

R&D COLLABORATION PORTAL AND COMMUNICATION PLATFORM FOR REMOTE ROBOT CONTROL

Florin Daniel ANTON¹

Lucrarea descrie un proiect științific ce are ca scop construirea unui portal de colaborare C&D a unei platforme de comunicare pentru integrarea roboților în structuri flexibile de producție, la cerere, și a unor mijloace de învățare prin practică. Platforma IRVIP este un produs software proiectat pentru a suporta conexiuni multiple cu un număr de controlere Adept Technology, localizate fie în rețeaua locală fie în Internet. În plus, IRVIP permite transferul de imagini multiple capturate de la sistemul de vedere atașat robotului la un server IBM XSeries pentru procesarea paralelă de mare viteză a imaginilor în concordanță cu algoritmii existenți sau cu noi algoritmii (ghidare vizuală pentru roboți (GVR), și inspecție vizuală automată (AVI) folosind unele vizuale bazate pe măsurări).

The paper describes a scientific project aiming to build up an R&D collaboration portal and communication platform for robot integration in on demand, flexible manufacturing structures and hands-on team training. The IRVIP platform is a software product designed to support multiple remote connections with a number of Adept Technology V+ robot-vision controllers, either located in a local network or via Internet. Additionally, IRVIP allows the transfer of multiple captured pictures to an IBM xSeries server for high-speed, parallel processing of images according to existing or new algorithms (guidance vision for robots (GVR), and automated visual inspection (AVI) from measurements based on vision tools).

Keywords: Web-based robot integration, open architecture, networked manufacturing, visual inspection and guidance.

1. Introduction

Production flows are nowadays modular, which means that each module in the enterprise is specialized and used to achieve a particular task. In many cases the modules are connected and materials are sequentially processed in each module resulting, in a final, unique product or assembly. One typical such production module is a flexible cell/system using multiple robots.

Usually robot (-vision) controllers are masters over local workstations or cells, because robot manipulators connect two important material flows: the *processing* flow and the *transportation* flow. One solution to integrate these two flows with on-line *quality control* in the manufacturing module, further networked

¹ PhD Student, Dept. of Automation and Industrial Informatics, University Politehnica of Bucharest, Romania

with the design and planning modules, is to adopt a unified feature-based description of:

- Parts and assemblies (*recognition pattern – R_P*);
- Technological operations (*processing pattern – P_P*);
- Geometric & surface quality control (*measure pattern – Q_P*);
- Grasping and manipulating (*grasping pattern – G_P*);

It is possible to generate this feature-based description by image processing from multiple area cameras (stationary, arm-mounted, down looking the conveyor belt, etc), in a three-step information management process:

1. Definition of mechanical processing features (e.g. hole, pocket etc) and translation into visual anchor-, point-, edge-, and combined features extracted from images.
2. Modelling: part classes for recognition in all processing stages; collision-free grasping styles for each part class, processing stage and presentation objective; scene and foreground appearance (constrained part location, parts touching or overlapping); measurement patterns for automated inspection (using vision tools)
3. Implementing a relational database $R_P:P_P:Q_P:G_P$ the associations of which are checked at runtime against predefined links and parameter values. Task-driven part search, identification, measurement and locating will confirm a normal follow-up of the pre planned production sequence or alter it.

The paper describes a system which can be used to unify, control and observe the cell's devices (each robot-vision system) from a remote location, e.g. the CAM/CAQC server linked to other design and planning compartments.

The system is now under construction as a software product designed to support multiple remote connections with a number of Adept Technology robot-vision controllers, either located in a local network, or via Internet.

Additionally, it allows the transfer of multiple captured pictures to an IBM xSeries server for high-speed, parallel image processing according to existing or new algorithms (image enhancement, pattern recognition, feature extraction for part description, robot guidance vision – GVR, automated visual inspection (AVI) from measurements with vision tools). The IBM Server is used to process multiple images at pixel level for future use in feature-based modelling and monitoring, and quality control of material flows.

The system has multiple functions:

- Observing *locally* the *foreground* of robot's workplaces (processing area, storage, conveyor belt, part buffer, pallet) using multiple area cameras – stationary or mobile, arm mounted, and *globally* the robot *workstation*;

- Set up of the *operating environment* (lighting pattern, virtual camera configuration, feature selection for material flow description) and learning the *model parameters* for scene description, part recognition and measuring, part grasping and gripper fingerprints for collision avoidance;
- Editing, debugging and downloading application data and programs;
- Remote shared control of multiple robot systems from a central point and event-driven supervision of robot actions including reaction and emergency routines launching;
- Access via a Lotus Domino-based messaging and collaboration portal to a team workspace, addressing hands-on team training and authenticated e-learning in the areas of computer aided design, planning, control and quality inspection for networked manufacturing workstations/cells.

2. The Structure Of The System

The system is composed by three applications (Fig. 1):

1. **The Server Application (SA):** Remote visual control and monitoring of multiple robot controllers from mobile and stationary matrix cameras.

- *Visual control:* the Server Application supports almost all V+ and AdeptVision program instructions and monitor commands. The robot training and control is interactive – menu-driven and acknowledged by image display in a VISION window. Some of the main functions available in this window are: choice of the physical and virtual cameras and of the image buffers; selecting the display mode and resolution; histogram and average curve contrast analysis; selection of switches and parameters for virtual camera construction; display of vision system status; training and planning multiple ObjectFinder models for recognition and locating; learning fingerprint models for collision-free grasping; editing, saving, loading and running V+ programs.
- *Monitoring:* a Monitoring/Treatment scheme can be defined for each Client/Station (the latter can be selected from a drop-down list of robot controllers connected to the server). For each client a list of events and controller variables to be monitored according to a user-definable timing and precedence, and reacted at by user-definable actions/sequences can be specified in an Automatic Treatment Window.
- *Access to image pixels:* Images taken over from clients are stored in a standard format allowing accessing the individual pixels for specialized, deep computing IPA treatment; the processed images, extracted features or computed measurements can be stored or transferred back to the client for further use.

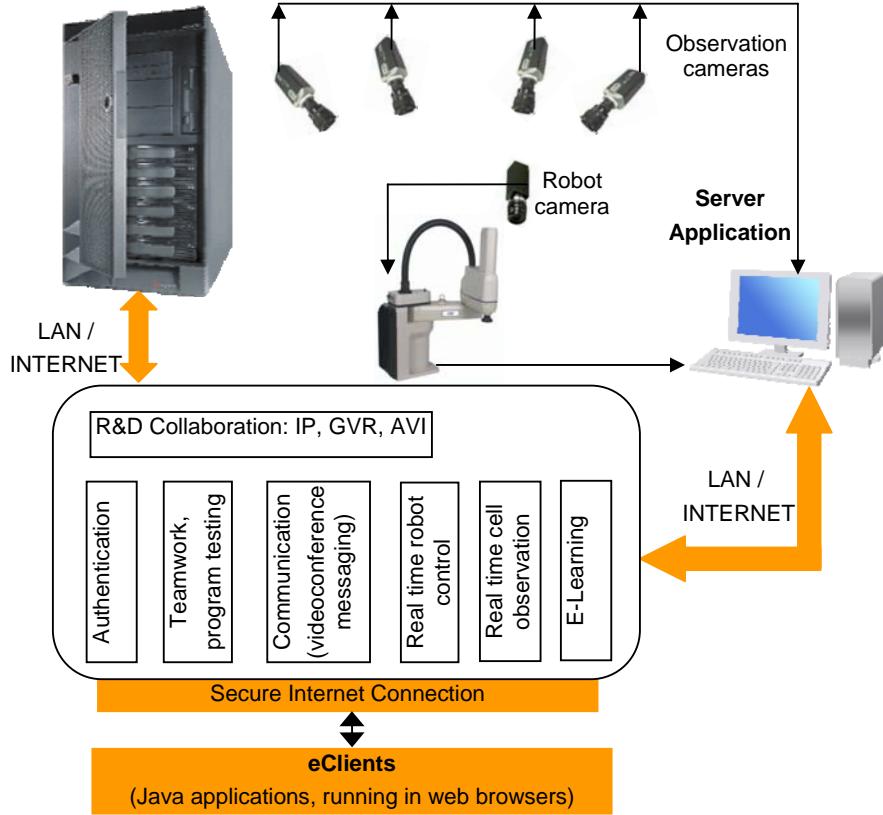


Fig. 1. The IVRIP Structure.

- *Communication management:* the Server Application manages the communication with the robot controllers and the observation cameras, transfers real-time images from the cameras observing the robot workplace and production environment, reports status information, stores in a database and displays images taken by the robot camera via its controller. Finally, the SA outputs commands which are received from the eClients or acknowledges task execution.

2. The **eClients Applications (eCA)**: Java applications running in web browsers. They provide portal services and the connection of networked production agents: image data and RV program / report management; real-time robot control and cell / workplace observation.

For team training and application development, the IVRIP pilot system allows accessing related documents, presentations and multimedia materials, Web conferencing, instant messaging and teamwork support for efficient and security-oriented creation and use of group workspaces (students / trainers, researchers).

- The **Image Processing Application (IPA)**: Task-oriented filtering, image transforms and enhancement by histogram modification; multiple threshold binarization; detection of special features (gauge predicates); signature analysis; rule-based inspection; advanced shape analysis; visual servoing of robots.

The Server and eClients Applications run on an IBM PC workstation on which IBM Lotus software offerings are customized and integrated with other applications (error-free data and command transfer, V+ program editing, real time image display and refresh in multiple Client windows, replication functions, Client authentication, etc) in a virtual training and research laboratory across geographical boundaries. Lotus-oriented solutions have been considered for transferring messages, status reports, data and video camera images, interpreting them and accessing databases created by all partners. The final objective of the platform is to develop an E-Learning component allowing students to access and download technical documentation, create, test, debug and run RV and AVI programs, attend real-time laboratory demonstrations, and check their skills in proposed exercises.

Thus, IBM Lotus software unifies all three Application modules, providing the necessary management and interconnecting tools for distributed industrial controllers, and the collaborative tools with back-end relational databases for team training and research.

3. R&D Collaboration Concept

The strong impact of the project is in stimulating the cooperation between different networked areas of an enterprise.

3.1 Technical background

The task is to build a system that provides access to public information for a wider audience and, at the same time, supports collaboration between registered members, provides safe access to protected contents and enables the publication and editing of contents. The system can be accessed from a great variety of places. Therefore great care was taken to also ensure optimal use in case of lower bandwidth and poorer quality hardware. The high level of security and availability are key components. This was ensured by the selection of high quality technical devices and the well-planned loading. As the portal must be prepared for a growing number of users, the tool must be highly scalable. It needs to integrate contents and services and provide access for document repositories. User groups must be able to access personalised contents.

A first step to install a framework is to create the infrastructure that is necessary for the safe operation and use of the portal (Fig. 2).

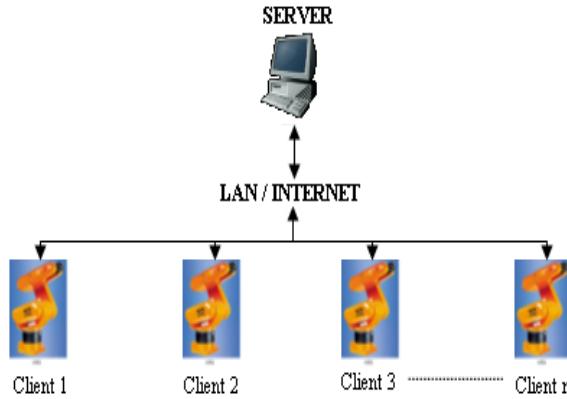


Fig. 2. The connexion between the server application and Adept controllers.

High-level availability must be guaranteed. The system must be scalable according to loadness and requirements. The use of NLBS (Network Load Balancing System) provides a solution for an even load of the network. The portal user interface need to be customized, templates must be created and uploaded. The authorisations and user groups must be defined.

3.2 The communication structure and protocol

The application was built to remotely command and supervise robot systems that can be located into an industrial LAN or Internet. The materials flow is supervised locally, for each station, and the results are sent to the server application which stores them into a data base.

The server uses the TCP/IP protocol to establish the connexion with the clients, the communication being achieved using messages and confirmations [1]. The length of one message cannot be more than 128 bytes; this limit is imposed by the operating system of Adept controllers. The message has two parts: the header of the message (two bytes), and the body (0 to 126 bytes)(see Fig. 3).

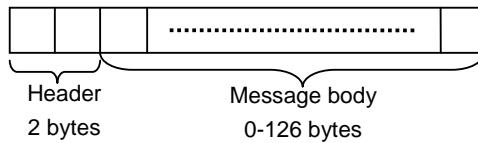


Fig. 3. The message structure.

The header represents the type of the message and specifies the way in which the body of the message will be utilised. For example the headers can be:

1. Headers sent from the server:

- '/C' – message containing a monitor command; the message is processed by the client as an instruction that will be executed immediately;

- '/V' – message containing a monitor command; this message is different from the previous one because the command is used to acquire a new image, and that image must be sent to the server;
- '/T' – message containing a variable and signal list; the variables and signals must be supervised and if a modification is detected the server is informed;
- '/L' – marks a new line from a list of instructions which form a program;
- '/E' – message which mark the end of a program;
- '/S' – monitor commands requiring a special execution mechanism;

2. Headers sent from the clients:

- '/M' – message to be displayed on the server terminal (can be a user message, a error message, or a message generated by a instruction or a command);
- '/A' – ACK message;
- '/V' – Vision ACK message (the client tell the server that the image was acquired, and is ready to be loaded to the server);

As a safety measure, but at the same time to ease the use of the application, the clients which will be connected to the configuration of the server must be specified. The client is identified using a name and an IP address (Fig. 4).

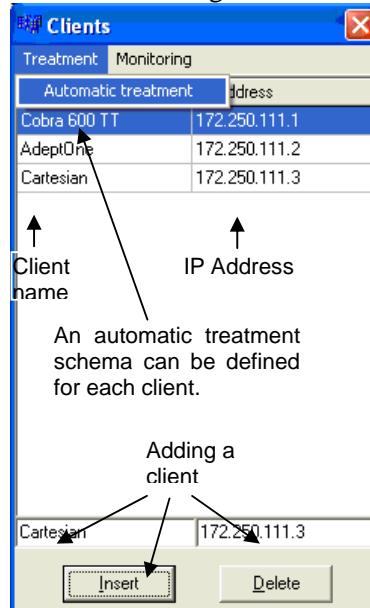


Fig. 4. Automatic treatment.

4. The Command System

The robot stations are commanded using the command line and the menus. When a client is connected, the IP address is checked and, if the client is accepted, the name attached to the IP address is added to a drop down list from which the

user can select what client he wishes to command. When a client who has a video camera attached the VISION button is enabled and if it is pressed the VISION Window will open.

From the VISION window, vision commands can be issued by selecting the wanted actions from the menus (Fig. 5). The most important functions are:

- selecting the physical and virtual cameras, and the virtual image buffers;
- selecting the display mode and the resolution;
- image acquisition;
- issuing primary operations (histogram, thresholding, etc.);
- displaying the vision system status;

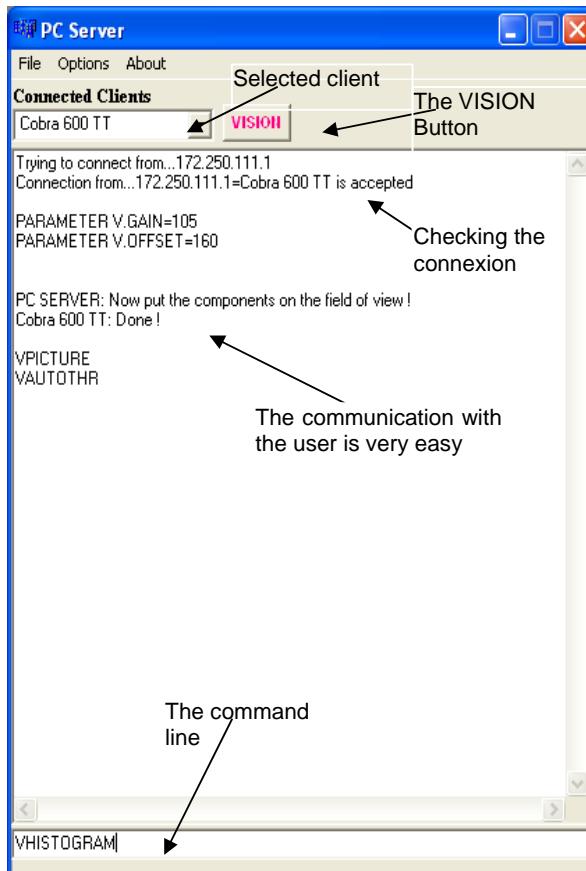


Fig. 5. The main window of the SA.

- training models;
- switches and parameters configuration for virtual camera set-up.

The advantage of the Vision window is that all commands can be issued using menus, but also the fact that the image acquired by the camera and sent to

the server can now be accessed at pixel level. Another major advantage is that the training of the part recognition and grasping models become a single-step process during which a unique window is used for parameters and constraints specification.

The client application can acquire full or partial images via the VGETPIC V+ operation and send them to the server [2].

Captured image can be processed via the menus (filtering, binarization, convolution, morphing, etc.), saved into a common format and sent to a specialized image processing application. After processing, the image can be sent back to the client and integrated to the image processing board using the VPUTPIC operation, for further use (an application using this mechanism is in course to be developed, and consists in a new part identifying algorithm based on skeleton computation and matching).

A basic approach to representing the structural shape of a manufactured part is to reduce it to a graph. This reduction was carried out by obtaining the *skeleton of the region* via a thinning (also called skeletonizing) algorithm.

The *medial axis transformation* (MAT) of a region R with boundary B is defined as follows: for each point p in R , find its closest point in B . If p has more than one such neighbour, it is said to belong to the MAT (*skeleton*) of R . The results of the medial axis transformation depend on the particular distance metrics that is selected. The MAT is sensitive to local distortions of the contour and small 'bumps' can give rise to extraneous skeletal lines. Fig. 6 illustrates the MAT of a metallic part having a 'bump'.

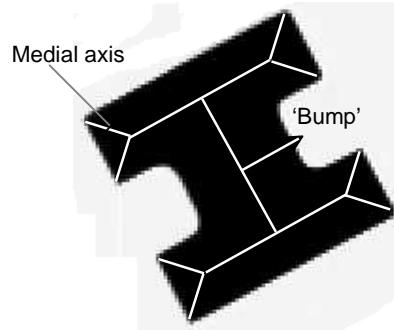


Fig. 6. Medial axis transform (MAT) of a 'T'-shaped part with a lateral 'bump'.

An algorithm producing the medial axis representation of a region with improved computational efficiency was chosen based on *thinning* methods that iteratively delete edge points of a region subject to the constraints that deletion of these points: (i) does not remove end points; (ii) does not break connectedness; (iii) does not cause excessive erosion of the region [3] [4].

Binary region points were assumed to have value 1 and background points to have value 0. Successive passes of two basic steps were applied to the contour points of the given region, where a contour point is any pixel of value 1 and having at least one 8-neighbour valued 0. Fig. 7 shows the definition adopted for the 8-neighbourhood.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

0	1	1
0	p_1	0
1	1	0

Fig. 7. Definition (left) and example (right) of the 8-neighbourhood for blob thinning.

STEP 1. A contour point p is flagged for deletion if the following conditions hold:

- 1.1 $2 \leq N(p_1) \leq 6$;
- 1.2 $S(p_1) = 1$;
- 1.3 $p_2 \cdot p_4 \cdot p_6 = 0$;
- 1.4 $p_4 \cdot p_6 \cdot p_8 = 0$,

where $N(p_1)$ is the number of nonzero neighbours of p_1 , i.e. $N(p_1) = \sum_{i=2}^9 p_i$, and

$S(p_1)$ is the number of $0 \rightarrow 1$ transitions in the ordered sequence of the bits $p_2, p_3, \dots, p_8, p_9$. For the example in Fig. 7, $N(p_1) = 4$ and $S(p_1) = 2$.

Step 1 is applied to every contour point in the binary region of interest. If one or more of conditions (1.1) – (1.4) are violated, the value of the respective point is not changed. However, the point is not deleted until *all* contour points have been processed, which prevents changing the structure of the data during execution of the algorithm. After applying step 1 to all border points, those which were flagged are deleted (changed to 0).

STEP 2. A contour point p is flagged for deletion if the following conditions hold:

- 2.1 $2 \leq N(p_1) \leq 6$;
- 2.2 $S(p_1) = 1$;
- 2.3 $p_2 \cdot p_4 \cdot p_8 = 0$;
- 2.4 $p_2 \cdot p_6 \cdot p_8 = 0$.

Step 2 is applied to data resulting from step 1 in exactly the same manner. After step 2 has been applied to all border points, those that were flagged are deleted (changed from 1 to 0).

Thus one iteration of the thinning algorithm consists of:

1. Applying step 1 to flag contour points for deletion.
2. Deleting the point flagged in step 1.

3. Applying step 2 to flag the remaining contour points for deletion.
4. Deleting the points flagged in step 2.

This basic cycle is applied iteratively until no further points were deleted, at which time the algorithm terminates yielding the skeleton of the region.

Such types of applications, for which the complexity of the computation is very high, were developed on a powerful IBM xSeries server instead of the robot vision controller.

5. The Supervising Function

The server application is capable to command and supervise multiple client stations. The material flow is supervised using the client stations and the status from each station is recorded into a data base.

For the supervising function a variable/signals list is attached to each client (Fig. 8). The variables/signals are verified by the clients using a chosen strategy, and if a modification occurs, the client sends to the server a message with the current state. Supervising can be based on a predefined timing or permanent.

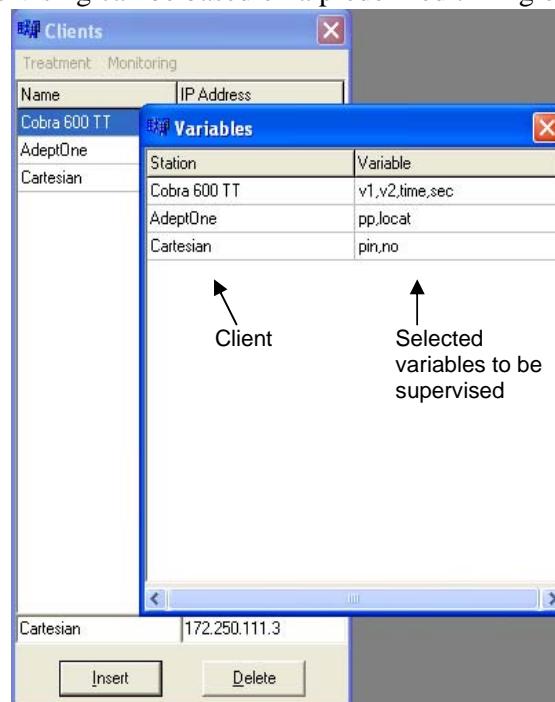


Fig. 8. Selecting the variables and the signals to be supervised.

If the status of a signal/variable is changed the server analyse the situation and take a measure to treat the event, so each client has a list of conditions or events that are associated with a set of actions to be executed (Fig. 9).

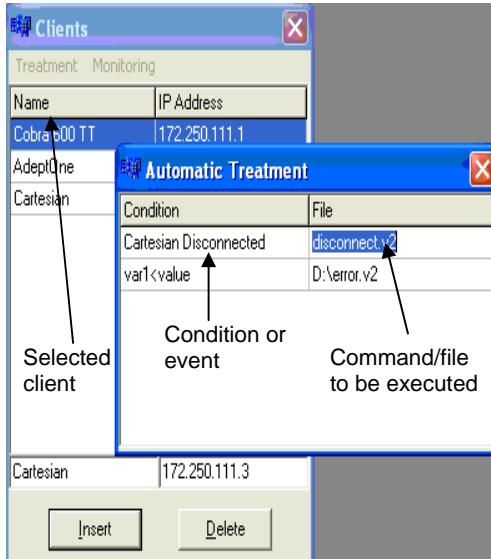


Fig. 9. Selecting the events and the actions to be automatically taken.

This feature removes much more from the human intervention, the appropriate measures being taken by a software supervisor.

When the *supervise* mode is selected, the server sends to the client (using a message with header ‘/T’) the variable list to be supervised and the time interval when the client must verify the status of the variables (in the case when the supervise mode is periodical).

The events which trigger response actions can be produced by reaction programs run by the controller (REACTI or REACTE type) or by special user commands from the terminal. The list of actions contains direct commands for the robot (ABORT, KILL 0, DETACH, etc) and program execution commands (EXECUTE, CALL).

6. Lotus Domino Platform And E-Learning Aspects

The eClient application is a Java applet that runs under the Lotus Domino Server, and implements a security access policy to the virtual workspace. The access to the eClient application is granted based on the Domino defined ACL’s (Access Control Lists), such that in order to connect to the application the user must specify a user name and a password. There were defined two classes of privileges:

1. A **user class** where the operator can observe images acquired from the observation web cameras and images from the VISION system taken by multiple area cameras; he can also view the commands issued by the trainer and watch the results of the commands;
2. A **trainer class** where the operator is authorized to issue commands for every connected robot system, upload, download and modify programs. The trainer can also take pictures from an individual camera and use the specific vision tools to process that image. The observation cameras can also be moved and positioned in the desired location by the trainer. The trainer can give full or partial permissions to users for program testing purposes.

The communication between the users is achieved by help of the integrated console (text mode) or using an Instant Messaging and Web Conferencing application (Sametime).

IBM Lotus Domino server software was used to combine enterprise-class messaging and calendar / scheduling capabilities with a robust platform for collaborative applications on a wide variety of operating systems. The design of the Lotus Domino Server made available three offerings: Domino Messaging Server (messaging only), Domino Utility Server (applications only), and Domino Enterprise Server (both messaging and applications) [5].

The most important Lotus Domino features that were used are:

- Encryption, signing, and authentication using the RSA public-key technology, which allows to mark a document in such a way that the recipient of the document can decisively determine that the document was unmodified during transmission.
- Access Control Lists (ACLs) determining who can access each database (application) and to what extent.
- Usage of Domino's new features to reduce network utilization. Network compression reduced the number of bytes sent during transactions by up to 50 percent. Connections across heavily loaded links, such as WANs and XPCs, will see the most benefit.
- Availability for the Windows NT and XP platforms, automatic fault recovery after shutdown and server restart without administrator intervention after the occurrence of an exception. Fault recovery uses operating system resources, like message queues.

Because the application eClient is accessed over the Internet, security represented a critical element. The access to different levels of the application is controlled by xACLs (extended ACLs) to allow or disallow access. The existing database Access Control Lists (ACLs) and the new ACL file feature ensure that application-private databases remain secure. In addition, file protection documents for the Domino Web server which is used to serve the eClient (Java application) provide additional access control for files accessed via HTTP.

7. Conclusion And Future Work

The project is under construction; most of the Server Application's functions are already implemented and tested on a pilot platform in the Laboratory of 'Robotics and AI' of the University Politehnica of Bucharest.

The research project will provide a communication and collaboration portal solution for linking the existing pilot platform with multiple V+ industrial robot-vision controllers from Adept Technology located in University Labs from various countries (UVHC Valenciennes-France, Quebec-Canada, FH Konstanz - Germany, and TU Cassino-Italy).

The SA module is basically finished and successfully tested. Currently, the eCAs are under construction, as well as the IPA library designed together with research groups in the partner universities and tested on the IBM xSeries Server in the Linux Competence Centre in Bucharest.

R E F E R E N C E S

- [1] *Borangiu Th., Tomas Balibrea L.M., Gonzales Contreras L.A., Manu M.*, 1997, Object Oriented Model of an Open Communication Architecture for Flexible Manufacturing Control, Lecture Notes in Computer Science 1333 - Computer Aided System Theory, EUROCAST '97, Berlin, 292-300.
- [2] *Adept Technology Inc.*, 2001, AdeptVision Reference Guide Version 14.0 Part Number 00964-03000, San Jose, Technical Publications.
- [3] *Borangiu Th.*, 2004, Intelligent Image Processing in Robotics and Manufacturing, Bucharest, Romanian Academy Press.
- [4] *Borangiu Th., Anton F.D., Tunaru S., Ivanescu N.-A.*, 2005, Integrated Vision Tools And Signature Analysis For Part Measurement And Recognition In Robotic Tasks, IFAC World Congress, Prague.
- [5] *Brooks K., Dhaliwal R., O'Donnell K., Stanton E., Sumner K., Van Herzele S.*, 2004, Lotus Domino 6.5.1 and Extended Products Integration Guide, IBM RedBooks.