. novice) x 2 (session: first vs. second) x 2 (creation manner: direct vs. indirect) MANOVA, with the last variable manipulated within participants.

Hypothesis 4

Results could not confirm Hypothesis 4 (experts use the button method more than novices do), $F < 1$.

Hypothesis 5

Confirming Hypothesis 5 (button use will be higher in the second session), we found an interaction effect between session and the use of a particular method, $F(1, 24) = 50.51, p < .01$. More specifically, button use increased from session 1 ($M = .105, SD = .129$) to session 2 ($M = .364, SD = .133$). See figure 32.

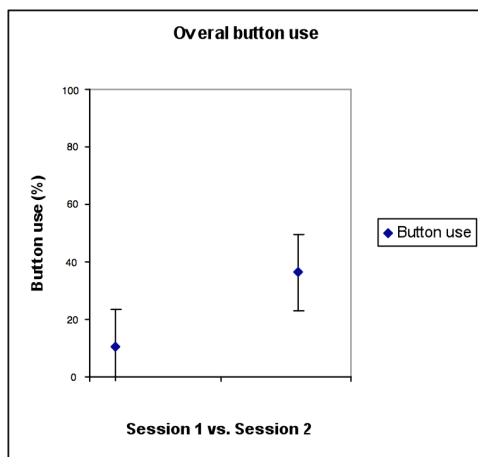


Figure 32. Button use per session.

Hypothesis 6

Hypothesis 6 (novices show a relatively higher increase of button use than experts in the second session) could not be confirmed, $F < 1$.

Hypothesis 7

Confirming Hypothesis 7 (experts create cases in a more direct manner than novices do), we found an interaction effect between expertise and the manner of creating a case, $F(1, 23) = 5.22, p < .05$ (figure 33). More specifically, experts create cases in a more direct manner ($M = .323, SD = .076$) than novices do ($M = .257, SD = .066$).

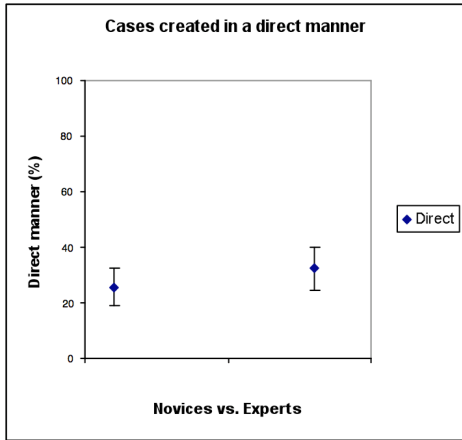


Figure 33. Experts create cases in a more direct manner.

Hypothesis 8

Hypothesis 8 (cases are created more direct in the second session than in the first session) could not be confirmed, $F < 1$.

Hypothesis 9

Hypothesis 9, which states that novices show a relatively higher increase of creating case loggings more direct than experts in the second session, could not be confirmed, $F(1, 23) = 1.04$.

Interaction effect

In accordance to our assumption, we found a three-way interaction between expertise level when compared to the case creation method (filter/button) and manner (direct/indirect), $F(1, 13) = 8.20, p = .013$. More specifically, experts use a direct creation manner when using the filter ($M = .369, SD = .179$) and an indirect creation manner when using a button ($M = .211, SD = .154$). See figure 34. However, novice users do not notably use different creation manners for different case methods, $F < 1$.

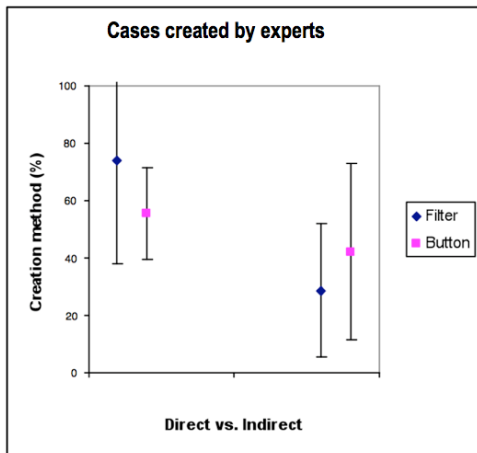


Figure 34. A comparison of creation method and creation manner for experts.

Mouse/keyboard use

Another remarkable finding is that during the user testing, expert users ($M = 18.60$, $SD = 15.165$) switch more often between their mouse and keyboard than the novice users did ($M = 8.13$, $SD = 9.403$), $F(1, 28) = 1.842$, $p < .05$. Figure 35 shows the number of times users switched between mouse and keyboard.

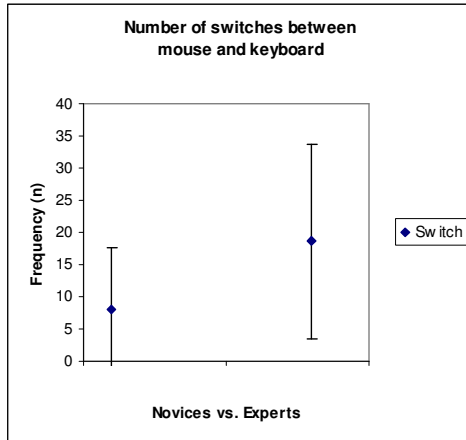


Figure 35. Experts switch more often between mouse and keyboard than novices do.

Mixed model analysis

To further investigate the time differences between creating a logging via the buttons or the filters, we used a mixed model analysis. This additional mixed model analysis had to be used because our logging time data points are independent. Results indicated that the used creation method has a major influence on the logging times, $F(1, 41) = 6.952$, $p < .05$ (figure 36). More specifically, using a shortcut button for creating a case took significantly less time ($M = 53.325$, $SD = 35.145$) than using a filter ($M = 64.893$, $SD = 46.804$).

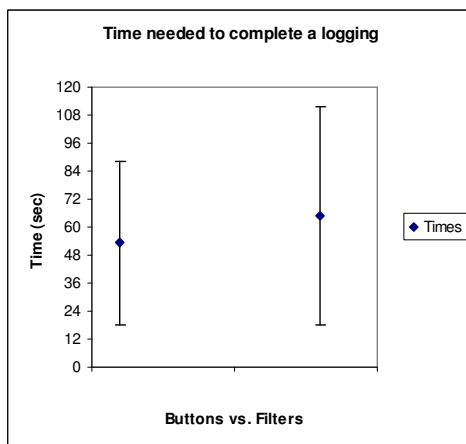


Figure 36. The time needed to complete a case logging using a button is significantly shorter than when using a filter.

The analysis furthermore indicated a marginally significant influence of expertise on logging times, $F(1, 26) = 3.582, p = .07$. This influence was however very small, and further research, using more participants, is most likely needed to investigate its true influence on logging times.

5.4 Discussion

In order to map the differences (eg. working methods) between novice and expert agents, but also to measure performance improvements (eg. case creation times) for each agent, an experiment was set up. This experiment consisted of 2 sessions during which participants had to perform a set of tasks. In total, 9 Hypotheses were used during this experiment.

Hypotheses 1-3

The results could not confirm Hypothesis 1, meaning there was no proof that experts are quicker at creating case loggings than novices. Also Hypothesis 2 could not be confirmed, since there was no proof that participants perform the set of tasks quicker in the second session. Results did however show that novices have relatively improved more than experts in the second session. This confirms Hypothesis 3.

Not finding a difference in logging times between experts and novices (H1) is somewhat peculiar because, as stated by Cellier (1997), you would expect that users who have more experience know their way around the software program, thereby needing less time to create a logging. On the other hand, having deeper knowledge of the program could very well be the explanation for the found results: experts might think longer in order to choose an appropriate case topic while novices search through the logging tree, which could turn out to be faster. These findings explain why hypothesis 2 was not confirmed, but they are in contrast with those of Farrington et al. (2006) who stated that experts score poorly when focusing on matters such as accuracy. Novices have more room for improvement when compared to experts. Because (some) novices perform poorly during their first interaction with a system, they have the ability to improve relatively more than an expert. This explains why we were able to confirm hypothesis 3.

Hypotheses 4-6

Hypothesis 4, which assumes that experts use buttons more than novices, could not be confirmed. Still, both expertise groups did use the button method more during the second session, which is in accordance with Hypothesis 5. The expected higher button use for the novice group during the second session could not be detected, though.

This refutes Hypothesis 6.

One would expect that users try to use the easiest and/or the quickest method for performing particular tasks (Cellier, 1997). In this experiment, using the buttons is both easier and quicker, because you only have to press one button. Since novices are new to the software package, they might not yet possess the knowledge to use the buttons. However, experts also did not use the buttons all of the time, at least not during the first session. During the second session both expertise groups started to use the buttons more than the filter. One possible explanation for not being able to confirm hypothesis 4 might be that, at the start of our research, some departments were using old profiles that did not

have all 16 buttons programmed. Also, many of these buttons were programmed for old case topics, which were not relevant anymore. During the second session participants could use the new profiles, which consisted of relevant buttons with a high frequent occurrence. Therefore they were more prone of using this method rather than searching through the filter. By doing this, the participants confirmed hypothesis 5. However, novices did not start using the buttons to a higher degree than experts did. This is probably caused by the fact that during the training that novices get, they are instructed to use such shortcut buttons, whereas expert users might not have received this information. This could then explain why hypothesis 6 was not confirmed. Therefore, this is an indication that special attention has to be paid to this type of shortcuts because they can help to improve case creation times. This will than eventually lead to a lower AHT scores (Average Handling Time is a measure used by Vodafone to indicate how fast an agent can complete, or handle, a conversation; lower AHT scores are better).

Hypotheses 7-9

Experts created case loggings in a more direct manner than the novices did. This confirms Hypothesis 7. Hypothesis 8 (cases are created more direct in the second session) could however not be confirmed. Also Hypothesis 9, which states that novices show a relatively higher increase of creating case loggings more direct than experts in the second session, could not be confirmed.

That experts choose a direct manner for creating a case logging could mean that they choose accurateness over handle time (Glaser et al., 1988). It furthermore indicates that expert users know where to find the right case logging because they have a better insight in the software's architecture (LaFrance, 1989). This explains why hypothesis 7 could be confirmed. The reason why we could not confirm hypothesis 8 and 9 could be that the time between the first and the second session was too short (especially for novices) to pick up a new creation manner. Using the buttons is a rather simple thing to learn, but being able to understand the structure of the used logging tress is another thing. Further research has to be conducted to test whether these assumptions are indeed the explanation for the found results.

Interaction effect

We found an interaction effect that was present in the expert group. It turned out that experts use a direct manner for creating a logging via the filter, and an indirect manner when they use the buttons. Our explanation for this result is that whenever the agent knows exactly where to find the appropriate case topic, he will swiftly open the logging filter and select the topic. However, when the agent is unsure about where to find the topic, he will use the buttons to navigate to the right logging filter, based on the category of that button. For example, the button "incoming calls" is used to navigate to the category "calls" and than the user can select "outgoing calls". The buttons are thus in an indirect manner used to quickly switch to a specific category in the filter.

Another result we found was that experts switch about twice as much as novices between mouse and keyboard. This could be an indication that experts are more accurate when filling out particular data forms. There is however no proof for this assumption. Also, you would expect that switching less between mouse and keyboard would improve the

logging time, but novices took longer to create all cases. Again, there is no proof for this result.

Besides the abovementioned results, we also counted the number of times participants used the text filter and whether or not they filled out the obligated questionnaire that was presented with some case loggings. Furthermore, we checked if participants used the intranet website which could be used to get help, and we counted how often they used keyboard shortcuts. Hellas, only a few participants actually performed any of these actions. Therefore we could not perform analysis on this data.

Buttons vs. Filters

The mixed model analysis showed that there was a significant difference in logging time when using one of both logging methods. Using the filter for creating a case logging took on average 12 seconds longer than using the button method. These results are astonishing because this means that great improvements can be accomplished by regularly updating the shortcut button profiles in Ciboodle. This updating is necessary because, for example, promotions are only valid for a certain time. When programming such a promotion as a shortcut button, you should remove this shortcut when the promotion has ended. Besides this regularly updating one can ask himself whether adding more buttons (how much?) would improve usability, because more buttons means presenting more information in one window. This can lead to informational overload, which might cause for agents to have to search longer for one particular topic. On the other hand, shortcut buttons have proven to work faster, and therefore are beneficial improvements for both Vodafone agents (more than half of the participants wished to continue using the new Ciboodle profile) as well as for Vodafone itself (shorter logging times means that more customers can be assisted in the same amount of time).

The discussed user-testing showed that you can improve the usability of Ciboodle by simply updating the user profiles. Because agents could only use the improved profile for about a week before the second session took place, we expect that agents will even improve more when they get used to the new profile. Besides Ciboodle, we also would like to get feedback from users about the usability of Gemini. Chapter 6 will describe our results of the usability questionnaire we used for this purpose.

Example:

*“Given that in the month September 2008 around 37000 Ciboodle cases² were logged of which 8500 could be logged by using the buttons rather than the filters, you could save 8500*12 seconds which equals to about 28 hours per month. This is equal to 0.16 FTE, which is quite substantial!”*

² These data are copied from the file ‘Contact Drivers’ provided by the department ‘reports’

6 Usability questionnaire

6.1 Introduction

The experiment discussed in Chapter 5 was setup to get insight in the usability of the software package Ciboodle. In order to investigate the usability of the other software package, Gemini, a questionnaire was used.

We used the data that was collected during the heuristic evaluations (Mosier et al., 1986) and the unstructured observations to formulate 50 questions (35 multiple-choice and 15 open) that were related to usability issues/problems.

The main questions we would like to answer in this section are what differences there exist in the way different expertise levels look at the presented usability topics. Do experts focus more on substantial problems (eg. performing trivial steps) rather than on superficial flaws (eg. used colors and fonts) (Glaser et al., 1988)? Are there issues that are judged as equally important by both expertise levels?

After completing the second session of the experiment, participants were presented with the questionnaire. Mind that not all participants were able to fill out the Gemini usability questionnaire because they did not use Gemini for their work. For example, the department Prepaid only uses the software package Ciboodle.

6.2 Method

Design

In order to measure the usability of Gemini, we used a 7 (usability topic: BAN, help, memo, redundant, short key, size, use) by 2 (expertise level: novice vs. expert) MANOVA, with the last variable between-subjects. Each topic consisted of several multiple-choice questions. The total score of all topics combined will serve as the usability measure. Also, the answers to the open questions will be taken into consideration for the usability measurement.

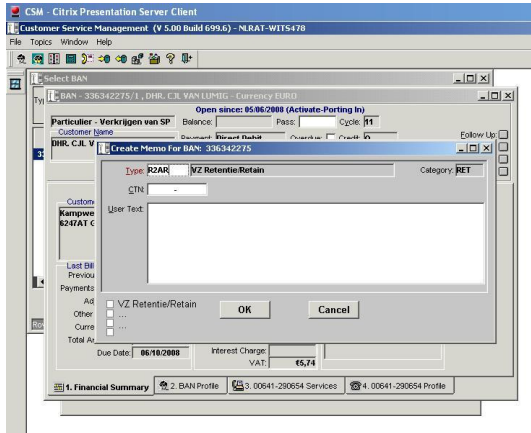
To be sure that the questions within each topic were related to each other, a reliability test was conducted. This test indicates whether different questions measure the same topic (eg. “Do you miss a help function?” and “Gemini offers plenty of help”). The reliability score of each topic is presented in the “Materials” section behind the description of that particular topic.

Participants

Initially we intended to use the same participants as during the experiment. However, as noted before, participants of the department Prepaid do not use Gemini. Therefore the resulting number of participants that filled out the usability questionnaire was 20.

Materials

In total, we formulated 50 questions of which 35 were multiple-choice and 15 were open questions (Appendix 9.2). Most of the open questions were presented with a colour image that visualised the particular problem/issue.



An example of an open question was “I would like to use shortcut buttons to create a memo” (figure 37). Participants could select an answer on a 5-point scale, ranging from “totally disagree” to “totally agree”.

Figure 37. Shortcut buttons for creating memos.

The multiple-choice questions were categorised into the following 7 topics:

- *BAN screen (Reliability score: 0.606)*

The BAN screen is a small window that displays all sorts of information regarding a Vodafone customer (eg. which kind of services the customer uses, where he lives).

- *Help (Reliability score: -0.465)*

This section contained questions about the help that was available in the software packages. Remember that both Ciboodle and Gemini do not have a traditional help menu. Gemini does have a menu, which is called “help”, but the user cannot select any options in this menu. Whenever an agent does have a question or a problem, he can use the intranet site Coach, or he can ask another agent/team coach to help him out.

- *Memo (Reliability score: 0.435)*

A couple of questions were used to investigate whether the available memo types were useful and descriptive enough.

- *Redundant (Reliability score: 0.752)*

Agents have to perform several actions during a conversation with a Vodafone customer. To check whether some actions might be found redundant, we created several questions that tested this matter.

- *Shortkey (Reliability score: 0.527)*

It is interesting to check whether agents know that shortkeys are available in the software packages. Using such keys can help make their work easier, and improve efficiency.

- *Size (Reliability score: 0.520)*

Gemini consists of several different windows of varies sizes. To determine whether the size of the used windows is large enough, several questions were assembled.

- *Use (Reliability score: 0.340)*

The last topic mainly consisted of questions that help to score the overall use of Gemini.

After all questions were formulated, they were printed on white paper and then bundled.

Procedure

After completing the second session of the previously discussed experiment (Chapter 5), participants were presented with the questionnaire, which they could fill out in the same room as the one the experiment took place in. On average, participants took about 15 minutes to complete the questionnaire. A complete version of the used questionnaire can be found in Appendix 9.2.

6.3 Results

In total, 20 participants filled out this questionnaire. We will first discuss the multiple choice questions, and then the open questions.

Multiple-choice questions (1-35)

In order to measure the usability of Gemini, the answers to the usability questions were submitted to a 7 (usability topic: BAN, help, memo, redundant, short key, size, use) x 2 (expertise level: novice vs. expert) MANOVA, with the last variable between-subjects.

Analyses showed a main effect of expertise level. That is, the usability topics were rated differently by novices and experts, $F(1, 18) = 5.28, p < .05$. More specifically, experts rated the usability of Gemini higher ($M = 3.630, SD = .261$) when compared to novices ($M = 3.296, SD = .379$). Neither a main effect of usability topic, nor an interaction between expertise and usability topic were found.

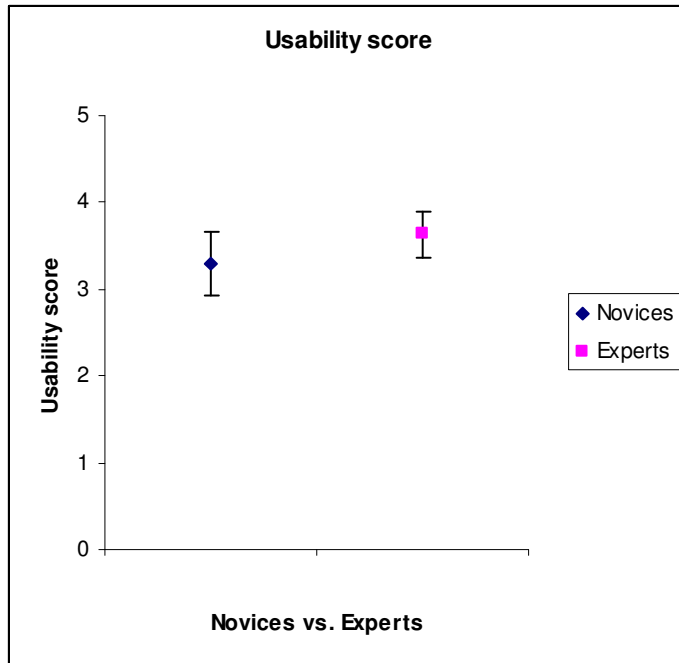


Figure 38. Usability score for each expertise group.

Open questions (36-50)

Of the 15 open questions, 2 questions are about the BAN screen (39, 46), 4 about help (41, 43, 45, 50), 1 about the memo screen (44), 2 about redundant actions (40, 47), 2 about shortcuts (36, 48), 3 about the used screen size (37, 38, 42), and 1 about the use of Gemini (49).

BAN screen

Adjusting the BAN screen such that it shows all the information needed during the verification process has turned out to be found a positive improvement. Only 3 participants indicated that this adjustment would not cause a huge improvement, although one of them admitted that it would shorten the verification process with about 2 seconds.

Placing an indicator on the first tab of the BAN screen that shows whether the customer still has a commitment with Vodafone was rated to be a small improvement. Most participants said it only saves one mouse click.

Help

15 Participants replied that it would be useful to present icons with a text label by default. At the moment, agents have the possibility to enable these text labels, but the labels are disabled again when the agent logs off.

After trying to close an opened BAN, the user sometimes receives an error message, which states that in order to close the BAN, he should close all associated windows. This error occurs when a window is minimized to the bottom of the screen. All participants indicated to understand the error, and knew how to solve the problem.

Currently, a one-word description becomes visible when the agent moves the mouse pointer over an icon. The proposed improvement (rather than only presenting one

word, give a short description of a function) was found positive by 18 participants. The 2 remaining participants indicated they would not need such a description, but novice users might appreciate it.

To the question which source of information (coach vs. colleagues) participants rather used, 40 percent answered neutrally, but another 40 percent indicated to prefer asking a colleague.

Memo

When the memo screen is opened, and next the BAN screen is selected, the memo screens disappears behind the BAN. Trying to return to the memo screen by using the menu option results in a warning “This window is already open.”. All participants indicated to understand the warning, but 4 of them answered that clicking the OK button would solve the problem. This is however not the case. When you click OK, the BAN screen reappears. The only way of making the memo screen visible again is by moving the BAN screen aside.

Redundant

Some icons are used more times for different functions. Only 3 participants could correctly describe the function of both icons. 8 participants thought that both icons had the same function. The remaining participants were not sure about one of both icons.

Whenever an agent wants to change the current commitment, all options are viewed by default. However, depending on the current commitment, only particular options can be selected. 18 Participants indicated that it would be better to only show allowed options, because this would decrease the chance of making an error, and also makes the option list more clarifying. Also, it would save time because then you would not have to search through the whole list in order to find valid options.

Shortkey

Based on the shortcut buttons in Ciboodle, we presented the participants with a similar list for creating memos in Gemini using buttons. 16 Participants welcomed the idea of being able to instantly create a memo via a button. The other participants indicated that they knew all relevant memo codes by heart after using them for a long time.

16 Participants answered positive to the suggestion of implementing a menu bar-a-la-Windows© into Gemini. This menu bar displays all active windows, and can be used to quickly jump between windows. This would avoid the previously discussed issue with overlaying windows.

Size

Maximizing the BAN screen does not lead to the expected result. Instead of enlarging the whole BAN screen, only the border is widened. All participants found this irritating because they expected to get more information in one screen when using the maximise button.

Because the used window size of the letter request screen is too small to read the full description, it could be possible to select the wrong option. Only 1 participant admitted to have made a mistake because of this. The rest of the participants indicated

that they almost never send out letter request, and when they do, they take their time and read the description by scrolling the window sideways.

The last question about the used window sizes in Gemini related to the memo window. In the system and user text field, only 2 lines are visible at one moment. Using the scrolling function of the mouse does not result in the expected result (the field, rather than the text, moves up and down). 18 Participants answered that important information can be missed because the information field is too small. Since there is plenty of space left on the screen, it is advisable to make this memo field bigger.

Use

When an agent wants to send a specified bill to a customer, he has to manually charge the customer for this bill. The proposed improvement would add an option to the billing window, making it possible to instantly select the appropriate charges. Seventeen participants would like to see such an option, because the current method required you to open another window, which takes longer.

6.4 Discussion

The used questionnaire has shed some light on several usability issues that were detected during the heuristic evaluations (Chapter 3) and the unstructured observations (Chapter 4). Participants were presented with possible improvements and were asked to reflect on these improvements.

Experts scored higher on the usability measure (questions 1-35) than novices did, which replicated the results of Mendoza et al. (2005). Mendoza et al. (2005) indicated that this effect can be ascribed to the level of frustration, which drops after a certain period of time. Usability issues become less important, and therefore users rate the overall usability higher. This same effect was visible in the second part of the questionnaire (questions 36-50); experts rated the systems' usability higher than novices did. These findings confirm that most of the detected usability issues are especially of concern for novice users. After using Gemini for a longer time, agents get used to the layout of all the screens, and manage to find their own way through the system. This does however not mean that the current interface of Gemini is fine as it is. Even experts admitted that several issues could be avoided by implementing the proposed improvements.

Novice agents reported that the biggest problem was how to figure out were to find particular functions. This replicates earlier research by Cellier et al. (1997). The quickest method (used in 40% of all cases) is asking a colleague for help. Agents use this method because of the time pressure that is present because the customer is still on the phone. However, team coaches (TC) instruct their agents to use Coach whenever they need help. This is somewhat of a concern. Some TC's are stricter than others. When an agent turns to a colleague for help, some TC's interrupt, and redirect the agent in question to Coach. The best method to avoid such situations is to improve the training that agents receive at the beginning of their career. As Glaser et al. (1988) indicates, pointing users at all different shortcuts, shortkeys, and other tips and tricks gathered from experienced users helps beginning users to get a better insight in the system, and its' possibilities.

The used questionnaire provided information that can be used to improve the interface of Gemini. Most of the proposed improvements might only seem small in nature, but they can still influence the overall usability of Gemini in a more serious way due to number of users that interact with the system every single day. And although Mendoza et al. (2005) indicated that expert users seem to care less about particular usability issues, this does not mean that it has no effect at all, let alone the effect it does have on novice users.

In Chapter 7 we will present our final discussion in which we will talk about the influence that user expertise has on the usability experience. Also, we will give our recommendations for improving the interfaces of both Ciboodle and Gemini.

7 Discussion & Recommendations

7.1 Discussion

The intention of this study was to investigate the influence of user expertise on the usability experience. Additionally, we intended to map the usability of the software programs used by Vodafone call center agents. The study started by evaluating the system, using heuristics. Next, unstructured observations were used, followed by an experiment in which we tested 2 types of interfaces. Lastly, we used a usability questionnaire to receive additional input from the current users.

We chose heuristic evaluations (Nielsen et. al, 1990) as a method to quickly get an overview of the programs Ciboodle and Gemini. This helped us to answer the first research question (What are the design flaws of Ciboodle and Gemini?). The most important issues concerned with Ciboodle are the outdated user profiles, and the double use of shortcuts. Issues in Gemini concern the verification process that takes too long because information has to be collected from several screens. As opposed to Ciboodle, Gemini uses icons to represent more functionalities. Furthermore, both programs lack an auto-save function for error prevention, and they lack a help menu to assist users when this is needed. As indicated by Mosier et al. (1986) and Nielsen et al. (1990) these functionalities are very important and should therefore definitely be implemented in the current software. For a list of all usability issues, we would like to redirect you to the results section of Chapter 3. Using the evaluation method thus helped us to not only report, but also to actually experience usability issues. This way we could imagine what kinds of situations Vodafone agents encounter during their normal work. During this part of the research, we were not yet able to investigate the differences between the two user expertise levels (experts and novices) used in this study.

We continued our study by performing unstructured observations (Wixon et al., 1990). This particular method enabled us to experience how actual users deal with usability issues in real life. Also, we started to gain more information about the various types of users that use the software packages Ciboodle and Gemini, which helped us to answer the second research question (What are the main differences between novice and experts users?). We learned that not all lay-mans' definitions of a particular user group were true. For instance, experts are not necessarily quicker at performing their job than novices. Also, not all experts preferred to use provided shortcuts and shortcuts. Glaser et al. (1988) however indicated that one can expect that an expert agent tries to be as efficient as possible. Novices on the other hand also did not always comply to the stereotype of a "beginner". Some novices learned quicker than others. It seems that whomever is classified as a novice (most likely someone that has just started working) can also turn out to be an expert after all. It all depends on the previous experience a user has with software programs, as indicated by Cellier et al. (1997). Glaser et al. (1988) furthermore states that users who use a computer all day are more likely to develop a working strategy that helps them to quickly accomplish certain tasks. Other factors that influence user performances are mood (eg. happy or tired) but also the time of day. We noted that during some hours of our observation, when the workload was high, agents can

become tired, thereby not performing as well as they would have when there was little workload. This replicates the results of Yang (2003). Obviously, this was to be expected at a call center, where a customer can call in at any time. Brown et al. (1992) indicates that the workload will be higher at times at which customers do not have to work themselves. Last, but certainly not least, we noted that a particular part of the software program Ciboodle was outdated. Because we were able to update this part of Ciboodle, it led us to conduct an experiment in which we tested 2 user interfaces with the help of 30 Vodafone agents.

The experiment consisted of a user-testing (Nielsen, 1994) in which we tested the old interface against an improved version of that interface. This testing was done in 2 sessions. The focus of this user-testing was on the differences in performance between expert and novice agents when presented with an interface with improved usability. Also, we investigated achieved improvements from session 1 to session 2 for each user group. The results of this experiment showed that there was no real difference in logging times between novice and expert agents. These results are in contrast with the notion of Wickens et al. (2004) that experts are more efficient than novices. Also, these results can be an indication that the type of user defined in this research as “expert” is in fact just a novice user that has more experience but uses a working method that feels familiar to them. By using this method, they prevent themselves from learning new methods; methods that might be faster as the ones they got used to. Experts did however create case loggings in a more direct manner, indicating that they know the system better, because they can locate the appropriate topic using fewer steps than novices do. These results confirm the findings of LaFrance (1989) where experts proved to be more functional and have a better view of a programs’ internal structure. These findings also confirm the definition of an expert as described by Glaser et al. (1988). When comparing session 1 with session 2, novices seemed to have improved more than experts. Also, both user groups switched to a quicker logging method in the second session (by using buttons rather than filters). The expectation that novices would show a relatively higher improvement, as described in Wickens et al. (2004), thus only turned out to be true for logging times, but not for creation method and creation manner.

The last method, a usability questionnaire, was used to collect user feedback about the usability of Gemini. The questionnaire was based on the usability issues that were detected during the heuristic evaluations (Chapter 3) and the unstructured observations (Chapter 4). Users were presented with interface improvements and they were asked to comment on them. Although several usability issues were presented, experts rated the usability of Gemini higher than novices, thereby confirming the findings of Mendoza et al. (2005) that some usability issues become less important after a while. Furthermore, results showed that only 3 participants could correctly name the function of a particular icon. This could be an indication that also experts are affected by a system that has degraded learnability properties. This is in contrast to Shneidermans’ (2002) notion that only novices have the need for a system with a high learnability degree.

Concluding, the aforementioned results are an indication that the improved user interface has lead to better performance, as well as more satisfied user (more than half of the participants are still using the new profile). Some improvements are more valuable for novice users; others are more valuable for expert users. It is however questionable

whether the difference between the used user groups is large enough to call one group “novices” and the other group “experts”, as results have turned out.

7.2 Recommendations

Throughout this study we encountered several situations in which usability improvements can be achieved. These situations helped us to answer the third research question (How can the usability of both software packages be improved?). Before I will present our recommendations, I would first like to quote my Vodafone supervisor Tom van Buitenen, who came up with a brilliant, yet simple solution for solving the problem of having to switch back and forth between the programs Ciboodle and Gemini.

“Waarom geef je niet alle adviseurs twee schermen, zodat ze beide programma’s tegelijk kunnen zien!”
“Why won’t you give all agents two screens, so they can view both programs simultaneously!”

As for our proposed improvements, they can be achieved by following a couple of steps which will now be presented.

To improve the usability of Ciboodle, we would advise to regularly update the shortcut button profiles. This can easily be achieved by programming the most frequent occurring cases under the 16 logging buttons. However, care has to be taken not to change the whole set of buttons each month (eg. if a certain promotion only lasts a weeks you should not program this topic as a shortcut button) because that would mean agents have to learn the button layout every month all over again, thereby affecting the learnability of the system (Nielsen, 1993). Another thing that definitely needs to be taken care of is the double use of shortcuts. As Mosier et al. (1986) indicates, a designer should choose a different character for each functionality.

Improving the usability of Gemini is another thing. From time to time, Gemini is updated, but high costs are concerned with these updates (or so called Changes). Taking this fact into consideration, we still advise Vodafone to look into the possibility of making changes to the interface in order to improve usability. Although most of these changes might only lead to small improvements of the interface, it still adds to a positive experience the user gets when using the program. Also, many agents use the program on a daily basis, thereby profiting from a better interface. The first thing we would change when having the opportunity to redesign Gemini would be the used BAN screen (both its’ size, as well as the information that is presented in it). At the moment, the used window area is far from optimal (Mosier et al., 1986). Therefore, we would also advise to look at the possibility of saving changes that users make in their Gemini account. These changes range from adjusted window size to the position of windows on the screen. At the moment, each user has to change these settings every time they log into Gemini.

Besides improving both software interfaces, we would also advise Vodafone to look into the training that agents get when they start working at the call centers. A lot of expert agents noted that the current training is a lot more elaborate than the one they got some time ago. Therefore regularly having agents follow training would keep them updated to the current areas of concern. For instance, if agents develop a method for

solving particular problems, this information should be shared with colleague agents. Trainings are ideal for these sorts of situations.

Another thing that needs close attention is availability of help options. The intranet site Coach contains thousands of pages, many of which are outdated and contain irrelevant information. This is mainly caused by the fact that some departments can edit Coach themselves. Also, there is no real Webmaster that, so to speak, coaches “Coach” itself. Because agents can only turn to Coach in case they need help, it is important that they can locate information quickly (especially because usually a customer is on the phone when the agent uses Coach to look up certain information). Therefore, we advise Vodafone to provide agents with appropriate help by updating Coach.

To address the last research question (What impact can the improvements have on cost reduction?), we would like to redirect you to the last paragraph of the discussion of Chapter 5. Here we presented a simple calculation that showed what the consequences are when you make better use of the systems’ capabilities. Making just a small adjustment turn out to cause great improvements on a yearly basis (12 seconds less per case logging equals to an 0.16FTE improvement).

8 References

1. Allen, P. A., Wallace, B., & Weber, T. A., (1995). Influence of case type, word frequency, and exposure duration on visual word recognition. *Journal of Experimental Psychology: Human Perception & Performance*, 21(4), 914-934.
2. Bagozzi, R. P., Davis, F. D., & Warshaw, P. R. (1992). Development and test of a theory of technological learning and usage. *Human Relations*, 45(7), 660-686.
3. Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., & Zhao, L. (2002). *Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective*. Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6340.
4. Cellier, J., Eyrolle, H., & Marine, C., (1997). Expertise in dynamic environments. *Ergonomics* 40, 28–50.
5. Bevan, N. (1994) ISO 9241-11 Ergonomic Requirements for Office Work With VDTs. http://www.usabilitynet.org/tools/r_international.htm#9241-11.
6. Danino, N., (2001). *Heuristic Evaluation – a Step by Step Guide*. <http://www.sitepoint.com/print/heuristic-evaluation-guide>.
7. Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, 35, 982-1003.
8. Dix, A., Abowd, G., Beale, R. and Finlay, J. (1998), *Human-Computer Interaction*, Prentice Hall Europe.
9. Farrington-Darby, & Wilson, J.R. (2006). The nature of expertise: A review. *Applied Ergonomics* 37, 17-32.
10. Fishbein, M., & Ajzen, I. (1975). *Belief, attitude, intention, and behavior: An introduction to theory and research*. Reading, MA: Addison-Wesley.
11. Glaser, R., & Chi, M., (1988). Overview. In: Chi, M., Glaser, R., Farr, M. (Eds.), *The Nature of Expertise*. Lawrence Erlbaum, NJ, 15-36.
12. LaFrance, M. (1989). The Quality of Expertise: Implications of Expert-Novice Differences for Knowledge Acquisition. *SIGART Newsletter* 108, 6-14.
13. Mendoza, V. & Novick, D. G. (2005). Usability over Time. *SIGDOC*, 151-158.

14. Mosier, J. N., & Smith, S. L. (1986). Guidelines for designing user interface software. The MITRE Corporation Bedford, Massachusetts, USA.
15. Nielsen, J. (1993), Usability Engineering. Academic Press, London.
16. Nielsen, J. (2005) First principles of interaction design. ISSN 1548-5552 Useit.com.
17. Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. CHI '90 Proceedings, Denmark.
18. Rogers, T. G., & Pegden, C. D. (1977). Formatting and organizing of a human engineering standard. Human Factors, 19(1), 55-61.
19. Shneiderman, B. (1992) Designing the user interface: Strategies for effective human-computer interaction (2nd edition). Reading, MA: Addison-Wesley.
20. Shneiderman, B. (1998). Designing the user interface: Strategies for effective human-computer interaction (3rd ed.) . Reading, MA: Addison-Wesley Publishing.
21. Van Welie, M, Van der Veer, G. C., & Eliëns, A. (1999) Breaking down Usability. Faculty of Computer Science, Vrije Universiteit Amsterdam.
22. Van Welie, M., van der Veer, G.C. and Eliëns, A. Breaking Down Usability, in Proceedings of INTERACT 99 (Edinburgh, August 1999), IOS Press, 613-620.
23. Wickens, C., Lee, J, Liu, Y., & Gordon Becker, S. An introduction to human factors engineering (2nd edition). Pearson Education Inc., New Jersey; 2004.
24. Wixon, D., Holtzblatt, K., & Knox, S. (1990). Contextual design: An emergent view of system design. CHI '90 Proceedings, 329-336.
25. Yang, C.C. (2003). Decision Support Systems, 35(1), 89. <http://il.proquest.com>.
26. Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength stimulus to rapidity of habit formation. Journal of Comparative Neurological Psychology, 18, 459-482.

9 Appendix

9.1 Ciboodle: user testing tasks

First session:

Takenlijst Ciboodle - Distributie

Dit deel van het onderzoek bestaat uit het aanmaken van (niet bestaande) cases in Ciboodle. Hieronder vind je een aantal omschrijvingen van zogenaamde gesprekken die je met een klant voert. Het is de bedoeling dat je voor **ieder** gesprek een case in Ciboodle aanmaakt. Probeer de cases zo zorgvuldig mogelijk in te vullen (i.e. zoals je tijdens de training geleerd hebt). In ieder verplicht tekstveld kun je de tekst “test case” zetten. Je hoeft echter **geen cases toe te wijzen**. Als contact type kun je “telefoon in” kiezen. Alle cases dienen opgeslagen en gesloten te worden. Mocht je nog vragen hebben, stel ze gerust.

Alle cases kunnen aangemaakt worden onder één test klant met CTN:
0652673363 (Naam: Spice)

1. Een klant wil graag weten wat zijn PUK code is. Noteer hieronder de laatste 2 cijfers van de PUK code en maak een case aan in Ciboodle.
PUK code: ...
2. Een klant wil graag weten wat zijn mobiele nummer is.
3. De klant ervaart routeringsproblemen wanneer hij gebeld wordt.
4. Er wordt een vraag gesteld over een factuur.
5. Het ontvangen van een SMS geeft problemen.
6. Het versturen van een MMS geeft problemen.
7. De klant wil graag van prepaid naar postpaid porteren.
8. De klant ervaart problemen nadat hij geporteerd is.
9. Een klant geeft aan niet te kunnen bellen.
10. Een klant vraagt om bepaalde kosten te crediteren.
11. De klant heeft een SIM unlock klacht.
12. Een klant geeft aan niet te kunnen bellen en gebeld worden.

Takenlijst Ciboodle - Postpaid

Dit deel van het onderzoek bestaat uit het aanmaken van (niet bestaande) cases in Ciboodle. Hieronder vind je een aantal omschrijvingen van zogenaamde gesprekken die je met een klant voert. Het is de bedoeling dat je voor **ieder** gesprek een case in Ciboodle aanmaakt. Probeer de cases zo zorgvuldig mogelijk in te vullen (i.e. zoals je tijdens de training geleerd hebt). In ieder verplicht tekstveld kun je de tekst “test case” zetten. Je hoeft echter **geen cases toe te wijzen**. Als contact type kun je “telefoon in” kiezen. Alle cases dienen opgeslagen en gesloten te worden. Mocht je nog vragen hebben, stel ze gerust.

Alle cases kunnen aangemaakt worden onder één test klant met CTN:
0652673363 (Naam: Spice)

1. Het verzenden van een SMS geeft problemen.
2. De klant ervaart routeringsproblemen wanneer hij gebeld wordt.
3. De klant heeft een SIM unlock vraag.
4. De klant heeft een vraag over het aansluiten van OnderOns.
5. Er treden problemen op nadat de klant naar Vodafone geporteerd is.
6. Een klant wil graag weten wat zijn PUK code is. Noteer hieronder de laatste 2 cijfers van de PUK code en maak een case aan in Ciboodle.
PUK code: ...
7. De klant meldt dat zijn toestel defect is.
8. Een klant vraagt je om een probleem met zijn factuur uit te zoeken.
9. De klant krijgt een foutmelding wanneer via nummer 1200 optie 1 gekozen wordt.
10. Een klant geeft aan niet gebeld te kunnen worden.
11. Het tegoed van de klant wordt niet bijgewerkt.
12. Een klant heeft een vraag over verzenden met SMS Direct.

Takenlijst Ciboodle - Prepaid

Dit deel van het onderzoek bestaat uit het aanmaken van (niet bestaande) cases in Ciboodle. Hieronder vind je een aantal omschrijvingen van zogenaamde gesprekken die je met een klant voert. Het is de bedoeling dat je voor **ieder** gesprek een case in Ciboodle aanmaakt. Probeer de cases zo zorgvuldig mogelijk in te vullen (i.e. zoals je tijdens de training geleerd hebt). In ieder verplicht tekstveld kun je de tekst “test case” zetten. Je hoeft echter **geen cases toe te wijzen**. Als contact type kun je “telefoon in” kiezen. Alle cases dienen opgeslagen en gesloten te worden. Mocht je nog vragen hebben, stel ze gerust.

Alle cases kunnen aangemaakt worden onder één test klant met CTN:
0652673363 (Naam: Spice)

1. De klant heeft een vraag over online opwaarderen.
2. Een klant verzoekt een SIM replace via de IZI client.
3. Een klant wil een vervoersblokkade laten plaatsen.
4. De klant heeft een vraag over online registreren.
5. Een klant geeft aan niet gebeld te kunnen worden.
6. Er komt een vraag binnen van een klant die meer weten wil over het activeren/deactiveren van zijn voicemail via 12330/12331.
7. De klant belt voor de specificaties van zijn SIM kaart.
8. De klant heeft een vraag over zijn saldo.
9. De klant wil graag van prepaid naar postpaid porteren.
10. De klant heeft een vraag over het nummer van de SMS-Centrale.
11. De klant heeft een algemene vraag over mobiel opwaarderen.
12. De klant heeft een vraag over de opwaardeerbon.

Second session:

Takenlijst Ciboodle - Distributie

Dit deel van het onderzoek bestaat uit het aanmaken van (niet bestaande) cases in Ciboodle. Hieronder vind je een aantal omschrijvingen van zogenaamde gesprekken die je met een klant voert. Het is de bedoeling dat je voor **ieder** gesprek een case in Ciboodle aanmaakt. Probeer de cases zo zorgvuldig mogelijk in te vullen (i.e. zoals je tijdens de training geleerd hebt). In ieder verplicht tekstveld kun je de tekst “test case” zetten. Je hoeft echter **geen cases toe te wijzen**. Als contact type kun je “telefoon in” kiezen. Alle cases dienen opgeslagen en gesloten te worden. Mocht je nog vragen hebben, stel ze gerust.

Alle cases kunnen aangemaakt worden onder één test klant met CTN:
0652673363 (Naam: Spice)

1. De klant wil graag naar Vodafone porteren.
2. De klant kan niet SMSen.
3. De klant ervaart problemen met de netwerk dekking.
4. De klant wil eerder overstappen naar Vodafone.
5. Een klant geeft aan niet gebeld te kunnen worden.
6. De klant geeft aan een bepaald nummer niet te kunnen bellen.
7. De klant krijgt een foutmelding wanneer via nummer 1200 optie 1 gekozen wordt.
8. De klant ervaart problemen tijdens het doorlopen van de unlockprocedure.
9. Het verzenden van een SMS geeft problemen.
10. Een klant vraagt je om een probleem met zijn factuur uit te zoeken.
11. De klant wil graag weten wat zijn SIM unlock code is.
12. De klant ervaart routeringsproblemen tijdens het bellen.

Takenlijst Ciboodle - Postpaid

Dit deel van het onderzoek bestaat uit het aanmaken van (niet bestaande) cases in Ciboodle. Hieronder vind je een aantal omschrijvingen van zogenaamde gesprekken die je met een klant voert. Het is de bedoeling dat je voor **ieder** gesprek een case in Ciboodle aanmaakt. Probeer de cases zo zorgvuldig mogelijk in te vullen (i.e. zoals je tijdens de training geleerd hebt). In ieder verplicht tekstveld kun je de tekst “test case” zetten. Je hoeft echter **geen cases toe te wijzen**. Als contact type kun je “telefoon in” kiezen. Alle cases dienen opgeslagen en gesloten te worden. Mocht je nog vragen hebben, stel ze gerust.

Alle cases kunnen aangemaakt worden onder één test klant met CTN:
0652673363 (Naam: Spice)

1. Een klant geeft aan niet te kunnen bellen en gebeld worden.
2. Er treden problemen met het toestel van de klant op.
3. Een klant heeft klachten over het ontvangen van ongewenste SMSen.

4. Er wordt een vraag gesteld over een factuur.
5. De klant heeft een vraag over de passport tarieven.
6. Er komt een vraag binnen van een klant die meer weten wil over het activeren/deactiveren van zijn voicemail via 12330/12331.
7. De klant heeft een klacht over de weergave van zijn tegoed.
8. De klant heeft problemen met zijn toestel. Je hebt de klant doorverwezen naar de betreffende afdeling.
9. De klant ervaart problemen met de netwerk dekking.
10. De klant heeft problemen met zijn SIM kaart en denkt dat deze defect is.
11. De klant heeft een vraag over het nummer van de SMS-Centrale.
12. Het ontvangen van een SMS geeft problemen.

Takenlijst Ciboodle - Prepaid

Dit deel van het onderzoek bestaat uit het aanmaken van (niet bestaande) cases in Ciboodle. Hieronder vind je een aantal omschrijvingen van zogenaamde gesprekken die je met een klant voert. Het is de bedoeling dat je voor **ieder** gesprek een case in Ciboodle aanmaakt. Probeer de cases zo zorgvuldig mogelijk in te vullen (i.e. zoals je tijdens de training geleerd hebt). In ieder verplicht tekstveld kun je de tekst “test case” zetten. Je hoeft echter **geen cases toe te wijzen**. Als contact type kun je “telefoon in” kiezen. Alle cases dienen opgeslagen en gesloten te worden. Mocht je nog vragen hebben, stel ze gerust.

Alle cases kunnen aangemaakt worden onder één test klant met CTN:

0652673363 (Naam: Spice)

1. Een klant heeft een vraag over de geldigheid van zijn nieuwe tegoed.
2. De klant heeft een vraag over een BloX.
3. Er treedt een transactie fout op tijdens online opwaarderen.
4. De klant heeft een klacht over online opwaarderen.
5. Een klant wil graag weten wat zijn PUK code is. Noteer hieronder de laatste 2 cijfers van de PUK code en maak een case aan in Ciboodle.
PUK code: ...
6. Omdat het opwaarderen via een bon niet lukt heb jij dit voor de klant gedaan.
7. De klant heeft een vraag over het aanmelden voor mobiel opwaarderen.
8. De klant wil graag naar Vodafone porteren.
9. Een klant wil graag weten welke opwaardeermogelijkheden er zijn.
10. De klant heeft een SIM unlock vraag.
11. Je hebt op verzoek van de klant een Admin Bar geplaatst.
12. Een klant geeft aan niet te kunnen bellen en gebeld worden.

9.2 Gemini: usability questionnaire

Evaluatie onderzoek Gemini

In de onderstaande lijst worden verschillende vragen gesteld die betrekking hebben op menus en functies in **Gemini**. Probeer iedere vraag zo zorgvuldig mogelijk te beantwoorden.

In hoeverre ben jij het met de volgende uitspraken eens:

	Uitspraken	Volledig Oneens				Volledig Eens
1.	De gebruikte menu opbouw is logisch en duidelijk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	Het is duidelijk welke informatie er in de 4 tabbladen van het BAN scherm te vinden is	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	Het verschil tussen het eerste tabblad van het BAN scherm en het account management scherm is duidelijk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Het memo scherm is te klein	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	Het is duidelijk dat je via rechtermuis-klik informatie uit de verschillende menuvelden kan kopiëren	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	Alle afkortingen die in memos gebruikt worden zijn duidelijk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	De meest voorkomende functies zijn gemakkelijk uit te voeren	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Het verifiëren van klanten gebeurt op een te omslachtige manier	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	Er moet een 'automatisch aanvullen' functie in Gemini komen (bv. hints bij het aanmaken van een memo)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Ik heb wel eens voorstellen voor verbeteringen gedaan aan mijn TC/TCVD/manager	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	Ik pas de grootte van het BAN scherm wel eens aan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12.	Ik maak gebruik van de help functie in Gemini	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13.	Gemini werkt te traag	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.	Het aanmaken van memos is nuttig	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15.	Ik weet dat ieder tabblad van het BAN scherm verschillende menu opties heeft	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16.	De informatie schermen (bv. BAN scherm) zijn groot genoeg	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17.	Ik raadpleeg Coach wanneer ik niet verder kom in Gemini	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18.	De omschrijving van de benodigde functies is duidelijk genoeg	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Uitspraken	Volledig Oneens				Volledig Eens
19.	Ik maak gebruik van sneltoetsen (bv. ctrl/alt + letter) om menu opties te kiezen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20.	De foutmeldingen in Gemini zijn duidelijk, en ik weet hoe ik problemen kan oplossen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21.	Ik gebruik de 'All Rows' knop om alle pending calls zichtbaar te krijgen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22.	Ik kan de klant, naar mijn mening, snel genoeg te woord staan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23.	Er moeten tijdens een gesprek te veel handelingen verricht worden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
24.	De training die in het begin gegeven wordt is uitgebreid genoeg	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
25.	Er treden vaak fouten op	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
26.	De popups die tevoorschijn komen bij het openen van een BAN zijn hinderlijk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
27.	Ik gebruik zowel Gemini als Ciboodle	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
28.	Ik zou graag een spellingcontrole in het memo scherm willen hebben	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
29.	Ik zou het handig vinden om via snelknoppen memos aan te kunnen maken (net als in Ciboodle)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
30.	Ik gebruik Gemini graag	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
31.	Ik zou graag extra training krijgen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
32.	Sommige sneltoetsen zijn verwarrend (ctrl+c) omdat ze anders reageren dan verwacht	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
33.	Er zijn te veel memo opties waardoor het lang duurt voordat de juiste optie gevonden is	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
34.	Het toevoegen van de extra 0 voor een 06-nummer is een overbodige actie	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
35.	Er moeten te veel handelingen verricht worden voordat de klant te woord gestaan kan worden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Evaluatie onderzoek Gemini - vervolg

Dit gedeelte van de vragenlijst bevat enkele praktijk voorbeelden. Verder worden er aanpassingsvoorbeelden gegeven. Deze dienen echter alleen ter *evaluatie* van de software, en zullen op het moment niet doorgevoerd worden.

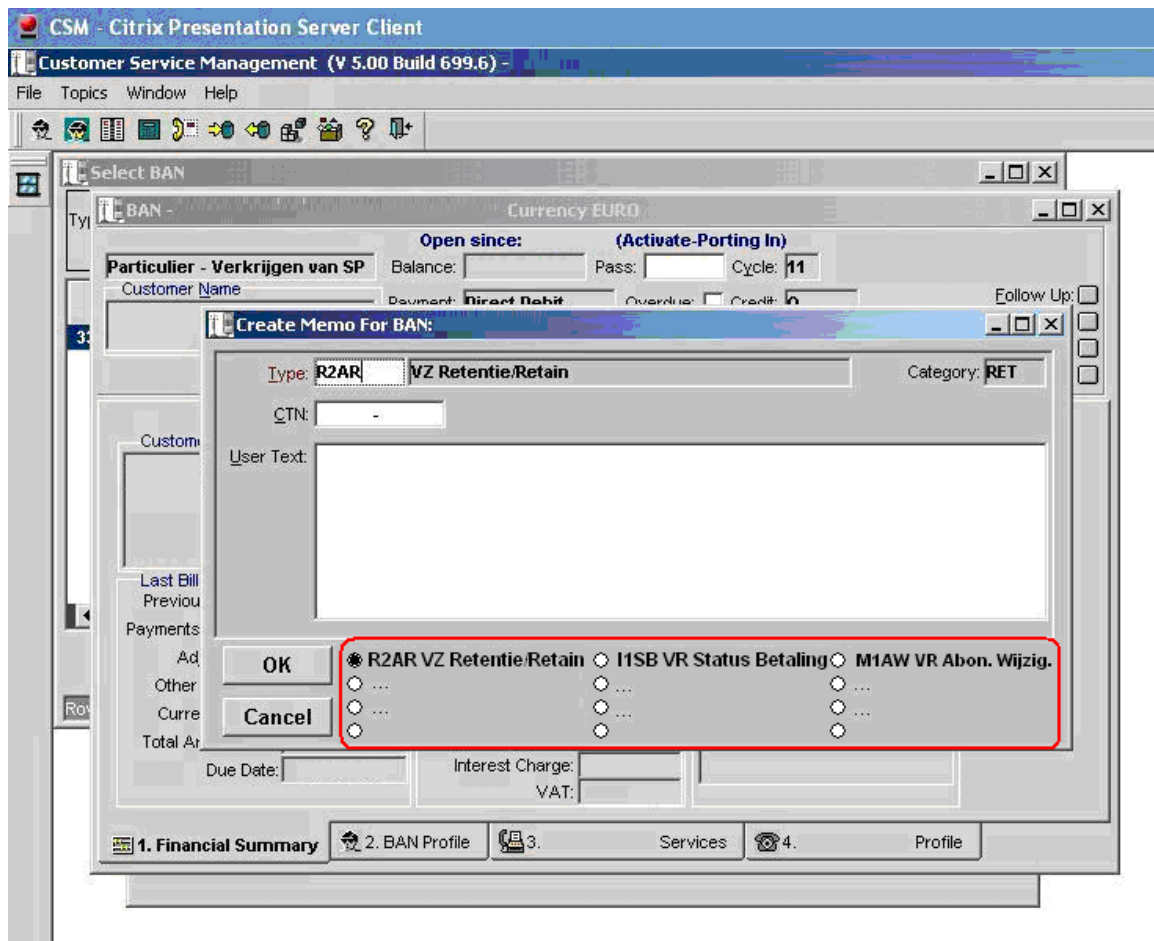
Probeer bij ieder punt een korte motivatie voor je antwoord te geven. Omcirkel het gewenste antwoord.

36. In de figuur hieronder zie je een mogelijke verbetering voor het memo scherm in Gemini. In plaats van zelf de code in te voeren kun je vaak voorkomende memos selecteren via de snelknoppen onder het tekst veld. Zou je hier gebruik van maken?

JA / NEE want

.....

.....

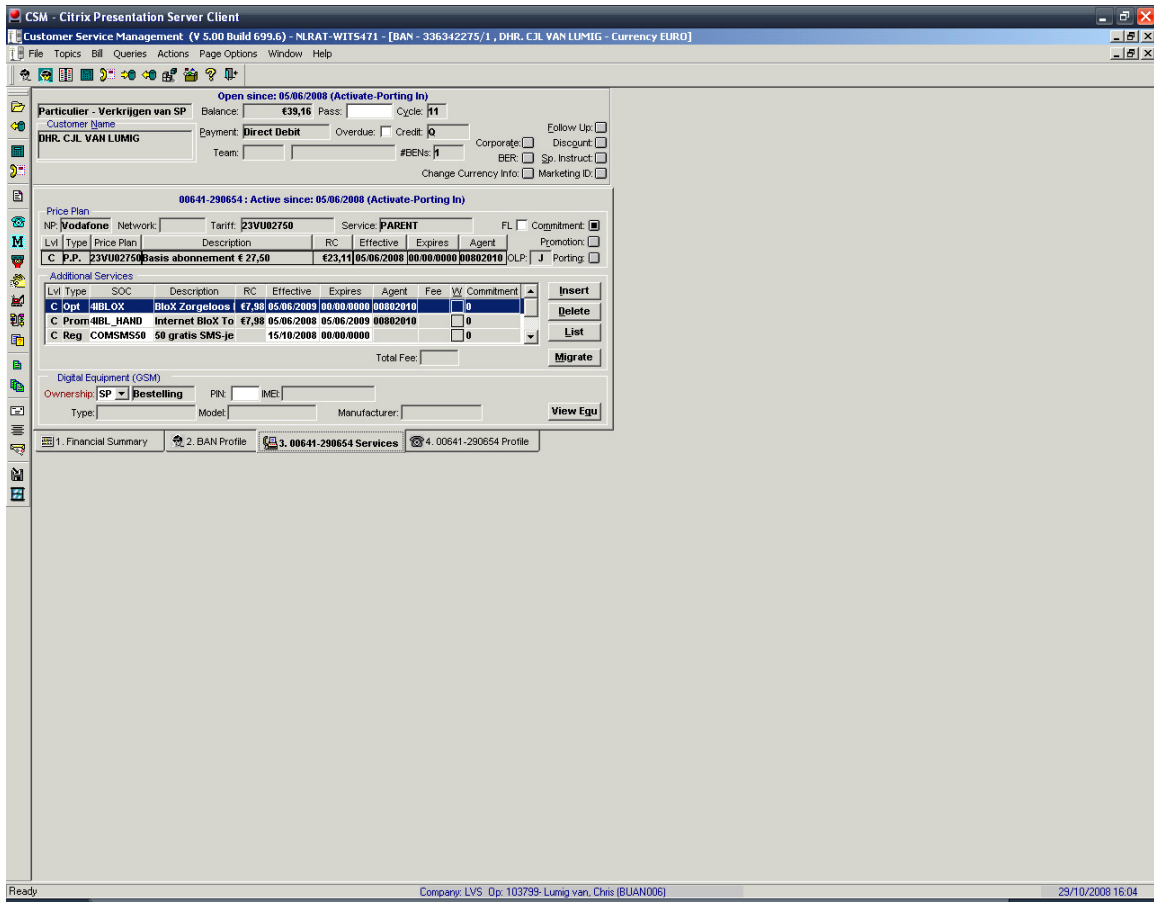


37. Wanneer je een scherm binnen Gemini maximaliseert krijg je onderstaand resultaat te zien. Heb je deze situatie zelf al eens meegemaakt? Zo ja, was dit het resultaat dat je verwachtte?

JA / NEE want

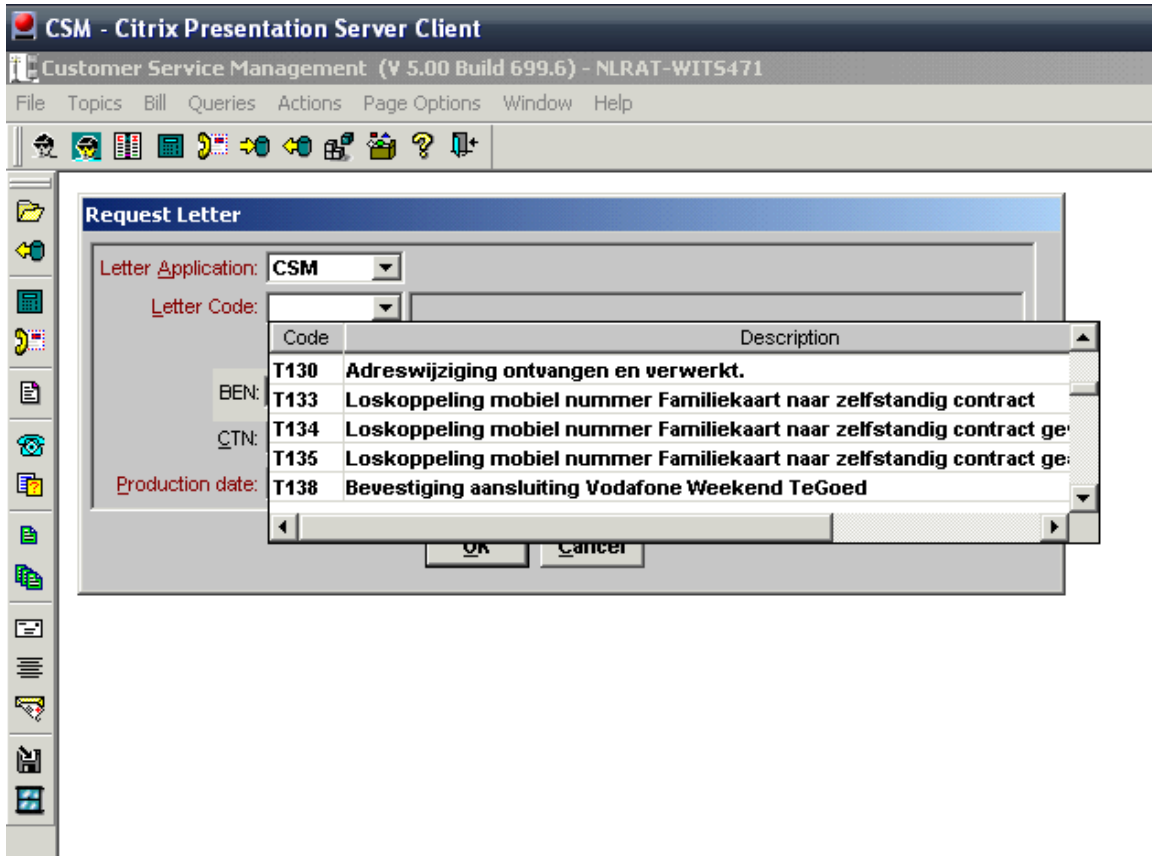
.....

.....



38. Onderstaande figuur laat een gedeelte van het “Request Letter” scherm zien. Zoals je kunt zien is de tekst niet volledig leesbaar. Denk je dat je hierdoor al eens een verkeerde brief naar een klant gestuurd hebt? (bv. brief product/dienst geannuleerd in plaats van geactiveerd verstuurd)

JA / NEE / WEET NIET



39. In de figuur hieronder zie je een aanpassing van het 1ste tabblad van het BAN scherm. Alle informatie die nodig is om een klant te verifiëren kan hierdoor in een scherm terug gevonden worden. Denk je dat deze aanpassing zin heeft?

JA / NEE want

.....

.....

The screenshot shows a software window titled 'BAN - Currency EURO'. The main content area is divided into several sections:

- Customer Details:** A red box highlights the text: 'Particulier - Verkrijgen van SP', 'MEVR. TEST SPICE', 'Avenue Ceramique 300', '6221KX MAASTRICHT', and '05/06/1970'. Other fields include 'Open since: 06/06/2008 (Restore-S-bar opheffen)', 'Balance:', 'Pass:', 'Cycle: 5', 'Payment: Direct Debit', 'Overdue: ', 'Credit: 0', 'Team:', '#BENs: 1', 'Corporate: ', 'BER: ', 'Sp. Instruct: ', 'Follow Up: ', 'Discount: ', 'Change Currency Info: ', and 'Marketing ID: .
- Last Activity:** 'CTN vrijgegeven. (, 06/11/2008)'. Below this are 'Customer Address' (Avenue Ceramique 300, 6221KX MAASTRICHT) and 'Current Invoice Details' (06 Mobiel: €0.21, Vast in NL: €0.63).
- Last Bill Information:** Invoice Number: 000100646031, Held Bill: , Date: 22/10/2008, Sts: Archived, Type: Regular, Doc.Type: Inv., Freq.: Monthly.
- Bill sent to:** MEVR. TEST SPICE, Avenue Ceramique 300, 6221KX MAASTRICHT.
- Last Bill Summary:** Previous Balance: [empty], Payments Received: [empty], Adjustments: [empty], Other Activities: [empty], Current Invoice: €62,58, Total Amount Due: €62,58, Due Date: 29/10/2008.
- Charges:** Recurring Charges: €57,04, One Time Charges: [empty], Equipment Charges: [empty], Credit Amount: €5,29, Discounts: [empty], Interest Charge: [empty], VAT: €9,99.

At the bottom, there are four tabs: '1. Financial Summary' (selected), '2. BAN Profile', '3. 00652- 3363 Services', and '4. 00652- 3363 Profile'.

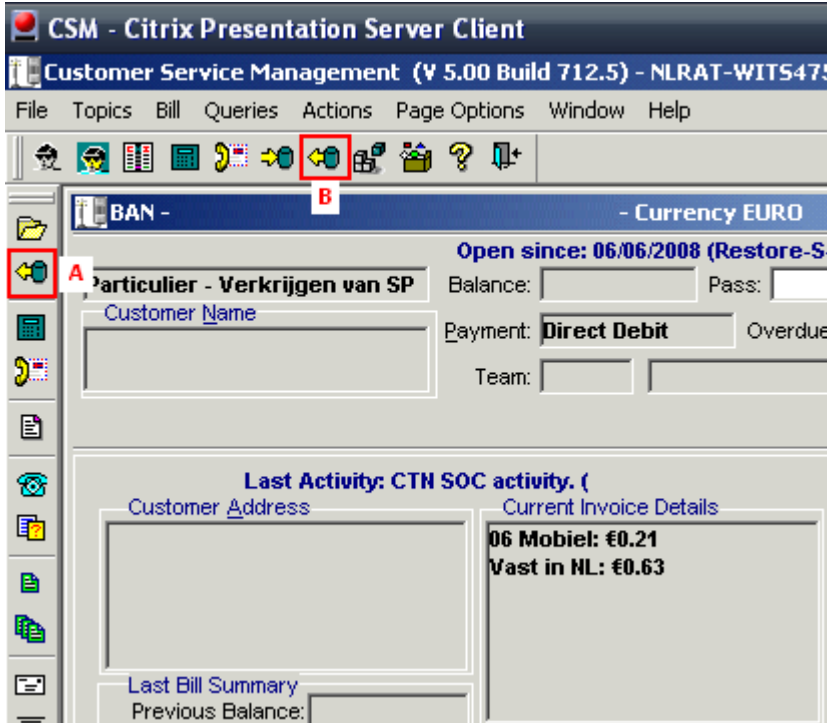
40. Weet je welke functie de onderstaande knoppen (A en B) hebben?

A

B

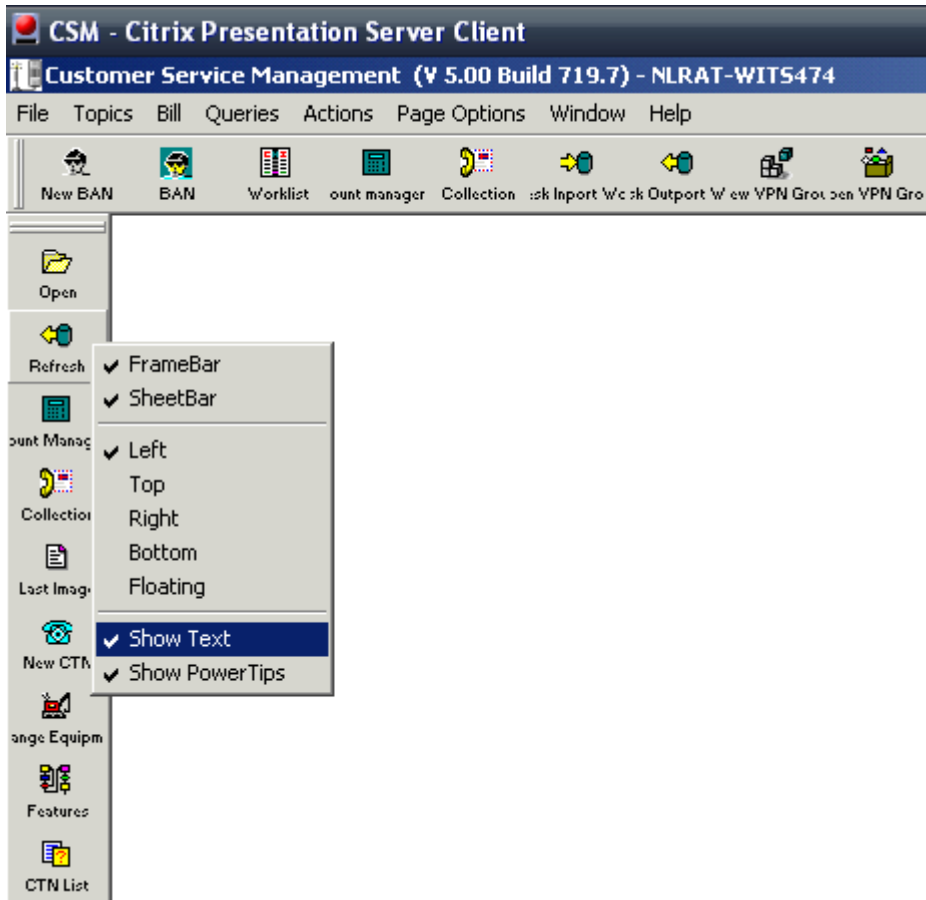
Ben je vaker soortgelijke situaties tegengekomen waarbij een icoontje/sneltoets meerdere functies heeft?

.....



41. Zou je het fijn vinden als standaard een tekst bij de icoontjes gepresenteerd wordt?

JA / NEE want



42. In het System/User Text veld worden niet automatisch scrollbalken weergegeven. Deze worden pas zichtbaar wanneer je in het tekst veld zelf klikt. Het kan zijn dat hierdoor belangrijke informatie gemist wordt. Zou je het fijn vinden als deze tekst velden groter waren zodat deze informatie direct zichtbaar is?

JA / NEE want

.....

.....

Creation Date	Type	Category	Subscriber	Created
26/09/2008 13:18:03	Tariff gewijzigd	CTN	00652- 3363 100	Ar
26/09/2008 13:14:59	Register OLP-failed	CTN	00652 3363 100	Ar
26/09/2008 13:14:59	Tariff gewijzigd	CTN	00652 3363 100	Ar
25/09/2008 14:25:56	Pers. geg. gewijzigd	BAN	- 100	Marce
25/09/2008 14:25:03	Pers. geg. gewijzigd	BAN	- 100	Marce
25/09/2008 14:17:08	Register OLP-failed	CTN	00652. 3363 100	Ar
25/09/2008 14:17:08	Tariff gewijzigd	CTN	00652 363 100	Ar
25/09/2008 13:33:16	Register OLP-failed	CTN	00615- 1949 100	Ar
25/09/2008 13:33:16	Tariff gewijzigd	CTN	00615- 9949 100	Ar
25/09/2008 12:06:22	CTN SOC activity	SRV	00615- 1949 2223 - EAI API System	
17/09/2008 16:52:38	Discount Update	BAN	- 10	Ar

System Text: **BAN Discount Information insert: CTN=, Discount type = Regular, Price Plan/SOC = APN020, Charge type = UC, Percentage=100,00, Effective date = 17/09/2008. Open Ended Reason = DISFX2.**

User Text:

Rows Retrieved: 454 out of Total: 454

System Text: **BAN Discount Information insert: CTN=, Discount type = Regular, Price Plan/SOC = APN020, Charge type = UC, Percentage=100,00,**

User Text:

System Text: **Price Plan/SOC = APN020, Charge type = UC, Percentage=100,00, Effective date = 17/09/2008, Open Ended Reason = DISFX2.**

User Text:

43. Tijdens het sluiten van een BAN scherm komt onderstaande melding tevoorschijn. Weet je waardoor deze veroorzaakt wordt, en hoe je het probleem kan oplossen?

JA / NEE namelijk

The screenshot shows a Citrix Presentation Server Client window titled "Customer Service Management (V 5.00 Build 719.7) - NL RAT-WITS468". The main window displays account management information for "MEVR. PKA VOORNEVELD". A modal dialog box titled "Close BAN (#6900194)" is open, showing an error message: "In order to close this BAN, you must close all the associated open windows." The dialog has an "OK" button. The background window shows various fields for customer information, payment details, and a table of charges.

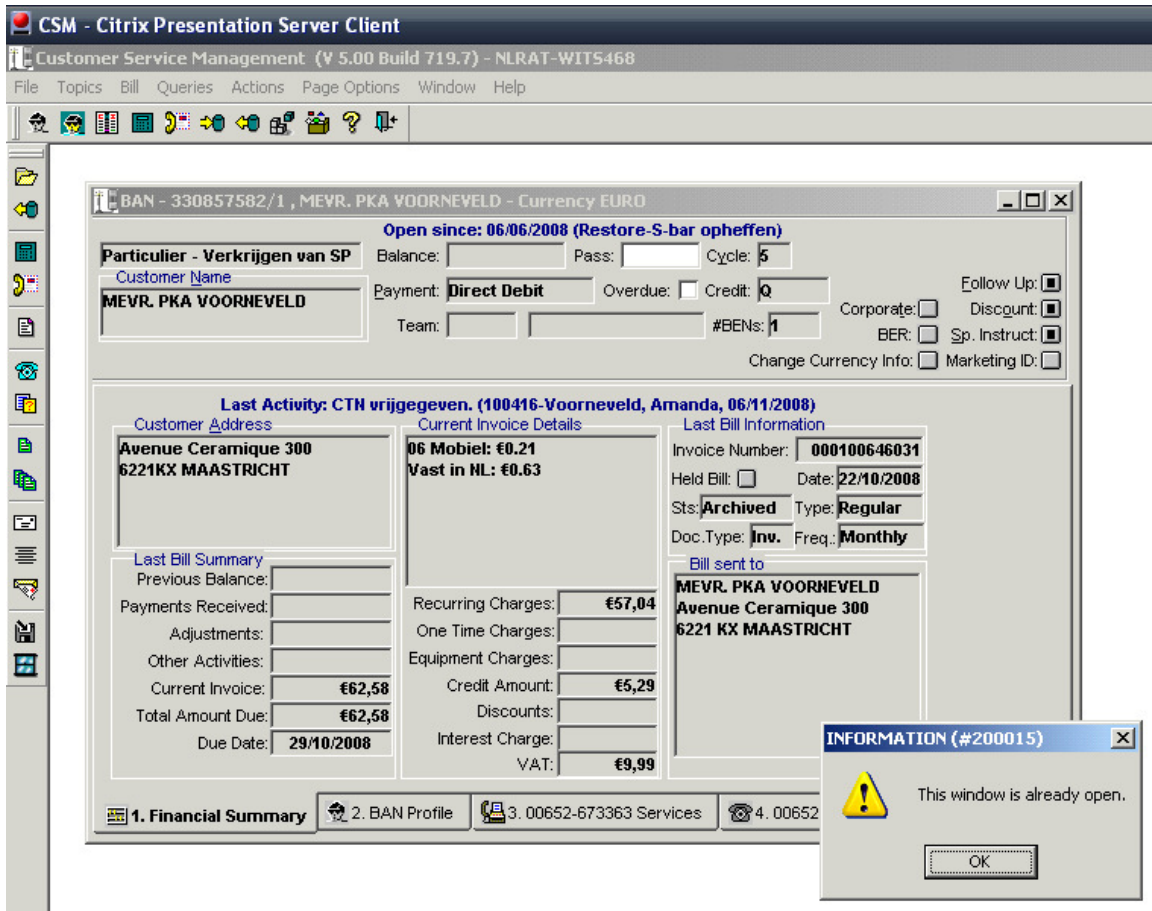
Last Bill Summary	
Previous Balance:	
Payments Received:	
Adjustments:	
Other Activities:	
Current Invoice:	€62,58
Total Amount Due:	€62,58
Due Date:	29/10/2008

Current Invoice Details	
06 Mobiel:	€0.21
Vast in NL:	€0.63
Recurring Charges:	€57,04
One Time Charges:	
Equipment Charges:	
Credit Amount:	€5,29
Discounts:	
Interest Charge:	
VAT:	€9,99

Last Bill Information	
Invoice Number:	000100646031
Held Bill:	<input type="checkbox"/>
Date:	22/10/2008
Sts:	Archived
Type:	Regular
Doc.Type:	Inv.
Freq.:	Monthly

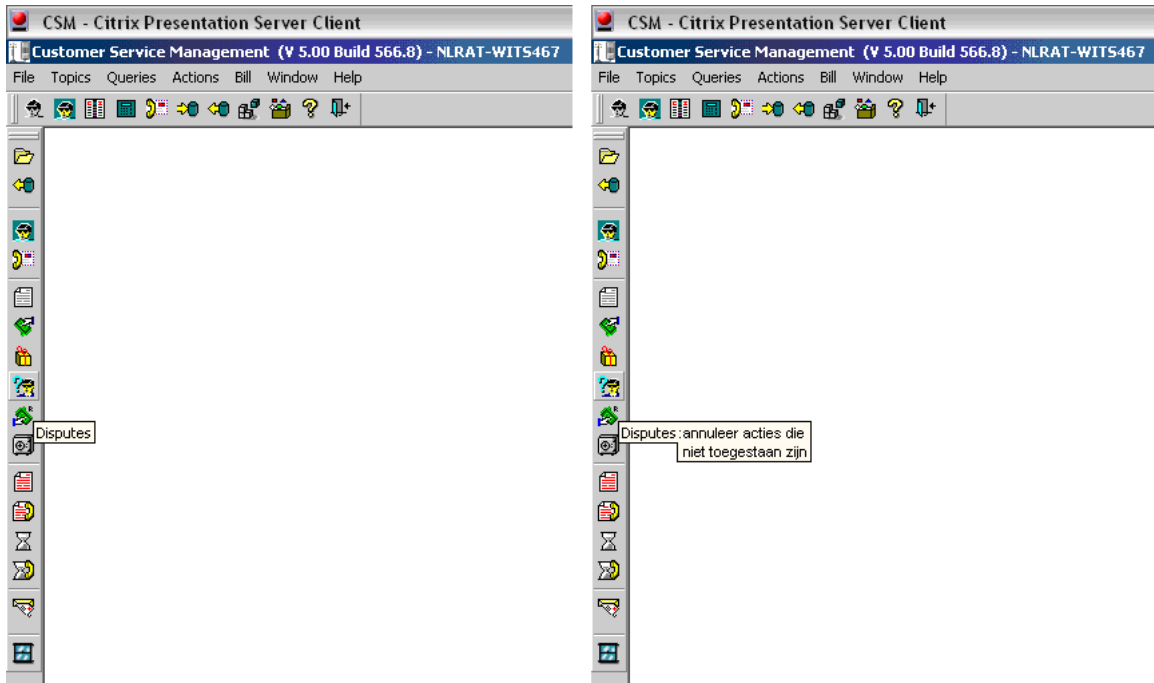
44. Je was juist bezig met het aanmaken van een nieuwe memo, maar je moet eerst nog wat gegevens opzoeken in het BAN scherm. Na de informatie opgezocht te hebben klik je weer op “Actions -> New Memo” (SHIFT+M). Als gevolg krijg je onderstaand beeld te zien. Nadat je op “Ok” klikt krijg je het BAN scherm weer te zien. Herken je dit probleem, en hoe los je dit normaal op?

JA / NEE oplossing



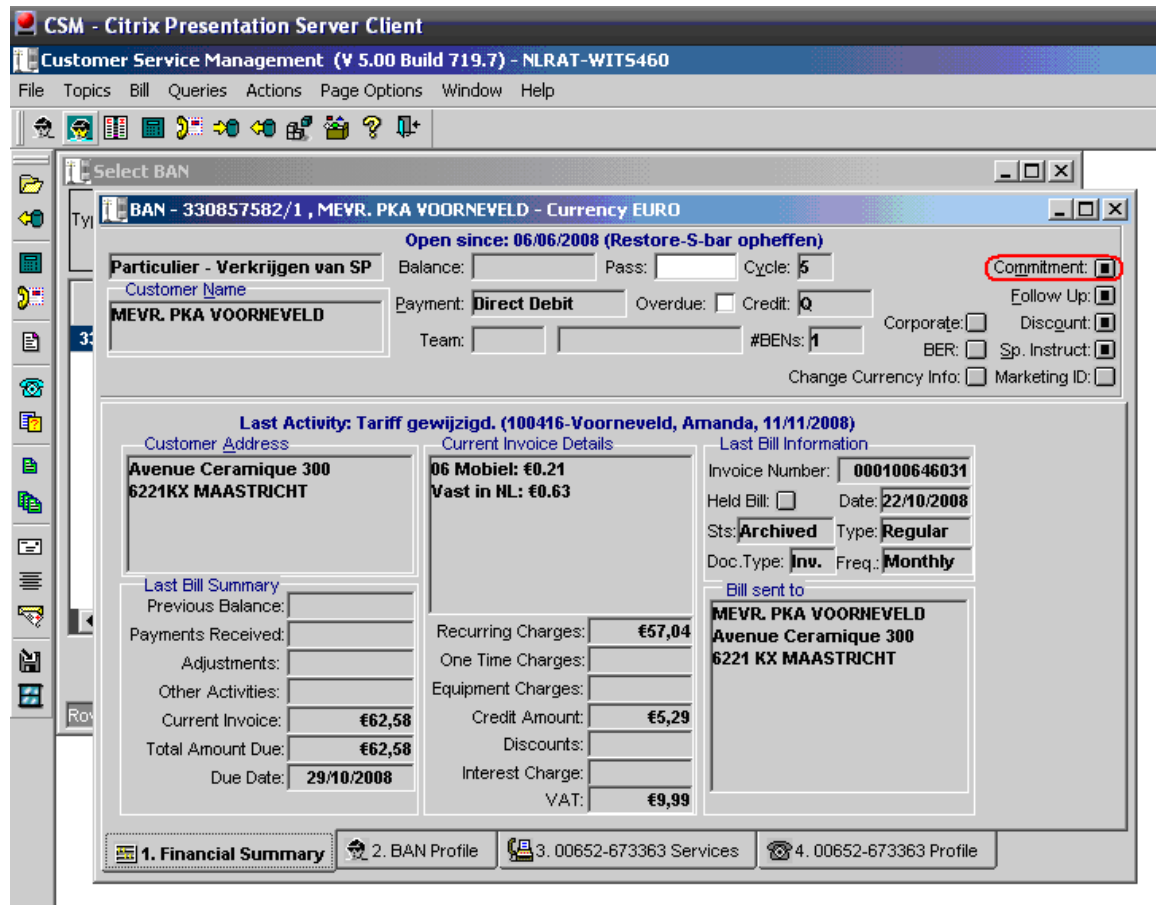
45. In de figuur hieronder zie je het resultaat wanneer je met de muis over een knop beweegt. Links is de oude situatie afgebeeld en rechts wordt een mogelijke aanpassing weergegeven. In plaats van slechts de functie weer te geven wordt er tevens een korte omschrijving toegevoegd. Helpt dit om functies duidelijker te maken? Zijn er eventueel nog andere voordelen?

JA / NEE want



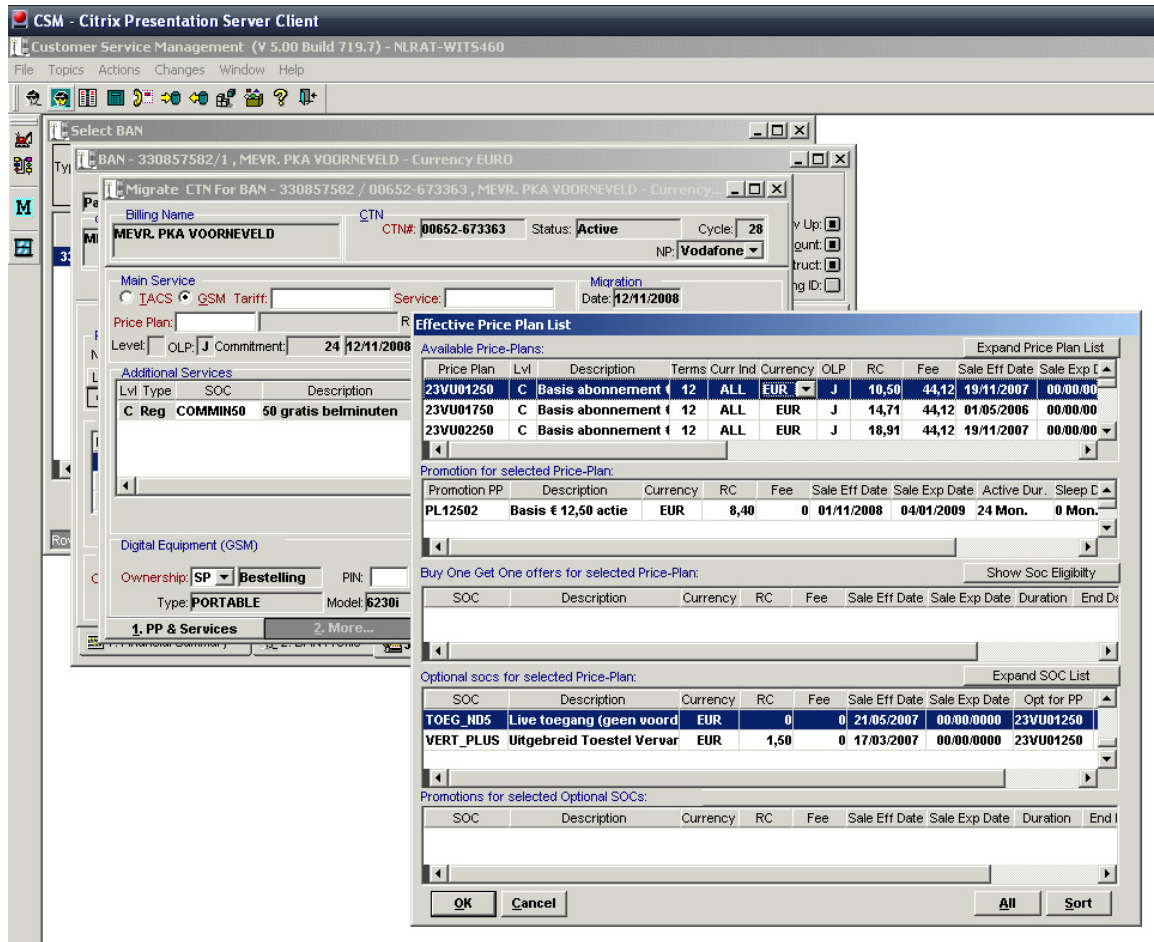
46. Klanten stellen geregeld vragen over de einddatum van hun contract. Momenteel kun je deze informatie vinden in het derde tabblad. In het scherm hieronder is de 'commitment' indicator verplaatst naar het eerste tabblad van het BAN scherm, waardoor je snel kunt zien of de klant al kan verlengen. In hoeverre denk je dat dit een verbetering is?

GEEN / KLEINE / GROTE VERBETERING want



47. Wanneer je de abonnementsvorm wil veranderen wordt standaard de hele lijst van abonnementen getoond. Je kunt echter normaal maar naar een beperkt aantal prijsplannen migreren. Verder kun je in onderstaand scherm niet zien welk prijsplan de klant momenteel heeft. Zou het beter zijn om standaard alleen de mogelijke opties te tonen, en eventueel via een aparte knop alle abonnementsvormen zichtbaar te kunnen maken?

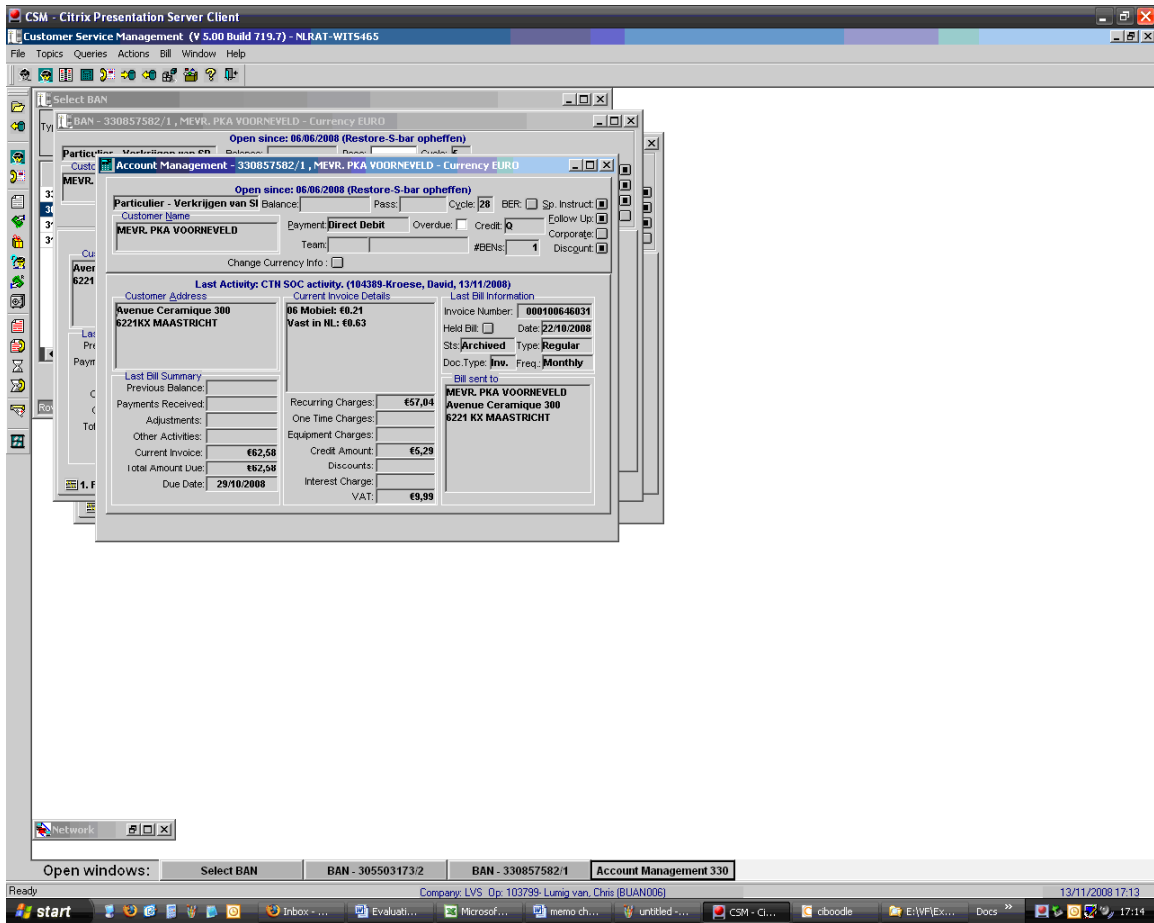
JA / NEE want



48. Als je veel schermen tegelijkertijd geopend hebt kan het voorkomen dat je het overzicht kwijt raakt. Zou je het fijn vinden als je gebruik kon maken van een taakbalk (zoals in Windows)?

JA / NEE want

.....

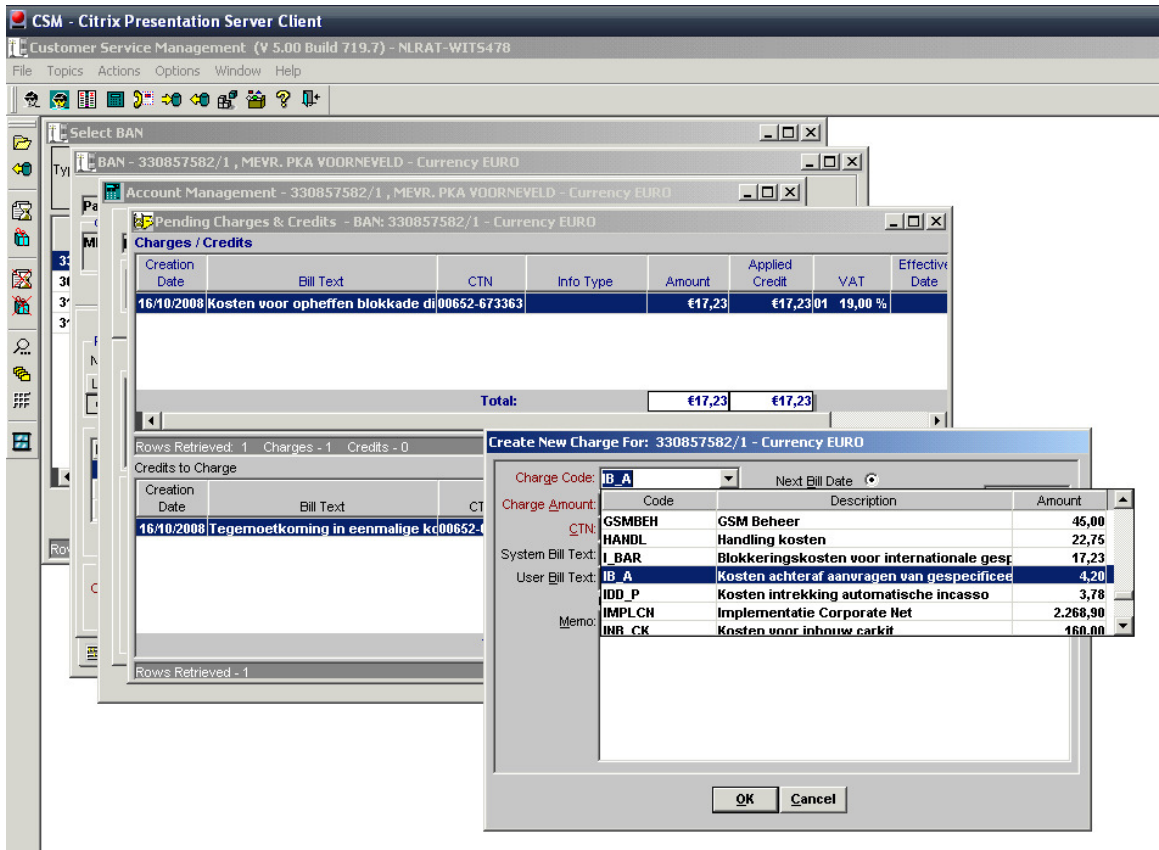


49. Wanneer je een gespecificeerde rekening naar de klant opstuurt moet je de kosten hiervoor achteraf handmatig in rekening brengen. Is het beter als deze optie selecteerbaar is tijdens het versturen van de rekening?

JA / NEE want

.....

.....



50. Tijdens een gesprek vraagt de klant vast wel eens dingen die jij zelf (nog) niet weet. Om de klant toch te kunnen helpen zal je waarschijnlijk op Coach kijken, of even gauw een collega vragen.

Geef op de schaal hieronder aan in welke verhouding je gebruik maakt van coach/collega's voor het verkrijgen van de informatie:

Voorbeeld: "Ik kijk zelden op Coach, maar vraag mijn collega's om hulp"

100% Coach o o o o ● o 100% Collega's

100% Coach o o o o o o 100% Collega's

Hieronder kun je op- en/of aanmerkingen over beide software programma's kwijt:

Ciboodle

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Gemini

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Ten slotte kun je aangeven wat je van dit onderzoek vond:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....