

A Multi-Agent Architecture for Sharing Knowledge and Experimental Data about Waste Water Treatment Plants through the Internet

Eugène J.H. Kerckhoffs *
Department of Technical Informatics
Faculty of Technical Mathematics
and Informatics
Delft University of Technology
the Netherlands
email: eugene@kgs.twi.tudelft.nl

Hans L.M. Vangheluwe
Department of Applied Mathematics, Biometrics and
Process Control
Faculty of Agricultural
and Applied Biological Sciences
University of Ghent
Belgium
email: Hans.Vangheluwe@rug.ac.be

Abstract

In this paper we present a first prototype of a multi-agent architecture for the sharing of knowledge and experimental data about waste water treatment plants through the Internet, using WWW protocols. Applying a net browser such as Netscape, a user currently has access to advice on waste water cleaning technologies (a CLIPS expert system) and to files containing experiment data. The local prototype discussed in this article is part of a proposed global agent architecture.

1 Introduction

The environmental impact of products and processes affects everyone: raw materials, manufacturing processes, and the use and disposal of products have environmental consequences and costs. Collaboration to share information about environmentally sound products and processes is therefore vital to us all. The primary mission of the CERES Global Knowledge Network (GKN) initiative is the creation of a loosely interconnected, globally distributed, and locally administered set of knowledge bases on environmentally sound product development processes. The GKN should be accessible through a variety of user interfaces to accommodate the diverse user community. A prototype of the network is currently under development by a consortium of universities, research laboratories, and industrial organizations around the world. Consortium members work independently to create local knowledge sources and access mechanisms through funding from their own governments and local industrial sponsors and users. The CERES-GKN Foundation has board members from universities and research institutes in Belgium, Germany, India, Italy, Japan, the Netherlands, Russia, the U.K., and the U.S.

*Currently part-time professor at the University of Ghent

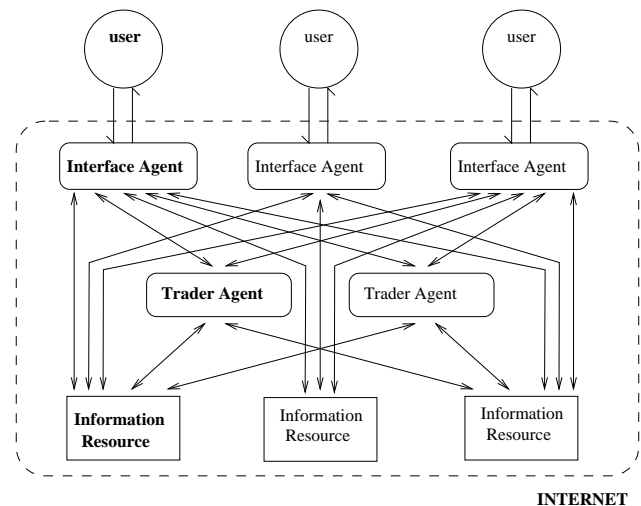


Figure 1: Proposed CERES Global Knowledge Network Meta-Architecture

Since the information revolution there has been a rapidly increasing number of information services by far exceeding the retrieval and consumption of that information. It may take several hours or even days to get the information, that is if at least it was stored in some machine.

Computers currently respond only to what interface designers call direct manipulation. Nothing happens unless a person gives commands from a keyboard, mouse or touch screen. The computer is merely a passive entity waiting to execute specific, highly detailed instructions; it provides little help for complex tasks. In the future, rather than manipulating a keyboard and mouse, people will speak to agents in order to get something done. In response, agents will appear as "living" entities on the screen, conveying their current state and behavior with animated facial expressions or body language rather than windows with text, graphs and figures [Mae95].

2 Proposed CERES-GKN Global Agent Architecture

In Figure 1 we present a conceptual model for the proposed global level agent architecture of CERES-GKN. The model presents three main modules (Interface Agent, Trader Agent, Information Resource) and their interrelations. The user is ignorant of modules other than the Interface Agent; all interaction between the user and the Internet goes through the Interface Agent. However, it should be noted that the user shown in the figure is not restricted to human users, any other module in the model can be a user. For example, a Trader Agent can access other Trader Agents by becoming a user and going through an Interface Agent.

Interface Agent: The Interface Agent must assist the user in browsing through the Internet. The help can be in two ways. First, several pages of information could be browsed ahead in the background. Second, possible interesting pages from those browsed would be presented to the user. To be able to do that the Interface Agent must have some knowledge of both the interests and environment¹ of the user. When the Interface Agent is accessing some Information Resource it is said to be in the Discovery Phase. Machine learning techniques could be used to construct users interest models [KMK96]. However, at the global level building such agent is very hard, due to the fact that it has to deal with a wide range of knowledge domains. The solution is to delegate some responsibility to Interface Agents at the local level where the domain of knowledge is more uniform. The Interface Agent has to be up-to-date with recent changes in the Internet; for that it has to contact the Trader Agent.

Trader Agent: The Trader Agent should help Interface Agents in identifying possible interesting Information Resources. When the Interface Agent consults a Trader Agent it is said to be in the Navigation Phase, a phase in which the Interface Agent navigates and selects information sources but does not access them. The Trader Agent must have up-to-date meta-knowledge about the local level Information Resources. The difficulty in building such agent is the way in which the meta-knowledge should be organized (i.e., building a Taxonomy) and represented. Search mechanisms have to be developed.

Information Resource: The Information Resources at the global level are actually the local level multi-agent architecture (see section 3). At the local level any kind of information system can be built, as long as it is

¹The computer environment and software/hardware capabilities.

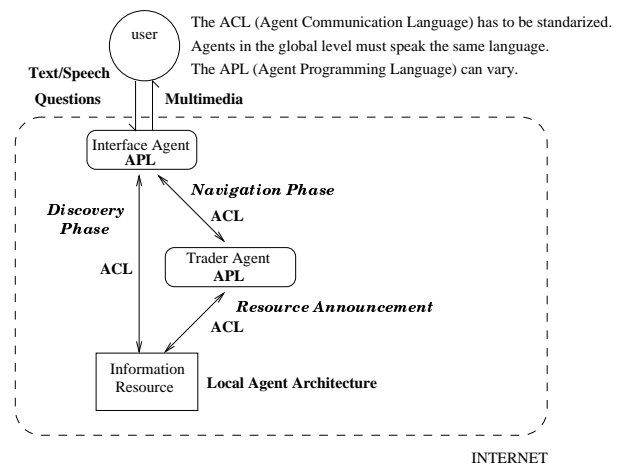


Figure 2: Details of proposed CERES Global Knowledge Network Meta-Architecture, presented in Figure 1.

accessible through one of the Internet service protocols. Most local levels however will be made accessible through the WWW. This Information Resource is the most dynamic module in the global agent architecture. A lot of Information Resources could be created, others changed or removed. Therefore, it is up to the builders of the Information Resources to keep at least one Trader Agent up-to-date with their latest situation, a phase called Resource Announcement (see Figure 2).

The agent characteristic of social ability implies that agents need some appropriate agent communication language (ACL) to communicate with each other. A standardized ACL facilitates the interoperability between agents [GSS94]. An ACL should be syntactically simple (readable by people), semantically described, efficiently implementable and should support reliable and secure communication among agents. With KQML and KIF communication is a declarative approach. Agents can also communicate by the exchange of procedures. This is currently a popular approach due to the popularity of scripting languages like Tcl. A scripting language is an interpreted language. It is less platform dependent because it doesn't have to be compiled.

3 CERES-GKN Local Agent Architecture

The local level architecture can be as simple as a few static web pages linked together, or a bit more complex as CGI programs interfacing some database query systems. What we recommend is a more advanced local architecture, see Figure 3. The CERES-GKN local agent architecture would be an instance of the class of global level architecture (to borrow the Object Oriented way of thinking) with the Me-

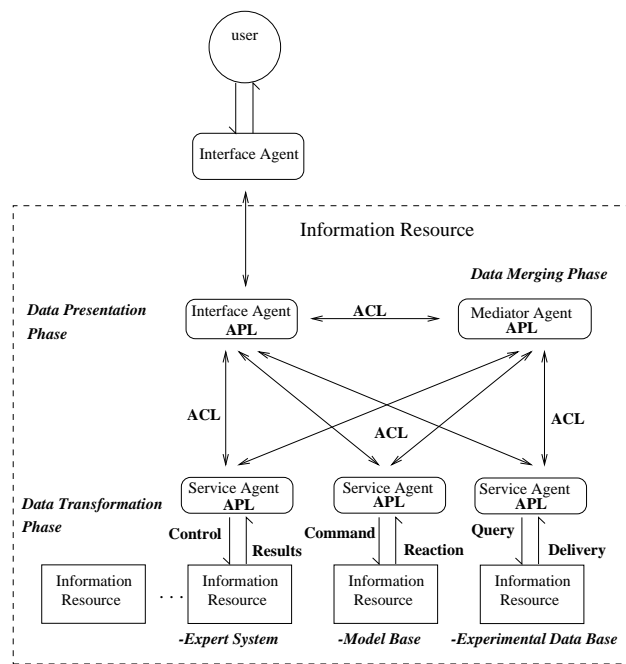


Figure 3: Proposed Generic Local Level Agent Architecture

diator Agent over-riding the Trader Agent and the addition of the Service Agent. In other words the local level architecture could be seen as a more detailed and specified copy of the global level architecture.

Interface Agent: At the local level the Interface Agent would be more aware of what its information sources have to offer and less aware of the user's needs and environment. Negotiations between the two Interface Agents at the global and local level would result in some common ground arrangement. Then the local Interface Agent could offer the wanted information in the proper format, a process named Data Presentation Phase. However, the result of the negotiations could be negative, saying that the user needs could not be satisfied. The Interface Agent has to go to the Mediator Agent for advice on what the local level architecture has exactly to offer as a single entity.

Mediator Agent: Since in the local level the domain of knowledge is unified, merging of information is now possible. Data Merging Phase is the phase in which different pieces of information semantically related are drawn from different local information resources and merged into one. This is a task of the Mediator Agent. Moreover, in the suggested local agent architecture the Mediator acts as Trader Agent as well. It helps the Interface Agent to navigate the local Information Source instances.

Service Agent: The Service Agent has the responsibility of interacting with an application specific Information

Resource. Direct manipulation, and information format translation is a must if the Information Resource is to be accessible through a Phase of Data Transformation. The Service Agent can be seen as agentifying the Information Resource. The way is to interface between existing information resources and the other agents, accepting messages from agents, translating them to a native communication protocol, manipulating and commanding, accepting responses from the information resource, and sending results back to other agents. This has an advantage of taking information resources as they are (legacy) with no changes to be done.

Information Resource instance: The Information Resource instance can be any kind of data or knowledge base, it can be a static text file in ASCII format, SQL database, executable C/C++ program, simulation program, or AI expert system. As long as knowledge about their communication behavior and protocol is known to the Service Agent, all kinds of systems can be included.

4 The CERES-EXP Local Agent Architecture

CERES-EXP is a subproject of CERES-GKN focussing on a prototype local multi-agent architecture for the sharing of knowledge and experimental data about waste water treatment plants through the WWW. CERES-EXP is undertaken by the Delft University of Technology (the Netherlands) and the University of Ghent (Belgium). CERES-EXP should result in an experimental version of "CERES-GKN in-the-small" in order to gain sufficient experience with agent methods and techniques; this experience will be a basis for further developments in the framework of CERES-GKN.

We aim at an environment as depicted in Figure 3, where the Information Resources are expert system, simulation model base, and experimental data bases in the field of waste water cleaning. The Information Resources are distributed and stored in computers at Delft University of Technology and the University of Ghent.

At the moment of writing this paper only simple experimental data bases (in the form of ASCII text files) and a small expert system (advising on the kind of cleaning technologies that should be used in a specified situation) are operational. The rest of the paper will focus on these; i.e., the architecture of Figure 3 in which an expert system and ASCII text files (with experimental waste water treatment plant data) are included as Information Resources, the Service Agent and Interface Agent included as indicated, and

for the time being the Mediator Agent is a simple switch.

5 AQUA-clean: A Prototype for Selection of Cleaning Technologies

AQUA-clean is based on the PROPER-method (Pollution Reducing Options in Processes: an Efficacy Ranking method), see [Els95], used to select an appropriate combination of cleaning technologies for a network of waste water streams. The prototype consists of three main components: the Interface Agent, the Service Agent, and a CLIPS expert system that implements the PROPER-method. The access to the prototype is through a graphical Internet browser.

The PROPER Expert System

Often plants produce pollutants which are drained into the surface water. Because of restrictions imposed by the government the amount of pollutants in the end-of-pipe stream are to be reduced. One possibility is to take the end-of-pipe stream and concentrate and destruct the polluting substances. A more sophisticated approach is to use cleaning technologies as part of the production process. To assist in this approach, an expert system is under development at the Delft University of Technology.

The expert system takes as its input a description of a network of waste water streams and finds for every stream a cleaning technology so that the end-of-pipe stream is as clean as possible. The expert system is based on the aforementioned PROPER method. Instead of describing all the pollutants which are present in the waste water streams, PROPER only describes groups of pollutants. Every member in such a group has an identical chemical and physical behaviour. In PROPER the pollutants are called components and a group of components is called a pseudo-compound. This reduces the amount of information significantly, but it is still specific enough for the selection of cleaning technologies.

The streams in the network of waste water streams are each described by the following properties:

capacity,	expressed in liters per hour	[l/hr]
temperature,	expressed in celsius	[degrees C]
pressure,	expressed in bar	[bar]
acidity,	expressed in pH	[pH]
sludge,	whether the stream is sludgy.	

A stream may consist of several pseudo-compounds, represented by:

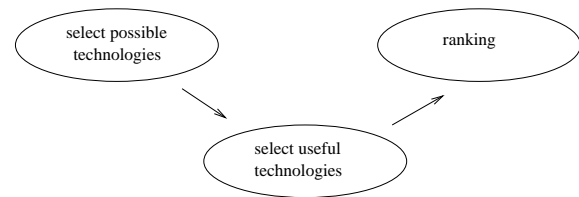


Figure 4: The PROPER Working Method

type of pseudo-compound (ionic, volatile, etc.)
concentration, in milligram per liter [mg/l].

The PROPER working method consists of three stages for any stream (see Figure 4). In the first stage, a selection is made of all the possible technologies which can be applied. Technologies which cannot be applied because of certain properties of the current stream, or because of the presence of a pseudo-compound which disturbs the working of the technology, are deleted. Then, a selection is made of technologies which operate on at least one pseudo-compound in the current stream. All other technologies will be deleted. Finally, a ranking of the selected technologies will be made, resulting in one technology being selected for the current stream. Some ranking criteria are efficiency of removal, amount of pseudo-compounds to be removed, costs of applying the technology, and acquaintance with the technology. Also a combination of several criteria can be used. Up to now it is only possible to choose a technology according to its efficiency of removal. The technology with the highest efficiency is selected for the current waste water stream.

For every stream, starting with the source streams, a technology is selected. Then, the technology is applied to the stream concerned, resulting in changes in the concentration(s) of the pseudo-compound(s) on which the technology operates. Whenever two streams join into one, all the properties of the resulting stream must be computed whenever technologies have been applied to both joining streams. After computing these properties, a technology can be selected for the joined stream. Finally, the end-of-pipe stream is reached.

Some additional features of the system include:

- *Editing the network:* There is the possibility to edit the network. Streams can be added or removed, and the expert system should find a solution for this changed network.
- *Selecting a different technology:* The user has the possibility to select a different technology for a certain stream.

The PROPER expert system has been implemented in CLIPS 6.0. In the framework of the CERES-EXP project

the expert system is made available to the World-Wide Web. For this, several agents are in development to guide the user in his effort to find the answers to his question.

The Interface Agent

To make the PROPER expert system accessible to the Internet, in addition to the Service Agent discussed in next section, an Interface Agent has been implemented which is in charge of handling the user's needs and connecting him to the Service Agent. The following functionalities are included:

- The Interface Agent provides fill-in forms to interact with the user; currently, this interface is based on HTML.
- The Interface Agent gives the user the possibility to present property values of the waste water stream(s); it deals with unknown property values by providing an advised range for these. Furthermore, it checks the validity of the values and gives feedback to the user.
- The Interface Agent also checks for the consistency of the entered waste water stream network; this network must converge to one final (end-of-pipe) stream. A graphical drawing of the waste water streams network is provided to the user (see Figure 5).
- The Interface Agent offers the user the possibility to ask for more information about the selected cleaning technologies. For example, the user could be interested to know whether there are experimental data available for the selected cleaning technology, requiring eventually a link to the H2O (see section 6).

The PROPER Service Agent

The Service Agent (see Figure 3) is responsible for interaction with the knowledge source. The PROPER Service Agent is an agent for controlling the afore-mentioned expert system written in CLIPS. It has the following basic functionalities (see Figure 6):

- Transform the data received from the Interface Agent to the format used by the PROPER expert system.
- Control the execution of the expert system.
- Transform the answers given by the expert system to a data format understood by the Interface Agent.

The Interface Agent stores all the data about the waste water network in an array. In order for the PROPER expert

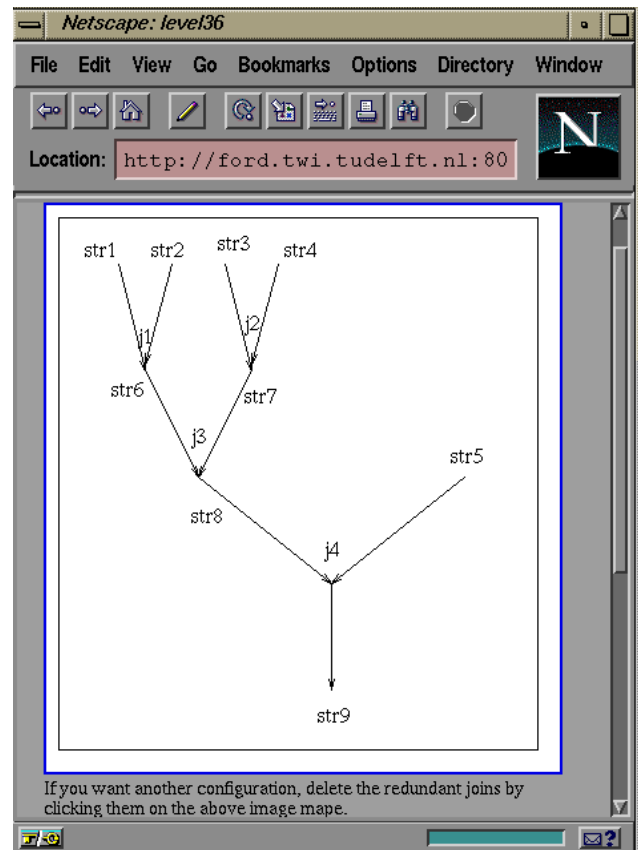


Figure 5: AQUA-clean Netscape View

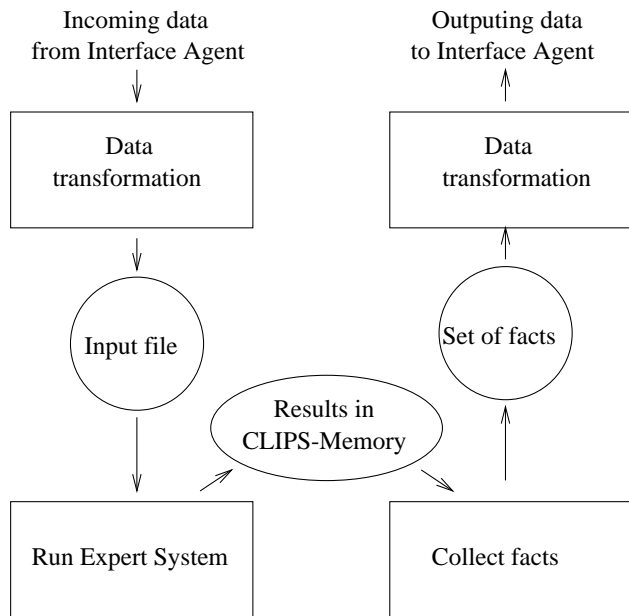


Figure 6: The Service Agent Functionalities

system to work on the data, these must first be transformed to an input file. This input file is stored in a special directory. Then, the expert system is started, given the saved input file. When the expert system has finished reasoning and constructing an answer, the Service Agent has to search through all the facts stored in the CLIPS Memory for the facts which describe the wanted answer. These facts are stored in a temporary array, and then transformed into an array which can be read by the Interface Agent.

The Service Agent has been programmed in Tcl 7.3 in which CLIPS 6.0 was embedded. All communication is done by Remote Procedure Calls implemented in the Tcl-extension Tcl-DP.

An Example Session

The user has to input a waste water network, with all the properties of the various source streams. This is handled by the Interface Agent. For the considered example, a network has been chosen with two source streams joining into one end-of-pipe stream. All properties of the two source streams can be entered in a form presented by Netscape (see Figure 7). When pressing the "Run ES" button the data are transferred to the expert system, and the results are transferred back and printed on the screen. As an illustration, in Figure 8 the data of the end-of-pipe stream is presented; also the time it took to construct an answer, the reasoning time of the expert system, is printed.

The screenshot shows a web form for inputting source stream properties. The fields are as follows:

- Stream name: str4
- Temperature [C]: 45
- Pressure [bar]: 2
- Capacity [l/hr]: 0.4
- PH: 7
- Pseudo-compound name: ps41
- State: ionic
- Concentration [mg/l]: 15

Buttons for 'Run ES' and 'Reset' are visible at the bottom.

Figure 7: Form-based input of source stream properties

The screenshot shows the reasoning results for stream str7. The reasoning time was 0.032 sec.

Information about stream: str7		
Stream name	str7	
Temperature	45.0 [C]	
Pressure	2.0 [bar]	
Capacity	1.6 [l/hr]	
PH	7.0	
Pseudo-compound name before application of technology	State	Concentration
gen385	ionic	0.075 [mg/l]
Technology	no-technology experiment based on no-technology	
Pseudo-compound name after application of technology	State	Concentration
gen386	ionic	0.075 [mg/l]

The reasoning time was: 0.032 sec

Figure 8: ES reasoning results

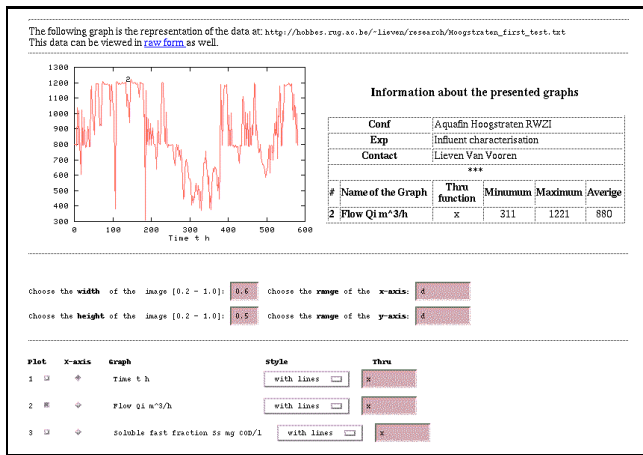


Figure 9: H2O Netscape View

6 H2O: A Prototype for the Sharing of Experimental Data

A prototype for the sharing and meaningful presentation of experimental data has been developed. The prototype is called H2O, Hypermedia access to models of waste water treatment plants (experimental data) [Cal96]. Compared to the set-up of Figure 3, for the time being the experimental databases are simply ASCII data files. H2O is built-up of two agents, a Trader Agent and a Mediator Agent; so compared to the set-up of Figure 3, the Mediator Agent (here supposed also to act as Trader Agent) is split up in a separate Trader Agent and Mediator Agent (for the time being there is no Service Agent to the Information Resource: experimental data files). These agents are implemented in Tcl [Ous94] and are accessible through the World Wide Web using a graphical Internet interface, such as Netscape.

H2O Agents

The Trader Agent enables users around the world to share experimental data. Users have the possibility to register new experimental data and they can select experimental data. The Trader Agent has a database that contains the header information of registered experimental data files. Registration of an experimental data file is done by providing the http-address² of an experimental data file. The Trader Agent then retrieves this experimental data file and adds all the header information of that file to the database. Selection of experimental data is done on the basis of these header information of registered experimental data. The data itself are not added to the database because there is no need to select experimental data on basis of these data and experimental data files can be very large which makes

²A http-address is an instance of the URL, it is the address of a document on the World Wide Web

it difficult to manage them in a database. The result of a selection is the http-address of the desired experimental data file. After a user has made a selection s/he wants to have access to these data. This is done by using the Mediator Agent.

The http-address of a selected experimental data file is input for the Mediator Agent. This agent consecutively retrieves the experimental data file and provides a meaningful presentation of these data. The experimental data will be presented as a graph and the user has the possibility to change the presentation according to his/her preferences. The user can:

- Change the size of the graph.
- Choose a particular range on both x-axis and y-axis.
- Choose which columns from the datafile have to be plotted in the graph.
- Choose which column represents the values on the x-axis.
- Plot a graph which is a simple mathematical expression of two other columns, e.g. plot the sum of column 2 and 3.
- Pipe all the values from a column through a function before they are plotted in the graph.
- Choose how the graph should be displayed: solid line, dashed line or impulses.

The Mediator Agent will also offer the user a hyperlink that points to the original numerical data file. On the basis of the user's preferences the Mediator Agent writes a plotting-script that is used by the plotting program *gnuplot* to draw the graph. The Mediator Agent in turn includes this graph in the document which is returned to the user.

Both the Trader Agent and the Mediator Agent are Common Gateway Interface (CGI) programs. The CGI is an interface between a World Wide Web server and an executable program. Such a program is addressable by means of an http-address. When the address is requested by a user (using a World Wide Web client), the program is run and the result of the program is returned to the World Wide Web client and presented to the user. It is possible to enter data at the client side³ and to send this data to the CGI program. In this way it is possible to provide information that is 'generated on the fly'. The agents are implemented in the interpreted language Tcl (Tool Command Language) [Ous94] and the Tcl-DP (Distributed programming) package. With Tcl, communication between software packages (in H2O between the World Wide Web server, a database

³This is done by using the FORM in an HTML document.

and a plotting program) is made easy. The Tcl-DP package is used by the Mediator Agent to retrieve an experimental data file on background.

Example of H2O User Session

Imagine a researcher, who works at a waste water treatment plant somewhere in the Netherlands, is interested in experimental data from a waste water treatment plant in Belgium. He starts Netscape on his computer and from the CERES-EXP home page he enters the World Wide Web document of the Trader Agent which enables him to select experimental data. On the basis of shown attribute/value pairs the researcher can give his preferences. The Trader Agent returns a list of http-addresses of available experimental data files from the Belgian waste water treatment plant concerned; from this list the researcher can make his final selection. Now the Mediator Agent is started-up automatically and the http-address of the selected experimental data file is used by the Mediator Agent to retrieve this data.

The Mediator Agent presents the data by means of a graph (see Figure 9). By default the Mediator Agent will plot column 2 of the data file against column 1. The researcher has the possibility to change the graphical presentation according to his preferences. He can change the size (width and height) of the image, he can choose a particular range on both the x-axis and y-axis, and he can choose which columns have to be presented graphically (check boxes on the bottom of the figure, not all shown). He also can choose which column of the experimental data file must represent values on the x-axis. And if the researcher wants to see the original experimental data, he can click on the raw form hyperlink, shown on the top of the document.

On the right side of the graph the header lines of the experimental data file are given. Whenever there is in one of the value parts a string that represents an http-address, this string will automatically be made a hyperlink when it is printed beside the graph. This hyperlink can be used to provide easy access to other information, e.g. to the home page of the experimentator or to context information of the experiment.

The researcher chooses to alter the range on the x-axis by typing [100:300] in the field after the text 'Choose the range of the x-axis'. He also prefers to see a bigger graph and alters the height of the image to 0.8 instead of the default value 0.5. He then clicks on the submit-button on the bottom of the document and waits for the reply of the Mediator Agent. The Mediator Agent returns a graphical presentation in which the range of the x-axis and the height of the graph are adapted.

Registration of an experimental data file can be done by providing the http-address of the data file. After clicking

on the submit button, the Trader Agent will return a document informing the user whether the registration was successful or not. If the registration is done successfully, the header information of that particular data file is added to the Trader Agent's database and other users have the possibility to select these experimental data on the basis of these header information.

7 Conclusion

The CERES-GKN project aims to provide all kinds of environment related information on the Internet to product developers and product designers to facilitate them to make their products more environmentally friendly. A subproject of CERES-GKN is CERES-EXP dealing with the sharing of knowledge and experimental data in the field of waste water treatment plants. In the paper the proposed general outline of the global and local multi-agent based architectures (in the framework of CERES-GKN) is presented, as well as a first prototype of the CERES-EXP environment that of course should fit the proposed general set-up. The CERES-EXP prototype currently consists of the H2O system (dealing with the sharing and meaningful presentation of experimental data from waste water treatment plant measurements) and the Aqua-clean system, an agent-based system with PROPER expert system advising on cleaning technologies to be used in specified situations. Both systems have been designed and built separately, and then merged to one system. The system described is a first prototype that needs of course more development and extension (e.g. inclusion of model bases as Information Resources). The prototype is currently running successfully and is being used and tested by researchers at the University of Ghent involved in experiments on waste water treatment plants.

Acknowledgment

The authors would like to thank their former students G. Frenk, F. Faik, and especially W. van de Calsijde and Raid I.K. Abu Yaman for the fruitful contributions to the research described in this paper. R.A. Vingerhoeds is gratefully acknowledged for providing the PROPER expert system developed under his supervision.

References

- [GSS94] Michael Genesereth, Narinder Singh, and Mustafa Syed, *A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation*, Communications of the ACM, Volume 37, No. 7 (July 1994).

- [Mae95] Pattie Maes, *Intelligent Software*, Scientific American, Vol. 273, No. 3, pp. 84-86, Scientific American, Inc., September 1995.
- [EW94] Oren Etzioni and Daniel Weld, *A Softbot-Based Interface to the Internet*, Communications of the ACM, pp. 72-76, Volume 37, No. 7 (July 1994).
- [Els95] Ester Elsinger Selection of cleaning-technologies, MSc. report, Delft University of Technology (Faculty of Technical Mathematics and Informatics), 1995.
- [Ous94] John K. Ousterhout. *Tcl and Tk Toolkit*, Development Group, 1995.
- [Cal96] Wijnand van de Calseijde Agents in a Global Knowledge Network, MSc. report, Delft University of Technology (Faculty of Technical Mathematics and Informatics), 1996.
- [KMK96] H.C.M. de Kroon, T.M. Mitchell, E.J.H. Kerckhoffs, Improving learning accuracy in information filtering, in: J. Herrmann, V. Moustakis (Eds.): Proce. of Pre-Conference Workshop on "Machine Learning meets Human Computer Interaction" (HL meets HCI), ICML'96: 13th Int. Conf. on Machine Learning, Bari (Italy), July 3-6, 1996.