Expert System for ALICE experiment

Use AI::ExpertSystem;

Arsen Hayrapetyan
ahairape@mail.yerphi.am
Artem Harutyunyan
hartem@mail.yerphi.am
Presented by: Predrag Buncic

About expert systems

A type of application program that makes decisions or solves problems in a particular field by using knowledge and analytical rules defined by experts in the field. The system mimics an expert's reasoning process when he or she makes decisions concerning particular problem.

An example of expert system: MS Windows operating system troubleshooting software, designed to provide solutions and suggestions to problems which the user may face throughout using the OS

Possible use case in Alice

E.g., it could be used by offline shifters to identify and troubleshoot the problems related to different parts of the system being monitored and controlled (shuttle operation, raw data registration, data transfer to T1s, etc.)

The expert system reduces the number of cases in which the shifter should contact the experts directly but *does not eliminate them*

If the information in the knowledge base is not sufficient for providing a definitive answer, the issue should be presented to the consideration of corresponding expert

How the system works

Coordinator/dispatcher

1. Asks for first question
3,5,...Forwards agent's answer

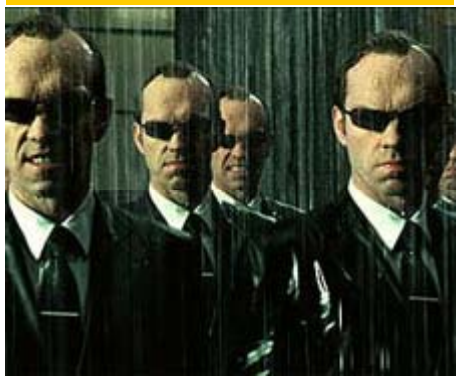
2,4,.... Forwards question/gets the status of the system component

Knowledge base

Asks question or provide final conclusion

Logs the action

Agents



Logger

Log the action

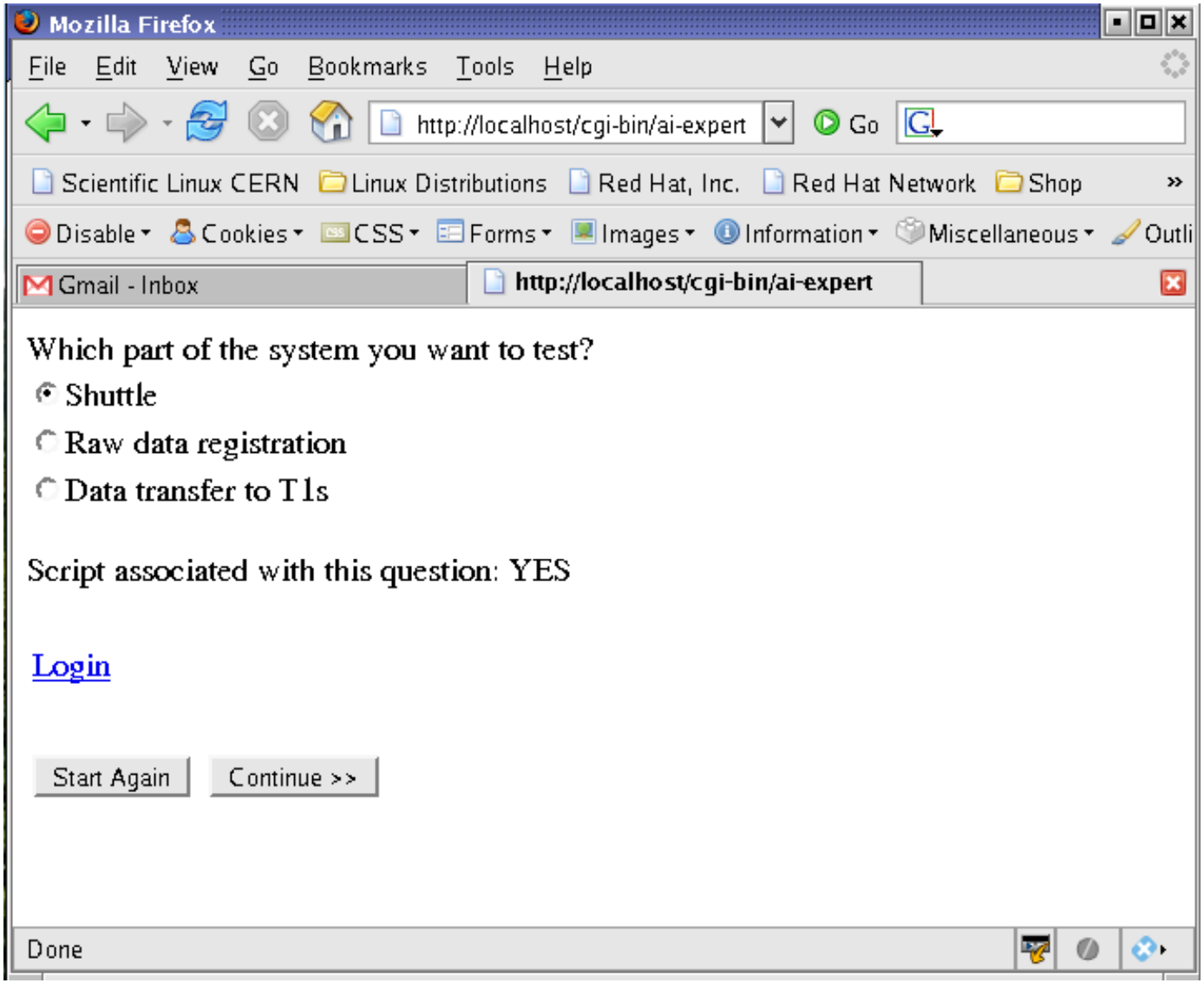
Review of the results

Logger

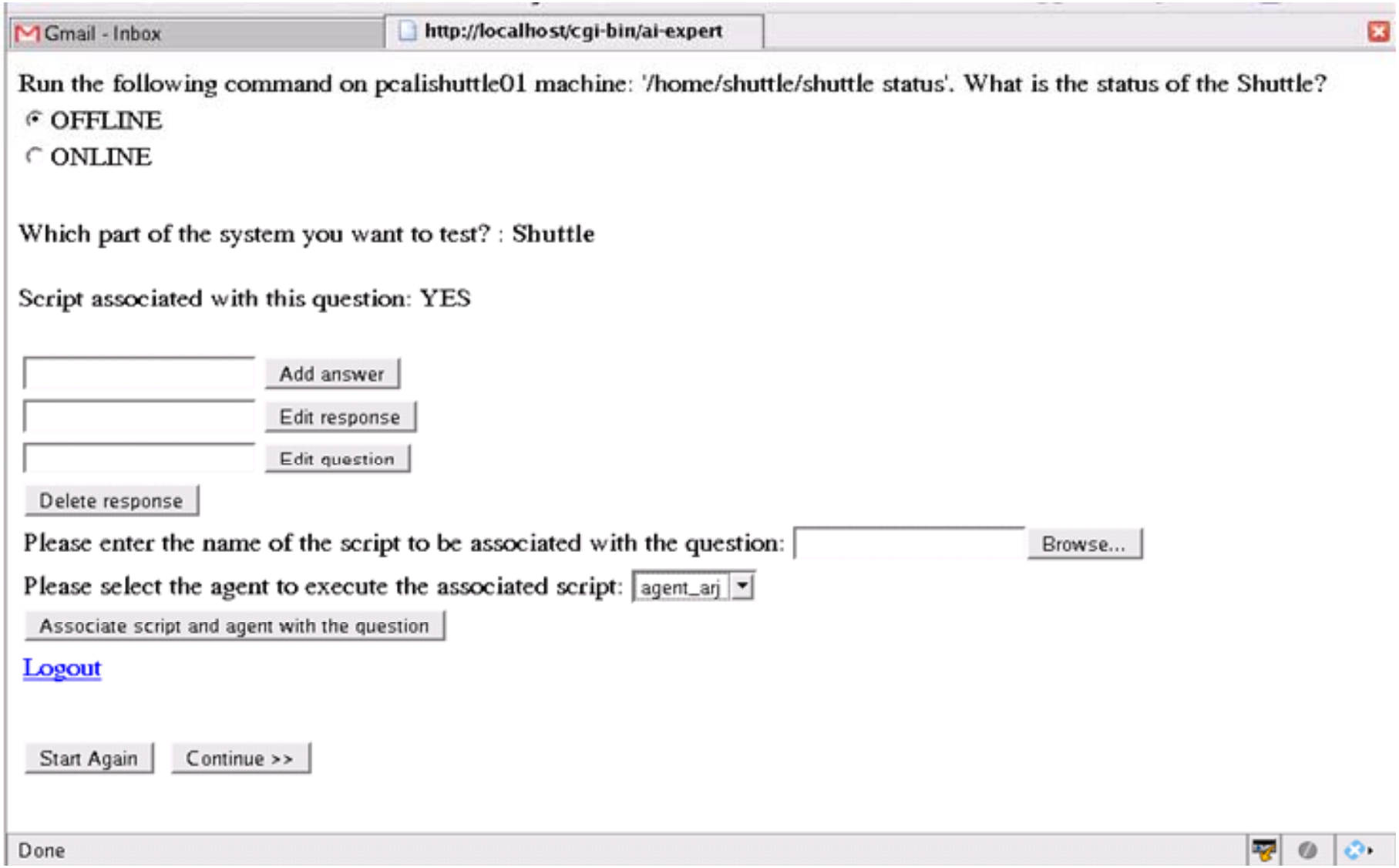
Reads logged messages on her/his terminal

Shifter and/or expert

User interface to the knowledgebase



Expert interface to the knowledgebase (adding/modifying information)



Logging


Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[pcepalice66] /home/ahairape/XMPP_scripts > /opt/alien/bin/perl global_logger.pl
Server [pcalientest04.cern.ch]:
Port [5222]:
Username [global_logger]:
Password [testpass]:
client_logger: Wed Mar 26 20:14:09 2008 : Successfully connected to jabber server
client_logger: Wed Mar 26 20:14:11 2008 : Sending the content of the file with following GUID to agent user6:
6a2dd7c4-fad4-11dc-aed6-0011117fc4ac
client_logger: Wed Mar 26 20:14:12 2008 : Sending the content of the file with following GUID to agent user7:
36162798-f75a-11dc-a03b-0011117fc4ac
client_logger: Wed Mar 26 20:14:13 2008 : Sending the content of the file with following GUID to agent agent_arj:
42116e26-f75b-11dc-ae47-0011117fc4ac
agent_arj_logger: Wed Mar 26 20:13:22 2008 - Got message from client
agent_arj_logger: Wed Mar 26 20:13:22 2008 - Info: Signature verification OK
agent_arj_logger: Wed Mar 26 20:13:23 2008 - Info: Trying to execute the script
agent_arj_logger: Wed Mar 26 20:13:24 2008 - Info: Script returned '1'. Returning '1' to client
client_logger: Wed Mar 26 20:14:17 2008 : Broadcasting the following message to agents:
6
Shell
Shell No. 2
Shell No. 3
Shell No. 4
Shell No. 5

Summary

The expert system

✓is quite easy to *install* (written in Perl, almost all the dependencies are resolved in the current AliEn version (2.14)) and *configure* (components of the system can be tuned via Apache-style configuration files)

✓ has a simple yet powerful communication mechanism for its components: **coordinator**, **agents** and **logger** are communicating over Extensible Messaging and Presence Protocol (XMPP aka jabber); **coordinator** and **knowledgebase** communicate over HTTP

✓ allows retrieving the status information via network of agents: each question in **knowledgebase** should be associated a script which will be executed by an agent to find out the answer to that question

✓ has a basic authentication mechanism: messages sent by **coordinator** to **agents** are digitally signed with an asymmetric key