

Human-Computer Debating System Evaluation & Development

Kristján Ævarsson

Supervisor: Tangming Yuan

Faculty of Information Technology,
University of Akureyri

Submitted April 2006, in partial fulfilment of
the conditions of the award of the degree BSc.

I hereby declare that this dissertation is all my own work, except as indicated
in the text:

Signature _____

Date ____ / ____ / _____

Abstract

The document describes the work on the development of a human computer debating System. This system was originally developed by Dr. Tangming Yuan and is a game that allows human user to debate against a computer. The necessary steps in the evolving process were to evaluate the system with potential users. Therefore a user evaluation was carried out on students in the University of Akureyri. The evaluation results gave information on the status of the system. In the light of the evaluation, modifications were made. A further user based evaluation was conducted afterwards, which confirm the modification made. It is anticipated that this work will contribute toward the development of human computer dialogue in general and computer based learning in particular.

Acknowledgements

I would like to express my gratitude to Dr. Tangming Yuan, of the University of Akureyri for his help, guidance, reading, correcting, commenting and support through the project. Also I would like to express my gratitude to Dr. Nicola J. Whitehead, of the University of Akureryi for her help and comments.

I will thank the students at the University of Akureyri for the participation in the user evaluation.

Thanks to my family, Anna, Pétur and Ævar, also my parents for their support.

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Project description	1
1.3	Project objectives	2
1.4	Importance & Contribution	3
1.5	Motivation	3
1.6	Overview	4
2	The debating system	5
2.1	System architecture	5
2.2	Input and output facilities	8
2.3	The game	9
2.4	Summary	12
3	The User Evaluation	13
3.1	Evaluation purpose	13
3.2	Evaluation design	14
3.2.1	Participants	15
3.2.2	Method & style	15
3.2.3	Time & script	16
3.2.4	Questions	16
3.2.5	Equipment & task	18
3.3	Procedure	19
3.4	Analysis	19
3.5	Results	20
3.5.1	User interface	20
3.5.2	System intelligence	22
3.5.3	Users feeling	23
3.5.4	System potential	23
3.6	Summary	24

4	The Modifications	25
4.1	Evaluation reflection	25
4.2	Output facility modifications	26
4.3	Input facility modifications	28
4.4	Other modifications	29
4.5	Summary	31
5	Further Evaluation	32
5.1	User evaluation	32
5.1.1	User evaluation purpose	32
5.1.2	User evaluation design	33
5.1.3	User evaluation process	33
5.2	User evaluation results	34
5.3	Summary	34
6	Conclusions	36
6.1	Objectives reflection	36
6.2	Further work	37
6.3	Project contribution	37
6.4	Conclusion	37
References		39
A	Appendix A Code Listing	40
A.1	AboutFrame.java	40
A.2	AssertionStrategist.java	43
A.3	Blink.java	46
A.4	BooleanRuleBase.java	46
A.5	ChallengeStrategist.java	57
A.6	CommitmentManager.java	58
A.7	CommitmentStore.java	60
A.8	ConflictSet.java	77
A.9	DialogueHistory.java	81
A.10	DialogueManager.java	82
A.11	Filler.java	87
A.12	FocusShiftDemolishManager.java	89
A.13	FocusShiftManager.java	90
A.14	GUI.java	91
A.15	InputGuideListener.java	98
A.16	InputManager.java	99
A.17	InterfaceManager.java	106

A.18 KBSManager.java	107
A.19 MessageDialogue.java	109
A.20 Move.java	111
A.21 MoveChoiceInitialiser.java	114
A.22 Plan.java	116
A.23 PlanSet.java	121
A.24 Planner.java	124
A.25 Proposition.java	127
A.26 QuestionStrategist.java	132
A.27 Rule.java	135
A.28 RuleProp.java	139
A.29 TacticManager.java	139
A.30 Tips.java	141
A.31 WithdrawStrategist.java	143
B Appendix B Evaluation script	145
C Appendix C Conversation between human and computer	147
D Appendix D Interview questions	156
E Appendix E Interview transcripts	159

List of Figures

2.1	Human computer debating system architecture	6
2.2	Human computer debating system user interface	7
2.3	Input facility	9
2.4	Ask the computer question	10
2.5	Withdraw commitment	11
4.1	The new user interface	27
4.2	The new output facility	28
4.3	The new input facility	29
4.4	Challenge the computer in modified user interface	29
4.5	Withdraw statement in modified user interface	30
4.6	Help text in modified user interface	30
4.7	Status bar in modified user interface	30

List of Tables

3.1	Information window result	20
3.2	User interface result summary	22
3.3	System intelligence result summary	23
3.4	User feeling of the system result summary	23
3.5	System potential result summary	24
5.1	Second evaluation result summary I	34
5.2	Second evaluation result summary II	34

Chapter 1

Introduction

1.1 Introduction

Debate is a method for people to represent their arguments. Debates are common in the western society. Politician debate about topics that involves their society, lawyers debate whether the laws have been broken or not. Debate can also be presented as a game, for example schools compete against each other. When debating, a person starts off with argument that the opponent answers, and the opponent may also give back an argument, then they go on, attacking and defending. This has similar style as chess game. In the game of chess, the computer is very powerful in playing the game, for example the IBM Deep Blue[5]. Intelligent chess playing computers are common now days. Although chess and debating is similar in structure there has not been much development of computer systems that can debate with humans.

There have been some games specified in the area philosophy of argumentation e.g.[3],[6],[1],[10]. How, the only computational utilisation of philosophical game that can be found in the literature is Yuan [11]. He utilizes Mackenzie's DC system and developed a fully functional prototype in his PhD work that allows a human to debate against a computer. The debate is presented as a game and the debate topic is capital punishment. This project intends to continue Yuan's [11] work in the development of a human computer debating system.

1.2 Project description

Yuan suggested further work on the evolving debating system that can be carried out.

In the short term, the basic human computer debate system needs to be further evaluated, to assess the user acceptability of the system, and to establish in what ways the system is effective and ineffective and why, and thus to enable iterative refinements to take place[11].

Then Yuan recommends possible means of improvements on the input facilities

A further enhancement considers the system input facilities. The current system enables users to select their inputs from a number of prescribed menu options. Perhaps, the most attractive means of communicating with the system is to use natural language.

it might be feasible to permit freer user input, initially via an option to enter fresh propositional content in addition to selecting from those made available by the system [11].

The system had been evaluated by Human Computer Interaction (HCI) experts but not future users.

During the process of arriving at this prototype, four independent HCI experts from Leeds Metropolitan University helped to evaluate and thus improve the initial prototypes [11].

Therefore the next step to evolve the system is to get information from future users. Information that are needed are: the intelligence of the system like how the computer respond to the user (game strategy), ease of use, enjoyment and the potential of the system.

This project will therefore focus on the evaluation of the system and assessment of the user acceptability of the system, establishing the effectiveness and the ineffectiveness of the system and the reason why. Then in the light of the evaluation necessary modification will be made. Special focus will be on the input facilities, like estimating the users opinion on free input, also investigating how the natural language of the users will affect there experience of the system.

1.3 Project objectives

The overall aim of this project is to evaluate and develop Yuan's [11] human computer debating system. The following objectives were set to achieve the overall aim of the project.

1. Detailed review of the human computer debating system. It is important to understand how the system work and how the game is played, because this will help the achievement of the subsequent objectives.
2. Conduct user based evaluation of the system.
3. Collect evidence, regarding the appropriate changes of the system, make modification of the system as necessary.
4. Conduct further user based evaluation of the modified system, and obtain approval on the modifications.

The evaluation are expected to give results on the system status. From that results some changes may be necessary to evolve the system. Therefore the next objective of the project was to modify the system according to the user evaluation. The intent of modifying the system according to the user evaluation results was not to change every single thing that the user would complain about. The results of the evaluation should state what is the key item that the users talked about and what is the necessary steps from that.

After modifying the system according to the user evaluation, the next step has to be, to confirm the modifications. Therefore the last objective of the project were to get information from future users about the changes. The same users have to be in the first user evaluation and the second evaluation. Because the aim of the second evaluation is to measure how effective the modification has been.

1.4 Importance & Contribution

This project is very important in the overall project of development of the human computer debating system. The current status of the overall project depends on the results of this project to evolve. Therefore this project is a one step forward to the overall project.

This project contribution can be considered as two parts. The first one is the contribution to education. This system is developed for educational purpose. Therefore any research and modification will help in developing the system such that it can be used for education. The potential system is expected to develop student's argumentation and debating skills, and development of their domain knowledge. The second contribution is to the area of argumentation and human computer dialogue. By developing the system depending on the users opinion and research on the conversation between user and the computer, this can benefit to both the computer science and argumentation.

1.5 Motivation

The major motivation for the project is to be a part of the development of the human computer debating system. It is interesting to take over other peoples work and evolve it such that a valuable contribution can be made. Having the opportunity to perform a user based evaluation and getting to know HCI studies is interesting. Before this project, my coding experience has been in developing my own programs from beginning, therefore it is interesting to work with other peoples code. The human computer debating system is interesting and it is a challenge to work on this project.

1.6 Overview

The remaining of the document is structured as follows. The first chapter provides the project description, project objectives, importance and the contribution of the project and motivation. Chapter 1 also provides an overview of the document. Background information is provided in chapter 2. The background information consists of the system architecture of the debating system, and how to play the debating game. Chapter 3 provides information about the technical aspect of the evaluation such as the design, the process and analysis method. Chapter 3 provides information about the user evaluation. The purpose, design, procedure, analysis and the results. Chapter 4 is focused on the modification that were made to the system. The modifications made was from the result of the user evaluation. Chapter 5 provides information about the second iteration of the user evaluation, then describes the purpose of the evaluation, the evaluation design and the process. Chapter 5 provides the results from the user evaluation. Chapter 6 provides the conclusions to the document, and recommendation for further work on the debating system.

Chapter 2

The debating system

This chapter introduces the human computer debating system developed by Yuan [11]. Since the evaluation of the human computer debating system is one of the objectives of the project it is important to understand how the system work and how the game is played. The architecture of the system is provided first, then information about the input and the output facilities at last information on how the game is played will be provided. The system architecture and the information about the facilities is documented in Yuan [11]

2.1 System architecture

The architecture of the system can be seen in figure 2.1. The system is divided into five main units. These units are:

- The interface unit
- The dialogue unit
- The scheduling unit
- The commitment unit
- The knowledge base unit

The interface unit provides the user interface that can be seen in figure 2.2. The user interface provides a dialogue history that records the debate. A commitment stores that show both the users and computers commitment store contents. Input facilities are provided in the user interface, the user has to select from the two drop down box's the move type and the move content to generate an input. The interface manager holds the record of the debate history and provides help facilities.

The dialogue unit as the transmission center of the dialogue interactions. The unit consists of input manager, a dialogue manager, a referee, and an output manager. The input manager

provides support for the user input and deliver the user input to the dialogue manager. The dialogue manager controls the turn taking of the interaction and is in charge of the input manager, output manager, referee, the commitment unit and the scheduling unit. Every move has to go to the referee for validation. If the referee accepts the move the commitment manager is called to update the commitment stores, and the output manager update the history. The scheduling unit is to make a move on behalf of the computer. If the move is illegal, the referee notifies the user by requesting another move.

The scheduling unit is responsible for generating the computers dialogue move. The commitment unit updates the users and the computers commitment stores. And he goes on to explain the knowledge base unit. The knowledge base unit consists of a knowledge base manager and a dedicated system knowledge base.

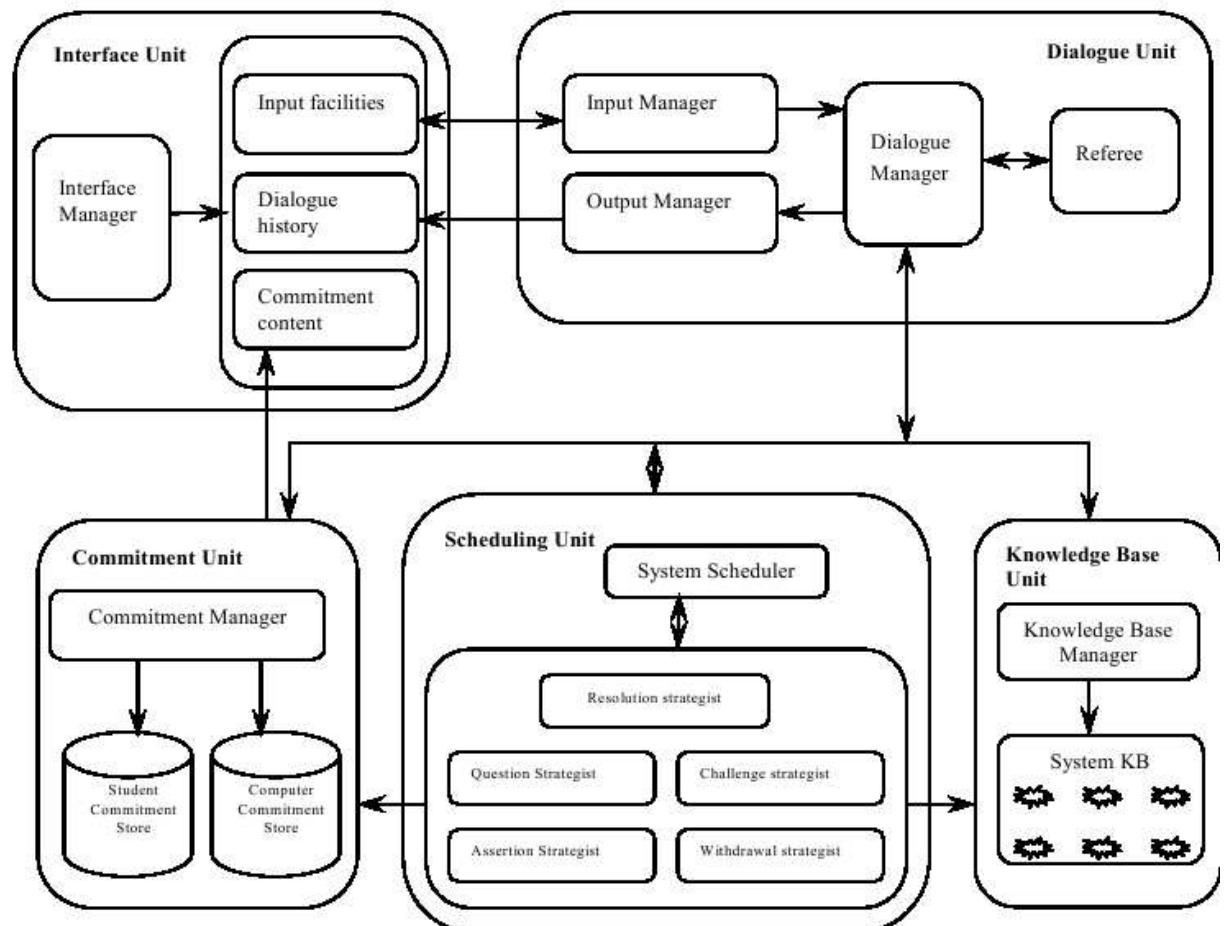


Figure 2.1: Human computer debating system architecture

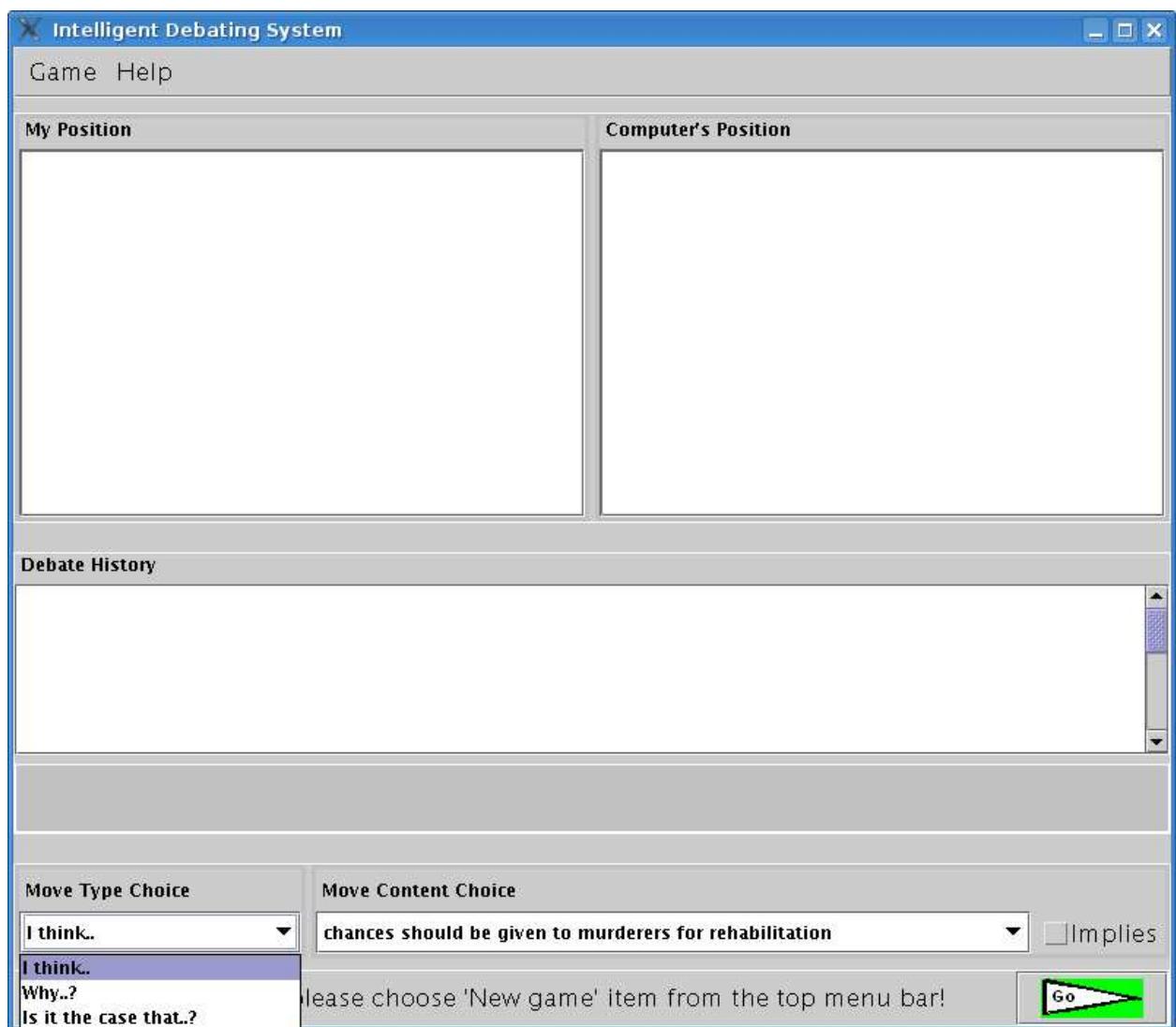


Figure 2.2: Human computer debating system user interface

2.2 Input and output facilities

The input facility is menu driven, menu driven input facility can aid the system such that the debate get more interesting[7]. Menu driven input facility require that the user make a double selection. That is available move types and list of prescribed propositions:

- 1 When the computer asks a question, there are three moves available. *Yes, no and I am not sure about it.*
- 2 When the computer makes a resolution demand, there are two moves available. Withdrawal of either of the conflicting statements.
- 3 When the computer make a resolution demand after the user has challenged or withdraw. There are three moves available. Withdrawal of either the conflicting statements or affirmation of the disputed consequence.
- 4 When the compute make a challenge, then *Because*, and *I don't know why* are available.
- 5 When the computer has withdraw a statement. The move types question, challenge and withdrawal are available.

When a move type has been chosen, the user need to select a propositional content. The information about the propositional content depending on the nature of the move type:

- 1 The move for resolution demand and challenge can be selected from the computers commitment store.
- 2 The move contents for withdrawal can be selected from the usurers commitment store.
- 3 The move contents for assertion and question can be selected from the list of propositions (by selecting the *implies* check box).

The systems output facility concerns of the generation of argumentation text in English. Then he goes on

The ingredients of the argumentation text are a set of move types and a set of propositions. Essentially, the output manager of the system will receive a *move* object from the system scheduler and then generate the argumentation text. A *move* object is designed to have three attributes: turn, move type and move content. The turn taker could be the user or the computer or the referee. The move type might be an *assertion*, a *question*, a *challenge*, a *no commitment* or a *resolution demand*. The move content could be a proposition, a conditional, or a conflict set which might contain a set of propositions or conditionals. A conditional is expressed as the connection of two propositions with the *implies* or a *prima facie reason for* link. A conflict set is expressed as each element (proposition and/or conditional) of the set with the word *and* link.

2.3 The game

The human computer debating system is presented as a game. Were human can debate against a computer. The computer ask the human in the beginning if *capital punishment is acceptable*, the computer then adopt the opposite opinion, depending on what answer the user gave. For example the user may choose that capital punishment is acceptable then the computer will argue that capital punishment is not acceptable. The debate go on until either party has adopted the opponents opinion.

To illustrate how to play the game an example will be provided. This example provides information on how to use the facilities, both the input and the output. To start the game the user has to go to the *Game* menu at the top of the screen and select *new game*. After starting the game the computer asks the question if capital punishment is acceptable.

01: C>Is it the case that CP is acceptable?

In our example the user answers by

02: S>Yes, I think CP is acceptable.

When the user choose if he think capital punishment is acceptable or not, he choose from the left drop down box called *Move Type Choice*. The next step is that the computer prompt the user with her opinion

03: C>I think CP is not acceptable.

Then the user can select from four possible move type choice. First the user select *I think..* then by selecting that type of move the user has to select a content, in the right drop down box. Figure 2.3 show the two drop down box's. The next move from the user is

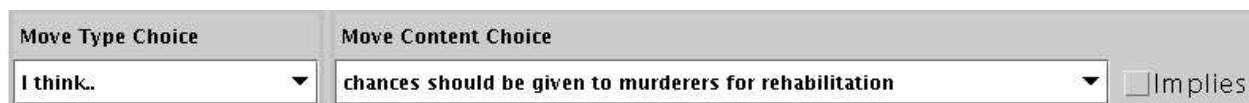


Figure 2.3: Input facility

04: S>I think CP is a good deterrent.

The computer respond with

05: C>But CP having countries like USA has higher murder rate than UK.

The user now selects the *is it the case that..?* then the content, asking the computer the question

06: S>Is it the case that execution of murderers is fair for the people being murdered?

The computer responds with

07: C>Yes, I think execution of murderers is fair for the people being murdered.

The user then select the type *Why is it the case that..?* and now can select from the computers commitment store as seen in figure 2.4. and asks the question

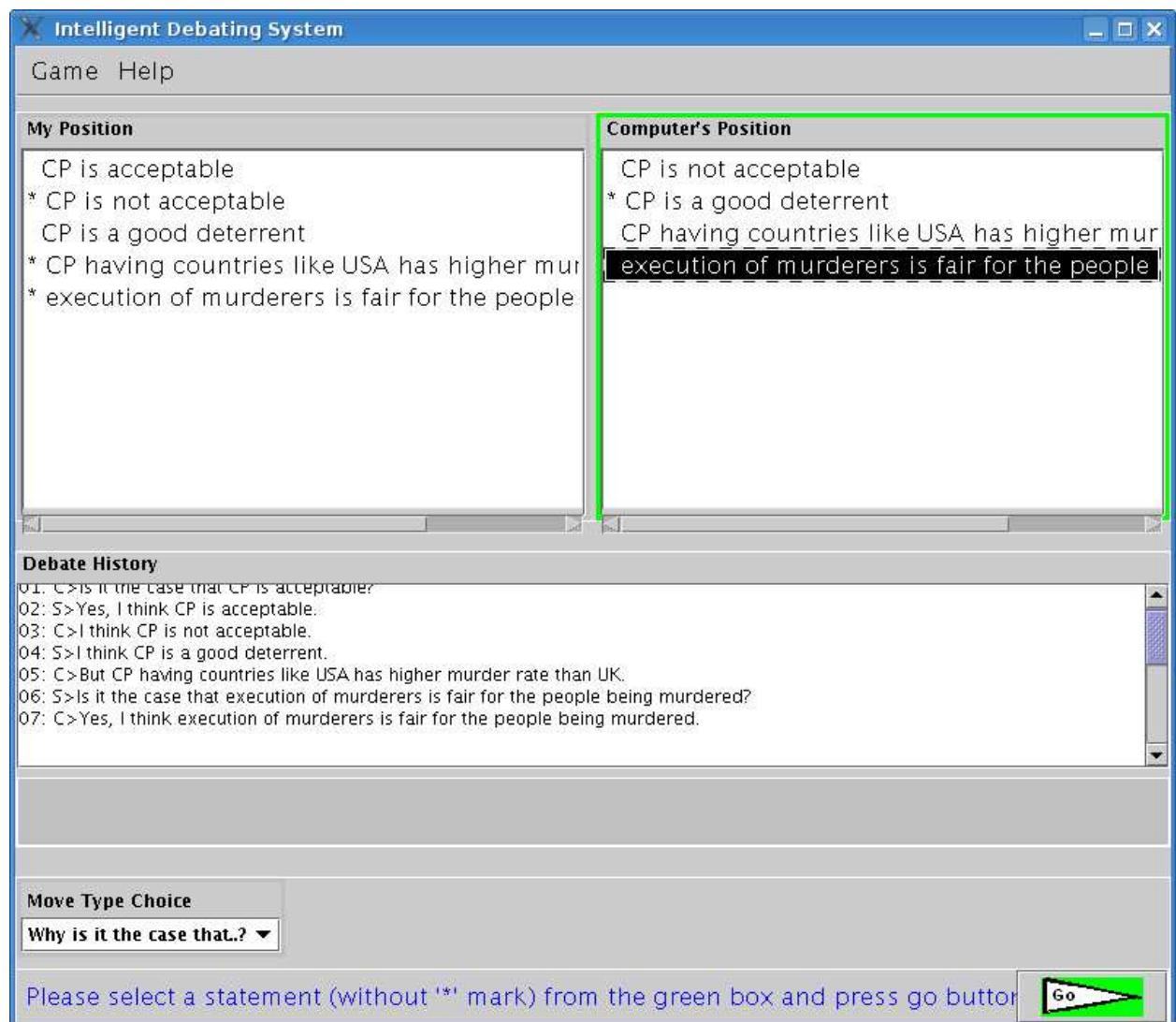


Figure 2.4: Ask the computer question

08: S>Why is it the case that execution of murderers is fair for the people being murdered?

In the computer commitment store the statements without the * icon is the computer commits in the game so far. The statements with the * icon are from the human user. The computer responds with

09: C>I don't know why execution of murderers is fair for the people being murdered.

So the computer don't know the answer to this question. Then the user selects the type *I don't think* and then selects from the human commitment store as can be seen in figure 2.5.

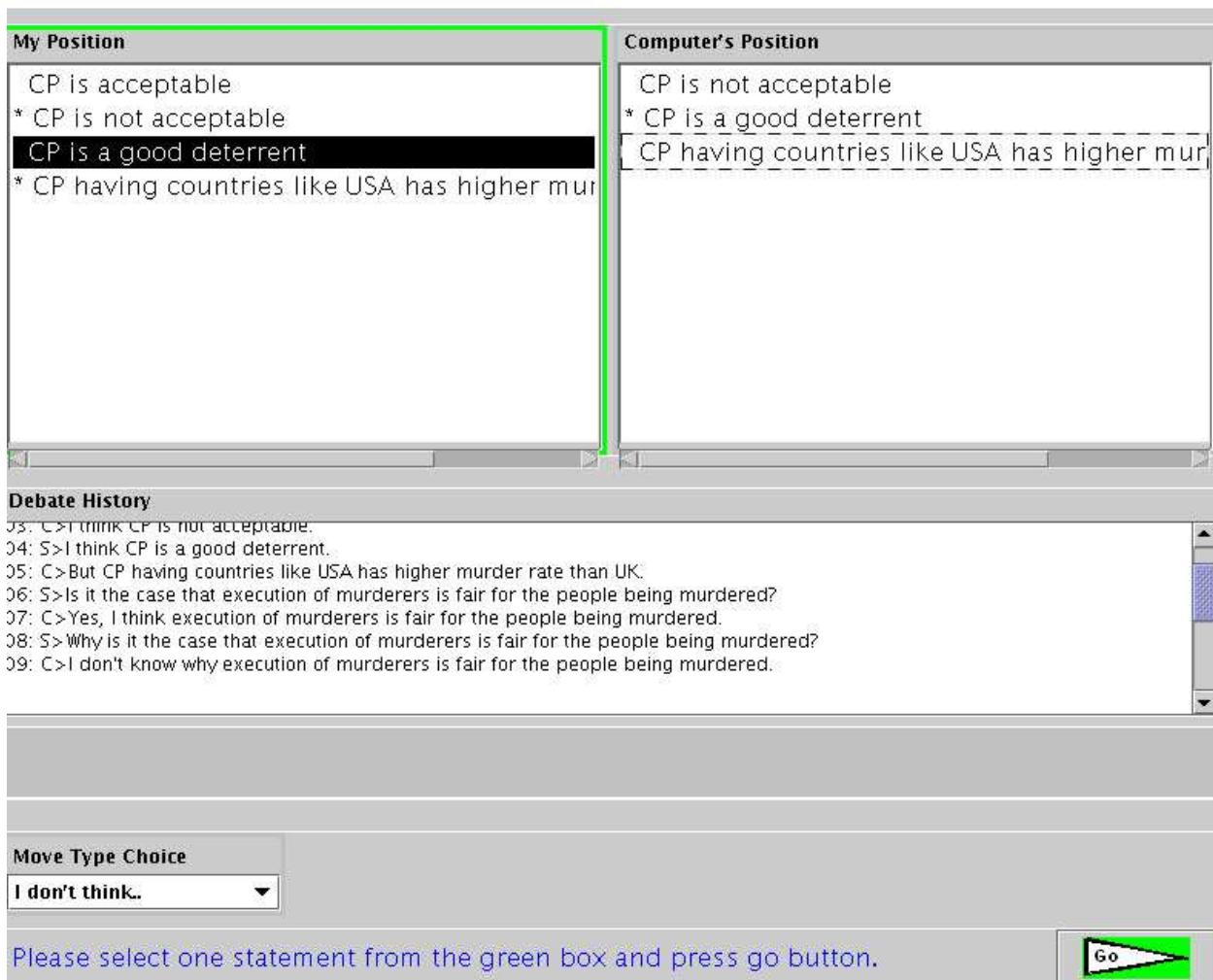


Figure 2.5: Withdraw commitment

He can only select the statements without the * icon, because the statements with the icon

are generated by the computer, and therefore he can not withdraw something that he did not argue. With this option he withdraws previous input that he had made.

10: S>I don't think CP is a good deterrent.

The computer responds with a question

11: C>Is it the case that CP is acceptable?

asking the user if he has change his mind. The user has not change his mind and select *Yes*, telling the computer that he thinks CP is still acceptable. One option is to use the *implies* check box. The user select the move type *I think..* and the content check the *implies* box and select another content from the right drop down box (Move Content Choice).

16: S>I think 'CP is a good deterrent' implies 'CP is acceptable'.

The computer respond by

17: C>I think it is wrong to take a human life.

This example describes the possible move types the user can select from, and how the game is played. The debate then continue until either the computer or the human user change there mind.

2.4 Summary

This chapter discussed the five unit architecture of the system. The main component of the system, the input and the output facilities were discussed and exemplified. As discussed before in this document the objective is to evaluation the system, to do that a potential user would be tested playing the game as shown in the example above. The next chapter will discuss the evaluation of the system.

Chapter 3

Evaluation

This chapter documents the user based evaluation of the human computer debating system. This part of the project is a critical and the most time consuming. Information about the design and the setup are provided in this chapter. Then information how the data from the user evaluation were analysed, also the results from the user evaluation will be presented.

3.1 Evaluation purpose

The purpose with the project evaluation is to assess the system and test it so it meets the user requirements. Or in fact to measure the users requirement, because this is the first evaluation on future users. Then the evaluation can give information on what the users require.

To assess the extent and accessibility of the system's functionality, to assess users' experience of the interaction, and to identify any specific problems with the system[2].

User testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested[8].

Since user testing is the fundamental method a user evaluation were conducted. A test plan was made beforehand to get overview of the issues need to address:

- The goal is to get information from the future users about the system, to establish in what ways the system is effective and ineffective.
- The test should take place as soon as possible, and should take place were the tester will be comfortable, were he is familiar with the environment. Such that the environment will not have affect on the testing.

- Each session should start with introduction from test script and then 15 minutes in testing the system. This time can changes depending on the pilot study.
- Computer that can run Java software are needed, also needs to be connected to the Internet. The screen should be at least big enough that the interface of the system fits easily.
- The system needs to be fully functional, therefore testing on the system need to be done.
- The computer that the tester use should have memory above 64 megabyte to prevent any technical failure.
- Definition on future users have to be clear so that right participants are used.
- What method to use of getting the participants.
- Get information about the necessary number of participants from HCI textbooks.
- Decide how the task in the evaluation should be.
- The use of on line dictionary will be provided to aid the testers because it is most likely that the testers native language will not be English.
- The experimenter will not be allowed to help the tester in any why other then reading the evaluation script that introduce the necessary information to the tester.
- The conversation between tester and the computer will be saved
- To get information from the testers a interview will be conducted.
- The interview will be recorded with a audio recorder.
- The questions for the interview should be focused on the interface, system strategic, ease of use and the potential use of the system.

The test plan gave clear ideas what would be needed for the evaluation, therefore the design process aimed to solve these issues.

3.2 Evaluation design

The design of the user evaluation are split up to participants, method & style, time & script, questions and equipment & task. Each of these parts are discussed below.

3.2.1 Participants

When defining who the future users are, it is necessary to realise the purpose of the system. It is aimed to aid education, for example such that students can train there dialectics. Then the biggest group of future users will be students. Since this project is a part of final year project in the University of Akureyri. Students in the University of Akureyri will be the group that participants can be chosen from. The majority of the university students are Icelandic, and therefore have Icelandic as there native language. The requirement that the participant has to fulfill is that he is willing to participate according to the time frame and is interested in debating. It is important that the participants are interested in debating, because the aim is to have future users, so participant that is not interested in debating is not likely to be in the group of future user. The requirement that the tester would have native language as Icelandic was decided, such that there would not be any conflict of language in the evaluation. Also to investigate how user that does not have English as there native language would like the input facility and how the idea of free input.

The method used to get participants to participate was controlled. The experimenter setup the evaluation in Sólborg the main building in the University of Akureyri, then went out to the hallway randomly picking students asking them if they would like to participate and if they had interest in debating. The experimenter also looked for participants in the library, located in the same building. The controlled part of the recruit is that, the experimenter tried to get participants from as many faculty's as possible. The same method was used in Borgir (which is other building at the university), as in Sólborg.

The number of participants were decided based on HCI studies on evaluation. Nielsen and Landauer argued that testing with one participant result in finding a third of the usability problems, and that there is little to be gained from testing with more than five[9]. This suggestion on number of participants involved observation. Using five experts is good practice, and when using non experts the number should be doubled[2]. Therefore ten participants would be as testers in the project evaluation.

3.2.2 Method & style

The evaluation style used was laboratory. Laboratory studies are described as

users are taken out of their normal work environment to take part in controlled test, often in a specialist usability laboratory (although the 'lab' may simply be a quiet room)[2].

The reason for laboratory studies is that observing testers in there working environment would not be affective for the evaluation. The type of data measurement from the evaluation have to be qualitative rather than quantitative. The reason for qualitative is that detailed information are needed that would be hard to get from numeric data.

To get this qualitative information as aimed for, a interview method was used. Interview and questioner are methods that studying how users use system, what they like or dislike. Questioner is better suited for quantitative measure,

Interviews are well suited to exploratory studies where one does not know yet what one is looking for, since the interviewer can adjust the interview to the situation. Interviews typically include many open-ended questions where users are encouraged to explain themselves in depth, often leading to colorful quotes that can be used to enliven reports and presentations to management [8].

Interview is what suits this evaluation best, giving tester the freedom of expressing their opinion and ideas, that will be valuable information in the continuous development of the system.

3.2.3 Time & script

To determine how much time spent on task in the evaluation, the time 15 minutes was chosen. The reason for 15 minutes were just a blind guess. Therefore to determine the length of the experiment a pilot test had to be performed, two pilot tester are needed[8]. The reason is that it is the first time that the experimenter performs a user evaluation. So two pilot tester can give more information about the test plan, rather than one.

A test script was made and can be seen in appendix B. This script is an introduction to the evaluation. In the script there are information about the project and what the purpose of the project is. Because the purpose of the project is to evaluate the system, the script states that the tester part is very valuable. Then discuss about the anonymity of the evaluation. Such that when the conversation is saved it does not involve the name of the tester, also about the interview, the name of the tester will not be linked to the interview audio file. And states that there are no observations going on while testing. The task and the time to solve the task are introduced. Explanation about the equipments, such as the computer, audio recorder and the online dictionary provided. The testers were told that they are free to stop at any time.

3.2.4 Questions

The interview design focused on how to structure the questions and what to ask, then what method to use when recording the interview. The questions were split up into four groups. Each group focuses on specific parts of the system. The groups are:

- The interface
- System intelligence
- Feeling (liking)

- Future use

The interface group will look at the pros and cons of the user interface. Special attention is on the input and output facilities. The Second group is the system intelligence. This group focus on the system strategies, and are to determine if the system can debate with human such that the user think that the system is intelligent enough to be worthy opponent. The group feeling is how the tester like the system in general. Do they enjoy playing and would they like to have access to a system like that. The last group is about the future use of the system. The idea about this group is to see if the tester see potential for the system to become more than prototype. Also to gather information on what kind of use the system can be expected to. By dividing the question up to this groups and line them up such that for example the first five question is about the interface then the next five is about the system intelligence and so on. It will be easier to analyse the data, and give the flexibility of analyse each group at the time. The questions asked can be seen in Appendix D. The questions are in Icelandic, therefore a loosely translation will be given on the main questions. Each question has a follow up depending on the answer that direct the interview to a conversion. Therefore the questions were open-ended questions. These follow up questions can be seen also in Appendix D. The questions are as discussed before in four groups. The questions were:

- A question about the information window that appears when the program starts. The aim was to find how the tester would rate the information window. If it helped, and also to see how users react to help like this.
- A Question on how difficult it was to start the game. And were the tester got the information how to start the game.
- A question about the Go button that is to confirm the input from user were asked.
- One question about the help text that blinks at the bottom of the screen were asked, to determine how much that text help in the game.
- A question on how the 3 windows in the system were rated by the testers.
- Questions about the input facility were asked. How the testers liked to select from these two drop down boxes. Then if they would like to have free input, for both English and Icelandic.
- How the tester liked the access to information and settings.
- How fast the computer responded.
- How intelligent the computers response were. Followed by a discussion on the computers strategic.

- How the tester liked the system.
- Would they use it for fun.
- Did they think that this system would be aid in education. If so how they did see that, with example from there experience.
- How much numbers of debating topics matter.
- A question about the user getting more control over the system, so it would be possible to add and modify the debate topic.
- A question focused on the language part.
- One tricky question at the end about how certain the tester was if he was talking to computer. The reason for this question was to get feedback on what the tester thought about the computers answers. If they were similar to what a human would answer.

The first question is about communication programs like MSN for example. The tester were asked if they had used such program before. If they had, there were asked to give some comparison on communicating with other person and with the human computer debating system. At the end of each question group the user were free to add comments if they wanted.

3.2.5 Equipment & task

The experimenter provided his own computer to the evaluation. Because the test was performed in the testers environment, therefore a laptop computer were desirable. That solved also many issues like, having Java installed, connection to the Internet, and the screen is a 12 inch so the interface can easily fit into that size of screen. The memory of the computer is one gigabyte so there should not be any problems about memory.

To record the interview a audio recorder was used. The reason for the use of audio recorder was that the experimenter had access to such equipment. A MP3 player with the capability of recording voices was used. The idea of recording the interview or even the test it self on video came up. But the the equipments are hard to get and the time to setup and analyse the video records were limited.

The task that the tester were asked to perform was to start the game and play against the computer in debating about capital punishment. The system had already be started up, so the tester would not be tested on if they could start the system, that information was thought off as not relevant for this evaluation. The experimenter should not help the tester when he was playing. So the only information that the tester got was from the evaluation script that can be seen in Appendix B.

3.3 Procedure

The user evaluation were performed in two locations. In Sólborg and Borgir, these are buildings at the University of Akureyri. The first step was to setup the evaluation. Class room were used in both of the buildings to perform the user evaluation. The class rooms are quiet and no other person was in the room. The computer were placed on a table with the system on the screen. An on line dictionary were provided to the users, therefore a web browser with the English-Icelandic on line dictionary that are available at <http://www.ordabok.is> was open, but minimize and visible in the task bar.

After the setup and locating participant. An introduction the evaluation script was read to the tester, introducing the necessary information. The tester had first 15 minutes to carry out the task. After the first pilot study the time was extended to 20 minutes. The task was to start the game and debate against the computer.

When the task had been carried out an interview was performed. Where the experimenter asked the tester questions. Before the interview a audio record were introduced and started to record the interview. After the interview an interview transcript were made, such that would be better to analyse the interview. The interview transcripts can be seen in Appendix E. Next section introduce the methods used to analyse the data from the user evaluation.

3.4 Analysis

There were two methods to analyse the data gathered from the user evaluation. With tools like analysing software, or manual analysing the data. The second method were chosen, that is to analyse manually. There was limited time, and by using tools to analyse the data, would entail time learn to use the software.

The next step was to determine how to analyse the data. There are three methods to do that. First to take each interview and analyse it separately. The other one is to analyse each group of question (interface, intelligence, feeling, future use) not depending on tester. The third method would be to analyse each question one by one. The second method was used, to analyse depending on the four question groups. Such that the focus would be on one group at the time, gathering information from all the interview transcripts.

The preparation was to cut all the interview transcripts into four groups. Then for each group there would be ten papers snippets. One group at the time were analysed, first there was the interface group with information about the user interface. The ten papers snippets were lined up on a table. In the first round all the statistical data were gathered. Then for the next round each question in the group would be analysed in detail. Connecting answer to other testers answer to find pattern. All findings were noted down on a separate paper. All the comments were noted down, and if there were comments about items that was in other group, all of those comments were noted down on a separate paper. This technique was used for all the groups in same manner as discussed above.

All the papers that were gathered to produce the final results of the user evaluation that will be discussed in the next section.

3.5 Results

This section provides information on the user evaluation conclusion. The method used to analyse the data from the user evaluation as described in previous section were to divided the data according to the four groups. The conclusion of the evaluation will be presented in same manner. First conclusion about the interface, then the system intelligence, users felling for the system and the system potential.

3.5.1 User interface

When the system starts a information window appear on the top of the user interface. This window has tips for the user how the can play the game. 90% of the testers liked to have window like that pop up in the beginning. Although they liked to have this window, only half of the tester read it all. 70% of the tester closed the window and the other 30% minimized the window. No one checked or read over the information window again. The results are summariest in Table 3.1 There were some comments about this window. The testers liked

	Yes	No
Liked	90%	30%
Read	50%	50%
Closed	70%	30%
Revisit	0%	100%

Table 3.1: Information window result

the information provided, but some would also like to have a tutorial with examples showing screen shots to aid. The testers would only see the information window once or twice, and after that it would be annoying if the window would appear above the user interface. The testers liked the width of the window, the reason for that is that the user did not have to move his eyes when reading the text from the window. From the information gathered about the information window from the user evaluation it would not be good to display the information window above the user interface. It looks like when the user are given information he ignores them. Therefore it would not affect the game if the information window would just be in the help menu. Comment from tester when asked if he could find a help window in the system, he responded that it would be no problem because the menu is just a standard menu.

All the tester had no problem with starting the game, and 70% of the testers got help from the blinking text at the bottom of the screen to start the game.

To send an input, the user has to press a button. This button is in the bottom right corner on the screen. The testers were positive on confirming there input with pressing a button. It gave them flexibility to change what they had selected, reducing mistakes. Although they liked to confirm there input, they did not like the button it self. Many comments were about the position of the button. And some tester pointed out that the button should be closer to the input facility. Then comments about the look of the button, like the size, color and the image on the button was something that the testers did not like.

The blinking text at the bottom of the screen, is to aid users in playing the game. Telling what is the next step, displaying guidelines for the user. This method of providing information to the user was very successful. All of the testers found that the text was helpful, and as can be seen when the testers started the game. 90% of the testers did use the text to get help when they were playing. But some comments were made about how fast he blinks, and how he changes colors were a problem for one tester.

In the menu there are options like changing colours in the debating history window and view the knowledge base. Only one tester did actually notice this settings.

Then the real problems started when discussing the 3 main windows in the output facility. Most of the tester time spent in the user evaluation were to figure out where the conversation took place. Was it in the human commitment store or in the computers store, or could they just focus on the debating history window were comments from the testers. Sometimes the user has to select from the human or the computers commitment stores in the two top windows, so by doing that it would confuse them. Also the history window did not auto scroll so the tester had to scroll manually. In the beginning of each interview the tester were asked if they had used communication program like MSN to chat with other people. 90% of the tester had and use programs like that. Therefore many of the tester commented that they are used to chat in just one window. So they would like to see just one window like in MSN.

The next real problem was the input facility. Half of the tester or 50% did not scroll down in the right drop down box the *Move Content Choice* as can be seen in the conversation log in Appendix C. Therefore they just used six to seven statements to select from. The other tester that did scroll down in the drop down box, exploring every possible selection of statements did not like the *Move Content Choice* drop down. There is to many items in the box so it was diffult to search. To inform, there is 37 statements in this dropdown. When tester had to select from the commitment stores they did not understand why there were star * icon infront of a statement. Therefore they avoided to select from this two windows. One tester came up with the idea of colour the sentence that have the star * icon, that would be better way to recognise thouse statements. The input facility had to many items in the other drop down and selecting from more then just one place input bothered the testers.

When the tester were introduced the idea of typing in there one sentences from a command line, only 30% of the tester liked that idea. The others were confortable with the menu method. But if the system would be in the testers native language (Icelandic) 70% of them would then like to type in there one input from a command line. Still 30% did not like that.

General comments about the user interface was that it is ugly, dull, no colours and there are no fun in it. The testers did not like the user interface. And the interface was the reason the testers could not fully play the game.

Summary of the evaluation result of the user interface can be seen in Table 3.2

Information window	No need to display the window at start up
Start game	easy, blinking text helped
Go button	Good to confirm, bad location and look
Blinking text	Very good, blink to fast, to many colour
Settings	Not important
Output facility	Not good, confusing, one window required
Input facility	To many item in drop down, * icon confusing
General	Ugly and dull

Table 3.2: User interface result summary

3.5.2 System intelligence

The disappointment part of this project was that not enough information could be gathered about the systems intelligence. By system intelligence it is how the system play the game, like the strategic. The tester were so distracted from the user interface that in depth analyse on the systems intelligence were not possible. The analysis on the system intelligence suffered from the time spent on figuring out where the conversation took place and half of the tester did only use small part of the input possibility's. One comment from a tester when asked about the system intelligence (translated from Icelandic) "I never reach that kind of speculation".

All of the testers agreed that the computer responded very fast, so no one had to wait. Also every one agreed that the system were intelligent such that it was a worthy opponent. Discussion about how aggressive the system is, was made and some of the tester would like to see the system more aggressive and attack more.

When the tester were asked if they thought that they have been talking to computer or human. No one thought that it was human that they were talking to. But half of the tester thought that it could be human they were talking to, but they were opt for that it was computer. The other half was sure that it was computer. The reason for this finding from the testers were often because how fast the system responded. It is therefore fair to say that the half of the testers look at the system as so intelligent that it could be human that they were debating against.

Summary of the evaluation result of the system intelligence can be seen in Table 3.3

Speed	No one had to wait
Intelligence	Very intelligent
Mode	Attacking more

Table 3.3: System intelligence result summary

3.5.3 Users feeling

The group, user feeling is how the user like the system. And the results from that group were in favour of the system. Only 2 of the tester did not like the system, and the reason was how difficult it was to play the game with this user interface, like selecting input. The other enjoy playing, therefore no one disliked the game and the idea of debate against the computer.

All of the testers agreed on adding another domains or debating topics would be very valuable for the system. But they did not like to have the control of adding domain or manipulate the rules. The uncertainty of the system response were interesting to the testers.

One idea came up from the evaluation. That is, to be able to see how the user is doing in the debate. To have indicator if the user is for example in better position then the system an vice versa.

The tester would like to see the game as a Internet game were they could have access to the game. 90% of the testers would like to play the game again after some improvements.

Summary of the evaluation result of the users feeling can be seen in Table 3.4

Liked the system	80%
Reason for disliking	User interface
Domain	Adding new domain would benefit the system
Control	Do not want control over the system
Idea	Status indicator
Play again	90%

Table 3.4: User feeling of the system result summary

3.5.4 System potential

The group system potentials is how the user see the future potential for the system. And the result from that group were in favour of the system. Everyone mention the potential for the system to be an aid to education. An idea came up from one tester that is a student in the business faculty. In his faculty there are module called dialectics. He could see this system as an aid for that model. Then other potential for the system came from tester from the

faculty of education. The tester would like to see the system tailored for kids to aid them in many areas. Also to help people in practicing augmentation and just a game to have fun with.

Summary of the evaluation result of the system potential can be seen in Table 3.5

Potential	Aid in education
Tailored	For kids
Practice	Aid in argumentation
Game	for entertainment;

Table 3.5: System potential result summary

3.6 Summary

This chapter discussed the user evaluation. The user evaluation was designed as a laboratory, gathering qualitative data through interviewing the testers. The testers were students in The University of Akureyri. An interview transcript were made to analyse. The analysis were done by hand and information were gathered to evaluate the system. Also in this chapter were discussion about the user evaluation result. The user interface was so confusing to the testers, that they had a difficult time playing the game. The input facility and the output facility were the most confusing parts of the user interface. Therefore it is clear that changes to this two components are important. The tester could not value the systems intelligence in any depth because of how the user interface were bothering them. But although they confirmed that the system is intelligent. How the tester liked the game and the idea of debating against computer very in favour of the system. Also the potential for the system were something that the testers discussed. From the user evaluation result it is apparent that the user evaluation needs to change. The next chapter discuss and inform the modifications made to the systems user interface.

Chapter 4

Modifications

This chapter introduces information about how the system was modified. The modification on the system are one of the project objectives. All the changes made on the system were according to the user evaluation result discussed in previous chapter. Before the modification, decided was to keep the same layout on the user interface, also to use the same programming language, the programming language is Java. There were no comments about the layout from the user evaluation, therefore the same layout was kept. What parts of the user interface can be changed will be discussed in this chapter. Then the changes made on each part of the interface will be discussed.

4.1 Evaluation reflection

From the user evaluation result there was problem with the user interface. The main problems were the output and the input facilities. Therefore changes to those two parts were critical.

According to the user evaluation the output facility has to be in only one window. Such that the conversation takes place in one window and there should not be required selection from the output facility when generating an input.

The testers from the user evaluation did not like the input facility. From the user evaluation result it can be seen that only half of the tester did scroll down in the drop down list with the move content choice. Therefore a new representation on the input facility is needed.

The *GO* button, that the user press when confirming his input, has to change. From the user evaluation the problems with the button were the location and the look.

The blinking text at the bottom of the screen that displays information to aid users in the game had to be changed. Such that he would not blink so rapidly. Also change how many colour there are in the text. Now the text switch between three colours.

An indicator have to me added to the user interface, that displays the status how well

the players are doing. The functionality of the indicator were ignored. Because to add the functionality to the indicator a function have to me implemented that determine how well the players are doing. Such function value function is hard to implement into the game and when designing such a function the dialog studies have to explored in details.

While modifying the system, a clean up on the code should be made. Structure the layout of the code, implement methods to solve common tasks to reduce duplication of code and give clear structure on the system.

The changes needed on the system can be summarised as:

- Output facility
- Input facility
- Go button
- The blinking text
- Add indicator
- Clean and structure code.

After changing this parts the new user interface can be seen in figure 4.1. The next sections discuss in detail what was changed and how.

4.2 Output facility modifications

The problem with the output facility is the three windows. The aim was to change the output such that only one window wold be used for the conversation. Similar to communication programs like MSN. The human position and computer position windows at the top of the user interface were removed. These two windows were confusing to the testers in the way that they did not recognise the debate history window as the conversation window. When challenging the computer or withdrawing position a selection had to be made in either of those two windows. This selection were moved to the input facility since the selection is a part of generating an input.

The debate history window is now the only output window. The history window was changed so it would be a JTextPane component. JTextPane is part of the java swing library, and use vectors for input. Vectors in Java is expandable array. JTextPane gives the flexibility of varied styles. A StyledDocument have to be added to the JTextPane to be able to use the style features. The StyledDocument is also a part of the Java swing library and style with StyleConstants was added to the StyledDocument. The styles added was so the use of colours were possible in the debating history window.

The use of colour were used to colour code the owner of each sentence in the output. Human was coloured with blue and the computer with red. By using colour it can aid the

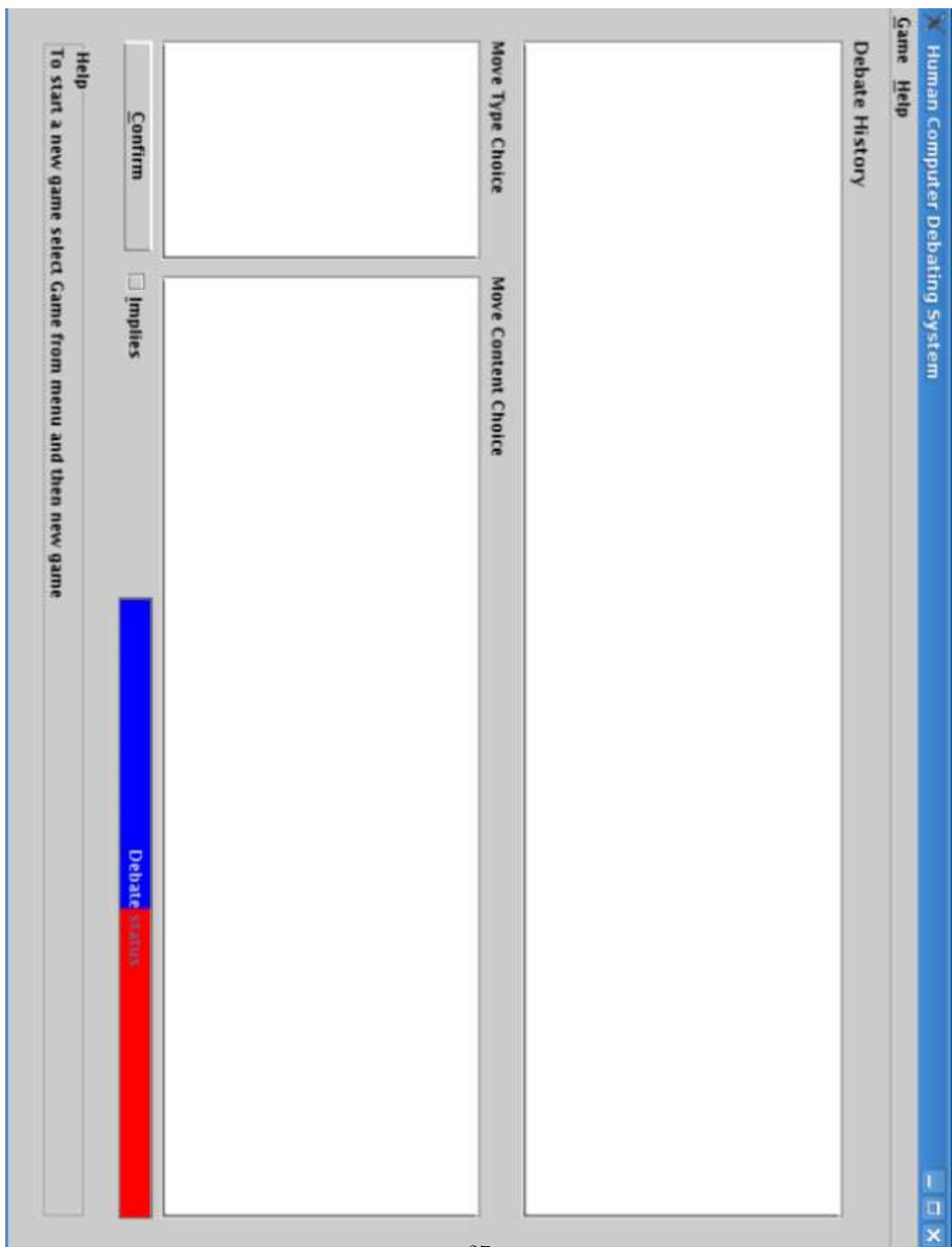


Figure 4.1: The new user interface

user to find what sentence belongs to him or the computers. In the old debate history box the computer were indicated by *C* and the user as *S*. For example this line is an output made by the user

04: S>I think CP is a good deterrent.

The indication on the owner of a sentence were made so that the full name of the owner is displayed. Now the computer is *Computer* and the user is *Human*. The same example as above is now

04: Human>I think CP is a good deterrent.

The window did not scroll down when the output had filled up the box and a new sentence were added. Therefore the new window auto scrolls with the newest sentence added. The new output facility can be seen in figure 4.2

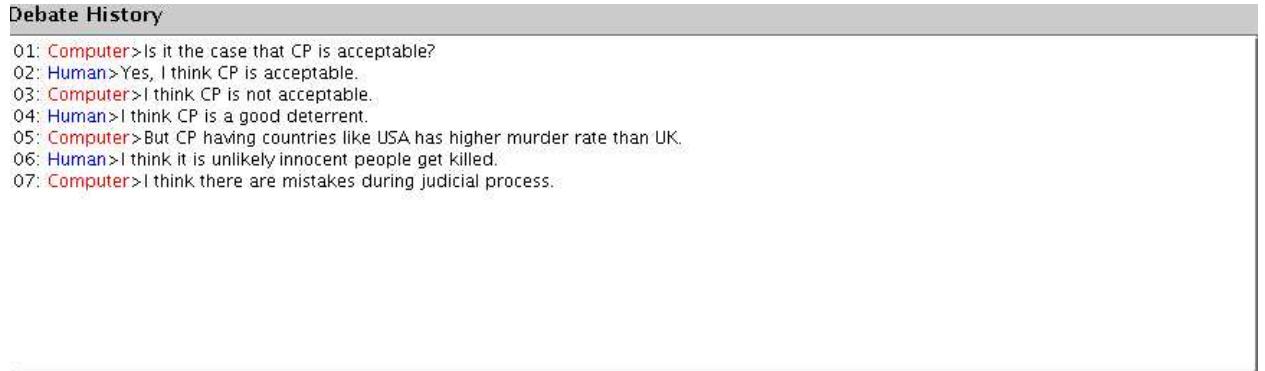


Figure 4.2: The new output facility

4.3 Input facility modifications

The input facilities had to be changed to prevent similar situation like half of the tester experienced. They only used a part of the move content choices because the input facility is made up by two drop down boxes and the move content choices has 37 item to select from. Therefore some other structure of the input facility had to be made. The one that was used is two open text boxes. The first one is the move type choice and are the same as the move type choice drop down in the old user interface. Now, all the options in the move type choice are visible to the user. The second one is the move content choice. This open text box has the same content as the move content choice drop down in the old user interface. Now, 14 items are visible to the user, and the scroll bar is more obvious presented in a box like this rather then the in a small drop down. The two text boxes are JList components that are part of the Java swing library. The new input facility can be seen in figure 4.3

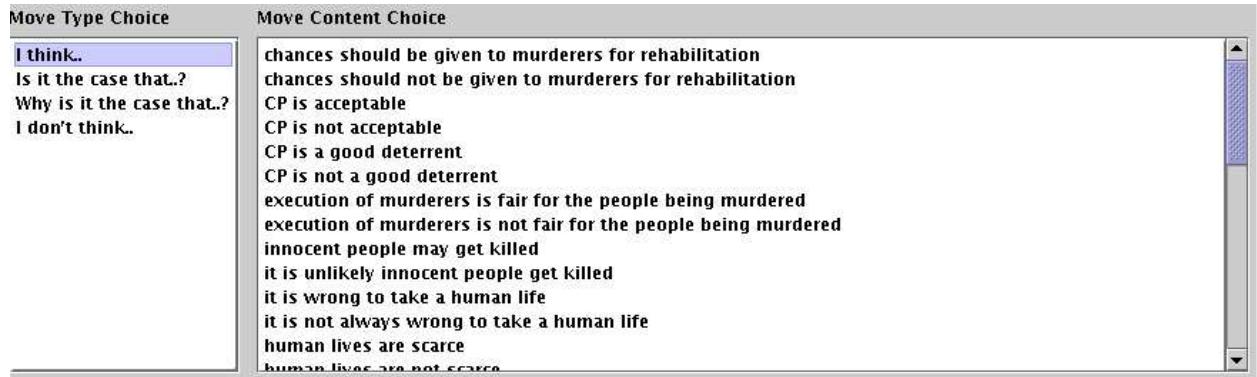


Figure 4.3: The new input facility

As discussed the human and the computer position windows were removed and placed in the input facility. Now when the user select *Why is it the case that..?* to challenge the computer, the possible statements to select are now displayed in the Move content choice box, the statements that were marked by star * icon in the old user interface are not displayed. An example when the user challenge the computer can be seen in figure 4.4

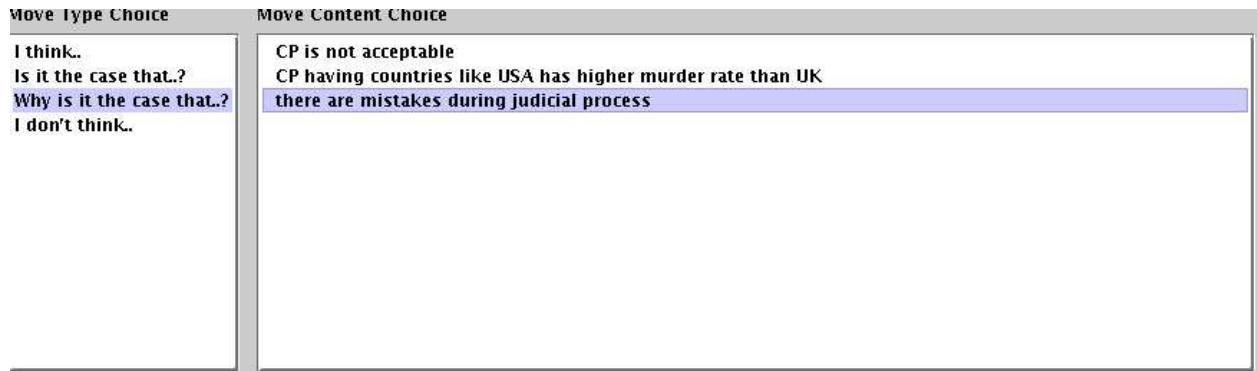


Figure 4.4: Challenge the computer in modified user interface

The same was done with the withdraw from user as challenging the computer. After selecting *I don't think..* from the move type choice the user gets the possible statements to select. And only the users commitment are displayed, no stars, no icons. An example when the user withdraw can be seen in figure 4.5

4.4 Other modifications

The *Go* button was an issue that the tester commented about in the user evaluation. Therefore the button was moved under the move type choice box solving the location problem.

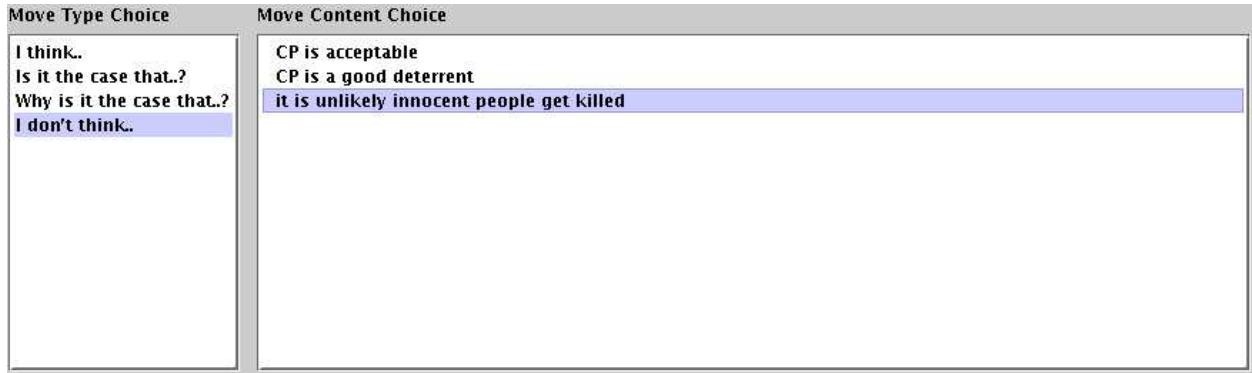


Figure 4.5: Withdraw statement in modified user interface

The name was changed to *Confirm* and the image removed from the button.

The blinking text at the bottom of the screen were very helpful to the testers in the user evaluation. Therefore to place emphasis on the text a border were placed around the text with the header *Help* that explains what the text does. The text changed colours, using three colours, now the text only change colours between black and red. To solve the comment about how rapidly the text blinks, the sleep on the thread that controls the blinking text were increased. The modified blinking text can be seen in figure 4.6

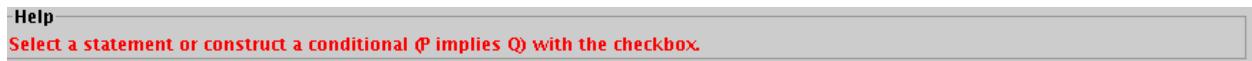


Figure 4.6: Help text in modified user interface

The idea of an indicator to show the status in the debate was solved by a progress bar. The bar is JProgressBar that is a part of the Java swing library. The bar has two colours, blue for human and red for the computer. This are the same colours as used in the debate history window to indicate the owner of a statement. These colours were used for the progress bar to link with the colours in the debating history window, given blue the representation of the human in the system and red as the computer. The progress bar can be seen in figure 4.7. There are now functionality behind the bar, it is placed on the user interface as reminder in further development.



Figure 4.7: Status bar in modified user interface

A short cut keys was set to the menu bar and all the menu items, also on the confirm button. In the menu bar the option to change background colour on the debating history

window was removed. The tester in the user evaluation did not use this option and it does not suit the new debating history window to allow this option.

The frame that contains the user interface was set with fixed size and that is not possible to resize the frame. The code were reformatted so the code layout would fit standard layout of coding. By modifying the system as discussed above the code counted in lines went from 6401 lines in the old version to 5440 lines in the modified version. The line of code had reduced by nearly 1000 lines.

4.5 Summary

The changes on the system were according to the user evaluation. From the user evaluation a need to change the output and the input facility was establish. Now the output facility has one window to display the outputs and the input facility is open showing more contents to select from. The selection on committed statements from the computer or the user are now in the input facility.

The *Go* button issues was solved by moving the button and change the name and the look of the button. The blinking text was changed from testers comments and a status bar added to indicate the status of the debate.

The system keeps the same structure as before, only one class was removed, the DebatingSystemInterface class. That class were responsible of creating the user interface and was the driver of the system. A new class were made called GUI that performs similar tasks as the DebatingSystemInterface class. The new class creates the new GUI and then creates objects of other classes like the DebatingSystemInterface class did. New method was added to the GUI class that the DebatingSystemInterface class did not have. These methods are used in the new user interface solve tasks related to the new output facility such as find index on selection. The source code of the system can be seen in Appendix A, also on the CD included with this document. The author of the code is Tangming Yuan but modified by Kristján Ævarsson. On the CD there are executable version of the human computer debating system. An Internet version of the game was made and can be seen on <http://kristjan.nix.is/>.

The new version of the human computer debating system was made according to testers comments in the user evaluation. The new user interface looks more alive after adding colours and should be more appealing to the user. To confirm the changes a further user evaluation have to be done on the new version. The next chapter discuss how the changes was evaluated.

Chapter 5

Further Evaluation

This chapter introduces information about the second user evaluation on the human computer debating system. Second evaluation is one of the objectives of this project. The aim is to confirm the modifications on the system as discussed in last chapter. The tester were the same group as in the first user evaluation. Information about the purpose, design and the setup are provided in this chapter. Then the results from the user evaluation will be presented.

5.1 User evaluation

From Chapter 4 the changes made were on the input facility and the output facility. The conversation now takes place in only one window as MSN and other communication programs has. The commitments from the computer and the user now is displayed in the input facility. The input facility was changed such that now there are open text box's that the user select from and when challenging or withdrawing the commitment to select from is displayed in one of the text box's in the input facility. These are the main changes made on the system user interface.

5.1.1 User evaluation purpose

The purpose of this evaluation is to get approval. That is to let the same tester as before in the first user evaluation test the new system and answer questions that give information on how efficacious the changes are. Or to confirm the changes made to the system. From those information the next steps in the ongoing development of the human computer debating system can be determine.

5.1.2 User evaluation design

When designing the second user evaluation two key issues had to be addressed. The issues are:

- Time
- Type of information

Questionnaire are good for task analysis and follow-up studies[8]. Since this iteration of the user evaluation is a follow-up on the modifications, questionnaire would be ideal. The fact that time is limitation to the project, then evaluation using questionnaire can be performed fast. Quantitative data were required from the second evaluation, since it would save time and give enough information on the tester opinion on the modifications. Questionnaire is easy to analyse for quantitative data. The question asked in the questionnaire were:

1. How hard was it to start the game ?
2. How useful is the blinking text ?
3. How do you like to have one conversation window ?
4. How do you like the new input facility compared with the old one ?
5. How do you like the new user interface ?
6. Is it easier to play the game now ?
7. Do you enjoy the game more now ?
8. Do the changes made improvement the system ?

With this question a result could be made whether the modifications have been to improve the system.

5.1.3 User evaluation process

The questioner and the system was send via email to the testers. This method was used because it was hard to find the testers and book time to test the system. By sending it via email, the testers could play the game and answer the questioner any time they liked. The testers then send the questioner back also via email to the experimenter.

5.2 User evaluation results

The response from the testers were printed out. Lined up on a table, then one by one, answers for each question was counted. The results from the evaluation can be seen in table 5.1 and table 5.2. Table 5.1 has questions that could be answered by three options. Table 5.2 present the yes and no questions.

Question	Bad	OK	Good
How hard was it to start the game	0%	10%	90%
How useful is the blinking text	0%	40%	60%
How do you like to have one conversation window	0%	10%	90%
How do you like the new input facility compared with the old one	0%	10%	90%
How do you like the new user interface	0%	30%	70%

Table 5.1: Second evaluation result summary I

Question	Yes	No
Is it easier to play the game now	100%	0%
Do you enjoy the game more now	90%	10%
Do the changes made improvement the system	100%	0%

Table 5.2: Second evaluation result summary II

The result from the evaluation does confirm that the modifications made has improved the system. The problem of having three windows have been solved and 90% of the tester confirmed that one window is good. No one thought that one conversation window were bad. The same result were for the input facility, confirming that the changes made are successful. The attitude agains the user interface has changes with the new user interface. Such that none of the testers found the new interface bad. The most importand finding was that 100% of the testers thought that was easier to play the game now with the new user interface. Also 100% of the testers confirmed that the changes made improve the system.

5.3 Summary

This chapter has discussed the second user evaluation. The purpose of confirming the modifications by using questionnaire for the same group as participated in the firs user evaluation. Then displaying and discussing the result from the second user evaluation. The result was very promising, and confirmed that the modifications made has been successful. The result

added more joy to the project, and it was good to have fully functional system that the potential users like.

Chapter 6

Conclusions

This chapter introduces the conclusions of the project. In the beginning of the project the objectives was made of how to carry out the project. This chapter reflects back on this objectives and discuss how and if they have been achieved. Then further work on the human computer debating system will be discussed. At the end the project contribution will be discussed following the last words.

6.1 Objectives reflection

The objectives set out in chapter 1 have been achieved. The system has been evaluated with user evaluation as discussed in chapter 3. The user evaluation examined four parts of the human computer debating system. These parts were: the user interface, system intelligence, user feeling toward the system and system potentials. Participants were students in The University of Akureyri, the interview was laboratory study followed by an interview. An interview transcript was made from the interviews to analyse the user evaluation. The result from the user evaluation confirmed that the tester found the game interesting and enjoyed playing. They saw potential for the system to aid education and argumentation. The user interface prevented the tester to play the game so an intelligent conversation could be made. Although the testers confirmed that the system is intelligent, even so that many of the tester was not sure if they were talking to human or computer. The user interface had many drawbacks, so changes to that part of the system was necessary.

The modification on the system based on the user evaluation result is the next objective that was discussed in chapter 4. The objective was achieved by changing the user interface, solving the problems that the tester had with the system, such as the input and facilities.

To confirm the modifications on the system a further evaluation was set as one objective of the project. This objective was achieved with performing a further evaluation on the human computer debating system with the same participants as in the first user evaluation. The evaluation was performed by sending questionnaire to testers with questions so that it

would be possible to determine if the tester confirmed the changes as discussed in chapter 5.

Although these objectives have been achieved, the human computer debating system is not completed, it is ongoing development and further work are needed.

6.2 Further work

There are many ways that can be carried out on the human computer debating system. The recommendation on further work that would benefit the human computer debating system are:

1. Implement value method that can evaluate how the players are doing in the debate, and connect the method to the status bar in the bottom right corner of the user interface.
2. From the user evaluation it was clear that the enjoyment of playing the game would be much more if the possibility of selecting different domains were possible. Therefore to add debate topics would benefit the system.
3. Implement facility with user interface to able adding in new domain.
4. Develop the input facility such that it can handle natural language as input
5. Asses the educational values of the system.
6. Make the system more aggressive so it would attack the user more.
7. Asses the system strategic

6.3 Project contribution

Much has been done, much left. The project are contributed new version of the human computer debating system to the overall project of the human computer debating system. The new version contribute to the intrinsic human-computer dialog and argumentation. It contribution to the educational application area, with a fully functional new version of the human computer debating system.

6.4 Conclusion

The evaluation on the human computer debating system was in favour of the system, excluding the user interface that has now been modified and confirmed by user evaluation. The result is a fully functional debating system that can aid the users and the contribution made increased the enjoyment of the project. This work have given my experience and knowledge

in many areas that I was not familiar with before, therefore it give me confidence to take on projects in the future that are challenging. With confidence I believe that the human computer debating system can develop into a product that will be widely used, therefore I am proud of my contribution towards the development.

Bibliography

- [1] Bath, E. M & Krabbe, E 1982, *From Axiom to Dialogue: A Philosophical Study of Logics and Argumentation*, Walter de Gruyter, Berlin and New York.
- [2] Dix, A, Finlay, J, Abowd, Gregory & Beale, R ,2004, '*Human-Computer Interaction*', 3rd edn, Person Prentice Hall, England. pp.319, 329, 327-328,
- [3] Hamblin, C. 1971, *Mathematical Models of Dialogue*, Theora, 37: 130-155.
- [4] Hintikka, J. ,1979, '*Quantifiers in Natural Languages: Some Logical Problems*'. In Esa Saarinen, D, et al. (eds.) *Game-Theoretical Semantics*, pp.81-117.
- [5] IBM 1997, Research Deep Blue, viewed 17 April, 2006, <<http://www.research.ibm.com/deepblue/>>.
- [6] Mackenzie, J. 1979, *Begging the Question in Non-Cumulative Systems*, Journal of Philosophical Logic, 8:117-133.
- [7] Moore, D ,1993, '*Dialogue Game Theory for Intelligent Tutoring Systems*', Unpublished Doctoral Dissertation, Leeds Metropolitan University, UK.
- [8] Nielsen, J ,1993, '*Usability Engineering*', Academic Press, San Diego, USA. pp.165, 174, 211
- [9] Nielsen, J & Landauer, T.K ,1993, '*A mathematical model of the finding of usability problems*', Proceedings of ACM INTERCHI'93 Conference, Amsterdam, The Netherlands. pp.206-213
- [10] Walton, D ,1984, '*Logical Dialogue Games and Fallacies*', University Press of America.
- [11] Yuan, T ,2004, '*Human Computer Debate, a Computational Dialectics Approach*', PhD thesis, Leeds Metropolitan University, Leeds, UK. pp.132, 133, 81, 84, 76-77,

Appendix A

System source code

Appendix A includes the whole source code of the prototype. Files are displayed in alphabetical order, and are separated with the header of the next file.

A.1 AboutFrame.java

```
/*
 * Standard "about" window for the program.
 * Displays information about the developers of this program.
 *
 * Created: 23/03/05 by Kristj?n.
 * author Kristj?n ?varsson
 * version 1.0.5, 06/04/2005 - Kristj?n
 */
/*
 * AboutFrame.java
 *
 * Created on April 1, 2006, 6:09 PM
 */

/**
 *
 * author      kristjan
 */
public class AboutFrame extends javax.swing.JFrame {  
    20
    /**
     * Creates new form AboutFrame */
    public AboutFrame() {
        initComponents();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setVisible(true);
        this.show();
    }
    30
    /**
     * This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Generated Code ">//GEN-BEGIN:initComponents

```

```

private void initComponents() {
    buttonPanel = new javax.swing.JPanel();
    headerPanel = new javax.swing.JPanel();
    headerLabel = new javax.swing.JLabel();
    centerPanel = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTextPanel1 = new javax.swing.JTextPane();
    closeButton = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    org.jdesktop.layout.GroupLayout buttonPanelLayout = new org.jdesktop.layout.GroupLayout(buttonPanel);
    buttonPanel.setLayout(buttonPanelLayout);
    buttonPanelLayout.setHorizontalGroup(
        buttonPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(0, 92, Short.MAX_VALUE)
    );
    buttonPanelLayout.setVerticalGroup(
        buttonPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(0, 0, Short.MAX_VALUE)
    );
    headerLabel.setFont(new java.awt.Font("Dialog", 1, 14));
    headerLabel.setText("Human-Computer Debating System");

    centerPanel.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
    jTextPanel1.setFont(new java.awt.Font("Dialog", 1, 12));
    jTextPanel1.setText("This is a prototype based on the PhD work of Tangming Yuan. Modified by Kristjan AEvars");
    jScrollPane1.setViewportView(jTextPanel1);

    org.jdesktop.layout.GroupLayout centerPanelLayout = new org.jdesktop.layout.GroupLayout(centerPanel);
    centerPanel.setLayout(centerPanelLayout);
    centerPanelLayout.setHorizontalGroup(
        centerPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(centerPanelLayout.createSequentialGroup()
            .addContainerGap()
            .add(jScrollPane1, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 346, Short.MAX_VALUE)
            .addContainerGap())
    );
    centerPanelLayout.setVerticalGroup(
        centerPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(centerPanelLayout.createSequentialGroup()
            .addContainerGap()
            .add(jScrollPane1, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 129, Short.MAX_VALUE)
            .addContainerGap())
    );
    closeButton.setMnemonic('c');
    closeButton.setText("Close");
    closeButton.setToolTipText("Close the about window");
    closeButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            closeButtonActionPerformed(evt);
        }
    });
    closeButton.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyPressed(java.awt.event.KeyEvent evt) {
            closeButtonKeyPressed(evt);
        }
    });
}

org.jdesktop.layout.GroupLayout headerPanelLayout = new org.jdesktop.layout.GroupLayout(headerPanel);
headerPanel.setLayout(headerPanelLayout);

```

```

        headerPanelLayout.setHorizontalGroup(
            headerPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                .add(headerPanelLayout.createSequentialGroup()
                    .add(headerPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                        .add(headerPanelLayout.createSequentialGroup()
                            .add(59, 59, 59)
                            .add(headerLabel))
                        .add(org.jdesktop.layout.GroupLayout.TRAILING, headerPanelLayout.createSequentialGroup()
                            .addContainerGap()
                            .add(centerPanel, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE)
                            .add(headerPanelLayout.createSequentialGroup()
                                .add(149, 149, 149)
                                .add(closeButton)))
                    .addContainerGap()))
            );
        headerPanelLayout.setVerticalGroup(
            headerPanelLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                .add(org.jdesktop.layout.GroupLayout.TRAILING, headerPanelLayout.createSequentialGroup()
                    .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .add(headerLabel)
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(centerPanel, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE)
                    .add(17, 17, 17)
                    .add(closeButton))
            );
        org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                .add(layout.createSequentialGroup()
                    .add(layout.createSequentialGroup()
                        .add(137, 137, 137)
                        .add(buttonPanel, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap(171, Short.MAX_VALUE))
                    .add(headerPanel, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE)
                );
            layout.setVerticalGroup(
                layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                    .add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()
                        .add(headerPanel, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE)
                        .add(buttonPanel, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap())
            );
        );
        pack();
    } // </editor-fold> // GEN-END:initComponents

    private void closeButtonKeyPressed(java.awt.event.KeyEvent evt) { // GEN-FIRST:event_closeButtonKeyPressed
        this.dispose();
    } // GEN-LAST:event_closeButtonKeyPressed

    private void closeButtonActionPerformed(java.awt.event.ActionEvent evt) { // GEN-FIRST:event_closeButtonActionPerformed
        this.dispose();
    } // GEN-LAST:event_closeButtonActionPerformed
} // Variables declaration - do not modify // GEN-BEGIN:variables
private javax.swing.JPanel buttonPanel;
private javax.swing.JPanel centerPanel;
private javax.swing.JButton closeButton;
private javax.swing.JLabel headerLabel;
private javax.swing.JPanel headerPanel;

```

```

    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextPane jTextPanel1;
    // End of variables declaration//GEN-END:variables
}

}

```

A.2 AssertionStrategist.java

```

 $\begin{array}{l} \text{/**} \\ \text{* Class: AssertionStrategist} \\ \text{* Author: Tangming Yuan} \\ \text{* Modification: Kristj"sn \r Evarsson} \\ \text{**}/ \\ \text{import java.util.*;} \\ \text{public class AssertionStrategist \{} \\ \text{    \/**} \\ \text{        * Method of plan when facing with an asserted proposition} \\ \text{        */} \\ \text{        * param dilv} \\ \text{        * param selfStore} \\ \text{        * param partnerStore} \\ \text{        * param selfKBS} \\ \text{        * param planner} \\ \text{        * return} \\ \text{    */} \\ \text{    public static Vector planAssertion(DialogueHistory dilv, CommitmentStore selfStore, CommitmentStore partnerStore, Boolean} \\ \text{        \/**} \\ \text{            Vector rmv=new Vector();} \\ \text{            Move previous=(Move)dilv.lastElement();} \\ \text{            Proposition prevProp=previous.getMoveProp();} \\ \text{            RuleProp prevRP=previous.getObject();} \\ \text{        \/**} \\ \text{            1. Start of the game, Computer would automatically adopt the negative view.} \\ \text{                Computer would assert the opposite thesis.} \\ \text{        */} \\ \text{        if(dilv.size()==2) \{} \\ \text{            Move first=(Move)dilv.elementAt(0);} \\ \text{            //user answer yes} \\ \text{            if(previous.getType()=="Concession"&&previous.getContent()==first.getContent()) \{} \\ \text{                Proposition p=new Proposition("CP is acceptable",new Boolean(false));} \\ \text{                rmv.add(new Move('C',"Assertion",p));} \\ \text{                planner.computerThesis=p;}} \\ \text{\}} \end{array}$ 

```

```

    }
    //user answer no
    else {
        Proposition p=new Proposition("CP is acceptable",new Boolean(true));
        rmv.add(new Move('C',"Assertion",p));
        planner.computerThesis=p;
    }
}

}//end of game set up

/*
2. There is a plan under execution, then try to execute the plan
*/
else if (planner.currentPlan!=null) {
    /*
    2.1 Try to execute the plan
    If the plan can be continuously executed, then execute it
    */
    Move m=planner.currentPlan.execute(dilv, planner, partnerStore, selfStore, selfKBS);
    if(m!=null)rmv.add(m);
}

/*
2.2 Else, plan is abandoned, Computer needs to check whether there are level 3 methods available to either build or demolish strategy given the current dialogue state. If there are level 3 available, then retain the current focus
Otherwise, call the focus shift manager
*/
else {
    Move m2=FocusShiftManager.execute(dilv, selfStore, partnerStore, selfKBS, planner);
    if (m2!=null)rmv.add(m2);
}

}//End of there is a plan under execution

/*
2. If there is no plan under execution and previous is a proposition
*/
else if (planner.currentPlan==null&&prevProp!=null) {
    /*
    Check the previous statement supports S's view or against computer's
    Check whether the current focus is retained
    */
    if (prevProp.equals(planner.computerThesis.denial())||selfKBS.support(prevProp, planner.computerThesis.denial()))
        /*
        Addressing previous move
        */
        Move m1=null;
        if(selfStore.onAssertion(prevProp)==false)
            m1=TacticManager.getAction(prevProp, dilv, selfStore, partnerStore, selfKBS, planner);
        if (m1!=null)rmv.add(m1);
}

```

```

    ^M
    ^M
    if(rmv.isEmpty()) {^M
        Move m3=FocusShiftManager.execute(dilv, selfStore, partnerStore, selfKBS, planner);^M 100
        if (m3!=null)rmv.add(m3);^M
        ^M
    } //end of S address previous move^M
    ^M
} //end of opposite statement^M
^M
/*^M
Else If computer has uttered statement against itself^M
^M
*/^M
If the statement supports C's thesis, then start an immediate plan built C's thesis^M
*/^M
else if(selfKBS.support(prevProp, planner.computerThesis)) {^M
    planner.currentPlan=selfKBS.getPlan(prevProp);^M
    planner.currentPlan.add(0, prevProp);^M
    planner.currentPlan.reform();^M
    planner.currentPlan.removeElementAt(0);^M
    Move m=planner.currentPlan.startImmediatePlan();^M
    planner.currentPlan.show();^M
    ^M
    if (m!=null)rmv.add(m);^M
}^M
^M
/*^M
Otherwise, C may assess whether C still stick with the thesis^M
*/^M
else {^M
    rmv.add(new Move('C',"Question",planner.computerThesis.denial()));^M
    ^M
}^M
^M
} //end of no plan under execution^M
^M
/*^M
There is no plan under execution and S surprisingly gives a conditional^M
*/^M
else {^M
    Move m=FocusShiftManager.execute(dilv, selfStore, partnerStore, selfKBS, planner);^M
    if (m!=null)rmv.add(m);^M
}^M
^M
return rmv;^M
} //end of static method^M
} //end class^M
^M

```

A.3 Blink.java

```

/***
 * Class: Blink
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import java.awt.*;
public class Blink implements Runnable {
    Thread activity = new Thread(this);
    GUI gui;
    public Blink(GUI gui_) {
        gui=gui_;
    }
    public void run() {
        while(true) {
            blink();
            try{Thread.sleep(4000);} catch(InterruptedException ie){}
            blink();
        }
    }
    private void blink() {
        for(int i=1;i<=4;i++) {
            try{Thread.sleep(1000);} catch(InterruptedException ie){}
            gui.messageLabel.setForeground(Color.red);
            try{Thread.sleep(1000);} catch(InterruptedException ie){}
            gui.messageLabel.setForeground(Color.black);
        }
    }
}
//end run
}//end of Blink

```

A.4 BooleanRuleBase.java

```
/**  
 * Class: BooleanRuleBase  
 * Author: Tangming Yuan  
 * Modification: Kristj"sn Evarsson  
 */  
  
import java.util.*;  
import java.awt.*;  
import java.awt.event.*;  
  
//class  
//author Tangming Yuan  
  
public class BooleanRuleBase {
```

```

String name;//name of the rule base
Vector propositionList = new Vector();// standalone proposition in the rule base
Vector rulePropList = new Vector();
Vector ruleList = new Vector();// list of all rules
TextArea textArea1=new TextArea();//
public BooleanRuleBase(String name) {
    this.name=name;
}
/**
* Sets the text area for the display of information.
*
* param txtArea the TextArea to be used for the display of information
*/
public void setDisplay(TextArea txtArea) {
    textArea1 = txtArea;
}
/**
* Adds the given text to the trace information.
*
* param text the String that contains the text to be displayed as part of the
* trace information
*/
public void trace(String text) {
    if (textArea1 != null) {
        textArea1.append(text);
    }
}
/**
* Displays all the propositions in the given text area.
*
* param textArea the JTextArea where the propositions are
* displayed
*/
public void displayProposition(TextArea textArea) {
    textArea.append("\n" + name + " Knowledge Base proposition: " + "\n\n");
    Enumeration enum = propositionList.elements();
    
while (enum.hasMoreElements()) {
        Proposition temp = (Proposition) enum.nextElement();
        temp.display(textArea);
    }
}

/**
* Displays all the rules and in text format in the given text area.
*
* param textArea the JTextArea where the rules are displayed
*/
public void displayRules(TextArea textArea) {

```

```

textArea.append("\n" + name + " Rule Base: " + "\n\n");  

Enumeration enum = ruleList.elements();  

while (enum.hasMoreElements()) {  

    Rule temp = (Rule) enum.nextElement();  

    temp.display(textArea);  

}  

}//end method  

/**  

 * Adds a proposition to this rule base proposition list.  

 */  

* param proposition the proposition to be added to the rule base  

*/  

public void addProposition(Proposition prop) {  

    if (propositionList == null) {  

        propositionList = new Vector();  

    }  

    propositionList.addElement(prop);  

}  

/**  

 * Adds a rule to this rule base  

 */  

* param rule the Rule to be added to the rule base  

*/  

public void addRule(Rule r) {  

    if (ruleList == null) {  

        ruleList = new Vector();  

    }  

//add rule  

ruleList.addElement(r);  

//adding propositions to rule proposition list  

rulePropList.add(r.getAntecedent());  

rulePropList.add(r.getConsequent());  

try {  

    rulePropList.add(r.getWarrent());  

} catch(NullPointerException e){}  

}  

/**  

 * Adds a rule to this rule base  

 */  

* param rule the Rule to be added to the rule base  

*/  

public void removeRule(Rule obj) {  

    Rule r=null;  

try{r=matchRule(obj);} catch(NullPointerException e){}  

}

```

```

if(r!=null) {M
    //remove ruleM
    ruleList.removeElement(r);M
    M
    //adding propositions to rule proposition listM
    rulePropList.removeElement(r.getAntecedent());M
    rulePropList.removeElement(r.getConsequent());M
    try {M
        rulePropList.removeElement(r.getWarrent());M
    } catch(NullPointerException e){};M
}M
}//end of remove RuleM
M
/**M
 * Find the match proposition in KBS for a given propositionM
 * Only from the rulePropList, stick with the rulesM
 */M
public Proposition matchProp(Proposition p) {M
    Enumeration enum = rulePropList.elements();M
    Proposition match=null;M
    while (enum.hasMoreElements()) {M
        try {M
            Proposition test = (Proposition) enum.nextElement();M
            if(test.equals(p)){match=test;};M
        } catch(NullPointerException e){};M
    }//end whileM
    return match;M
}//end methodM
M
/*M
match a proposition from the proposition listM
standalone propositionM
*/M
public Proposition matchProposition(Proposition p) {M
    Enumeration enum = propositionList.elements();M
    Proposition match=null;M
    while (enum.hasMoreElements()) {M
        Proposition test = (Proposition) enum.nextElement();M
        if(test.getName()==p.getName()&&test.check().equals(p.check())) {M
            match=test;M
        };M
    }//end whileM
    return match;M
}//end methodM
M
/**M
 * Find the match rule in KBS for a given ruleM
 * Only from the ruleListM
 */M
public Rule matchRule(Rule r) {M
    Enumeration enum = ruleList.elements();M
    Rule match=null;M
    while (enum.hasMoreElements()) {M

```

```

Rule test = (Rule) enum.nextElement();  

if(test.getAntecedent().getContent() == r.getAntecedent().getContent() && test.getConsequent().getContent() == r.getContent())  

    match = test;  

}  

}//end while  

return match;  

}//end method  

/*  

 * A rule or proposition is agreeable  

 */  

public boolean agreeable(RuleProp rp) {  

    boolean agree = false;  

    if (rp.getClass().getName() == "Proposition") {  

        Proposition p = (Proposition) rp;  

        if(matchProposition(p) != null){agree = true;}  

    }  

    else {  

        Rule r = (Rule) rp;  

        if(matchRule(r) != null)agree = true;  

    }  

    return agree;  

}  

/**  

 * Check if a given proposition is in the KBS  

 * Only from the proposition list, If the proposition  

 * is in the KBS, return true, else not exist  

 */
public boolean checkProp(Proposition p) {  

    Enumeration enum = propositionList.elements();  

    boolean agree = false;  

    while (enum.hasMoreElements()) {  

        Proposition test = (Proposition) enum.nextElement();  

        if(test.getName() == p.getName() && test.check().equals(p.check())) {  

            agree = true;  

        }  

    }  

}//end while  

return(agree);  

}//end method  

/*  

 * Find the negation of a given proposition in the KBS  

 * we are going to object about it  

 * return the negative proposition  

 */
public Proposition negationProp(Proposition p) {  

    Enumeration enum = propositionList.elements();  

    Proposition negation = null;  

    while (enum.hasMoreElements()) {  


```

```

Proposition test = (Proposition) enum.nextElement();  

if(test.getName() == p.getName()) {  

    if((test.check().booleanValue() == true&&p.check().booleanValue() == false) || (test.check().booleanValue() == false&&p.check().booleanValue() == true))  

        negation = test;  

    }  

}  

//end while  

return negation;  

//end of method  

/*  

* Find all the ground proposition of a given proposition in the KBS  

*/  

public Vector groundProps(Proposition p) {  

    Proposition p1=matchProp(p);  

    Vector ground =new Vector();  

    Vector ruleVector=new Vector();  

    try{ruleVector=p1.getRule();} catch(NullPointerException e){}  

    if(ruleVector.isEmpty() == false) {  

        Enumeration enum = p1.getRule().elements();  

        while (enum.hasMoreElements()) {  

            Rule test = (Rule) enum.nextElement();  

            if ((test.check().booleanValue() == true&&p1.equals(test.getConsequent())&&checkProp(test.getAntecedent()))  

                ground.add(test.getAntecedent());  

        }  

    }  

}  

return ground;  

//end method  

/*  

* Find all the acceptable ground proposition of a given proposition in the KBS  

*/  

public Vector acceptableGroundProps(Proposition p, CommitmentStore partnerStore) {  

    Vector acceptableGrounds =new Vector();  

    Vector allgrounds=groundProps(p);  

    if(allgrounds.isEmpty() == false) {  

        Enumeration enum = allgrounds.elements();  

        while (enum.hasMoreElements()) {  

            Proposition gp =(Proposition)((Proposition) enum.nextElement()).clone();  

            /*  

            Avoid begging the question. Acceptable would be  

            (i) a new commitment,  

            (ii) commitment,  

            (iii) or de facto commitment  


```

230 240 250 260 270

```

with respect to partner's store
*/
if(partnerStore.onRecord(gp)===false||(partnerStore.onTotal(gp)||partnerStore.deriveFromTotal(gp))) {/*
    acceptableGrounds.add(gp);
}
}//end while
}//end all grounds is not empty
/*
return acceptableGrounds;
}//end method
/*
* Ground, random select one and fire the rule
** Only one item is put to the ground proposition vector and
** returned as the relevant move
*/
public Vector groundProp(Proposition p) {
    Proposition p1=matchProp(p);
    Vector returning=new Vector();
    /*
    Vector ruleVector=new Vector();
    try{ruleVector=p1.getRule();} catch(NullPointerException e){}
    if(ruleVector.isEmpty()===false) {
        Enumeration enum = ruleVector.elements();
        Vector ground =new Vector();
        Vector rule=new Vector();
        while (enum.hasMoreElements()) {
            Rule test = (Rule) enum.nextElement();
            if ((test.fired===false)&&(test.check().booleanValue()==true&&p1.equals(test.getConsequent())&&checkProp
                ground.addElement(test.getAntecedent());
                rule.addElement(test);
        }
    }
    int j=ground.size();
    if(j>=1) {
        int i= (int ) ( Math.random() * j );
        returning.addElement(ground.elementAt(i));
        ((Rule)rule.elementAt(i)).fire();
    }
}
return returning;
}//end of get the ground method
/*
* Check whether a statement is supported by other statement
*/
public boolean issupported(Proposition p) {
    boolean supported=false;
}

```

```

Enumeration enum = ruleList.elements();  

340  

341  

while (enum.hasMoreElements()) {  

    Rule test = (Rule) enum.nextElement();  

    342  

    if (test.getConsequent().equals(p)&&checkProp(test.getAntecedent())) {  

        supported=true;  

    }  

}  

343  

344  

return supported;  

}  

345  

346  

347  

348  

349  

350  

351  

352  

353  

354  

355  

356  

357  

358  

359  

360  

361  

362  

363  

364  

365  

366  

367  

368  

369  

370  

371  

372  

373  

374  

375  

376  

377  

378  

379  

380  

381  

382  

383  

384  

385  

386  

387  

388  

389  

390  

391  

392  

393  

394  

395  

396  

397  

398  

399  

400  

401  

402  

403  

404  

405  

406  

407  

408  

409  

410  

411  

412  

413  

414  

415  

416  

417  

418  

419  

420  

421  

422  

423  

424  

425  

426  

427  

428  

429  

430  

431  

432  

433  

434  

435  

436  

437  

438  

439  

440  

441  

442  

443  

444  

445  

446  

447  

448  

449  

450  

451  

452  

453  

454  

455  

456  

457  

458  

459  

460  

461  

462  

463  

464  

465  

466  

467  

468  

469  

470  

471  

472  

473  

474  

475  

476  

477  

478  

479  

480  

481  

482  

483  

484  

485  

486  

487  

488  

489  

490  

491  

492  

493  

494  

495  

496  

497  

498  

499  

500  

501  

502  

503  

504  

505  

506  

507  

508  

509  

510  

511  

512  

513  

514  

515  

516  

517  

518  

519  

520  

521  

522  

523  

524  

525  

526  

527  

528  

529  

530  

531  

532  

533  

534  

535  

536  

537  

538  

539  

540  

541  

542  

543  

544  

545  

546  

547  

548  

549  

550  

551  

552  

553  

554  

555  

556  

557  

558  

559  

560  

561  

562  

563  

564  

565  

566  

567  

568  

569  

570  

571  

572  

573  

574  

575  

576  

577  

578  

579  

580  

581  

582  

583  

584  

585  

586  

587  

588  

589  

590  

591  

592  

593  

594  

595  

596  

597  

598  

599  

600  

601  

602  

603  

604  

605  

606  

607  

608  

609  

610  

611  

612  

613  

614  

615  

616  

617  

618  

619  

620  

621  

622  

623  

624  

625  

626  

627  

628  

629  

630  

631  

632  

633  

634  

635  

636  

637  

638  

639  

640  

641  

642  

643  

644  

645  

646  

647  

648  

649  

650  

651  

652  

653  

654  

655  

656  

657  

658  

659  

660  

661  

662  

663  

664  

665  

666  

667  

668  

669  

670  

671  

672  

673  

674  

675  

676  

677  

678  

679  

680  

681  

682  

683  

684  

685  

686  

687  

688  

689  

690  

691  

692  

693  

694  

695  

696  

697  

698  

699  

700  

701  

702  

703  

704  

705  

706  

707  

708  

709  

710  

711  

712  

713  

714  

715  

716  

717  

718  

719  

720  

721  

722  

723  

724  

725  

726  

727  

728  

729  

730  

731  

732  

733  

734  

735  

736  

737  

738  

739  

740  

741  

742  

743  

744  

745  

746  

747  

748  

749  

750  

751  

752  

753  

754  

755  

756  

757  

758  

759  

760  

761  

762  

763  

764  

765  

766  

767  

768  

769  

770  

771  

772  

773  

774  

775  

776  

777  

778  

779  

780  

781  

782  

783  

784  

785  

786  

787  

788  

789  

790  

791  

792  

793  

794  

795  

796  

797  

798  

799  

800  

801  

802  

803  

804  

805  

806  

807  

808  

809  

810  

811  

812  

813  

814  

815  

816  

817  

818  

819  

820  

821  

822  

823  

824  

825  

826  

827  

828  

829  

830  

831  

832  

833  

834  

835  

836  

837  

838  

839  

840  

841  

842  

843  

844  

845  

846  

847  

848  

849  

850  

851  

852  

853  

854  

855  

856  

857  

858  

859  

860  

861  

862  

863  

864  

865  

866  

867  

868  

869  

870  

871  

872  

873  

874  

875  

876  

877  

878  

879  

880  

881  

882  

883  

884  

885  

886  

887  

888  

889  

890  

891  

892  

893  

894  

895  

896  

897  

898  

899  

900  

901  

902  

903  

904  

905  

906  

907  

908  

909  

910  

911  

912  

913  

914  

915  

916  

917  

918  

919  

920  

921  

922  

923  

924  

925  

926  

927  

928  

929  

930  

931  

932  

933  

934  

935  

936  

937  

938  

939  

940  

941  

942  

943  

944  

945  

946  

947  

948  

949  

950  

951  

952  

953  

954  

955  

956  

957  

958  

959  

960  

961  

962  

963  

964  

965  

966  

967  

968  

969  

970  

971  

972  

973  

974  

975  

976  

977  

978  

979  

980  

981  

982  

983  

984  

985  

986  

987  

988  

989  

990  

991  

992  

993  

994  

995  

996  

997  

998  

999  

9999
```

```

} // end of method supported
/*
 * Given a proposition, if it support others, return all the statements
 * it support from lower hierachy to hiher hierachy
 */
public Plan getPlan(Proposition p) {
    Plan supported=new Plan();
    while (supportOthers(p)) {
        supported.add(getDirectSupported(p));
        p=getDirectSupported(p);
    }
    return supported;
} // end of method supported
/*
 * Check whether one proposition rp supports another
 * The first parameter supports the second parameter r
 */
public boolean support(RuleProp rp, Proposition r) {
    Proposition p;
    if (rp.getClass().getName() == "Rule") p=((Rule)rp).getConsequent();
    else p=(Proposition)rp;
    boolean support=false;
    PlanSet planSet=getPlanSet(r);
    Enumeration enum = planSet.elements();
    while (enum.hasMoreElements()) {
        Plan plan=(Plan) enum.nextElement();
        Enumeration enum1 = plan.elements();
        while (enum1.hasMoreElements()) {
            Proposition test = (Proposition) enum1.nextElement();
            if(test.equals(p)) {
                support=true;
                break;
            }
        }
    }
    return support;
} // end of method support
/*
 */

```

390

400

410

420

430

```

        * If there is substantive objections,*
        * Build the negation of that view*
        */
public PlanSet getPlanSet(Proposition p) {/*
    PlanSet planSet=new PlanSet();/*
    /*
    Enumeration enum = propositionList.elements();/*
    /*
    while (enum.hasMoreElements()) {/*
        /*
        Proposition test = (Proposition) enum.nextElement();/*
        /*
        //Make the statement is at the bottom of the hierarchy/*
        if(issupported(test)==false) {/*
            /*
            Plan temp=getPlan(test);/*
            /*
            if (temp.isEmpty()==false&&temp.contains(p)) {/*
                /*
                Plan plan=new Plan();/*
                int i=temp.indexOf(p);/*
                /*
                //The bottom one/*
                plan.add(test);/*
                /*
                for(int j=0;j<=i;j++) {plan.add(temp.elementAt(j));}/*
                planSet.add(plan);/*
            }*/
        }*/
    //}
    return planSet;/*
}
/*
/*Check whether a proposition against proponent's support*/
/*
/*
/*
public boolean againstSupport(RuleProp rp, Proposition r) {/*
    /*
    Proposition p;/*
    /*
    if (rp.getClass().getName() == "Rule") p=((Rule)rp).getConsequent();/*
    else p=(Proposition)rp;/*
    /*
    boolean against=false;/*
    /*
    PlanSet planSet=getPlanSet(r);/*
    Enumeration enum = planSet.elements();/*
    /*
    while (enum.hasMoreElements()) {/*

```

440 450 460 470 480 490

```

Plan plan=(Plan) enum.nextElement();  

Enumeration enum1 = plan.elements();  

while (enum1.hasMoreElements()) {  

    Proposition test = (Proposition) enum1.nextElement();  

    if(p.equals(test.denial())||support(p, test.denial())) {  

        against=true;  

        break;  

    }  

}//end of while  

}//end of while  

return against;  

}//end of method  

/*  

 * Check whether a proposition is challengable or not  

 * This will depends on whether it is a fact,  

 * or whether there is the possibility to rebut it  

 */  

public boolean challengable(RuleProp rp, CommitmentStore partnerStore, CommitmentStore selfStore) {  

boolean challengable=true;  

String name=rp.getClass().getName();  

if(name=="Proposition"&&partnerStore.onAssertion(rp)) {  

    Vector v=groundProps((Proposition)rp);  

    Proposition p=(Proposition)rp;  

    /*  

     * It cannot be further supported  

     */  

    if (v.isEmpty())challengable=false;  

    else if(selfStore.isConsequence(p))challengable=false;  

}  

else challengable=false;  

return challengable;  

}//end of method challengable  

/*  

 *      Get the soul support of his thesis,  

 *      try to make things controversial  

 */  

public Vector getSoulSupport(Proposition p) {  

    Vector v=new Vector();  

}

```

```

if(p.equals(new Proposition("CP is acceptable",new Boolean(true)))) {M
    v.add(new Proposition("CP is a good deterrent",new Boolean(true)));M
    v.add(new Proposition("most people want CP back",new Boolean(true)));M
} else {M
    v.add(new Proposition("innocent people may get killed",new Boolean(true)));M
    v.add(new Proposition("it is wrong to take a human life",new Boolean(true)));M
}M
M
return v;M
}M
M
//display kbsM
public void show() {M
    final Frame kbsFrame=new Frame();M
    //Window closedM
    kbsFrame.addWindowListener( new WindowAdapter() {M
        public void windowClosing( WindowEvent e ) {M
            kbsFrame.dispose();M
        }M
    });M
    Text Area ruleTextArea = new TextArea();M
    Text Area propTextArea = new TextArea();M
    kbsFrame.setSize(new Dimension(300, 400));M
    kbsFrame.setTitle("Computer Knowledge Base");M
    kbsFrame.setLayout(new BorderLayout());M
    kbsFrame.add("North",ruleTextArea);M
    kbsFrame.add("South",propTextArea);M
    displayRules(ruleTextArea);M
    displayProposition(propTextArea);M
    kbsFrame.setVisible(true);M
}M
M
public void showRuleProp() {M
    Enumeration enum = rulePropList.elements();M
    while (enum.hasMoreElements()) {M
        Proposition temp = (Proposition) enum.nextElement();M
        {M
            try{} catch(NullPointerException e){}M
        }M
    }M
}M
}//endM
}//end classM

```

550 560 570 580

A.5 ChallengeStrategist.java

```

/**M
 * Class: ChallengeStrategistM
 * Author: Tangming YuanM
 * Modification: Kristj"sn EvarssonM
**/M

```

```

import java.util.*;M
public class ChallengeStrategist {M
    M
    //Method of plan when facing with an asserted propositionM
    public static Vector planChallProp(Proposition prevProp,char currentTurn,CommitmentStore selfStore,CommitmentStore
    M
    Vector rmv=new Vector();          M
    /*M
    1      If the grounds are already in the opponent's store,M
    M
    */
    ConflictSet partnerConflictSet=partnerStore.getRealPremises(prevProp);M
    Vector acceptableGrounds=selfKBS.acceptableGroundProps(prevProp, partnerStore);M
    M
    if (partnerConflictSet.isEmpty()==false) {M
        Proposition p=partnerConflictSet.mergeToProp();M
        M
        rmv.add(new Move(currentTurn,"Resolve",p,partnerConflictSet));M
    }M
    M
    M
    /*M
    2      Give a groundM
    */
    else if(acceptableGrounds.isEmpty()==false) {M
        int j=acceptableGrounds.size();M
        int i= ( int ) ( Math.random() * j );M
        M
        rmv.add(new Move('C',"Ground", (Proposition)acceptableGrounds.elementAt(i)));M
    }M
    M
    M
    /*M
    * 3      Withdraw if no reason can be providedM
    */
    M
    if(rmv.isEmpty())      rmv.add(new Move('C',"Withdraw", prevProp));M
    M
    return rmv;M
}//end of challenge methodM
}//end of challenge strategist classM
M

```

20

30

40

A.6 CommitmentManager.java

```

/**M
 * Class: CommitmentManagerM
 * Author: Tangming YuanM
 * Modification: Kristj"sn EvarssonM
**/M

```

```

import java.util.*;M
M
class CommitmentManager {M
    // Method commitmentM
    public static void commit(Move currentMove, Move previousMove, CommitmentStore currentStore, CommitmentStore opponentStore) {
        // Assertions onlyM
        if (currentMove.getType() == "Assertion" || currentMove.getType() == "Ground") {
            // Current move proposition or ruleM
            RuleProp crp = ((Move) currentMove.clone()).getObject();M
            M
            // ground givenM
            if (currentMove.getType() == "Ground") {
                // previous move proposition or ruleM
                RuleProp prp = ((Move) previousMove.clone()).getObject();M
                M
                // if it is a rule, only the warrant is addedM
                if (prp.getClass().getName() == "Rule") {
                    Rule match = currentStore.matchRule((Rule) prp);M
                    match.setWarrent((Proposition) crp);M
                }
                M
                else // it is a propositionM
                {
                    // previous propositionM
                    Proposition p = (Proposition)((Proposition) prp).clone();M
                    currentStore.addAssertion(new Rule((Proposition) crp, p));M
                    opponentStore.addConcession(new Rule((Proposition) crp, p));M
                }
            }
            M
            // end of Challenge and standaloneM
            M
            // adding to the current storeM
            currentStore.addAssertion(crp);M
            opponentStore.addConcession(crp);M
        }
        M
        // Current move type is withdrawM
        else if (currentMove.getType() == "Withdraw") {
            RuleProp crp = currentMove.getObject();M
            currentStore.withdraw(crp);M
        }
        M
        // Current move type is challengeM
        else if (currentMove.getType() == "Challenge") {
            RuleProp crp = currentMove.getObject();M
            currentStore.withdraw(crp);M
        }
        M
        else if (currentMove.getType() == "Concession") {
            RuleProp crp = currentMove.getObject();M
            currentStore.addAssertion(crp);M
            opponentStore.addConcession(crp);M
        }
        M
    }
}

```

```

        currentStore.repaint(c);  

        opponentStore.repaint(o);  

    }  

}

```

60

A.7 CommitmentStore.java

```

/**  

 *  Class: CommitmentStore  

 *  Author: Tangming Yuan  

 *  Modification: Kristj"sn Avarsson  

**/  

import java.util.*;  

import java.io.*;  

import java.awt.*;  

import javax.swing.*;  

import java.awt.event.*;  

//class  

//author Tangming Yuan  

public class CommitmentStore extends BooleanRuleBase {  

    //The non-cumulative stack  

    Vector claimStack=new Vector();  

    //What they said  

    Vector assertionList = new Vector();  

    //Silent concession  

    Vector concessionList = new Vector();  

    //total commitment list  

    Vector totalList = new Vector();  

    /*  

     *In order to ban the line of question-begging,  

     *we put up an altogether list of commitments occurs  

    */  

    Vector record=new Vector();  

    //Constructor  

    public CommitmentStore(String name) {  

        super(name); //this.name=name;  

    }  

    public Vector getAssertionList() {  

        return assertionList;  

    }
}

```

10

20

30

40

```

M
/** M
 * Displays all the rules and propositions in text format in the given dialogue choice.M
*M
 * param the choice where the rules and propositions are displayedM
 * In the interface, if it is assertion, S BEFORE, IF IT IS A CONCESSIONM
 * C BEFOREM
*/M
public void repaint(java.awt.List c) {M
    //Empty it firstM
    c.clear();M
M
    Enumeration enum = totalList.elements();//totalList.elements();M
    while (enum.hasMoreElements()) {M
        RuleProp temp = (RuleProp) enum.nextElement();M
        {M
            M
            if(onAssertion(temp)) {M
                c.add("      "+temp.getContent());M
            } else c.add("* "+temp.getContent());M
            }M
        }M
    }//end methodM
M
/** M
 * Displays all the rules and propositions in text format in the given dialogue choice.M
*M
 * param the choice where the rules and propositions are displayedM
*/M
public void showAssertion() {M
    //Empty it firstM
    //c.removeAll();M
    //System.out.println("Assertions:\n");M
    Enumeration enum = assertionList.elements();M
    while (enum.hasMoreElements()) {M
        RuleProp temp = (RuleProp) enum.nextElement();M
        {M
            System.out.println(temp.getContent());M
        }M
    }M
}i//end methodM
M
//M
public void showConcession() {M
M
    System.out.println("Concessions:\n");M
    Enumeration enum = concessionList.elements();M
    while (enum.hasMoreElements()) {M
        RuleProp temp = (RuleProp) enum.nextElement();M
        {M
            System.out.println(temp.getContent());M
        }M
    }M
}M

```

```

} //end M
/** M
 * Adds a assertion proposition or rule to a commitment store. M
 * M
 * the assertion will go to the concession set and the claim stack as well M
 */ M
public void addAssertion(RuleProp obj) { M
    //The record will be added M
    record.add(obj); M
    M
    if(onClaimStack(obj)==false) { M
        claimStack.addElement(obj); M
    } M
    M
    //adding the object to the assertionList and totallist if it is not there M
    if(onAssertion(obj)==false) { M
        assertionList.addElement(obj); M
    } M
    M
    //For an assertion, the negation should be removed from the concession set M
    if(obj.getClass().getName()=="Proposition") { M
        M
        Proposition p=((Proposition)obj).denial(); M
        if(onConcession(p)==true) { M
            //System.out.println("Good dog//"); M
            withdrawConcession(p); M
            //System.out.println("Good dog again//"); M
        } M
        M
        if(onTotal(obj)==false) { M
            totalList.addElement(obj); M
        } M
        M
        //adding to the reasoning part M
        if (obj.getClass().getName()=="Proposition") { M
            super.addProposition((Proposition)obj); M
        } else { M
            super.addRule((Rule)obj); M
        } M
    } M
} //end of adding assertion M
M
M
/** M
 * Adds a concession proposition or rule to a commitment store. M
 * M
 * the assertion will go to the concession set only, M
 * need to further check and delete the negation M
 */ M
public void addConcession(RuleProp obj) { M
    M
    record.add(obj); M
    M

```

100
110
120
130
140
150

```

//adding the object to the concession list and the claim stack
if(onConcession(obj)==false) {  

    concessionList.addElement(obj);  

}  

if(onTotal(obj)==false) {  

    totalList.addElement(obj);  

}  

//adding to the reasoning part
if (obj.getClass().getName() == "Proposition") {  

    super.addProposition((Proposition)obj);  

} else {  

    super.addRule((Rule)obj);  

}  

}  

}  

}//end of adding concession  

/*  

Withdraw from concession rather than concession
public void withdrawConcession(RuleProp obj) {  

    //remove the object to the concession list and the claim stack
    try{concessionList.removeElement(obj);} catch (ArrayIndexOutOfBoundsException e){};  

    if (onAssertion(obj)==false) {  

        try{totalList.removeElement(obj);} catch (ArrayIndexOutOfBoundsException e){};  

    }  

    //Remove a proposition
    if (obj.getClass().getName() == "Proposition") {  

        Proposition p=(Proposition)obj;  

        if(super.checkProp(p)) {  

            Proposition match=super.matchProposition(p);  

            //remove from proposition list
            super.propositionList.removeElement(match);  

        }  

        try{concessionList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

        if (onAssertion(match)==false) {  

            try{totalList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

        }  

    }  

}  

}//end proposition part
else //rules
{  

    Rule r=(Rule)obj;  

    Rule match=null;  

    try{match=super.matchRule(r);} catch(NullPointerException e){}  


```

```

        if(match!=null) {  

            super.removeRule(match);  

            try{concessionList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

            if (onAssertion(match)==false) {  

                try{totalList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){}; 210  

            }  

        }  

    } //end of rule part  

}  

}//end of withdraw concession  

/*  

 * Withdraw a commitment proposition or rule from a commitment store.  

 */  

/*  

 * the object will removed from the concession set and the claim stack as well  

 */  

public void withdraw(RuleProp obj) {  

    //remove the object to the concession list and the claim stack  

    try {assertionList.removeElement(obj);} catch (ArrayIndexOutOfBoundsException e){};  

    try{concessionList.removeElement(obj);} catch (ArrayIndexOutOfBoundsException e){};  

    try{totalList.removeElement(obj);} catch (ArrayIndexOutOfBoundsException e){}; 230  

    //Remove a proposition  

    if (obj.getClass().getName()=="Proposition") {  

        Proposition p=(Proposition)obj;  

        if(super.checkProp(p)) {  

            Proposition match=super.matchProposition(p);  

            //remove from proposition list  

            super.propositionList.removeElement(match);  

            try {assertionList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){}; 240  

            try{concessionList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

            try{totalList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

        }  

    } //end proposition part  

}  

else //rules  

{  

    Rule r=(Rule)obj;  

    Rule match=null;  

    try{match=super.matchRule(r);} catch(NullPointerException e){};  

    if(match!=null) {  

        super.removeRule(match);  

        try {assertionList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

        try{concessionList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){};  

        try{totalList.removeElement(match);} catch (ArrayIndexOutOfBoundsException e){}; 250  

    }  

}

```

```

    }M
}//end of rule partM
M
}//end of withdrawM
M
M
/**M
 * Check if a given proposition or rule is on the assertion listM
 */M
public boolean onAssertion(RuleProp rp) {M
    boolean on=false;M
M
    if(rp.getClass().getName() == "Proposition") {M
        Proposition p=(Proposition)rp;M
M
        Enumeration enum = assertionList.elements();M
M
        while (enum.hasMoreElements()) {M
            RuleProp test = (RuleProp) enum.nextElement();M
M
            if(test.getName() == "Proposition") {M
                Proposition ptest=(Proposition)test;M
                if(p.equals(ptest)) {M
                    on=true;M
                }M
            }M
        }M
    }M
} //end propositionM
M
//RulesM
else {M
    Rule r=(Rule)rp;M
M
    Enumeration enum = assertionList.elements();M
M
    while (enum.hasMoreElements()) {M
        RuleProp test = (RuleProp) enum.nextElement();M
M
        if(test.getName() == "Rule") {M
            Rule ptest=(Rule)test;M
            if(r.equals(ptest)) {M
                on=true;M
            }M
        }M
    }M
} //end rule compareM
M
return on;M
} //end of on assertionM
M
/**M
 * Check if a given proposition or rule is on the CLAIM STACKM
 */M
public boolean onClaimStack(RuleProp rp) {M

```

```

boolean on=false;†
†
if(rp.getClass().getName() == "Proposition") {†
    Proposition p=(Proposition)rp;†
    †
    Enumeration enum = claimStack.elements();†
    †
    while (enum.hasMoreElements()) {†
        RuleProp test = (RuleProp) enum.nextElement();†
        †
        if(test.getClass().getName() == "Proposition") {†
            Proposition ptest=(Proposition)test;†
            if(p.equals(ptest)) {†
                on=true;†
            }†
        }†
    }//end while†
}//end proposition†
†
//Rules†
else {†
    Rule r=(Rule)rp;†
    †
    Enumeration enum = claimStack.elements();†
    †
    while (enum.hasMoreElements()) {†
        RuleProp test = (RuleProp) enum.nextElement();†
        †
        if(test.getClass().getName() == "Rule") {†
            Rule ptest=(Rule)test;†
            if(r.equals(ptest)) {†
                on=true;†
            }†
        }†
    }//end while†
}//end of rule compare†
return on;†
}//end of on assertion†
†
/**†
 * Check if a given proposition or rule is on the concession list†
 */
public boolean onConcession(RuleProp rp) {†
    boolean on=false;†
    †
    if(rp.getClass().getName() == "Proposition") {†
        Proposition p=(Proposition)rp;†
        †
        Enumeration enum = concessionList.elements();†
        †
        while (enum.hasMoreElements()) {†
            RuleProp test = (RuleProp) enum.nextElement();†
            †

```

310

320

330

340

350

360

```

if(test.getClass().getName() == "Proposition") {  

    Proposition ptest = (Proposition)test;  

    if(p.equals(ptest)) {  

        on=true;  

    }  

}  

}//end while  

}//end proposition  

//Rules  

else {  

    Rule r = (Rule)rp;  

    Enumeration enum = concessionList.elements();  

    while (enum.hasMoreElements()) {  

        RuleProp test = (RuleProp) enum.nextElement();  

        if(test.getClass().getName() == "Rule") {  

            Rule ptest = (Rule)test;  

            if(r.equals(ptest)) {  

                on=true;  

            }  

        }  

    }  

}//end while  

}//end of rule compare  

return on;  

}//end on concession  

/** * Check if a given proposition or rule is on the total list  

 */
public boolean onTotal(RuleProp rp) {  

    boolean on=false;  

    if(rp.getClass().getName() == "Proposition") {  

        Proposition p = (Proposition)rp;  

        Enumeration enum = totalList.elements();  

        while (enum.hasMoreElements()) {  

            RuleProp test = (RuleProp) enum.nextElement();  

            if(test.getClass().getName() == "Proposition") {  

                Proposition ptest = (Proposition)test;  

                if(ptest.equals(p)) {  

                    on=true;  

                }  

            }  

        }  

    }  

}//end while  

}//end proposition  

//Rules

```

370

380

390

400

410

```

else {  

    Rule r=(Rule)rp;  

    Enumeration enum = totalList.elements();  

    while (enum.hasMoreElements()) {  

        RuleProp test = (RuleProp) enum.nextElement();  

        if(test.getClass().getName() == "Rule") {  

            Rule ptest=(Rule)test;  

            if(ptest.equals(r)) {  

                on=true;  

            }  

        }  

    } //end while  

} //end of rule compare  

return on;  

} //end of on total  

/** * Check if a given proposition or rule is on the total list  

 */
public boolean onRecord(RuleProp rp) {  

    boolean on=false;  

    if(rp.getClass().getName() == "Proposition") {  

        Proposition p=(Proposition)rp;  

        Enumeration enum = record.elements();  

        while (enum.hasMoreElements()) {  

            RuleProp test = (RuleProp) enum.nextElement();  

            if(test.getClass().getName() == "Proposition") {  

                Proposition ptest=(Proposition)test;  

                if(ptest.equals(p)) {  

                    on=true;  

                }  

            }  

        } //end while  

    } //end proposition  

//Rules  

else {  

    Rule r=(Rule)rp;  

    Enumeration enum = record.elements();  

    while (enum.hasMoreElements()) {  

        RuleProp test = (RuleProp) enum.nextElement();  

        if(test.getClass().getName() == "Rule") {  


```

420

430

440

450

460

```

        Rule ptest=(Rule)test;/*
        if(ptest.equals(r)) {/*
            on=true;/*
        }*/
        }/*
    }//end while
}//end of rule compare
return on;
}//end of on total
/*
/*To block the circular argument,*/
 * all the claims on the speaker's assertion, but not on
 * safe the partner's total
*/
public boolean deriveFromTotal(RuleProp rp) {
    boolean derived=false;
    if(onTotal(rp)) {
        derived=true;
    }
    /*
    //If it is a proposition
    else if(rp.getClass().getName() == "Proposition") {
        Proposition p=(Proposition)rp;
        Vector ground=groundProps(p);
        /*
        //First level ground is found
        if (ground.isEmpty() == false){derived=true;} else{}
        //Check if there is match rule
    }
    return derived;
}
}//end of derive from total
/*
Method for checking general conflict for P,?P and P, R>?P
*/
public ConflictSet getConflictSet() {
    ConflictSet conflict=new ConflictSet();
    int n=totalList.size();
    /*
    //check P AND ?P first
    for(int i=0; i<=n-1; i++) {
        RuleProp rp=(RuleProp)totalList.elementAt(i);
        if(rp.getClass().getName() == "Proposition") {
            Proposition denial=((Proposition)rp).denial();
            if(onTotal(denial)) {
                conflict.addElement(rp);
                conflict.addElement(denial);
                break;
            }
        }
    }
}
}//end for loop

```

```

 $\hat{M}$ 
//If there is no P AND ?P conflict, check P and R->?P           conflict $\hat{M}$ 
if (conflict.isEmpty() == true) { $\hat{M}$ 
    for(int i=0; i<=n-1; i++) { $\hat{M}$ 
        RuleProp rp=(RuleProp)totalList.elementAt(i); $\hat{M}$ 
        if(rp.getClassName().getName() == "Rule") { $\hat{M}$ 
            Rule r=(Rule)rp; $\hat{M}$ 
            Proposition ante=r.getAntecedent(); $\hat{M}$ 
            Proposition cons=r.getConsequent(); $\hat{M}$ 
 $\hat{M}$ 
            Proposition denial=((Proposition)cons).denial(); $\hat{M}$ 
            if(onTotal(denial)&&onTotal(ante)&&r.check().booleanValue()) { $\hat{M}$ 
                conflict.addElement(ante); $\hat{M}$ 
                conflict.addElement(r); $\hat{M}$ 
                conflict.addElement(denial); $\hat{M}$ 
                break; $\hat{M}$ 
            } $\hat{M}$ 
        } $\hat{M}$ 
    } //end for loop $\hat{M}$ 
} //end of checking for P, R, R->?P $\hat{M}$ 
 $\hat{M}$ 
return conflict; $\hat{M}$ 
}//end method conflict set $\hat{M}$ 
 $\hat{M}$ 
 $\hat{M}$ 
/* $\hat{M}$ 
 *This method is specific for computer use to avoid unnecessary conflict $\hat{M}$ 
 *It is worth noting that P,?P is considered to be only one has said, $\hat{M}$ 
 *because the underlying logic is defeasible, we take only one explicitly $\hat{M}$ 
 *said as conflict $\hat{M}$ 
 */ $\hat{M}$ 
public ConflictSet getRealConflictSet() { $\hat{M}$ 
    ConflictSet conflict=new ConflictSet(); $\hat{M}$ 
    int n=assertionList.size(); $\hat{M}$ 
 $\hat{M}$ 
//check P AND ?P first $\hat{M}$ 
for(int i=0; i<=n-1; i++) { $\hat{M}$ 
    RuleProp rp=(RuleProp)assertionList.elementAt(i); $\hat{M}$ 
    if(rp.getClassName().getName() == "Proposition" && onAssertion(rp)) { $\hat{M}$ 
        Proposition denial=((Proposition)rp).denial(); $\hat{M}$ 
        if(onAssertion(denial)) { $\hat{M}$ 
            conflict.addElement(rp); $\hat{M}$ 
            conflict.addElement(denial); $\hat{M}$ 
            break; $\hat{M}$ 
        } $\hat{M}$ 
    } $\hat{M}$ 
} //end for loop $\hat{M}$ 
 $\hat{M}$ 
 $\hat{M}$ 
// WE TEMPARARILY NOT CONSIDER THIS KIND OF CONFLICT $\hat{M}$ 
//If there is no P AND ?P conflict, check P and R->?P           conflict $\hat{M}$ 
/*if (conflict.isEmpty() == true) $\hat{M}
{ $\hat{M}$$ 
```

```

for(int i=0; i<=n-1; i++) {
    RuleProp rp=(RuleProp)assertionList.elementAt(i); // (RuleProp)totalList.elementAt(i);
    if(rp.getClass().getName() == "Rule") {
        Rule r=(Rule)rp;
        Proposition ante=r.getAntecedent();
        Proposition cons=r.getConsequent();
        Proposition denial=((Proposition)cons).denial();
        if(onTotal(denial)&&onTotal(ante)&&r.check().booleanValue()) {
            conflict.addElement(ante);
            conflict.addElement(r);
            conflict.addElement(denial);
            break;
        }
    }
}
} // end of checking for P, R, R->?P
}

public ConflictSet getPremises(Proposition p) {
    ConflictSet premises=new ConflictSet();
    int n=totalList.size();
    for(int i=0; i<=n-1; i++) {
        RuleProp rp=(RuleProp)totalList.elementAt(i);
        if(rp.getClass().getName() == "Rule") {
            Rule r=(Rule)rp;
            Proposition ante=r.getAntecedent();
            Proposition cons=r.getConsequent();
            // The antecedent and the rule's consequence is the given proposition
            if(onTotal(ante)&&cons.equals(p)&&r.check().booleanValue()) {
                premises.addElement(ante);
                premises.addElement(r);
                break;
            }
        }
    }
    return premises;
}
} // end method getPremises
}

// Method for getting the premises for a given proposition

```

```

//Such as given P, see if R and R->P is in the store
//The purpose is to fulfil the challenge and withdraw a consequence
//Can be followed by either withdraw or affirmation,
//This one concerns assertion set only
public ConflictSet getRealPremises(Proposition p) {
    ConflictSet premises=new ConflictSet();
    int n=assertionList.size();
    //Search the total list
    for(int i=0; i<=n-1; i++) {
        RuleProp rp=(RuleProp)assertionList.elementAt(i);
        if(rp.getClass().getName() == "Rule") {
            Rule r=(Rule)rp;
            Proposition ante=r.getAntecedent();
            Proposition cons=r.getConsequent();
            //The antecedent and the rule's consequence is the given proposition
            if(onTotal(ante)&&cons.equals(p)&&r.check().booleanValue()) {
                premises.addElement(ante);
                premises.addElement(r);
                break;
            }
        }
    }
    return premises;
}

//Method of get student input for resolution demand
//from the assertion list
public ConflictSet getInput(int [] a) {
    ConflictSet input=new ConflictSet();
    for(int i=0;i<=a.length-1;i++) {
        input.add(assertionList.elementAt(a[i]));
    }
    return input;
}

/*Method of get supported statement for a further challenge
*/
public Proposition getSupportedProp(RuleProp p) {
    Proposition support=null;
    //A rule currently is not concerned
    if(p.getClass().getName() == "Rule"){support=((Rule)p).getConsequent();}
    else {
        Proposition pp=(Proposition)p;
        int n=assertionList.size();
    }
}

```

```

    }
    //Search the total list
    for(int i=0; i<=n-1; i++) {
        RuleProp rp=(RuleProp)assertionList.elementAt(i);
        if(rp.getClass().getName() == "Rule") {
            Rule r=(Rule)rp;
            Proposition ante=r.getAntecedent();
            Proposition cons=r.getConsequent();
            ...
            //The antecedent and the rule's consequence is the given proposition
            if(onTotal(cons)&&ante.equals(pp)&&r.check().booleanValue()) {
                support=(Proposition)cons.clone();
                break;
            }
        }
    }
    //end for loop
}
//end support for a proposition
...
return support;
}

// end method getSupportedProp
700

/*
 * For question begging, for a statement, try to get the backward chain
 * In order to attack, if there is no such a chain
 * return itself
 */
public boolean derivable(RuleProp rp) {
    boolean derived=false;
    if(onTotal(rp)) {
        derived=true;
    }
    ...
    //Let's deal with proposition first
    else if(rp.getClass().getName() == "Proposition") {
        ...
        Proposition goal=(Proposition)rp;
        Vector v=getRules(goal);
        int s=v.size();
        ...
        //No rules match as a consequence
        if(v.isEmpty()){derived=false;}
        while(v.isEmpty() == false && derived == false) {
            //Lower level rules
            Vector lv=new Vector();
            ...
            for(int i=0;i<s;i++) {
                Rule r=(Rule)v.elementAt(i);
                ...
                //derived
                if(onTotal(r.getAntecedent())&&r.check().booleanValue()){derived=true;break;}
            }
            ...
            //adding the lower level rule to
        }
    }
}
710
720
730

```



```

        Rule temp = (Rule) enum.nextElement();  

        {  

            lv.add(temp);  

        }  

    }  

} //end else  

} //end for loop  

v=lv;  

}//end while  

}//end proposition  

return derived;  

/*  

 * Check whether a proposition or a conditional is supported  

 it is necessary to check every statement in your partner's assertion list  

 */
public boolean supported(RuleProp rp) {  

    boolean supported=false;  

//Let's deal with proposition first  

if(rp.getClass().getName() == "Proposition") {  

    Proposition goal=(Proposition)rp;  

    Vector v=getRules(goal);  

    int s=v.size();  

//No rules match as a consequence  

if(v.isEmpty()){supported=false;}  

while(v.isEmpty() == false & supported == false) {  

    //Lower level rules  

    Vector lv=new Vector();  

for(int i=0;i<s;i++) {  

    Rule r=(Rule)v.elementAt(i);  

//derived  

if(onTotal(r.getAntecedent())){supported=true;break;}  

//adding the lower level rule to  

else {  

    Vector element=getRules(r.getAntecedent());  

    Enumeration enum = element.elements();  

    while (enum.hasMoreElements()) {  

        Rule temp = (Rule) enum.nextElement();  

        {  

            lv.add(temp);  

        }  

}

```

790

800

810

820

830

```

        }M
    }//end elseM
}//end for loopM
M
v=lv;M
M
}//end whileM
}//end propositionM
M
//Rules, temporarily like thisM
else {M
    Rule goal=(Rule)rp;M
    M
    Proposition backing=null;M
    M
    try{backing=goal.getWarrent();} catch (NullPointerException e){}M
    M
    //Backing exist and backing in the storeM
    if(backing!=null) {M
        if (onTotal(backing)){supported=true;}M
    }M
}//end ruleM
M
return supported;M
M
}//end of derive from totalM
M
M
/*M
 * Get the rule for a given proposition which is a consequenceM
 */M
public Vector getRules( Proposition p) {M
    Vector v=new Vector();M
    M
    int s=totalList.size();M
    for(int i=0;i<s;i++) {M
        RuleProp tem=(RuleProp)totalList.elementAt(i);M
        M
        //try to find a ruleM
        if(tem.getClass().getName() == "Rule") {M
            Rule temp=(Rule)tem;M
            if(temp.getConsequent().equals(p)&&temp.check().booleanValue()) {M
                v.add(temp);M
            }M
        }M
    }//end for loopM
    M
    return v;M
}//end methodM
M
/*M
Check whether a proposition is supported, not by rulesM
But if there is a supporting proposition in its storeM

```

```

    support itM
*/M
public boolean issupported(Proposition p, BooleanRuleBase selfKBS) {M
    boolean issupported=false;M
M
    Vector v=selfKBS.groundProps(p);M
M
    /*M
     * It cannot be further supportedM
*/M
    if (v.isEmpty() == false) {M
        Enumeration enum = v.elements();M
        while (enum.hasMoreElements()) {M
            Proposition test = (Proposition) enum.nextElement();M
            if(onAssertion(test)) {M
                issupported=true;M
                break;M
            }
        }M
        } //end of whileM
    } //end of there are groundM
M
    return issupported;M
}M
}M
} //end classM

```

900

910

A.8 ConflictSet.java

```

/**M
 * Class: ConflictSetM
 * Author: Tangming YuanM
 * Modification: Kristj"sn AvarssonM
**/M
import java.util.*;M
M
public class ConflictSet extends Vector {M
M
//To propose a method to produce a proposition for aM
//given vector of rules and propositionsM
M
public Proposition mergeToProps() {M
M
    String message=null;M
    if(isEmpty() == false) {M
        message="," + ((RuleProp)elementAt(0)).getContent() + ", ";M
        int n=size();M
M
        for(int i=1;i<=n-1;i++) {M
            RuleProp temp = (RuleProp) elementAt(i);M
M
            message=message+ " and " + temp.getContent() + ", ";M
        }
    }
}

```

10

20

```

    }  

    return new Proposition(message,new Boolean(true));  

}//end of method merge to proposition  

//Check if a given proposition is in the conflict set  

public boolean includes(RuleProp rp) {  

    boolean in=false;  

    Enumeration enum = elements();  

    while (enum.hasMoreElements()) {  

        RuleProp temp = (RuleProp) enum.nextElement();  

        if(temp.getClass().getName() == "Proposition" && rp.getClass().getName() == "Proposition") {  

            Proposition pt=(Proposition)temp;  

            Proposition pr=(Proposition)rp;  

            if(pt.equals(pr)) {  

                in=true;break;  

            }  

        } else if(temp.getClass().getName() == "Rule" && rp.getClass().getName() == "Rule") {  

            Rule pt=(Rule)temp;  

            Rule pr=(Rule)rp;  

            if(pt.equals(pr)) {  

                in=true;break;  

            }  

        }  

    }  

    return in;  

}//end of method  

//Check if the 2 size set is R, R->P style  

public boolean isRRP(Move previous) {  

    boolean is=false;  

    Proposition pc=(Proposition)previous.getMoveProp().clone();  

    RuleProp rp1=(RuleProp)elementAt(0);  

    RuleProp rp2=(RuleProp)elementAt(1);  

    if((rp1.getClass().getName() == "Rule" && rp2.getClass().getName() == "Proposition") || (rp2.getClass().getName() == "Rule"  

        if(rp1.getClass().getName() == "Rule" && rp2.getClass().getName() == "Proposition") {  

            Rule r1=rp1; Proposition p1=(Proposition)rp2;  

            if(r1.check().booleanValue() && r1.getConsequent().equals(pc) && r1.getAntecedent().equals(p1))  

                is=true;  

        }  

    if((rp2.getClass().getName() == "Rule" && rp1.getClass().getName() == "Proposition")) {  

        Rule r2=rp2; Proposition p2=(Proposition)rp1;  

        if(r2.check().booleanValue() && r2.getConsequent().equals(pc) && r2.getAntecedent().equals(p2))  

            is=true;  

    }  

    return is;  

}//end method  

//check if the 2 size set is P, ?P style  


```

```

public boolean isPNP() {  

    boolean is=false;  

    RuleProp rp1=(RuleProp)elementAt(0);  

    RuleProp rp2=(RuleProp)elementAt(1);  

    if(rp1.getClass().getName()=="Proposition"&&rp2.getClass().getName()=="Proposition") {  

        Proposition p1=(Proposition)rp1;Proposition p2=(Proposition)rp2;  

        if( p1.negate(p2))is=true;  

    }  

    return is;  

}  

//CHECK If the 3 size set is R, R->P, ?P style  

public boolean isPRNP() {  

    boolean is=false;  

    for(int i=0; i<=2; i++) {  

        RuleProp rp=(RuleProp)elementAt(i);  

        if(rp.getClass().getName()=="Rule") {  

            Rule r=(Rule)rp;  

            Proposition ante=r.getAntecedent();  

            Proposition cons=r.getConsequent();  

            Proposition denial=((Proposition)cons).denial();  

            if(includes(denial)&&includes(ante)&&r.check().booleanValue()) {  

                is=true;  

                break;  

            }  

        }  

    } //end for loop  

    return is;  

} //end method  

// Get consequent from the 3 size set P,R, R->?P  

// and 2 size R,R->?P  

public Proposition getConseq() {  

    Proposition p=null;  

    Enumeration enum = elements();  

    while (enum.hasMoreElements()) {  

        RuleProp temp = (RuleProp) enum.nextElement();  

        if(temp.getClass().getName()=="Rule") {  

            Rule r=(Rule)temp;  

            if(r.check().booleanValue())  

                p=(Proposition)(r.getConsequent()).clone();  

        }  

    }  

    if(size()==2)return p;  

    else return(p.denial());  

} //end of get  

/**  

 * Check if a given proposition or rule is on the total list  

 */  

public boolean onConflict(RuleProp rp) {  


```

```

boolean on=false;†
†
if(rp.getClass().getName() == "Proposition") {†
    Proposition p=(Proposition)rp;†
    †
    Enumeration enum = elements();†
    †
    while (enum.hasMoreElements()) {†
        RuleProp test = (RuleProp) enum.nextElement();†
        †
        if(test.getClass().getName() == "Proposition") {†
            Proposition ptest=(Proposition)test;†
            if(ptest.equals(p)) {†
                on=true;†
            }†
        }†
    }//end while†
} //end proposition†
†
//Rules†
else {†
    Rule r=(Rule)rp;†
    †
    Enumeration enum = elements();†
    †
    while (enum.hasMoreElements()) {†
        RuleProp test = (RuleProp) enum.nextElement();†
        †
        if(test.getClass().getName() == "Rule") {†
            Rule ptest=(Rule)test;†
            if(ptest.equals(r)) {†
                on=true;†
            }†
        }†
    }//end while†
} //end of rule compare†
return on;†
} //end of on total†
†
/*†
Check whether all the elements of user posed resolution demand†
are on C's assertion list†
*/†
public boolean onAssertion(CommitmentStore store) {†
    boolean on=true;†
    Enumeration enum = elements();†
    while (enum.hasMoreElements()) {†
        RuleProp temp = (RuleProp) enum.nextElement();†
        if (store.onAssertion(temp)==false) {†
            on=false;†
            break;†
        }†
    }†
}

```

140
150
160
170
180

```

        return on;
    } //end of class
}

public void show() {
    Enumeration enum = elements();
    while (enum.hasMoreElements()) {
        RuleProp temp = (RuleProp) enum.nextElement();
    }
}

```

190

A.9 DialogueHistory.java

```

/***
 * Class: DialogueHistory
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import java.util.*;
public class DialogueHistory extends Vector
{
    /**
     *Find whether a question has already been asked
     */
    public boolean contains(Move m)
    {
        boolean contain=false;
        Enumeration enum = elements();
        while (enum.hasMoreElements())
        {
            Move test = (Move) enum.nextElement();
            if(test.equals(m))
            {
                contain=true;
                break;
            }
        }
        return contain;
    }

    public void show()
    {
        Enumeration enum = elements();
        while (enum.hasMoreElements())
        {
            Move test = (Move) enum.nextElement();

```

10

20

30

```

        }
    }

    System.out.println(test.getContent());
}

}

```

40

A.10 DialogueManager.java

```

/*
 * Class: DialogueManager
 * Author: Tangming Yuan
 * Modification: Kristijan Evarsson
 */
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.text.BadLocationException;
class DialogueManager implements ActionListener,ItemListener{
    // Interface as the attribute
    GUI gui;
    BooleanRuleBase computerKBS=KBSManager.getCompKBS();
    Planner computerPlanner;
    //Two commitment store
    CommitmentStore studentStore;
    CommitmentStore computerStore;
    //Computer Relevant move
    private Vector crelv=new Vector();
    //Dialogue history vector,it is an ordered collection of move
    private DialogueHistory dilv=new DialogueHistory();
    //Whether the dialogue ends
    boolean end=false;
    //When the input is the resolve move type and move content is
    //from the computer's store
    ConflictSet compConflictSet=new ConflictSet();
    //Line
    int line=2;
    //Dialogue manager constructor
    public DialogueManager(GUI gui-){

```

10

20

30

```

        gui=gui_-;  

    }  

//Do something when triggered  

public void actionPerformed(ActionEvent ae){  

    if (ae.getSource()==gui.kbs){computerKBS.show();}  

    if (ae.getSource()==gui.newGame){  

        try {  

            start();  

        } catch (BadLocationException ex) {  

            ex.printStackTrace();  

        }  

    }  

//end of start game  

// Dialogue is going on when button "Go" is pressed  

if(ae.getSource()==gui.confirmButton && dilv.size()!=0){  

    if(end==false){  

        studentMove();  

        String message=endingMessage();  

        if(message!=null)endDebate();  

    }  

    if(end==false){  

        computerMove();dilv.show();  

    }  

}  

//end of actionperformed  

//Start game  

public void start() throws BadLocationException{  

    gui.historyText.remove(0,gui.historyText.getLength());  

// URL bla = GUI.class.getResource("emptyHtml.html");  

    end=false;  

    line=1;  

    gui.historyText.insertString(gui.historyText.getLength(),"01: Computer>Is it the case that CP is acceptable?\n",gu  

    gui.historyText.setCharacterAttributes(4, 8, gui.historyText.getStyle("Red"), true);  

//Create a planner for the compter  

    computerPlanner=new Planner('C');  

//Clean up the move type vector or choice  

    dilv.removeAllElements();  

    gui.kbs.enable();  

    studentStore=new CommitmentStore("Student");  

    computerStore=new CommitmentStore("Computer");  

//Dialogue history vector upgrade  

    dilv.addElement(new Move('C',"Question",new Proposition("CP is acceptable",new Boolean(true))));  

//Turn message  

    gui.messageLabel.setText("It is your turn now, select the move type choice from the left box.");  

//Add pop up menu listener  


```

```

InputGuideListener inputGuideListener = new InputGuideListener(gui);  

gui.moveTypeChoice.addMouseListener(inputGuideListener);  

//Initialise input choice and move type choice  

MoveChoiceInitialiser.initialiseMoveType(gui);  

MoveChoiceInitialiser.initChoice(gui);  

InputManager.prefixInputChoice(gui,dilv,computerStore);  

}  

//Method move  

public void studentMove(){  

String message=null;  

Move previous=(Move)dilv.lastElement();  

char currentTurn;  

//decide who is going to make a move  

if(previous.getTurn()=='C')currentTurn='S';  

else currentTurn='C';  

//Make sure it is his turn  

if(currentTurn=='S'&&end==false){  

    Move studentMove=InputManager.getInput(gui, dilv, computerStore, studentStore);  

    //((Move)relv.elementAt(i));  

//Conduct dialogue and make move  

line+=1;  

try {  

    studentMove.display(line,gui.historyText,previous.getType(),previous.getContent());  

} catch (BadLocationException ex) {  

    ex.printStackTrace();  

}  

//check whether it is a legal one  

message=Filler.getMessage(studentMove, dilv, computerStore, studentStore);  

//The move is legal  

if(message==null){  

    //Upgrade commitment store  

CommitmentManager.commit(studentMove,previous,studentStore,computerStore,gui.studentCommitmentStore,  

dilv.addElement(studentMove);  

gui.messageLabel.setText("Computer is thinking about it...");  

} else{  

    final Move m=new Move('R',"Message",new Proposition(message,new Boolean(true)));  

Runnable updateAComponent = new Runnable() {  

    public void run() {  

        try{Thread.sleep(1000);} catch(InterruptedException ie){}  

        line+=1;  

        try {  

            m.display(line,gui.historyText,"Assertion", " ");  

        } catch (BadLocationException ex) {  

            ex.printStackTrace();  

    }  

}

```

100

110

120

140

```

        };}
    };  

    SwingUtilities.invokeLater(updateAComponent);  

}  

}//end referee  

}//end student turn  

}//end student move  

public void computerMove()//Computer(Chris makes a move)  

{  

    String message=null;  

final Move previous=(Move)dilv.lastElement();  

//Make sure it is his turn  

if(previous.getTurn()=='s'){  

    Runnable updateAComponent = new Runnable(){  

        public void run(){  

            if(end==false){  

                try{Thread.sleep(1000);} catch(InterruptedException ie){}  

                Vector relv=computerPlanner.produceRelevantMove(dilv,computerStore,studentStore,computerKBS);  

                int j=relv.size();  

                int i= ( int ) ( Math.random() * j );  

                Move currentMove=(Move)relv.elementAt(i);  

//Conduct dialogue and make move  

                line+=1;  

                try {  

                    currentMove.display(line,gui.historyText,previous.getType(),previous.getContent());  

                } catch (BadLocationException ex) {  

                    ex.printStackTrace();  

                }  

//Upgrade commitment store  

                CommitmentManager.commit(currentMove,previous,computerStore,studentStore,gui.computerComm);  

//dialogue history vector upgrade  

                dilv.addElement(currentMove);  

//turn message  

                gui.messageLabel.setText("It is your turn now, select the move type choice from the left");  

//Computer will support student to fix the move type to avoid rule breaking  

                InputManager.prefixInputChoice(gui,dilv,computerStore);  

//test  

                dilv.show();  

                if(endingMessage()!=null)endDebate();  

            }  

        }  

    };  

    SwingUtilities.invokeLater(updateAComponent);  

}

```

```

        } //end computer's move
    } //end of making a move
}

//Method when imply checkbox is selected
public void itemStateChanged(ItemEvent e){
    if(e.getSource() == gui.implyCheckbox){
        InputManager.setTempInput(gui);
    }
} //end Item stateChanged

//Decide win or loss
public String endingMessage() {
    String message=null;
    Move currentMove=(Move)dilv.lastElement();
    Proposition stuThesis=(Proposition)((Move)dilv.elementAt(1)).getMoveProp().clone();
    Proposition compThesis=stuThesis.denial();
    if (currentMove.getTurn()=='S') {
        if(studentStore.onAssertion(stuThesis)==false) {
            message="You seem to have given up your view.";
        } else if(studentStore.onAssertion(compThesis)) {
            message="You seem to have changed your view.";
        }
    }
    if (currentMove.getTurn()=='C') {
        if(computerStore.onAssertion(compThesis)==false) {
            message="Many Congratulations, you are doing very well.";
        } else if(computerStore.onAssertion(stuThesis)) {
            message="Congratulations, You win.";
        }
    }
    if(currentMove.getType() == "Withdraw" && dilv.size() == 2) {
        message="You have no view on the issue, so we can not debate any more, you can try later when you g
    }
    return message;
}
} //end end Debate method

//Decide win or loss
public void endDebate() {
    final String message=endingMessage();
    Runnable updateAComponent = new Runnable() {
        public void run() {
            if(end==false) {
                try{Thread.sleep(1000);} catch(InterruptedException ie){}
                end=true;
                Move m=new Move('R',"Message",new Proposition(message,new Boolean(true)));
                line+=1;
            }
        }
    };
}

```

200
210
220
230
240
250

```

        try {^M
            m.display(line.gui.historyText," ", " ");^M
        } catch (BadLocationException ex) {^M
            ex.printStackTrace();^M
        }^M
        gui.messageLabel.setText("Game Over ! To start a new game, choose 'New game' item from the Game menu");^M
    };^M
    SwingUtilities.invokeLater(updateAComponent);^M
}//end end Deabte method^M
^M
^M
}//end of class dialogue manager

```

260

A.11 Filler.java

```

/**^M
 * Class: Filler^M
 * Author: Tangming Yuan^M
 * Modification: Kristján Ávarsson^M
 */^M
import java.util.*;^M
^M
class Filler {^M
    // A lot of work to do in this part: Parameter: dialv,both commitment store,kbs^M
    public static String getMessage(Move studentMove,Vector dilv,CommitmentStore computerStore,CommitmentStore studentMoveStore)
    {
        String message=null;^M
        Move previous=(Move)dilv.lastElement();^M
        ^M
        //We get the previous proposition and make a copy^M
        RuleProp rp=studentMove.getObject();^M
        ^M
        Proposition stuThesis=null;//(Proposition)((Move)dilv.elementAt(1)).getMoveProp().clone();^M
        Proposition compThesis=null;//stuThesis.denial();^M
        if(dilv.size()>1) {^M
            stuThesis=(Proposition)((Move)dilv.elementAt(1)).getMoveProp().clone();^M
            compThesis=stuThesis.denial();^M
        }^M
        ^M
        /*^M
        1 Illegal challenge^M
        */^M
        if(studentMove.getType() == "Challenge" && computerStore.onAssertion(rp) == false) {^M
            message="The statement '" + rp.getContent() + "' is stated by you and agreed by the computer, you can only agree by the computer";^M
        }^M
        ^M
        /*^M
        2.Begging the question^M
        */^M
    }^M
}

```

20

30

```

if(studentMove.getType() == "Ground") {  

    Proposition ps=(Proposition)rp;  

    if((computerStore.derivable(ps)==false)&&studentStore.onClaimStack(ps)) {  

        //Using statements one fails to prove  

        if(studentStore.onAssertion(ps)==false) {  

            message="You are begging the question '" + ps.getContent() + "', which you have failed to prove";  

        } else {  

            message="You are begging the question '" + ps.getContent() + "', which is under dispute, please try again";  

        }  

    } /*  

    3. Illegal asking for resolve  

*/  

/*  

if(studentMove.getType() == "Resolve") {  

    ConflictSet compConflictSet=studentMove.getConflictSet();  

    System.out.println(compConflictSet);  

    compConflictSet.show();  

    if(compConflictSet.size()==2&&compConflictSet.isPNP()==false) {  

        message="The contents of resolution should be P,?P or P, R, R implies not P style, please try again";  

    } else if(compConflictSet.size()==3&&compConflictSet.isPRNP()==false) {  

        message="The contents of resolution should be P, ?P, or P, R, R->?P style, please try again";  

    }  

} //end of illegal resolve  

*/  

5.      Repeat statement, if S assert C's thesis, it is ok  

temporarily use for on both assertions  

if(studentMove.getType() == "Assertion"&&computerStore.onAssertion(rp)&&studentStore.onAssertion(rp)) {  

    Proposition pp=null;  

    if(rp.getClass().getName() == "Proposition") {  

        pp=(Proposition)rp;  

    }  

    if (pp.equals(compThesis)==false) {  

        message="The statement '" + rp.getContent() + "' , has already in both positions, it is not necessary to add it again";  

    }  

}

```

```

/*M
6      Rule is not well formed exceptionM
*/M
if(rp.getClass().getName() == "Rule") {M
    Rule rs=(Rule)rp;M
    M
    if(rs.getAntecedent().equals(rs.getConsequent())) {M
        message="This is not a sound conditional, the head and the tail should be different (e.g. R imp
    } else if(rs.getAntecedent().negate(rs.getConsequent())) {M
        message="It is not a valid conditional, please try again!";M
    }M
}M
return message;      M
}M
M
}//end of class FillerM

```

A.12 FocusShiftDemolishManager.java

```

/**M
 * Class: FocusShiftDemolishManagerM
 * Author: Tangming YuanM
 * Modification: Kristj"sn EvarssonM
**/M
/*M
This class is defined to deal with the demolish strategyM
when C has run out of the build strategyM
*/
import java.util.*;M
public class FocusShiftDemolishManager {M
    M
    public static Move act(DialogueHistory dilv, CommitmentStore selfStore, CommitmentStore partnerStore, BooleanRuleBase
    Move m=null;M
    M
    Enumeration enum = partnerStore.assertionList.elements();M
    while (enum.hasMoreElements()&&m==null) {M
        RuleProp temp = (RuleProp) enum.nextElement();M
        M
        /*M
        Find a proposition on partner's assertion list, which isM
        either a support of C's thesis or a against S's viewM
        */
        if (temp.getClass().getName() == "Proposition") {M
            Proposition focus=(Proposition)temp;M
            if((focus.equals(planner.computerThesis.denial())&&selfKBS.challengable(planner.computerThesis.denial(), par
                m=new Move(planner.turn, "Challenge", planner.computerThesis.denial());M
            } else if (selfKBS.support(focus, planner.computerThesis.denial())||selfKBS.againstSupport(focus, planner.com
                m=TacticManager.getAction(focus, dilv, selfStore, partnerStore, selfKBS, planner);M
            }M
        }M
    }//end whileM

```

```

    M
    return m;M
}//end of method actM
M
}//end of class FocusShiftDemolishManager
```

A.13 FocusShiftManager.java

```

/***
 * Class: FocusShiftManager
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import java.util.*; 

public class FocusShiftManager { 
    public static Move execute(DialogueHistory dilv, CommitmentStore selfStore, CommitmentStore partnerStore, BooleanRule
        Move response=null; 
        PlanSet ps=selfKBS.getPlanSet(planner.computerThesis); 
        int i= ( int ) ( Math.random() * 2); 
        /* 
         The priority is given to open a subtopic 
        */ 
        switch (i) { 
            case 0: 
                if(response==null) { 
                    response=ps.startSubtopic(selfStore, partnerStore); 
                } 
                /* 
                 2) Otherwise build a plan 
                */ 
                if (response==null) { 
                    response=ps.startBuildPlan(dilv,planner); 
                } 
            case 1: 
                /* 
                 2) Otherwise build a plan 
                */ 
                if (response==null) { 
                    response=ps.startBuildPlan(dilv,planner); 
                } 
                if(response==null) { 
                    response=ps.startSubtopic(selfStore, partnerStore); 
                } 
        } //end switch
}

```

```

/*
 * Two steps build plan
 */
if(response==null) {
    response=ps.getQuestionMove(dilv, planner);
}

if(response==null) {
    response=new Move('C',"Assertion", new Proposition("if you have anything more to say, you can go on"));
}

return response;
}//end of execute focus shift method
}

```

50 60

A.14 GUI.java

```

/*
 * GUI.java
 *
 * Created on April 1, 2006, 3:02 PM
 */

/**
 *
 * author      kristjan
 */
import java.awt.*;
import java.io.*;
import javax.swing.text.BadLocationException;
import javax.swing.text.Style;
import javax.swing.text.StyleConstants;
import javax.swing.text.StyleContext;
import javax.swing.text.StyledDocument;

public class GUI extends javax.swing.JFrame {

    InterfaceManager interfaceManager;
    DialogueManager dialogueManager;
    public StyledDocument historyText;

    private FileDialog openD =new FileDialog(this,"open");
    private FileDialog saveD =new FileDialog(this,"Save as",FileDialog.SAVE);

    //Student commitmentStore
    public java.awt.List studentCommitmentStore = new java.awt.List(10,true);
    public java.awt.List computerCommitmentStore = new java.awt.List(10,true);

    public boolean visible = false;//if inputChoice is empty

    int index = 0;
    Style styleII;
    Style     style;

```

10 20 30

```

/** Creates new form GUI */
public GUI() {
    initComponents();
    historyText = textPane.getStyledDocument();
    textPane.setEditable(false);
    //addStylesToDocument(doc);

    //Create interface Manager
    interfaceManager = new InterfaceManager(this);
    dialogueManager = new DialogueManager(this);

    help.addActionListener(interfaceManager);
    exit.addActionListener(interfaceManager);
    save.addActionListener(interfaceManager);
    aboutMenu.addActionListener(interfaceManager);

    //dialogmanager
    newGame.addActionListener(dialogueManager);
    kbs.addActionListener(dialogueManager);
    confirmButton.addActionListener(dialogueManager);
    implyCheckbox.addItemListener(dialogueManager);

    //Blink message
    Blink blink=new Blink(this);
    blink.activity.start();

    styleII = textPane.addStyle("Blue", null);
    StyleConstants.setForeground(styleII, Color.blue);

    style = textPane.addStyle("Red", null);
    StyleConstants.setForeground(style, Color.red);
}

/*
 * This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
//<editor-fold defaultstate="collapsed" desc=" Generated Code ">//GEN-BEGIN:initComponents
private void initComponents() {
    jSeparator3 = new javax.swing.JSeparator();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jScrollPane = new javax.swing.JScrollPane();
    moveTypeChoice = new javax.swing.JList();
    jScrollPane1 = new javax.swing.JScrollPane();
    inputChoice = new javax.swing.JList();
    jPanel2 = new javax.swing.JPanel();
    confirmButton = new javax.swing.JButton();
    progressBar = new javax.swing.JProgressBar();
    implyCheckbox = new javax.swing.JCheckBox();
    blinkingPanel = new javax.swing.JPanel();
    messageLabel = new javax.swing.JLabel();
    jPanel3 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jScrollPane1_historyText = new javax.swing.JScrollPane();
    textPane = new javax.swing.JTextPane();
    jMenuBar1 = new javax.swing.JMenuBar();
    fileMen = new javax.swing.JMenu();
    newGame = new javax.swing.JMenuItem();
}

```

```

save = new javax.swing.JMenuItem();
sepMen1 = new javax.swing.JSeparator();
exit = new javax.swing.JMenuItem();
helpMen = new javax.swing.JMenu();
help = new javax.swing.JMenuItem();
kbs = new javax.swing.JMenuItem();
sepMenu2 = new javax.swing.JSeparator();
aboutMenu = new javax.swing.JMenuItem();                                100

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Human Computer Debating System");
setResizable(false);                                                 110

jLabel2.setLabelFor(Jscroll);
jLabel2.setText("Move Type Choice");

jLabel3.setLabelFor(inputChoice);
jLabel3.setText("Move Content Choice");

moveTypeChoice.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
Jscroll.setViewportView(moveTypeChoice);                                120

inputChoice.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
jScrollPane1.setViewportView(inputChoice);

confirmButton.setMnemonic('c');
confirmButton.setText("Confirm");
confirmButton.setToolTipText("Confirm your input");
confirmButton.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED)); 130

progressBar.setBackground(java.awt.Color.red);
progressBar.setForeground(java.awt.Color.blue);
progressBar.setValue(50);
progressBar.setString("Debate status");
progressBar.setStringPainted(true);

implyCheckbox.setMnemonic('i');
implyCheckbox.setText("Implies");
implyCheckbox.setToolTipText("Make an implementation statement");
implyCheckbox.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
implyCheckbox.setMargin(new java.awt.Insets(0, 0, 0, 0));           140

org.jdesktop.layout.GroupLayout jPanel2Layout = new org.jdesktop.layout.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(jPanel2Layout.createSequentialGroup()
            .add(confirmButton, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 155, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(14, 14, 14)
            .add(implyCheckbox)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 173, Short.MAX_VALUE)      150
            .add(progressBar, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 459, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(confirmButton))
    );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(jPanel2Layout.createSequentialGroup()
            .add(confirmButton, org.jdesktop.layout.GroupLayout.BASELINE)
            .add(implyCheckbox)
            .add(progressBar, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 25, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(confirmButton))                                         160
);

```



```

void aboutMenu() {
    new AboutFrame();
}
//To set and get the correct index when selecting from computer or human store.
public void setIndex(int input){
    index = input;
}
public int getIndex(){
    return index;
}
public void matchIndex(int i,int ind){
    if(i==1){//withdraw
        for(int k=0; k < studentCommitmentStore.getItemCount();k++){
            if(studentCommitmentStore.getItem(k).equals(InputGuideListener.studentVector.get(ind))){
                setIndex(k);
            }
        } else{//Challenge
            for(int k=0; k < computerCommitmentStore.getItemCount();k++){
                if(computerCommitmentStore.getItem(k).equals(InputGuideListener.computerVector.get(ind))){
                    setIndex(k);
                    System.out.println("Index is then: "+k);
                }
            }
        }
    }
}

private StyledDocument StyledDocument(StyleContext sc) {
    return null;
}

// Variables declaration - do not modify//GEN-BEGIN:variables
public javax.swing.JScrollPane jScrollPane;
private javax.swing.JScrollPane jScrollPane_historyText;
public javax.swing.JMenuItem aboutMenu;
private javax.swing.JPanel blinkingPanel;
public javax.swing.JButton confirmButton;
public javax.swing.JMenuItem exit;
public javax.swing.JMenu fileMen;
public javax.swing.JMenuItem help;
private javax.swing.JMenu helpMen;
public javax.swing.JCheckBox implyCheckbox;
public javax.swing.JList inputChoice;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparator3;
public javax.swing.JMenuItem kbs;
public javax.swing.JLabel messageLabel;
public javax.swing.JList moveTypeChoice;
public javax.swing.JMenuItem newGame;
private javax.swing.JProgressBar progressBar;
public javax.swing.JMenuItem save;
private javax.swing.JSeparator sepMen1;
private javax.swing.JSeparator sepMenu2;
public javax.swing.JTextPane textPane;
// End of variables declaration//GEN-END:variables

```

A.15 InputGuideListener.java

```

/*
 * Class: InputGuideListener
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.util.Vector;

public class InputGuideListener implements MouseListener{
    // Interface as the attribute
    GUI gui;
    static Vector studentVector = new Vector();
    static Vector computerVector = new Vector();
    //Dialogue manager constructor
    public InputGuideListener(GUI gui_){
        gui=gui_;
    }
    public void mouseClicked(MouseEvent mouseEvent) {
        //Hide the move content choice
        int i=gui.moveTypeChoice.getSelectedIndex();
        String mt=(String)InputManager.dynamicMoveType.elementAt(i);
        if(mt=="Challenge"||mt=="Resolve"||mt=="Withdraw"){
            if(mt=="Withdraw") {
                gui.messageLabel.setText("Please select one statement from the right box and press confirm button.");
                studentVector.clear();
                String input;
                for(int k=0; k < gui.studentCommitmentStore.getItemCount();k++){
                    input = gui.studentCommitmentStore.getItem(k);
                    if(input.charAt(0)!=('*')) {
                        studentVector.add(input);
                    }
                }
                gui.inputChoice.setListData(studentVector);
            } else if(mt=="Challenge") {
                gui.messageLabel.setText("Please select a statement from the right box and press confirm button.");
                computerVector.clear();
                String inputC;
                for(int j=0; j < gui.computerCommitmentStore.getItemCount();j++){
                    inputC = gui.computerCommitmentStore.getItem(j);
                    if(inputC.charAt(2)=='\'' ) {
                        System.out.println("Fann stk");
                    }
                }
                if(inputC.charAt(0)!=('*')) {

```

```

        if(inputC.charAt(2)!=(\'\\')) {
            computerVector.add(inputC);
        }
    }
    gui.inputChoice.setListData(computerVector);
} else {
    gui.messageLabel.setText("Select a conflict set(e.g. (P and not P) or(P, R, R implies not P)) and
    gui.visible=false;
}
else if(mt=="Assertion"||mt=="Question") {
    if(gui.visible==false){
        gui.inputChoice.setListData(MoveChoiceInitialiser.getMixer());
        gui.visible = true;
    }
    gui.messageLabel.setText("Select a statement or construct a conditional (P implies Q) with the checkbox
}
//Others such as 'yes', 'no', withdraw, withdraw after a resolve or challenge,
else {
    gui.inputChoice.removeAll();
    gui.messageLabel.setText("Press confirm button.");
}
}

public void mouseEntered(MouseEvent mouseEvent) {
}
public void mouseExited(MouseEvent mouseEvent) {
}
public void mousePressed(MouseEvent mouseEvent) {
}
public void mouseReleased(MouseEvent mouseEvent) {
}
}

}

```

70

80

A.16 InputManager.java

```

/***
 * Class: InputManager
 * Author: Tangming Yuan
 * Modification: Kristj"sn Avarsson
 */
import java.util.*;
import javax.swing.JOptionPane;
import javax.swing.*;

public class InputManager {
    static Move tempMove;
}

}

```

10

```

static Vector dynamicMoveType=new Vector();  

static Vector moveContentVector=MoveChoiceInitialiser.getMoveContent();//new Vector();  

static Vector emptyVector = new Vector();  

public static void prefixInputChoice(GUI gui,Vector dilv,CommitmentStore computerStore) {  

    emptyVector.add("");  

    dynamicMoveType.removeAllElements();  

    setMoveType(gui,dilv,computerStore);  

}  

//method of dynamic setting of move type  

private static void setMoveType(final GUI gui,Vector dilv,CommitmentStore computerStore) {  

    //Guideline message  

    String message=null;  

    Move previousMove=(Move)dilv.lastElement();  

    String previousMoveType=previousMove.getType();  

    RuleProp prevrp=previousMove.getObject();  

    /*  

    Previous move is question, the available move types are  

    Yes-No-No commitment  

*/  

    if(previousMoveType.equals("Question")) {  

        //dynamicMoveType.add("--Select--");  

        dynamicMoveType.add(" Yes");  

        dynamicMoveType.add(" No");  

        dynamicMoveType.add(" I am not sure about it");  

        gui.inputChoice.setListData(emptyVector);  

        gui.visible = false;  

        showDynamicMoveType(gui,dilv);  

    }  

    /*  

    Previous move is challenge, it follows by a ground,  

    No commitment option  

    Or a resolution demand option  

*/  

    else if(previousMoveType.equals("Challenge")) {  

        //give a reason  

        //dynamicMoveType.add("--Select--");  

        dynamicMoveType.add("Assertion");  

        dynamicMoveType.add("I don't know why "+previousMove.getContent()+" .");  

        //If it is a consequence in computer store, prompt resolve move type  

        Proposition prevp=(Proposition)prevrp;  

        if(computerStore.getRealPremises(prevp).isEmpty()==false) {  

            dynamicMoveType.add("Resolve");  

    }
}

```

20

30

40

50

60

```

}Â
Â
//Make the interface niceÂ
dynamicMoveType.add("");Â
Â
showDynamicMoveType(gui,dilv);Â
}Â
Â
/*Â
previous move is resolve, then follows 2 or three optionsÂ
*/
Â
else if(previousMoveType.equals("Resolve")) {Â
Â
ConflictSet con=previousMove.getConflictSet();Â
Â
Enumeration enum = con.elements();Â
while (enum.hasMoreElements()) {Â
    RuleProp temp = (RuleProp) enum.nextElement();Â
    dynamicMoveType.add("I don't think "+temp.getContent()+".");Â
}
Â
//If it is the second type of resolve, allow a statementÂ
Â
if(con.size()==2&&con.isPNP()==false) {Â
    dynamicMoveType.add("I think "+con.getConseq().getContent()+".");Â
}
Â
Â
//Make the interface niceÂ
dynamicMoveType.add("");Â
Â
showDynamicMoveType(gui,dilv);Â
}Â
Â
/*
Previous move is an assertionÂ
*/
Â
else if(previousMoveType.equals("Assertion")||previousMoveType.equals("Concession")||previousMoveType.equals("Gr
//dynamicMoveType.add("--Select--");Â
dynamicMoveType.add("Assertion");Â
dynamicMoveType.add("Question");Â
dynamicMoveType.add("Challenge");Â
dynamicMoveType.add("Withdraw");Â
Â
//Allowed when there is a conflictÂ
if(computerStore.getRealConflictSet().isEmpty()==false) {Â
    dynamicMoveType.add("Resolve");Â
}
Â
showDynamicMoveType(gui,dilv);Â
}Â
Â

```

```

/*
The last option is a withdrawal
*/
else {
    //dynamicMoveType.add("—Select—");
    dynamicMoveType.add("Assertion");
    dynamicMoveType.add("Question");
    dynamicMoveType.add("Challenge");
    dynamicMoveType.add("Withdraw");
}

/*
Allowed when there is a conflict
if(prevrp.getClassName() == "Proposition") {
    Proposition prevp=(Proposition)prevrp;
    if((prevp!=null && computerStore.getRealPremises(prevp).isEmpty()==false)||(computerStore.getRealConf...
        dynamicMoveType.add("Resolve");
}
showDynamicMoveType(gui,dilv);
}

//Pop up the menu
Runnable updateAComponent = new Runnable() {
    public void run() {
        gui.moveTypeChoice.setVisible(true);
    }
}
SwingUtilities.invokeLater(updateAComponent);
}//end set move type method

//Method of show move type
private static void showDynamicMoveType(GUI gui, Vector dilv) {
    gui.moveTypeChoice.removeAll();
    Vector moveVector = new Vector();
    Enumeration enum =dynamicMoveType.elements();
    while (enum.hasMoreElements()) {
        String test = (String) enum.nextElement();
        if(test.equals("Assertion")) {
            if(((Move)dilv.lastElement()).getType() == "Challenge"){moveVector.add("Because. .");} else{moveVector.add("...
            } else if(test.equals("Challenge")){moveVector.add("Why is it the case that. .?");} else if(test.equals("Quest...
        }
        gui.moveTypeChoice.setListData(moveVector);
    }
}//end method

//Method of get the full move when Button "Go" is pressed
public static Move getInput(GUI gui, Vector dilv, CommitmentStore computerStore,CommitmentStore studentStore) {
    Move fullMove=null;
    Move previousMove=(Move)dilv.lastElement();
    String previousType=previousMove.getType();
    RuleProp prevrp=previousMove.getObject();
}

```

```

    //
//Make sure it is not a rule
if (tempMove==null) {
    //
//get move type from the gui
int i=gui.moveTypeChoice.getSelectedIndex();
String mt=(String)dynamicMoveType.elementAt(i);
if(mt==" Yes"||mt==" No"||mt==" I am not sure about it") {
    if(mt==" Yes"){fullMove=new Move('S',"Concession",prevrp);}//fullMove.setType("Concession");}
    else if(mt==" No") {
        if(prevrp.getClass().getName()!="Proposition")
            fullMove=new Move('S',"Concession",((Proposition)prevrp).denial());
        else {fullMove=new Move('S',"Concession",((Rule)prevrp).denial());}
    } else {fullMove=new Move('S',"Withdraw",prevrp);}
}
//
//get input from computer's store
else if(mt=="Resolve") {
    int[] a=gui.inputChoice.getSelectedIndices();
    ConflictSet compConflictSet = computerStore.getInput(a);
    if(compConflictSet.isEmpty()==true||compConflictSet.size()==1) {
        String message="If you ask the computer to resolve conflicts, you need \n"
                    +"to select two (P, not P) or three (P, R, R implies not P)\n"
                    +"conflict statements from computer's positions to resolve.";
        JOptionPane.showMessageDialog(gui,message);
    } else if(compConflictSet.size()>3) {
        String message="Sorry, this advanced feature--over 3 conflict elements is \n"
                    +"not currently available in this version, try to select \n"
                    +"two (P, not P) or three (P, R, R implies not P) options \n"
                    +"from computer's positions to resolve.";
        JOptionPane.showMessageDialog(gui,message);
    }
    else if((compConflictSet.size()==2&&compConflictSet.isPNP()==false)||compConflictSet.size()==3&&compConflictSet.isPNP()==true) {
        String message="If you ask the computer to resolve conflicts, you need \n"
                    +"to select two (P, not P) or three (P, R, R implies not P)\n"
                    +"conflict statements from computer's positions to resolve.";
        JOptionPane.showMessageDialog(gui,message);
    }
    else {
        Proposition p=compConflictSet.mergeToProp();
        compConflictSet.show();
        fullMove=new Move('S',"Resolve", p, compConflictSet);
    }
}

```

```

    M
//Get input from computer's assertion listM
else if(mt=="Challenge") {M
    int[] a=gui.inputChoice.getSelectedIndices();M
    gui.matchIndex(2,a[0]);M
    a[0]=gui.getIndex();M
    if(a.length==0) {M
        String message="If you want to make a challenge, you needs to select one \n" +
                      +"statement from computer's positions, which are indeed \n" +
                      +"advanced by the computer.";M
    M
    JOptionPane.showMessageDialog(gui,message);M
} else if(a.length>1) {M
    String message="To challenge several positions together is not currently \n" +
                  +"available, please select one statement from computer's \n" +
                  +"positions which are indeed stated by the computer.";M
    M
    JOptionPane.showMessageDialog(gui,message);M
} else {M
    RuleProp rp=(RuleProp)computerStore.totalList.elementAt(a[0]);M
    if(rp.getClass().getName()!="Rule") {M
        String message="To challenge a conditional (e.g. P->Q) is not currently \n" +
                      +"available, if you have problem with this conditional, you \n" +
                      +"can withdraw it or try other options.";M
    M
    JOptionPane.showMessageDialog(gui,message);M
} else {M
    fullMove=new Move('S', mt, rp);M
}M
}M
} else if(mt=="Withdraw") {M
    int[] a=gui.inputChoice.getSelectedIndices();M
    gui.matchIndex(1,a[0]);M
    a[0]=gui.getIndex();M
    if(a.length==0) {M
        String message="If you want to make a withdrawal, you needs to select one \n" +
                      +"statement from your own positions. \n";M
    M
    M
    JOptionPane.showMessageDialog(gui,message);M
} else if(a.length>1) {M
    String message="You can withdraw only one options once, please select one\n" +
                  +"statement from your own positions.";M
    M
    M
    JOptionPane.showMessageDialog(gui,message);M
} else {M
    RuleProp mc=(RuleProp)studentStore.totalList.elementAt(gui.inputChoice.getSelectedIndex());M
    fullMove=new Move('S', mt, mc);M
}M
}M
}

```

230 240 250 260 270

```

    }
    else if(mt=="Assertion") {
        Proposition mc=(Proposition)((Proposition)moveContentVector.elementAt(gui.inputChoice.getSelectedIndex()));
        //
        //Illegal ground exception
        if(previousType=="Challenge") {
            if(gui.impliesCheckbox.isSelected()==true) {
                String message="You can only use a proposition as a ground, please select one\n"+statement from the move content choice.";
                JOptionPane.showMessageDialog(gui,message);
            } else fullMove=new Move('S', "Ground", mc);
        } else fullMove=new Move('S', mt, mc);
    }
    else if(mt=="Question") {
        Proposition mc=(Proposition)((Proposition)moveContentVector.elementAt(gui.inputChoice.getSelectedIndex()));
        fullMove=new Move('S', "Question", mc);
    }
    else {
        //either after a resolution demand or the NC after challenge
        if(previousType=="Challenge"&&i==1)fullMove=new Move('S',"Withdraw",prevrp);
        //
        //Handle the withdraw input after a resolve
        if(previousType=="Resolve") {
            ConflictSet con=previousMove.getConflictSet();
            //
            //Input exception
            if((con.size()==2&&con.isPNP()&&i==2)||i==3) {
                String message="You need to select a move choice.\n";
                JOptionPane.showMessageDialog(gui,message);
            }
            JOptionPane.showMessageDialog(gui,message);
        }
        //Second type of resolve, for affirmation
        if(con.size()==2&&con.isPNP()==false&&i==2)fullMove=new Move('S',"Assertion",con.getConseq());
        else fullMove=new Move('S',"Withdraw",RuleProp.con.elementAt(i));
    }
}
}

//If the implies choice is selected, which means it is a rule
else {
    //get rule consequence
    Proposition con=(Proposition)((Proposition)moveContentVector.elementAt(gui.inputChoice.getSelectedIndex())).clone();
    if(tempMove.getType()!="Assertion"&&previousType=="Challenge")
        fullMove=new Move('S',"Ground",new Rule(tempMove.getMoveProp(),con));
    else fullMove=new Move('S',tempMove.getType(),new Rule(tempMove.getMoveProp(),con));
}
}

//clear

```

```

tempMove=null;M
gui.impliesCheckbox.setSelected(false);M
M
return fullMove;M
}//end get input methodM
M
//Method of get and set the temp move when checkbox is selectedM
public static void setTempInput(GUI gui) {M
    if(gui.impliesCheckbox.isSelected()==true) {M
        //get move type from the GUI
        String mt=(String)dynamicMoveType.elementAt(gui.moveTypeChoice.getSelectedIndex());M
        M
        //get move content from the GUI
        Proposition mc=(Proposition)((Proposition)moveContentVector.elementAt(gui.inputChoice.getSelectedIndex())).clone();
        gui.messageLabel.setText(gui.inputChoice.getSelectedItem().toString());M
        tempMove=new Move('S',mt,mc);M
    } else tempMove=null;M
}// end methodM
}

```

340

A.17 InterfaceManager.java

```

/**M
 * Class: InterfaceManagerM
 * Author: Tangming YuanM
 * Modification: Kristj?sn ?varssonM
 **/M
import java.awt.event.*;M
import javax.swing.text.BadLocationException;M
M
/**M
 * Manage action from Menu in GUI classM
 * In agent, the GUI does not handle what to do on action.M
 *M
 * Modified: 01/04/06 by Kristj?n ?varsson & Tangming YuanM
 */M
M
class InterfaceManager implements ActionListener {M
    // Interface as the attributeM
    GUI gui;M
    M
    /**M
     * Constructor.M
     */M
    public InterfaceManager(GUI gui_) {M
        gui=gui_;M
    }M
    M
    /**M
     * Determines what action to doM

```

10

20

```

30
 */  

public void actionPerformed(ActionEvent e){  

    //get help  

    if (e.getSource()==gui.help) {  

        new Tips();  

    }  

    //Quit the game  

    else if (e.getSource()==gui.exit) {  

        System.exit(0);  

    }  

    //Save the context in text area  

    else if (e.getSource()==gui.save) {  

        try {  

            gui.save();  

        } catch (BadLocationException ex) {  

            ex.printStackTrace();  

        }  

    }  

    //Show about window  

    else if (e.getSource()==gui.aboutMenu) {  

        gui.aboutMenu();  

    }  

}  

}//end of actionPerformed method
}

```

A.18 KBSManager.java

```

50
10
20
/**/  

 * Class: KBSManager  

 * Author: Tangming Yuan  

 * Modification: Kristj"sn Avarsson  

**/  

import java.util.*;  

//  

public class KBSManager {  

    static BooleanRuleBase computerKBS;  

//class method of getting the computer KBS  

    public static BooleanRuleBase getCompKBS() {  

        BooleanRuleBase current=null;  

        computerKBS = new BooleanRuleBase("Computer");  

        initKBS(computerKBS);  

        current=computerKBS;  

        return current;  

}  

}//end of get KBS METHOD

```

```

    /*
    //To initialise the CP not acceptable KBS
    public static void initKBS(BooleanRuleBase rb) {
        //Opinions
        Proposition p1=new Proposition(rb,"CP is acceptable",new Boolean(true),"opinion");
        Proposition p2=new Proposition(rb,"CP is acceptable",new Boolean(false),"opinion");
        Proposition p3=new Proposition(rb,"CP is a good deterrent",new Boolean(true),"opinion");
        Proposition p4=new Proposition(rb,"CP is a good deterrent",new Boolean(false),"opinion");
        //Fact
        Proposition p5=new Proposition(rb,"CP having countries like USA has higher murder rate than UK",new Boolean(true));
        //Evidence
        Proposition p6=new Proposition(rb,"statistics shows an increase in murder rate since the abolition of CP");
        //Opinions
        Proposition p7=new Proposition(rb,"CP makes people less likely commit serious crimes",new Boolean(true),"opinion");
        Proposition p8=new Proposition(rb,"nobody is willing to die",new Boolean(true),"opinion");
        Proposition p9=new Proposition(rb,"nobody is willing to die",new Boolean(false),"opinion");
        //Fact
        Proposition p10=new Proposition(rb,"suicide bombers want to die",new Boolean(true),"fact");
        //Opinions
        Proposition p11=new Proposition(rb,"innocent people may get killed",new Boolean(true),"opinion");
        Proposition p12=new Proposition(rb,"innocent people may get killed",new Boolean(false),"opinion");
        Proposition p13=new Proposition(rb,"there are mistakes during judicial process",new Boolean(true),"opinion");
        Proposition p14=new Proposition(rb,"political and racial bias often causes prejudices",new Boolean(true),"opinion");
        Proposition p15=new Proposition(rb,"scientific techniques will increase the success of justice",new Boolean(true));
        Proposition p16=new Proposition(rb,"most people want CP back",new Boolean(true),"opinion");
        //Evidence
        Proposition p17=new Proposition(rb,"the recent survey shows that 60% British people support CP");
        //Opinions
        Proposition p18=new Proposition(rb,"it is wrong to take a human life",new Boolean(true),"opinion");
        Proposition p19=new Proposition(rb,"it is wrong to take a human life",new Boolean(false),"opinion");
        Proposition p20=new Proposition(rb,"a state has the right to deliberately execute the wrong people",new Boolean(true));
        Proposition p21=new Proposition(rb,"human lives are scarce",new Boolean(true),"opinion");
        //Opinions
        Proposition p22=new Proposition(rb,"murderers should receive capital punishment",new Boolean(true),"opinion");
        Proposition p23=new Proposition(rb,"execution of murderers is fair for the people being murdered",new Boolean(true));
        //Fact
        Proposition p24=new Proposition(rb,"CP does not give murderers opportunities for rehabilitation");
        //Evidence
        Proposition p25=new Proposition(rb,"statistics shows that repeat offences of murders are extremely low");
        //Opinions
        Proposition p26=new Proposition(rb,"there are mistakes during judicial process",new Boolean(false),"opinion");
    }
}

```

```

//Fact
Proposition p27=new Proposition(rb,"CP definitely stops murderers commit further crimes",new Boolean(true),
M
//Opinions
Proposition p29=new Proposition(rb,"murderers should receive capital punishment",new Boolean(false),"opinion"
Proposition p30=new Proposition(rb,"chances should be given to murderers for rehabilitation",new Boolean(true),
M
// Define Consequence links
Rule r1 = new Rule(rb, "R1", p3, p1);
Rule r2 = new Rule(rb, "R2", p5,p4);
Rule r3 = new Rule(rb, "R3", p6,p3);
Rule r4 = new Rule(rb, "R4", p7,p3);
Rule r5 = new Rule(rb, "R5", p8,p7);
Rule r6 = new Rule(rb, "R6", p10,p9);
Rule r7 = new Rule(rb, "R7", p11,p2);
Rule r8 = new Rule(rb, "R8", p26,p12);
Rule r9 = new Rule(rb, "R9", p15,p26);
M
Rule r10 = new Rule(rb, "R10", p13,p11);
Rule r11 = new Rule(rb, "R11", p14,p13);
Rule r12 = new Rule(rb, "R12", p16,p1);
Rule r13 = new Rule(rb, "R13", p17,p16);
Rule r14 = new Rule(rb, "R14", p18,p2);
Rule r15 = new Rule(rb, "R15", p21,p18);
Rule r16 = new Rule(rb, "R16", p20,p19);
Rule r17 = new Rule(rb, "R17", p22,p1);
Rule r18 = new Rule(rb, "R18", p23,p22);
Rule r19 = new Rule(rb, "R19", p24,p2);
M
Rule r20 = new Rule(rb, "R20", p27,p1);
M
Rule r22 = new Rule(rb, "R22", p30,p29);
Rule r23 = new Rule(rb, "R23", p25,p30);
}
}//end
M
//to initialise CP acceptable KBS
public static void initProponentKBS(BooleanRuleBase rb) {}
}
//end of class

```

A.19 MessageDialogue.java

```

/***
 * Class: MessageDialogue
 * Author: Tangming Yuan
 * Modification: Kristj"sn Avarsson
 */
// File: MessageDialog.java
// Tommy : May 2001
// Client side program
// Access oracle database in LMU
// Dialog for putting message to user
M

```

10

```

import java.awt.*;/*
import javax.swing.*;
import java.awt.event.*;

public class MessageDialogue extends JDialog implements ActionListener {
    //Component object
    JButton okBut;/*
    JLabel messageLab,blank1,blank2;/*
    Font messFont=new Font("SansSerif",Font.PLAIN,14);/*
    /*
    //Constructor, parent JDialog
    public MessageDialogue(JDialog parent,String message) {
        super(parent, "Message", message :"",true); //model
        initLayout(message);/*
    }/*
    /*
    //Constructor, parent JFrame
    public MessageDialogue(JFrame parent,String message) {
        super(parent, "Message", message :"",true); //model
        initLayout(message);/*
    }/*
    /*
    //Method layout the dialog box
    public void initLayout(String message) {
        //Layout message label
        messageLab=new JLabel(message);/*
        messageLab.setFont(messFont);/*
        getContentPane().add(messageLab,BorderLayout.NORTH);/*
        /*
        //Leave left hand of ok button blank
        blank1=new JLabel("");/*
        getContentPane().add(blank1,BorderLayout.WEST);/*
        /*
        //Leave right hand of ok button blank
        blank2=new JLabel("");/*
        getContentPane().add(blank2,BorderLayout.EAST);/*
        /*
        //Layout ok button
        okBut=new JButton("Ok");/*
        okBut.addActionListener(this);
        getContentPane().add(okBut,BorderLayout.CENTER);/*
        /*
        //Calcualte, location
        pack();/*
        setLocation(200,250);/*
    }/*
    /*
    public void actionPerformed(ActionEvent e) {
        setVisible(false);/*
    }/*
}

```

20

30

40

50

60

A.20 Move.java

```

/***
 * Class: Move
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import javax.swing.text.BadLocationException;
import javax.swing.text.StyledDocument;
class Move {
    String moveType;
    RuleProp rpmc; //move content is a rule
    char turn; //computer or student
    ConflictSet conflict=null; //Constructor when move content is a Rule or proposition
    public Move(char turn, String mt, RuleProp mc) {
        this.turn=turn;
        moveType=mt;
        rpmc=mc;
    }
    //Constructor when move content is a Rule or proposition
    public Move(char turn, String mt, RuleProp mc, ConflictSet conflict) {
        this.turn=turn;
        moveType=mt;
        rpmc=mc;
        this.conflict=conflict;
    }
    //method set turn
    public void setTurn(char s) {
        turn=s;
    }
    //method get turn
    public char getTurn() {
        return turn;
    }
    //Method of set move type
    public void setType(String s) {
        moveType=s;
    }
    //method get move type
    public String getType() {
        return moveType;
    }
    //method get move content proposition
    public Proposition getMoveProp() {
        10
        20
        30
        40
        50
    }
}

```

```

if(rpmc.getClass().getName() == "Proposition") {
    return (Proposition)rpmc;
} else return null;
}

//method get move content if it is a rule
public Rule getMoveRule() {
    if(rpmc.getClass().getName() == "Rule") {
        return (Rule)rpmc;
    } else return null;
}

//clone a move
public Object clone() {
    return new Move(getTurn(),getType(),rpmc,conflict);
}

//display the move on the interface
public void display(int line, StyledDocument ta, String prevType, String prevContent) throws BadLocationException {
    String con = getWholeMove(prevType, prevContent);
    String turnString = "";
    if(turn == 'S') {turnString = "Human";} else if(turn == 'C') {turnString = "Computer";} else {turnString = "Referee";}
    int begin = ta.getLength();
    String inserter = line + ":" + turnString + ">" + con + "\n";
    if (line < 10) ta.insertString(ta.getLength(), "0" + line + ":" + turnString + ">" + con + "\n", ta.getStyle("regular"));
    else ta.insertString(ta.getLength(), inserter, ta.getStyle("regular"));
    //Change color of ID
    if(turn == 'S') {
        ta.setCharacterAttributes(begin + 4, 5, ta.getStyle("Blue"), true);
    } else if(turn == 'C') {
        ta.setCharacterAttributes(begin + 4, 8, ta.getStyle("Red"), true);
    }
}

public ConflictSet getConflictSet() {
    return conflict;
}

//Method get move content as an proposition or rule(object)
public RuleProp getObject() {
    return rpmc;
}

//get rule or proposition name
public String getName() {
    return rpmc.getClass().getName();
}

//Method get move content as a string
public String getContent() {
    return rpmc.getContent();
}

```

60 80 90 100

```

    /**
     * Method of comparation of two moves
     */
    public boolean equals(Move m) {
        boolean equal=false;
        if (getTurn()==m.getTurn()&&getType()==m.getType()&&getContent()==m.getContent())
            equal=true;
        return equal;
    }

    /**
     * Method of deny the move, e.g. no
     */
    public void negate() {
        if(getMoveRule()==null){getMoveProp().negate();} else{getMoveRule().negate();}
    }

    /**
     * Method get whole move
     */
    public String getWholeMove(String prevType,String prevContent) {
        String wholeMove=null;
        //move type is concession
        if(getType()=="Concession") {
            //yes no question
            if(prevType=="Question") {
                if(getContent().equals(prevContent)) {
                    wholeMove="Yes, "+ "I think "+getContent()+".";
                } else{wholeMove="No, "+ "I think "+getContent()+".";}
            }
            //end of concession
        }
        //move type is assertion
        else if(getType()=="Assertion") {
            if(getName()!="Proposition" && getMoveProp().isEvidence()) {
                wholeMove="But "+getContent()+".";
            } else wholeMove="I think "+getContent()+".";
        }
        //end of assertion
        //move type is assertion
        else if(getType()=="Ground") {
            wholeMove="Because "+getContent()+".";
        }
        //end of assertion
        //Challenge
        else if(getType()=="Challenge") {
            wholeMove="Why is it the case that "+getContent()+"?";
        }
        //end of assertion
        //Question
        else if(getType()=="Question") {
            wholeMove="Is it the case that "+getContent()+"?";
        }
    }

```

110 120 130 140 150

```

//Resolve
else if(getType() == "Resolve") {
    if(conflict.size() == 2 && conflict.isPNP() == false && turn == 'C') wholeMove = "You already know this, please resolve";
    else wholeMove = "Please resolve " + getContent() + " in your positions.";//"Do you agree that " + getContent();
}

//Withdraw
else if(getType() == "Withdraw") {
    if(prevType == "Question") wholeMove = "I am not sure about it.";
    else if(prevType == "Challenge") wholeMove = "I don't know why " + getContent() + ".";
    else wholeMove = "I don't think " + getContent() + ".";
}

//Message or ending
else {
    wholeMove = getContent();
}

return wholeMove;
}//end of get whole move

}//end class

```

170

A.21 MoveChoiceInitialiser.java

```

/***
 * Class: MoveChoiceInitialiser
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import java.util.*;
public class MoveChoiceInitialiser {

    static Vector moveContentVector = new Vector();
    static Vector moveTypeVector = new Vector();
    static Vector mixer = new Vector();

    public static void initChoice(GUI gui) {
        initialiseMoveType(gui);
        initialiseMoveContent(gui);
    }

    public static Vector getMoveContent() {
        return moveContentVector;
    }

    public static Vector getMoveTypeContent() {
        return moveTypeVector;
    }

    //Method of show move type
    public static void initialiseMoveType(GUI gui) {

```

10

20

```

moveTypeVector.add("I think. . .");  

moveTypeVector.add("Why.. ? ");  

moveTypeVector.add("Is it the case that.. ?");  

moveTypeVector.add("I don't know whether. . .");  

gui.moveTypeChoice.setListData(moveTypeVector);  

}//end method  

//Method of setting of move content  

public static void initialiseMoveContent(GUI gui){  

    moveContentVector.add(new Proposition("chances should be given to murderers for rehabilitation",new Boolean(true)));  

    moveContentVector.add(new Proposition("chances should be given to murderers for rehabilitation",new Boolean(false)));  

    moveContentVector.add(new Proposition("CP is acceptable",new Boolean(true)));  

    moveContentVector.add(new Proposition("CP is acceptable",new Boolean(false)));  

    moveContentVector.add(new Proposition("CP is a good deterrent",new Boolean(true)));  

    moveContentVector.add(new Proposition("CP is a good deterrent",new Boolean(false)));  

    moveContentVector.add(new Proposition("execution of murderers is fair for the people being murdered",new Boolean(true)));  

    moveContentVector.add(new Proposition("execution of murderers is fair for the people being murdered",new Boolean(false)));  

    moveContentVector.add(new Proposition("innocent people may get killed",new Boolean(true)));  

    moveContentVector.add(new Proposition("innocent people may get killed",new Boolean(false)));  

    moveContentVector.add(new Proposition("it is wrong to take a human life",new Boolean(true)));  

    moveContentVector.add(new Proposition("it is wrong to take a human life",new Boolean(false)));  

    moveContentVector.add(new Proposition("human lives are scarce",new Boolean(true)));  

    moveContentVector.add(new Proposition("human lives are scarce",new Boolean(false)));  

    moveContentVector.add(new Proposition("most people want CP back",new Boolean(true)));  

    moveContentVector.add(new Proposition("most people want CP back",new Boolean(false)));  

    moveContentVector.add(new Proposition("murderers should receive capital punishment",new Boolean(true)));  

    moveContentVector.add(new Proposition("murderers should receive capital punishment",new Boolean(false)));  

    moveContentVector.add(new Proposition("a state has the right to deliberately execute the wrong people",new Boolean(true)));  

    moveContentVector.add(new Proposition("a state has the right to deliberately execute the wrong people",new Boolean(false)));  

    moveContentVector.add(new Proposition("nobody is willing to die",new Boolean(true)));  

    moveContentVector.add(new Proposition("nobody is willing to die",new Boolean(false)));  

    moveContentVector.add(new Proposition("scientific techniques will increase the success of justice",new Boolean(true)));  

    moveContentVector.add(new Proposition("scientific techniques will increase the success of justice",new Boolean(false)));  

    moveContentVector.add(new Proposition("CP makes people less likely commit serious crimes",new Boolean(true)));  

    moveContentVector.add(new Proposition("CP makes people less likely commit serious crimes",new Boolean(false)));

```

```

80
    moveContentVector.add(new Proposition("political and racial bias often causes prejudices",new Boolean(true));
    moveContentVector.add(new Proposition("political and racial bias often causes prejudices",new Boolean(false));
    100
    moveContentVector.add(new Proposition("there are mistakes during judicial process",new Boolean(true)));
    moveContentVector.add(new Proposition("there are mistakes during judicial process",new Boolean(false)));
    110
    //Fact
    moveContentVector.add(new Proposition("CP having countries like USA has higher murder rate than UK",new Boolean(true));
    moveContentVector.add(new Proposition("CP does not give murderers opportunities for rehabilitation",new Boolean(true));
    moveContentVector.add(new Proposition("CP definitely stops murderers commit further crimes",new Boolean(true));
    moveContentVector.add(new Proposition("suicide bombers want to die",new Boolean(true)));
    //Evidence
    moveContentVector.add(new Proposition("statistics shows that repeat offences of murders are extremely low"));
    moveContentVector.add(new Proposition("statistics shows an increase in murder rate since the abolition of"));
    moveContentVector.add(new Proposition("the recent survey shows that 60% British people support CP",new Boolean(true));
    //gui.inputChoice.setListData(moveContentVector);
    Enumeration enum =moveContentVector.elements();
    while (enum.hasMoreElements()){
        Proposition p= (Proposition) enum.nextElement();
        String s= p.getContent();
        mixer.add(s);
    }
    gui.inputChoice.setListData(mixer);
    //end of set move content
    static public Vector getMixer(){
        return mixer;
    }
}

}

```

A.22 Plan.java

```

10
/***
 * Class: Plan
 * Author: Tangming Yuan
 * Modification: Kristj"sn Avarsson
 */
import java.util.*;
public class Plan extends Vector {
    //Find whether a proposition is on the plan
    public boolean contains(Proposition p) {
        boolean contain=false;
        Enumeration enum = elements();

```

```

 $\hat{M}$ 
while (enum.hasMoreElements()) { $\hat{M}$ 
    Proposition test = (Proposition) enum.nextElement(); $\hat{M}$ 
 $\hat{M}$ 
    if(test.equals(p))contain=true; $\hat{M}$ 
} $\hat{M}$ 
return contain; $\hat{M}$ 
} $\hat{M}$ 
 $\hat{M}$ 
//Find the position of proposition on the plan $\hat{M}$ 
public int indexOf(Proposition p) { $\hat{M}$ 
    int position=-1; $\hat{M}$ 
    int size=size(); $\hat{M}$ 
    for(int i=0;i<size;i++) { $\hat{M}$ 
        Proposition temp=(Proposition)elementAt(i); $\hat{M}$ 
        if(temp.equals(p))position=i; $\hat{M}$ 
    } $\hat{M}$ 
    return position; $\hat{M}$ 
} $\hat{M}$ 
//end of index $\hat{M}$ 
 $\hat{M}$ 
//This method is for start a plan $\hat{M}$ 
public Move start() { $\hat{M}$ 
    Move response=null; $\hat{M}$ 
    reform(); $\hat{M}$ 
    RuleProp rp=(RuleProp)elementAt(0); $\hat{M}$ 
    response=new Move('C',"Question",rp); $\hat{M}$ 
    return response; $\hat{M}$ 
} $\hat{M}$ 
//end of plan start $\hat{M}$ 
 $\hat{M}$ 
//Start an immediate plan $\hat{M}$ 
public Move startImmediatePlan() { $\hat{M}$ 
    Move response=null; $\hat{M}$ 
    RuleProp rp=(RuleProp)elementAt(0); $\hat{M}$ 
    response=new Move('C',"Question",rp); $\hat{M}$ 
    return response; $\hat{M}$ 
} $\hat{M}$ 
//end of plan start $\hat{M}$ 
 $\hat{M}$ 
/* $\hat{M}$ 
 * Method of plan execution $\hat{M}$ 
 */ $\hat{M}$ 
public Move execute(Vector dilv, Planner planner, CommitmentStore partnerStore, CommitmentStore selfStore, Boolean
    Move response=null; $\hat{M}$ 
    Move previous=(Move)dilv.lastElement(); $\hat{M}$ 
    RuleProp prevrp=previous.getObject(); $\hat{M}$ 
    Move secondLast=(Move)dilv.elementAt(dilv.size()-2); $\hat{M}$ 
 $\hat{M}$ 
    planner.currentPlan.show(); $\hat{M}$ 
 $\hat{M}$ 
/* $\hat{M}$ 
 1. If the user say yes, go on to execute plan, the plan executed ok. $\hat{M}$ 
   If there is only one item remaining, then state the item and set the plan to null $\hat{M}$ 
   If the plan does not reach the end, then continuously pose the question $\hat{M}$ 
 */ $\hat{M}$ 

```

```

if(previous.getType() == "Concession" && previous.getContent().equals(secondLast.getContent()) // && size() > 0) {
    // plan success and get rid of the first element
    removeElementAt(0);
    if(size() > 1) {
        RuleProp rp = (RuleProp)elementAt(0);
        response = new Move('C', "Question", rp);
    } else if(size() == 1) {
        RuleProp rp = (RuleProp)elementAt(0);
        response = new Move('C', "Assertion", rp);
        planner.currentPlan = null;
    }
}
/* */
2. The user say withdrawal after a question, temporarily dropped according to Moore(1993 chapter 9 pp26)
current plan is set to null. The line has been dropped. However, before the dropping,
C may challenge the negation of the statement being build, if it is challengable
*/
else if(previous.getType() == "Withdraw" && secondLast.getType() == "Question") {
    Proposition last = (Proposition)elementAt(size() - 1);
    /* */
    C may needs to challenge it if the current focus is not current being built
    /*
    planner.currentPlan = null;
}
/* */
3. The user gives an unwanted statement,
If there is a direct conflict, then pose a resolution demand
Or challenge the unwanted proposition
*/
else if(previous.getType() == "Concession" && previous.getContent().equals(secondLast.getContent()) == false) {
    /*
    3.1 If there is an direct conflict, then pose the resolution demand
    */
    ConflictSet cs = partnerStore.getRealConflictSet();
    if(cs.isEmpty() == false && cs.onConflict(prevrp)) {
        Proposition p = cs.mergeToProp();
        response = new Move('C', "Resolve", p, cs);
    }
    // end conflict
}
/* */
3.2 If without direct conflict, then check the unwanted statement is challengable
If it is, then challenge it
*/
else if (selfKBS.challengable(prevrp, partnerStore, selfStore)) {
    response = new Move('C', "Challenge", prevrp);
}
}

```

```

 $\hat{M}$ 
/* $\hat{M}$ 
3.3 If without direct conflict, and the statement is not challengable $\hat{M}$ 
then abandon the current line of question. $\hat{M}$ 
*/ $\hat{M}$ 
else { $\hat{M}$ 
    planner.currentPlan=null; $\hat{M}$ 
} $\hat{M}$ 
}//end of unwanted statement $\hat{M}$ 
 $\hat{M}$ 
/* $\hat{M}$ 
4. The user says a withdrawal after a resolution, need to check whether $\hat{M}$ 
the appropriate answer is withdrawn $\hat{M}$ 
*/ $\hat{M}$ 
else if(previous.getType() == "Withdraw" && secondLast.getType() == "Resolve") { $\hat{M}$ 
     $\hat{M}$ 
    RuleProp rp=(RuleProp)elementAt(0); $\hat{M}$ 
     $\hat{M}$ 
    //Wanted withdraw $\hat{M}$ 
    if((rp.getClass().getName() == "Proposition" && prevrp.getClass().getName() == "Proposition" && ((Proposition)rp)
        response=new Move('C',"Question",rp); $\hat{M}$ 
    } $\hat{M}$ 
     $\hat{M}$ 
    else if((rp.getClass().getName() == "Rule" && prevrp.getClass().getName() == "Rule" && ((Rule)rp).denial().equals(((Rule)prevrp).denial())) {
        response=new Move('C',"Question",rp); $\hat{M}$ 
    } $\hat{M}$ 
     $\hat{M}$ 
    //Unwanted withdraw $\hat{M}$ 
    else { $\hat{M}$ 
        //Drop the line of question, $\hat{M}$ 
        planner.currentPlan=null; $\hat{M}$ 
         $\hat{M}$ 
    } $\hat{M}$ 
}//end of withdraw after a resolution demand $\hat{M}$ 
 $\hat{M}$ 
/* $\hat{M}$ 
5. The user says a withdrawal after a resolution, need to check whether $\hat{M}$ 
the appropriate answer is withdrawn $\hat{M}$ 
*/ $\hat{M}$ 
else if(previous.getType() == "Withdraw" && secondLast.getType() == "Challenge") { $\hat{M}$ 
    RuleProp rp=(RuleProp)elementAt(0); $\hat{M}$ 
    response=new Move('C',"Question",rp); $\hat{M}$ 
} else if(previous.getType() == "Ground" && secondLast.getType() == "Challenge") { $\hat{M}$ 
    //Drop the line of question, $\hat{M}$ 
    planner.currentPlan=null; $\hat{M}$ 
     $\hat{M}$ 
}//end of withdraw after a resolution demand $\hat{M}$ 
return response; $\hat{M}$ 
} $\hat{M}$ 
/* For a bottom up plan, may need to be reformed $\hat{M}$ 

```

```

* in order to be ready for questioning
*/
public void reform() {
    Plan temp=(Plan)clone();
    this.removeAllElements();
    this.add(temp.elementAt(0));
    for (int i=0;i<(temp.size()-1);i++) {
        Proposition ap=null;Proposition cp=null;
        try{
            ap=(Proposition)temp.elementAt(i);
            cp=(Proposition)temp.elementAt(i+1);
        } catch(ClassCastException ce){}
        if(ap!=null&&cp!=null) {
            Rule r=new Rule(ap,cp);
            this.add(r);
        }
    }
    this.add((Proposition)temp.elementAt(temp.size()-1));
}
//Clone a plan
public Object clone() {
    Plan temp=new Plan();
    for (int i=0;i<size();i++) {
        temp.add((RuleProp)elementAt(i));
    }
    return temp;
}
//Check whether a statement is on current plan
public boolean onPlan(RuleProp rp) {
    boolean on=false;
    Enumeration enum=elements();
    while(enum.hasMoreElements()) {
        RuleProp test = (RuleProp) enum.nextElement();
        if(test.getContent()==rp.getContent())on=true;
    }
    return on;
}
//Show the elements in the plan
public void show() {
    Enumeration enum1 = elements();
    while (enum1.hasMoreElements()) {
        RuleProp temp1 = (RuleProp) enum1.nextElement();
    }
}

```

180

190

200

210

220

```

//test
public static void main(String arg[])
{
    Plan p1=new Plan();
    p1.add(new Proposition("1", new Boolean(true)));
    p1.add(new Proposition("2", new Boolean(true)));
    p1.add(new Proposition("3", new Boolean(true)));
    Enumeration enum = p1.elements();
    while (enum.hasMoreElements())
    {
        Proposition test = (Proposition) enum.nextElement();
    }
    Plan temp=(Plan)p1.clone();
    Enumeration enum1 = temp.elements();
    while (enum1.hasMoreElements())
    {
        Proposition test = (Proposition) enum1.nextElement();
    }
    p1.reform();
    Enumeration enum2 = p1.elements();
    while (enum2.hasMoreElements())
    {
        RuleProp test = (RuleProp) enum2.nextElement();
    }
}
//end test
}

```

230

240

250

10

A.23 PlanSet.java

```

/***
 * Class: PlanSet
 * Author: Tangming Yuan
 * Modification: Kristján Þórvártsson
 */
import java.util.*;
public class PlanSet extends Vector
{
    /*
     * Since the plan set will get all the possible plan,
     * this method will produce a hard evidential move
     */
    public Move getHardEvidentialMove(CommitmentStore selfStore)
    {
        Move evidentialMove=null;
        Enumeration enum =elements();
        while (enum.hasMoreElements())
        {
            Plan temp = (Plan) enum.nextElement();
    }
}

```

```

if(temp.size()==2) {  

    Proposition p=(Proposition)temp.elementAt(0);  

    if(p.isEvidence()&&selfStore.onClaimStack(p)==false) {  

        evidentialMove=new Move('C',"Assertion", p);  

        break;  

    }  

}  

}//end while  

return evidentialMove;  

}//end of get hard evidence  

/*  

Since the plan set will get all the possible plan, if the support cannot  

be further supported, this method will get a question plan, two elements only  

*/  

public Move getQuestionMove(DialogueHistory dilv, Planner planner) {  

    Move questionMove=null;  

    Enumeration enum =elements();  

    while (enum.hasMoreElements()) {  

        Plan temp = (Plan) enum.nextElement();  

        Proposition p=(Proposition)temp.elementAt(0);  

        if(temp.size()==2)//&&p.isEvidence()==false)  

        {  

            //Make sure the plan has not been used  

            if(dilv.contains(new Move('C',"Question",p))==false) {  

                planner.currentPlan=temp;  

                questionMove=planner.currentPlan.start();  

                break;  

            }  

        }  

    }  

}//end while  

return questionMove;  

}//end of get hard evidence  

/*  

Since the plan set will get all the possible plan, this method will produce a rebuttal move, state the support which can be further supported  

*/  

public Move getRebuttalMove(CommitmentStore selfStore, CommitmentStore partnerStore) {  

    Move rebuttalMove=null;  

    Enumeration enum =elements();  

    while (enum.hasMoreElements()) {  

        Plan temp = (Plan) enum.nextElement();  

        if(temp.size()>2) {  

            Proposition last=(Proposition)temp.elementAt(temp.size()-1);  

            Proposition secondlast=(Proposition)temp.elementAt(temp.size()-2);  

            20  

            30  

            40  

            50  

            60  

            70
        }
    }
}

```

```

    }
    if(selfStore.onClaimStack(last)===false&&(selfStore.onTotal(last)&&partnerStore.onTotal(last))==false) {
        rebutalMove=new Move('C',"Assertion", last);
        break;
    } else if(selfStore.onClaimStack(secondlast)===false&&(selfStore.onTotal(secondlast)&&partnerStore.onTotal(secondlast))==false) {
        rebutalMove=new Move('C',"Assertion", secondlast);
        break;
    }
}
}/*end while*/
return rebutalMove;
}//end of get hard evidence
/*
If its support can be further supported
This method is produced to get the subtopic under discussion
It is used for Computer's build strategy
*/
public Move startSubtopic(CommitmentStore selfStore, CommitmentStore partnerStore) {
    Move m=null;
    Vector subtopic=new Vector();
    Enumeration enum =elements();
    while (enum.hasMoreElements()) {
        Plan temp = (Plan) enum.nextElement();
        if(temp.size()>2) {
            Proposition secondlast=(Proposition)temp.elementAt(temp.size()-2);
            if(selfStore.onClaimStack(secondlast)===false&&(selfStore.onTotal(secondlast)&&partnerStore.onTotal(secondlast))==false) {
                subtopic.add(secondlast);
            }
        }
    }
}
}/*end while*/
/*
Random basis
*/
if(subtopic.isEmpty()===false) {
    int i= (int) ( Math.random() * subtopic.size() );
    m=new Move('C',"Assertion", (Proposition)subtopic.elementAt(i));
}
return m;
}//end of get hard evidence
/*
If it can be further supported, but not by a hard evidence
C can build a plan
*/
public Move startBuildPlan(DialogueHistory dilv, Planner planner) {
    Move questionMove=null;
    Vector questionPlan=new Vector();
    Enumeration enum =elements();
    while (enum.hasMoreElements()) {

```

80

90

110

120

```

Plan temp = (Plan) enum.nextElement();  

Proposition p=(Proposition)temp.elementAt(0);  

if(temp.size()>2)//&&p.isEvidence()==false)  

{  

    //Make sure the plan has not been used  

    if(dilv.contains(new Move('C',"Question",p))==false) {  

        questionPlan.add(temp);  

    }  

}  

}//end while  

/*  

Randomly choose one  

*/  

if(questionPlan.isEmpty()==false) {  

    int i= (int) ( Math.random() * questionPlan.size());  

    planner.currentPlan=(Plan)questionPlan.elementAt(i);  

    questionMove=planner.currentPlan.start();  

}  

return questionMove;  

}//end of get build strategy  

//May need some work to pick up a suitable plan  

public Plan pickAplan() {  

    Plan currentPlan=null;  

    currentPlan=(Plan)elementAt(0);  

    removeElementAt(0);  

    return currentPlan;  

}  

public void show() {  

    Enumeration enum =elements();  

    while (enum.hasMoreElements()) {  

        Plan temp = (Plan) enum.nextElement();  

        Enumeration enum1 = temp.elements();  

        while (enum1.hasMoreElements()) {  

            RuleProp temp1 = (RuleProp) enum1.nextElement();  

        }  

    }  

}//end show
}

```

130 140 150 160 170

A.24 Planner.java

```

/**  

* Class: Planner  

* Author: Tangming Yuan

```

```

        * Modification: Kristj"sn Evarsson*
**/Â
import java.util.*;Â
Â
public class Planner {Â
    char turn;Â
    Â
    /*Â
     * Current plan, supposed to be a line of questionsÂ
     */
    Plan currentPlan=null;Â
    Â
    /*Â
     * Computer's thesisÂ
     */
    Proposition computerThesis=null;Â
    Â
    boolean exhausted=false;Â
    Â
    /*Â
     *Planner constructorÂ
     */
    public Planner(char turn) {Â
        this.turn=turn;Â
        Â
    }Â
    Â
    /*Â
     *Method of set thesisÂ
     */
    public void setThesis(Proposition p) {Â
        computerThesis=p;Â
    }Â
    Â
    // A lot of work to do in this part: Parameter: dialv,both commitment store,kbsÂ
    public Vector produceRelevantMove(DialogueHistory dialv,CommitmentStore selfStore,CommitmentStore partnerStore,Boolean
        Vector rmv=new Vector();Â
        Â
        Move previous=(Move)dilv.lastElement();Â
        String prevType=previous.getType();Â
        RuleProp prevrp=previous.getObject();Â
        Â
        /*Â
         * When facing the situation of ASSERTION, or GROUNDÂ
         */
        if(previous.getType()=="Assertion"||previous.getType()=="Ground"||previous.getType()=='Concession') {Â
            rmv=AssertionStrategist.planAssertion(dilv, selfStore,partnerStore,selfKBS,this);Â
        }Â//end of facing assertionÂ
        Â
        /*Â
         * When facing WITHDRAW a proposition or ruleÂ
         */
        if(previous.getType()=='Withdraw') {Â

```

```

        rmv=WithdrawStrategist.planWithdraw(dilv, selfStore, partnerStore, selfKBS, this);  

    } //end of facing assertion  

    /*  

    III      When facing the CHALLENGE, challenge a rule is not currently available  

    */  

    else if(previous.getType() == "Challenge") {  

        if(previous.getName() == "Proposition") {  

            Proposition prevProp=(Proposition) previous.getMoveProp().clone();  

            rmv=ChallengeStrategist.planChallProp(prevProp, turn, selfStore, partnerStore, selfKBS);  

        }  

        //this feature is not available  

        else {}  

    } //end of facing challenge  

    /*  

    IV      Facing the QUESTION  

    */  

    else if(previous.getType() == "Question") {  

        rmv=QuestionStrategist.planQuestion(dilv, selfStore, partnerStore, selfKBS);  

    } //end of facing question  

    /*  

    V      Facing a legal RESOLUTION demand  

    */  

    else if(previous.getType() == "Resolve") {  

        rmv=planResolution(previous, dilv, turn, selfStore, partnerStore, selfKBS);  

    } //end resolve  

    return rmv;  

} //end of produce move  

//Method of plan when facing with a withdrawn rule  

private Vector planResolution(Move previous, Vector dilv, char currentTurn, CommitmentStore selfStore, CommitmentStore  

    Vector rmv=new Vector();  

    ConflictSet selfConflictSet=previous.getConflictSet();  

    //we deal with P, ?P and R, R->P first  

    if(selfConflictSet.size()==2) {  

        //if the conflict is P, ?P  

        if(selfConflictSet.isPNP()) {  

            Proposition p1=(Proposition) selfConflictSet.elementAt(0);  

            Proposition p2=(Proposition) selfConflictSet.elementAt(1);  

            Proposition favor=null;  

            Proposition against=null;  

            // A bit judge which answer on C's side and which is not  

        }  

    }

```

```

if(selfKBS.support(p1, computerThesis)||selfKBS.againstSupport(p1,computerThesis.denial())) {110
    favor=p1;M
    against=p2;M
} else {M
    favor=p2;M
    against=p1;M
}M
M
if (selfKBS.acceptableGroundProps(favor, partnerStore).isEmpty()==false) {M
    rmv.add(new Move(currentTurn,"Withdraw", against));M
} else rmv.add(new Move(currentTurn,"Withdraw", favor));M
M
}M
}//end P, ?P conflictM
M
/*M
1. Affirm the consequenceM
*/M
else {M
    //We get the previous proposition and make a copyM
    Proposition pc=(Proposition)selfConflictSet.getConseq().clone();M
    rmv.add(new Move(currentTurn,"Assertion",pc));M
    RuleProp p1=(RuleProp)selfConflictSet.elementAt(0);M
    RuleProp p2=(RuleProp)selfConflictSet.elementAt(1);M
    rmv.add(new Move(currentTurn,"Withdraw", p1));M
    rmv.add(new Move(currentTurn,"Withdraw", p2));M
}M
M
}//end of 2 size conflictM
M
//The conflict is as such P,R, R->?PM
/*M
1      Decide winning orM
    Withdraw which has no supportM
*/M
else {M
    RuleProp rp1=(RuleProp)selfConflictSet.elementAt(0);M
    RuleProp rp2=(RuleProp)selfConflictSet.elementAt(1);M
    RuleProp rp3=(RuleProp)selfConflictSet.elementAt(2);M
    rmv.add(new Move(currentTurn,"Withdraw", rp1));M
    rmv.add(new Move(currentTurn,"Withdraw", rp2));M
    rmv.add(new Move(currentTurn,"Withdraw", rp3));M
}M
M
return rmv;M
}//end of resolveM
}M
}//end of class relevance managerM

```

A.25 Proposition.java

```

/**M
 * Class: PropositionM

```

```

* Author: Tangming Yuan
* Modification: Kristj"sn Evarsson
*/
import java.util.*;
import java.awt.*;
/*
 */
/***
 * The <code>Proposition</code> class implements the antecedent, warrent and
 * consequent parts of a rule.
 */
* author Tangming Yuan
*/
public class Proposition extends RuleProp {
    BooleanRuleBase rb;
    Vector ruleRefs=new Vector();
    String name; //the name of the proposition
    String content;// the content of the emove
    String position=null; // position in the rule
    Boolean truth; // states = null(unknown), true or false
    /*
     *We add a new attribute to identify whether it is an direct evidence
     * A fact or opinions
     */
    String qualification=null;
    /*
     *Create a standalone proposition for a rule base
     */
    public Proposition(BooleanRuleBase rb, String name, Boolean b) {
        this.rb=rb;
        this.name=name;
        truth = b;
        rb.propositionList.addElement(this);
    }
    /*
     *Create a proposition
     */
    public Proposition(String name, Boolean b) {
        this.name=name;
        truth = b;
    }
    /*
     *Create a standalone proposition for a rule base
     */
    public Proposition(BooleanRuleBase rb, String name, Boolean b, String qualification) {
        this.rb=rb;
        this.name=name;
        truth = b;
        rb.propositionList.addElement(this);
        this.qualification=qualification;
    }
    /*
     *Create a proposition
     */
    public Proposition(String name, Boolean b, String qualification) {
        this.name=name;
        truth = b;
    }

```

```

        this.qualification=qualification;^
    }^
^
//Method of getting qualification^
public String getQualification() {^
    return qualification;^
}^
^
^
//A parsing method for proposition^
public String getContent() {^
    String content=null;^
    if(check().booleanValue()==false) {^
        if(getName()=="a state has the right to deliberately execute the wrong people") {^
            content="a state does not have the right to deliberately execute the wrong people";}^
        ^

        else if(getName()=="chances should be given to murderers for rehabilitation") {^
            content="chances should not be given to murderers for rehabilitation";}^
        ^

        else if(getName()=="CP is acceptable") {^
            content="CP is not acceptable";}^
        ^

        else if(getName()=="CP is a good deterrent") {^
            content="CP is not a good deterrent";}^
        ^

        else if(getName()=="CP makes people less likely commit serious crimes") {^
            content="CP does not make people less likely commit serious crimes";}^
        ^

        //opinion^
        else if(getName()=="CP stops murderers commit further crimes") {^
            content="CP does not stop murderers commit further crimes";}^
        ^

        else if(getName()=="execution of murderers is fair for the people being murdered") {^
            content="execution of murderers is not fair for the people being murdered";}^
        ^

        else if(getName()=="human lives are scarce") {^
            content="human lives are not scarce";}^
        ^

        else if(getName()=="innocent people may get killed") {^
            content="it is unlikely innocent people get killed";}^
        ^

        else if(getName()=="it is wrong to take a human life") {^
            content="it is not always wrong to take a human life";}^
        ^

        else if(getName()=="most people want CP back") {^
            content="most people don't want CP back";}^
        ^

        else if(getName()=="murderers should receive capital punishment") {^
            content="murderers should not receive capital punishment";}^
        ^

        else if(getName()=="nobody is willing to die") {^

```

60

80

90

100

```

        content="some people are willing to die";}Â
    Â
    else if(getName() == "political and racial bias often causes prejudices") {Â
        content="political and racial bias will not cause prejudices";}Â
    Â
    else if(getName() == "scientific techniques will increase the success of justice") {Â
        content="scientific techniques will not increase the success of justice";}Â
    Â
    else if(getName() == "there are mistakes during judicial process") {Â
        content="mistakes rarely happen during judicial process";}Â
    Â
    else if(getName() == "suicide bombers want to die") {Â
        content="suicide bombers don't want to die";}Â
    Â
    else if(getName() == "CP does not give murderers opportunities for rehabilitation") {Â
        content="CP gives murderers opportunities for rehabilitation";}Â
    Â
    else if(getName() == "CP having countries like USA has higher murder rate than UK") {Â
        content="it is not the case that CP having countries like USA has high murder rate than UK";}Â
    Â
    else if(getName() == "statistics shows an increase in murder rate since the abolition of CP") {Â
        content="it is not the case that statistics shows an increase in murder rate since the abolition of CP";}Â
    Â
    else if(getName() == "the recent survey shows that 60% British people support CP") {Â
        content="it is not the case that the recent survey shows that 60% British people support CP";}Â
    Â
    else if(getName() == "statistics shows that repeat offences of murders are extremely low") {Â
        content="it is not the case that statistics shows that repeat offences of murders are extremely low";}Â
    Â
    else content="denial of this is not specified";Â
} else {content=getName();}Â
return content;Â
}
Â
/**Â
 * Assign a truth object to a propositionÂ
 */
public void setTruth(Boolean b) {
    truth=b;
}
Â
/**Â
 * Registers the given rule with this proposition.Â
 */
public void addRuleRef(Rule ref) {
    ruleRefs.addElement(ref);
}
Â
/**Â
 * Set the position of the proposition in the rule.Â
 */

```

```

 $\ast/\hat{M}$ 
public void setPosition(String s) { $\hat{M}$ 
    if (s=="PREMISE" | s=="CONCLUSION" | s=="WARRENT") { $\hat{M}$ 
        position=s; $\hat{M}$ 
    } $\hat{M}$ 
} $\hat{M}$ 
 $\ast/\hat{M}$ 
 $\ast \text{ Check whether it is an evidence} \hat{M}$ 
 $\ast/\hat{M}$ 
public boolean isEvidence() { $\hat{M}$ 
    boolean evidence=false; $\hat{M}$ 
     $\hat{M}$ 
    if(getQualification()=="evidence")evidence=true; $\hat{M}$ 
    return evidence; $\hat{M}$ 
} $\hat{M}$ 
 $\hat{M}$ 
 $\ast/\hat{M}$ 
 $\ast \text{ Check whether it is a fact} \hat{M}$ 
 $\ast/\hat{M}$ 
public boolean isFact() { $\hat{M}$ 
    boolean fact=false; $\hat{M}$ 
     $\hat{M}$ 
    if(getQualification()=="fact")fact=true; $\hat{M}$ 
    return fact; $\hat{M}$ 
} $\hat{M}$ 
 $\hat{M}$ 
 $\ast/\hat{M}$ 
 $\ast \text{ Performs a test of the proposition.} \hat{M}$ 
 $\ast/\hat{M}$ 
public Boolean check() { $\hat{M}$ 
    return truth; $\hat{M}$ 
} $\hat{M}$ 
 $\hat{M}$ 
 $\ast/\hat{M}$ 
 $\ast \text{ Get the name of the proposition.} \hat{M}$ 
 $\ast/\hat{M}$ 
public String getName() { $\hat{M}$ 
    return name; $\hat{M}$ 
} $\hat{M}$ 
 $\hat{M}$ 
 $\ast/\hat{M}$ 
 $\ast \text{Retrieves the rule that owns this proposition} \hat{M}$ 
public Vector getRule() { $\hat{M}$ 
    return ruleRefs; $\hat{M}$ 
} $\hat{M}$ 
 $\hat{M}$ 
 $\ast/\hat{M}$ 
 $\ast \text{Compare two propositions} \hat{M}$ 
 $\ast/\hat{M}$ 
public boolean equals(Proposition p) { $\hat{M}$ 
    if (name==p.getName()&& truth.equals(p.check())) { $\hat{M}$ 
        return true; $\hat{M}$ 
    } else return false; $\hat{M}$ 
} $\hat{M}$ 

```

170

180

190

200

210

```

/*
 * if a proposition is negate with another
 */
public boolean negate(Proposition p) {  

    boolean negate=false;  

    if (name==p.getName()) {  

        if((truth.booleanValue()==true&&p.check().booleanValue()==false)|| (truth.booleanValue()==false&&p.check().bo  

            negate=true;  

    }  

    } //end if  

    return negate;  

} //end check  

//clone a move
public Object clone() {  

    return new Proposition(getName(),check(), getQualification());  

}  

//deny the proposition
public void negate() {  

    if(truth.booleanValue()==true){setTruth(new Boolean(false));} else{setTruth(new Boolean(true));}  

}  

//the denial of a proposition
public Proposition denial() {  

    Boolean deny;  

    if(truth.booleanValue()==true){deny=new Boolean(false);} else{deny=new Boolean(true);}  

    return new Proposition(getName(),deny);  

}  

//Displays this proposition in text format in the given text area.
void display(TextArea textArea) {  

    textArea.append(" "+getContent()+"\n");  

}
}//end of class proposition

```

A.26 QuestionStrategist.java

```

/**  

 * Class: QuestionStrategist  

 * Author: Tangming Yuan  

 * Modification: Kristj"sn Avarsson  

**/br/>
import java.util.*;  

public class QuestionStrategist {  

    //Method of plan when facing with an asserted proposition
    public static Vector planQuestion(DialogueHistory dlyv, CommitmentStore selfStore, CommitmentStore partnerStore, Boolean  

}

```

```

Vector rmv=new Vector();  

Move previous=(Move)dilv.lastElement();  

RuleProp prevrp=previous.getObject();  

/*  

When a proposition is questioned  

*/  

if(dilv.size()==1) {  

    rmv.add(new Move('C',"Concession",planner.computerThesis));  

}//end of game set up  

else if(prevrp.getClass().getName() == "Proposition") {  

    Proposition prevProp=(Proposition)prevrp;  

/*  

1 If both (P, ?P) cannot be found in KBS, then say No Commitment  

*/  

if(selfKBS.checkProp(prevProp)==false&&selfKBS.checkProp(prevProp.denial())==false) {  

    rmv.add(new Move('C',"Withdraw", prevProp));  

}  

/*  

2 If only one of them (P, ?P) can be found in KBS.  

If C has uttered No commitment to the found statement before, then no commitment  

Else, C would assert the found statement  

*/  

else if(selfKBS.checkProp(prevProp)==true&&selfKBS.checkProp(prevProp.denial())==false) {  

    if (dilv.contains(new Move('C',"Withdraw", prevProp))) {  

        rmv.add(new Move('C',"Withdraw", prevProp));  

    }  

    else {  

        rmv.add(new Move('C',"Concession", prevProp));  

    }  

}//end of one statement check  

/*  

else if(selfKBS.checkProp(prevProp)==false&&selfKBS.checkProp(prevProp.denial())==true) {  

    if (dilv.contains(new Move('C',"Withdraw", prevProp.denial())))) {  

        rmv.add(new Move('C',"Withdraw", prevProp.denial()));  

    }  

    else {  

        rmv.add(new Move('C',"Concession", prevProp.denial()));  

    }  

}//end of second statement check  

/*  

3. Both (P, ?P) can be found in the KBS  

*/  

else {  

    Proposition favor=null;  

    Proposition against=null;  

}

```

```

/*M
A bit judge which answer on C's side and which is notM
*/M
if(selfKBS.support(prevProp, planner.computerThesis)||selfKBS.againstSupport(prevProp,planner.computerTh
    favor=prevProp;M
    against=prevProp.denial();M
} else {M
    favor=prevProp.denial();M
    against=prevProp;M
}M
M
/*M
3.1 If C has acceptable support for C's favor statement, then state its favorable oneM
*/M
if(selfKBS.acceptableGroundProps(favor, partnerStore).isEmpty() == false) {M
    rmv.add(new Move('C',"Concession", favor));M
}M
M
/*M
3.2 If C has no acceptable support for C's favor statement,M
and C has got reasonable support in his assertion to support "against" M
C would surrender and state "against" M
M
    OR against is a consequence of its storeM
*/M
else if (selfStore.issupported(against,selfKBS)||selfStore.isConsequence(against)) {M
    rmv.add(new Move('C',"Concession", against));M
}M
M
else {M
    rmv.add(new Move('C',"Withdraw", prevProp));M
}M
}M
//END OF PROPOSITIONM
M
/*M
A rule is challengedM
*/M
else {M
    Rule prevRule=(Rule)prevrp;M
    /*M
        Since current ssystem, a conditional is not challengable,M
        C would assert a conditional if it can be found in KBSM
        Otherwise, C would say no commitment to it.M
    */M
    if(selfKBS.matchRule(prevRule)!=null){rmv.add(new Move('C',"Concession", prevRule));} M
    else {rmv.add(new Move('C',"Withdraw", prevRule));}M
}M
return rmv;M
}//end methodM
}//end class

```

A.27 Rule.java

```

/***
 * Class: Rule
 * Author: Tangming Yuan
 * Modification: Kristj"sn Evarsson
 */
import java.awt.*;
/*
 * The <code>Rule</code> class implements the antecedent, warrent and
 * consequent parts of a rule.
 */
* author Tangming Yuan
*/
public class Rule extends RuleProp {
    BooleanRuleBase rb;
    String name; //name of the rule
    Proposition antecedent; //only 1 consequent proposition allowed
    Proposition consequent; //null=unknown, true, or false)
    Proposition warrent; //cretae a rule for a knowledge base
    boolean fired=false;
    /*
    */
    /*
    */
    public Rule(BooleanRuleBase rb, String name, Proposition lhs, Proposition rhs, Proposition wrt) {
        this.rb = rb;
        this.name = name;
        /*
        */
        antecedent = lhs;
        antecedent.setPosition("ANTECEDENT");
        antecedent.addRuleRef(this); //Register the proposition with this rule
        rb.rulePropList.addElement(antecedent);
        /*
        */
        consequent = rhs;
        consequent.setPosition("CONSEQUENT");
        consequent.addRuleRef(this);
        rb.rulePropList.addElement(consequent);
        /*
        */
        warrent=wrt;
        warrent.setPosition("WARRENT");
        warrent.addRuleRef(this);
        rb.rulePropList.addElement(warrant);
        /*
        */
        //add self to rule list
        rb.ruleList.addElement(this);
        /*
        */
    }
}

```

```

//the truth of the rule
truth = new Boolean(true);  

}  

//create a rule for a knowledge base
public Rule(BooleanRuleBase rb, String name, Proposition lhs, Proposition rhs) {  

    this.rb = rb;  

    this.name = name;  

//premise
antecedent = lhs;  

antecedent.setPosition("ANTECEDENT");  

antecedent.addRuleRef(this);           //Register the proposition with this rule  

rb.rulePropList.addElement(antecedent);  

//consequent
consequent = rhs;  

consequent.setPosition("CONSEQUENT");  

consequent.addRuleRef(this);  

rb.rulePropList.addElement(consequent);  

//add self to rule list
rb.ruleList.addElement(this);  

//the truth of the rule
truth = new Boolean(true);  

}  

//Create a rule with name, and warrent
public Rule(String name, Proposition lhs, Proposition rhs, Proposition wrt) {  

    this.name = name;  

//premise
antecedent = lhs;  

antecedent.setPosition("ANTECEDENT");  

antecedent.addRuleRef(this);           //Register the proposition with this rule  

//consequent
consequent = rhs;  

consequent.setPosition("CONSEQUENT");  

consequent.addRuleRef(this);  

//warrent
warrent=wrt;  

warrent.setPosition("WARRENT");  

warrent.addRuleRef(this);  

//the truth of the rule
truth = new Boolean(true);  

}  

//Create a proposition with only premise and consequent
public Rule(Proposition lhs, Proposition rhs) {  


```

```

//premise
antecedent = lhs;//
antecedent.setPosition("ANTECEDENT");//
antecedent.addRuleRef(this); //Register the proposition with this rule
//
//consequent
consequent = rhs;//
consequent.setPosition("CONSEQUENT");//
consequent.addRuleRef(this);//
//
//the truth of the rule
truth = new Boolean(true);//
}
//
//clone a move
public Object clone() {
    return new Rule(getAntecedent(),getConsequent());
}
//
/***
 * Retrieves the antecedent propositions in this rule.
 */
* return the of antecedents
*/
public Proposition getAntecedent() {
    return antecedent;
}
//
/***
 * Retrieves the warrent propositions in this rule.
 */
* return the warrent of antecedents
*/
public Proposition getWarrent() {
    return warrent;
}
//
//set warrent
public void setWarrent(Proposition p) {
    warrent=p;
}
//
/* Retrieves the consequent propositions in this rule.
 */
* return the of consequent
*/
public Proposition getConsequent() {
    return consequent;
}
//
/***
 * Retrieves the truth of this rule.
 */

```

110

120

130

140

150

```

    *M
    * return the truthM
*/M
public Boolean check() {M
    return truth;M
}M
M
/**M
 * Assign a truth object to a propositionM
*M
*/M
public void setTruth(Boolean b) {M
    truth=b;M
}M
M
//negate the truth valueM
public void negate() {M
    if(truth.booleanValue()==true){setTruth(new Boolean(false));} else{setTruth(new Boolean(true));}M
}M
M
//denial a ruleM
public Rule denial() {M
    Rule denial=new Rule(antecedent,consequent);M
    denial.negate();M
    return denial;M
}M
M
M
/**M
 * Get the string format content of the ruleM
*/M
public String getContent() {M
    String content=null;M
    if(truth.booleanValue()==true)content=","+antecedent.getContent()+" implies "+consequent.getContent()+",";M
    else content="it is not the case that "+antecedent.getContent()+" implies "+consequent.getContent()+",";M
    return content;M
}M
M
/**Compare two rules,M
 */
public boolean equals(Rule r) {M
    if ((getAntecedent().equals(r.getAntecedent()))&&(getConsequent().equals(getConsequent()))&& truth.equals(r.check()))M
        return true;M
    } else return false;M
}M
/**M
 * Displays this rule in text format in the given text area.M
*M
 * param textArea the JTextArea where the rule is displayedM
*/M
public void display(TextArea textArea) {M
    textArea.append(name + ": ");M
    textArea.append(antecedent.getContent());M
}M

```

160 170 180 190 200 210

```

        textArea.append("\n      implies\n    );  

        textArea.append(" + consequent.getContent() + "\n");  

    }  

    public void fire(){fired=true;}  

}  

} //end of class rule

```

A.28 RuleProp.java

```

/**  

 * Class: RuleProp  

 * Author: Tangming Yuan  

 */  

abstract class RuleProp{  

    abstract String getContent();  

    //abstract RuleProp denial();  

}

```

A.29 TacticManager.java

```

/**  

 * Class: TacticManager  

 * Author: Tangming Yuan  

 * Modification: Kristj"sn Avarsson  

 */  

import java.util.*;  

public class TacticManager {  

    Move response=null;  

    public static Move getAction(Proposition focus, DialogueHistory dilv, CommitmentStore selfStore, CommitmentStore par  

        Move response=null;  

        /*  

         * If the previous move is S's thesis, then switch to the thesis level decision.  

         */  

        if (focus.equals(planner.computerThesis.denial())){response=null;}  

        /*  

         * Fire the heuristics against the previous move  

         */  

        else {  

            /*  

             * If there is a conflict, and the demolish target is an element,  

             * then issue the resolution demand  

             */  

            ConflictSet cs=partnerStore.getRealConflictSet();  

            if(cs.isEmpty()==false&&cs.onConflict(focus)) {  

                Proposition p=cs.mergeToProp();  

                response=new Move('C',"Resolve",p,cs);  

            }
        }
}

```

20

```

    } // end conflictM                                         30
M
    /*M
2) If there is a contradicted hard evidence, and the hard evidence has notM
   been asserted before, then assert the hard evidenceM
    */M
PlanSet ps=selfKBS.getPlanSet(focus.denial());M
ps.show();M
M
if(response==null) {M
    response=ps.getHardEvidentialMove(selfStore);M
}M
/*M
3) If there is a support which can be further supported.M
   If the demolish target is not on the assertion set, then assert the negation of the demolish targetM
   else assert the supportM
*/M
int i= ( int ) ( Math.random() * 2);           M
switch (i) {M
    case 0:M
        if (response==null) {M
            response=ps.startSubtopic(selfStore, partnerStore);M
        }M
        if(response==null) {M
            response=ps.startBuildPlan(dilv,planner);M
        }M
    case 1:M
        if(response==null) {M
            response=ps.startBuildPlan(dilv,planner);M
        }M
        if (response==null) {M
            response=ps.getRebutalMove(selfStore, partnerStore);M
        }M
    }M
/*M
4) If there is a support which cannot be further supportedM
   Then form a plan, the plan has not been executedM
*/M
if(response==null) {M
    response=ps.getQuestionMove(dilv, planner);M
}M
/*M
5) If the demolish target is challengable, then challenge it.M
*/M
if(response==null&&selfKBS.challengable(focus, partnerStore, selfStore)) {M
    response=new Move('C', "Challenge", focus);M
}M
} // End of maintain the current focus to demolishM
M
return response;M
} // End of get level 3 methodM
} // end class

```

A.30 Tips.java

```
/**  
 * Class: Tips  
 * Author: Tangming Yuan  
 * Modification: Kristj"sn Evarsson  
 */  
  
import javax.swing.*;  
import java.awt.event.*;  
import java.awt.*;  
import java.net.URL;  
import java.io.IOException;  
  
public class Tips extends JFrame implements ActionListener {  
    public static void main(String arg[]) {  
        new Tips();  
    }  
  
    //Not panel  
    private JPanel buttonPane;  
    private JButton introduction;  
  
    //Editor pane  
    JEditorPane editorPane;  
    private JPanel southPane;  
  
    //constructor  
    Tips() {  
        //Title and location  
        setTitle("Tips for a quich start");  
        setSize(350,540);  
        setResizable(false);  
        setLocation(200,40);  
  
        //Not panel  
        buttonPane=new JPanel();  
        introduction=new JButton("Close");//(new ImageIcon("start"+ System.getProperty("file.separator")+"Introduction.gif"))  
        buttonPane.setLayout(new GridLayout(6,1));  
        buttonPane.add(introduction);  
  
        //South pane  
        southPane=new JPanel();  
        editorPane=new JEditorPane();  
        editorPane.setBorder(BorderFactory.createMatteBorder(8,8,8,20,Color.white));  
        southPane.setLayout(new BorderLayout(0,5));  
        southPane.add("Center",new JScrollPane(editorPane));  
        southPane.setBorder(BorderFactory.createMatteBorder(0,2,2,2,Color.lightGray));  
    }  
}
```

```

    }
    //Layout the window
    getContentPane().setLayout(new BorderLayout());
    getContentPane().add("South",buttonPane);
    getContentPane().add("Center",southPane);
}

//Default information
setContent("Introduction.html");
}

//Add action listner to these items
introduction.addActionListener(this);

//pack();
setVisible(true);

addWindowListener( new WindowAdapter() {
    public void windowClosing( WindowEvent e ) {
        dispose();
    }
});;

}

//Close window actionPerformed
public void actionPerformed(ActionEvent e) {
    if(e.getSource()==introduction) {
        dispose();
    }
}

private void setContent(String htmlname) {
    //JEditorPane editorPane = new JEditorPane();
    editorPane.setEditable(false);
    String s = null;
    try {
        URL helpURL = GUI.class.getResource(htmlname);
        displayURL(helpURL, editorPane);
    } catch (Exception e) {
        System.err.println("Couldn't create help URL: " + s);
    }
}

private void displayURL(URL url, JEditorPane editorPane) {
    try {
        editorPane.setPage(url);
    } catch (IOException e) {
        System.err.println("Attempted to read a bad URL: " + url);
    }
}
}

```

60

70

80

90

100

A.31 WithdrawStrategist.java

```

/***
 * Class: WithdrawStrategist
 * Author: Tangming Yuan
 * Modification: Kristijan Åvarsson
 */
import java.util.*;
public class WithdrawStrategist {
    /**
     * Method of plan when facing with an asserted proposition
     */
    public static Vector planWithdraw(DialogueHistory dilv, CommitmentStore selfStore, CommitmentStore partnerStore, Boolean rmv) {
        Vector previous=(Move)dilv.lastElement();
        RuleProp prevrp=previous.getObject();
        Proposition supported=null;
        /* Get supported proposition */
        Proposition prevProp=null;
        if(prevrp.getClassName().getName() == "Proposition") {
            prevProp=(Proposition)prevrp;
            supported=partnerStore.getSupportedProp(prevrp);
        }
        /* Check whether the plan is under execution, if it is, then carry on */
        if (planner.currentPlan!=null) {
            /* Try to execute the plan */
            if (the plan can be continuously executed, then execute it)
                Move m=planner.currentPlan.execute(dilv, planner, partnerStore, selfStore, selfKBS);
            if(m!=null)rmv.add(m);
            /* Else, plan is abandoned, Computer would switch the current focus */
            else
                Move m2=FocusShiftManager.execute(dilv, selfStore, partnerStore, selfKBS, planner);
                if (m2!=null)rmv.add(m2);
        }
        //End of there is a plan under execution
        /* If there is no plan under execution */
        else if (planner.currentPlan==null) {
            /* If the withdrawn statement is a unique support, then challenge it */
        }
    }
}

```

```

if (supported!=null&&selfKBS.challengable(supported, partnerStore, selfStore)&&supported.equals(planner.compu
    rmv.add(new Move('C',"Challenge",supported));  

}/*  

If the withdrawn statement supports S's view, then assess whether C adhere with its thesis  

*/  

else if (prevProp!=null&&planner.computerThesis.denial().equals(prevProp)==false&&(selfKBS.support(prevrp,
    rmv.add(new Move('C',"Question",planner.computerThesis.denial()));  

}  

else if (prevProp!=null&&planner.computerThesis.denial().equals(prevProp)) {  

    rmv.add(new Move('C',"Question",planner.computerThesis));  

}  

}  

/*  

A final option would be to switch the current focus  

*/  

else {  

    Move m=FocusShiftManager.execute(dilv, selfStore, partnerStore, selfKBS, planner);  

    if (m!=null)rmv.add(m);  

}  

}//end of no plan under execution  

return rmv;  

}//end method  

}//end class
}

```

60

70

Appendix B

Evaluation script

Appendix B includes the test script which are the introduction document read to the tester in the beginning of the evaluation.

//Introduction
Velkominn og takk fyrir að koma.

Ég heiti Kristján Ævarsson. og er að vinna að lokaverkefninu mínu sem kallast Human Computer Debating System. Hluti af verkefninu er að kanna hvað notendum finnst um frumgerð af samræðu hugbúnaði, til að sjá betur hvaða hlutar purfa frekari próun.

Ég vill taka það fram að þetta er frumgerð og er í vinnslu. Ef þér t.d. finnst hluti/hlutar frumgerðarinnar erfiðir í notkun, þá mun öðrum notendum finnast það lika. Það er mín vinna að finna slika hnögра og laga.

Pegar þú ferð að nota hugbúnaðinn þá er ekkert verið að fylgjast með þér, það er ekkert tekið upp á meðan á prófun standur. Eftir að þú hefur lokið við prófun þá mun ég vista samtal þitt við tölvuna. Úr þeim gögnum get ég ekki séð hver það er sem á í hlut. P.e. ég safna öllum samtölunum saman og greini. Eg hef engan áhuga á að vita hver það er sem ákvæðið?i? samtal. Og því mun hvert samtal vera nafnlaust. Eftir prófun mun ég spyrja þig nokkura spurninga sem ég tek upp. Viðtalid verður síðan velritað upp, öllum nöfnum verður breytt ef þau koma fyrir. Eftir það verður hljóð skránni af viðtalinnu eytt.

Prófunin mun standa í 20 mínútur, en ef þú hefur ekki áhuga á að vera allan þann tíma þá er þér auðvitað frjálst að hætta á hvaða tímapunkti sem er. Eftir það mun ég spyrja þig spurninga sem tekur einungis nokkrar mínútur.

//Explain Equipments (explanation of the product and media equipments
Þessi frumgerð sem þú ert að fara að prófa er fram sett sem leikur. Þú ert að rökræða við tölvuna, og umræðuefnid er dauðarefsignar (Capital Punishment=CP). Ég ætla ekki að fara nánar út í hvernig hann gengur fyrir sig. Því þú færð allar þær leiðbeiningar sem þú þarf þegar þú ræsir hugbúnaðinn.

Ég vil taka fram að þessi frumgerð er á ensku og því hef ég vafra opinn á tölvunni á meðan á prófun standur. Í vafranum er vefsíða sem er orðabók, ef þú átt vilt þýða ákvæðið orð þá getu þú slegið orðið inn í glugga sem er efst á síðunni og valið svo enter takkan.

Pegar að vitalinu kemur mun ég taka það upp með mp3 spilar.

//Introduce tasks.
Það sem ég vill að þú gerir er að setjast fyrir framan tölvuna og prófir þennan hugbúnað í 15 mínútur. Mundu að þér er frálst að hætta hvenær sem er. Ef þú vilt hætta þá kemur þú bara fram til mína og lætur mig vita. Annars mun ég koma að 15 mínútum liðnum og við förum að spjalla

saman. Ekki loka forritinu eftir að þú ert búinn að prufa það. Núna er ég búinn að opna forritið í tölvunni. Einungis er búið að ræsa það þannig að næsta skref er að prófa.

Gangi þér vel !!!

Appendix C

Conversation between human and computer

Appendix C includes all conversation between the testers and the computers. The conversations are displayed in testers numerical order, and are separated with the header of the next conversation.

Tester 1

01: C>Is it the case that CP is acceptable?
02: S>No, I think CP is not acceptable.
03: C>I think CP is acceptable.
04: S>Why is it the case that CP is acceptable?
05: C>Because CP definitely stops murderers commit further crimes.
06: S>Is it the case that chances should be given to murderers for rehabilitation?
07: C>Yes, I think chances should be given to murderers for rehabilitation.
08: S>Why is it the case that CP definitely stops murderers commit further crimes?
09: C>I don't know why CP definitely stops murderers commit further crimes.
10: S>I think 'chances should be given to murderers for rehabilitation' implies 'chances should be given to murderers for rehabilitation'.
11: R>This is not a sound conditional, the head and the tail should be different (e.g. R implies P), please try again!.
12: S>I think chances should be given to murderers for rehabilitation.
13: C>Is it the case that execution of murderers is fair for the people being murdered?
14: S>No, I think execution of murderers is not fair for the people being murdered.
15: C>Is it the case that statistics shows an increase in murder rate since the abolition of CP?
16: S>No, I think it is not the case that statistics shows an increase in murder rate since the abolition of CP.
17: C>I think murderers should receive capital punishment.
18: S>Is it the case that chances should be given to murderers for rehabilitation?
19: C>Yes, I think chances should be given to murderers for rehabilitation.
20: S>Why is it the case that murderers should receive capital punishment?
21: C>Because execution of murderers is fair for the people being murdered.
23: S>I don't think execution of murderers is fair for the people being murdered.
24: C>I think CP is a good deterrent.
25: S>I don't think CP is a good deterrent.
26: C>I think most people want CP back.
28: S>Why is it the case that most people want CP back?
29: C>Because the recent survey shows that 60% British people support CP.
30: S>Is it the case that chances should be given to murderers for rehabilitation?
31: C>Yes, I think chances should be given to murderers for rehabilitation.
32: S>Why is it the case that the recent survey shows that 60% British people support CP?
33: C>I don't know why the recent survey shows that 60% British people support CP.
34: S>I think chances should be given to murderers for rehabilitation.

35: R>The statement 'chances should be given to murderers for rehabilitation', has already
in both positions, it is not necessarily to say it, please try other options.
36: S>I think CP is a good deterrent.
37: C>Is it the case that 'CP is a good deterrent' implies 'CP is acceptable'?
38: S>Yes, I think 'CP is a good deterrent' implies 'CP is acceptable'.
39: C>I think CP is acceptable.
41: S>Why is it the case that CP is a good deterrent?
42: C>Because statistics shows an increase in murder rate since the abolition of CP.
43: S>Why is it the case that most people want CP back?
44: C>I don't know why most people want CP back.
45: S>Is it the case that CP is a good deterrent?
46: C>Yes, I think CP is a good deterrent.
47: S>Why is it the case that CP is a good deterrent?
48: C>Because statistics shows an increase in murder rate since the abolition of CP.
49: S>I think CP is acceptable.
50: R>You seem to have changed your view.

Tester 2

01: C>Is it the case that CP is acceptable?
02: S>No, I think CP is not acceptable.
03: C>I think CP is acceptable.
04: S>I think chances should not be given to murderers for rehabilitation.
05: C>Is it the case that CP is not acceptable?
06: S>Yes, I think CP is not acceptable.
07: C>I think most people want CP back.
08: S>I think chances should be given to murderers for rehabilitation.
09: C>Please resolve 'chances should not be given to murderers for rehabilitation'
and 'chances should be given to murderers for rehabilitation' in your positions.
10: S>I don't think chances should not be given to murderers for rehabilitation.
11: C>I think CP is a good deterrent.
12: S>I think CP is not a good deterrent.
13: C>But statistics shows an increase in murder rate since the abolition of CP.
17: S>I think CP is not a good deterrent.
18: C>Is it the case that nobody is willing to die?
19: S>Yes, I think nobody is willing to die.
20: C>Is it the case that 'nobody is willing to die'
implies 'CP makes people less likely commit serious crimes'?
21: S>No, I think it is not the case that 'nobody is willing to die'
implies 'CP makes people less likely commit serious crimes'.
22: C>I think murderers should receive capital punishment.
24: S>I think CP is not a good deterrent.
25: C>I think CP makes people less likely commit serious crimes.
26: S>I don't think CP makes people less likely commit serious crimes.
27: C>Is it the case that statistics shows an increase in murder
rate since the abolition of CP?
28: S>No, I think it is not the case that statistics shows an increase in murder
rate since the abolition of CP.
29: C>Is it the case that execution of murderers is fair for the people being murdered?
30: S>No, I think execution of murderers is not fair for the people being murdered.
31: C>Is it the case that the recent survey shows that 60% British people support CP?
32: S>I am not sure about it.
33: C>Is it the case that CP definitely stops murderers commit further crimes?
34: S>No, I think denial of this is not specified.
35: C>I think if you have anything more to say, you can go on.
36: S>I think chances should be given to murderers for rehabilitation.
37: C>Why is it the case that chances should be given to murderers for rehabilitation?
38: S>Because CP is not acceptable.
39: C>I think if you have anything more to say, you can go on.

Tester 3

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
06: S>I think execution of murderers is fair for the people being murdered.
07: C>I think it is wrong to take a human life.
08: S>I think chances should not be given to murderers for rehabilitation.
09: C>Is it the case that CP is acceptable?
10: S>Yes, I think CP is acceptable.
11: C>Is it the case that human lives are scarce?
12: S>No, I think human lives are not scarce.
13: C>I think innocent people may get killed.
14: S>I think CP is acceptable.
15: C>Is it the case that political and racial bias often causes prejudices?
16: S>I am not sure about it.
17: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
18: S>Yes, I think CP does not give murderers opportunities for rehabilitation.
19: C>Is it the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'?
20: S>No, I think it is not the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'.
21: C>I think if you have anything more to say, you can go on.
22: S>I think execution of murderers is fair for the people being murdered.
23: C>I think if you have anything more to say, you can go on.
24: S>Is it the case that chances should not be given to murderers for rehabilitation?
25: C>No, I think chances should be given to murderers for rehabilitation.
26: S>Is it the case that execution of murderers is fair for the people being murdered?
27: C>Yes, I think execution of murderers is fair for the people being murdered.
28: S>Is it the case that CP is acceptable?
29: C>No, I think CP is not acceptable.
30: S>Is it the case that execution of murderers is fair for the people being murdered?
31: C>Yes, I think execution of murderers is fair for the people being murdered.
32: S>Is it the case that it is unlikely innocent people get killed?
33: C>No, I think innocent people may get killed.
34: S>I think CP makes people less likely commit serious crimes.
35: C>Why is it the case that CP makes people less likely commit serious crimes?
36: S>Because statistics shows an increase in murder rate since the abolition of CP.
37: C>I think if you have anything more to say, you can go on.
38: S>I think the recent survey shows that 60% British people support CP.
39: C>I think if you have anything more to say, you can go on.
40: S>I think suicide bombers want to die.
41: C>Is it the case that CP is acceptable?
42: S>Yes, I think CP is acceptable.
43: C>I think if you have anything more to say, you can go on.
44: S>I think CP makes people less likely commit serious crimes.
45: C>I think if you have anything more to say, you can go on.
46: S>Is it the case that scientific techniques will increase the success of justice?
47: C>Yes, I think scientific techniques will increase the success of justice.
48: S>Is it the case that most people want CP back?
49: C>Yes, I think most people want CP back.
50: S>Is it the case that it is not always wrong to take a human life?
51: C>No, I think it is wrong to take a human life.
52: S>Is it the case that CP is a good deterrent?
53: C>No, I think CP is not a good deterrent.
54: S>Is it the case that chances should not be given to murderers for rehabilitation?
55: C>No, I think chances should be given to murderers for rehabilitation.
56: S>Is it the case that CP is acceptable?
57: C>No, I think CP is not acceptable.
58: S>I think CP is acceptable.
59: C>I think if you have anything more to say, you can go on.
61: S>I think CP is not acceptable.
62: R>You seem to have changed your view.

Tester 4

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
04: S>I think murderers should receive capital punishment.
05: C>I think chances should be given to murderers for rehabilitation.
06: S>Why is it the case that chances should be given to murderers for rehabilitation?
07: C>Because statistics shows that repeat offences of murders are extremely low.
09: S>Why is it the case that statistics shows that repeat offences of murders
are extremely low?
10: C>I don't know why statistics shows that repeat offences of murders are extremely low.
11: S>I think 'scientific techniques will increase the success of justice'
implies 'scientific techniques will increase the success of justice'.
12: R>This is not a sound conditional, the head and the tail should be different
(e.g. R implies P), please try again!.
13: S>I think 'scientific techniques will increase the success of justice'
implies 'it is unlikely innocent people get killed'.
14: C>I think it is wrong to take a human life.
15: S>I think execution of murderers is fair for the people being murdered.
16: C>Is it the case that human lives are scarce?
17: S>No, I think human lives are not scarce.
18: C>I think innocent people may get killed.
20: S>Why is it the case that it is wrong to take a human life?
21: C>Because human lives are scarce.
22: S>Why is it the case that human lives are scarce?
23: C>I don't know why human lives are scarce.
24: S>Is it the case that 'it is unlikely innocent people get killed'
implies 'human lives are scarce'?
25: C>I am not sure about it.
26: S>Why is it the case that chances should be given to murderers for rehabilitation?
27: C>I don't know why chances should be given to murderers for rehabilitation.
28: S>I think 'CP does not make people less likely commit serious crimes'
implies 'statistics shows that repeat offences of murders are extremely low'.
29: C>Is it the case that political and racial bias often causes prejudices?

Tester 5

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
04: S>I think execution of murderers is fair for the people being murdered.
05: C>Is it the case that political and racial bias often causes prejudices?
06: S>Yes, I think political and racial bias often causes prejudices.
07: C>Is it the case that 'political and racial bias often causes prejudices'
implies 'there are mistakes during judicial process'?
08: S>I am not sure about it.
09: C>Is it the case that human lives are scarce?
10: S>Yes, I think human lives are scarce.
11: C>Is it the case that 'human lives are scarce'
implies 'it is wrong to take a human life'?
12: S>No, I think it is not the case that 'human lives are scarce'
implies 'it is wrong to take a human life'.
13: C>I think innocent people may get killed.
15: S>I think execution of murderers is fair for the people being murdered.
16: C>I think it is wrong to take a human life.
17: S>Why is it the case that it is wrong to take a human life?
18: C>Because human lives are scarce.
19: S>I think CP is acceptable.
20: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
21: S>Yes, I think CP does not give murderers opportunities for rehabilitation.
22: C>Is it the case that 'CP does not give murderers opportunities for rehabilitation'

implies 'CP is not acceptable'?

23: S>No, I think it is not the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'.

24: C>I think if you have anything more to say, you can go on.

25: S>Why is it the case that human lives are scarce?

26: C>I don't know why human lives are scarce.

27: S>I think it is not always wrong to take a human life.

28: C>Why is it the case that it is not always wrong to take a human life?

29: S>Because human lives are not scarce.

30: C>I think if you have anything more to say, you can go on.

31: S>I think it is not always wrong to take a human life.

32: C>I think if you have anything more to say, you can go on.

33: S>Is it the case that it is wrong to take a human life?

34: C>No, I think it is not always wrong to take a human life.

35: S>I think it is unlikely innocent people get killed.

36: C>I think there are mistakes during judicial process.

37: S>I think it is unlikely innocent people get killed.

38: C>Why is it the case that it is unlikely innocent people get killed?

40: S>Because a state has the right to deliberately execute the wrong people.

41: C>I think if you have anything more to say, you can go on.

44: S>I think nobody is willing to die.

45: C>Is it the case that suicide bombers want to die?

46: S>No, I think suicide bombers don't want to die.

47: C>I think if you have anything more to say, you can go on.

48: S>Why is it the case that political and racial bias often causes prejudices?

49: R>The statement 'political and racial bias often causes prejudices' is stated by you and agreed by the computer, you can only challenge the statement the computer have asserted (without a '*'), please try again!

Tester 6

01: C>Is it the case that CP is acceptable?

02: S>Yes, I think CP is acceptable.

03: C>I think CP is not acceptable.

04: S>I think execution of murderers is fair for the people being murdered.

05: C>I think it is wrong to take a human life.

06: S>I think CP is a good deterrent.

07: C>But CP having countries like USA has higher murder rate than UK.

09: S>Why is it the case that CP having countries like USA has higher murder rate than UK?

10: C>I don't know why CP having countries like USA has higher murder rate than UK.

11: S>I think chances should not be given to murderers for rehabilitation.

12: C>Is it the case that CP is acceptable?

13: S>Yes, I think CP is acceptable.

14: C>Is it the case that political and racial bias often causes prejudices?

15: S>Yes, I think political and racial bias often causes prejudices.

16: C>Is it the case that 'political and racial bias often causes prejudices' implies 'there are mistakes during judicial process'?

17: S>Yes, I think 'political and racial bias often causes prejudices' implies 'there are mistakes during judicial process'.

18: C>Is it the case that 'there are mistakes during judicial process' implies 'innocent people may get killed'?

19: S>Yes, I think 'there are mistakes during judicial process' implies 'innocent people may get killed'.

20: C>Is it the case that 'innocent people may get killed' implies 'CP is not acceptable'?

21: S>Yes, I think 'innocent people may get killed' implies 'CP is not acceptable'.

22: C>I think CP is not acceptable.

24: S>Why is it the case that CP is not acceptable?

25: C>Because innocent people may get killed.

26: S>Why is it the case that innocent people may get killed?

27: C>Because there are mistakes during judicial process.

30: S>Is it the case that chances should not be given to murderers for rehabilitation?

31: C>No, I think chances should be given to murderers for rehabilitation.
33: S>I think execution of murderers is fair for the people being murdered.
34: C>Is it the case that human lives are scarce?
35: S>No, I think human lives are not scarce.
36: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
37: S>No, I think CP gives murderers opportunities for rehabilitation.
38: C>I think if you have anything more to say, you can go on.
39: S>I think CP is a good deterrent.
40: C>Is it the case that CP having countries like USA has higher murder rate than UK?
41: S>I am not sure about it.
42: C>I think if you have anything more to say, you can go on.
43: S>I think CP is a good deterrent.
44: C>Why is it the case that CP is a good deterrent?
45: S>Because execution of murderers is fair for the people being murdered.
46: C>I think if you have anything more to say, you can go on.
47: S>Is it the case that execution of murderers is fair for the people being murdered?
48: C>Yes, I think execution of murderers is fair for the people being murdered.
49: S>I think execution of murderers is fair for the people being murdered.
50: R>The statement 'execution of murderers is fair for the people being murdered',
has already in both positions, it is not necessarily to say it, please try other options.
51: S>I think CP is acceptable.
52: C>I think if you have anything more to say, you can go on.
53: S>Why is it the case that human lives are not scarce?
54: R>The statement 'human lives are not scarce' is stated by you and agreed by the computer,
you can only challenge the statement the computer have asserted (without a '*'), please try again!
55: S>Is it the case that CP is acceptable?
56: C>No, I think CP is not acceptable.
57: S>Why is it the case that CP is not acceptable?
58: C>Because it is wrong to take a human life.
59: S>Is it the case that CP is acceptable?
60: C>No, I think CP is not acceptable.
61: S>I think execution of murderers is fair for the people being murdered.
62: R>The statement 'execution of murderers is fair for the people being murdered',
has already in both positions, it is not necessarily to say it, please try other options.
63: S>I think chances should not be given to murderers for rehabilitation.
64: C>Is it the case that CP is acceptable?
65: S>Yes, I think CP is acceptable.
66: C>I think if you have anything more to say, you can go on.
67: S>Is it the case that chances should not be given to murderers for rehabilitation?
68: C>No, I think chances should be given to murderers for rehabilitation.
69: S>Why is it the case that chances should be given to murderers for rehabilitation?
70: C>Because statistics shows that repeat offences of murders are extremely low.
71: S>I think execution of murderers is fair for the people being murdered.
72: R>The statement 'execution of murderers is fair for the people being murdered',
has already in both positions, it is not necessarily to say it, please try other options.

Tester 7

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
04: S>I think CP is a good deterrent.
05: C>But CP having countries like USA has higher murder rate than UK.
06: S>Why is it the case that CP having countries like USA has higher murder rate than UK?
07: C>I don't know why CP having countries like USA has higher murder rate than UK.
08: S>I think CP is acceptable.
09: C>Is it the case that political and racial bias often causes prejudices?
10: S>Yes, I think political and racial bias often causes prejudices.
11: C>Is it the case that 'political and racial bias often causes prejudices'
implies 'there are mistakes during judicial process'?
12: S>No, I think it is not the case that 'political and racial bias often causes prejudices'
implies 'there are mistakes during judicial process'.

14: C>Is it the case that human lives are scarce?
15: S>No, I think human lives are not scarce.
16: C>I think innocent people may get killed.
17: S>Why is it the case that innocent people may get killed?
18: C>Because there are mistakes during judicial process.
19: S>Why is it the case that there are mistakes during judicial process?
20: C>Because political and racial bias often causes prejudices.
21: S>I think CP is acceptable.
22: C>I think it is wrong to take a human life.
23: S>I don't think it is wrong to take a human life.
24: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
25: S>Yes, I think CP does not give murderers opportunities for rehabilitation.
26: C>Is it the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'?
27: S>No, I think it is not the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'.
28: C>I think if you have anything more to say, you can go on.
29: S>I think execution of murderers is fair for the people being murdered.
30: C>I think if you have anything more to say, you can go on.
31: S>I think if you have anything more to say, you can go on.
32: C>I think execution of murderers is fair for the people being murdered.
33: S>I think if you have anything more to say, you can go on.
34: C>I think execution of murderers is fair for the people being murdered.
35: S>I think if you have anything more to say, you can go on.
36: C>I think if you have anything more to say, you can go on.
37: S>Why is it the case that it is wrong to take a human life?
38: C>Because human lives are scarce.
39: S>I think chances should not be given to murderers for rehabilitation.
40: C>Is it the case that CP is acceptable?
41: S>Yes, I think CP is acceptable.
42: C>I think if you have anything more to say, you can go on.

Tester 8

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
04: S>I think chances should not be given to murderers for rehabilitation.
05: C>Is it the case that CP is acceptable?
06: S>Yes, I think CP is acceptable.
07: C>I think innocent people may get killed.
08: S>I don't think innocent people may get killed.
09: C>I think it is wrong to take a human life.
10: S>I think it is not always wrong to take a human life.
11: C>Is it the case that human lives are scarce?
12: S>No, I think human lives are not scarce.
13: C>Is it the case that political and racial bias often causes prejudices?
14: S>Yes, I think political and racial bias often causes prejudices.
15: C>Is it the case that 'political and racial bias often causes prejudices' implies 'there are mistakes during judicial process'?
16: S>Yes, I think 'political and racial bias often causes prejudices' implies 'there are mistakes during judicial process'.
17: C>Is it the case that 'there are mistakes during judicial process' implies 'innocent people may get killed'?
18: S>Yes, I think 'there are mistakes during judicial process' implies 'innocent people may get killed'.
19: C>Is it the case that 'innocent people may get killed' implies 'CP is not acceptable'?
20: S>Yes, I think 'innocent people may get killed' implies 'CP is not acceptable'.
21: C>I think CP is not acceptable.
22: S>Is it the case that CP does not give murderers opportunities for rehabilitation?
23: C>Yes, I think CP does not give murderers opportunities for rehabilitation.
24: S>I think execution of murderers is fair for the people being murdered.

26: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
27: S>Yes, I think CP does not give murderers opportunities for rehabilitation.
28: C>Is it the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'?
29: S>Yes, I think 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'.
30: C>I think CP is not acceptable.
31: S>I think murderers should receive capital punishment.
32: C>I think chances should be given to murderers for rehabilitation.
33: S>Is it the case that CP makes people less likely commit serious crimes?
34: C>Yes, I think CP makes people less likely commit serious crimes.
35: S>I think CP definitely stops murderers commit further crimes.
36: C>I think if you have anything more to say, you can go on.
37: S>Is it the case that the recent survey shows that 60% British people support CP?
38: C>Yes, I think the recent survey shows that 60% British people support CP.
39: S>I think scientific techniques will increase the success of justice.
40: C>I think if you have anything more to say, you can go on.

Tester 9

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
04: S>I think execution of murderers is fair for the people being murdered.
05: C>Is it the case that human lives are scarce?
06: S>Yes, I think human lives are scarce.
07: C>Is it the case that 'human lives are scarce' implies 'it is wrong to take a human life'?
08: S>No, I think it is not the case that 'human lives are scarce' implies 'it is wrong to take a human life'.
09: C>I think it is wrong to take a human life.
11: S>Why is it the case that it is wrong to take a human life?
12: C>Because human lives are scarce.
13: S>Why is it the case that human lives are scarce?
14: C>I don't know why human lives are scarce.
15: S>I think 'execution of murderers is fair for the people being murdered' implies 'execution of murderer is fair for the people being murdered'.
16: R>This is not a sound conditional, the head and the tail should be different (e.g. R implies P), please try again!.
17: S>I think 'execution of murderers is fair for the people being murdered' implies 'CP is acceptable'.
18: C>Is it the case that political and racial bias often causes prejudices?
19: S>Yes, I think political and racial bias often causes prejudices.
20: C>Is it the case that 'political and racial bias often causes prejudices' implies 'there are mistakes during judicial process'?
21: S>Yes, I think 'political and racial bias often causes prejudices' implies 'there are mistakes during judicial process'.
22: C>Is it the case that 'there are mistakes during judicial process' implies 'innocent people may get killed'?
23: S>Yes, I think 'there are mistakes during judicial process' implies 'innocent people may get killed'.
24: C>Is it the case that 'innocent people may get killed' implies 'CP is not acceptable'?
25: S>No, I think it is not the case that 'innocent people may get killed' implies 'CP is not acceptable'.
26: C>I think innocent people may get killed.
27: S>I think 'CP is acceptable' implies 'execution of murderers is fair for the people being murdered'.
28: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
29: S>Yes, I think CP does not give murderers opportunities for rehabilitation.
30: C>Is it the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'?

31: S>No, I think it is not the case that 'CP does not give murderers opportunities for rehabilitation' implies 'CP is not acceptable'.
32: C>I think if you have anything more to say, you can go on.
33: S>Why is it the case that it is wrong to take a human life?
34: C>I don't know why it is wrong to take a human life.
35: S>Why is it the case that innocent people may get killed?
36: C>Because there are mistakes during judicial process.
37: S>I don't think it is not the case that 'innocent people may get killed' implies 'CP is not acceptable'.
38: C>I think if you have anything more to say, you can go on.
39: S>I think CP is acceptable.
40: C>I think if you have anything more to say, you can go on.
41: S>Is it the case that 'CP is acceptable' implies 'execution of murderers is fair for the people being murdered'?
42: C>I am not sure about it.

Tester 10

01: C>Is it the case that CP is acceptable?
02: S>Yes, I think CP is acceptable.
03: C>I think CP is not acceptable.
04: S>Why is it the case that CP is not acceptable?
05: C>Because innocent people may get killed.
06: S>I don't think innocent people may get killed.
07: C>Is it the case that human lives are scarce?
08: S>No, I think human lives are not scarce.
09: C>Is it the case that political and racial bias often causes prejudices?
10: S>No, I think political and racial bias will not cause prejudices.
11: C>I think it is wrong to take a human life.
12: S>I don't think it is wrong to take a human life.
13: C>Is it the case that CP does not give murderers opportunities for rehabilitation?
14: S>No, I think CP gives murderers opportunities for rehabilitation.
15: C>I think if you have anything more to say, you can go on.
16: S>Is it the case that it is not always wrong to take a human life?
17: C>No, I think it is wrong to take a human life.
18: S>I don't think it is wrong to take a human life.
19: C>I think if you have anything more to say, you can go on.
20: S>I don't think it is wrong to take a human life.
21: C>I think if you have anything more to say, you can go on.
22: S>I think human lives are not scarce.
23: C>I think if you have anything more to say, you can go on.
24: S>Why is it the case that if you have anything more to say, you can go on?
25: C>I don't know why if you have anything more to say, you can go on.
26: S>I think CP makes people less likely commit serious crimes.
27: C>Why is it the case that CP makes people less likely commit serious crimes?

Appendix D

Interview questions

Appendix D includes the questions asked in the interview of the first user evaluation.

Questions for User Evaluation (first round)

Groups:

1. Interface

How it the interface, pros & cons

2. System intelligence

Stupid or Intelligence.

3. Feeling (liking)

Fun or boring

4. Future use

For user and potential.

Question 0.

Hefur þú notað samkskipta forrit áður ?

Follow up: Já - eins og ?

Follow up: Getur þú gefið mér samanburð á því sem þú hefur notað og forritinu ?

Follow up: Nei - hvað me MSN ?

Question 1.

Í byrjun kom upplýsinga gluggi. Hvernig fannst þér að fá svona upplýsingar í byrjun ?

Follow up: Gott - Hvernig þá ?

Slæmt - Að hvaða leiti ?

Hlutlaus - Er þetta nauðsynlegt ?

Question 2.

Hvernig fannst þér upplýsingarnar, skýrðu þér nægilega út fyrir þér ?

Follow up: Gott - Er þetta of mikil af upplýsingum ? Skipta allar máli ? Ef nei: Hverjar ?

Follow up: Bad - Hvert var vandamálið ? Hvernig myndir þú leysa það ?

Follow up: Hlutl. -

Question 3.

Ef þú myndir nota forritið aftur myndir þú lesa aftur yfir þessar upplýsingar ?

Follow up: Já - Getur þú gert þér grein fyrir hversu oft, í hvað mörg skipti ?

Follow up: Nei - Hvað finnst þér þá um að láta gluggann alltaf byrtast ?

Question 4.

Hvað gerðir þú við gluggann eftir að þú varst búinn að lesa það sem var í honum ?

Follow up: Loka - 1. Skoðaðir þú gluggann aftur ?

Follow up: Hvað var það sem þú vildir skoða aftur ?

2. Getur þú sagt mér hvernig hægt er að fá gluggann upp aftur ?

Question 5.

Er eitthvað sem þér þykir athugavert við þennan glugga, eitthvað sem angraði þig ?

Follow up: Já - Hvað þá ?

Question 6.

Hvernig gekk að byrja leikinn ?

Follow: Vel - Tókstu eftir textanum sem blikkaði neðst á skjánum áður en þú byrjaðir ?

Follow: Illa - Hvað var það sem ruglaði þig ?

Question 7.

Hvernig finnst þér að stafesta hvert val með því að íta á takka í hvert skipti ?

Follow. Illa: Hvað vildir þú sjá frekar ?

Question 8.

Hjálpaði textinn sem blikkaði neðst á skjánum þér ?

Follow: Hvernig þá ?

Question 9.

Hvernig gekk að átta sig á þessum 3 aðal gluggum, þ.e. hver táknað hvað ?

Follow: Illa - Hvernig þá ?

Question 10.

Hausarnir á hverjum hluta af valmyndinni, eins og t.d. gluggunum og val-boxunum sögðu þeir þér hvað þessir hlutar gera/standa fyrir ?

Follow: Nei - Hver þá ?

Question 11.

Hvernig líkaði þér að velja setningar úr svona dropdown boxum ?

Follow: Hvernig fannst þér valmöguleikarnir, of margir eða fáir ?

Follow: Myndir þú frekar vilja skrifa setningarnar sjálf/ur ?

Já: Afhverju ?

Nei: En ef þetta væri á íslensku ?

Já/nei: Afhverju ?

Follow: Er einhver annar valmöugleiki sem þér dettur í hug til að koma þínu á framfæri við tölvuna.

Question 12.

Hvernig finnst þér aðgangur að valmöguleikum og upplýsingum ?

Follow: Hvað þá ?

Question 13.

Er eitthvað sem þú tókst eftir sem ekki hefur komið fram sem þér þótt athugavert varðandi viðmótið ?

Question 14.

Hversu hratt fannst þér tölvan ná að svara þér ?

Lentir þú í að bíða ?

Question 15.

Hvernig fannst þér tölvan svara ?

Follow: Gáfulega - Náði hún að halda uppi samræðum við þig, allan tíman ?

Drasl -Getur þú sagt mér meira frá því ? Hvað fannst þér að ?

Question 16.

Hvernig líkaði þér við forritið ?

Follow: Afhverju ?

Follow: Hafiðr þú gaman af því að spjalla við tölvuna ?

Follow: Hvað svona helst var það sem þú hafðir gaman af ?

Follow: Er umhverfið skemmtilegt ?

Follow: Hvað þá ?

Question 17.

Myndir þú vilja nota þetta kerfi þér til gamans ?
Follow: Afhverju ?

Question 18.

Telur þú að það sé hægt að nota þetta kerfi svo gagn sé að ?
Follow: Hvernig þá ?

Question 19.

Sérðu möguleika á að breyta kerfinu á einhvvern hátt svo að það sé hægt að nota það ekki einungis til skemmtunar ?

Question 20.

Núna var bara eitt umræðuefni, dauðarefsingar. Myndu þú vera líklegrí til að nota kerfið ef hægt væri að velja úr nokkrum umræðuefnum ?

Question 21.

Hefðir þú áhuga á að bæta inn umræðuefnum og fá meiri sjórn yfir kerfinu ?

Question 21.

Ef kerfið væri á íslensku værir þú líklegrí til að nota það ?

Question 22.

Já, næst síðasta spurningin.

Hversu viss ertu að það var tölva sem þú varst að tala við ?

Follow: Viss - Ef ég segði þér að þú værir að tala við mig allan tíman ? Værir þú hissa ?
Litið - Heldur þú að það sé hægt að búa til í framtíðinni hugbúnað sem gerir þér kleypt að tala við tölvuna eins og þessi leikur ?

Question 23.

Er eitthvað sem þú vilt bæta við sem hefur ekki komið fram hér á undan ?

Þá er þetta búið !!

Ég vil pakka þér fyrir páttökuna, þín pátttaka mun koma að miklu gagni.

Appendix E

Interview transcripts

Appendix E includes all the interview transcripts made from the first user evaluation. The transcripts are displayed in transvers numerical order, and are separated with the header of the next transcript.

Test 1.
PILOT

Kristján: Hefur þú notað samskiptaforrit áður ?

Tester: Jaa, já MSN og svona.

Kristján: Hvernig er samanburðurinn, gagnvart þessu forriti ?

Tester: Mér fannst þetta miklul ruglingslegra

Kristján: Þetta ?

Tester: Já

Kristján: Já

Tester: Glugginn, þinn gluggi, þarna uppi Mér fannst það svoltið ruglingslegt allavega svona í byrjun.

Kristján: Já, en hérna í byrjun kom upplýsinga gluggi, hvernig fannst þér að fá svona glugga í byrjun ?

Tester: Það er nauðsynlegt fyrir þetta forrit allavega.

Kristján: Já, og hérna, fannst þér þetta skýra nægilega út fyrir þér ?

Tester: Nei, vantaði hvað implies gerði, sem maður tékkaði í.

Kristján: Var það svona eina sem þú manst eftir ?

Kristján: Ef þú myndir nota forritið oftar, myndir þú lesa yfir pennan glugga aftur ?

Tester: Já,

Kristján: Hvert skipti ?

Tester: Ekki hvert skipti, ef mér vantaði upplýsingar þá mundi ég kíkja á gluggann

Kristján: Hvað gerðir þú við gluggan þegar þú varst búinn að lesa hann yfir í fyrsta skipti ?

Tester: Ég las aftur yfir hann, ég lokaði honum ekki eftir fyrsta skiptið, en eftir annað skiptið þá lokaði ég honum.

Kristján: Purftir þú að opna hann aftur ?

Tester: Nei.

Kristján: Heldur þú að þú gætir fundið hvernig á að opna hann aftur ?

Tester: Jú ég gæti fundið út úr því, það er svona standard menu

Kristján: Er eitthvað athugavert við þennan glugga, eitthvað sem var að angra þig ?

Tester: Já, stafsetningarvillan efst uppi. Það var það eina

Kristján: Hvernig gekk þá að byrja leikinn ?

Tester: Það gekk hægt.

Kristján: Já, en svona starting game

Tester: Já ég fann það strax

Kristján: Tókstu eftir blikkandi textanum þarna neðst ?

Tester: Já já

Kristján: Hjálpaði hann þér ?

Tester: Já já

Kristján: Hvernig finnst þér að stafesta hvert val, með því að íta á takka, pessi GO takki þarna niðri.

Tester: Það er bara eins og enter, það er nauðsynlegt að gera það, t.d. ef þú gerir vitleysu þá er svo vont að geta ekki skoðað það áður en þú staðfestir.

Kristján: Skal ég segja þér, hvernig gekk að átta þig á þessum 3 aðalgluggum.
Þú, tölvan og pessi störi, sagan ?

Tester: Jú, mér fannst ruglingslegt þetta minn gluggi hennar gluggi, hvað stjarnan þýddi.
Svolitið að vefjast fyrir mér.

Kristján: Pagar þú varst búinn að nota þetta aðeins, var þetta þá OK ?

Tester: Já, það kom svona eftir nokkur skipti þá var þetta að koma.

Kristján: Dettur þér eitthvað annað í hug, eins og fyrir stjörnuna.

Tester: uuuuu

Kristján: Myndi það hjálpa þér ef t.d. það væru litir á setningum með stjörnum ?

Tester: Já !

Kristján: Hausarnir á hverjum valmöguleika, fannst þeir ná að skíra út hvað þeir gerðu ?

Tester: Já já þeir gerðu það.

Kristján: Hvernig fannst þér að velja setningarnar úr svona tveimur dropdown boxum,
að tala þannig ?

Tester: Það var svolitið, það ruglaði mig svolitið til að byrja með, að velja úr þessum dropdown boxi "I think" og petta dæmi. Og síðan átti maður annaðhvort að velja úr hinum þarna uppi eða úr hinum dropdown boxinu.

Kristján: En hvað með valmöguleikana sem þú gast búið til setningarnar úr, vöru þeir margir eða fáir ?

Tester: Nei nei, mér fannst það fint.

Kristján: Petta var, ekki nóg skýrt hvað átti að velja ?

Tester: Vantaði aðeins útskýringar á þessu.

Kristján: Myndir þú vilja skrifa setningarnar sjálfur

Tester: Nei, fint að velja svona úr.

Kristján: Dettur þér annan möguleika til að koma þínu á framfæri við tölvuna ?

Tester: Ég eiginlega veit það ekki.

Kristján: Hvernig fannst þér aðgangur að öllum valmöguleikunum og upplýsingum ??

Tester: Ég skoðaði það ekki, ég meina ég spilaði bara leikinn. Það er bara eins og í öllum öðrum forritum.

Kristján: Er eitthvað sem þú tókst eftir, eitthvað varðandi viðmótið ?

Tester: Mér fannst það algerlegt litalaust. Bara dull

Kristján: Eitthvað sem gæti poppað þetta upp ?

Tester: Litur í stað sjörnu og mismunandi litir á setningum. Litur á já t.d.

Kristján: Hversu hratt fannst þér tölvun svara ?

Tester: Bara mjög hratt, ekkert hægt að kvarta undan því.

Kristján: En hvernig fannst þér hún svara ? Hvernig fannst þér hún tala við þig ?

Tester: Mjög skringilega. Kannski allt í einu, með og svo á móti.

Kristján: Fannst þér hún ekki nægilega samkvæm sjálfum sér ?

Tester: já, einmitt

Kristján: En fannst þér hún svara gáfulega þegar þú spurðir hana af eitthverju, eða spurði þig af einhverju gáfulegu ?

Tester: Já já, kom með statistics og svona.

Kristján: Svona til að ná þessu þá fannst þér hún vaða úr einu í annað.

Tester: Já einmitt.

Kristján: Hvernig líkaði þér svo við forritið ?

Tester: Aaaa svona bara, allt í lagi.

Kristján: Umhverfið hvernig fannst þér það ? Þú ert búin nað segja að það hafi verið dull. Eitthvað annað ?

Tester: Petta umhverfi var bara ekki að heilla mig sko.

Kristján: En ef það væri fleiri umræðuefnir, ef ég myndi setja inn fleiri umræðuefnir, myndi þú þá vilja að nota þetta frekar ?

Tester: Já já, þá mundi ég hafa mjög gaman af þessu.

Kristján: En telur þú að það sé hægt að nota kerfið svona til gagns eftir þróun ?

Tester: Já að sjálfsögðu, til kennslu t.d. til að láta fólk koma með rök fyrir málí sínu. Standa á sinni skoðun og bara til að læra að rökræða.

Kristján: Þú myndir vilja að sjá fleiri umræðuefnir. Mundir þú vilja getað bætt sjálfur inn umræðuefni ?

Tester: Já það væri snild.

Kristján: Þannig að þú fengir meiri stjórn yfir kerfinu ?

Tester: Já svona notendavænt admin kerfi.

Kristján: Ef kerfið væri á íslensku, væri það betra fyrir þig, mundi líkurnar að þú myndir nota það og hafa gaman af aukast ?

Tester: Nei nei, ekki endilega, væri notendavænna fyrir íslenska notendur.

Kristján: En fyrir þig sjálfan ?

Tester: Sennilega íslensku útgáfuna.

Kristján: Jæja næst síðasta spurninginn. Hversu viss ertu að þetta hafi verið tölva sem þú varst að rökræða við, ekki manneskja annar staðar í húsinu ?

Tester: He he ég get ekki verið viss, en miðað við hversu fljótt hún var að svara þá held ég að þetta hafi verið tölva.

Kristján: Er eitthvað sem þú vilt bæta við sem ekki hefur komið fram.

Tester: Nei nei

Kristján: En hvað finnst þér um tíman sem þú fékkst, þessar 15 mínútur ?

Tester: Það mætti vera aðeins meira sko.

Kristján: Ok ! heyrðu þá er þetta bara búið, ég vill þakka þér fyrir þátttökuna.

Test 2.

PILOT

Kristján: Góðan daginn og þakka þér fyrir að koma.

Tester: takk.

Kristján: Ok, hefur þú notað samskipta forrit áður ?

Tester: Ekki við tölvu, heldur bara fólk eins og MSN og svona.

Kristján: Í byrjun kom upplýsinga glugginn fyrst, hvernig fannst þér að fá svonna glugga í byrjun ?

Tester: Bara gott !

Kristján: Hvernig þá ?

Tester: Það svona gaf mér betri innsýn út í hvað ég var að fara.

Kristján: Hvað með þessar upplýsingar sem þú fékkst skýrðu þær nægilega út fyrir þér ?

Tester: Af því að það var á ensku þá var þetta erfitt, en það skýrði sennilega nógu mikið út fyrir mér.

Kristján: Fannst þér of mikið af upplýsingum ?

Tester: Nei nei,

Kristján: Fannst þér þær allar skipta máli ?

Tester: Þegar ég var byrjuð fannst mér að ég hefði átt að lesa meira í byrjun.

Kristján: En gátu þær kennt þér nægilega ?

Tester: (ekkert svar)

Kristján: Ef þú notaðir forritið aftur, og reglulega myndir þú lesa aftur yfir gluggann ?

Tester: Já, en ég held að það sé nóg einu sinni tvisvar.

Kristján: En hvað gerðir þú við gluggann, eftir að þú varst búinn að lesa hann ?

Tester: Ég minnkaði hann, setti hann niður.

Kristján: Þannig að þú lokaðir honum ekki ?

Tester: Nei, en ég skoðaði hann ekki aftur.

Kristján: Heldur þú að þú gætir kallað gluggann fram aftur ef þú hefir lokað honum ?

Tester: Ekki í þessu umhverfi.

Kristján: Tókstu eftir valmöguleikunum uppi ?

Tester: Nei, var bara að spila.

Kristján: Er eitthvað sem þér þykir athugavert við upplýsinga gluggann, eitthvað sem angrar þig ?

Tester: Nei, ég held ekki. (Hlátur) hann var bara það besta af þessu.

Kristján: Já,(fliss)

Kristján: Hvernig gekk svo að byrja leikinn ?

Tester: Ekki nágu vel, mér fannst of mikið af gluggum.

Kristján: afhverju ?

Tester:mér fannst of mikið af gluggum. Var ekki allveg viss hvað var hvað.

Kristján: En ég meina að starta leiknum. Start new game ?

Tester: Ég tók auðvitað eftir textanum uppi sem sagði mér að þarna færi ég.

Kristján: Tókstu eftir blikkandi textanum neðst á skjánum þegar þú varst að byrja ?

Tester: Já,

Kristján: Hann sagði þér að fara upp og velja nýjan leik

Tester: einmitt

Kristján: Þegar þú varst farinn að spila, hvernig fannst þér að sko staðfesta hvert val með því að íta á takka ?

Tester: Mér fannst það allt í lagi. Já

Kristján: Var það eitthvað að angra þig ?

Tester: Nei miklu frekar þessir 3 gluggar þarna.

Kristján: En hérna, hjálpaði textinn þessi blikkandi þér í leiknum ?

Tester: Já, hann gerði það, stundum var ég í vandræðum og fannst ég stundum ráðvilt kannski bara tungumálið en og nýtti mér þá þarna textann.

Kristján: Einmitt, hvernig gekk að átta sig á þessum 3 aðalgluggum ?

Tester: Ég var svo lítið ringluð. Ég var sko, það er nóg að hafa einn.

Kristján: Það er í líkingu við það sem þú hefur notað ?

Tester: Já, ég var ekki viss hvar samræðurnar fóru fram. Betra að hafa eitthvað sem maður pekkir. Fyrst hélt ég að samræðurnar færu fram uppi, en svo fór ég að nýta mér neðri dálkinn. Og ég gekk út frá því í lokinn að þær væru þar. Ég var ekki allveg að átta mig á þessu. Allavega já, og. Þannig að ég var með athygilina út um allt og fór mikið þúður í það.

Kristján: Má þá segja að það hafi farið mikill tími í að átta sig á umhverfinu í staðinn fyrir að spila leikinn.

Tester: Já, einmitt.

Kristján: Hver gluggi og dropdown boxinn hafa allir haus. Skýrðu þeir út fyrir þér hvað þeir gerðu ?

Tester: Husarnir ?

Kristján: T.d. glugginn í vinstra horninu er Human, eða það sem þú segir !

Tester: Já, sko ég vildi hafa þessa hausa en bara 1 glugga, það þarf að koma fram.

Kristján: Hvernig líkar þér að setja saman setningar úr svona 2 dropdown boxum ?

Tester: Mér fannst það mjög gott

Kristján: En hérna, hvernig fannst þér valmöguleikarnir þar, of fair eða margir ?

Tester: Bara góðir mættu ekki vera fleiri.

Kristján: Myndi þú vilja skrifa setningarnar sjálf ?

Tester: Það væri skemmtilegra, og gæfi meiri möguleika í rökrðunum, en ég veit það ekki.

Kristján: Já ! En ef þú fengir meiri stjórn og gætir skrifað sjálf rök að umfjöllunarefni sem þú velur ?

Tester: Ég væri til í að hafa þennan valmöguleika.

Kristján: Ef þetta væri á íslensku ?

Tester: Það væri tvennt ólikt

Kristján: Myndir þú frekar vilja skrifa sjálf eða velja úr svona dropdown boxum ?

Tester: Þá myndi ég vilja skrifa sjálf.

Kristján: Er einhvað sem þér dettur í hug annað en þetta tvennt. Þ.e. dropdown box og að skrifa sjálf.

Tester: Já, hafa bara eitt box þar sem væru setningarnar með fyrri hlutanum lika, svona heilar setningar.

Kristján: Hvernig finnnst þér aðgangur að valmöguleikum og upplýsingum ? Stillingum og þess háttar.

Tester: Ég veit það ekki, ég spilaði bara leikinn.

Kristján: Er eitthvað athugavert við viðmótið sem hefur ekki komið fram ?

Tester: Neeee, ég var aðalega að átta mig á þessum 3 gluggum. Og eins og þegar það poppaði upp gluggi þar sem ég átti að velja upp. Ég vara góð er þetta.

Kristján: Hversu hratt fannst þér tölvun svara ?

Tester: Hún var rosalega fljót.

Kristján: En hvernig fannst þér hún svara eða spyrja ?

Tester: Bara mjög skemmtilega sko.

Kristján: Fannst þér hún sem jafninginn þinn svona gáfulega ?

Tester: Jaaa, spilaði svolitið inn tungumálið en já. Ég upplifði að ég væri ekki að tala við tölvu.

Kristján: Já, bara eins og þú værir að tala við vin þinn ?

Tester: Já eins og ég væri að rökræða við manneskju.

Kristján: Náði hún að halda upp samræðum við þig allann tíman ?

Tester: Já, það stóð frekar á mér heldur en henni (hlátur).

Kristján: Já, en hvernig líkaði þér að spila, þ.e. leikurinn ?

Tester: Mér fannst þetta rosalega spennandi, og að rökræða svona. En mér fannst umhverfið svo lítið ógvrekjandi en samt spennandi að rökræða við tölvuna. Kannski að maður venjist þessu. Mjög spennandi og flott sko.

Kristján: En hafðir þú gaman af þessu ?

Tester: Já, mjög spennandi. Gaman að sjá hvað kemur á móti.

Kristján: En finnst þér umhverfið skemmtilegt ?

Tester: Nei,

Kristján: Hvernig ? hvað gerði það ekki skemmtilegt ?

Tester: Ég held að það, jú, nei, það var þetta með þessa 3 glugga og fyrstu viðbrögð voru þetta...

Kristján: Fannst þér það notarlegt ? hrátt ?

Tester: Flókið !

Kristján: Varstu hrædd við það ?

Tester: Þínu, veit ekki hvort að það var tungumálið eða umhverfið.

Kristján: Segðu mér að ef ég myndi laga hnökrana, myndir þú notað þér forritið þér til gamans ?

Tester: Já tvímæla laust. Mér finnst gaman að rökræða og hérna og þetta var svona já þetta var skemmtilegt.

Kristján: Þú telur að þú gætir haft gaman af ?

Tester: Já gagn og gaman.

Kristján: Telur þú að það sé möguleiki fyrir svona kerfi að koma til gagns, verða að gagni ?

Tester: Já

Kristján: Hvernig ?

Tester: Bara í skólakerfinu, ég sé þetta allveg í grunnskólum og í leikskólum líka.

Kristján: Hvernig sérðu það fyrir þér eins og í leikskólunum ?

Tester: Bara svona síðferðislegar spurningar, stela, muninn á réttu og röngu. Og fyrir hugtök og þess háttar.

Kristján: Einmitt. Telur þú að það sé hægt að breyta kerfinu þannig að það sé hægt að hafa það þannig að það komi vel að gagni ?

Tester: Já svona síðferðislegar umræður, ef það er skýrt vel út, þ.e. hvað er verið að fara að gera. Og í leikskólunum verður að vera setningar sem hægt er að velja eins og núna.

Kristján: Núna er bara eitt umræðuefn, dauðarefsingar. Værir þú líklegrí til að nota kerfið ef það væru fleiri ?

Tester: já, auðvtiað. Hversu mikið nennir maður að rökræða um eitt umræðuefn. Já já

Kristján: Og skemmtunar gildið mundi aukast ?

Tester: Ég held að það verði að vera valmöguleikar

Kristján: En að bæta sjálf inn t.d. umræðuefn og setja sjálf inn rök með og á móti ?

Tester: Ég myndi vilja geta valið viðfangsefni en ekki setja það inn. Eitt að því sem mér fannst skemmtilegt við þetta að það kom mér á óvart. Og ég vildi ekki vita svona fyrirfram hvað kæmi.

Kristján: Nei, einmitt. En hérna ef þetta væri svona, og þú sem kennari myndir þú vilja að hafa þennan möguleika að bæta inn ?

Tester: Já, held það

Kristján: Ef kerfið væri á íslensku værir þú líklegrí til að nota það ?

Tester: já !

Kristján: Hversu viss varstu að það var tölvua sem þú varst að tala við ekki önnur manneskja. T.d. að þú værir að tala við aðra mannesku yfir netið ?

Tester: Ég veit það ekki, ég trúi þér (hlátur) en ég er ekki viss.

Kristján: Heldur að það sé hægt að búa til hugbúnað í framtíðinni til að rökræða við tölvuna, eins og þú varst að gera ?

Tester: Já, ég held það bara.

Kristján: Er eitthvað sem þú vilt bæta við, sem ekki hefur komið fram hér að undan ?

Tester: Nei, held að þetta sé bara komið. Ég hlakka bara til að prófa þegar búið er að þróa þetta áfram.

Kristján: Þá er þessu lokið og vil ég þakka þér fyrir þátttökuna.

Test 3.

Kristján: Góðan daginn og takk fyrir að taka þátt.

Kristján: Ég ætla fyrst að spyrja þig hvort að þú hefur notað samskipta forrit áður eins og MSN ?

Tester: Já, MSN

Kristján: Getur þú sagt mér hvernig samanburðurinn er, svona samskiptinn ?

Tester: Miðað við MSN ?

Kristján: Já

Tester: þarna eru setningarnar fyrirfram ákveðnar fyrir mann. MSN getur maður stitt út, lagt saman, og fer bara eftir aðalanum sem maður er að tala við

Kristján: Já

Tester: Hvernig í raun tungumál á MSN maður notar, hérrna er það fyrirfram ákveðið tungumál. Ágætis tungumál þannig maður þarf bara smá tíma til að ná því eins og með alla aðra á MSN.

Kristján: Já já, en hvernig er umhverið miðað við MSN-ið

Tester: Mér finnst þetta mjög gott umhverfi og í raun, og ef ég miða þetta við MSN þá er ég mjög ánægður það er allt bullshittið farið.

Kristján: Eins og ?

Tester: fullt af iconum sem maður hefur ekkert með að gera og veit ekkert hvað þau eiga að gera, þarna er þetta simple stupid, fjórir takkar og maður er enga stund að læra á þá ?

Kristján: Ekkert bells and wisels ?

Tester: Ekkert bells and wisels, maður fær ekkert svona (úfffffff) þegar maður horfir á þetta. Maður bara tekst á við að fara í þetta forrit og lærir á það. Í MSN þá er þetta svona, maður lærir á 5,6 functionir og hitt skipti ekki máli er í raun bara grafík.

Kristján: Já, í byrjun kom upplýsinga gluggi, hvernig fannst þér að fá svona glugga í byrjun ?

Tester: Mjög gott, Hann var samt ekki tæmandi á útskýringar fannst mér. En mjög gott,

hann leiddi mann á stað og svo gatur maður fiktað sig áfram.

Kristján: Já, hvað fannst þér vanta svona í hann. Fannst þér upplýsingar vanta í hann sem þú svo rakst á í leiknum ?

Tester: Uuu, það er eina sem ég sé í raun og veru, sem ég sé til að gera hann tæmandi þá hefði ég viljað sjá er í raun og veru, svona lítinn glugga með exempla að hvernig þú notar hlutina. En það kom strax, það tók bara 2-3 mínútur að koma sér inn í hann. Og þetta er afleiðing af því að maður hafði svo stuttan tíma í prófuninni að maður varð óánægður með gluggann. En ef maður hafði allan daginn þá væri það fint.

Kristján: Værir þú frekar til í að fá svona glugga svona sem tutorial með myndum og ?

Tester: Já bara stuttum til að koma sér af stað og og en en það mætti ekki vera langt, en eins og ég segi þá vara þetta finn gluggi, stuttur og hnittmiðaður gluggi. En samt hefði ég vilja fá svona eitt hratt yfir hvað er hvað.

Kristján: Já, skil þig. Ég er að velta fyrir mér að ef þú notar forritið aftur, reglulega segjum það, myndir þú lesa yfir gluggann aftur, ef þú t.d. notar það á morgun ?

Tester: Án ef, ég myndi nota upplýsinga gluggan 1-2 í viðbót. Á meðan ég er að ná allveg fulkommu valdi á forritinu. En eftir það myndi ég ekki lesa hann.

Kristján: Telur þú að hann myndi angra þig ef hann kæmi alltaf upp í hvert skipti svona fremst ?

Tester: Ég myndi vilja hafa val um hvort að hann komi upp.

Kristján: En hvað gerðir þú við gluggan eftir að þú varðst búinn að lesa hann ?

Tester: Lokaði honum

Kristján: Lokaðir honum ?

Tester: Já

Kristján: Vildir þú fara aftur í hann og skoða hann ?

Tester: Nei,

Kristján: En sástu einhvern möguleika hvort að það væri hægt ?

Tester: Ég skoðaði það ekki einu sinni, ég var bara kominn í það að fikta í leiknum.

Kristján: Einmitt, er eitthvað sem þér þykir athugavert við þennan glugga, eitthvað sem var að angra þig ? ég meina fyrir utan það sem þú sagðir með myndir ?

Tester: Upplýsinga gluggann ?

Kristján: Já, eitthvað annað sem þér dettur í hug.

Tester: Ég meina, það sem ég ég, já ég get sagt heilmargt, hann var rosalega vel hannaður upp á það að maður gat lesið hann án þess að hreyfa augun

Kristján: Já

Tester: Maður var ekkert að böggast við að leyta í honum, maður bara las hann niður. Hann kom fyrir miðju skjásins þannig að maður var ekkert að leita af honum til hægri eða vinstri. Og maður fór í raun og veru að skoða forritið fyrr en maður var búinn að lesa hann.

Kristján: Hann náði athygli þinni ?

Tester: Hann náði athyglinni. Og hann náði að halda manni við athyglina að maður fór ekkert að kíkja til hlíðar hvað gæti þetta og hitt verið, maður las hann í gegn, svo hugsaði maður til baka hvað stóð í honum.

Kristján: Hvernig gekk þér að byrja í leiknum, svona að starta leiknum ?

Tester: Jú jú, það gekk ágætlega en ég ég tók ekki eftir að það væru fleiri setningar, en það kom fyrsta val upp, ég tók ekki eftir scrollbarinum nógu snemma.

Kristján: Já

Tester: En það sem fór allveg með mig, var að mótspilarinn minn kom með setningu, eftir að búið var að leggja árar í bát, s.s. if you, já já já

Kristján: Já

Tester: If think you have any thing more to say, you can go on.

Kristján: lína 23

Tester: Já, lína 23, en þetta kom nokkrum sinnum fyrir.

Kristján: Já

Tester: Þetta er eins og svona diss. Þetta er svona

Kristján: Já

Tester: Forritið er að ég ætla ekkert að tala við þig meira um þetta. Í staðinn fyrir að finna rök sem koma á móti.

Kristján: Já

Tester: Svo er það annað, það kemur spurning finnst þér ekki eitthvað

Kristján: Já

Tester: Maður getur ekki spurt til baka afhverju ætti mér að finnast það

Kristján: OK

Tester: Pannig að það vantar tvöfalda rökfærslu í forritið.

Kristján: Umhum

Tester: Pannig að maður verður að svara á móti svari. Eða s.s. að maður verður að svara spurningunni. Maður getur ekki hvernigð spurningu og fengið staðreyndarfestinu eftir spurninguna.

Kristján: hvernig fannst þér að núna bjóstu til setningar úr þessum dropdown boxum, hvernig fannst þér að staðfesta val þitt með því að íta á takka ?

Tester: Það var allt í lagi

Kristján: var það ekkert að angra þig ?

Tester: Maður er vanur úr öðrum samskipta forritum að þurfta að senda, og þetta er í raun bara send í stað GO.

Kristján: uu, textinn sem blikkar þarna neðst á skjánum, hjálpaði hann þér í einhverjum tilvikum.

Tester: Nei,

Kristján: Nei ok

Tester: nei ekki neitt, nema núna, að nún get ég byrjað nýjan leik. Það var í raun eini textinn sem hefur hjálpað mér, því ég gafst upp í leiknum (kemur meira sem skipti ekki máli).

Kristján: Einmitt, Hvernig gekk að átta sig á þessum 3 aðalgluggum í leiknum ?

Tester: vel

Kristján: Hver táknað hvað og svona?

Tester: vel, ég er vinstra megin, mótpilarinn minn hægra megin og svo sagan, hvað við erum að ræða, mér fannst eiginlega gott við þetta að maður getur tekið stöðunu hvar maður er í leiknum.

Kristján: Já,

Tester: Og hvaða taktík maður getur notað

Kristján: Stundum kom grænt box utan um, tókstu eftir því ?

Tester: Já,

Kristján: og stjörnurnar og það ?

Tester: Nei ég hafði ekki hugmynd hvað pessi grænu box virka og þessar stjörnur, ég nýtti mér það ekki.

Kristján: OK, hver hluti af valmyndinni, þessir gluggar og dropdown boxinn, hafa svona hausa, hjálpaði það þér eitthvað, skýrir það út fyrir þér hvað þeir gera ? t.d. eins og computer posision skilur, fynnst þér þetta nögu lýsandi ?

Tester: Já, mér finnst þetta nögu lýsandi, ef maður les í gegnum þetta þá sér maður hvað gerist í raun og veru. Og eftir að ég skoða þetta núna þá er þetta að maður í raun og veru, já að maður getur tekið stöðun og hvað eru helstu mótrökin, og hvað hún sytur föstust við og hver taktíkin er.

Kristján: Já,

Tester: Forritið hefur pennann glits, að það kemur ekki sko, spurning og síðan staðhæfing fyrir spurninguna, það kemur ekki staðhæfingar færsla fyrir spurningarnar(svo meira um línu 23).

Kristján: En hvernig likar þér að velja svona setningar úr svona dropdown boxum ?

Tester: Bara fint

Kristján: Fannst þér valmöguleikarnir of margir eða fáir ?

Tester: Petta er bara mjög skemmtilegur textaleikur, kosturinn við þetta er að maður þarf ekki að hamast við að setja saman setningu sem að talvan skilur.

Kristján: Já, næsta spurning er einmitt, hvort að þú vildir frekar skrifa setninguna sjálfur ?

Tester: Ekki í þessu tilfelli.

Kristján: OK

Tester: Ef maður miðar þetta við gömlu textaleikina, þá var maður kannski hálfan dag til að komast í gegnum eina setningu, þannig að þetta er í raun mun skemmtilegra upp á það að gera að maður er að reina að rökfæra eftir forsendum.

Kristján: Ef þetta væri á íslensku, myndir þú frekar vilja skrifa setningarnar sjálfur ?

Tester: Ég held að maður mundi lenda í sömu vandræðum hvort sem þetta er á ensku eða íslensku upp á það að talvan skilji framsetninguna og ég held líka að leikurinn mundi hægjast því hún væri að reikna út setninguna. Þá er maður farinn að býða eftir svari og þá væri þetta ekkert skemmtilegt.

Kristján: Já, nákvæmlega, eru einhverjir fleiri möguleikar til að koma þínu á framfæri, annað en að skrifa setninguna sjálfur eða velja úr svona dropdown boxum ?

Tester: Það er náttúrulega sko, það væri gaman að spila afbrigði af þessu þannig að hægt væri að setja inn random answer

Kristján: Já

Tester: Þar sem í raun og veru, búinn að setja fyrir að maður er allveg á móti og ég ætla ekki að breyta út af því, talvan er farinn að setja, og maður er aðeins að stopp og maður getur haft möguleika til að setja inn random answer til að hjálpa manni áfram.

Kristján: Uuu ok, en hvernig fannst þér aðgangur að valmöguleiku, settingins og upplýsingum, tókstu eitthvað eftir því, fórstu eitthvað að fikta í því ?

Tester: Þetta er mjög aðgenilegt, og í alla staði hægt að komast að öllu.

Kristján: OK, en er eitthvað sem þú tókst eftir sem hefur ekki komið fram, varðandi viðmótið, interfaceið ?

Tester: Já, hérna (pása) mér fannst eitt voðalega skrítið þegar ég byrjaði leikinn,

Kristján: Já

Tester: þegar að boxið kom upp og ég lokaði, þá kom smá byð pangað til að ég gat farið að nota forritið

Kristján: Hvernig þá ?

Tester: bara 2-3 sec áður en að forritið fór að responsa á móti

Kristján: Þá í byrjun ?

Tester: Já, þá hefur það verið að loka glugganum og hugsa sinn gang.

Kristján: Nákvæmlega

Tester: Það var það eina

Kristján: Hversu hratt fannst þér tölvun svara þér ?

Tester: Nággilega hratt

Kristján: Hvernig fannst þér hún svara þér, þá er ég að spá í þessum intelligent hluta ?

Tester: Hún svaraði mjög vel, þeim hlutum sem voru auðljóslega, sem voru augljóslega auðsvaranlegir miðað við það viðmóti sem maður var búinn að setja sér upp.

Kristján: Náði hún að halda samræðum við þig allan tíman ?

Tester: Já, hún náði pangað til að hún fór að með þess hundleiðinlegu setningu þarna.

Kristján: Já

Tester: If you have anything more to say, please continue, þetta er svona

Kristján: Þetta er diss eins og þú segir

Tester: Já þetta er dissið, og þetta er eina sem ég sé gallan í samræðunum

Kristján: Já

Tester: Ef maður væri að rökræða við manneskju þá væri þetta tímapunkturinn sem maður segir ok "Þú ert ekki viðræðuhæfur lengur".

Kristján: Skil þig, en hvernig líkaði þér svo við forritið svona almennt ?

Tester: Vel, og þetta er textaleikur sem ég get hugsað mér að halda áfram að leik með.

Kristján: Þú kafið gaman að spjalla við tölvuna ?

Tester: Já ég hafði gaman af þessu, og maður var farinn hugsað sér taktíkt hvað setningar á ég að segja, í hvaða röð á ég að segja þær til að brjóta hana niður.

Kristján: Já

Tester: Pannig að það var komið (hóst) strategia í þetta maður var farinn að hugsa sér game play, maður var farinn að hugsa 5 til 10 leiki fram í tíman.

Kristján: Einbeita þér að reina að brjóta hana ?

Tester: Já, og maður var farinn að passa sig líka á því að segja ekkert vitlaust.

Kristján: Já, en hvernig finnst þér umhverfið sem að þú situr við ? Myndir þú segja að það væri skemmtilegt ?

Tester: Skemmtilegt ? Grafiks lega séð er það hundleiðinlegt (hlátur), sem textaleikur er þetta mjög þæginlegt umhverfi.

Kristján: Já

Tester: Og þetta er, ef maður hugsar þetta út frá textaleikjum þá myndi ég segja að þetta væri skemmtilegt.

Kristján: Já,

Tester: En ef þetta væri leikur sem væri að koma út í dag og ég ætti að fara út í búð að kaupa hann á 2000 kall, ekki séns !

Kristján: hvað þá út af umhverfinu, myndir þú vilja sjá ?

Tester: Nei, mér finnst þetta leikur sem ég ætti að fá gefins á netinu

Kristján: já einmitt

Tester: En en, mér finnst þetta skemmtileg uppsetning á textaleiknum og mér finnst þetta í raun og veru minna á gömlu, PC mennirnir sem byjuðu á 186 og voru að spila golf í textaformi og og og kingsquest og þessa leiki þar sem ein setning var upp á skjánum í einu. Ef maður hugsar það út frá því þá er þetta miklu meira af upplýsingum sem þú hefur, maður getur farið til baka og skoðað það sem maður er búinn að gera þó svo að það sé ekki á grafísku formati.

Kristján: Mundi það auka skemmtanagildið þitt ef það væri meira lifandi, t.d. eins og fleiri litir ?

Tester: Nei, ekki, uu ég myndi ekki segja það í þessu formati. Fyrir mig myndi í raun

og veru skemmtilegra að ef það væri maður á mann og í raun að talvan dómarí

Kristján: Svona eins og MORFÍS ?

Tester: já, og leikurinn gæti orðið skemmtilegri þannig. En miðað við það að það er maður á tölvu þá held ég að það séu mjög fáar breytingar sem hægt væri að gera.

Kristján: Já

Tester: En ef maður horfir á þetta þá er þetta eins og maður sé að fara í ritvinnslu,

Kristján: Já (hlátur)

Tester: (hlátur) ekki miskilja mig þetta er mjög flott miðað við að þetta sé textaleikur.

Kristján: Já, en myndir þú nota þetta kerfið þér til gamans ?

Tester: Já allveg tvímælaust.

Kristján: Já, en telur þú að hægt sé að nota þetta kerfi svo gagn sé að ? lært af þessu.

Tester: já, ég myndi nota þetta kerfi til að reyna að brjóta það niður, ná mínum vilja fram. Ég hugsa að ég myndi ekki læra af því að ég myndi taka rökum

Kristján: En telur þú að það sé hægt að nota það í skólakerfinu til að efla rökhusun og svoleiðis ?

Tester: Já, það væri hægt, en þá bara með því að hún myndi ekki svara manni þannig "það er ekki til neitt svar hjá mér þú verður að segja eitthvað meira, til að ég geti svarað þér"

Kristján: Nú er bara eitt umræðuefni, dauðarefsingar. Myndir þú vera líklegri til að nota kerfið ef hægt væri að velja úr mörgum umræðuefnum ?

Tester: Allveg án efa.

Kristján: Og hérna, uuu myndir þú hafa áhuga á að þú værir svona administrator, gætir bætt inn umræðuefnum, hafa svona meiri sjórн á forritinu ?

Tester: Ég myndi frekar vilja geta að senda inn og sagt "hverníg væri að ræða þetta" í staðinn fyrir að búa til hlutinn í hrингum það.

Kristján: Já, en myndir þú vilja, ertu að setja að þú vildir t.d. að velja úr 10 umræðuefnum ?

Tester: Segjum sé svo að það sé nú, að það sé val um 4 umræðuefni, og ég sé búinn að fara nokkrum sinnu í gegnum það og fleiri örugglega og svo dættur mér í hug að það er eitthvað að gera í heiminum, Írak stríðið og eitthvað svona, þá get ég sent inn sem eru að gera þetta, "hvað með að setja upp module fyrir Írak stríðið ?"

Kristján: Því væri þá bætt inn af öðrum og þú myndir svo nota

Tester: Já, ég myndi vera tilbúinn að vera notandi.

Kristján: Ef kerfið væri á íslensku, myndi þú vera líklegri til að nota það, ef þú hefðri valmöguleika að nota íslenska og enska útgáfu ?

Tester: Uuuu, nei ég væri ekki líklegri til þess.

Kristján: það bara kæmi út á það sama ?

Tester: Frekar fyrir mitt leiti væri það fráhrindandi.

Kristján: Jæja þá er það næst síðasta spurningin, hversu viss ertu að það hafi verið

tölvu sem þú varst að tala við ekki einhver yfir netið ?

Tester: (hlátur) (hlátur) (barið í borðið) Ég er mjög viss. Ef þetta hafi ekki veirð talva, þá hafi ekki komið oft í röð setningar til að láta mann halda áfram. Human response hafi verið að segja eitthvað við mann.

Kristján: Ef ég segji þér að þú varst að tala við mig allan tíman annars staðar í húsinu, myndi þú vera hissa ? Myndi það koma þér á óvart ?

Tester: Já það myndi koma mér á óvart !

Kristján: Ok, en hérrna, bara að láta þig vita að þú varst að tala við tölvuna, bara tékka á þessum möguleika hvað fólk finnst. Er eitthvað sem þú vilt bæta við sem hefur ekki komið fram hérrna á undan ?

Tester: (þögn) Já, það mætti vera scoreboard,

Kristján: Já

Tester: Það mætti vera svona scoreboard, ef maður er að spila leikinn oft

Kristján: Já

Tester: þú veist kemur þú hefur unnið 2 sinnum og þú hefur tapað þetta oft, og möguleiki til að reseta þannig að maður getur byrjað á nálli þegar maður er illa hallandi undan vindu.

Kristján: Já,

Tester: Jú það mætti vera svona indicator

Kristján: Uuuu

Tester: Svona er maður að ná henni upp, eða er maður allvega að tapa.

Kristján: Pannig að þú myndir vilja fá svona stig eftir röksemadafærslum og hvernig er að gera það, t.d að þú værir kominn með 5 stig og talvan 3, og þú værir kominn með yfirhöndina.

Tester: Já

Kristján: Já

Tester: En samt ekki gefa upp hve mikill munurinn er, ég myndi ekki vilja vita að ég væri með 5 stig og talvan 3 stig. Ég myndi vilja sjá það grafiskt.

Kristján: Er þá að tala um súlur eða ertu að tala um process bara 0 til 100 ?

Tester: Nei ekki, ég er að tala um segjum bara analog mæli, öðrumeginn talvan og ég hinum megin.

Kristján: Ok, heyrdu þá er þetta búið, ég vill þá þakka þér bara fyrir.

Test 4.

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Jæja, u fyrsta spurning hefur þú notað samskiptaforrit áður ?

Tester: Nei

Kristján: hvað með að tala við aðra, persónur MSN og svona ?

Tester: Jú jú

Kristján: Petta er samskiptaforrtið, þarna ertu að tala við tölvuna á MSN ertu að tala við fólk, getur þú gefið mér svona samanburð á þessum samskiptaforritum ?

Tester: Allaveg petta forrit hefur þú afmarkað domain. Afmarkað space, þú hefur bara ákveðin svör sem þú getur notað, getur ekki sagt sem þú vilt segja, þú hefur þú þarf að velja úr svörunum.

Kristján: Ok, en hvað með svona viðmótið, guið, er mikill munur þar á ?

Tester: Já, það er svona svipað, það eru gluggar og gluggar, yfirleitt er það bara í einum glugga.

Kristján: Umhum, finnst þér það betra eða verra að hafa þetta í einum glugga ?

Tester: Ég myndi segja að það væri betra, þegar ég var að nota þetta forrit hjá þér núna, þá var ég að lesa í 3 gluggum alltaf

Kristján: Já

Tester: Til að finna samtalið

Kristján: En hérna í byrjun kom svona upplýsinga gluggi þegar þú startar forritinu

Tester: Nauðsynlegt

Kristján: Afhverju ?

Tester: Þú hefur ekki hugmynd annars hvað þú ert að fara að gera við þetta.

Kristján: Umhum, en hvernig fannst þér upplýsingarnar, skýrðu þær næjilega út fyrir þér ?

Tester: Já, þær skýrðu ágætlega út já, en það vantaði tilganginn, að útskýra fyrir notandanum tilganginn með leiknum, whats the goal.

Kristján: en fékkstu þú nægilegar upplýsingar til að spila leikinn.

Tester: Já já, þarf að vera svona kynning,

Kristján: Ef þú myndir nota forritið aftur, hversu oft telur þú að þú myndir þú nota svona upplýsingar glugga

Tester: Ég held að ég myndi nota í svona fyrstu 2-3 skiptinn.

Kristján: Fannst þér þetta of mikið af upplýsingum, var eitthvað þarna sem skipti ekki máli ?

Tester: Petta var heilmikil langloka já, það má allveg örugglega klippa þetta svolitið niður, ef ég má koma með athugasemd svona almennt pá truflaði það mig að enskan var ekki fullkominn.

Kristján: Já, t.d. stafestningarávilla í byrjun

Tester: orðalags villur aðalega

Kristján: Hvað gerðir þú við gluggann eftir að þú varst búinn að lesa hann ?

Tester: Ég lokaði honum ekki, til þess að hafa ef ég þyrfti að kíkja á hann aftur.

Kristján: Skoðaðir þú hann aftur ?

Tester: Nei

Kristján: Og hérna segðu mér, er eitthvað athugavert við þennan upplýsinga gluggann annað en orðalaga og þess háttar ?

Tester: Miðað við magnið á textanum þá var hann of lítill, óþægilegt að scrolla niður.

Kristján: Þegar þú varst kominn í gegnum þennan upplýsinga glugga hvernig gekk að byrja leikinn, þú veist að starta honum ?

Tester: Það var mjög skýrt

Kristján: Tókstu eftir blikkandi textanum þarna niðri

Tester: Já, mjög pirrandi

Kristján: Finnst þér hann pirrandi já ?

Tester: Já (hlátur)

Kristján: En hvernig finnst þér að staðfesta hvert val með því að ita á takka, þennan go takka í hvert skipti ?

Tester: Mér finnst það nauðsynlegt, vegna þess að, það er valmöguleikinn þarna til vinstri "I think" og það allt saman, ég var kannski búinn að velja eitt svona, svo var ég búinn að skoða hvað var í boði í miðju glugganum, svo fór ég til baka og breytt þá, samsetningin til að búa til míni input,

Kristján: um hum

Tester: ég vildi ekki vera búinn að festa niður fyrsta "I think" "Why is that the case", ég vildi ekki negla það niður áður en hitt var ákveðið

Kristján: OK, hvernig gekk, ég meina þessi blikkandi texti sem þér fannst pirrandi, hjálpaði hann þér eitthvað ?

Tester: Já já maður komst nú ekki hjá því að kíkja ekki á hann, jú hann hjálpaði, en bara óþæginlegt að hafa hann blikkandi, og ég held líka sérstaklega litarvalið og óþæginlegt að hafa svona 2 liti.

Kristján: Pannig að ef svona texti væri þarna niðri án þess að blikka og væri bara í einum lit. Telur þú að það myndi hjálpa þér ?

Tester: Já

Kristján: Hvernigekk að átta sig á þessum 3 aðalgluggum, þú veist hver táknað hvað ?

Tester: Já já áttaði mig allveg á því

Kristján: Hver hluti, gluggarnir, dropdown boxinn hafa haus, sem segja hvað þeir gera, standa fyrir, fannst það skýrt, hvað hver hluti gerði ?

Tester: hvaða dropdownin gerði ?

Kristján: Já bara gluggarnir og allt petta

Tester: Já ef computer var með a implies b, og ég sá ekki fyrr en í restina að það væri hægt að scrolla til hliðar til að sjá restinga af setningunni. Það var bara ekki mjög áberandi að það væri hægt að scrolla. Sumar setningarnar voru með stjörnu og aðrar ekki. Ég var búinn að lesa í leiðbeiningunum hvað það væri. Út ég myndi jafnvel segja að það matti bara ef þú ætlað að halda áfram þessa 3 glugga að hafa þá 2 skipta. Annars vegar sem er gilt og hinsvegar sem er ógilt. Af því að sjörnunar, þetta kemur inn á milli, þú eruð með 4-5 setningar og 1 er ekki með stjörnu og 2 er með, 3 setninginn ekki og 4 með.

Þannig að það er verið að groupa saman þá sem eru með og án stjörnu

Kristján: Já, skil þig. En hvernig líkar þér að velja setningarnar úr svona 2 dropdown boxum ?

Tester: Fyrsti hlutinn "I think" "I dont think" var mjög skýrt og þægilegt.

Kristján: Já

Tester: Hinn hlutinn það var mögj erfitt að fara í gegnum hann, því það var ekkert skipurlag á setningunum. Nema að þær eru 2 og 2 saman.

Kristján: Já

Tester: með og á móti

Kristján: Já

Tester: Þær eru ekki í stafrófsröð, það hefði hjálpað. Það mætti vera svona heading yfir hverjum, þetta fjallar um stjórnmál, þessi fjallar ummorðingja og það mætti groupa þær saman eftir umfjöllunarefnini, stafrófsröð

Kristján: Það myndi hjálpa ?

Tester: Það myndi hjálpa aðeins, sérstaklega ef þú myndir groupa eftir umfjöllunarefnini.

Kristján: Fannst þér valmöguleikarnir of fáir eða margir ?

Tester: of margir til að ná yfirsýn yfir þetta.

Kristján: En þeir voru það margir þannig að þú gast komið þínu á framfæri ?

Tester: Já, það var eitt sem mér fannst vanta, mjög óþæginlegt því ég veit p implies q og almennur notandi á ekki eftir að skilja neitt hvað það þýðir.

Kristján: Hum hum.

Tester: Hinsvegar ef það á að setja það upp. Ég gerð feil, fyrsta skipti sem ég notaði implies, man ekki hvað ég notaði "I think" eða eitthvað. Svo valdi ég eitthvað og ítt á implies og þá ætti að koma annar gluggi, þú ættir að sjá fyrri hlutan, til að geta haft það til hliðsjónar á meðan þú ert að velja q.

Kristján: Einmitt, myndir þú vilja skrifa setningarnar sjálf inn ?

Tester: Já, það væri rosalega fint, en það væri rosalega,

Kristján: En ef þetta væri á íslensku, myndir það skipta máli fyrir þig ?

Tester: Ekki fyrir mig

Kristján: já, en er einhver annar valmöguleiki sem þér dettur í hug til að koma þínu á framfæri við tölvuna, við erum búinn að tala um dropdown box og skrifa sjálf inn ?

Tester: Það er spuring með radiobuttons, það gæti verið svolitið mikið til að velja úr. Annað varðandi implies

Kristján: Já

Tester: Þá datt mér með dropdowninn með því sem að velja úr, ég myndi vilja sjá hérna svipað sem væri hægt að merkja við bara not, negation á því sem þú ert að velja, það eru 2 og 2 setnignar með og á móti, þær eru ekki orðaðar þannig að þær séu á móti hvor annari.

Kristján: Hvað fannst þér aðgangur að valmöguleikum og upplýsingum, setttings og svona,

skoðaðir þú það eitthvað ?

Tester: Nei,

Kristján: Nei, Er eitthvað sem ekki hefur komið fram sem þér þótt athugavert varðandi viðmótið guð ?

Tester: Það er sérstaklega petta að þegar ég er að búa til constructing conditional, að þá finnst mér að það pyrfti að vera, að sjá báða, sjá fyrri hlutan þegar þú smellir á implies ættiru að sjá annan dropdown

Kristján: OK, en hversu hratt fannst þér tölvan svara þér ?

Tester: bara ágætlega

Kristján: Lentir þú aldrei í því að bíða ?

Tester: Nei nei.

Kristján: En hvernig fannst þér tölvan svara og spyrja ef því er að skipta ?

Tester: Yfirlieitt ekki út í hött, var svolítið mikið "I do not know why this/that" þegar ég byrjaði á einum leik á undan og spurði, fyrsta spurningin er "Is CS..", hvernig er setningin aftur

Kristján: Er CS acceptable

Tester: Já, og þá var það sennilega hvoru meginn þú ætlar að vera,

Kristján: Já

Tester: Og ég sett "I am not sure"

Kristján: Umhum

Tester: Þá sagði tölvan bara ja, ég vill ekki leika við þið, þú þaft bara að gera upp hug þinn og koma aftur síðar.

Kristján: Nákvæmlega, hvað myndir þú vilja sjá frekar er þú svara þeim möguleika

Tester: Tölvan ætti að reyna að sannfæra mig á sína hlið.

Kristján: OK

Tester: Svolítið varðandi, minnir mig á það, í upplýsingum. Þá er capital punishment skammstafað sem CP, computer er skammstafð sem C og you are skammstafað sem S

Kristján: Já

Tester: Afhverju ?

Kristján: Student

Tester: Já, en það, þú ert ekkert alltaf að fá þá til að gera þetta.

Kristján: En hvernig líkaði þér við forritið svona heilt yfir ?

Tester: Það er, það þarf töluvert að slípa það til.

Kristján: Afhvaða leiti ?

Tester: Um leið og það er búið að laga enskuna, það mundi það hækka um 2 heila í

einkunn hjá mér

Kristján: Já, enskan að laga, gui-ið ?

Tester: Gui-ið er, það vantar þegar maður er að búa til conditional að þá þarf tu að sjá hinn á undan

Kristján: Umhum

Tester: Nauðsynlega uu, mér fannst gott að þú gast valið fyrrihlutan "Why is that a case" eða datt daddara eitthvað svona

Kristján: Já

Tester: Og og flett eitthvað á milli og farið svo til baka og breytt og sagt æi nei "I think" eitthvað

Kristján: Já

Tester: Það er gott, uu, þetta með að skipta gluggunum upp í, OK ég kannski skil frekar gluggana 3 þegar þú ert með samtalið í gangi, en mér finnst minnsta áherslan vera á því, á samtalini, það er miklu stærra letur á hinum 2 gluggunum hvort að þú sért með eða á móti einhverju

Kristján: já

Tester: Það er spurning hvort að það sé ekki hægt þarna meða og á móti að koma því einhvernveginn öðruvisi, byrta það einhvernveginn öðruvisi

Kristján: En þetta umhverfi fannst þér það skemmtilegt, höfðaði það til þín ?

Tester: Nei nei, þetta er ekki skemmtilegt, þetta er mjög þurt.

Kristján: já, svona dull eins og maður segir

Tester: já

Kristján: En hafðir þú gaman að spjalla við tölvuna ?

Tester: Þetta er ekki nokkuð sem ég hef áhuga að spjalla um, capital punishment þannig að

Kristján: En svona hafðir þú gaman að því að argumenta við hana ?

Tester: Það er, fyrir mér er þetta svona stærðfræði dæmi að reyna að finna út dæmi sem tölvan gæti ekki svarað,

Kristján: Já

Tester: Ég var í raun ekki að leika með tilfiningunum, heldur reyna að finna út equation sem virkaði

Kristján: En eftir frekari þróun, heldur þú að þú myndir vilja nota svona kerfið þér til gamans. Þá væri fleiri umræðuefnir og svona, og þú gætir valið úr umræðuefnir ?

Tester: Persónulega sé ég það ekki að ég myndi hafa gaman af því, en ef þú ert að spá í það þannig að þetta gæti verið þjálfunartól fyrir fólk sem á erfitt með að tjá sig

Kristján: Já

Tester: Sem á erfitt með að standa á rökum sinum, sem að jú væri nokkuð sem ég ætti að gera.

Kristján: Það var einmitt næsta spurning, hvernig kerfið kæmi til gagns. En sérðu möguleika á að breyta kerfinu á einhvern hátt þannig að það komi til gagns ekki bara til gamans.

Tester: Já, þjálfa fólk i að halda upp rökræðum.

Kristján: Nú er einungis eitt umræðuefní dauðarefsingar, mundir þú vera líkleg til að nota kerfið ef hægt væri að velja úr nokkrum umræðuefnum

Tester: Já það myndi ég gera.

Kristján: En hefðir þú sjálf sko áhuga að fá að bæta inn umræðuefni, fá meiri stjórn yfir kerfinu, velja umræðuefni sem að þér líkaði og þú myndir setja inn rök og annað ?

Tester: Já ég held að það gæti einmitt verið hluti að því að þjálfa í rökfræði tækni.

Kristján: En hérna segðu mér ef kerfið væri á íslensku væri þú líklegri til að nota það ?

Tester: ef það væri á íslensku ?

Kristján: Já

Tester: Ég sé engan mun á því persónulega.

Kristján: Hversu viss ertu að það var tölvan sem þú varst að tala við ?

Tester: Viss (vissi af þessu)

Kristján: Er eitthvað sem þú vilt bæta við sem hefur ekki komið fram hér á undan ?

Tester: Heyrðu jú, það kom fram þegar ég var að spila. ég reyndi að challenge conditional og þá kom "not implemented" ég gat valið

Kristján: OK

Tester: Ég gat valið ef ég ætlaði að "Why is that the case", hægra meginn úr glugga tölvunar get ég valið fleira en eitt, það var bara óvart og það kemur gluggi að það er not implemented. Og það kom hvergi fram í kynningunni að þetta væri í þróun og

Kristján: Já, heyrðu þá er þetta búið ég vill þakka þér fyrir.

Test 5.

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Hefur þú notað samskiptaforritið áður svona til að tala við einhvern ?

Tester: Nei

Kristján: Svona MSN og svoleiðis ?

Tester: Nei

Kristján: Í byrjun kom gluggi svona upplýsinga gluggi, hvernig finnst þér að fá svona glugga í byrjun, þegar þú ert að prófa nýtt forrit ?

Tester: Hundleiðinlegt

Kristján: OK, gott, er hann bara fyrir þér ?

Tester: Eitthvað sem ég ekki nota

Kristján: Já, skoðaðir þú gluggan núna ?

Tester: Ég svona bara aðeins renndi yfir hann fljótlega og sá að þetta var eitthvað sem ég nennti ekki að lesa

Kristján: Hvað gerðir þú við gluggan eftir að þú .. ?

Tester: Setti hann bara niður

Kristján: Skoðaðir þú eitthvað gluggann aftur, fórstu eitthvað aftur í hann ?

Tester: Nei

Kristján: Ekkert þannig, eeee, þannig að hvernig gekk að byrja leikinn, starta honum ?

Tester: Það tók mig svona 10 sec, 20 sec, ég var ekki allveg að kveikja

Kristján: OK, en tókstu eftir blikkandi textanum þarna niðri

Tester: (hlátur) það var ekki fyrr en ég tók eftir honum

Kristján: En hvernig fannst þér, núna bjóstu til setningar úr þessum 2 gluggum þarna niðri, hvernig fannst þér að staðfesta val þitt með því að íta á takka þarna ?

Tester: Go, hann má vera meira eitthvað svona center, og litill og of langt frá

Kristján: En svona almennt um textan sem blikkar þarna neðst á skjánum, hjálpaði hann þér eitthvað meirna en í byrjun ?

Tester: já hann hjálpaði mér

Kristján: Fannst þér böggandi að hann blikkaði

Tester: Nei

Kristján: hvernig gekk að átta sig á þessum 3 aðalgluggum, hver táknað hvað og ?

Tester: Já, ég hafði ekki hugmynd þarna fyrst og vissi heldur ekkrert úti hvað ég var að fara, en ég var fljótur að fatta það út á hvað þetta gengi. Mér fannst reyndar galla að þessi gluggi fylgir ekki hér [bendir á history] þegar þú ert kominn niður hættir þú bara að sjá

Kristján: Já hann scrollar ekki með ?

Tester: já scrollar ekki með

Kristján: Hausarnir á hverjum glugga og svona skýrðu þeri út fyrir þér hvað þeir gera

Tester: Já já

Kristján: Hvernig líkar þér að velja svona, núna varstu að tala við tölvuna og input þitt það sem þú segir úr 2 dropdown boxum, að tala þannig ?

Tester: Það vara bara allt í lagi, en ég var ekki búinn að kveikja nógum snemma að það væru fleiri möguleikar [scroll á dropdown]

Kristján: Já ok, þá er næsta spurning um möguleikana, fannst þer þeir of margir eða of fáir ?

Tester: Þeir eru ekki sko, ég hugsa ef maður er búinn að vera í þessu í smá tíma að þá væru þeir of fáir.

Kristján: mindir þú vilja getað skrifað setningarnar sjálfur ?

Tester: Ég myndi vilja hafa þann möguleika.

Kristján: En t.d. ef petta væri á íslensku, og þú hefðir þann möguleika að nota dropdown eða skrifa sjálfur ?

Tester: Ef ég þyrfti að velja á milli, þá myndi ég vilja skrifa sjálfur

Kristján: Hvernig fannst þér aðgangur að valmöguleikum og upplýsingum, settings og stillingar, skoðaðir þú það eitthvað ?

Tester: Ég skoðaði það ekki

Kristján: Var eitthvað sem böggði, fannst gott þig varðandi viðmótið ?

Tester: Það eina sem böggði mig var hérna Go takkinn. Og hérna það svona, böggði það mig aðeins, mér fannst fint þarna græni,

Kristján: Já fannst þér þágilegt að nota hann.

Tester: ef ég er að velja eitthvað hérna, þá kemur hann og þú verður að velja úr þessu hérna.

Kristján: hann hjálpar þér

Tester: Já

Kristján: Uuu, hversu hratt fannst þér tölvun svara þér ?

Tester: Mjög hratt

Kristján: lentir þú í því að bíða ?

Tester: nei

Kristján: hvernig fannst þér hún svara þér, spyrja þig, sovna gáfulega séð ?

Tester: Mér fannst hún þokkalega gáfuð nema þegar hún kom með sko eitthvað, ef þú hefur eitthvað meira að segja.

Kristján: Já, hvernig leið þér, þegar þú fékkst þá setningu ?

Tester: Ég fekk hana upp 2 eða 3. Þá var ég einhvernveginn.. Þá fannst mér bara vanta, erum við þá bara sammála

Kristján: umhum fannst þér það vera diss, upplifðir þú það þannig ?

Tester: nei ekkert diss, nei bara eithvað svo, hún var ekkert að halda áfram að ögram mér, eða koma með statement svo að þú purfir að svara því

Kristján: ok, þannig að hún beið eftir að þú mundir sækja.

Tester: Já já

Kristján: En hvernig likar þér við þetta forrit ?

Tester: Bara fínt, ekkert út á það að setja

Kristján: Hafðir þú gaman að rökræða við tölvuna ?

Tester: Já já

Kristján: Hvað var svona það helsta sem gaman var að, bara að rökræða ?

Tester: Maður er að prófa eitthvað nýtt, þetta var ekkert langur tími, ég get allveg ímyndað mér að sitja við þetta í langan tíma.

Kristján: Hvernig er eins og með umhverfið, finnst þér það skemmtilegt ?

Tester: Nei, þetta er ekkert aðlagandi

Kristján: en svona hvernig myndir þú lísa því ?

Tester: Þetta er allveg eins og þú sagðir, þetta er svona prototype. Þetta er ekkert sérstaklega aðlagandi.

Kristján: En svona eftir frekari þróun, gætir þú hugsað þér að nota svona kerfi þér til gamans ?

Tester: Já, ég hugsa að ég myndi aldrei nota það mikið, en hérrna svona til þess að ræða við tölvuna, aðeins droppa inn á það.

Kristján: En telur þú að hægt sé að nota það til gagns t.d. í skólakerfinu ?

Tester: Skil ekki

Kristján: T.d. sérðu möguleika fyrir krakka og unglings, ekki endilega þetta umhverfi, svo þau hafa gagn af því ?

Tester: Já meinar það, já sko, það væri hægt að koma inn ákvæðnum gildum hjá krökkum. En þá þýðir ekkert að hún sé á móti, að maður getur valið það

Kristján: Já

Tester: Þá verður tölvan að vera búinn að velja sitt gildi

Kristján: Já, fyrirfram

Tester: Já, þannig að ef þú myndir segja er rétt að lemmja skólabróður þinn þá mundi ekkert tölvan, þá myndi tölvan segja nei, en ef þú myndir segja að það sé ekki rétt þá víst.

Kristján: Skil hvort þú ert að fara, að reyna að koma því góða inn.

Tester: Já þaning að já, ef það ætti að nota það í þeim tilgangi.

Kristján: Nú er bara eitt umræðuefni, dauðarefsingar, værir þú líklegrí til að nota kerfið ef þú gætir valið úr nokkrum umræðuefni?

Tester: Ef ég fengi valið um ?

Kristján: Já ef það kæmi upp nokkur í byrjun 10 eða 15 til að velja úr

Tester: Værí ég líklegrí til að nota það ?

Kristján: Já,

Tester: Já ég myndi vera líklegrí til að nota það, maður myndi gefast fljótt upp á því að tala alltaf um sama hlutinn.

Kristján: Nákvæmlega, en myndir þú hafa áhuga að bæta sjálfur inn umræðuefnum og vera meira sovna sjórnandi ?

Tester: Já,

Kristján: Já

Tester: Ef þetta væri bara svona online og ég gæti bara komið með hvaða umræðuefnið sem er

Kristján: Já og þú myndir velja umræðuefnið og aðrir sjá um að græja það

Tester: Já einmitt

Kristján: Uu, ef allt kerfið væri á íslensku værir þú líklegri til að nota það á íslensku frekar en ensku ?

Tester: Já

Kristján: Og hérna, þá er það næst síðasta spurningin. Hversu viss ertu að það hafi verið tölva sem þú varst að tala við ekki einhver annar í næsta herbergi, yfir netið ?

Tester: Ég veit ekkert um það

Kristján: Hver er svona fílingurinn, hvort heldur þú að það hafi verið tölvan eða ég á móti þér ?

Tester: Ég held nú frekar að þetta hafi verið tölvan.

Kristján: Hvað svona byggir þú á því ?

Tester: Ekki neitt, Hún var mjög fljót að svara, en ég segji það ekki að ef maður sem væri vanur að nota þetta forrit gæti verið mjög fljótur að svara líka

Kristján: Já

Tester: En ég meina, ég held að þú værir bara miðað við þetta að velja hér og velja s vona annarstaðar og íta svo á go þú nærd því ekki, og því held ég að þetta hafi verið tölvan.

Kristján: Er eitthvað sem þú vilt bæta við sem hefur ekki komið fram hér á undan, hversu gáguleg talvan er, viðmótið eða bara hvað sem er ?

Tester: Mér fannst hún allveg meika sens, þangað til að hún fór að koma með þarna, ef þú vilt bæta einhverju við

Kristján: Já

Tester: Það fannst mér, það var ekki til þess að halda þér í forritinu. Ef ég væri að selja þetta forrit, þá vill ég að fólk sé í því helst sem lengst

Kristján: umhummm

Tester: nota það nógum mikið, þegar kemur upp þetta, þá sko það er eitthvað

Kristján: Það var þetta og svo go takkinn sem var.

Tester: Já

Kristján: Heryðu þá þakka ég þér fyrir

Test 6

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Hefur þú notað samskipta forrit áður, að einhverjum hætti ?

Tester: Er það MSN og svoleiðis ?

Kristján: Já

Tester: Já

Kristján: Sko í byrjun kom upplýsinga gluggi, hvernig finnst þér að fá svo upplýsinga glugga í byrjun þegar þú opnar forrit ?

Tester: Það var mjög fint

Kristján: Hvernig þá ?

Tester: Bara líka glugginn neðst niðri [á við blikkandi] hann útskýrði, ég hélt að það ætti að byrja strax, en það stóð þarna fara í new game

Kristján: Já

Tester: Ég svo las ekki þennan upplýsinga glugga

Kristján: Skoðaðir þú hann eitthvað

Tester: gerði bara close

Kristján: Þú lokaðir honum bara strax

Tester: Já

Kristján: Purfitr þú eihvertiman, vantaði þér eihvertiman hjálp þannig að þú þurfir að fara í gluggann aftur þ.e. upplýsinga gluggann.

Tester: Nei, hefði átt að gera það því ég sá ekki alla spurninguna.

Kristján: Hvernig gekk að byrja í leiknum, að strarta honum ?

Tester: Það var ekkert mál,

Kristján: Já

Tester: eftir að ég byrjaði þá virkaði ekkiert, þá las ég þarna niðri að maður átti að fara í new game

Kristján: Á blikkandi textanum

Tester: já, það var ekkert mál

Kristján: þannig að hann hjálpaði þér. Nú býrðu til setningar úr 2 dropdown boxum hvernig fannst þér að staðfesta val þitt með því að íta á takka

Tester: Það var bara mjög, það var ekkert vandamál

Kristján: Fannst þér bara fint að gera það ?

Tester: Já

Kristján: OK, varðand blikkandi textan parna í byrjun

Tester: já

Kristján: hjálaði hann seinna í leiknum ?

Tester: já, kom ekki þarna ef maður vara að spryrja sömu spurninguna aftur þegar maður var búinn að fá tölvuna til að snúa skoðunn sinni.

Kristján: Já kom það í blikkandi textanum

Tester: Já mig minnir það, eða koma það þarna fyrir ofan.

Kristján: En fannst pér eitthvað böggandi að hann blikkaði ?

Tester: Nei nei.

Kristján: já, hvernig gekk að átta sig á þessum 3 aðalgluggum ?

Tester: Það var ekki mál

Kristján: Þannig að þú veist allveg hvað hver táknaði hvað og svona ?

Tester: Ég er með mínn svör einum megin og hanns einu megin og svo dregst allt saman í miðjuna, það var mjög auðvelt

Kristján: Hver hlutur hefur haus, sem á að skýra út hvað hann gerir, tókstu eftir því ?

Tester: Uuu skil ekki hvað þú ert að meina.

Kristján: hvernig gekk að velja setningar úr svona 2 dropdown boxum ?

Tester: Það var ekki vandamál, en einhverju vegna áttaði ég mig ekki á því að scrolla niður til þess að velja fleiri möguleika, vara bara með einhverja 6 möguleika.

Kristján: Já, fannst pér það of mikið eða lítið þessir 6 ?

Tester: Mjög lítið

Kristján: Í restina sástu alla möguleikana, og telur þú það vera í lagi

Tester: Þá hefði ég náð að rökræða að meira viti við tölvuna.

Kristján: Myndir þú frekar vilja skrifa setningarnar bara sjálfur ?

Tester: nei ekki endilega, þetta eru logicar spurnignar

Kristján: En ef þetta væri á íslensku myndir þú frekar vilja skrifa þá sjálfur ?

Tester: nei, bara velja

Kristján: Dettur pér eitthvað í hug, nú kemur þú þinum á framfæri með því að velja úr þessum 2 dropdown, er einhver annar valmöguleiki sem pér dettur í hug, annar en að skrifa sjálfur og dropdown ?

Tester: Nei, ekki svona í fljótubragði

Kristján: En hvernig fannst pér að gangur að valmöguleikum og upplýsingum, þú lokaðir upplýsinga glugganum strax og náðir aldrei í hann aftur. Skoðaðir þú svona stillingar ?

Tester: Nei, ég var bara að debata við tölvuna, ég les aldrei upplýsingar.

Kristján: Er eitthvað sem þú tókst eftir sem ekki hefur komið fram varðandi viðmót, eitthvað sem böggæði þig eða fannst gott ? við erum búinn að tala um blikkandi textan hann var góður, eitthvað sem böggar þig ?

Tester: Alltaf leiðinlegt að það sé ekki hægt að setja saman spurningar, þar sem tölvun tekur ekki við, svo þarf maður alltaf að velja frá honum.

Kristján: Já já

Tester: (talar um að gott væri að hafa bara neitun, svona negation á eitthvað sem tölvun segir)

Kristján: Hversu hratt fannst þér tölvan svara þér ?

Tester: Bara mjög hratt

Kristján: lentir þú aldrei í því að býða ?

Tester: Nei

Kristján: En hversu gáfulega fannst þér tölvan svara þér ?

Tester: Já, þegar maður var farinn að bulla bara eitthvað þá svaraði hún manni bara, if you have anything more to say you can go on, það sem búið var að ræða, og það var ekkert við í því þá sagði hún manni bara að reina eitthvað annað.

Kristján: En náði hún að halda upp samræðum við þig allan tíman ?

Tester: Já, svo náði ég að snúa henni í einum lið, og þá gat maður ekkert spurt meira út í það, því maður var bara búinn að vinnna það.

Kristján: Hvernig líkar þér við forritið í heild sinni ?

Tester: Bara mjög vel.

Kristján: Hafðir þú gaman að spjalla við tölvuna ?

Tester: Það væri hægt að snúa þessu upp í marga hluti. Setja inn, setja upplýsingar inn í þetta og snúa upp í marga hluti, rökfærðin mjög skemmtileg. Hægt að snúa þessu upp í íslensku, pannig að rökræða t.d. með eða á móti kárahnjúkavirkun,

Kristján: Fannst þér umhverfið skemmtilegt ?

Tester: Já, ekkert of spaceað, miklu betra að hafa þetta simple.

Kristján: pannig að þú hafðir gaman af því að nota kerfið ?

Tester: Já

Kristján: Þú talaðir um að það er hægt að breyta eitthvað, telur þú að eftir þróun og breytingar að það sé hægt að nota þetta til einhvers gagns ?

Tester: Já það væri allveg hægt að nota þetta sem kennslutæki, t.d. ég er á viðskiptabraut og þar er áfangi sem heitir rökfræði, það væri hægt að nota það í því

Kristján: Já, sérðu möguleika á að breyta kerfinu pannig að það sé hægt að nota það ekki bara til skemmtunar ?

Tester: Til menntunaruna

Kristján: Já, en hefur þú hugmyndi hvernig hægt sé að breyta ?

Tester: Já, nei í rauninni ekki, ekki nema með meira efni, fleiri umræðuefnum.

Kristján: Já það er næsta spurning, það er bara eitt umræðuefni núna, værir þú liklegri til að nota kerfið ef þú gætir valið úr mörgum umræðuefnum ?

Tester: Já ég mundi segja það

Kristján: Telur þú að það mundi auka skemmtunargildi ?

Tester: Rifist um hvor er betri Carragher eða Gerard

Kristján: T.d. hefur þú áhuga að setja inn umræðuefni, þannig að þú hafðir meiri sjórn yfir kerfinu ?

Tester: Já, ég efast um að ég myndi hafa þolinmæði að búa til svörin

Kristján: Þú vilt bara fá þetta tilbúið og já ?

Tester: Já, ég er latur

Kristján: Ef kerfið væri á íslensku væri þú líklegri til að nota það ?

Tester: Ekkert endilega, jú það væri auðveldara að nota það, það voru sum orð þarna sem maður þurfti að fara í orðabókina.

Kristján: Ef við gefum okkur það að þú hefðir valmöguleika hvort að þú viljir hafa á íslensku eða ensku í byrjun ?

Tester: Allveg í heildina litið þá held ég að það væri betra að hafa það á ensku upp á notenda

Kristján: Þá er það næst síðasta spurningin, hversu viss ertu að það hafi verið tölvva sem þú varst að tala við ekki einhver annar í öðru herbergi í húsinu ?

Tester: Ég er ekkert svo viss um það, ég trúi þér til alls, en ég ætla að trúá því að petta hafi verið talva

Kristján: Er eitthvað sem þú vilt bæta við sem hefur ekki komið fram um þetta, hvernig í raun og veru, það sem ég er að fiska eftir hversu gáfluleg talvan er, hvernig umhverið er o.s.frv.

Tester: Ég veit ekki hvað maður á að segja

Kristján: Pannig að þú telur að þú sért búinn að koma þínu á framfæri ?

Tester: Já er það ekki

Kristján: Ok, takk fyrir

Test 7

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Hefur þú notað samskipta forrit áður, ekki enilega við tölvu ?

Tester: já já MSN

Kristján: OK, í byrjun kom upplýsinga gulaggi, hvernig fannst þér að fá svona glugga ?

Tester: Allveg nauðsynlegt, ég vissi ekki neitt hvað ég var að fara í

Kristján: Skýrðu þær nájilega fyrir þér ?

Tester: Já

Kristján: Ég er að spá í magni af upplýsingum, voru allar upplýsignarnar sem skiptu máli.

Tester: Sko upplýsingarnar sem ég átti að svara ?

Kristján: í upplýsinga glugganum

Tester: Já ok, mér fannst þær vera nokkuð góðar

Kristján: Ef þú myndir fara að nota forritð aftur, myndir þú fara aftur í gegnum þennan glugga, upplýsinga gluggann

Tester: nei ég myndi ekki gera það.

Kristján: bara svona í fyrsta skipti, Myndi það eitthvað pirra þig ef hann byrtist alltaf fyrst ?

Tester: Nei ekki ef ég gæti bara itt á close.

Kristján: En hvað gerðir þú við gluggann eftir að þú varst búinn að lesa hann ?

Tester: Bara close

Kristján: Purftir þú að skoða gluggann aftur ?

Tester: Nei

Kristján: Vildir..

Tester: Ég vara fann ekki þörf fyrir að skoða

Kristján: Er eitthvað athugavert, eitthvað sem angrar þig við upplýsinga gluggann ?

Tester: Nee nei.

Kristján: En hérna hvernig gekk svo að byrja leikinn, að starta ?

Tester: Það gekk bara mjög vel, bara út af upplýsinga glugganum, hann sagði manni að það væri blikkandi ljós sem sagði mann hvað maður þurfti að gera í raun og veru.

Kristján: Já, flott. Hvernig fannst þér að staðfesta hvert val með því að íta á takka, þú varst búinn að velja úr dropdown boxunum ?

Tester: Bara mjög gott

Kristjan: var eitthvað sem böggði þig við það ?

Tester: Nei alls ekki.

Kristján: Þú talaðir um blikkandi textan þarna neðst á skjánum, hjálpaði hann þér í einhverju öðru en að starta leiknum ?

Tester: Já bara, já já, og nátúrulega líka. Ég átti að velja statementið og svo statementið.

Kristján: Fannst þér eitthvað böggandi að hann blikkaði

Tester: nei alls ekki.

Kristján: Hvernig gekk að átta sig á þessum 3 aðalgluggum, hver táknað hvað ?

Tester: Það bara gekk mjög vel. Það var ég, tölvan og samtalið.

Kristján: Það hefur ekkert verið að rugla þig ?

Tester: Nei allst ekki, fljótur að ná því.

Kristján: Já, hver gluggi og dropdown boxinn hafa haus hvað þeir heita hvað þeir gera. Hjálpaði það þér eitthvað ?

Tester: Já já mjög mikil.

Kristján: Hverníg líkar þér að velja setningarnar úr svona 2 dropdown boxum ?
Tester: Mér fannst það mjög fint, mætti hafa fleiri möguleika og tengja ekki bara
dauðarefsingar bara við morðingja [ATH hann scrollaði ekki niður]

Kristján: Þú vilt frá fleiri möguleika. Myndir þú vilja skrifa sjálfur setningarnar ?

Tester: Já ef talvan myndi skilja hvað ég væri að segja.

Kristján: Ef petta væri á íslensku, myndir þú frekar vilja skrifa sjálfur ?

Tester: Skrifa sjálfur

Kristján: Hérna,

Tester: Ef það væru fleiri valmöguleikar þá væri það allt í lagi sko.

Kristján: Ef þú stæðir frami fyrir vali um að hvora aðferðina að velja. Dropdown með
hellings setningum eða skrifa sjálfur.

Tester: Skrifa sjálfur

Kristján: Hverníg fannst þér aðgangur að valmöguleikum stillingum ? Skoðaðir þú það ?

Tester: Nei ég skoðaði það ekki

Kristján: Var eitthvað sem hefur ekki komið, sem við höfum ekki talað um varðandi
viðmótið ? GUI-ið

Tester: Viðmótið ?

Kristján: gluggana og það dót, eitthvað sem angaraði þig ?

Tester: nei alls ekki.

Kristján: Hversu hratt fannst þér tölvun svara þér ?

Tester: mjög hratt

Kristján: Þú lendir aldrei í því að biða ?

Tester: nei

Kristján: En hversu fannst þér hún svara þér svona gáfulega séð ?

Tester: Hún var mjög sniðug að, ég skildi ekki eina setningu sem ég sagði þarna og
svaraði öfugt sem ég myndi gera. vanalega og síðan héldum við áfram að rökræða og svo
skelti hún fram í mig því sem ég hafði svarað þarna áður. Þannig að mér fannst þetta
bara mjög sniðugt.

Kristján: Hún náði að hala upp samræðum við þig allan timan ?

Tester: Já, þangað til í lokinn þá fór hún að segja við mig að ef ég hefði eitthvað
meira að segja ætti ég bara að halda áfram

Kristján: Myndir þú segja að hún hafi talað við þig gáfulega ?

Tester: Já hún talaði skynsamlega, þetta var allveg rökrétt.

Kristján: Já nákvæmlega. En hverníg líkar þér svona almennt við forritið ?

Tester: Bara, þetta lítur mjög vel út

Kristján: Hafir þú gaman af því að spjalla við tölvuna ?

Tester: já já

Kristján: Uuu, hvað svona helst hafðir þú gaman af ?

Tester: bara að talvan hafi, mér fannst fint að rökræða við tölvu því maður veit að hún á að hafa endalaust af möguleikum.

Kristján: En umhverið sem þú varst að vinna í, myndir þú segja að það sé skemmtilegt ?

Tester: Nei nei, bara venjulegt, skrautlegt að sjá svona rauða, græna blikkandi kerfi.

Kristján: Myndir þú geta hugsað þér að nota svona kerfi eftir einhverja þróun þér til gamans ?

Tester: Já

Kristján: En telur þú að það sé hægt að nota svona kerfi til gagns ekki einungis gamans.

Tester: Já

Kristján: Hvernig sérðu það, hvaða möguleika ?

Tester: Ef einhver hefur meiri upplýsignar en maður sjálfur, þá getur hann kennt manni.

Kristján: Góður, nú var bara eitt umræðuefni dauðarefsingar. Værir þú liklegri til að nota kerfið ef hægt væri að velja úr nokkrum umræðuefnunum ?

Tester: já já, sérstaklega ef það væri bara til gamans.

Kristján: Værir þú til í að geta bætt inn sjálfur umræðuefnum og fá meiri stjórn yfir kerfinu ?

Tester: já, en þyrfti ég þá ekki að mata tölvuna sjálfur upplýsingum lika. Já en. ?

Kristján: jú

Tester: En þá væri það nokkuð einhæft þá er ég í raun að rökræða við sjálfan mig.

Kristján: Ef kerfið væri á íslensku algerlega, væri líkurnar að þú mundir nota það aukast ?

Tester: já já, það væri mun auðveldara í notkunn en í ensku, því sum af þessum háfleigum orðum í ensku nær maður ekki allveg að setja í samhengi.

Kristján: Jæja næst síðasta spurning. Hversu viss ertu að það hafi verið tölva sem þú varst að tala við, ekki ég í næsta herbergi ?

Tester: Ég er mjög viss um það

Kristján: Að það hafi verið ?

Tester: Að það hafi verið talva. Mér fannst hún, petta var svo lítið stirt þarna í lokinn. Ef þú hafir verið hinum megin þá hafðir þú eflaust skrifað eitthvað, skiluru og við haldið áfram. Ég stoppaði þarna aðeins og hún skaut einhverju á mig og ég svaraði og hún sagði ef ég hefði einhvað meria að segja og svo svaraði ég einhverju öðru og þá kom hún með það sama. Mér fannst petta tölvu legt.

Kristján: Er eitthvað meria sem þú vilt koma á framfæri sem hefur ekki komið fram ?

Tester: Fleiri svarmöguleika, ekki tengja bara við morðingja og fleiri valmöguleika[á við umræðuefni]

Test 8

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Hefur þú notað samskipta forrit áður, ekki einungis við tölvu ?

Tester: MSN og svoleiðis ?

Kristján: Já t.d.

Tester: Já ég hef notað MSN

Kristján: Í byrjun kom upplýsninga gluggi, hvernig fannst þér að fá svona glugga í byrjun ?

Tester: bara mjög gott

Kristján: Já, og hérna fannst þér hann skýra út fyrir þér ?

Tester: Já

Kristján: fannst þér upplýsingarnar skýra næjilega út fyrir þér hvernig þú áttir að gera hlutina ?

Tester: Já mér fannst það

Kristján: Ef þú myndir nota forritið aftur, myndiru nota gluggann aftur ?

Tester: Nei, ég held ekki

Kristján: Petta er bara einu sinni og búið ?

Tester: Já, já

Kristján: En þegar þú varst búinn að lesa gluggann, hvað gerðir þú við hann ?

Tester: Ég lokaði honum.

Kristján: Lokaðir honum ?

Tester: Já,

Kristján: Skoðaðir þú gluggann aftur eftir að þú varst búinn að loka honum ?

Tester: nei

Kristján: Er eitthvað athugavert við gluggann, eitthvað sem böggði þig sem hefur ekki komið fram ?

Tester: Nei, nei

Kristján: En hvernig gekk að starta leiknum ?

Tester: Það gekk bara vel

Kristján: Tókstu eftir blikkandi textanum þarna neðst ?

Tester: já já, ég tók eftir honum, það kom mér

Kristján: Hjálpaði það þér ?

Tester: Já

Kristján: Hvernig fannst þér að staðfesta val í hvert skipti með því að íta á takka, parna go takkanum ?

Kristjan: Var það eitthvað að böggja þig ?

Tester: Nei nei

Kristján: Hjálpaði textinn sem blikkaði þér svona alment í leiknum ?

Tester: Já, bara mjög mikið

Kristján: Ok, þá er það næsta. Hvernig gekk að átta sig á þessum 3 aðalgluggum, hafðir parna 3 glugga. ? Ruglaði það þig eitthvað ?

Tester: Já dálitið.

Kristján: Hvað var það sem þér þótti ruglingslegt við þá.

Tester: Neðri glugginn stóð alltaf, ég var alltaf að horfa á það sama.

Kristján: Já, scrollaðist ekki niður

Tester: Já já, ég varð alltaf að skrolla sjálfur, ég horfði meira á það en gluggana að ofan.

Kristján: Ok, og hérna, hver gluggi hafði haus og dropdown boxinn líka, skýrði það út hvað þeir gerðu ?

Tester: Já, þeir skýrðu út hvað þeir gerðu

Kristján: hvenrig líkar þér að velja setningar, þú valdir setningar úr 2 hlutum, dropdown boxum, hvernig líkaði þér það ?

Tester: Það var svona þínu flókið

Kristján: Já, fannst þér valmöguleikarnir of margir eða fáir

Tester: Það mátti vera, ég var með, kannski eru þeir jafn margir.

Kristján: Þú scrollaðir niður sjálfur

Tester: Já já ég scrollaði niður

Kristján: En heldur þú að þú viljir skrifa setningarnar bara sjálfur ?

Tester: Nei, ég held ekki

Kristján: En ef petta væri á íslensku, mundir þú frekar vilja skrifa þá ?

Tester: Já kannski þá

Kristján: Segum að petta væri á íslensku og þú hefðir valmöguleika í byrjun, hvernig þú vilt tala. Úr þessum boxum eða skrifa sjálfur ?

Tester: Skrifa sjálfur

Kristján: En hvernig fannst þér aðgangur að valmöguleikum, stillingum ? Fórstu eitthvað að fikta í því ?

Tester: nei ég gerði það nú ekki

Kristján: En er eitthvað sem þú tókst eftir sem þér finnst athugavert varðandi

viðmótið, umhverið ?

Tester: Ekki neitt sérstaklega

Kristján: Þú minntist á scrollið, eitthvað fleira ?

Tester: Aðalega það, að fylgjast með þessum 3, það ruglaði mig, því ég var aðalega að fylgjast með þessum neðri.

Kristján: Fannst þér fara mikil athygli í það að fylgjast með hvar umræðan færi fram ?

Tester: Mér fannst það,

Kristján: Hveru hratt fannst þér tölvan svara ?

Tester: mjög hratt

Kristján: Lendir þú í því að bíða ?

Tester: nei

Kristján: Hvernig fannst þér tölvan svara, spyrja þig, þá er ég að fiska eftir gáfulega séð ?

Tester: bara nokkuð gáfulega

Kristján: Já, hún náði að halda uppi samræðum við þig allan tíman ?

Tester: já

Kristján: Leið þér eins og þú værir að tala við tölvu eða vin þinn, svona gáfulega séð ?

Tester: Petta var á rökum reist.

Kristján: Já, hvernig líkar þér við forritið svona alment ?

Tester: bara ágætlega

Kristján: Hafið þú gaman að því að spjalla við tölvuna ?

Tester: Já, kannski bara ef það væri annað debate, þá kannski, en jú jú

Kristján: En umhverið var það að heilla þig ?

Tester: Það var bara nokkuð skýrt

Kristján: En ef ég myndi segja orðið skemmtilegt ?

Tester: Ég veit það ekki hvort ég myndi nota það orð, en allavega aðgenileft

Kristján: Myndir þú vilja nota svona, eftir frekari þróun og lagfæringar, myndir þú vilja nota svona kerfi þér til gamans.

Tester: Já já !!

Kristján: En telur þú að hægt sé að nota svona kerfi að einhverju gagni, að einhver læri á því ?

Tester: Já jafnvel.

Kristján: já, skal ég segja þér, við töluðum um að það væri einungis eitt umræðuefn, dauðarefsingar. Myndir þú vera líklegri til að nota kerfið ef það væri fleiri umræðuefni ?

Tester: Já ! vissulega

Kristján: En hefir þú áhuga á að bæta inn umræðuefní sjálfur og svona fá meiri sjórn yfir því ?

Tester: Já !

Kristján: Ef kerfið væri alfarið á íslesku værir þú líklegri til að nota það ?

Tester: já

Kristján: var enskan mikið að trufla þig ?

Tester: örlitið jú

Kristján: Heyrðu þá er það næst síðasta spurningin. Hversu viss ertu að það hafi verið tölva sem þú varst að tala við, ekki ég hérna í öðru herbergi á hinum endanum ?

Tester: Nokkuð viss (hlátur)

Kristján: já ertu nokkuð viss ? Afhverju ?

Tester:(þögn)

Kristján: Fannst þér þetta tölvulega svarað ?

Tester: Já kannski, ég myndi segja það.

Kristján: Er eitthvað sem þú myndir vilja bæta við sem ekki hefur komið fram, sko eithvað um kerfið einhverjir hnógrar ?

Tester: Ekkert sem böggði mig neitt sérstaklega nema þarna

Kristján: þetta scroll

Tester: já, að fylgjast með hvað var, þegar komið var mikið þá var maður allveg orðinn lost bara.

Kristján: Já, þegar þú talar um að svörin hafa verið tölvuleg þá, ertu að meina formlega eða ?

Tester: Já eitthvað í þá áttina

Kristján: já maður vill hafa þetta

Tester: já einmitt í þá áttina

Kristján: Ok, þá pakka ég þér bara fyrir.

Tester: OK

Test 9

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Hefur þú notað samskipta forrit áður ?

Tester: Já

Kristján: Nefndu mér dæmi ?

Tester: Ég hef notað Yahoo messenger, MSN messenger og IPuls.

Kristján: Hvernig finnst þér þetta uuuu í samanburði, þarna hafðir þú samskipti við tölvu. umhverfið aðalega, getur þú gefið mér einvhern samanburð ?

Tester: Mér fannst aðalega ruglingslegt að sjá þessi statement box sitt hvoru megin. Annars fannst mér þetta allt í lagi.

Kristján: Í byrjun kom upplýsinga gluggi, hvernig finnst þér að fá svona upplýsingar í byrjun ?

Tester: Fint

Kristján: Hvenrig þá ?

Tester: Þá er bara þæginlegra að átta sig á forritinu

Kristján: Já, uuu hvernig fannst þér upplýsingarnar, skýrðu þær nægilega út fyrir þér ?

Tester: Já að mestu leyti

Kristján: Fannst þér þetta of mikið af upplýsingum

Tester: Nei mér fannst þetta bara nokkuð nett

Kristján: já, ef þú myndir nota forritið aftur, myndir þú lesa aftur yfir hann ?

Tester: Nei

Kristján: Ertu bara one timer, ?

Tester: Ha

Kristján: Notar þú bara svona upplýsignar einu sinni ?

Tester: Já

Kristján: Ok, ef glugginn myndi alltaf byrtast á toppnum, myndi það böggva þig ?

Tester: Já, ég myndi vilja hafa svona option

Kristján: Já

Tester: um að byrta hann

Kristján: Lastu hann allan ?

Tester: Ég las hann allan

Kristján: Og hvað gerðiri þú við hann

Tester: Ég held að ég hefði slökt á honum

Kristján: Lokaðir honum allveg?

Tester: Já

Kristján: Purftir þú að skoða hann aftur ?

Tester: Nei

Kristján: Er eitthvað athugavert við gluggan, sem þér fannst, eitthvað sem var að angra þig ?

Tester: Mér fannst hann kannskíiii, mætti að vera aðeins breiðari, svo að mðaur þyrfti ekki að scrolla eins mikið í honum.

Kristján: Hvernig gekk að byrja leikinn, starta ?

Tester: Það gekk nokkuð vel, en mér fannst svo óþágilegt að fara lengst niður í hornið og ita alltaf á go.

Kristján: Tókstu eftir blikkandi textanum sem var neðst á skjánum þegar þú varst að starta ?

Tester: Já

Kristján: Hjálpaði hann þér ?

Tester: Hann sagði mér hvað ég átti að gera já

Kristján: Fórstu bara eftir honum og startaðir þú leiknum ?

Tester: Já

Kristján: OK , það er einmitt næsta spurning með go takkan, hvernig finnst þér að stafesta hvert val með því að ita á go takkan i hvert skipti?

Tester: Mér finnst það óþolandí

Kristján: Óþolandí, er það þá bara staðsetningin ?

Tester: Já ég held það, þetta er svon 3 skipti glugga umhverfi, hann mætti allveg vera í miðjunni. Þessir valmöguleikar mættu allveg vera í miðjunni kannski

Kristján: Skil þig, þessi blikkandi texti, hjálpaði hann þér eitthvað meira en að starta ?

Tester: Nei, tók lítið eftir honum eftir start

Kristján: Hvernig gekk að átta sig á þessum 3 aðalgluggum, hver táknað hvað ?

Tester: Ágætlega, en mér fannst ruglingslegt þessar merkingar við setningarnar

Kristján: Stjarnan ?

Tester: Stjarnan

Kristján: Hver af þessum hlutum dropdown boxinn, gluggarnir höfðu haus, skýrðu þeir eitthvað út fyrir þér hvað þeir gerðu?

Tester: Tók ekki eftir því

Kristján: Hvernig líkaði þér að velja setningar úr dropdown boxum tveimur ?

Tester: Mér fannst of lítið af möguleikum [ATH scrollaði ekki niður]

Kristján: Þegar þú íttir á dropdown boxinn

Tester: Já

Kristján: Scrollaðir þú niður ?

Tester: Já það var bara 6-7-8 möguleikar

Kristján: OK, hérna, myndir þu vilja sjálfur skrifa setningarnar sjálfur inn,

hafa svona inputbox og skrifa ?

Tester: Það mætti vera möguleiki

Kristján: En ef þetta væri á íslensku, myndir þú frekar vilja skrifa setningarnar sjálfur ?.

Tester: já örugglega

Kristján: Myndi það skipta máli, íslensku eða ensku ?

Tester: Það væri ábyggilega auðveldara á íslensku

Kristján: Er einhver annar möguleiki sem þér dettur í hug til að koma þinu á framfæri.
Annað en dropdown eða skrifa sjálfur ?

Tester: Það mætti, kannski út af því að þetta eru rök uppbyggingar á setningu þá mætti
kannski vera einhver orð sem þú velur í byrjun sem leiða af sér, svo velur þú næstu orð
í setningunnini, væri byrjunar rök og þá færðu mengi af orðum [i.e. iPod dæmið]

Kristján: Hvernig fannst þér aðgangur að valmöguleikum og upplýsingum,
skoðaðir þú það eitthvað ?

Tester: nei, skoðaði það ekki

Kristján: Er eitthvað sem þú tókst eftir sem þér þykir athugavert varðandi viðmótið GUI-ið ?

Tester: Mér fannst græni takkinn mjög ljótur (hlátur)

Kristján: (hlátur), hversu hratt fannst þér tölvun svara þér ?

Tester: ég tók aldrei eftir því að ég væri að bíða neitt,

Kristján: Lendir aldrei í því að bíða. Hvernig fannst þér tölvun svara þér,
svona gáfulega séð ?

Tester: Ég náði aldrei svoleiðis pælingum í gang, ég sett fram rök og hún kom með á móti.
og ég veit ekki hvort að hún breytti um skoðun [umræða um stjörnuna, hann var ekki að skilja hvað það var]

Kristján: Hvernig líkaði þér við forritið svona alment ?

Tester: Ég hafði enga ánægu af því

Kristján: Hafðir þú ekkert gaman af því að spjalla við tölvuna ?

Tester: Nei,

Kristján: Nei

Tester: Ég held að það hafi aðalega verið útaf því að maður var fastur í einhverju boxi,
þú varst ekki, það var bara smá mengi sem þú varst bara að fara fram og til baka. Þú varst ekkert frjálsari í setningarvali

Kristján: Hvernig fannst þér umhverfið ? Fannst þér það skemmtilegt ?

Tester: Forritið

Kristján: Umhverfið, gui-ið

Tester: Mér fannst það bara ágætt

Kristján: Gætir þú hugsað þér að nota svona kerfi þér til gamans ?

Tester: nei, jú jú það væri allveg hægt til að byggja sjálfan sig upp fyrir rökræður,

en það þarf þá að vera frjálsara setningarval.

Kristján: Telur þú að það sé hægt að nota svona kerfi svo gagn sé að, eftir frekari þróun ?

Tester: Það er allveg séns, ef það er hægt að koma því í það level þannig að þú gætir haldið upp rökræðum til að æfa þig. Til að gera sjálfan þig betri í rökræðum.

Kristján: Nú er bara eitt umræðuefní dauðarefsingar, myndir þú vera líklegari til að nota svona kerfi ef hægt væri að velja úr umræðuefnum

Tester: örugglega

Kristján: Hvernig finnst þér sú hugmynd að þú gætir bætt inn umræðuefnum, þú veist hafir meiri sjórn yfir kerfinu.

Tester: Það er mjög gott, bara nauðsynlegt

Kristján: Ef kerfið væri alfarið á íslensku, væri þú líklegri til að nota það ?

Tester: Það væri örugglega auðveldara til að nota það, veit ekki hvort það sé sé líklegra

Kristján: Þá er það næst síðasta spurningin, hversu viss ertu að það hafi veirð tölvan sem þú varst að tala við en ekki ég í gömlu tölvunni minni ?

Tester: Ekki hugmynd

Kristján: Ekki hugmynd ?

Tester: Það gæti allveg verið einvher annar, það er svo lítið mengi af svörum og dóti, að talvan gæti allveg verið að velja eitthvað dunk dunk dunk.

Kristján: random ?

Tester: Já eða þú bara að koma með eitthvað á móti,

Kristján: Þú ert ekkert svo viss á þessu, en heldur þú að það sé hægt að búa svona hugbúnað þar sem þú getur rökrætt við tölvu ?

Tester: Það er hægt að spila skák við tölvu

Kristján: Já

Tester: Þá hlítur að vera hægt, rök og skák eru nánast sami hluturinn þá hlítur að vera hægt að byggja upp, ef hún hefur nægilegan skilning á orðunu eins og skák leikjum. Þá ætti að vera hægt að halda upp rökræðum.

Kristján: Et eitthvað sem þú vilt bæta við sem ekki hefur komið fram varðandi kerfið, gui-ið, kefrið ?

Tester: Græni takknn er ógeðslegur.

Kristján: Ok, þá þakka ég þér fyrir.

Test 10

Kristján: Góðan daginn og þakka þér fyrir að koma.

Kristján: Hefur þú notað samskipta forrit áður ?

Tester: Nei

Kristján: Til að eiga samskipti við fólk og svona ?

Tester: Já

Kristján: MSN og svona

Tester: Já

Kristján: Þarna ertu að hafa samskipti við tölvu, getur þú gefið mér samanburð á þessum, finnst þér petta eitthvað likt ?

Tester: Nei

Kristján: Hvað svona helsta ?

Tester: Uuuu aa, mér fannst óþægilegt að vita ekki í hvaða glugga ég á að skrifa eða hvað ég á að gera, maður er út um allan skjá

Kristján: OK, í byrjun kom upplýsinga gluggi, hvernig fannst þér að fá svona upplýsinga glugga í byrjun ?

Tester: Það var voða gott

Kristján: lastu hann allan ?

Tester: Nei

Kristján: Ok, hvernig fannst þér þessar upplýsingar, skýrðu þær nægilega út fyrir þér ?

Tester: Uuuu ég las hann ekki allan

Kristján: En það sem þú last ?

Tester: Já, það skýrði eitthvað út fyrir mér

Kristján: Ef þú myndir nota svona forrit aftur, myndir þú þá lesa upplýsingarnar aftur ?

Tester: Nei

Kristján: Lestu almennt svona upplýsingar ?

Tester: nei

Kristján: Nei, ok, hvað gerðir þú við gluggan eftir að þú varst búinn að lesa hann ?

Tester: Lokaði honum

Kristján: Íttir þú á close takkann ?

Tester: Ég man það ekki.

Kristján: OK, þurftir þú að skoða hann aftur ?

Tester: Nei

Kristján: Er eitthvað að angra þig við upplýsinga gluggan ?

Tester: Nei hann truflaði mig ekkert

Kristján: Hvernig gekk að byrja leikinn ?

Tester: Nokkuð vel

Kristján: Já

Tester: Ég vissi ekkert hvað ég ætti að gera, kannski afþví að ég las ekki allan upplýsinga gluggan

Kristján: Tókstu eftir textanum sem blikkandi á skjánum

Tester: Nei, ekki þegar ég var að byrja, bara eftir að ég var byrjaður..

Kristján: Hvernig finnst þér að staðfesta hvert val, með því að íta á takka í hvert skipti ?

Tester: Mér fannst það óþæginlegt, kannski er takkinn bara á óþægilegum stað

Kristján: OK, hjálpaði textinn sem blikkaði neðst á skjánum í leiknum ?

Tester: Já, svonan þegar ég fór að fatta hvað hann gerði

Kristján: Hvernig gekk að átta sig á þessum 3 aðalgluggum, hver táknað hvað ?

Tester: Illa

Kristján: hvernig þá ?

Tester: ég vissi ekkert hvað ég átti að velja úr hvaða glugga, eða úr þessum 2 efstu

Kristján: var erfitt að átta sig á hvar samtalið færi fram ?

Tester: já

Kristján: Ok, allir hlutir á viðmótinu hafa haus, hjálpuðu þeir eitthvað, skýrðu þeir út fyrir þér ?

Tester: Já, ég skildi hvað my position og computer position þýddi.

Kristján: Hvernig líkar þér að velja setningar úr svona 2 dropdown boxum, tala þannig ?

Tester: Uuu, stundum þarf maður að velja úr my position og stundum úr computer position, stundum úr dropdown boxinu, það fannst mér allveg einstaklega óþægilegt.

Kristján: En, hvernig fannst þér valmöguleikarnir sem þú hafðir, of margir eða fáir ?

Tester: Frekar margir sko

Kristján: Myndir þú frekar vilja skrifa setningarnar sjálfur inn ?

Tester: Já

Kristján: afhverju ?

Tester: þægilegra

Kristján: Ef þetta væri á íslensku, myndir þú vilja frekar skrifa inn sjálfur ?

Tester: Á íslensku ?

Kristján: Já ef þetta væri á íslensku

Tester: Skiptir ekki máli

Kristján: Hvernig fannst þér aðgangur að valmöguleikum, upplýsingum, stillingum, kiktir þú eitthvað á það ?

Tester: Já, ég kíkti aðeins á það, ég valdi eitthvað þarna og hún var frosinn þarna í smá tíma

Kristján: Er eitthvað sem þú tókst eftir sem ekki hefur komið fram varðandi viðmótið, eitthvað sem var að angra þig ?

Tester: Viðmótið angraði mig ekkert.

Kristján: Hversu hratt fannst þér tölvun svara þér ?

Tester: Bara hratt,

Kristján: Lentir þú í því að bíða ?

Tester: U ég lenti ekki í því að bíða þegar ég var að tala við hana. Bara þegar ég var að skoða aðra glugga.

Kristján: Hvernig fannst þér tölvun svar þér, gáfulega þá ?

Tester: Mér fannst hún pirrandi þá. Hún vildi ekki svara alltaf og bara með kjaft og leiðindi

Kristján: Kjaft, hvað þá, hvað meinar þú ?

Tester: Uuuu eins og, ummm þegar talvan var búinn að koma sínum pointum á framfæri, þá vildi hún ekkert rökræða það meira, og þá átti ég að koma með einhver point og vildi ekkert ræða það frekar.

Kristján: Hérrna hvernig líkar þér við forritið svona alment ?

Tester: Frekar pirrandi, hvað þá útaf samtalini og viðmótinu

Kristján: Hafið þú gaman að því að spjalla við tölvuna, argumenta við hana ?

Tester: Alls ekki

Kristján: Hvað með umhverfið, myndir þú segja að það væri skemmtilegt ?

Tester: Nei,

Kristján: Hvernig myndir þú lísa því ?

Tester: Bara klúðurslegt og leiðinlegt

Kristján: OK, uu eftir frekari þróun og lagfæringar, myndir þú geta hugsað þér að nota þetta kerfi þér til gamans ?

Tester: Ekki til ánægðu nei.

Kristján: U, en endurtek eftir frekari þróun, telur þú að það sé hægt að nota svona kerfi einhverjum til gagns ?

Tester: Já já, það tel ég

Kristján: Hvernig þá ?

Tester: uu, svona help desk eitthvað

Kristján: Ok, nú er bara eitt umræðuefní dauðrefsingar, væri þú liklegri til að nota svona kerfi ef það væri hægt að velja úr nokkrum umræðuefnum ?

Tester: Það sé ég ekki fyrir mér.

Kristján: Hefur þú áhuga að bæta inn umræðuefnum og hafa meiri stjórn yfir kerfinu ?

Tester: Nei nei

Kristján: Ef kerfið væri á íslensku, væri þú líklegri til að nota það

Tester: nei

Kristján: Hefur þú eitthvað fleira að bæta við sem hefur ekki komið fram ?

Tester: mér finnst bara leiðinlegt að tala við tölvur

Kristján: Ok, þá takka ég þér bara fyrir.