JAVA-TOOLS THAT SUIT ERICSSON TELECOM AB

Anette Hermansson

Göteborg May 1998

Master Thesis

Examensarbete II

Abstract
This master thesis concerns with finding a tool, which generates Java-code. While choosing a tool it is important to know who is using it and which requirements there are on a tool. To get to know that I have made a questionnaire among people in the organisation and some of them have been interviewed. To learn which functionality the different tools offer, the market has been investigated. To learn this I have read a lot about the different tools. It was also necessary to talk and write to the vendors.

With the questionnaire and the interviews as basis a requirement list has been made. The requirements in the list are put in three categories: Definite requirements, Not that definite requirements and Wishes. Among the definite requirements there are requirements about being a whole IDE (integrated development environment), requirements concerning how the organisation works, requirements about the generated code by the tool and requirements about help function and support.

For the reader having a survey of the tools on the market a lot of them are described with advantages and disadvantages. Also a judgement for matching the tools with the requirements has been done.

There are a number of tools, having fulfilling the most absolutely requirements from the organisation. Jbuilder from Borland, Parts for Java from Objectshare, Visual Age from IBM, PowerJ from Sybase and Visual Café from Symantec are five candidates.

Which tool is preferred depends on the priority point by the person making the decision. Some people in the organisation want both visually building of events and the possibility to make changes in debug mode (which is to be found among the Not that definite requirements). Therefor I recommend a tool which fulfil this and the choice will be to evaluate Visual Café from Symantec.

Denna magisteruppsats handlar om att hitta ett utvecklingsverktyg för Java som passar Ericsson Telecom AB. När man ska välja ett utvecklingsverktyg är det viktigt att veta vem det är som ska använda det och vad de personerna har för krav på verktyget. För att ta reda på detta har jag låtit en enkät gå ut i organisationen och några personer har blivit intervjuade. Jag har vidare undersökt marknaden för utvecklingsverktyg för att ta reda på funktionaliteten hos de olika verktygen. Metoden för detta har varit att läsa om dem men även ringa och skriva för att få reda på det som är väsentligt.

En lista med krav har framställts ur resultatet från enkäten och intervjuerna. Kravlistan består av krav på tre nivåer: Absoluta krav, inte så absoluta krav och önskemål. Bland de absoluta kraven finns krav som har att göra med att verktyget är ett helt IDE (integrated development environment), krav som har att göra med hur organisationen arbetar, krav på den av verktyget genererade koden och krav gällande hjälpfunktionen och support.

Vilka verktyg som finns på marknaden, deras fördelar och nackdelar samt en bedömning över hur väl de stämmer överens med kravlistan finns redovisat för den intresserade läsaren.

Det finns ett antal verktyg som uppfyller de mest absoluta kraven från organisationen. Jbuilder från Borland, Parts for Java från Objectshare, Visual Age från IBM, PowerJ från Sybase och Visual Café från Symantec är fem kandidater.

Vilket verktyg som föredras beror på vilka prioritieringsgrunder personen som väljer verktyg har. Några människor i organisationen har sagt att de skulle vilja ha tillgång till visuell länkning av händelser och ha möjlighet att göra ändringar i debugläge (vilket återfinns bland de inte så definitiva kraven). Min intention har varit att hitta ett verktyg som erbjuder detta och rekommendationen blir Symantecs Visual Café.

I would like to thank Stefan Åberg for initiating this work in a very short limit of time and for the interest he has shown for it.

I also would like to thank Thomas Kronberg, who has acted as my instructor for all the help and answering to all my questions.

Kari Wahll has been the instructor at the University of Gothenburg. I would like to thank her for the thoroughly reading and the thoughts about the work, which I have received from her.

There are also many other people in the organisation, especially the respondents to the questionnaire and the respondents to the interviews who have taken time to help me raising quality of this work. Thanks to all of you, who have in some way giving me advice!

INDEX

1 INTRODUCTION

1.1 BACKGROUND

This survey was initiated from Ericsson Telecommunication AB. At Ericsson there is an increasing demand for developing in Java. More and more the customer wants to have web-based applications. Systems for telecommunication are executed in distributed environments and there are different operating systems; Windows and Sun Solaris. Java is a suitable language in distributed environments because it is machine independent. So far Java is the only commercial cross-platform language.

Java is the newest technology and some developers want to develop with Java. So to keep the competent staff and to recruit new ones to the company Java is interesting because Java is new and exciting.

Up to now it has been little use in some Java development tools at Ericsson. The most used one is jdk 1.1.4. Some developers have been using Visual Café 1.0, Java workshop 1.0 and Visual J++. None of these were satisfying.

1.2 RESEARCH AREA

There is a need to know if there are development tools that are better and therefore can suit the business. To learn this it is important to know how the organisation works and which demands there is on a tool.

The question is which development tool is to be chosen. The answer is considered with the functionality offered by the tool and taste about the design of it. While choosing a tool there must be some criteria to look after. The criteria must come from the organisation, which is going to use it. So the organisation and its demands is one part of the research area. An important thing before choosing a tool is to get to know how people are working, how they think and act in the actual enterprise.

Today there are quite a lot of companies jumping into the era of Java and starting selling development tools for Java. Often the products are streamlined to earlier tools already developed for other languages.

Many companies are trying to raise efficiency and work faster. The companies selling the development tools have started to compete with promising the tool that will make the developer develop fast. New products are entering the market constantly because Java is hot. The products are the other part of the research area.

1.3 SCOPE AND LIMITATIONS

Developing systems and writing programs is a large area. The whole system development process from pre-study to test is to be done to get a system or program on the market.

Although it is interesting to have a tool covering the whole development process the primary focus will be on the programming phase. To look at the possibilities of having tools supporting the work in other phases of the development process and to integrate them with the other selected tools is of course interesting.

As the work is performed often the same persons work through the entire project. There are demand for more or less people in the different phases, which leads to that some people only join for example one or two phases and other join the whole project. In other words there are no people just working in the coding phase every time a new project is going on. This essay concern with the coding phase and the word programmer will be used for a person writing code.

1.4 PURPOSE

The purpose of this work is to recommend a Java development tool, which suits the enterprise. The following questions are accurate:

- Which requirements does the business have for a Java development tool?

To answer this question it is necessary also to answer to the following questions: How do they work? How do they act together? Which functionality is necessary? Which facilities are nice to have?

To answer to the questions a questionnaire was distributed among the employees and some of the employees have been interviewed.

Another question is if the requirements depend on the task a person is performing. Are the requirements not the same if you are programming user interface as if the task is programming logic parts or communicational parts of the application. The hypothesis is that they are not exactly the same. Therefore it is interesting to separate the programmers into groups and see what the different groups answer.

- Which tools do fulfil the requirements?

To answer this question and to find the best tool, the market must be investigated. Some tools must be investigated thoroughly and some others could be abandoned on an early stage of the investigation. The tools abandoned are the tools not fulfilling the most important requirements.

Finally there shall be a recommendation of a tool. The hypothesis is that at least one tool will be suitable. It is also possible that different programmers have different demands and therefore the chosen tool suit some programmers better than others. For example it can suit programmers writing user interface code better than those programming communication units.

1.5 DISPOSITION OF THE ESSAY

The outline of this master thesis is as follows:

Chapter two contains of some background on Java development tools.

In chapter three, Method, the method for doing this survey is explained.

Chapter four concern business requirements on a development tool. The result from a questionnaire and interviews will be presented and there will be a list of the requirements on a development tool for Java.

In chapter five some Java development tools will be presented. Facts about the tools and a judgement about the tool will be presented.

Chapter six contains the result from chapter four and chapter five. First there will be the requirements set up as points and after that some tables will be presented indicating how the tools match the requirements. Finally there will be a recommendation for further evaluating and a recommendation.

Finally in chapter seven, the work as a whole will be discussed and the conclusion and the result will be presented.

2 FRAME OF REFERENCE

2.1 SOFTWARE DEVELOPMENT

To create software concerns with the whole development process. The development process consists of pre-study, design-phase, coding and testing. In the coding phase there are mostly three major things that have to be done to let the program work. One thing is to write the code for the user interface, another thing is to write the logic code and the last thing is to write the code for communication. In a very small application it is probably one person writing the whole code. In big applications there are of course many people involved and sometimes one person only does one of those three things.

Software development is done both in distributed environments and in standalone machines. A machine, which stands alone, is either a PC or a mainframe computer. When developing applications to run in a standalone machine all the different code mentioned above has to be put at the same place. In a distributed environment it is different, there will be no more than two layers at the client side and two or three at the server side. A layer is either code for communication, code for logic programming or code for the user interface. At the client there is code for doing the user interface and communication with the server. At the server there is code for logic and code for communication with the client and code for communication with a database.

2.2 DEVELOPMENT TOOL

The development tool is the environment that the programmer is using while writing code. But the word “development tool” does not necessarily associate to the whole environment. A development tool for Java can for example contains classes of Java Beans. A debugger or a compiler is a development tool as well. A number of tools together build up the whole environment. An integrated development environment (IDE) is a more complete development tool. It shall at least contain an editor, a compiler, a debugger and a hierarchical browser.

A modern development tool is visual. It means that the programmer is able to see more on the screen than in the old days. It is optional to use menus and simply click somewhere to get the program compiled or to run it instead of writing down commands in an empty window. But there are shortcuts for the trained person who does not want to walk through many menus.

It is also possible to develop visually without coding and instead draw the user interface using the drag and drop technique. The palette is the place from where the buttons and so on is dragged. While doing so the code will be generated.

User friendliness of the IDE is important. In this case the programmer is the user.

2.3 RAPID APPLICATION TOOL

A rapid application tool (RAD) is a tool, which enhances the speed of application development in comparison to the traditional environment. Whether a tool is a RAD or not is point of discussion between vendors. Objectively it shall at least contain a visual form designer to be called RAD.

One way to get rapid application is to -in the programming phase of a project- use a powerful tool. “Rapid application development tool is a term for tools that are made to create applications in a shorter time. To shorten the development time reusable components, object-oriented language and a well laid-out IDE are important” [18]

A development tool, used to develop fast, is based on a component model where the components are reusable. “ A software component is a pre-built piece of encapsulated application code that can be provided with other components and with hand-written code to rapidly produce custom-applications.” [13] While drawing the user interface using the drag and drop-technique the programmer is using reusable components. The tool will use components to generate code. So the term “generated code” means that it is code auto-generated by the machine, and the term “reusable code” refers to the prewritten component, which is used to do the generation.

Mostly the components of the tools are Java Beans. Sun and IBM developed that technique. There are other ways to get components. Microsoft’s components are named ActiveX.

2.4 EVENT PROGRAMMING

One way to get code rapidly is to get event handlers generated as well. Earlier tools did only support user interface to be drawn while using reusable components. Then the code for event handling must be written. Now there are quite a lot of tools supplying having the event handlers generated.

Sometimes events involve several components. For example a push on a button shall raise event by another component. Some tools take care of this too and generate code.

2.5 THE MACHINE INDEPENDENCE OF JAVA

The language Java is developed by Sun, in the beginning secure imbedded products. Later it became a programming language for Internet and for solid applications, mostly in distributed environments. First it was Internet that was the interesting thing because that Java works cross-platforms and therefore could be executed within different web-browsers. Now as the language and the development tools are more mature, there is interest in using them in distributed environments, for the same reason, machine independence.

Machine independence means that the program once written on one machine is executable on another, and it shall behave and appear in the same way. If this is true the vendors have to follow the standard worked out by Sun. “Sun Microsystems Inc.’s Java is the de facto standard platform for network computing.” [23]

The source code that the tool generates must have a good structure. There is documentation from Sun about how the code should look like and there is also possibility to check if the code leads to programs that do what they should.

To translate and execute the Java byte code is the work of the virtual machine (VM). The VM consists of a just-in-time compiler, an interpreter and a runtime machine. The machine independence of the translation is a matter of how the VM works. There are two possibilities: either a vendor can create an own compiler and have it licensed from Sun or a vendor can choose to use the compiler integrated in jdk (Java development kit, which is an environment for developing in Java made by Sun). It is not only the developers’ virtual machine that matters, but also users virtual machine will matter. “Keep in mind though, that it’s the implementation of the Java runtime environment on the receiving system that determines your applets performance in the eyes of the end user. [22]

3 METHOD

This investigation is in two parts. The first part is to get to know what requirements the business has on development tools. The second step is to choose one from the market, which seems to suit according to the requirements and recommend that one.

3.1 METHOD CHOICE

The method choice is concerning with purpose, experience and resources. The resources can amongst others be time.

For the first part (which is to get to know the requirements of the business) the best thing is to combine two methods: questionnaire and deep interviews. The reason for this is that it is important to learn something generally and deep.

Getting to learn something generally indicates that many people are going to answer. It takes a lot of time to interview many people and it takes even more time to treat a great amount of material. If people can answer on their own when they have time it will go much faster than if an interviewer must meet everyone. This is the reason for doing a questionnaire.

The most important thing in this investigation is to get to know the requirements well. Which requirement people have is very much a question about taste and what they like. If a tool is suitable or not will at first concern desirable functionality. When there are several tools that fulfil the functionality requirements, the choice will concern with details. To know about details and what people like, the best is to ask them. They have already got questions in the questionnaire. But even though there are quite a lot of questions there could be more details that are important. There are probably things that people do not think about at first. Another thing that can happen in answering a questionnaire is that the respondent misunderstands some of the questions. To get deeper and to know that the answers not will be misunderstandings it is important to ask people in an interview.

The second part is about what is on the market and if there is a tool which fits the requirements.

To get to know what's on the market Internet is a good place to look at. All vendors have information about their products on the www. But how informative it is depends. So there will probably be need for calling and mailing to get to know special issues. To get a neutral idea about the good points of the tool it is necessary to read reviews about the tools. That could also be found on the Internet. The tool must fulfil important requirements. Otherwise the tool is not interesting.

Another way of doing such a survey or rather a complement could be going on demonstrations by the vendors. It could be a good alternative because it will be more

visual to look at it than just reading. In this case it is no alternative, because it will take more resources than is available. It will take more time and that is one of the limited resources.

3.2 PARTICIPATOR

To say something generally the best thing is to learn what every single person is doing and thinking about software development tools. So everyone who has the opportunity to answer is going to participate in the first step, which means that they are going to answer a questionnaire.

In the second step, in-depth interview, there are few participators. After the questionnaire is answered the whole population is divided into three groups because of which type of programming they are working with. One group is the programmers coding the user interface, another group is the programmers writing code for communication units and finally there is a group, which codes the logic. One person can be a participator of more than one group if he or she is doing more than one thing. Because the questionnaire has told the generality it is not important to ask many people. It is better to just let very few answer, and give them time to tell what they want. So if there are four persons who have time to participate it should be enough. The interviewer is interested in the persons who have answered in a common way and have experience from integrated development environment. Experience from Java is also interesting.

3.3 PROCEDURE

3.3.1 The questionnaire

3.3.1.1 The design of the questionnaire

The first thing that is important about the questionnaire is to get as much answers back as possible and the second thing is to treat them. So if the questionnaire is worked out with questions that could be answered with choosing among written alternatives it will be faster for the respondents to answer and probably easier to get them back in reasonable time. It is faster to treat. It is possible to look for patterns and see what the patterns tell.

After that the questionnaire will be worked out as statements, that people have to respond whether they agree or not. They have to choose one of four opportunities. The hope is to prohibit the ”middle-way-answer” while doing so. The four possible answers are that they think that the statement is a requirement, that it is a wish, or it is not that important or at last it is not necessary. See appendix A.

3.3.1.2 The questions

The questionnaire must tell about the respondents, which experiences they have. It shall also tell about the requirements they do have on development tools. So the questions is

first about which tools they have experience with, if they have experience with Java and which kind of programming they are doing.

After that there are many questions about how they want a development tool to look and behave. The questions are grouped into headings; contents of the tool, fault-localisation, development of the graphical interface, communication, help function, applet development, manual and other questions.

3.3.1.3 Treatment of the questionnaire

While treating the questionnaires a good question is what they are telling about the requirements. Which patterns are we looking for?

If half of the respondents say that one thing is a requirement this indicates that it is a requirement for the group as a whole. If the whole group is saying that it is either a requirement or wish it will also be treated as a requirement.

When very few persons say that they have a requirement and it does not bother other persons the differences can be explained of differences in their works. A requirement from a single person can be crucial to his or her work. If that is so and it will become a requirement for the whole organisation has to be figured out in the interviews.

Another question is if the requirements depend on what you are doing. Are they not the same if you are programming user interface as if you are programming logic parts or communicational parts of the application. The hypothesis is that they are not exactly the same. Therefore it is interesting to separate the programmers into groups and see what the different groups are answering. The material in the answers from the questionnaire will be treated in separate tables.

3.3.2 Interviews

3.3.2.1 Questions in the deep interviews

To get to know more about the requirements there shall be in-depth interviews. To know about people's feelings about a good tool is of interest. To get to know about the feelings of what is a good tool there have to be open questions. "When doing qualitative interviews questionnaire with prewritten questions is not used… Instead a list of question areas is to be written." (Freely translated) [2]

From the beginning the interviewer could try to only give one broad question such as:

- Which requirements do you have on a development tool?

If it is difficult for the respondent to answer one broad question, here are some suggestions to question areas:

- The overall impression about the visual tool, what the programmer would like to see on the screen.

- Content of the tool.
- How working with the tool.
- How to program a graphical interface.
- How build the event handlers.
- Fault localisation.
- The code generated by the tool.

What this investigation is about is to be able to choose a tool. So what is interesting to ask about must be concerned with which opportunities the tools on the market are able to offer. So it is interesting to know what the programmers like about some facilities. These questions are not so open. The answers to these questions shall tell very exactly how important different facilities are.

3.3.2.2 Treatments of the questions

The answers from the questions lead to a requirement's list. The answer shall tell what is important and what is likeable and suitable.

3.3.3 Tool investigation

The basis is the requirement list. The requirement list is written in three categories. First category is the most important requirements, although some are more important than others this category must show what is not able to manage without. These requirements should be fulfilled. Therefore it is important to get to know them and it could be necessary to write or speak to the vendor, even if the starting point is to read about them. Reading reviews will help finding out how good or bad a tool is. And if there is any customer satisfied with the product is also a good thing to know. If it could be seen in an early phase of reading about a tool that something important is missing, just leaving the tool behind without any further investigation is the right thing to do. So the amount of text about the tools is probably going to differ.

4 THE REQUIREMENTS ON A DEVELOPMENT TOOL

4.1 THE QUESTIONNAIRE

4.1.1. The respondents

The questionnaire where sent out to 70 persons. The expectation was not that everyone but perhaps half of them were going to answer. 16 people answered to the questionnaire. The whole population of developers at ETX /A/B (the actual department at Ericsson Telecom) is 70. Of this 70 there were 16 who has the opportunity and interest in answering. So 16 people are the whole population of the survey.

5 of the persons have experience with Java and 11 of them have not. 8 of the respondents have worked at Ericsson more than 5 years, 3 of them have worked there more than 1 year and five of them less than 1 year. 7 of the persons have worked within IDE's and 9 have not. The IDE's mentioned is Symantec Visual Café, Visual J++, Visual C++, Borland C++ (probably not at Ericsson), UIM/X and Visual Works.

In the questionnaire there was no question about how long people have been working at Ericsson, and it was a mistake not asking it. So the question has been asked afterwards. In appendix B the totally result from the questionnaire is to be found.

4.1.2 The answers from the whole group

Spontaneously looking on the answers indicates that there are two things that are important. That is that it must be easy to reach the source code from anywhere in the IDE and that it should be possible to modify the code if necessary.

The answers interesting are those where 8 or more say that it is a requirement, and those where all the people but one say that it is either a requirement or wish. The one person is the margin of error. There must be room for one of 16 to understand the question in another way than the expected for example.

Treating the answers in this way gives that at least eight persons want that:

- The tool must have an integrated editor.
- It must be possible to undo the last thing done.
- It must be easy to reach the source code from anywhere in the development tool.
- The debugger must be able to do several things such as run, breakpoints, single step and look for values in variables.
- The user interface should easily be drawn and there must be ability to modify the code by oneself and keep it while regenerating.
- The tool must support data base handling and it should be able to get code from other files.
- The help function must contain a search function, examples and must be on-line as well.

- It must be possible to save own classes to export them to the next program the programmer writes.
- It is important that the tool supports learning by doing by the user.
- The result must be fully machine independent.
- The technical support must be without complaints.

Adding the statements that the group told that it was either a requirement or wish gives that there are several more important things while looking at development tools:

- It must be visual and have an integrated debugger.
- Which parts there are in the tool should be easily understand.
- It must be possible to get hierarchical information about the classes you do of your own and the classes owned by the tool. These should be easily understood which they are.
- The tool must support both relational databases and object oriented databases.
- It must be possible to keep classes written by the developer in another program later on. There shall also be possibility to import Java classes from other suppliers.
- The debugger must be visual.
- While drawing the user interface the drag and drop method shall be used.
- It must be possible to send data to and from files.
- It is important that the tool supports minimising the download time for the user of the program. This is an interesting thing if the supplier can promise this.

The number of items here is not the same as in appendix B. Some of the questions in the questionnaire have been grouped together.

4.1.3 The answers from subgroups

Of the 16 people in the whole population 4 of them are spending lot of time to program user interfaces. A lot of time means that they use about half the time or more to that sort of activities. 13 of them use a lot of time to do logic programming and 5 of them use a lot of time to program communication units. The numbers of members in the different groups indicates that one person can do more than one of these things. Therefore adding the subgroups together does not adding to whole population.

Comparing subgroups with each other gives that there are very small differences. The number of people involved is very low and it is very hard to know if this actually tells something. Remember that one person can be a member in more than one group. It seems like the UI (user interface) group is more definite in their opinion about that the user interface should be generated from drawing it while compared to those in the communication group. Between the other groups it is hard to see any differences and even harder to see any significant ones.

One reason that there are so little differences is that one person who has answered that he spend 50% to develop communication units and 50% to develop logic has answered more like the persons in the user interface-group. This is probably because that person is open

minded. Two other persons are spending 50% on logic and 50% on user interfaces. They are probably more open minded as a result of their experiences.

That there are no significant differences between some of the subgroups or that the differences between subgroups are small and unsafe because of the low amount of material does not mean that there are no such differences. The only thing it tells is that such differences can not be safely proved within this investigation.

4.2 THE INTERVIEWS

4.2.1 The persons involved

The persons asked for interviews are some of those who have answered to the questionnaire. Three persons participated, one from each group representative for the answers from the whole group as a middle-answer-person. Some others who did not have the opportunity to answer the questionnaire were asked because of their experience in for example the language Java and their experience of doing user interface and working with IDE's.

The people involved in the interviews had something that they all thought in common and something's where the opinions differed.

4.2.2 Requirements needed for the way the business works

Some of the requirements have to do with how the organisation works and acts together. All the projects are organised to be teamwork and a product named Clear Case from Rational is the version handler. This version handler supports the teamwork. It is very important that the development tool and Clear Case are integrated, or can be integrated by the vendor. There is also a product named Rose from the same vendor used in an earlier phase in the development process, which is of interest to have integrated with the development tool.

4.2.3 Opinions about the visual interface of the tool

How the respondents want the visual interface is quite similar. It shall be as pure as possible. It is important to decide by him or her how big the windows are and how many there are at one time. It is better to have the windows beside another than overlapped. It shall be easy to work with, that is that it shall be possible to move around between different windows, to get to the source code from wherever you are.

4.2.4 Opinions about the content of the tool

The tool shall at least contain an editor, a debugger and hierarchical information about classes and methods. The debugger must be able to set break points, move freely and move step by step. It must also be possible to inspect the values of variables.

4.2.5 Opinions about wizards

It is of interest that the tool has wizards. Working with wizards can be on the one hand great help to get code generated, while just answering questions and the tool does the work. But it can on the other hand be very limiting, because the code is not written in the way the programmer wants it. So if you can work with them and change the code is the best.

4.2.6 Opinions about the ability to add and modify code

There are several ways to modify the code. Some tools do not allow the programmer to do it at all. If there will be own written code it will be overwritten by regenerating. Other tools let the programmer write his own code. Sometimes the programmer is only allowed to write between markers. If there will be code written by the programmer anywhere else it will be overwritten or there is no synchronisation between the visual designer and the source code.

The opinion about the possibilities of adding own code depends from thinking that the best thing is to let the generator do what ever it wants to and let that alone to the opinion that there shall be the ability to write everywhere where one wants. The reason for thinking in so different ways is that the experience of the respondents differs. Working with a tool, which does not generate fully satisfactory structured, or fully machine independent code will give that the programmer wants to control the code. Working with a better tool gives the thoughts about letting the generator do its own work, the programmer should write in one little square and let the tool puzzle it to the right place. What is a requirement from them all is that there must be opportunity to put some code of your own. It can not be only a tool that is based on components.

To summarise: The tool shall generate good structured code which is fully machine independent, and there should not be need for changing it. But there will always be need for writing own code. Therefore there shall be the ability to change the auto-generated code and to add own code. The own code shall be kept while regenerating and it shall be synchronised with the visual designer. A change in the visual designer shall lead to a change in the source code and vice versa directly.

4.2.7 Opinions about effectively working with the tool

One point when working with tools is to use prewritten code. Every one who has worked with the drag-and-drop-technique for drawing the interface is satisfied with the time saved compared to write it on their own. These tools contain a number of classes, some of them are in Suns originally library and some of them created by the tool. Suns classes are to be found in jdk (Java development kit which is an environment). Currently jdk 1.1 is available. Jdk 1.2 will be available in the summer. It is important that the tool is compliant with jdk 1.1. Otherwise it will not be machine independent.

To get as much reusable code as possible it is necessary to have more components than Sun and maybe also the IDE offers. So it is important that components can be added to the palette.

While using the classes provided by the tool it is easy to miss the classes from which they inherent methods. This will make breakdowns while moving the application to another machine, because the virtual machine is not able to translate it. So it would be of interest to get some help from the tool to package the needed files together in a Java Archive File (JAR-file).

To become a rapid working organisation there must be as much code as possible to be generated instead of the programmer writing it. Among the respondents there seems to be little or no experience in generating the events. But there was interest in getting more code generated. One person said it would be of interest to get the ability to wire logical components together as well.

4.2.8 Opinions about communication and distributed environment

It is important that the tool supports standards for communication with databases. It is also of interest that the tool supports standards for communication in a distributed environment. It is important that it does not only support products from one vendor. It should support standards like Java Data Base Connection (JDBC) and for example Common Object Request Brocer (CORBA) or Remote Method Invokation (RMI).

4.2.9 Opinions about help function

It is important to have help functions on-line, but it is also a wish to have a book. It then becomes optional which is being used.

4.3 THE REQUIREMENTS

To summarise the questionnaire and the interviews gives that the requirements for a tool are as follows.

4.3.1 Definite requirements

4.3.1.1 The content of the tool

The development tool shall contain all the tools that make it to a whole IDE. In chapter two there is an explanation what an IDE is. So some of the lines in this paragraph are about what is expected to find within an IDE. Some other lines are nearly related requirements.

- The tool must be visual.

- There must be an integrated editor.
- There must be an integrated visual debugger. The debugger must be able to run, set break points, go step by step and show values in variables.
- There must be hierarchical information to navigate through classes and methods.
- There must be a compiler integrated and the compiler must help with finding mistakes.
- The user interface shall easily be drawn while using a palette and the drag and drop technique. The code will be generated.
- The tool shall supply reusable components.
- There must be shortcuts for experienced users.

4.3.1.2 Requirements concerning organisation and generated code

- The tool must support working with Clear Case. The programmer must be allowed to leave his or her contribution developed by the tool to the team through Clear Case.
- The tool must run on Windows NT and preferably but not necessarily on Sun Solaris.
- The tool must be compliant with jdk 1.1.
- The tool must support generation of 100% pure Java code.
- The tool must generate good structured code. It shall as long as possible be no need for the programmer to modify it.
- The tool must generate comments. This is only necessary when the tool does not generate good structured code and there is need for modifying the code. Stating this as a definite requirement is based on the belief that such good tools do not exist.
- It is not possible to figure out everything that the user wants to do with the program. Needing more than common things there is no use for a development tool only based on components. Therefore there must be the ability to add own code.
- The manually added code and the visual designer should be synchronised.
- The code must not be overwritten while regenerating. The code written by the programmer shall be kept while regenerating.

- There must be ability to add components to the palette, either from another vendor or own written components.

4.3.1.3 Other requirements

- There must be support for database connection standard (JDBC).

- The tool must support generation of code that handles internationalisation.

- The tool must be easy to use. The tool must support experimental developing. It must be easy to start with a small and easy application. From this application it must be easy to continue to build a more sophisticated application.

- While visually building events the tool is not allowed to limit the event handling on components. All the methods in different classes must be available as possible events.

- There must be an on-line help function.

- The vendor must give support from Europe.

- There must be courses by the vendor.

4.3.2 Requirements that are not that definite

4.3.2.1 Interesting requirements which shall be used while classifying and choosing a tool

- The tool must put all files needed for execution into one JAR-file.

- There must be support for standard communication in distributed environments, for example CORBA and RMI.

- The tool must be smooth to work with. The source code must be easily reached. It must be easy to move around between the source code and the hierarchical information about classes. Moving around between several windows must be easily done. The source code must be easily reached for example by double clicking on a function in a browser.

- It is important that it is not cluttering. There must be ability to choose how many windows there shall be at one moment. The size of them must be eligible. There shall not be many overlapped windows. The menu and tool bar shall appear once as long as there is no need for more than the first menu and tool bar contents. There shall not be the same things on many places on the screen.

- The tool must support learning by doing. Not being afraid of doing things that are wrong, there must be a command like undo.

- There must be ability to build events visually.

- The tool must support incremental debugging. Doing changes while debugging will raise the effectiveness by the programmer. When adding more functionality to the application the application shall be executable at once.

- When adding new components to the palette it must be possible to use the existing components by the tool. Adding new components from other vendors must be easily done. New adding components shall be executable at once.

- There must be ability to test a small piece of code while debugging and if it is an applet it must be possible to test it without going through the www.

- The help function must content seek function and examples.

4.3.2.2 More requirements that are not that definite

- The tool must support jdk 1.2.

- The tool must support Rose.

- It must be possible to develop on both Unix and Windows.

- There must be the ability to include an invisible Ericsson identity. Every class must have a variable that says that it is an Ericsson class.

- There must be ability to include Ericsson copyright according to the law.

4.3.3 Other nice features that is "wishes"

- There must be documentation on paper as well.

- There must be wizards to work with.

- Reserved word must be highlighted.

- There must be possibilities to get help from the tool to prevent foolish errors.

- There must be possibility to get syntax error corrections.

- If the licenses are counted in the server, current users must be shown.

- There must be wizards called projects to handle all the files needed to get the executable program available. There must be hierarchical information about those projects.

- The text in the text editor must support the design rules of Ericsson.
  For example the design demands are to indent the text in a special way and that the square brackets will be put on the right place.

- The generated code must support the design demands of Ericsson as mentioned above.

5 THE SCAN OF THE MARKET FOR TOOLS

5.1 INTERESTING TOOLS

5.1.1 Java Workshop from SunSoft Incorporate

5.1.1.1 Description

The actual version is 2.0. It has a new Java compiler that reduces build time, and the application Profiler will helps watching how long time the applet or application spent on each method. The product includes the Visual GUI-builder, profiler as already mentioned, project and portfolio manager, source editor, source browser, debugger, Java compiler.

The earlier version of Sun Workshop for Java 1.0 was a bit unusual with its browser technology. Many developers were not satisfied with the interface of the product. “For example … the previous version required extensive paging back and forth to perform tasks.” [20] Sun has changed the new version so it looks more likes the other IDE’s on the market, although the browser technology remains.

Sun in Sweden say that this is not a very visual tool and many customers buy a 3rd party tool to compensate that. The tool is named Visaj and is sold by IST in England.

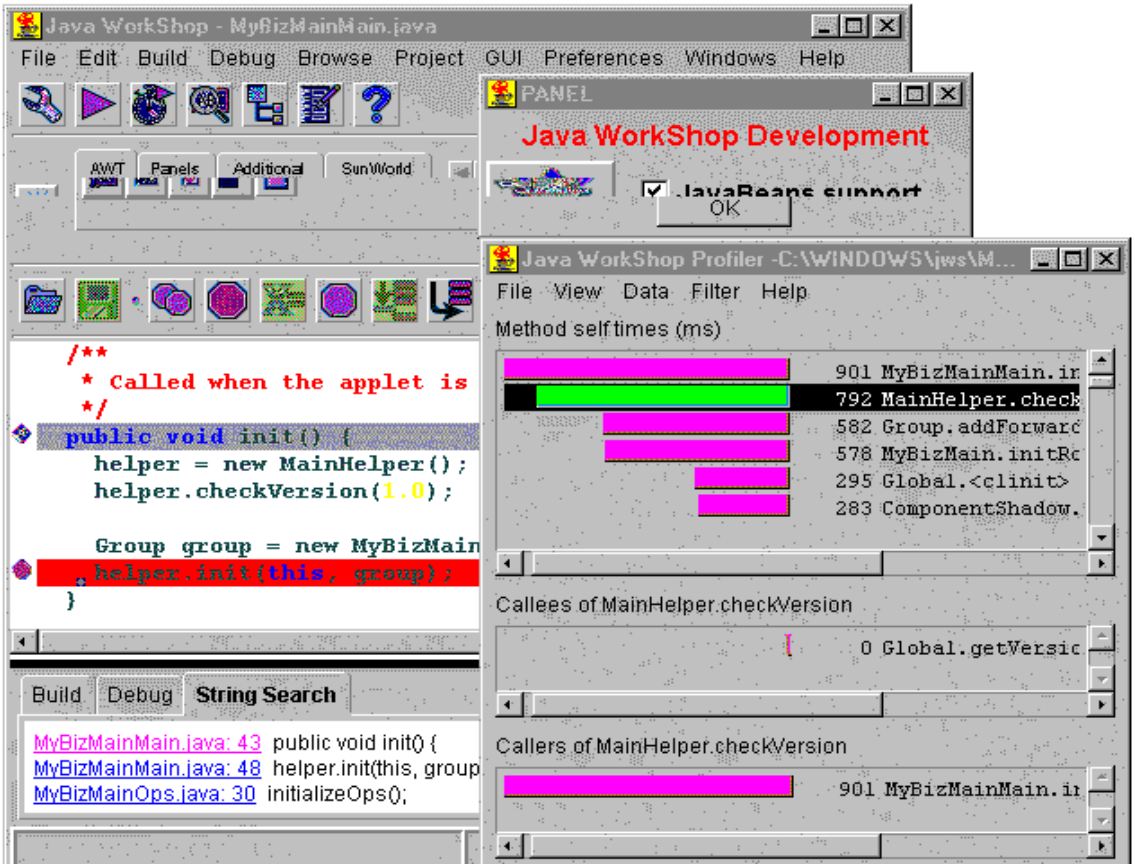

Figure 1: Java Workshop

5.1.1.2. Advantages

It runs on both Windows NT and Sun Solaris. It contains what is expected from an IDE. It generates good structured code with comments. The code is 100% Java. There is ability to add own code and keep it while regenerating. It is possible to add components to the palette. The help function is on-line and contains seek function and examples.

5.1.1.3 Disadvantages

They do not know if the product is compatible with Clear Case and therefore they have answered no. They have also said that this is something that has to be verified. So far at this writing they have not verified it anyway. There is no support for internationalisation and no support for CORBA. The code and the UI are not synchronised. The tool does not support visually building of events nor does it offer an incremental environment.

5.1.1.4. Judgement

The building of the tool to be executed within a browser makes it interesting. Some of the respondents in the questionnaire said that they did not want that. The argument is that it would not be fast enough. The most respondents though, said that it was interesting even if a few cover oneself with the argument as long as it will not take too much time running it.

If this one shall be chosen it will be for the reason that it run on both Windows and Sun and for the browser technology. There are quite many things that are not fulfilled within this tool. That the code and the user interface will not be synchronised is the most important thing that it does not supply, and that will be the reason not to use this tool.

5.1.2 Visaj from IST

5.1.2.1 Description

This product does not seem to be a whole IDE. The customers who give their opinions are appreciating it for being a complement to the other development tool. "The component tree is very good. Also it doesn't try to be an IDE." Nigel Sharples, Computing research Laboratory/New Mexico State University. [25]
"What I like about the Visaj product is the fact that it doesn't try to be every part of the development process. Visaj focuses on being an AWT GUI builder and doing it well. I also like that it generates real AWT code instead of bundled classes that are often poorly documented. Visaj is easily integrated with our existing tools and generates well understood industry standard code. To us that spells saved effort, improved efficiency and better profitability." David Westerdahl, Tricon Restaurants International, Inc. [25]

The product contains the following parts:
The visual class editor, which show the containment hierarchy and where the interface is build. The Layout editor is where the frames and dialogs is build. The Event editor let the programmer have code for events generated. The Resource Bundles handles internationalisation. Finally there is a dialog for selecting colours.

Visaj also offer a product called X-designer Java edition. For customers who have applications written in Motif and want to move them to Java they offer this product to be integrated with Visaj.

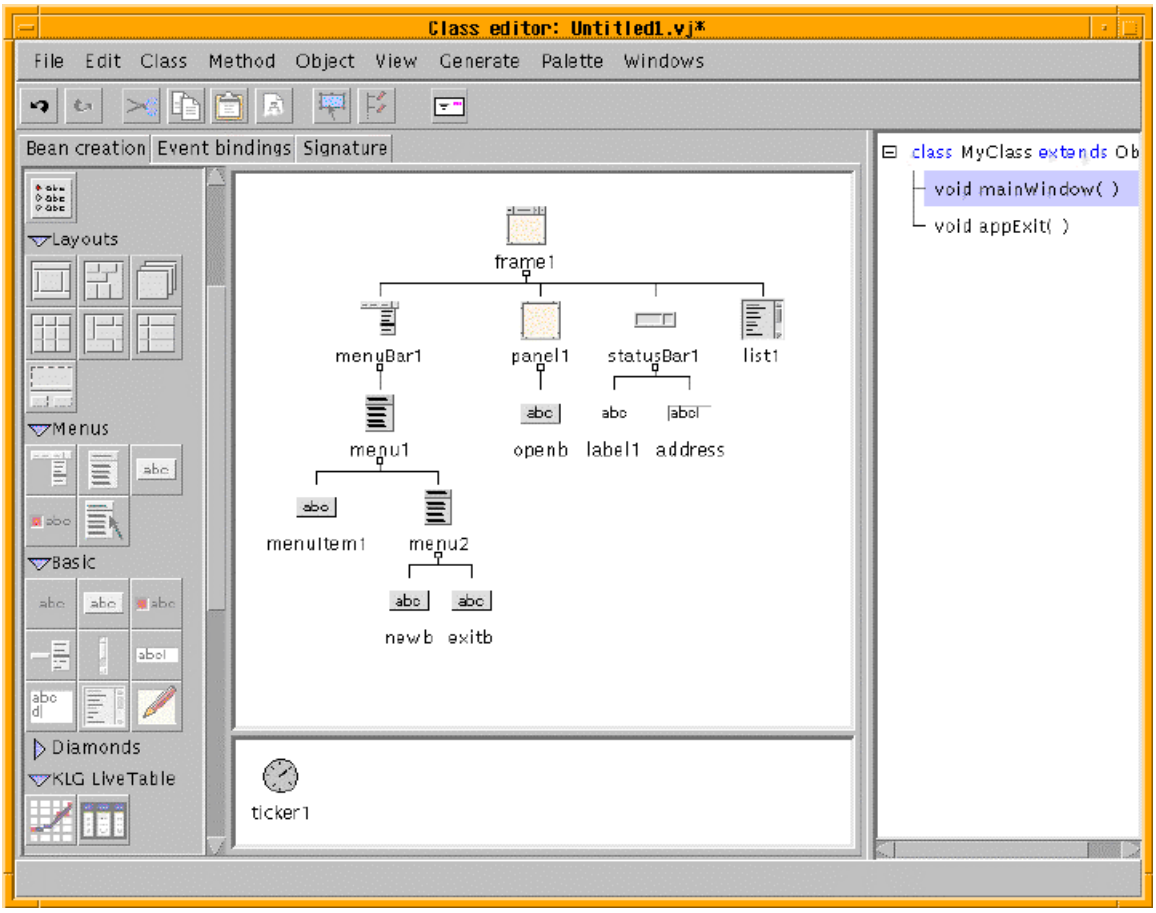

Figure 2: Visaj

5.1.2.2 Advantages

It generates 100% Java Code, runs on all platforms, supporting Java. So it will work on both Windows and Sun Solaris. There is possibility to have events visually built. It supports internationalisation.

5.1.2.3 Disadvantages

It is not a whole IDE. Connection with databases, connection in distributed environment such as CORBA and RMI and a debugger seems to missing.

5.1.2.4 Judgement

This is not a standalone product to get the whole development tool. To integrate it with Java Workshop is interesting because it lets the programmer have events generated and it will support internationalisation, which are two things that is not supplied from Java Workshop. Unfortunately I have not been able to figure out if there is ability to write own code and have it synchronised with the UI. If so this tool together with Java Workshop will be interesting, otherwise not.

5.1.3 Visual Age for Java from IBM

5.1.3.1 Description

It is an integrated development environment where you can develop visually both the interface and the events. The visually event handling is made of Parts from Objectshare. This tool is a fully integrated tool where the programmer does not have to leave the environment while doing changes.

So far it is the version 1.0. In the middle of 1998 there will be a new version.

This tool has got good criticism: "We were especially impressed by the clarity of the generated source code, which was well formatted and thoroughly commented for the convenience of both human readers and Java documentation tools" [24]

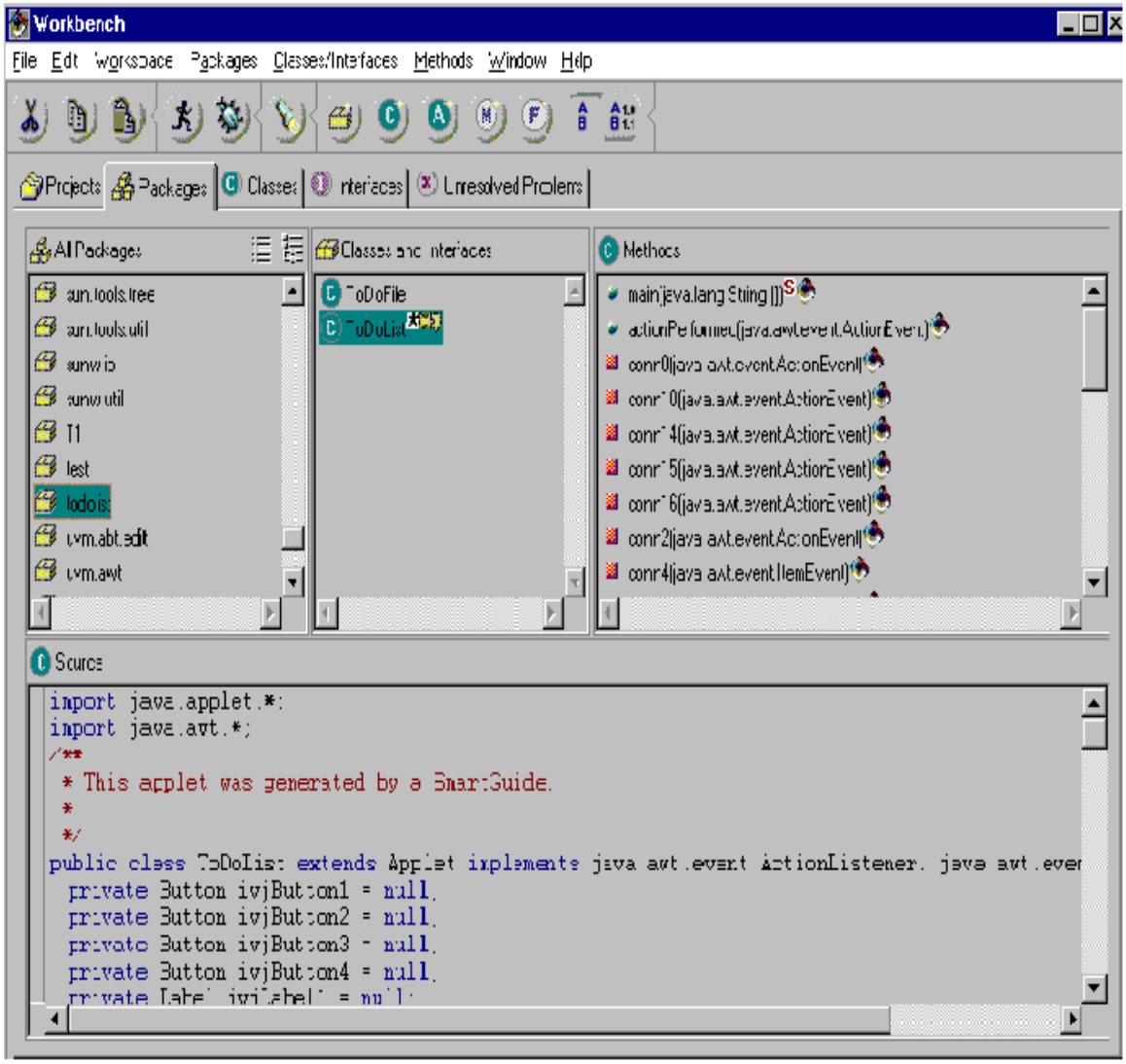

Figure 3: Visual Age

5.1.3.2 Advantages

The tool contains what is expected from an IDE. It generates 100% Java code. There is comments generated with the code. Own written code is kept while regenerating and there is ability to add components to the palette. There is ability to visually link events together and there are no limits on the event handling. The tool supplies incremental debugging.

The new version is going to support both Windows NT and Sun Solaris. The help function is on-line, contains of seek function and examples. There is support for CORBA and RMI . Numbers of windows are optional and the size of them is eligible.

5.1.3.3. Disadvantages

At the moment the product is not going to function together with Clear Case. IBM say that the current plan is to release a version with open API's for the integration with other vendor's products in the middle of 1998. They do not answering the question if they will guarantee that it will work.

At Ericsson it is important that there is ability to make changes in the code and have that reflected in the user interface. The person at IBM said that this is not such a tool where there is need for that. While persisting in the necessary having the code and UI synchronised she said that there was the ability to change colours and even other things and that it will be synchronised. I am hesitating about that answer.

5.1.3.4. Judgement

This tool seems to have a lot of desired functionality. After the new version is released and if there is possible to get the guarantee for Clear Case it could be an interesting tool. If it is a truly two-way tool, that is that the changes in the source code will be reflected in the UI and vice versa has to be proved though. But to answer to my questions they have to have the guarantee that the choice was not already done and they will tell me which functionality the customer needs. I will sort out this tool because of the contact with them.

5.1.4 PowerJ from Sybase

5.1.4.1 Description

Sybase offers an IDE, which includes much. One thing included is Jaguar. It is their own component, which handles objects in distributed environments. Jaguar can manage Java, Active X, C/C++ and CORBA. The tool also offers visual building of events. According to one reviewer this tool requires less Java knowledge on the part of the developer. The way of doing it is a bit strange compared to the other tools.

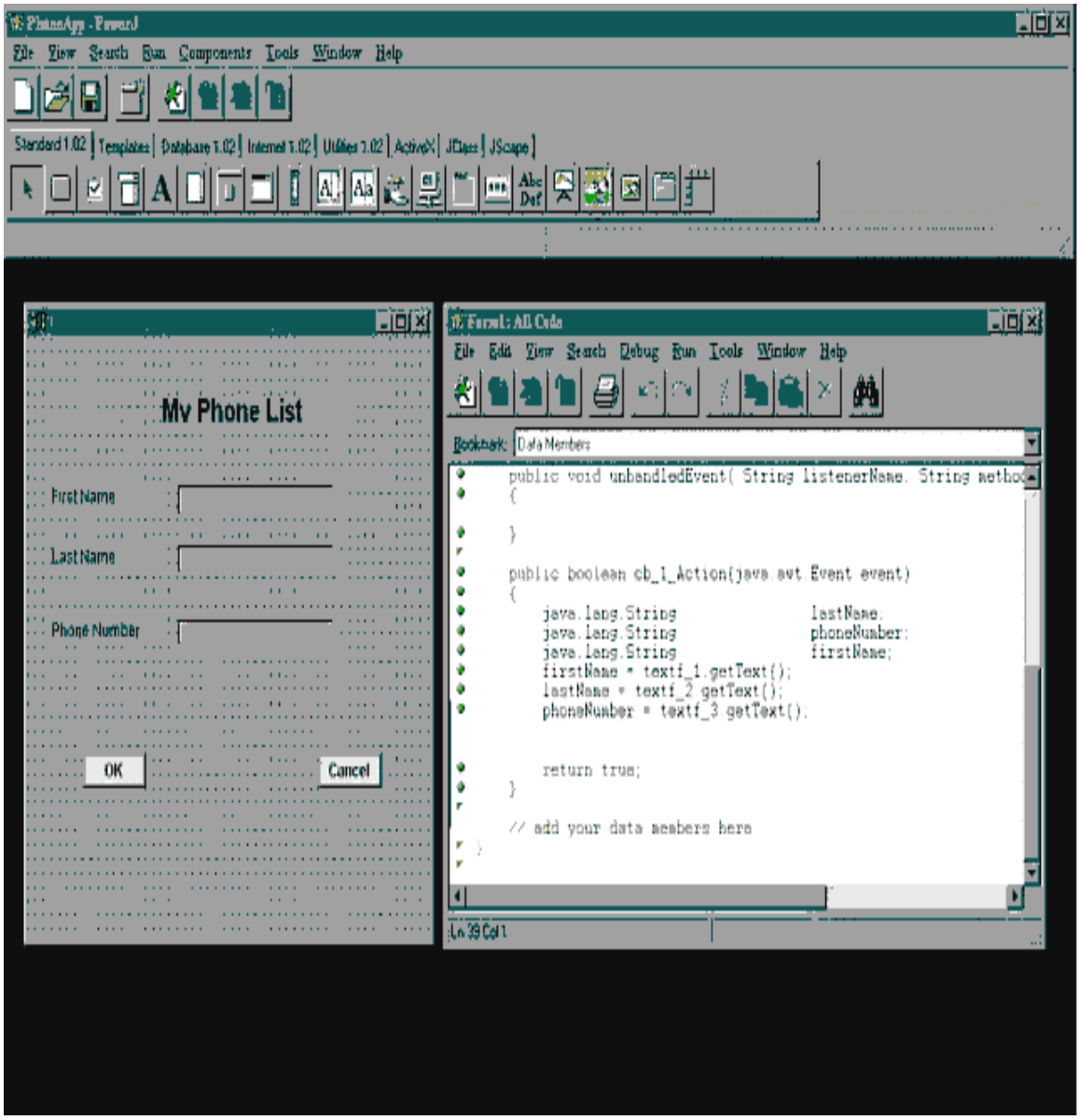

Figure 4: PowerJ

5.1.4.2 Advantage

This tool fulfils the most important requirements from the firm. Rational is a partner and they guarantee that Clear Case and PowerJ will work together. It contains what is expected from an IDE. The code generated is 100% Java code, good structured and there are comments. The code and the UI are synchronised. Own written code is kept while regenerating and there is ability to add components to the palette. This tool has the facility visually building events. It also supplies incremental debugging. It supports CORBA and RMI. Numbers of windows are optional and the size of them is eligible. The help function are on-line and contains of seek function and examples.

5.1.4.3 Disadvantage

There is no support for internationalisation. It will not run on Sun. It will not generate Ericsson classes. There is no possibility to see other users of licenses, which is a wish.

5.1.4.4 Judgement

This is a good tool, which would suit Ericsson well. It fulfils very many requirements. There is one thing worth mentioning though; the visually building of events is done in a strange way. While building the events visually the drag and drop-technique is used between a component in the visual form designer and the source code of that method. A couple of dialog boxes has to be worked through and the code is generated. I think that this way of doing it will require more knowledge in Java than the other way offered within other tools. That is why this tool is not proposed.

5.1.5 J Builder from Borland

5.1.5.1 Description

The most complete version is the Client/Server Suite 1.01. It has got integrated version control, visual query-building tools for databases and support for CORBA and RMI. This tool is a two-way tool, which means that it is reflecting changes in the source code to the UI form builder. It is possible to choose another editor and have the code and the UI synchronised even though. The component palette is tabbed, which makes the screen pure.

“Although its version number is 1.0, Borland’s Jbuilder feels like a second-generation Java development environment. This long-waited entry into the Java tools market excels in many important areas from support for JDK 1.1.1 and Java Beans to seamless two-way code creation”. [16]

“The bottom line: Very good” [15] “For an enterprise that is looking to move to Java or Web-distributed applications, Jbuilder is an excellent choice in a young market” [14]

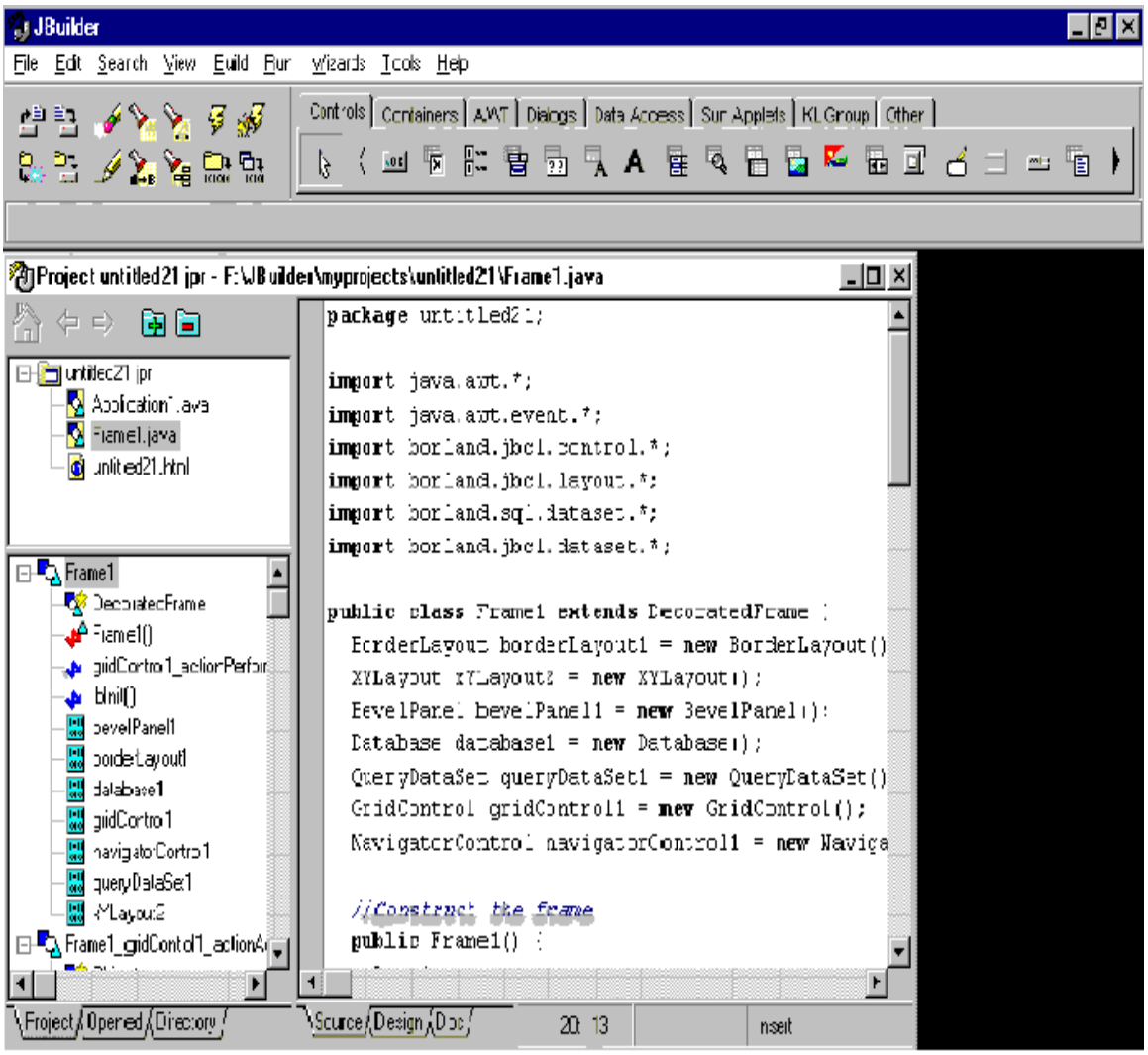

Figure 5: J Builder

5.1.5.2 Advantages

Borland say that they support the integration to Clear Case. It fulfils what is expected from an IDE. It is a two-way tool, which means that the user interface and the code will be synchronised. The manual added code would not be overwritten. The code generated by the tool is good structured and contains comments. It generates 100 % Java code. There is ability to add components to the palette. There is support for CORBA and RMI. The numbers of windows are optional and the size of them is eligible.

5.1.5.3 Disadvantages

There is no ability to build events visually and it is not offering incremental debugging. The tool does not support Rose and it will not run on Sun.

5.1.5.4. Judgement

This tool fulfils the most important requirements from the business. It will fulfil the most of the requirements. That it does not support visually building of events and not supply incremental debugging is a drawback however, and it will not be picked out as number one.

5.1.6 Parts for Java from Objectshare

5.1.6.1 Description

Objectshare is a division of Parc Place. They inform about their Parts for Java 2.0 on the web. They are shipping a new version, Professional Edition near the end of March. It has three primary differences from the base product: integrated version control, automated creation of database applications and a set of beans for database work, and support for the use and creation of ActiveX controls.

The integrated development environment has the following parts: Project manager, Visual designer, Class Master browser, Java debugger. There are some wizards such as CORBA wizard, RMI wizard, and the delivery assistant.

The code is structured in a way where the auto-generated code is put in super-classes and the manual added code is put in sub-classes. They say themselves the code style is close to how the code should look like according to Sun.

The judgement from reviewers about this product is good, even if there are some complaints about components missing in the visual designer and one person find the debugger clumsy to work with. “Overall, however, once you became accustomed to its quirks, Parts for Java is a solid tool.” [21]

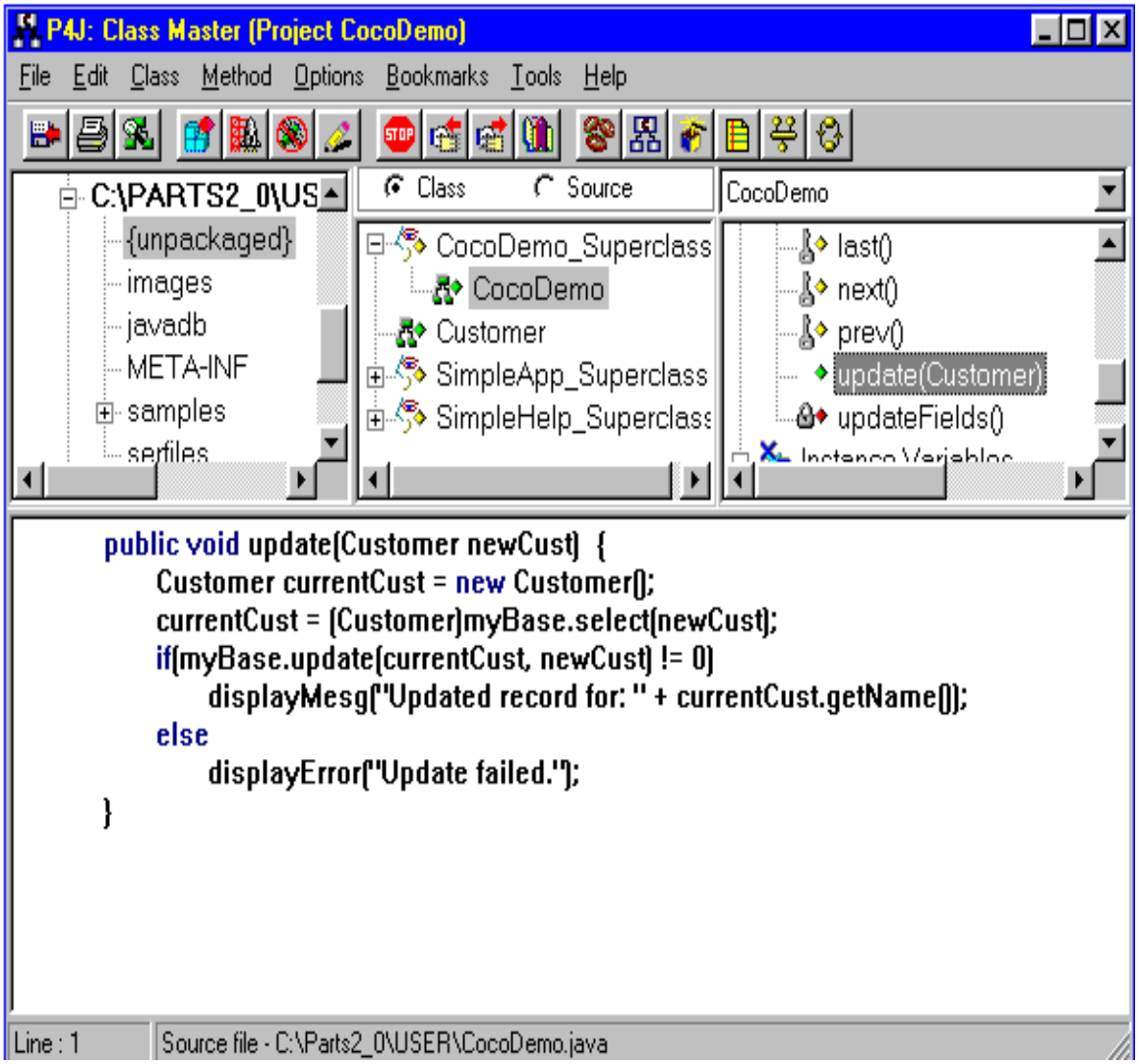

Figure 6: Parts for Java

5.1.6.2 Advantages

The tool contains what is expected from an IDE. They are working together with Rational and will support that the product will work together with Clear Case. It generates 100% Java code. The code is good structured with comments and there is ability to add own code. The tool will not override the code as long as the programmer conforms to the code style. The source code and the UI will be synchronised. There is ability to add components to the palette.

The tool support CORBA and RMI. The numbers of windows are optional and the size of them is eligible. There is ability to build events visually. The help function is on-line, contains seek function and examples.

5.1.6.3 Disadvantages

This tool does not support internationalisation. There is no incremental debugging. It will not run on Sun Solaris. Among the requirements, which are wishes there is no syntax corrections and no Ericsson design in the editor or in the source code.

5.1.6.4 Judgement

This is an interesting tool. Almost every important requirement is fulfilled. The only thing that is not fulfilled among them is support for internationalisation. It offers visually building of events, which is making the developing faster. This tool does not offer incremental debugging and therefore it is not on the top on the list but it will be a good candidate.

5.1.7 Visual Café from Symantec

5.1.7.1. Description

The latest version of Visual Café is 2.1 but in March there is a new version: 2.5 Professional Development Edition and the 2.5 Database Development Edition. The Database Edition is the complete one. Developers having bought the 2.0 or 2.1 version will have the opportunity to download the new version free.

One change in the 2.0 version is that there is a new compiler licensed from Sun. It is faster. “In contrast to javac, the Java compiler that ships with the jdk, Café’s compiler is fast, making the entire compile-test-debug cycle much quicker.” [22]

The version 2.5 has a tabbed formula where the screen will be pure and optimised. Another thing is that there is the ability to change between RAD mode and Source code mode. The RAD mode enables forms-based rapid application development with automatic code generation. The source code mode is without automatic code generation. The database development edition contains all the things, which are to be found in the professional edition plus local relational database and middleware server with unlimited database connectivity.

Several persons at Ericsson have tried the early version 1.0. Quite a few things were not satisfying. The code was not fully machine independent. The programmer was allowed to write everywhere in the code, but it will get out of synchronisation if he or she writes anywhere else than between the proprietary markers, and the result is that the programmer is abandoning the visual environment.

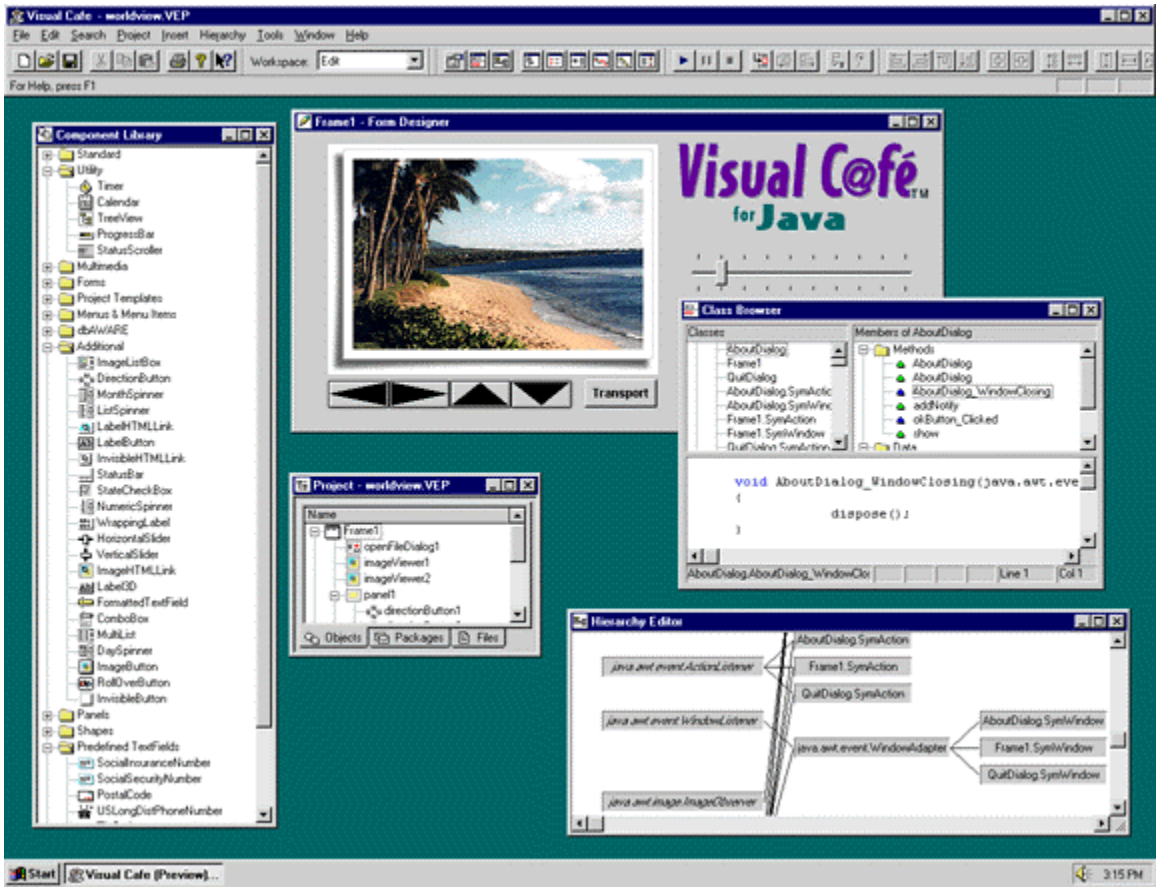

Figure 7: Visual Café for Java

5.1.7.2 Advantages

The tool contains what is expected from an IDE. It generates 100% Java code. The code is good structured with comments. There is possibility to add own code and the own written code will be kept while regenerating as long as the programmer conform to the code style. The source code and the UI will be synchronised. There is ability to add components to the palette. It supplies building events visually and there are no limits on event handling. It is an incremental environment where the developer can add functionality while debugging. The code will be compiled in the background. The numbers of windows are optional and the size of them is eligible. The help function is on-line and contains seek function and examples.

5.1.7.3 Disadvantages

It is a disadvantage that they do not say for sure that they support Clear Case. The person at Symantec answering said that there are a number of API's included in the product and

Rational is using them to integrate Rose and that he will find out if that is the case with Clear Case too. By this writing nothing has been heard.

5.1.7.4 Judgement

This tool is fulfilling the most necessary requirements on a development tool. It offers visually linking of events and incremental debugging. That is making this tool interesting. It could be, however, that the code written and compiled in the background would not behave in the same way when it is executed. So this has to be proved. While choosing this one the guarantee for that it works together with Clear Case has to be received from Symantec. The tool is putting markers where the programmer is allowed to write code. If the manual added code is put outside the markers it will not be parsed. So the programmer has to conform to the code style. The code style of the tool has to be evaluated to see if this is a drawback or not. I think this one is worth testing.

5.2 TOOLS THAT COULD BE INTERESTING IN THE FUTURE

5.2.1 Mojo from Penumbra Software

5.2.1.1 Description

The tool contains the Designer, the Visual Scripter and the Coder. The Visual Designer let the programmer build the user interface using drag and drop-technique. The Visual Scripter let the programmer add events, actions, functions and logic with the same technique. The Coder let the programmer access the code. Mojo Coder lets the programmer add his or her functions or objects supplied by another vendor as well. The unique point of this product is the ability to add logic components through the drag- and drop-technique.

5.2.1.2 Advantages

The tool does contain everything that makes it to an IDE, except that the debugger is not integrated but there is a link to one. There is the ability to add own written code, and it will be synchronised with the visual designer and not overwritten. The code is 100% Java code. There is ability to add components to the palette. It supports JDBC. It runs on both Windows NT and Sun Solaris. There is the command undo within the coding environment. The programmer can use the tool to build new components and they are executable at ones. The size of the windows is eligible.

5.2.1.3 Disadvantages

This product does not offer support from Europe, there are no on-line help, no seek functions or examples within the help function and there are no courses. The tool does not support CORBA and RMI nor does it support the integration with Clear Case. The generated code will not content comments. There is no ability to build events visually. They include a link to JDB with which rudimentary debugging is available. Among those

requirements which are wishes is worth mentioning that there are no wizards to work with, no possibility to have Ericsson design on the source code and in the editor. There are no projects and no syntax error correction.

Penumbra Software is a small company with 12 employees. It has been around since 1995. One test shows that their support was not to lean on. “We found one significant weakness with Mojo: poor technical support from Penumbra.” [19]

5.2.1.4 Judgement

This tool is supplying many things on the requirement list although there are some missing too. It is interesting because it has got reusable components even for events and logic and will probably raise the efficiency by the programmer. But good support from Europe is a very important thing and because of that this is not a tool to investigate by the moment. It is interesting to keep an eye on this tool if the company will place resellers in Europe.

5.2.2 Cosmo Code from Silicon Graphic

5.2.2.1 Description

While evaluating with Cosmo Code the interface shows menu and toolbar at the top, card panel which is a tabbed collection of panes in the bottom and between there are the source-display pane. The card panels content is grouped in three categories: development, debugging and compile/execute. It is the same construction as in Jbuilder, PowerJ and the latest version of Visual Café and is one way of getting a pure screen. The tool offers visually linking on events similar to some other tools; Parts for Java and Visual Age.

5.2.2.2 Advantage

It contains what is expected from an IDE. The debugger has got all the facilities, from the requirement list. It generates good structured code with comments. The code is 100% Java code. There is ability to add own written code and it will be kept while regenerating code. The source code and the UI will be synchronised. It is possible to add components to the palette. The help-function is on-line and contains seek function and examples. The numbers of windows are optional and the size of them is eligible. There is possibility to build events visually. The tool support incremental debugging.

5.2.2.3 Disadvantage

It runs on IRIX, but a version for Windows 95 and Windows NT is coming in May. There are limits on event handling while wiring components together but there is also another way doing it. This tool has no support from Europe. There are no courses and no books.

5.2.2.4 Judgement

Seems to be an interesting tool, which supports many of the requirements on the list. It is supplying with opportunities to work rapidly with this event programming. By this writing it is on the wrong platform however and that there is no support from Europe is another thing that make the tool not actual for now. But it is interesting to keep an eye on the company, if they will place resellers in Europe.

5.3 OTHER TOOLS

5.3.1 Object Quest Developer

Object quest Developer is a framework for building applications. It is based on the Microsoft Foundation Classes and to develop application products using Objectquest Developer a full development version of Microsoft Visual J++ is required

5.3.2 Super Cede 2.0 from Asymetrix

This one I have seen mentioned some times as a good visual tool. “Competition from similar tools, such as Symantec’s Visual Café and Asymetrix’s SuperCede is strong…” [14] Unfortunately I have not been able to find it on the web. There is maybe the opportunity that someone else has overtaken it. I will not investigate it any further.

5.3.3 Java Sun Studio

This is a component-based integrated development environment.
The disadvantage is that there is no ability to add code.
The ability to add code to customise an application is a necessary feature for most programmers. This is one important thing on the requirement list for Ericsson as well, so it will not be of interest.

5.3.4 J++ from Microsoft

5.3.4.1 Description

The information available on the net is about the version 1.1, which was released 3rd of March last spring and the new one just announced: the version 6.0. This one is a trial version. It will supply tools for developing Java applications for windows-based software. J/Direct is the technology that allows developers to build native windows-based applications using Java. There are new classes named WFC (Windows Foundation Classes). “Visual J++ is designed to provide easy access to the full power of the windows platform” [18]

This has got quite a lot criticism for being not visual enough. “…compared against the less fully graphical implementations of Java programming such as Microsoft’s J++ …the visual advantages of Jbuilder cannot be denied.” [14]

Microsoft has got good criticism about the fast just-in-time compiler and the judgement is that it is one of the fastest compilers on the market. “Microsoft has created one of the best Java virtual machines to date. In addition being 100% compatible, it is one of the fastest, most reliable, and the most functional virtual machine for Java.” [17]

Just as in other products from Microsoft, the component technique is named ActiveX and is different from most of the other vendors making Java development tools, using Java Beans components.

It is hard to look at the product J++ without getting into dispute between Microsoft and Sun about the standard of the Java language. With this new classes for Windows based applications it will from Sun’s point of view seems like Microsoft not want the language Java to run across platforms. This has escalated the war. There is a judgement in court that says that Microsoft is not allowed to use the logo from Sun anymore.

5.3.4.2 Judgement

Using the classes for Windows based applications means that the application will run in a strange way on a Sun Machine. That facility is only worth using when the application is going to run only on Windows. Ericsson wants to develop applications that runs on both Windows and Sun Solaris. So there is no need for these classes.

I am hesitating about the machine independence of Microsoft. It is probably not a bad tool but I will not investigate it any further.

5.4 THE CONTACT WITH THE ENTERPRISES

The answering from Objectshare is excellent. Someone at the office in USA answers the day after questioning. Borland is hard to get in contact with. Anyway they do not sell anything themselves to Sweden, they have different resellers here, so it will be a matter for the reseller. Sybase and Symantec answers without complaints. IBM was not willing to answer before they got the guarantee that the choice was not done and they also wanted to know the requirements from Ericsson.

6. RESULT

The result from this survey as a whole is the recommendation of a tool for Java. To learn about which tool is to be recommended there has been an investigation about the requirements on a tool. There has been an investigation about which tools there are on the market. So the result is threefold. First there is the result which tells about the requirements, after that there is the result which tells about possible tools and at last the recommendation of one tool.

6.1 THE RESULT ABOUT REQUIREMENTS ON A DEVELOPMENT TOOL

The table with the requirements (see chapter 4.3) is in three parts. First are the most definite requirements, after that not that definite requirements and at last “wishes”.

6.1.1 Definite requirements

6.1.1.1 Requirements concerning an IDE

In chapter two we learned that an IDE contains an editor, a compiler, a debugger and a hierarchical browser. Some of the requirements in the requirement list are considered with just checking if the product is an IDE. The points below contain those requirements and related requirements, such as how the debugger behaves. These requirements are important.

- The tool must be visual.
- There must be an integrated editor.
- There must be an integrated debugger.
- The debugger must be able to run independently.
- The debugger must be able to set break points.
- The debugger must be able to do single stepping.
- The debugger must be able to inspect variables.
- There must be hierarchical information about classes and methods.
- There must be an integrated compiler.
- The drag and drop-technique shall be used to draw the user interface.
- The tool shall supply reusable components.
- There must be shortcuts for experienced people, who not want to walk through number of menus.

6.1.1.2 Requirements concerning organisation and generated code

These requirements are very important to have fulfilled.

- The tool must function together with Clear Case.
- The tool must run on Windows NT.
- The tool must be compliant with jdk 1.1

- It must generate 100% Java code.
- The generated code must be well structured.
- The generated code must contain comments.
- There must be ability to add own code.
- The source code and the user interface must be synchronised.
- The own written code must be saved while regenerating.
- There must be ability to add components to the palette

6.1.1.3 Other definite requirements

- There must be support for JDBC.
- The tool must support internationalisation.
- It must be possible to develop experimentally.
- For the tools, which are offering visually linking of events, there shall not be limits in event handling.
- The help function must be on-line.
- There must be support from Europe.
- There must be courses by the vendor.

6.1.2 Requirements that are not that definite

6.1.2.1 Interesting requirements which will be used while classifying and choosing a tool

- The tool must put all files needed for execution into one JAR-file.
- The tool must support standard for communication in distributed environments for example CORBA
- The tool must support standard for communication in distributed environments for example RMI.
- The source code must be easily reached for example while double clicking on a function in a browser.
- The number of windows must be optional.
- The size of the windows must be eligible.
- There must be a command like undo.
- There must be ability to build events visually.
- The tool must supply incremental debugging.
- There must be possibility to use the tool to build components.
- New components must be usable at once.
- There must be possibility to test a piece of code while debugging.
- It must be possible to test an applet without going through www.
- The help function must contain seek function.
- The help function must contain examples.

6.1.2.2 More requirements, which are not that definite:

- The tool must support jdk 1.2.
- The tool must support Rose.
- The tool must run on Sun.
- The tool must generate variables that tell that the classes are Ericsson classes.
- The tool must generate Ericsson copyright notices.

6.1.3 Wishes

- There must be documentation in a book.
- There must be wizards to work with.
- Reserved words must be highlighted.
- There must be possibility to get help from the tool preventing foolish errors.
- There must be possibility to get syntax error corrections.
- If the licenses are counted in the server, the other users must be shown.
- There must be wizards called projects, handling all the files needed to have the executable program.
- There must be hierarchical information about the projects.
- The text in the text editor must support the design rules of Ericsson.
- The generated code must support the design rules of Ericsson.

6.1.4 Summarising

The most important requirements are that the tool is an IDE and that the generated code will be machine independent. To code shall be good structured and have comments. There shall at first be no need to modify the auto-generated code. If it is necessary, there shall be the ability and the changes in the source code shall immediately be reflected in the UI.

Some requirements not that important but interesting to have fulfilled is that the tool support visually building on events and supply incremental debugging. This will improve the efficiency of the programmer and will make the tool easier to work with. Other things that will improve the user friendliness are that the number of windows will be optional and the size of them will be eligible.

6.2 MATCHING OF THE REQUIREMENTS FOR SOME INTERESTING TOOLS

6.2.1 The tables

The facts presented in the tables below is what the vendors says themselves. The square is filled with a question mark if the vendors did not answer. The answers are subjective.

6.2.2 Definite requirements

6.2.2.1 Requirements concerning an IDE

Requirement	Parts for Java	J Builder	Visual Age	PowerJ	Java Wo-rkshop	Visual Café
Visual	Yes	Yes	Yes	Yes	Yes	Yes
integrated editor	Yes	Yes	Yes	Yes	Yes	Yes
integrated debugger	Yes	Yes	Yes	Yes	Yes	Yes
the debugger runs freely	Yes	Yes	Yes	Yes	Yes	Yes
the debugger make brakes	Yes	Yes	Yes	Yes	Yes	Yes
the debugger goes stepwise	Yes	Yes	Yes	Yes	Yes	Yes
The debugger shows values	Yes	Yes	Yes	Yes	Yes	Yes
hierarchical information	Yes	Yes	Yes	Yes	Yes	Yes
integrated compiler	Yes	Yes	Yes	Yes	Yes	Yes
reusable components	Yes	Yes	Yes	Yes	Yes	Yes
drag and drop to build UI	Yes	Yes	Yes	Yes	Yes	Yes
short cuts	Yes	Yes	Yes	Yes	Yes	Yes

6.2.2.2 Requirements concerning organisation and generated code

Requirement	Parts for Java	J Builder	Visual Age	PowerJ	Java Wo-rkshop	Visual Café
Clear Case	Yes	Yes	?	Yes	No	?
Windows NT	Yes	Yes	Yes	Yes	Yes	Yes
Jdk 1.1	Yes	Yes	Yes	Yes	Yes	Yes
100% Java code	Yes	Yes	Yes	Yes	Yes	Yes
Good structured code	Yes	Yes	?	Yes	Yes	Yes
Comments	Yes	Yes	Yes	Yes	Yes	Yes
Add own code	Yes	Yes	Yes	Yes	Yes	Yes
Code and UI synchronised	Yes	Yes	Yes	Yes	No	Yes
Keep own written code	Yes	Yes	Yes	Yes	Yes	Yes/No
Add components to palette	Yes	Yes	Yes	Yes	Yes	Yes

The question about Clear Case seems to be hard for some of them to answer. Sun answers that it has to be verified and Symantec that it will probably work but they will find out. IBM says that they have open API's but they do not answer to the question if they will support it.

IBM does not answer to the question about good structured code. Instead they do explain how the tool is built and that the programmer does not have to bother about files. Another thing, which I want to comment, is that it took me some time to explain for the lady at IBM that it is important to have the ability to make changes in the code and have that reflected in the user interface. First she said that this is not such a tool where there is need for that. While persisting in the importance of having the source code and the UI synchronised she said that there was the ability to change colours and even other things and it will be synchronised. I am hesitating about that answer.

6.2.2.3 Other definite requirements

Requirement	Parts for Java	J Builder	Visual Age	PowerJ	Java Wo-rkshop	Visual Café
Support for JDBC	Yes	Yes	Yes	Yes	jdk1.1	Yes
Internationalisation	No	Yes	Yes	No	No	Yes
Experimental developing	Yes	Yes	Yes	Yes	?	Yes
No limits in event handling	Yes	-	Yes	Yes	-	Yes
Help function on-line	Yes	Yes	Yes	Yes	Yes	Yes
Support from Europe	Yes	Yes	Yes	Yes	Yes	Yes
Courses by the vendor	Yes	Yes	Yes	Yes	?	Yes

6.2.3 Requirements that are not that definite

6.2.3.1 Requirements used to classyfing tools

Among the requirements that are not so important there are some interesting and will therefore make sense while classifying the tools. They are put in the table below.

Requirement	Parts for Java	J Builder	Visual Age	PowerJ	JavaWo-rkshop	Visual Café
Build JAR-files	Yes	Yes	Yes	Yes	Yes	Yes
CORBA	Yes	Yes	Yes	Yes	No	No
RMI	Yes	Yes	Yes	Yes	jdk1.1	Yes
Easy to reach the source code	Yes	Yes	Yes	Yes	Yes	Yes
Number of windows optional	Yes	Yes	Yes	Yes	Yes	Yes
Size of windows eligible	Yes	?	Yes	Yes	Yes	Yes
Undo	Yes	Yes	Yes	Yes	Yes	Yes
Build events visually	Yes	No	Yes	Yes	No	Yes
Incremental debugging	No	No	Yes	Yes	No	Yes
Use the tool to build components		Yes	Yes	Yes	Yes	Yes
Use new components immediately	Yes	Yes	Yes	Yes	Yes	Yes
Test a prototype	?	Yes	Yes	Yes	Yes	Yes
Test an applet without www	Yes	Yes	Yes	Yes	Yes	Yes
Help function seek function	Yes	Yes	Yes	Yes	Yes	Yes
Help function examples	Yes	Yes	Yes	Yes	Yes	Yes

6.2.3.2 More requirements that are not that definite.

Requirement	Parts for Java	J Builder	Visual Age	PowerJ	Java Wo-rkshop	Visual Café
Support jdk 1.2	Yes	Yes	No	?	?	?
Support Rose	Yes	No	will do	?	No	Yes
Run on Sun	No	No	will do	No	Yes	No
Generates Ericsson classes	Yes	I think so	Yes	No	No	Yes
Generates Ericsson copyright	Yes	Yes	?	Yes	Yes	Yes

6.2.4 Wishes

The following table shows things that are nice to have. This is a wish that the tool shall fulfil. It is not necessary and therefore the tools will not be classified because of the requirements here.

Requirement	Parts for Java	J Builder	Visual Age	PowerJ	Java Wo-rkshop	Visual Café
Book	Yes	Yes	Yes	Cdpaper	Yes	Yes
Wizard	Yes	Yes	Yes	Yes	Yes	Yes
Reserved words highlighted	Yes	Yes	Yes	Yes	Yes	Yes
Preventing foolish errors	?	?	Yes	?	?	Yes
Syntax correction	No	Yes	Yes	?	No	Yes
See other users of licenses	client license	?	No	No	No	client license
Projects	Yes	Yes	Yes	Yes	Yes	Yes
Hierarchical information about projects	Yes	Yes	Yes	?	Yes	?
Editor Ericsson design	No	Yes	?	?	No	No
Source code Ericsson design	No	Yes	Yes	?	No	Yes

6.2.5 Summarising the tables

When it comes to the content in the first table, which are important requirements, all the firms are saying yes to everything.

The second table does also contain important requirements. There are many requirements which are fulfilled within all tools even here. Something worth mentioning is if the firms are able to allow that their product will work together with Clear Case. Borland, Objectshare and Sybase are promising it. IBM does not answer to the question. Symantec says that they will find out and Sun says that it has to be verified. But after that nothing has been heard about it.

Further in the second table we are told that the code and the UI will not be synchronised while using Java workshop. Java Workshop, PowerJ and Parts for Java do not support internationalisation. Visual Café will in some cases not keep the code written by the programmer. If that matters depends on the code style. Parts for Java will also override the code if the programmer will write outside the markers. It does not matter because that it use the object oriented code style and where the superclasses are it is not allowed to put in own written code.

The forth table contains not that definite requirements. Result worth mentioning here is that Java Workshop and Visual Café does not support CORBA. Java Workshop and

J Builder do not support visually building of events. Incremental debugging are only supported by PowerJ, Visual Age and Visual Café.

6.3 CHOOSING A TOOL

The expectation was not that all the requirements could be fulfilled within one tool. That is not the result. The tools do fulfil the requirements more or less. There are a number of tools that could fit Ericsson. The interesting tools are Jbuilder from Borland, Parts for Java from Objectshare, Visual Age from IBM, PowerJ from Sybase, Visual Café from Symantec and maybe Java Workshop from SunSoft inc.

6.3.1 Classifying the tools

The recommended tool, has to fulfil the most definite requirements. After that there are some requirements which it would be nice to have fulfilled. Some of the requirements that are not that definite are not included in every tool in the table. The interesting things among the requirements where the functionality between tools differs is that it support visual building of events, it supports internationalisation, it supplies an incremental environment while supporting incremental debugging and that it supports communication like CORBA and RMI. From here the tools can be classified.

Functionality	Parts for Java	Jbuilder	Visual Age	PowerJ	Java Workshop	Visual Café
Visually building events	Yes	No	Yes	Yes	No	Yes
Internationalisation	No	Yes	Yes	No	No	Yes
Incremental debugging	No	No	Yes	Yes	No	Yes
CORBA	Yes	Yes	Yes	Yes	No	No
RMI	Yes	Yes	Yes	Yes	jdk1.1	Yes

Objectshare explains that they do not have incremental debugging because the virtual machine from Sun does not allow this. And they say that those who are supplying this is do it running against a “non-standard” Java VM.

6.3.2 The recommendation

It would be nice to recommend a tool which is fulfilling the most important requirements and supplying visually linking of events, incremental debugging and support for CORBA and RMI. There are two development tools, which are supporting all these. They are Visual Age (except that the guarantee that it will work together with Clear Case is not obtained) and PowerJ. PowerJ is the tool, fulfilling everything except support for internationalisation. The visually linking of events, though, seems like a hybrid with programming to me. That the programmer will work faster and do not need to know very much about programming while doing the visually linking on events is the argument for choosing a tool which offers that facility. I think that the way the other tools build events visually is supporting that the person using it knows less about programming. So PowerJ

is not my choice for this reason. Neither is Visual Age my choice. The reason for that is the contact with IBM.

To choose a tool which offer incremental debugging is a risk in one way. It could be that the changes is not fully machine independent. Because there is interest from the organisation to got this I think it is worth evaluating. While not willing to risk this the choice will fall among the other tools which are not offering incremental debugging. In that case I would recommend Parts for Java from Objectshare. The reason is that it fulfils the most basic requirements apart from that the code shall support internationalisation (which they say is high on their priority list). It has something more than some of the other, the visually linking.

The recommendation is Visual Café. There must be the guarantee that it works together with Clear Case, though. Visual Café is offering all the definite requirements with a reservation about that the code will be kept while regenerating. One thing among the “not that definitive” requirements that is not fulfilled is support for CORBA. That is a pity but that is the compromise. Two things that have to be evaluated are the code style of the tool and how changes made in debug mode will appear while executing the application or applet.

7 DISCUSSION

The question: which tools for Java suits Ericsson Telecom AB should be answered within this essay. So the starting point is the specific firm. To be able to answer to this it is important to know a lot about the organisation and one important question to have answered is the one raised in the beginning:

- Which requirements does the business have on a Java development tool?

In the beginning there was also other questions raised to answer to this. The questions are; how do people in the organisation work, how do they act together, which functionality is necessary and which facilities are nice to have?

There is also a question raised if programmers developing different kinds of programs have different demands on a development tool. The hypothesis is that for example the persons programming user interface do not have the same requirements as those programming communication units.

To be able to recommend one tool the following question has to be answered.

- Which tools do fulfil the requirements?

The hypothesis was that there is at least one tool, which shall suit.

7.1 The requirements

To learn which the requirements are the following questions have to be answered.

- Which requirements do the business have on a development tool for Java?
- Which functionality is necessary and which facilities are nice to have?
- How do people in the organisation work, how do they act together?
- Do programmer with experience from different kind of programming do have different requirements on a tool?

The most important requirements are that the tool is an IDE and that the generated code will be machine independent. The code shall be good structured and have comments. There shall at first be no need to modify the auto-generated code. If it is necessary, there shall be the ability to modify the code and the changes in the source code shall immediately be reflected in the UI. That there is good support from Europe is another important requirement.

Some requirements, which are not that important but is interesting to have fulfilled within the tool is that the tool support visually building on events and supply an incremental environment. This will raise the efficiency of the programmer. The incremental debugging will make the tool easier to work with for the programmer. Other things that will raise the user friendliness are that the number of windows will be optional and the size of them will be eligible. The whole list of the requirements is to be seen in chapter 6.

Every person spoken in this investigation will not recognise the requirement list as exactly the way they want to look at things. My hope is that the list will show all opinions combined. It has been interesting finding out what people really mean with what they say.

The people in the organisation are working together in projects. It is not only one person developing a program or system, and the way to work together is to leave their contribution in Clear Case, which is a version handler.

The hypothesis that the requirements on a development tool differs depending on which kind of programming the developer is doing could not be proved within this investigation. There were very few respondents to the questionnaire and that is probably the reason. Another reason could be that if a person do user interface and logical programming with the division of 50% on each, that person will have a broader view on things.

My opinion is that there must be differences in the requirements between programmers, depending on which kind of programs they are developing. The requirement of having the UI and the source code synchronised is important for a person who is programming user interface but not for a person who are programming logical parts. The person programming logical parts is perhaps more interested in having logic components generated. I also guess that a person who programs logical parts is not having the same demands on a tool. They think that they can write the code on their own. The reason for that is that there are a very few such tools on the market and it will maybe affect the person not to have that requirement.

While the result would have been that there were differences between different programmers, that will raise the question if the different demands on a tool should lead to different tools for different groups of programmers. The answer to that is considered with how people really work. Many of the respondents have told that they do a little of each kind of programming. The division of 50% on two kinds of programming is a common answer from the 16 respondents. So if it is so in the rest of the organisation the most reasonable solution would have been to find a tool, which is good for them all and not one for each, which means that the same person have to learn how to use much more tools.

7.2 The tools

- Which tools do fulfil the requirements?

The hypothesis was that there is at least one tool, which shall suit.

That is true, but the world is not that simple. The meaning with writing so many requirements on the list was not to find one that fulfil them all, rather to have the tools thoroughly investigated. There is no tool fulfilling all the requirements. So the one chosen depends on the priority point.

On the other hand there are quite a lot of tools that fulfil the most important requirements. Reading about them tells that there are no such big differences between them. And the result is that there is not possible to find the “big winner”.

In the near future there will be new ones, the competition between vendors will force them to develop new tools with more functionality.

Right now from this investigation is to be said that there are 5 good candidates; Jbuilder from Borland, Parts for Java from Objectshare, Visual Age from IBM, PowerJ from Sybase, Visual Café from Symantec. There is one more, namely Java Workshop from SunSoft Inc., which is presented among the interesting tools in the essay. Why I do not count that one here is because it does not fulfil the requirement about synchronisation between the source code and the user interface.

7.3 Choosing a tool

It would be of interest to find a tool which fulfils the most important requirements and some other nice features such as visually building of events, incremental debugging and support for standard communication in distributed environments for example CORBA and RMI.

The result of this investigation is to evaluate Visual Café. It is based on the belief that it is interesting to develop in an incremental environment and that it is interesting to have code for events generated. Not many people said that they want to have events visually built because of lack of experience. But a few did so. Unfortunately it is not supporting CORBA. But as I said earlier, the “big winner” does not exist. This is a compromise.

7.4 Generally requirements

Some of the requirements are easy to see that they are specific for the firm. That the tool and Clear Case shall be integrated is such one. Such a decision is a policy for each firm. Other requirements could be more general for most firms. I think that machine independence is such a thing that many firms would have high on their priority list. But it has not been the intention of this investigation to figure out which requirements are common and which are not.

Another thing that could be interesting for other firms is to have a version handler integrated in the tool. The most people at Ericsson were satisfied with only Clear Case. An integrated version handler will not be used to share files with other programmer. The decision is to use Clear Case to do that. But a version handler in the tool could be used to check different versions done by the programmer to go back to and so forth (even though Clear Case could be used for that too). There was no interest in that at Ericsson.

The evaluators writing in magazines about tools often talk about some things that are not interesting at Ericsson. For example that the tool does support jdk 1.0, that the debugger

is written in Java and that the tool is expressed in a tabbed formula seems to be interesting to them.

That the tool shall support jdk 1.0 is interesting for those who are writing applets for "normal" users at home. In that case the applet can be executed in all browsers. But that is no problem at Ericsson, the browsers are supporting jdk 1.1.

The reason for having the debugger written in Java, is that when the application or applet will not be fully machine independent (that it should be), then it can be debugged on the platform where it is going to be executed. No one has mentioned that requirement at Ericsson. I have missed asking about it, but I guess that it is not very important.

To have the starting point of the tool tabbed will make the screen pure and some of the tools mentioned in the investigation have that; Jbuilder, PowerJ and Visual Café. But it is not so important that it is done in that way. The important thing is that the developer can decide by him or her how many windows shall appear on the screen and the size of them.

7.5 Some finally conclusions

This investigation is done only with reading about tools, looking at pictures and a very few guided tours available on the web. I have not investigated the tools while programming anything with them myself. Next step would have been to do that. The last step investigating a tool is to test it in bigger scale in a project. The reader has to look at this investigation as a pre-study.

One thing that occur especially while evaluating a development tool for Java instead of any other language is that there is a requirement on the result being machine independent. It has taken me some time to figure out under which circumstances does it or does it not be machine independent and how the technique really works. An interesting question which I have not been able to answer is if it is true that the environment which uses incremental debugging is not fully machine independent. That is if the changes made in debug mode will behave in another way while executing.

Another question which I have thought very much about is if it does matter if the tool put in proprietary markers which tells where the programmer is allowed to write. In early versions of Visual Café this was a drawback because it followed that the programmer was abandoning the visual designer while writing outside the markers. The problem here was not the proprietary markers, rather the need to write outside them. Objectshare are using proprietary markers and this will not effect the programmer in a negative way. Between the proprietary markers the programmer is allowed to modify all the code because all the subclasses is put there. So if it matters depends on the code structure generated by the tool.

Further one thing, which is important and also mentioned in chapter 4 in the requirement list, is that the compiler shall give good messages. Good messages are message about the kind of errors and where the error occurs. Good messages are also messages, which is to prevent foolish errors when the programmer tries to write something that will probably

not lead to that the program does what it should do even though it is syntactically correct. I have missed to ask about this in the right way and have not reached any answers, which is usable in the table unfortunately.

Finally I would have liked to angle the essay much more concerning user friendliness. To read more research about that and how a development tool would be user friendly to the programmer is interesting and important for the developers. So maybe another master thesis can be more angled to this.

REFERENCE LIST

[1] Backman J (1985), Att skriva och läsa vetenskapliga rapporter, Studentlitteratur, Lund

[2] Trost, J (1993), Kvalitativa intervjuer, Studentlitteratur, Lund

[3] www.borland.com

[4] www.objectshare.com

[5] www.penumbrasoftware.com

[6] www.microsoft.com

[7] www.ibm.com

[8] www.sybase.com

[9] www.symantec.com

[10] www.sun.com

[11] www.ist.co.uk

[12] www.sgi.com

[13] Jothy Rosenberg, Javax an approachable examination of Java, Javabeans, Javascript and all the related technologies, wysiwyg://388://developer.netscape.com/library/wpapers/javax/javax.html, februari 1998

[14] Michael Carnell, Borland Jbuilder Client/Server Suite, DBMS Online, february 1998, http://www.dbmsmag.com/9802d08.html

[15] Maggie Biggs, Latest Jbuilder boasts well-integrated enterprise support, Info World Electric, http://www.infoworld.com/cgi-bin/displayArchive.pl?/98/02/rjbuilda.dat.html

[16] John Garris, Borland delivers a great Java compiler, cnet.com, http://www.cnet.com/Content/Reviews/JustIn/Items/0,118,247,00.html

[17] http://www.microsoft.com/visualtools/Egs/VisualJ.html february 1998

[18] Nancy Weil, Microsoft's latest Java move provokes outcry, Info World Electric, Mar 11 1998, http://infoworld.com/cgi-bin/displayStory.pl?980311.ehojavawar.html march 1998

[19] Tom Stearns, Lantimes Online, www.lantimes.com/96aug/608a058a.html

[20] Sun to launch Java Workshop 2.0
http://pubsys.cmp.com/iw/newsflash/nf647/0912_st5.html

[21] David Aubrey, Objectshare Parts for Java 2.0, Internetweek, 970922,
www.techweb.com/shopper/reviews/showReview?review_id=419.html

[22] PC Magazine, Java and Java Tools, Symantec café,
wysiwyg://465/http://www.zdnet.com/pcmag/features/1511/java2b.html

[23] Symantec Visual Café for Java wins 1998 New Media Magazine Hyper Award,
http://www.symantec.com/press/n980217.html

[24] Peter Coffee, IBM's Visual Age helps Java grow, zdnet,
wysiwyg://892/http://www.zdnet.com/wsources/content/0897/regrev6html]

[25] www.ist.co.uk/visaj/quotes.html

Appendix A

Hi,
My name is Anette Hermansson and I am doing my master thesis work at ETX/A/B. Brief the work is to find a Java tool that suits the organisation. To find out this I have to know which requirements you have for a good development tool, therefore I need your help. Please answer as quickly as you can, at least before February 10th. I would like the answers with name, to have the opportunity to ask more questions later in an interview.

Name:
Telephone number:

First some questions about what you are doing today.

I have used/am using Java tools. Yes/No
In that case: are you developing applets or applications?
Which tools do you use more? Mark on the list!

UIM/X	
Clear Case	
Purify/Quantify	
Workshop 21	
Netscape	
EMACS	
FrameMaker	
HP OV	
OTP	
ASAP	
Prolint	
BULL toolkit	
LoadRunner	
Xrunner	
MS Project	
MS Excel	
Sun C++ compiler	
HP C++ compiler	
Visual C++ compiler	
JDK 1.1.4	
Java Workshop	
Symantec Visual Café	
Acrobat	
Orbix	
Sun solaris	

HP-UX	
PC-NT	
APS-tools	

Do you use any other tools?
In that case which ones?

Which kind of programming are you doing?
Try to tell in percentage.

Programming graphical interface	
Write programs to handle communication units.	
Write the logical part of the program.	

Now some questions about desirable functionality by the development tool

Questions about the contents of the tool	**It is a requirement**	**It is a wish**	**It is not so important**	**It is unnecessary**
It shall be visual, that is for example window based with menus				
It shall content integrated version control				
It shall content integrated editor				
It shall content integrated debugger/tracer				
It shall be possible to work in teams with the product				
It shall content programs for editing syntax errors				
It shall be possible to regret the last thing done				
It shall content integrated picture handling				
It shall support thread handling				
It shall be possible to work with wizards				
It shall be possible to get hierarchical information about own generated classes				

Questions about the contents of the tool	**It is a require-ment**	**It is a wish**	**It is not so im-portant**	**It is un-neces-sary**
It shall distinctly evident which parts there are in the tool				
It shall be obvious which classes there are in the tool				
It shall be possible to get hierarchical information about the classes of the tool				
It shall be possible to switch between all windows and the source code				
It shall in general be easy to reach the source code				

Questions about fault-localising	**It is a require-ment**	**It is a wish**	**It is not so im-portant**	**It is un-neces-sary**
The debugger shall be visual				
The debugger shall move step by step				
The debugger shall be able to move backwards				
The debugger shall be able to travel free				
The debugger shall be able to make brakes				
It shall be possible to look for the value of a variable				
By syntax error it shall be possible to move to the error and edit it in the same window				
In the error message there shall be a time message about when the error occurred				
The error message shall tell possible solutions				
The tool shall content aids that supports the prevention from execution error				
It shall content control over if variables and classes are reached from wrong places				
It shall support control for memory allocation and deallocation				

Questions about development of the graphical interface	**It is a requirement**	**It is a wish**	**It is not so important**	**It is unnecessary**
It shall be possible to draw the graphical interface and get generated code				
When I am drawing I want to use the “drag and drop” method				
When I am drawing I want to use the “point and click” method				
It shall be possible to define methods and characteristics in the form of a table				
The generated code shall be highly effective so there is no need to add code of your own				
It shall be possible to add your own code to modify the generated one				
Your own code shall not be overwritten while regenerating				

Questions about communication	**It is a requirement**	**It is a wish**	**It is not so important**	**It is unnecessary**
The tool shall support data base treatment				
It shall support object oriented data base				
It shall support relational data base				
It shall support both way to organise the data base				
It shall content integrated file handling				
It shall be possible to send data to and from files				
It shall be possible to get components (code) from other files				
It shall be possible to translate C++ source code to Java				
It shall be possible to translate even other language to Java				
It shall be possible to import classes in Java from other suppliers				
It shall be possible to save own classes to export them to the next program you write				

Questions about the help function	**It is a requirement**	**It is a wish**	**It is not so important**	**It is unnecessary**
The help function shall content seek function				
The help function shall content examples				
The help function shall show the way while working				
The help function shall be on-line				

Questions about applet development	**It is a requirement**	**It is a wish**	**It is not so important**	**It is unnecessary**
A prototype shall be testable without going through www				
It shall support e-mail communication between the owner and the user of the side				
The tool shall support that the download time will be as short as possible for the user				

Questions about manual	**It is a requirement**	**It is a wish**	**It is not so important**	**It is unnecessary**
The tool shall be able to write a manual from the written comments				

Other questions	**Very important**	**Not so important**
The tool shall support “learning by doing” by that you can start with easy applications and build up your knowledge step by step		
The last result shall be absolutely machine independent		
There shall not be a need for great adjustments when moving from one machine to another		
It is important that the technical support from the supplier is satisfying		
It shall look like things that I have experience from		

Would you like to use the tool via your browser and load down when you need it?

Is it interesting if the tool gets input from the design phase and leave output to the test phase?

Is it anything else, which is important which I have forgotten to ask about?

Appendix B

Here are the results of the questionnaire for the group as a whole. The number of respondents are 16.

Here are some questions about desirable functionality by the development tool

Questions about the contents of the tool	**It is a require-ment**	**It is a whish**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
It shall be visual, that is for example window based with menus	7	8		1	
It shall content integrated version control		7	7	2	
It shall content integrated editor	9	3	2	2	
It shall content integrated debugger/tracer	7	8	1		
It shall be possible to work in teams with the product	4	9	3		
It shall content programs for editing syntax errors	3	6	3	2	2
It shall be possible to regret the last thing done	8	8			
It shall content integrated picture handling	2	6	6	2	
It shall support thread handling	6	8	2		
It shall be possible to work with wizards	2	6	5	1	2
It shall be possible to get hierarchical information about own generated classes	4	11	1		
It shall distinctly evident which parts there are in the tool	7	8			1
It shall be obvious which classes there are in the tool	7	8	1		
It shall be possible to get hierarchical information about the classes of the tool	6	9	1		
It shall be possible to switch between all windows and the source code	8	6		1	1
It shall in general be easy to reach the source code	9	6		1	

Questions about fault-localizing	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The debugger shall be visual	7	8		1	
The debugger shall move step by step	13	2	1		
The debugger shall be able to move backwards	4	7	4	1	
The debugger shall be able to travel free	11	2	1		2
The debugger shall be able to make brakes	13	3			
It shall be possible to look for the value of a variable	12	4			
By syntax error it shall be possible to move to the error and edit it in the same window	6	8	1		1
In the error message there shall be a time message about when the error occurred	1	5	7	3	
The error message shall tell possible solutions		7	7	2	
The tool shall content aids that supports the prevention from execution error		13	2		1
It shall content control over if variables and classes are reached from wrong places		14	1	1	
It shall support control for memory allocation and deallocation	5	7	2	1	1

Questions about development of the graphical interface	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
It shall be possible to draw the graphical interface and get generated code	8	6	1	1	
When I am drawing I want to use the “drag and drop” method	5	10	1		
When I am drawing I want to use the “point and clic” method	4	5	6	1	
It shall be possible to define methods and characteristics in the form of a table	1	8	3		4
The generated code shall be highly effective so there is no need to add code of your own	1	10	3	2	
It shall be possible to add your own code to modify the generated one	10	6			
Your own code shall not be overwritten while regenerating	13	3			

Questions about communication	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The tool shall support data base treatment	7	7	2		
It shall support object oriented data base	2	11	2	1	
It shall support relational data base	5	8	2	1	
It shall support both way to organize the data base	1	14		1	
It shall content integrated file handling	4	8	2	1	1
It shall be possible to send data to and from files	7	8			1
It shall be possible to get components (code) from other files	8	8			
It shall be possible to translate C++ source code to Java	1	8	5	2	
It shall be possible to translate even other language to Java		6	6	4	
It shall be possible to import classes in Java from other suppliers	6	9	1		
It shall be possible to save own classes to export them to the next program you write	8	8			

Questions about the help function	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The help function shall content seek function	9	7			
The help function shall content examples	8	8			
The help function shall show the way while working	2	9	5		
The help function shall be on-line	9	6	1		

Questions about applet development	**It is a requirement**	**It is a which**	**It is not so important**	**It is unnecessary**	No answer
A prototype shall be testable without going through www	7	6	2		1
It shall support e-mail communication between the owner and the user of the side	1	8	4		1
The tool shall support that the download time will be as short as possible for the user	3	12			1

Questions about manual	**It is a requirement**	**It is a which**	**It is not so important**	**It is unnecessary**	No answer
The tool shall be able to write a manual from the written comments	1	8	7		

Other questions	**Very important**	**Not so important**	No answer
The tool shall support “learning by doing” by that you can start with easy applications and build up your knowledge step by step	10	4	2
The last result shall be absolutely machine independent	12	4	
There shall not be a need for great adjustments when moving from one machine to another	16		
It is important that the technical support from the supplier is satisfying	16		
It shall look like things that I have experience from	3	13	

Would you like to use the tool via your browser and load down when you need it?

Yes:11
No: 4
No answer:1

Is it interesting if the tool gets input from the design phase and leave output to the test phase?

Yes: 11
No: 0
No answer: 5

Here are the results of the questionnaire for the subgroup which contents of the respondents which are programming user interface. The number of respondents are 4.

Here are some questions about desirable functionality by the development tool

Questions about the contents of the tool	**It is a require-ment**	**It is a whish**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
It shall be visual, that is for example window based with menus	2	2			
It shall content integrated version control		3		1	
It shall content integrated editor	2	1	1		
It shall content integrated debugger/tracer	3	1			
It shall be possible to work in teams with the product	2	1	1		
It shall content programs for editing syntax errors	1	1	2		
It shall be possible to regret the last thing done	2	2			
It shall content integrated picture handling	1	2	1		
It shall support thread handling	3	1			
It shall be possible to work with wizards	2				2
It shall be possible to get hierarchical information about own generated classes	2	2			
It shall distinctly evident which parts there are in the tool	3	1			
It shall be obvious which classes there are in the tool	4				
It shall be possible to get hierarchical information about the classes of the tool	4				
It shall be possible to switch between all windows and the source code	3	1			
It shall in general be easy to reach the source code	3	1			

Questions about fault-localizing	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The debugger shall be visual	2	2			
The debugger shall move step by step	4				
The debugger shall be able to move backwards	2	1	1		
The debugger shall be able to travel free	3				1
The debugger shall be able to make brakes	4				
It shall be possible to look for the value of a variable	4				
By syntax error it shall be possible to move to the error and edit it in the same window	2	2			
In the error message there shall be a time message about when the error occurred		2	1	1	
The error message shall tell possible solutions		2	1	1	
The tool shall content aids that supports the prevention from execution error		3			1
It shall content control over if variables and classes are reached from wrong places		4			
It shall support control for memory allocation and deallocation	1	2			1

Questions about development of the graphical interface	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
It shall be possible to draw the graphical interface and get generated code	4				
When I am drawing I want to use the “drag and drop” method	2	2			
When I am drawing I want to use the “point and clic” method		3	1		
It shall be possible to define methods and characteristics in the form of a table		2			2
The generated code shall be highly effective so there is no need to add code of your own		3	1		
It shall be possible to add your own code to modify the generated one	4				
Your own code shall not be overwritten while regenerating	4				

Questions about communication	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The tool shall support data base treatment	2	1	1		
It shall support object oriented data base	1	2	1		
It shall support relational data base	2	1	1		
It shall support both way to organize the data base	1	2		1	
It shall content integrated file handling	2	1		1	
It shall be possible to send data to and from files	2	1			1
It shall be possible to get components (code) from other files	2	2			
It shall be possible to translate C++ source code to Java	1	2		1	
It shall be possible to translate even other language to Java		1	1	2	
It shall be possible to import classes in Java from other suppliers	3	1			
It shall be possible to save own classes to export them to the next program you write	4				

Questions about the help function	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The help function shall content seek function	2	2			
The help function shall content examples	3	1			
The help function shall show the way while working	1	2	1		
The help function shall be on-line	3	1			

Questions about applet development	It is a requirement	It is a which	It is not so important	It is unnecessary	No answer
A prototype shall be testable without going through www	2	1			1
It shall support e-mail communication between the owner and the user of the side		1	1	1	1
The tool shall support that the download time will be as short as possible for the user			3		1

Questions about manual	It is a requirement	It is a which	It is not so important	It is unnecessary	No answer
The tool shall be able to write a manual from the written comments		2	1	1	

Other questions	Very important	Not so important	No answer
The tool shall support “learning by doing” by that you can start with easy applications and build up your knowledge step by step	3	1	
The last result shall be absolutely machine independent	3	1	
There shall not be a need for great adjustments when moving from one machine to another	4		
It is important that the technical support from the supplier is satisfying	4		
It shall look like things that I have experience from		4	

Would you like to use of the tool via your browser and load down when you need it?

Yes: 2
No: 1
Do not know: 1

Is it interesting if the tool gets input from the design phase and leave output to the test phase?

Yes: 2
No: 0
Do not know: 2

Here are the results of the questionnaire for the subgroup who are programming logical parts. The number of respondents are 13.

Here are some questions about desirable functionality by the development tool

Questions about the contents of the tool	**It is a require-ment**	**It is a whish**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
It shall be visual, that is for example window based with menus	6	6		1	
It shall content integrated version control		7	5	1	
It shall content integrated editor	8	2	1	2	
It shall content integrated debugger/tracer	5	7	1		
It shall be possible to work in teams with the product	4	8	1		
It shall content programs for editing syntax errors	2	5	2	2	2
It shall be possible to regret the last thing done	6	7			
It shall content integrated picture handling	1	5	5	2	
It shall support thread handling	4	7	2		
It shall be possible to work with wizards	1	6	4		2
It shall be possible to get hierarchical information about own generated classes	2	10	1		
It shall distinctly evident which parts there are in the tool	5	7			1
It shall be obvious which classes there are in the tool	6	7			
It shall be possible to get hierarchical information about the classes of the tool	5	8			
It shall be possible to switch between all windows and the source code	6	4		2	1
It shall in general be easy to reach the source code	7	4		2	

Questions about fault-localizing	It is a require-ment	It is a which	It is not so im-portant	It is un-neces-sary	No answer
The debugger shall be visual	6	6		1	
The debugger shall move step by step	11	1	1		
The debugger shall be able to move backwards	4	6	2	1	
The debugger shall be able to travel free	9	2			2
The debugger shall be able to make brakes	11	2			
It shall be possible to look for the value of a variable	10	3			
By syntax error it shall be possible to move to the error and edit it in the same window	6	6	1		
In the error message there shall be a time message about when the error occurred	1	4	5	3	
The error message shall tell possible solutions		6	4	3	
The tool shall content aids that supports the prevention from execution error		11	2		
It shall content control over if variables and classes are reached from wrong places		11	1	1	
It shall support control for memory allocation and deallocation	5	5	2		1

Questions about development of the graphical interface	It is a require-ment	It is a which	It is not so im-portant	It is un-neces-sary	No answer
It shall be possible to draw the graphical interface and get generated code	7	4	1	1	
When I am drawing I want to use the “drag and drop” method	4	7	2		
When I am drawing I want to use the “point and clic” method	5	3	5		
It shall be possible to define methods and characteristics in the form of a table	1	7	1		4
The generated code shall be highly effective so there is no need to add code of your own		8	2	2	1
It shall be possible to add your own code to modify the generated one	8	5			
Your own code shall not be overwritten while regenerating	10	3			

Questions about communication	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The tool shall support data base treatment	7	5	1		
It shall support object oriented data base	1	10	1	1	
It shall support relational data base	4	7	1	1	
It shall support both way to organize the data base		13			
It shall content integrated file handling	4	7	2		
It shall be possible to send data to and from files	7	6			
It shall be possible to get components (code) from other files	8	5			
It shall be possible to translate C++ source code to Java		9	2	1	
It shall be possible to translate even other language to Java		5	6	2	
It shall be possible to import classes in Java from other suppliers	5	8			
It shall be possible to save own classes to export them to the next program you write	7	6			

Questions about the help function	**It is a require-Ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The help function shall content seek function	7	6			
The help function shall content examples	5	8			
The help function shall show the way while working	1	6	6		
The help function shall be on-line	6	6	1		

Questions about applet development	It is a require-Ment	It is a which	It is not so im-portant	It is un-neces-sary	No answer
A prototype shall be testable without going through www	7	3	2		1
It shall support e-mail communication bet-ween the owner and the user of the side	1	5	5	1	1
The tool shall support that the download time will be as short as possible for the user	4	8			1

Questions about manual	It is a require-ment	It is a which	It is not so im-portant	It is un-neces-sary	No answer
The tool shall be able to write a manual from the written comments	2	5	6		

Other questions	Very impor tant	Not so impor tant	No ans-wer
The tool shall support “learning by doing” by that you can start with easy applications and build up your knowledge step by step	9	3	1
The last result shall be absolutely machine independent	10	3	
There shall not be a need for great adjustments when moving from one machine to another	13		
It is important that the technical support from the supplier is satisfying	13		
It shall look like things that I have experience from	3	10	

Would you like to use of the tool via your browser and load down when you need it?

Yes: 10
No: 3

Is it interesting if the tool gets input from the design phase and leave output to the test phase?

Yes: 9
No: 0
Do not know: 4

Here are the results of the questionnaire for the subgroup who does program comunication units. The number of respondents are 5.

Here are some questions about desirable functionality by the development tool

Questions about the contents of the tool	**It is a require-ment**	**It is a whish**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
It shall be visual, that is for example window based with menus	3	2			
It shall content integrated version control		2	3		
It shall content integrated editor	3		2		
It shall content integrated debugger/tracer	2	3			
It shall be possible to work in teams with the product	1	3	1		
It shall content programs for editing syntax errors		3		1	1
It shall be possible to regret the last thing done	4	1			
It shall content integrated picture handling		2	3		
It shall support thread handling	1	2	2		
It shall be possible to work with wizards		3	2		
It shall be possible to get hierarchical information about own generated classes	2	3			
It shall distinctly evident which parts there are in the tool	1	3			
It shall be obvious which classes there are in the tool	2	2	1		
It shall be possible to get hierarchical information about the classes of the tool	2	2	1		
It shall be possible to switch between all windows and the source code	2	2		1	
It shall in general be easy to reach the source code	2	2		1	

Questions about fault-localizing	It is a require-ment	It is a which	It is not so im-portant	It is un-neces-sary	No answer
The debugger shall be visual	3	2			
The debugger shall move step by step	4	1			
The debugger shall be able to move backwards	1	3	1		
The debugger shall be able to travel free	4		1		
The debugger shall be able to make brakes	4	1			
It shall be possible to look for the value of a variable	4	1			
By syntax error it shall be possible to move to the error and edit it in the same window	2	2	1		
In the error message there shall be a time message about when the error occurred		2	2	1	
The error message shall tell possible solutions		2	2	1	
The tool shall content aids that supports the prevention from execution error		5			
It shall content control over if variables and classes are reached from wrong places		5			
It shall support control for memory allocation and deallocation	2	3			

Questions about development of the graphical interface	It is a require-ment	It is a which	It is not so im-portant	It is un-neces-sary	No answer
It shall be possible to draw the graphical interface and get generated code	2	3			
When I am drawing I want to use the “drag and drop” method	1	4			
When I am drawing I want to use the “point and clic” method	1	2	1	1	
It shall be possible to define methods and characteristics in the form of a table	1	1	2		1
The generated code shall be highly effective so there is no need to add code of your own	1	3	1		
It shall be possible to add your own code to modify the generated one	2	3			
Your own code shall not be overwritten while regenerating	4	1			

Questions about communication	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The tool shall support data base treatment	2	3			
It shall support object oriented data base	1	4			
It shall support relational data base		5			
It shall support both way to organize the data base		5			
It shall content integrated file handling	1	4			
It shall be possible to send data to and from files	2	3			
It shall be possible to get components (code) from other files	2	3			
It shall be possible to translate C++ source code to Java		2	3		
It shall be possible to translate even other language to Java		1	3	1	
It shall be possible to import classes in Java from other suppliers	1	3	1		
It shall be possible to save own classes to export them to the next program you write	2	3			

Questions about the help function	**It is a require-ment**	**It is a which**	**It is not so im-portant**	**It is un-neces-sary**	**No answer**
The help function shall content seek function	4	1			
The help function shall content examples	3	2			
The help function shall show the way while working		3	2		
The help function shall be on-line	3	2			

Questions about applet development	It is a requirement	It is a which	It is not so important	It is unnecessary	No answer
A prototype shall be testable without going through www	2	3			
It shall support e-mail communication between the owner and the user of the side		4	1		
The tool shall support that the download time will be as short as possible for the user	1	4			

Questions about manual	It is a requirement	It is a which	It is not so important	It is unnecessary	No answer
The tool shall be able to write a manual from the written comments	1	3	1		

Other questions	Very important	Not so important	No answer
The tool shall support “learning by doing” by that you can start with easy applications and build up your knowledge step by step	2	2	
The last result shall be absolutely machine independent	2	1	
There shall not be a need for great adjustments when moving from one machine to another	5		
It is important that the technical support from the supplier is satisfying	5		
It shall look like things that I have experience from		5	

Would you like to use of the tool via your browser and load down when you need it?

Yes: 5
No: 0

Is it interesting if the tool gets input from the design phase and leave output to the test phase?

Yes: 5

Appendix C

ABBREVIATIONS AND GLOSSARY

GUI	Graphical User Interface
CORBA	Common Object Request Broker Architecture. This is a technique for communication in distributed environment. The applications on the client side and the server can be written in different languages. The person working on the client side does not have to bother about the objects being on the server or the client.
IDE	Integrated Development Environment. An IDE contains at least an editor, a compiler, a debugger and a hierarchical browser. It can also contain a version handler. There is a difference in having all these tools integrated compared to environments such as Unix where all these are separate programs.
JAR-file	Java Archive file. A Java Archive file includes all the files needed to run one application.
JDBC	Java Data Base Connection
JDK	Java Development Kit. An environment developed by Sun.
RAD-tool	Rapid Application Development tool.
RMI	Remote method invokation This is a technique for communication in distributed environment. The applications are written in Java. The person working on the client does not have to bother about the object is on the client or the server.
VM	Virtual Machine