

# AJA 1.0 Installation and Usage

## Installation and the Directory Structure

To install AJA 1.0 one need only to extract the file **AJA\_1.0.zip**. If the JDK 1.4 or higher is installed on the computer, AJA 1.0 can be compiled and used.

The file **AJA\_1.0.zip** contains all AJA 1.0 files. After extracting the archive, the following directory structure will be created:

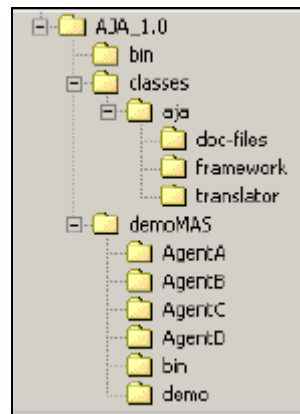


Figure 1 The directory structure after the extraction of **AJA\_1.0.zip**.

The directory **AJA\_1.0** contains two files:

- **AJA\_1.0\readme.txt** – briefly describes the files and directories in the archive.
- **AJA\_1.0\guide.pdf** – describes in details the usage of AJA 1.0 and implemented demo MAS.

In the directory **AJA\_1.0\bin** are MS Windows/DOS script files used to compile all AJA parts and to generate HTML javadoc documentation:

- **AJA\_1.0\bin\setJAVABIN.bat** – this batch file is called in all other AJA batch files in the archive. Edit this file and set the appropriate path of your JDK 1.4 bin directory as a value of the **JAVABIN** environment variable. The preset value is **C:\j2sdk1.4.1\bin**.
- **AJA\_1.0\bin\1\_comp.bat** – compiles the java package **aja.framework**.

- **AJA\_1.0\bin\2\_genstub.bat** – generates a RMI stub class used in the RMI communication between AJA Agents.
- **AJA\_1.0\bin\3\_compt.bat** – compiles the java package `aja.translator`.
- **AJA\_1.0\bin\4\_compmain.bat** – compiles the AJA translator program.
- **AJA\_1.0\bin\5\_gendoc.bat** – generates HTML javadoc documentation. The documentation files will be in the new directory **AJA\_1.0\doc**. The file **AJA\_1.0\doc\index.html** is the starting page of the documentation.

One can use these batch files as a template to write the appropriate script files for UNIX or some other operating system, if other operating system than MS Windows is used.

**AJA\_1.0\classes** directory contains subdirectories with all AJA source files, i.e. .java files. After the compilation of the AJA source files, the generated binary files, i.e. .class files, will also be located here.

- the directory **AJA\_1.0\classes\aja\framework** contains the files belonging to the package `aja.framework`.
- the directory **AJA\_1.0\classes\aja\translator** contains the files belonging to the package `aja.translator`.
- the directory **AJA\_1.0\classes\aja\doc-files** contains HADL grammar and the description of all the Java+ elements. These files are used in the generation of javadoc documentation.
- the directory **AJA\_1.0\classes\aja** contains the files belonging to the package `aja`. In this package is only the main program of the AJA translator.

**AJA\_1.0.zip** file contains an example MAS implemented in AJA, which demonstrates many AJA features. The files belonging to this example are located in the directory **AJA\_1.0\demoMAS**.

## Preparing AJA for the First Use

1. Edit the file **AJA\_1.0\bin\setJAVABIN.bat** and set the appropriate path of your JDK 1.4 bin directory as a value of the **JAVABIN** environment variable. The preset value is `C:\j2sdk1.4.1\bin`.
2. Execute the batch file **AJA\_1.0\bin\1\_compf.bat** in order to compile the package `aja.framework`. The file can be executed e.g. by double-clicking its icon in the Windows-Explorer.

```
C:\WINNT\System32\cmd.exe
C:\AJA_1.0\bin>call setJAVABIN.bat
C:\AJA_1.0\bin>SET JAVABIN=C:\j2sdk1.4.1\bin
C:\AJA_1.0\bin>cd ..\classes\aja\framework
C:\AJA_1.0\classes\aja\framework>C:\j2sdk1.4.1\bin\javac -d ..\.. @sourcefiles.txt
C:\AJA_1.0\classes\aja\framework>cd ..\..\bin
C:\AJA_1.0\bin>pause
Drücken Sie eine beliebige Taste . . .
```

Figure 2 The file `AJA_1.0\bin\1_comp.f.bat` has been executed.

3. Execute the batch file `AJA_1.0\bin\2_genstub.bat` in order to generate a RMI stub class for the class `aja.framework.Agent`.

```
C:\WINNT\System32\cmd.exe
C:\AJA_1.0\bin>call setJAVABIN.bat
C:\AJA_1.0\bin>SET JAVABIN=C:\j2sdk1.4.1\bin
C:\AJA_1.0\bin>cd ..\classes
C:\AJA_1.0\classes>C:\j2sdk1.4.1\bin\rmic -v1.2 -classpath . -d . aja.framework.Agent
C:\AJA_1.0\classes>cd ..\bin
C:\AJA_1.0\bin>pause
Drücken Sie eine beliebige Taste . . .
```

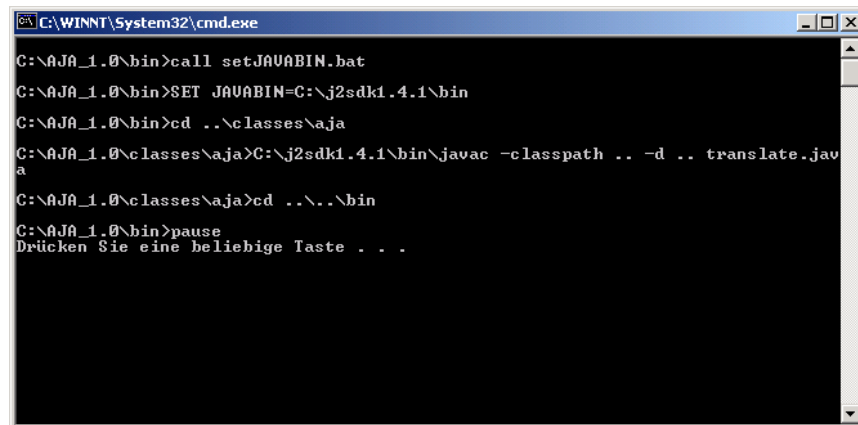
Figure 3 The file `AJA_1.0\bin\2_genstub.bat` has been executed.

4. Execute the batch file `AJA_1.0\bin\3_compt.bat` in order to compile the package `aja.translator`.

```
C:\WINNT\System32\cmd.exe
C:\AJA_1.0\bin>call setJAVABIN.bat
C:\AJA_1.0\bin>SET JAVABIN=C:\j2sdk1.4.1\bin
C:\AJA_1.0\bin>cd ..\classes\aja\translator
C:\AJA_1.0\classes\aja\translator>C:\j2sdk1.4.1\bin\javac -d ..\.. @sourcefiles.txt
C:\AJA_1.0\classes\aja\translator>cd ..\..\bin
C:\AJA_1.0\bin>pause
Drücken Sie eine beliebige Taste . . .
```

Figure 4 The file `AJA_1.0\bin\3_compt.bat` has been executed.

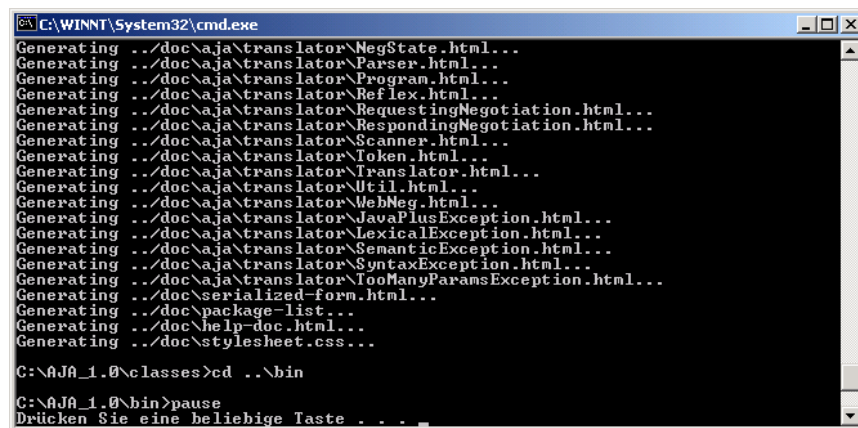
- Execute the batch file `AJA_1.0\bin\4_compmain.bat` in order to compile the translator main program (i.e. the class `aja.translate`).



```
C:\WINNT\System32\cmd.exe
C:\AJA_1.0\bin>call setJAVABIN.bat
C:\AJA_1.0\bin>SET JAVABIN=C:\j2sdk1.4.1\bin
C:\AJA_1.0\bin>cd ..\classes\aja
C:\AJA_1.0\classes\aja>C:\j2sdk1.4.1\bin\javac -classpath .. -d .. translate.java
C:\AJA_1.0\classes\aja>cd ..\..\bin
C:\AJA_1.0\bin>pause
Drücken Sie eine beliebige Taste . . .
```

Figure 5 The file `AJA_1.0\bin\4_compmain.bat` has been executed.

- To run the example MAS one need not the AJA documentation. However, the documentation will be useful later, when one start to develop his/her own MAS with AJA. Executing the batch file `AJA_1.0\bin\5_gendoc.bat` generates the AJA documentation.



```
C:\WINNT\System32\cmd.exe
Generating ../doc/aja/translator/NegState.html...
Generating ../doc/aja/translator/Parser.html...
Generating ../doc/aja/translator/Program.html...
Generating ../doc/aja/translator/Reflex.html...
Generating ../doc/aja/translator/RequestingNegotiation.html...
Generating ../doc/aja/translator/RespondingNegotiation.html...
Generating ../doc/aja/translator/Scanner.html...
Generating ../doc/aja/translator/Token.html...
Generating ../doc/aja/translator/Translator.html...
Generating ../doc/aja/translator/Util.html...
Generating ../doc/aja/translator/WebNeg.html...
Generating ../doc/aja/translator/JavaPlusException.html...
Generating ../doc/aja/translator/LexicalException.html...
Generating ../doc/aja/translator/SemanticException.html...
Generating ../doc/aja/translator/SyntaxException.html...
Generating ../doc/aja/translator/TooManyParamsException.html...
Generating ../doc/serialized-form.html...
Generating ../doc/package-list...
Generating ../doc/help-doc.html...
Generating ../doc/stylesheet.css...
C:\AJA_1.0\classes>cd ..\bin
C:\AJA_1.0\bin>pause
Drücken Sie eine beliebige Taste . . .
```

Figure 6 The file `AJA_1.0\bin\5_gendoc.bat` has been executed.

A generated file `AJA_1.0\doc\index.html` is the starting page of the documentation.

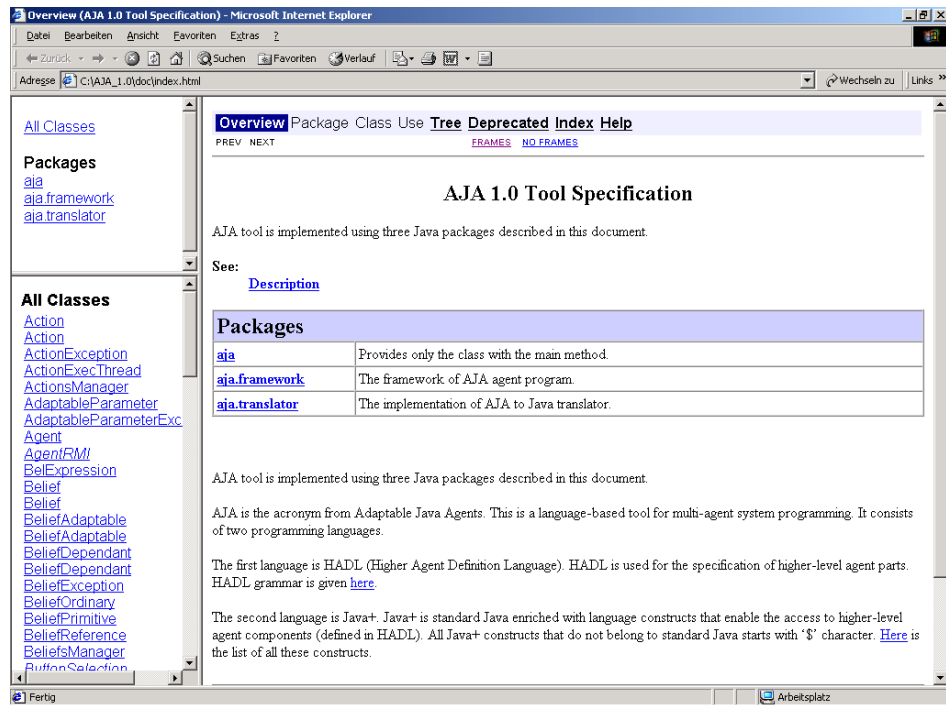


Figure 7 HTML documentation of AJA classes.

## How to Translate, Compile, Run, and Use the Example Agents

### Agents in the System and the Locations of their Files

There are four personal digital assistant (PDA) agents in the example MAS. It is a homogenous MAS, i.e. all four PDA agents are equal. They only differ in their names and URLs.

The first agent is named **A** and the source code of its AJA program is in the directory `AJA_1.0\demoMAS\AgentA`.

Similarly, the source codes of the agents **B**, **C**, and **D** are in the directories `AJA_1.0\demoMAS\AgentB`, `AJA_1.0\demoMAS\AgentC`, and `AJA_1.0\demoMAS\AgentD` respectively.

Since AJA supports the use of java in the implementation of AJA agents, a java package `demo` is made and used in the implementation of the example agents. The files belonging to the package `demo` are placed in the directory `AJA_1.0\demoMAS\demo`.

The directory `AJA_1.0\demoMAS\bin` contains fifteen batch files used to translate, compile and run the agents. In order to do this, AJA classes has to be compiled first.

### Starting the Agents

There are fifteen batch files in the directory `AJA_1.0\demoMAS\bin`. The name of each batch file starts with a number between 01 and 15. These numbers determine the order of the batch files execution:

1. As first, execute the batch file `AJA_1.0\demoMAS\bin\01_compDemo.bat`. A batch file can be executed e.g. by double-clicking its icon in the Windows-Explorer. The file `01_compDemo.bat` compiles the java package `demo`, which is used in the implementation of the example agents.

```

C:\WINNT\System32\cmd.exe
C:\AJA_1.0\demoMAS\bin>call ..\..\bin\setJAUABIN.bat
C:\AJA_1.0\demoMAS\bin>SET JAUABIN=C:\j2sdk1.4.1\bin
C:\AJA_1.0\demoMAS\bin>cd ..\demo
C:\AJA_1.0\demoMAS\demo>C:\j2sdk1.4.1\bin\javac -d .. @sourcefiles.txt
C:\AJA_1.0\demoMAS\demo>cd ..\bin
C:\AJA_1.0\demoMAS\bin>pause
Drücken Sie eine beliebige Taste . . .

```

Figure 8 AJA\_1.0\demoMAS\bin\01\_compDemo.bat has been executed.

- Execute the batch file AJA\_1.0\demoMAS\bin\02\_createKeys.bat in order to create keystores for agents, to create public-private key pairs and to exchange certificates among agents. All messages that agent exchange among themselves are digitally signed.

```

C:\WINNT\System32\cmd.exe
C:\AJA_1.0\demoMAS\bin>C:\j2sdk1.4.1\bin\keytool -import -noprompt -alias localh
ost_1099_D -keystore ..\AgentC\C.ks -storepass ckspass -file ..\AgentD\Dcertfile
.cer
Zertifikat wurde zu Keystore hinzugef³gt.
C:\AJA_1.0\demoMAS\bin>REM import other certificates in ..\AgentD\D.ks
C:\AJA_1.0\demoMAS\bin>C:\j2sdk1.4.1\bin\keytool -import -noprompt -alias localh
ost_1099_A -keystore ..\AgentD\D.ks -storepass dkspass -file ..\AgentA\Acertfile
.cer
Zertifikat wurde zu Keystore hinzugef³gt.
C:\AJA_1.0\demoMAS\bin>C:\j2sdk1.4.1\bin\keytool -import -noprompt -alias localh
ost_1099_B -keystore ..\AgentD\D.ks -storepass dkspass -file ..\AgentB\Bcertfile
.cer
Zertifikat wurde zu Keystore hinzugef³gt.
C:\AJA_1.0\demoMAS\bin>C:\j2sdk1.4.1\bin\keytool -import -noprompt -alias localh
ost_1099_C -keystore ..\AgentD\D.ks -storepass dkspass -file ..\AgentC\Ccertfile
.cer
Zertifikat wurde zu Keystore hinzugef³gt.
C:\AJA_1.0\demoMAS\bin>PAUSE
Drücken Sie eine beliebige Taste . . .

```

Figure 9 AJA\_1.0\demoMAS\bin\02\_createKeys.bat has been executed.

- Execute the batch file AJA\_1.0\demoMAS\bin\03\_transA.bat in order to translate the AJA source code of the agent A into Java.

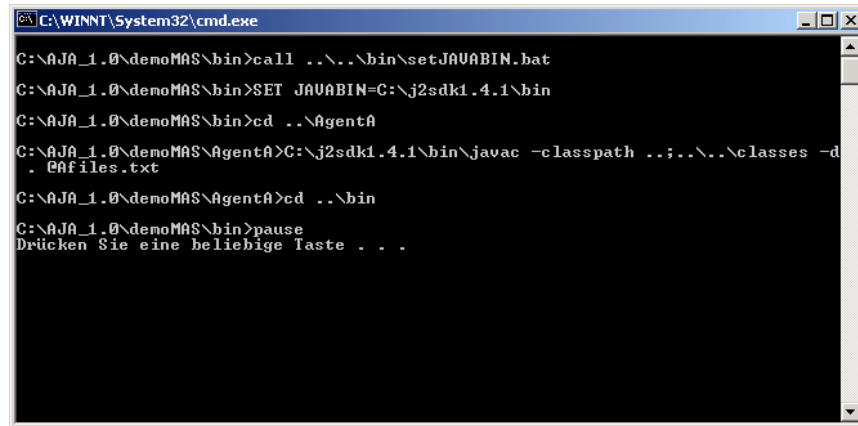
```

C:\WINNT\System32\cmd.exe
...FindReplacementReqNeg
...EngagementInitReqNeg
...RepeatedInformReqNeg
processing responding negotiations:
...InformResNeg
...GetBirthdayResNeg
...ReplacementResNeg
...EngagementInitResNeg
processing web negotiation
processing reflexes:
...eventAlertReflex
...birthdayAlertReflex
...removeOldEngagementsReflex
...backupTimeTableReflex
...backupEventAlertTimeReflex
...backupConsultationDurationReflex
processing initial agent part
Names of generated java files are listed in C:\AJA_1.0\demoMAS\AgentA\Afiles.txt
Translation successfully completed in 0.922 sec.
C:\AJA_1.0\demoMAS\AgentA>cd ..\bin
C:\AJA_1.0\demoMAS\bin>pause
Drücken Sie eine beliebige Taste . . .

```

Figure 10 AJA\_1.0\demoMAS\bin\03\_transA.bat has been executed.

- Execute the batch file `AJA_1.0\demoMAS\bin\04_compA.bat` in order to compile generated Java classes in the previous step.



```
C:\WINNT\System32\cmd.exe
C:\AJA_1.0\demoMAS\bin>call ..\..\bin\setJAUABIN.bat
C:\AJA_1.0\demoMAS\bin>SET JAUABIN=C:\j2sdk1.4.1\bin
C:\AJA_1.0\demoMAS\bin>cd ..\AgentA
C:\AJA_1.0\demoMAS\AgentA>C:\j2sdk1.4.1\bin\javac -classpath .;..\..\classes -d
. @files.txt
C:\AJA_1.0\demoMAS\AgentA>cd ..\bin
C:\AJA_1.0\demoMAS\bin>pause
Drücken Sie eine beliebige Taste . . .
```

Figure 11 `AJA_1.0\demoMAS\bin\04_compA.bat` has been executed.

- Execute the batch file `AJA_1.0\demoMAS\bin\05_transB.bat` in order to translate the AJA source code of the agent **B** into Java.
- Execute the batch file `AJA_1.0\demoMAS\bin\06_compB.bat` in order to compile generated Java classes in the previous step.
- Execute the batch file `AJA_1.0\demoMAS\bin\07_transC.bat` in order to translate the AJA source code of the agent **C** into Java.
- Execute the batch file `AJA_1.0\demoMAS\bin\08_compC.bat` in order to compile generated Java classes in the previous step.
- Execute the batch file `AJA_1.0\demoMAS\bin\09_transD.bat` in order to translate the AJA source code of the agent **D** into Java.
- Execute the batch file `AJA_1.0\demoMAS\bin\10_compD.bat` in order to compile generated Java classes in the previous step.
- Execute the batch file `AJA_1.0\demoMAS\bin\11_startRMireg.bat` in order to start the `rmiregistry`. `rmiregistry` will be started in a new minimized window.
- Execute the batch file `AJA_1.0\demoMAS\bin\12_runA.bat` in order to start the agent **A**. Before the agent window appears, a small dialog window pops up:

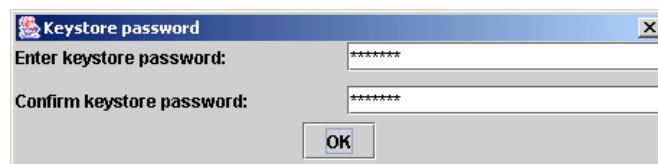
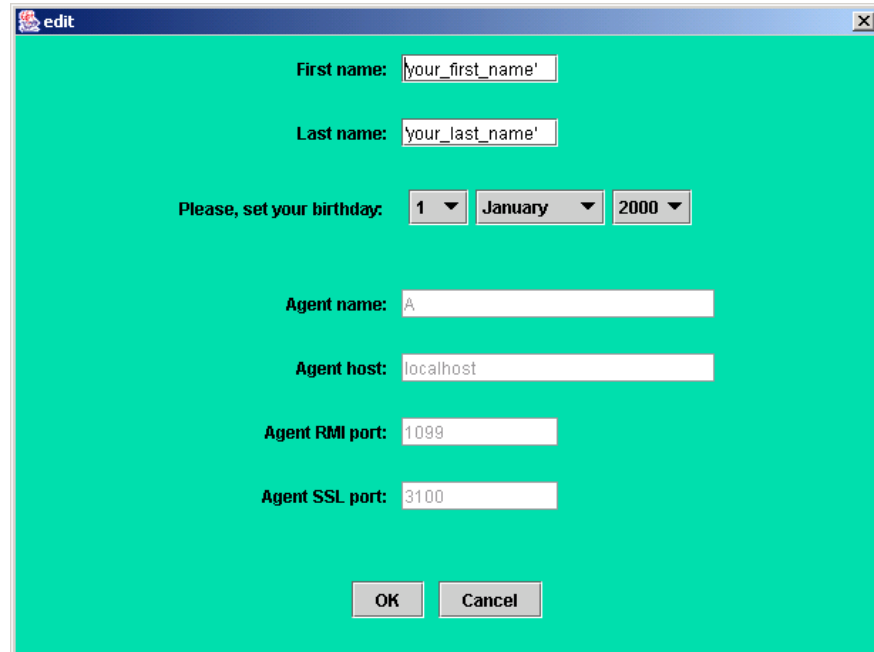


Figure 12 Dialog window with password text fields.

Enter and confirm the keystore password for the agent **A: akspass**. This window appears by all AJA agents in order to avoid password storing in the program source code.

A new dialog window appears:

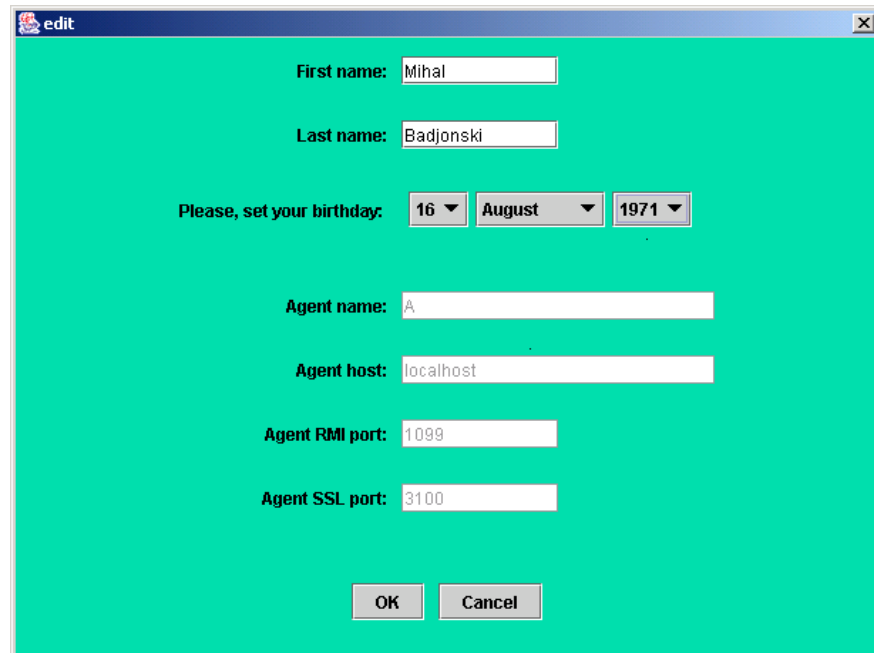


The screenshot shows a dialog window titled 'edit' with a blue title bar and a close button. The window has a light blue background. It contains the following fields and controls:

- First name:** A text input field containing the placeholder text 'your\_first\_name'.
- Last name:** A text input field containing the placeholder text 'your\_last\_name'.
- Please, set your birthday:** Three dropdown menus. The first shows '1', the second shows 'January', and the third shows '2000'.
- Agent name:** A text input field containing the value 'A'.
- Agent host:** A text input field containing the value 'localhost'.
- Agent RMI port:** A text input field containing the value '1099'.
- Agent SSL port:** A text input field containing the value '3100'.
- At the bottom, there are two buttons: 'OK' and 'Cancel'.

Figure 13 Entering owner data of the agent **A**.

Enter the first name, the last name, and the birthday of the agent owner and click OK. For example:



The screenshot shows the same 'edit' dialog window as in Figure 13, but with example data entered:

- First name:** A text input field containing the value 'Mihal'.
- Last name:** A text input field containing the value 'Badjonski'.
- Please, set your birthday:** Three dropdown menus. The first shows '16', the second shows 'August', and the third shows '1971'.
- Agent name:** A text input field containing the value 'A'.
- Agent host:** A text input field containing the value 'localhost'.
- Agent RMI port:** A text input field containing the value '1099'.
- Agent SSL port:** A text input field containing the value '3100'.
- At the bottom, there are two buttons: 'OK' and 'Cancel'.

Figure 14 Owner data for the agent **A**.

After clicking OK button, the main agent window finally appears:

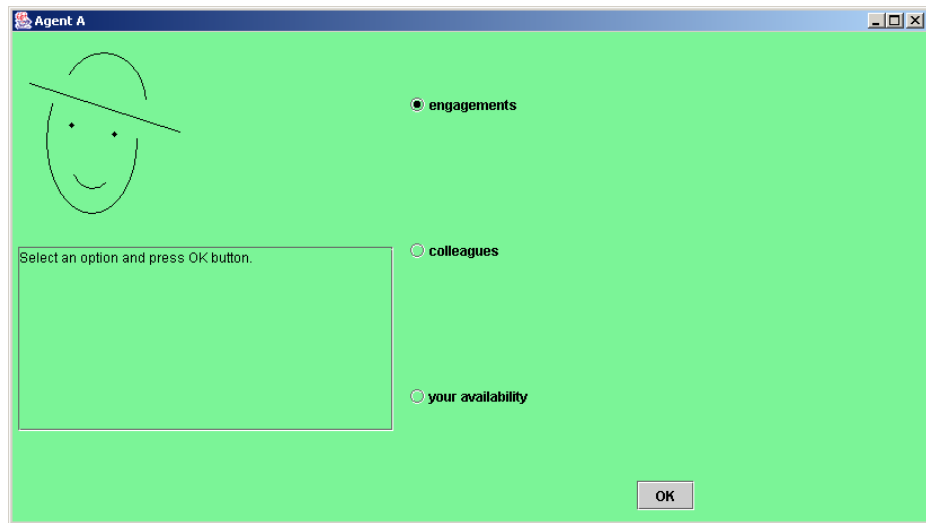


Figure 15 The main window of the agent A.

Debug information can be seen in the console of the agent A:

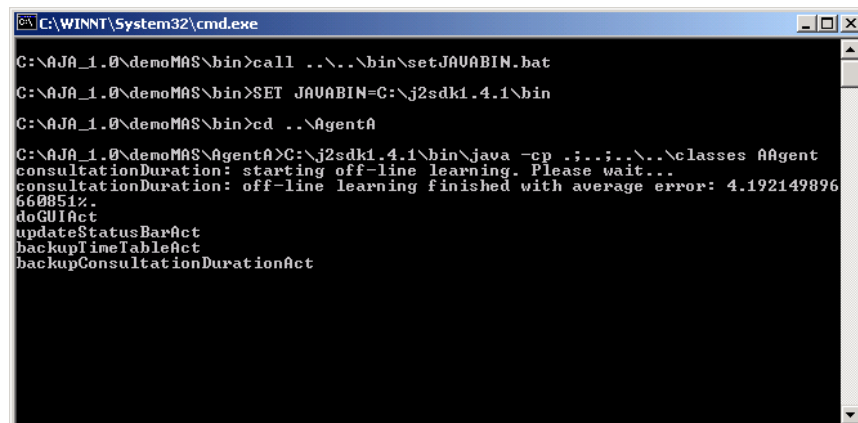


Figure 16 The console of the agent A.

13. Execute the batch file `AJA_1.0\demoMAS\bin\13_runB.bat` in order to start the agent B.

Keystore password of the agent B is **bkspass**.

Set the owner data for the agent B, e.g.:

edit

First name:

Last name:

Please, set your birthday:

Agent name:

Agent host:

Agent RMI port:

Agent SSL port:

Figure 17 Owner data for the agent **B**.

- Execute the batch file `AJA_1.0\demoMAS\bin\14_runC.bat` in order to start the agent **C**.

Keystore password of the agent **C** is **ckspass**.

Set the owner data for the agent **C**, e.g.:

edit

First name:

Last name:

Please, set your birthday:

Agent name:

Agent host:

Agent RMI port:

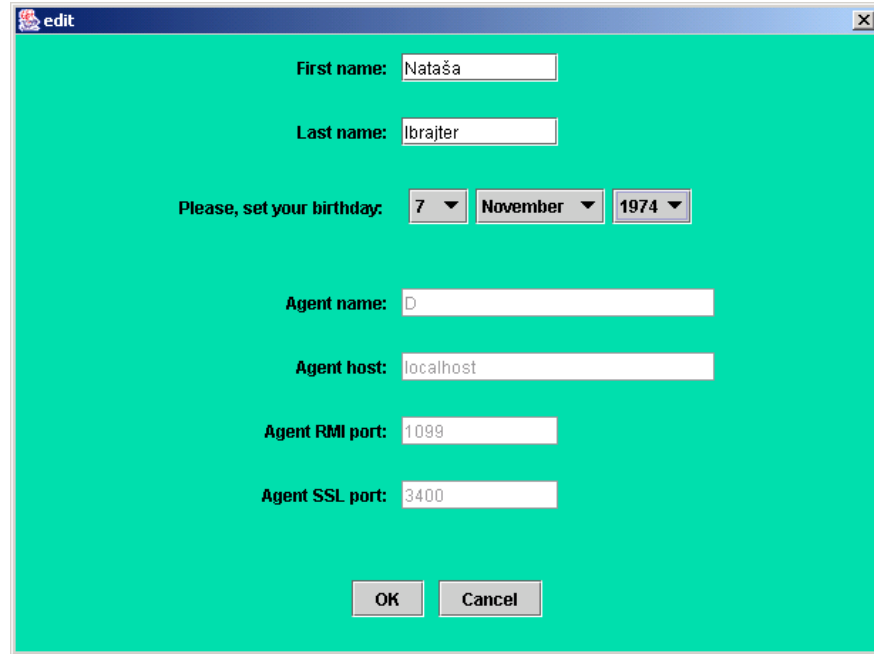
Agent SSL port:

Figure 18 Owner data for the agent **C**.

15. Execute the batch file `AJA_1.0\demoMAS\bin\15_runD.bat` in order to start the agent **D**.

Keystore password of the agent **D** is **dkspass**.

Set the owner data for the agent **D**, e.g.:



The screenshot shows a dialog box titled "edit" with a blue title bar. The dialog contains the following fields and controls:

- First name:** Nataša
- Last name:** Ibrajter
- Please, set your birthday:** 7 (dropdown), November (dropdown), 1974 (dropdown)
- Agent name:** D
- Agent host:** localhost
- Agent RMI port:** 1099
- Agent SSL port:** 3400
- Buttons:** OK, Cancel

Figure 19 Owner data for the agent **D**.

Now, all four PDA Agents run on localhost.

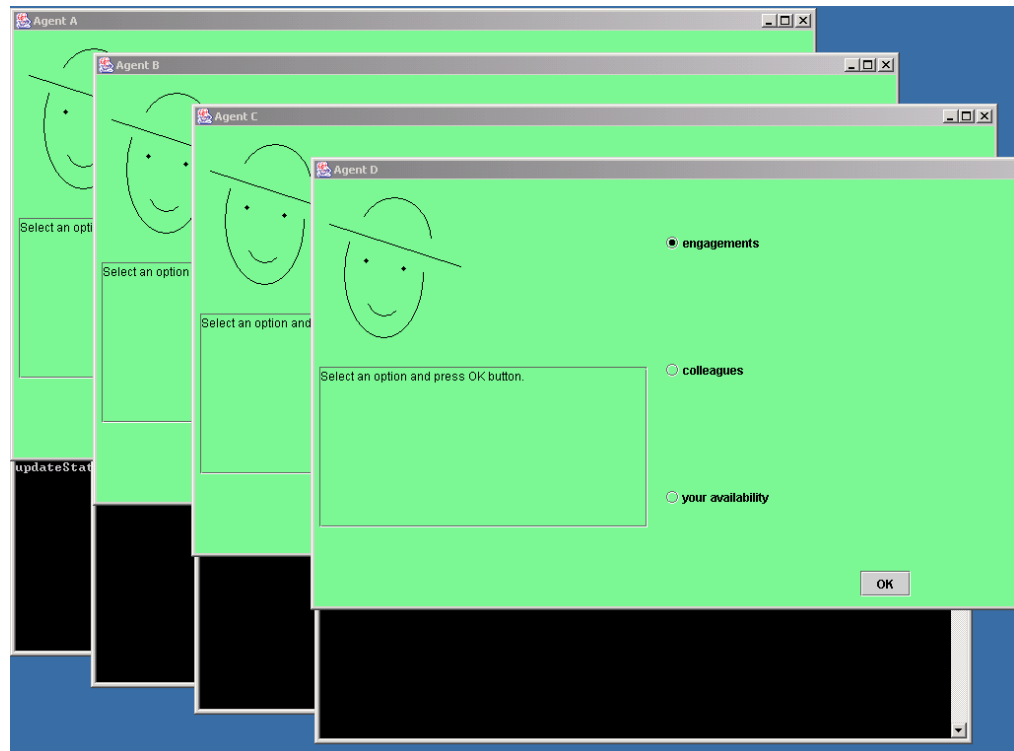


Figure 20 Four PDA Agents are executing on one computer.

## Using Example Agents

The example MAS consists of four PDA agents. Each agent belongs to one assistant or to one professor at one university. An agent manages the available time and engagements of its owner. In most of the engagements there are two or more participants. Agents in the MAS are able to negotiate the time of the joint engagement. A participant in an engagement can also find the replacement using his/her agent.

An AJA agent can be accessed via Internet. Students communicate with PDA agents using their Internet browsers. Here they can find out when their assistant or professor conducts the consultations. If a student wants to visit the consultation, he has to register himself/herself by the agent. The agent can then estimate the consultation duration and inform its owner about the consultation participants and the consultation topics.

## Managing Colleagues

To add, remove, or edit a colleague the agent owner has to select **colleagues** radio button in the main agent window and then to click the **OK** button.

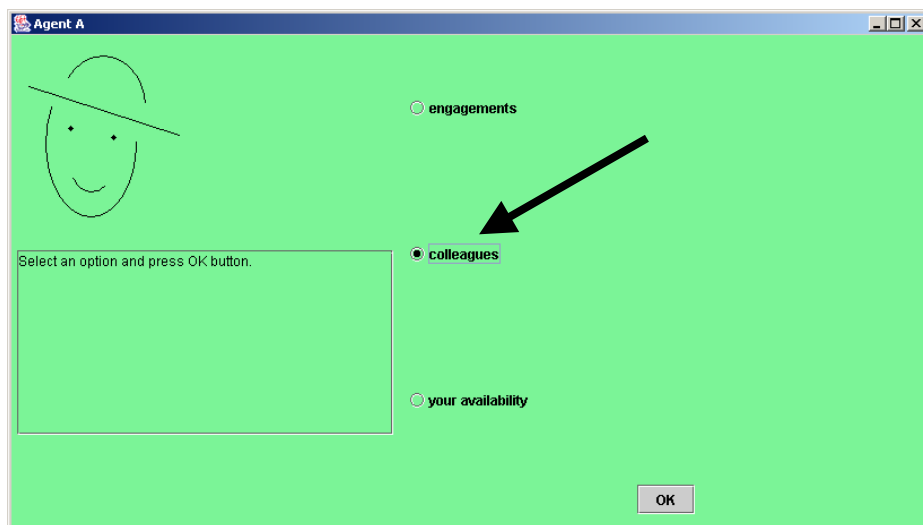


Figure 21 **colleagues** radio-button.

After selecting **colleagues** and clicking **OK** in the window of the agent **A**, the window looks like the one on the following figure.

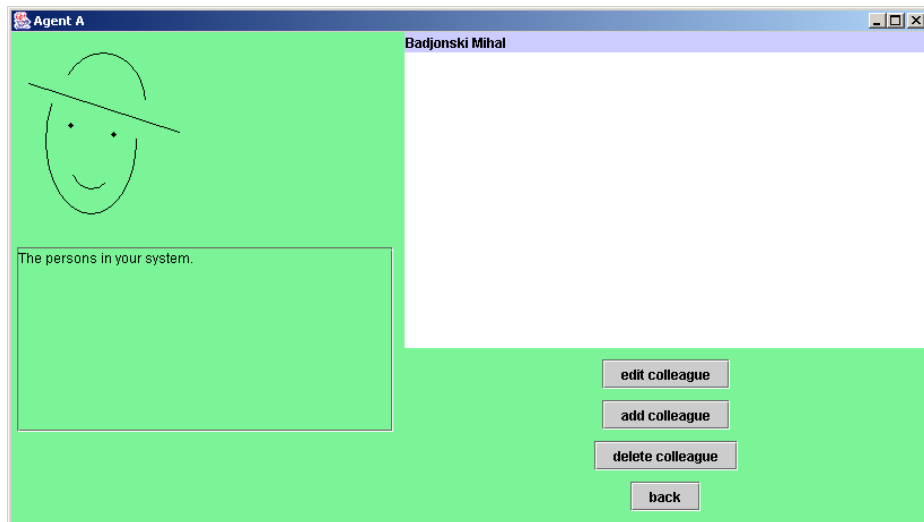


Figure 22 Initial colleagues list of the agent **A**.

The only registered person at the moment is the owner of the agent **A**. Clicking on the **add colleague** button creates the following dialog window:

Figure 23 **new colleague** dialog window.

To add a colleague Nataša Ibrajter, which is the owner of the agent **D**, the text fields has to be filled out as on the following figure.

Figure 24 The owner of the agent **A** adds the owner of the agent **D** to its list of colleagues.

Similarly can be added other two colleagues to the list of colleagues of **A**.

Figure 25 Colleagues list of the agent **A**.

Owners of the agents **B**, **C**, and **D** add their colleagues on the same way.

## Managing Available Times

To specify the time intervals when he/she is available for appointments and other engagements, the agent owner has to select **your availability** radio button in the main agent window and then to click the **OK** button.

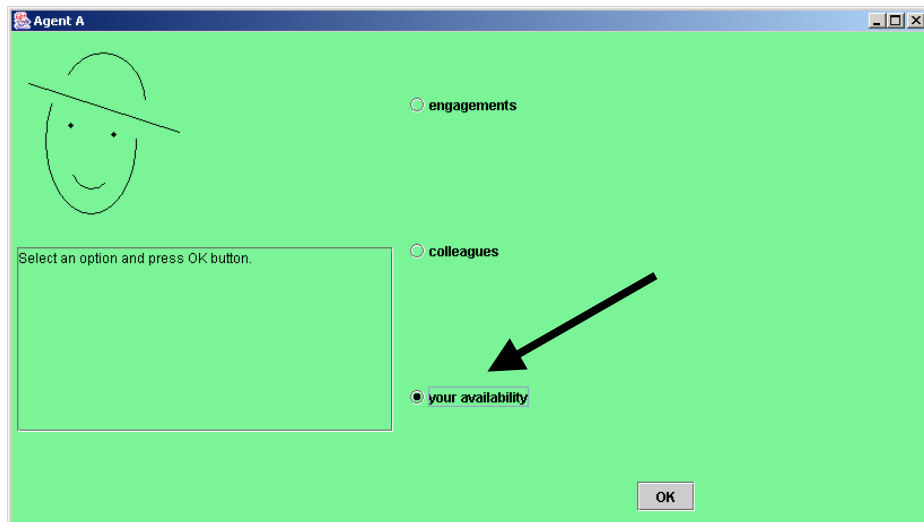


Figure 26 **your availability** radio-button.

After clicking **OK** in the agent **A** window, the window gets a new look.

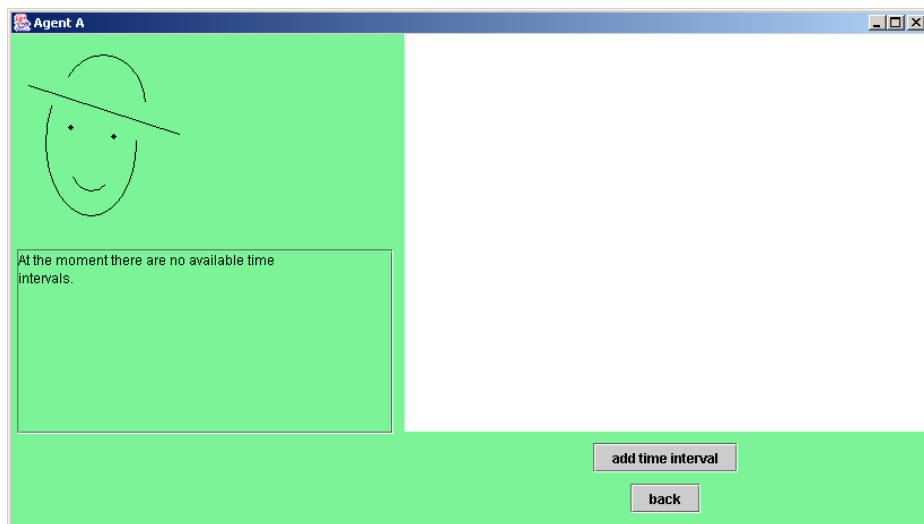


Figure 27 The agent **A** has at the beginning no registered available time intervals.

Click **add time interval** button in order to add a new available time interval. A new dialog window is created.

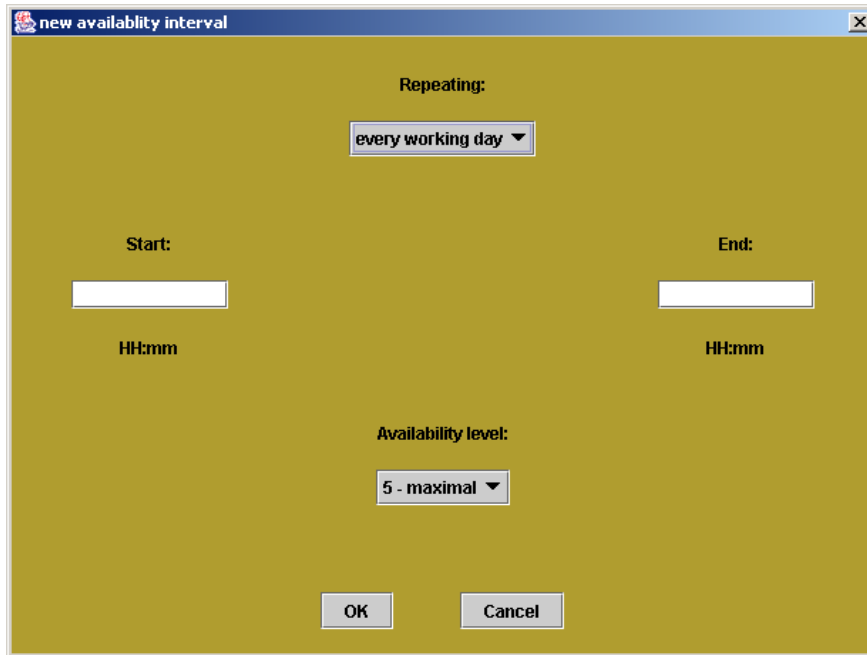


Figure 28 **new availability interval** dialog window.

There are four types of availability intervals:

- repeating every working day
- repeating every day
- not periodical
- repeating every weak

The availability level of an availability interval is a value between **1** (minimal availability) and **5** (maximal availability).

An availability interval with the availability **5**, from **9:00** until **15:00** in **every working day** can be specified as below.

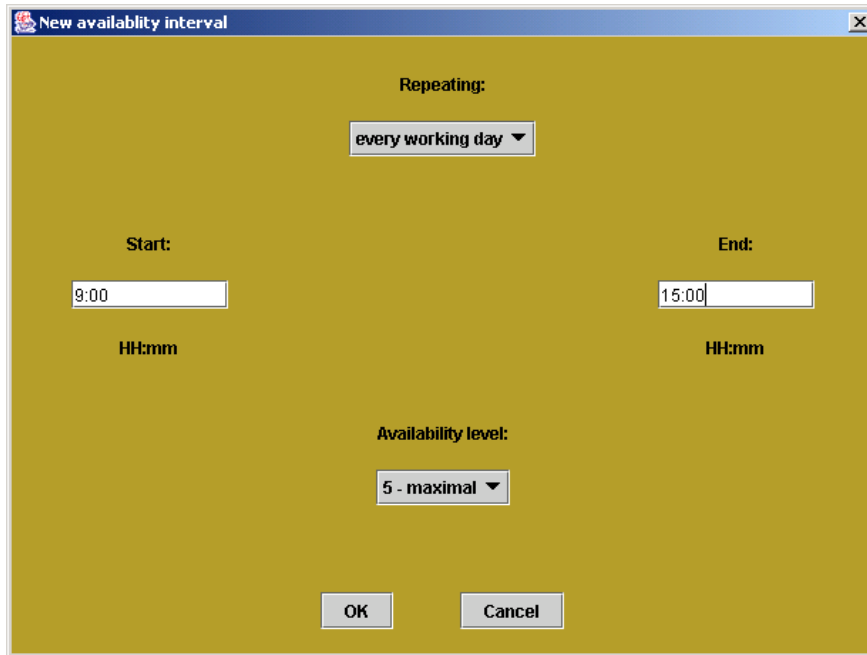


Figure 29 Defining an availability interval.

After clicking **OK** button, the list of the availability time intervals of the agent **A** has one element.

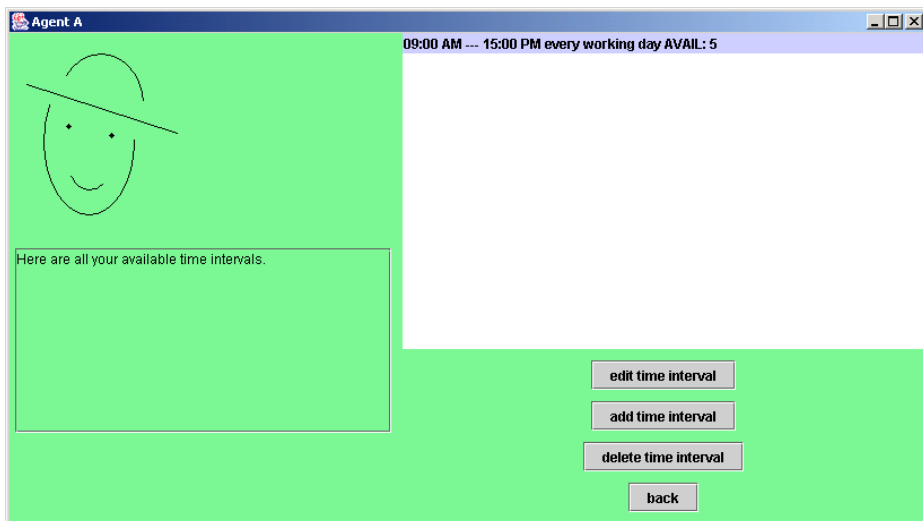


Figure 30 Available time intervals of the agent **A**.

Repeating above steps, other time intervals can be defined for the agent **A**, as well as for the other three agents in the example MAS.

## Managing Engagements

A person can initiate an engagement involving other persons. For example, the owner of the agent **A** (Mihal Badjonski) can initiate an engagement with the owner of the agent **B** (Mirjana Ivanović) and the owner of the agent **C** (Zoran Budimac).

To create an engagement, one has to select **engagements** radio-button and to click **OK** button.

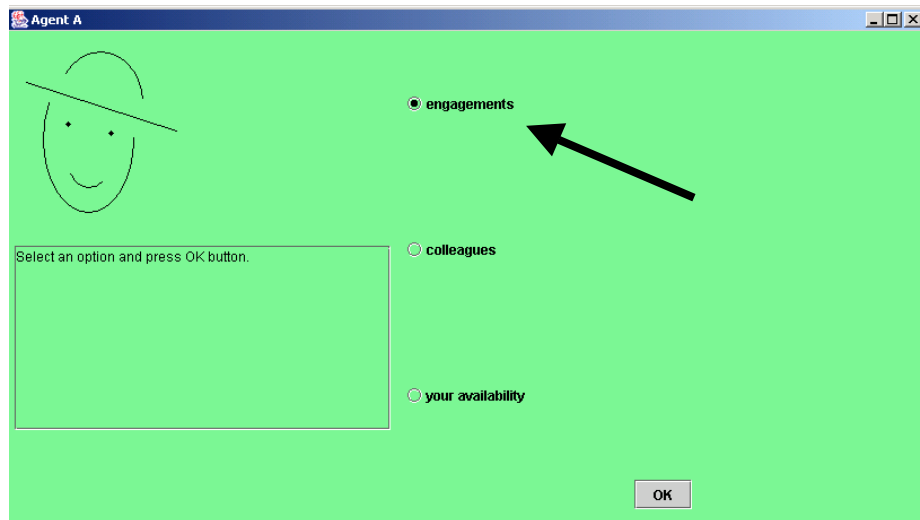


Figure 31 **engagements** radio-button.

Afterwards, a **new engagement** button has to be clicked.

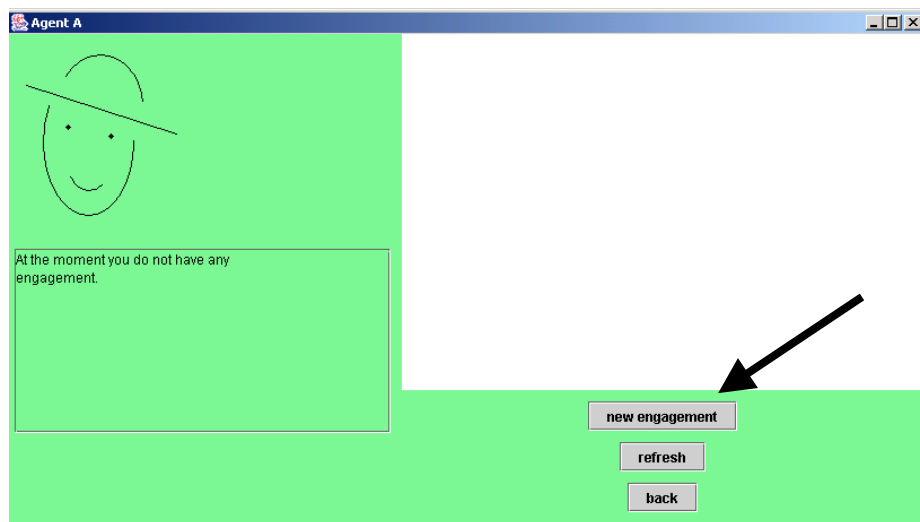


Figure 32 Currently there are no engagements.

A small dialog window appears.

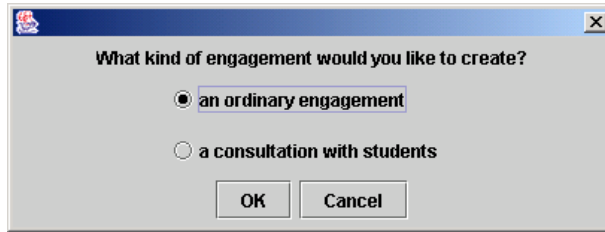


Figure 33 A window for the engagement type specification.

After selecting **an ordinary engagement** radio-button, the following dialog window pops up.

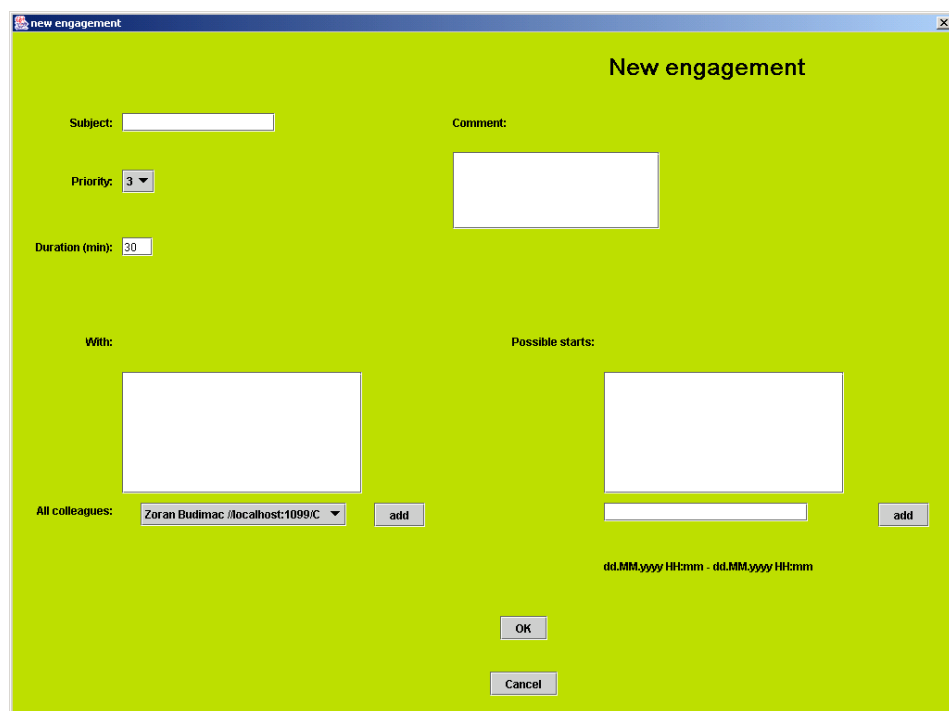


Figure 34 New engagement window.

Input elements in the above window can be filled out as in the figure below.

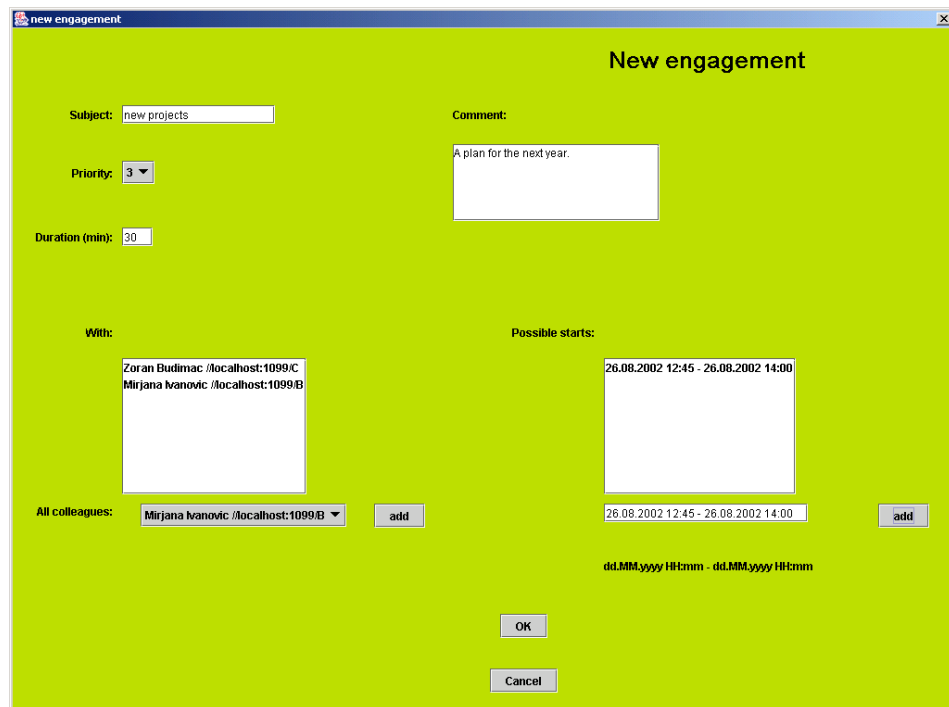


Figure 35 New engagement window after entering the engagement data.

If an intersection of available times of the invited persons in specified interval can be found, the engagement will be established. The owners of the agents **A** and **B** are notified using the status bar of their agent windows.

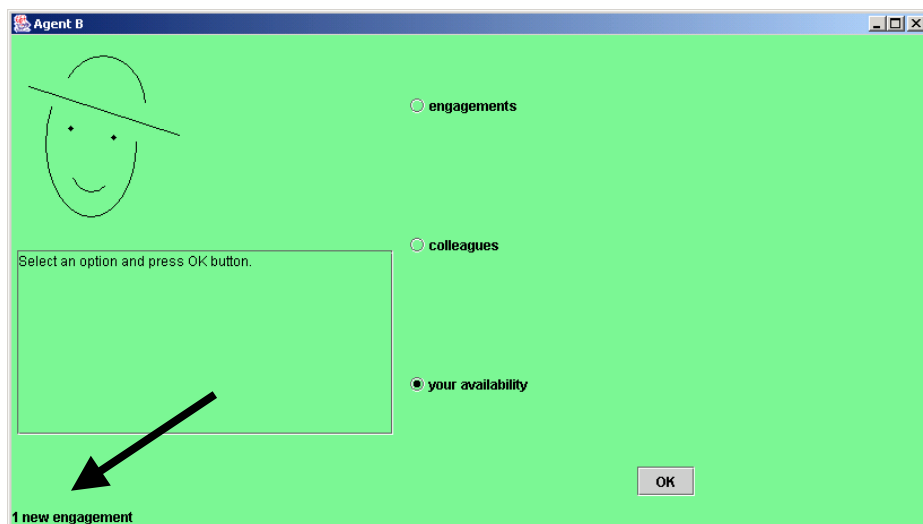


Figure 36 Agent **B** displays the new information in the status bar of its window.

The owner of the agent **B** can edit the engagement by selecting the **engagements** radio button, clicking on the **OK** button, and then clicking on the **edit engagement** button.

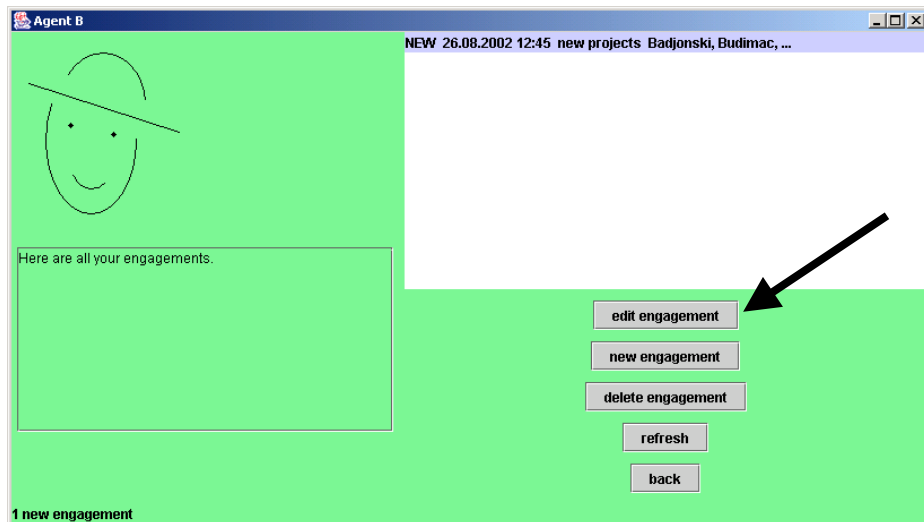


Figure 37 Engagements list of the Agent B.

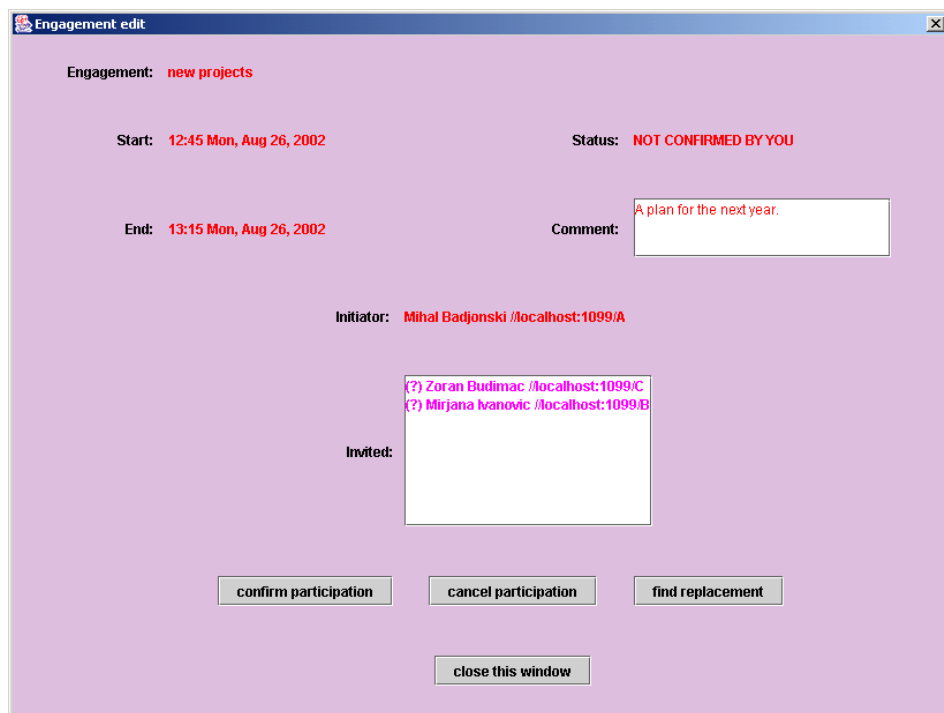


Figure 38 The owner of the agent B (Mirjana Ivanović) edits the new engagement.

Because of the fact that Zoran Budimac and Mirjana Ivanović have not yet confirmed their participation, the question marks appear near their names in the above window.

An invited participant can:

- confirm the participation,
- cancel the participation,
- find the replacement.

In all cases the other involved agents are informed about the new status of the engagement. They inform their owners about the engagement update using their status bar.

### Access via Internet Browser

AJA Agents can be accessed via World Wide Web (WWW). An agent in the example MAS allows students to register themselves via WWW for the consultations by the agent owner.

For example, the owner of the agent **A** defines two consultations. The first one takes place on August 27<sup>th</sup>, 2002 at 10:00.

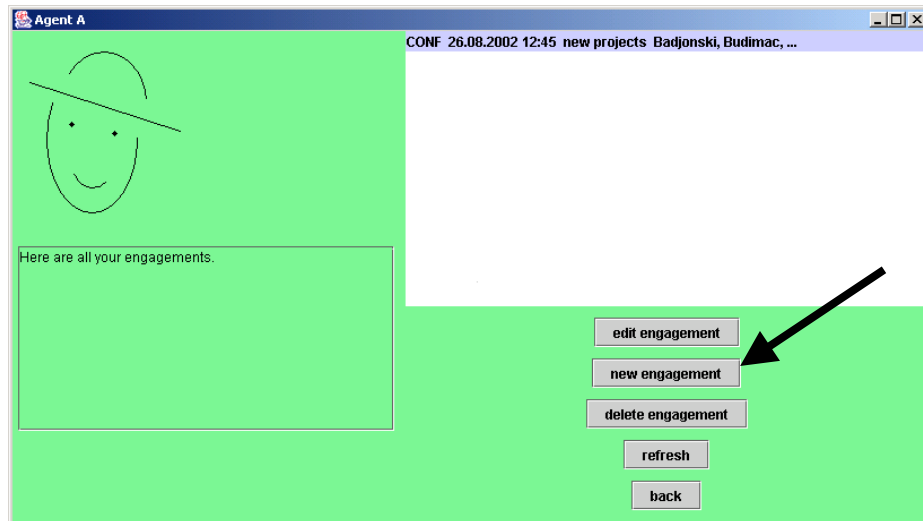


Figure 39 The first step in creating of a consultation.

After clicking the **new engagement** button, the radio button **a consultations with students** has to be selected.

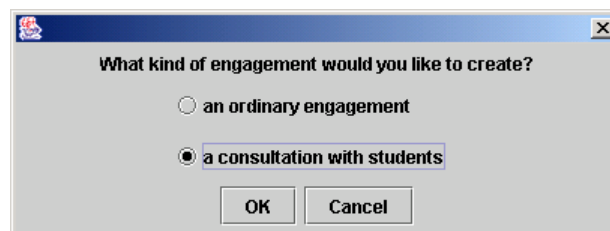


Figure 40 The second step in creating of a consultation.

A new window appears, where the consultation starting time has to be specified.

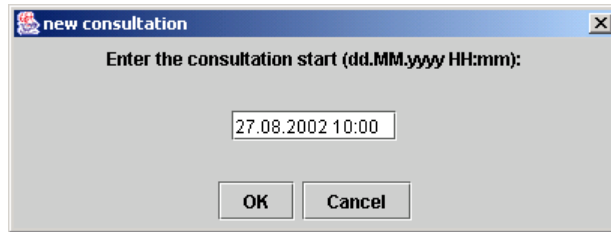


Figure 41 The third step in creating of a consultation.

After clicking **OK**, the consultation is created.

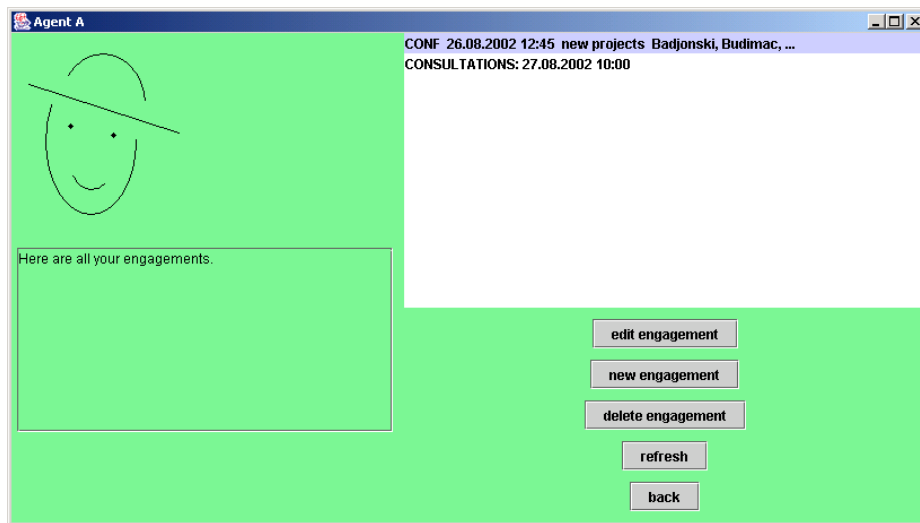


Figure 42 A consultation is created.

The second consultation date is, for example, August 30<sup>th</sup>, 2002 at 14:00. Repeating the above step, it can be added to the list of engagements.

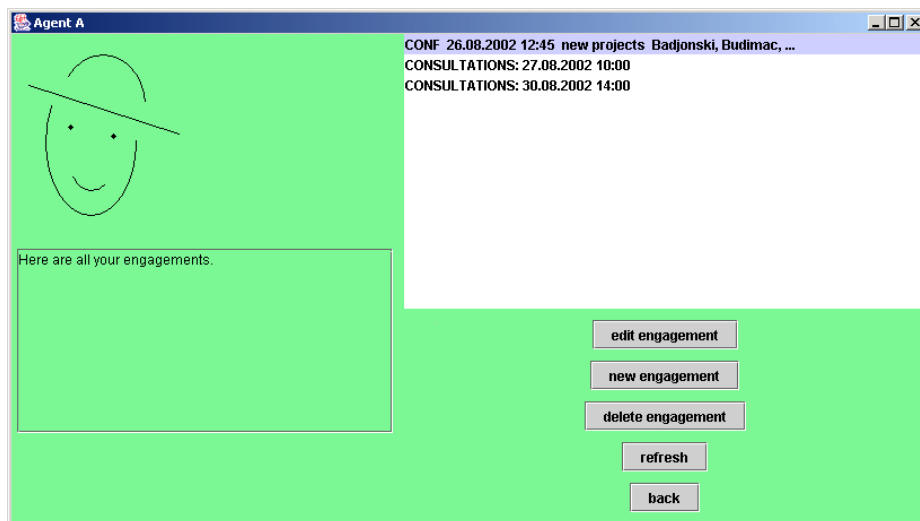


Figure 43 The second consultation is created.

The HTTP address of the agent **A** is <http://localhost:2100/A> . A student can use this address to access the agent over the Internet.

Of course, in the real life the computer name would not be **localhost**.

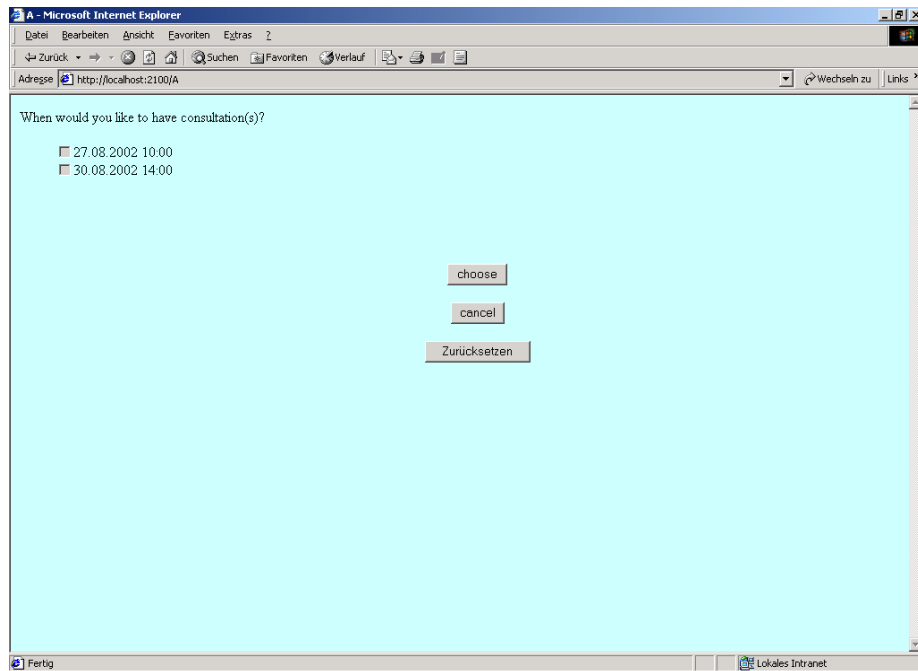


Figure 44 The first html page in the agent-student dialog.

A student chooses e.g. the first consultation date.

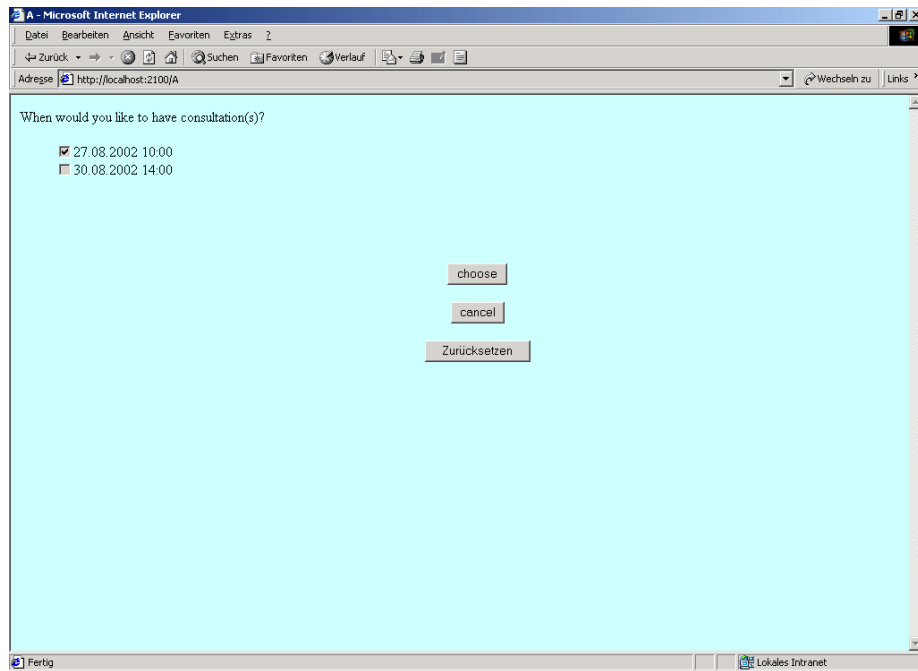


Figure 45 A check-box is selected.

After clicking **choose** button, a new web page is shown.

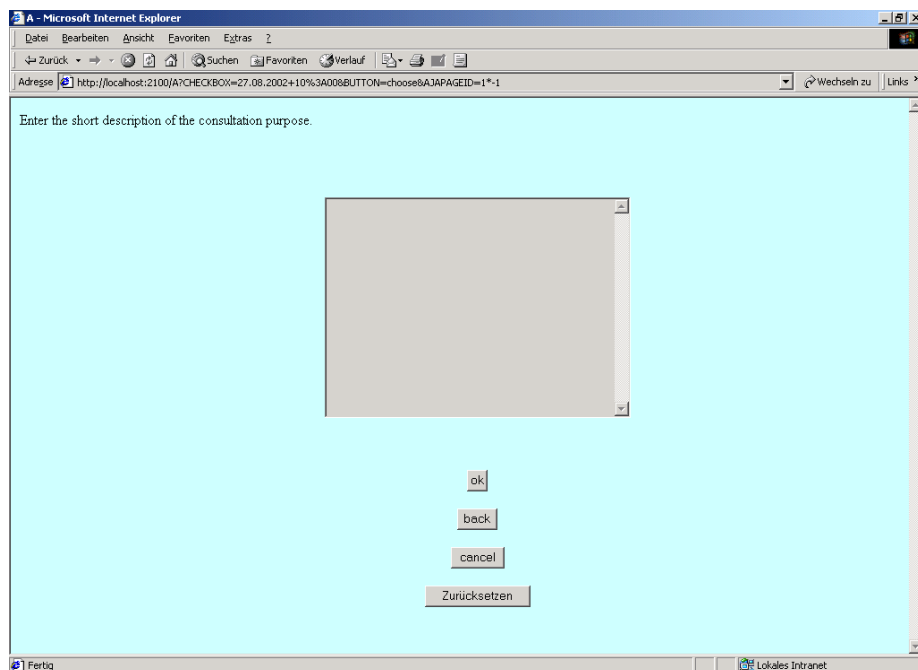


Figure 46 The second page.

A student enters the short description of the consultation purpose and clicks **ok** button.

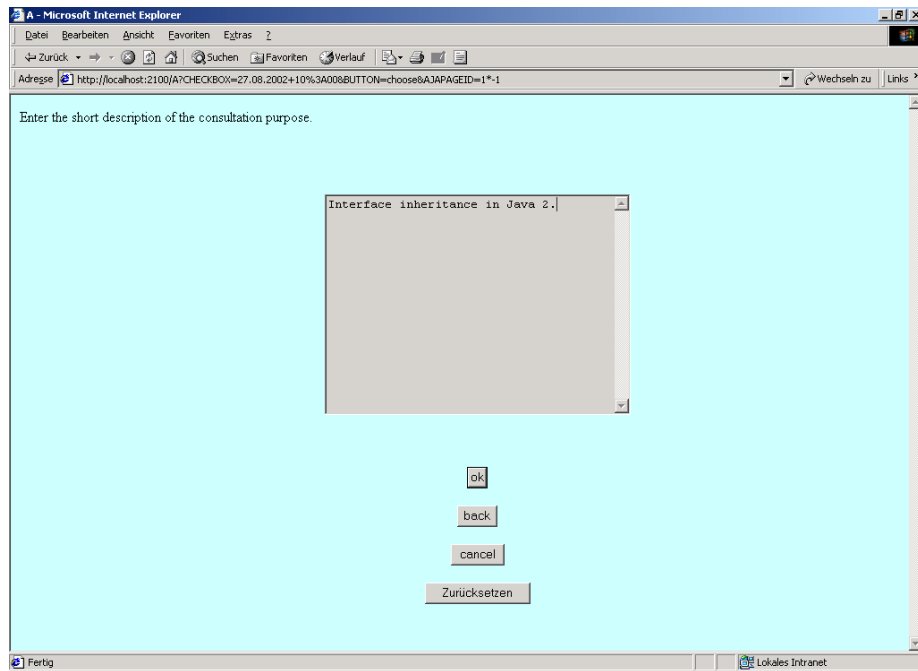


Figure 47 The purpose of the consultation.

A new page is shown.

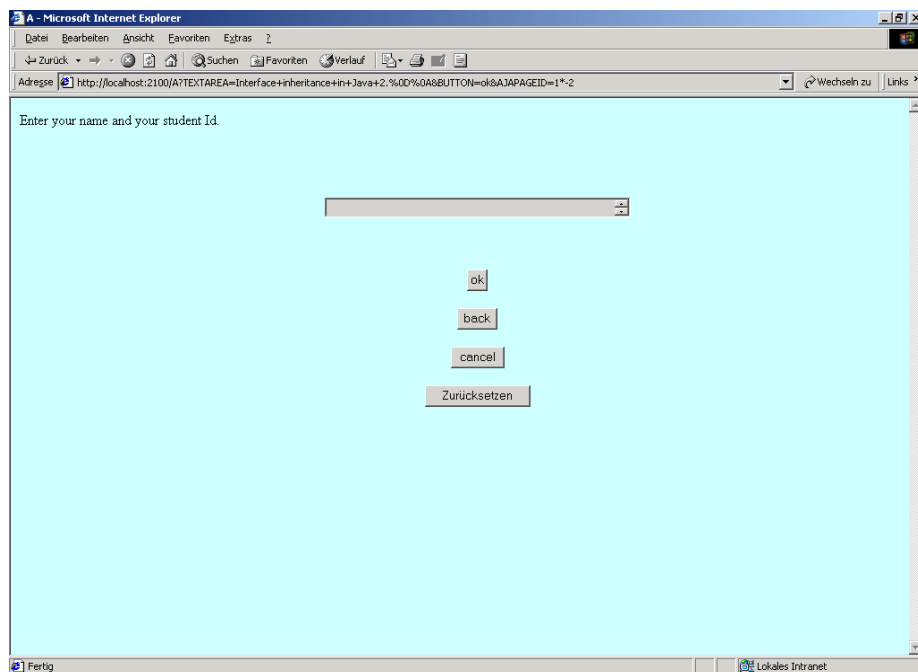


Figure 48 The third page.

A student enters her/his name and student Id.

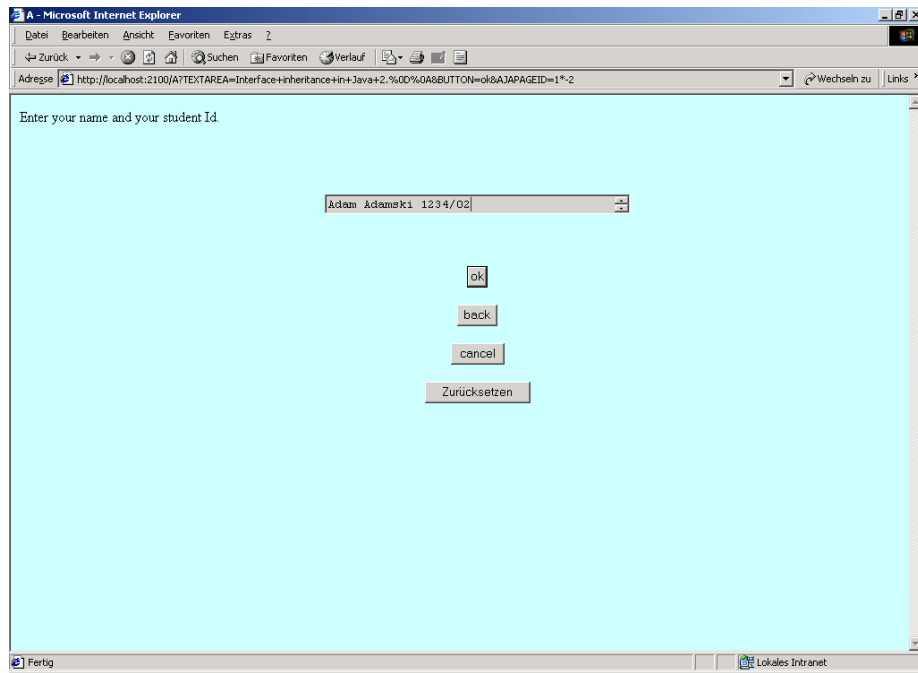


Figure 49 Student name and Id.

After clicking **ok**, the last page is shown.

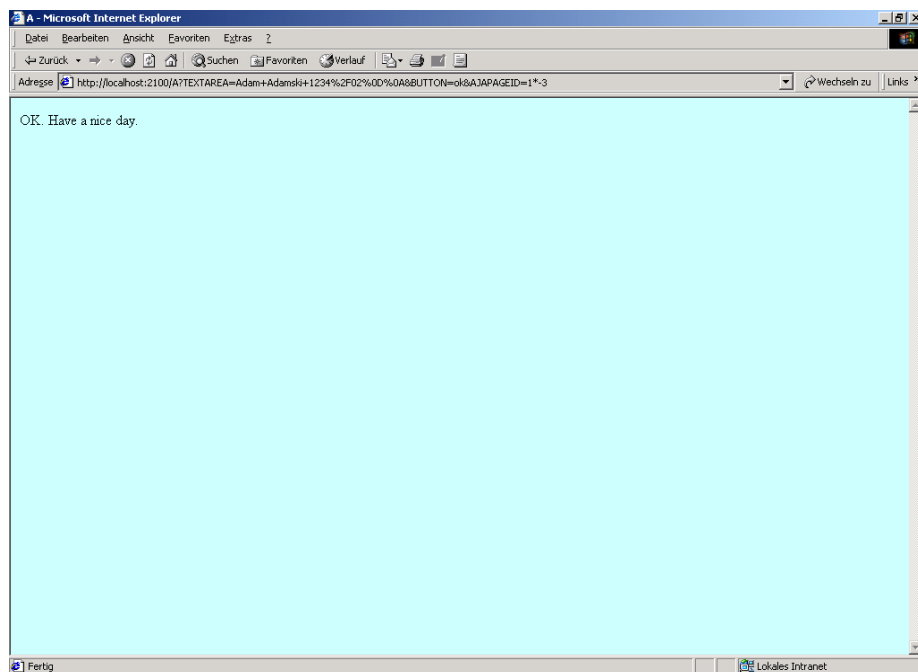


Figure 50 The last page.

The owner of the agent **A** can edit the consultation in agent window. The names of all students appointed for the consultation are displayed. In our example there is only one student appointed for the consultation.

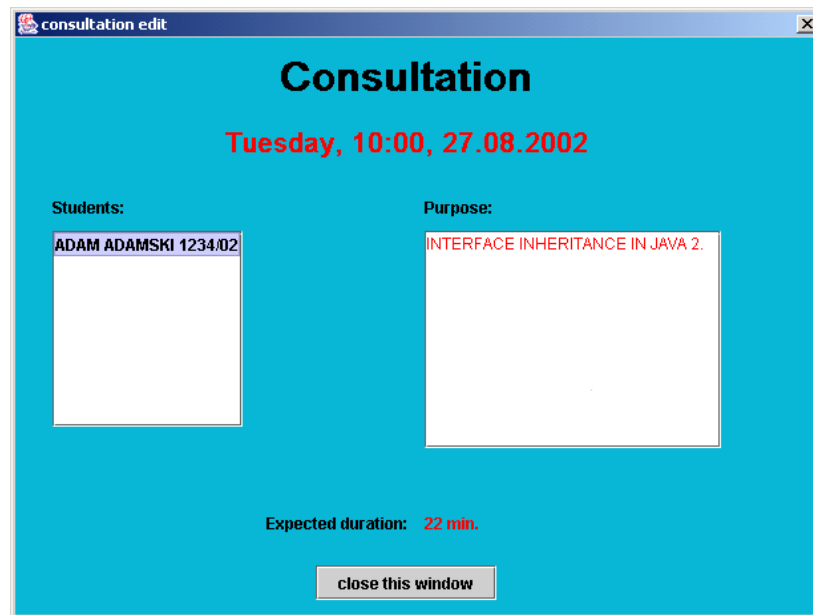


Figure 51 Editing the consultation.

### Other features

Besides the features mentioned above, an example agent has some additional features. Some of these features are shortly described below:

- Agent owner is notified, when a colleague has a birthday. The notification occurs a day before the birthday as well as at the day of the birthday.
- Duration of a consultation with students depends on two values:
  - number of appointed students, and
  - number of days before the next exams (the sooner the exams, the longer the consultations).

Duration of a consultation is therefore a dependant belief. The value of this belief is computed using neural network.

- Agent owner is alerted before the engagement start. Nevertheless, some user would like to be alerted sooner and some other earlier. The exact number of minutes before the engagement start, when the alert window is created, is computed using an adaptable parameter. Initially, the parameter has a value fifteen minutes. During the agent life its owner gives the feedback to the agent regarding the alert timing. Using this feedback (i.e. 'sooner' or 'later'), the adaptable parameter adapts its value to its user preferences.
- Agent reflexes are used to store the agent beliefs into the file after they have been changed. Although an agent should never be stopped, eventually it will be stopped. After the new start of the agent, it reads its belief from the file and continues to work.