IEEE Instrumentation and Measurement
Technology Conference
Budapest, Hungary, May 21-23, 2001.

A framework for interaction of distributed autonomous systems and human supervisors

H.J.W. Spoelder, D.M.Germans, L. Renambot, H.E. Bal,
Division of Physics and Astronomy
Division of Mathematics and Computer Sciences
Faculty of Sciences, Vrije Universiteit,
De Boelelaan 1081, 1081 HV Amsterdam
tel. +31 20 444 7866, e-mail: hs@nat.vu.nl

P.J. de Waal, F.C.A. Groen
Faculty of Sciences, Universiteit van Amsterdam
Kruislaan 413, 1098 SJ Amsterdam

Abstract ***– Autonomous systems are rapidly gaining importance in a large number of situations relevant to the general public. By their nature their need for external control is low but still necessary. In this paper we present a framework for interaction of distributed autonomous systems and human supervisors. This framework exploits progress made in a number of related areas and shows that they can be effectively combined into one single framework. To this end it combines an environment for computational steering with virtual reality techniques for visualization and wap-based communication for ubiquitous intervention. Given the described setup for the technology the current version must be seen as a prototype that shows the feasibility of this approach.***

Keywords ***– distributed measurements, human in the loop, WAP, autonomous systems***

I. INTRODUCTION

There is a growing demand for deploying distributed, autonomous, systems in distributed environments. Applications are in public safety, monitoring of traffic and environmental conditions, assistance and clean-up work in disaster areas [1]. A first task of such systems is *distributed measurement and reconnaissance*.

In many situations, interaction of the autonomous systems with human supervisors for critical decisions and monitoring activities is of vital importance, particularly in dynamic situations where decisions are time critical, the trade offs are important. An example is where to go for obtaining additional information on both the information accumulated in the world model and the supervisor's understanding of the situation. In such cases, the *human is a central part of the control loop* and their behavior becomes an important consideration in the system design [3].

Although the Internet is becoming almost ubiquitous in several of the abovementioned situations, Internet connectivity for all human supervisors is not a valid assumption. Wireless techniques, especially employed in second-generation mobile phones [4], are becoming the next vehicle for connectivity. Clearly they enjoy the benefit of their mobility. We see these wireless techniques as precursor of the mobile, networked, computer.

In this paper, we study the design of a framework to integrate Internet-based distributed measurement systems with human supervisors using second-generation mobile phones.

Our objective is twofold: (i) chart the structure of such a distributed measurement system given the current state of the technology and (ii) the possibility to support 'human-in-the-loop' interaction using a range of techniques ranging from CAVE to waptelephones. This case study is based on an implementation of this design using real robots, a CAVE/workstation and a waptelephone.

II. RELATED WORK

This study combines key technologies and developments in a number of fields. Among the most important related areas are the following.

A. Virtual Reality Techniques

The rapid advances in computer graphics have dramatically changed the way in which humans can interact with data.([2]). Instead of working at a low abstraction level with numerical data, it is now possible to present data in a natural and easily interpretable metaphor. An extreme example of this development is the CAVEs (Cave automated Virtual Environments). A more common and increasingly powerful example is the state-of-the-art graphics card of an off-the-shelf

PC. A typical example of such work is found in [1]. In both cases the visualization of the measured data facilitates the interpretation of the scene for the human observer. A limiting factor for the use of this technology is the delicate balance that must be struck between flexibility, performance, and ease-of-use of the software used and the problems in finding proper paradigms for visualization. The optimal choice of VR environment is highly dependent on the type of data studied.

B. Mobile systems and the internet

Internet robots are a typical example of distributed measurement systems[4], which combine the notion of a virtual instrument and a network. As with the virtual instrumentation, the framework used for the development of applications is open to debate. Interesting on the robots as measurement system is that they can exhibit an adaptive behavior and thus increase the accuracy of the measurement.

For the network, the inevitable latency and the limitations on bandwidth make it impossible to share all available information in the entire network forcing a distributed approach [5] and a logical structure for determining 'what has to be done where'.

C. Wireless communication techniques

Wireless communication techniques are increasingly becoming popular. The advent of the WAP [6] can be seen as a modest precursor of the mobile computer. In the current implementation, computing power, interface, and size of the display of wapphones are limited. However, often these drawbacks are easily compensated by the waptelephone ubiquitous nature.

III. GLOBAL SYSTEM OVERVIEW AND DESGIN OBJECTIVES

The hierarchical control structure on which our framework is based is shown in Figure 1. The human observer controls and takes part in a system consisting of a number of clusters of mobile and or fixed systems. A logical structuring of the clusters can be based on the latency of their communication.

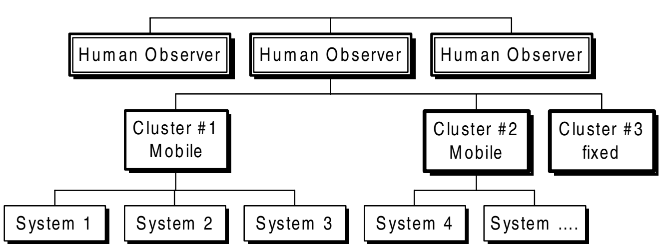

Figure 1 Control structure used for this framework.

Globally one can say that the system comprises four major components: the autonomous systems, the mobile communication (WAP), the VR environment for interaction and the interconnection software. A realization of the above structure, which has been used for our study, is given in Here two autonomous systems exchange information with a human observer in the Cave, another observer behind a workstation and a third, mobile, observer connected via WAP. All communications are as indicated bi-directional. The type of information exchanged is annotated in.

The main objective of this case study is to chart a possible framework for the interaction of human observers with distributed autonomous systems. This implies both a realization of a software infrastructure for the communication between the various components as well as an infrastructure for the – natural – interaction of human observers with the system and the data measured.

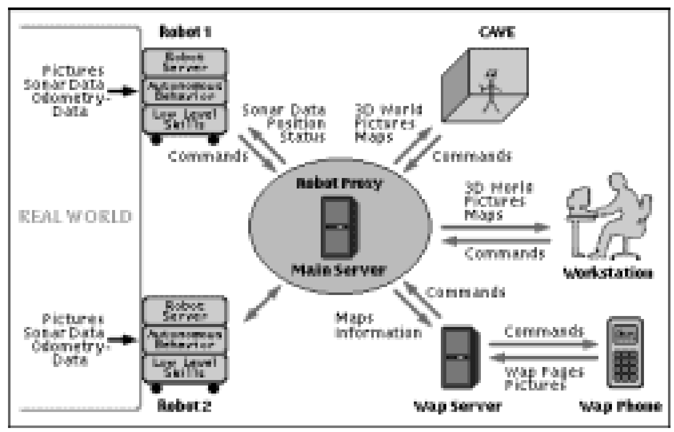

Figure 2 General architecture of the framework used in this study.

Our design objective in this is to reuse as much as possible existing software components and realize a setup that also allows the non-expert user to make minor modifications to the system.
The secondary objective of the case study is to test the usability of the setup for actual measurements. In the remainder of this paper we will present the main characteristics of our framework.

IV. SYSTEM DESCRIPTION

In this section we will focus on the design of the four major components described in the previous section.

A. CAVEStudy Interconnecting software

The interconnecting software plays a central role. In our framework we use a software environment, CAVEStudy[7], that has been developed in our group. This environment combines the power and the functionalities of computational steering and virtual reality. It consists of two major parts: a code generator and a VR framework. To minimize the programming for the control of the computational process and the data management a specification of the process is given in a file. This file is processed to generate two objects, a proxy and a server. The computational process is wrapped in a server object to control its execution. The server's interface provides methods to start, stop, pause, resume and read input. The data generated by the computational process are automatically propagated to the proxy object. This object can be seen as the local copy of the computational process.

The generalization from computational processes to autonomous systems is easily made. We regard the autonomous system as a process that can accept commands (input) and describe its state or do measurements (output).

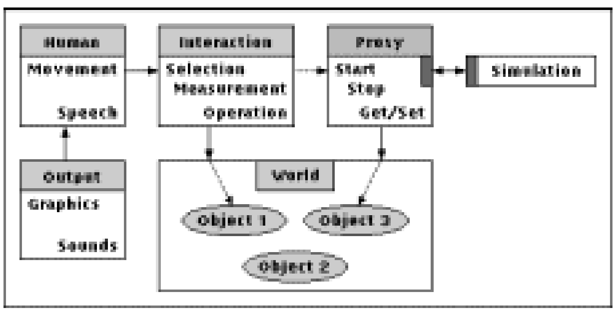

Figure 3 VR framework support by CAVEStudy

The VR framework of CAVEStudy, shown in Figure 3, is used to steer and visualize the data produced by computational processes. It consists of a shared-world where the outputs of the computational process are represented. An interaction module allows the user to scale, rotate, translate objects and select objects. Other interactions have to be programmed explicitly.

Figure 4 Nomad Scout Robot

B Autonomous Systems and their control

One of the autonomous systems that we have used in this case study is shown in Figure 4. All systems are uniquely characterized by a name. This Nomad Scout, jan, is controlled by a 300 MHz Pentium II system running Linux. The Linux system communicates via the RS-232 port with a 68000 Processor than controls the low level functionality of the system. It controls the various sensors attached like the 16 range sensors, a collision detection bumper. Other devices, like the color camera and IEEE-488 and RS-232 based measurement devices like the thermometer shown, are controlled by the Linux system. For communicational purposes the system uses a BreezeCom wireless Ethernet Unit which has a maximum bandwidth of 3 Mbit.

The communication in our setup is file based. The motivation to do so is that this highly facilitates browser based communication over the Internet. In view of the setup times of the communications involved, this choice is in know way limitative. An advantage of this approach is that logging and play back are trivially implemented.

Associated with each system is a file with identical name as the robot that contains the following info rmation.

<timestamp>
<name>
<status>
position (X Y)
Sensor_reading_1 (X Y)
……
Sensor_reading_16 (X Y)

Table 1 Information associated with autonomous systems

Status is a string having three values: *dead* (system is not functioning), *alive* (system is operational), *running* (system is busy executing a command) and *frozen* (needs attention). Both the current position of the system and the sonar readings are incorporated in this file. Therefore the information in this file is sufficient to build a local world model of the robot[8]. The information of the other sensors like the camera is all written onto a separate file. The control program running on the system currently supports a number of functions shown in Table 2.

The system can store a number of commands on a local stack. This stack can be flushed by means of the 'abort' command. When the system encounters a situation, which it cannot resolve, it freezes updates all associated output files and waits for human intervention.

The central server, shown in Figure 2, combines the information of all the separate systems and converts it into one global world model.

Navigate		move autonomous
Abort		flush command stack
Freeze		stop movement
Mark		store current pos
Goto	**<mark>**	goto mark
Move	**<spec>**	goto position
Rotate	**<spec>**	rotate over angle
Grab		grab image
Refrsh		refresh all output files

Table 2 Some commands implemented for the autonomous system.

The file-based control of the autonomous systems makes it very easy to integrate them into the CAVEStudy environment. Thus the ease-of-use of this environment can be used to control the cluster of systems and visualize their measurement on a variety of platforms.

C. WAP COMMUNICATION

The one remaining functionality to be inserted in the system is the ability for mobile control and intervention. This means that commands and status/pictorial information have to be exchanged between wapphone and robot. Since we have chosen Internet as the medium for control we the only thing we have to do is to map the input/output files onto HTML pages and make them available via http servers. At the server side a world file is maintained which contains the number of robots followed by theirs names, coordinates, time stamp and status. The name of this file is fixed. Reading this file can bootstrap a detailed reading of the status files of all participating systems.

```
<number of robots>
<robot name> <robot position> <robot time> <robot status>
…….
```

However for simple applications like setting up a positional map of all the systems this is not necessary. On the wapside a similar action is needed: the maintenance of a command file.

```
<time stamp> <robot spec><command>
…….
```

The usage of time stamps allows easy and fast check of possible updates of the file.

A first inventory of the possibilities of a wapphone to be used as a steering device in the context of 'human-in-the-loop' interactions with a mobile distributed autonomous system, led to a long list of restrictions and few opportunities. The current technical state of mobile Internet dictates one of the most important restrictions, the set-up time of a data connection based on GSM techniques. Furthermore, the second generation mobile telephones currently available that support the Wireless Application Protocol, the upcoming standard protocol for mobile internet, are advanced telephones but still poor computers. In this project we used a Nokia 7110 mobile telephone, the first product that supported WAP [3], launched in December 1999, and still one of the most popular.

The device is equipped with a small monochrome display with effective dimensions 96x45 pixels for displaying graphics, or five lines of text visual at a time. The standard numerical keyboard is extended with a vertical scrolling bar and two option buttons. As HTML counts as a standard mark-up language for internet, the equivalent for mobile internet is Wireless Mark-up Language (WML). The phone we used runs a browser implementation accepting WAP-WML version 1.1. [3] In fact, WML can be seen as a small subset of HTML, extended with some specific event handlers for mobile devices. Along with the WML parser comes a WML scripting interpreter, but the added functionality was not needed our case and we decided to do all the computing server-side in order to release the mobile device from further computing to increase its overall speed. In terms of memory the maximum length of a page or bitmap-graphics is about 1.5 kilobytes. The telephone can memorize up to 40 kilobytes of pages, but we can't use this caching memory since our pages are dynamically generated at server-side.

The connection with internet is set up by selecting an URL bookmark in the telephone menu. The phone calls a dedicated dial-in number and attempts to set up a data connection with the wapportal, using standard GSM data-transfer techniques at a 9600 baud rate. The connection is not permanent; it is set up again for every single request-response action, with an average response time around 10 seconds. Clearly this is the most important bottleneck confronting a real-time steering task. Whereas the displaying and computing capacities strongly influence the design of the interface, the above stated access limitations determine the general usability of the device as a steering component in the overall framework.

The mobile interactive visualization component consist of two parts: (i) the communication module and (ii) the interface.

The communication module is implemented in the server-side scripting language called ASP, running on a Windows NT 4.0 Internet information Server. The implementation consist of two parts: (i) the main loop and (ii) functions and subroutines. The main loop handles all request parameters calls the appropriate subroutines conform the requested viewpoint and command or action. The current stage of the project requested for a *single user environment* only, but the design and implementation accounts for future extending to a multi-user environment. To perpetuate single user access, in a flexible way, the script compares the IP address to the last logged user and responds a 'force logon' option in case the IP's didn't match. In any way the script holds to one single IP, stored in a logfile.

When the user is authorized, the main loop calls a selection of subroutines depending on the request and responds the constructed WML page. The subroutines discriminate on the chosen viewpoint (world or robot) and the availability of fresh data files on the remote main server. In case the user requests for an abort or command, the script will first write the command file and secondly try to retrieve fresh data. The command file is implemented as a standard text file, including time-stamp, robot name, command, parameters (optional). The command file will be checked frequently by the main server and in case the time-stamp states that the command file was updated, the content of the file will be processed and the actual command will be propagated to the autonomous system.

Each response (wml page) starts with a graphical visualization of the user chosen viewpoint and proceeds with a textual description of the most relevant state parameters and concludes with an options menu. Because of the limitations of the mobile phone, both in displaying and input facility, the script acts data-driven in the sense that the available information is first evaluated on various criteria before the response is constructed. Especially the user options menu participates on the availability of relevant data and the optional commands.

V. TEST CASE

As a first test for the system we have used it to chart an unknown building and measure a temperature profile. The latter part of the test is work currently in progress. For this test we have only used the sonar range data of the autonomous systems. The color camera is under explicit control of the user.

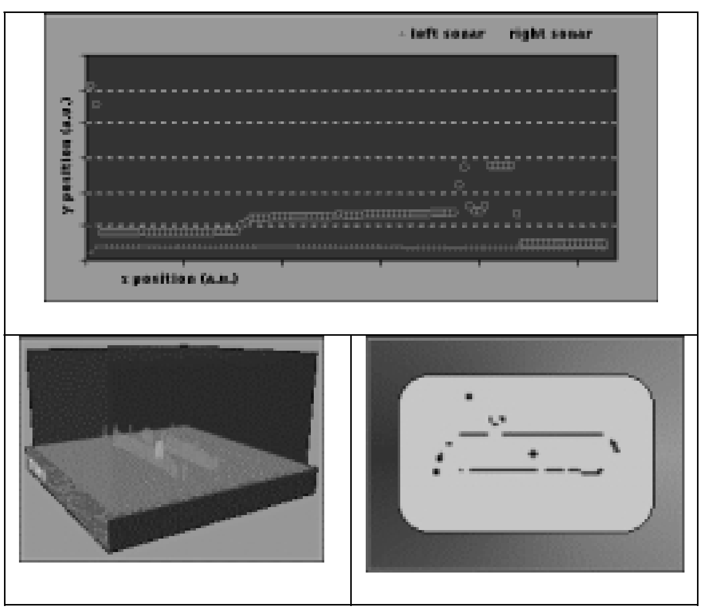

Table 3 Sonar reading and their visualization

The first picture of Table 3 shows the raw sonar data of the two sonar sensors on the left and right hand side of the robot while it is traversing a corridor. The jittery behavior of the upper line in this graph represents an opened door. Since no throughway exists at the end of the corridor the system decides to stop there, freeze, take an image of what lies in front and initiates human intervention.

For two human supervisors the graphical representation of the data in the CAVE and on the wapphone is shown in the lower part of Table 3. The observer in the CAVE not only has a superior view of the situation but has also the advantage of a latency in communication which is lower by orders of magnitude. However in a number of cases this advantage is easily balanced by the ubiquitous nature of the wapcommunication.

VI. RESULTS AND DISCUSSION

We have shown that a framework for interaction with distributed autonomous systems and human supervisors can be realized with state-of-the-art technology and existing software for computational steering. Wireless communication can be used in such a setup. Currently the setup time of the communication is the limiting factor. However given the rate at which wireless communication develops one may expect that in the near future this option becomes usable in real situations.

The choice for HTML file based information exchange looks promising although already somewhat outdated. Extension to the new XML standard can easily be realized and will facilitate the description of the data (output format) even further.

Lastly the notion of virtual instrumentation should be used both inside the autonomous systems and on the system as a whole to give the user easy support for distributed measurement.

REFERENCES

[1] D. Gracanin, K.P. Valavanis, N.C. Tsourveloudis, M. Matijasevic, "Virtual-environment-based navigation and control of underwater vehicles", IEEE Robotics and automation magazine, vol. 6, no. 2, June 1999, pp 53-62.

[2] See eg http://www.vrac.iastate.edu/research/index.html

[3] J.G. Kerasmas, "How will a robot change your life ?", IEEE Robotics and automation magazine, vol. 7, no. 1, March 2000, pp 57-62.

[4] "Robots on the web", special issue of IEEE Robotics and automation magazine, vol. 7, no. 1, March 2000.

[5] A Foster and C. Kesselman eds. , "The Grid: blueprint for a new computer infrastructure, Morgan Kaufmann 1998.

[6] WAP Forum Ltd, "WAP, WML Specification, Version 1.1", June 1999.

[7] L. Renambot, H.E. Bal, D. Germans and H.J.W. Spoelder, "Cavestudy: an infrastructure for computational steering in virtual reality environment". *In 9th IEEE International Symposium on High Performance Distributed Computing,* pp 239-246, Pittsburgh, PA, August 2000.

[8] B.J.A. Krose and R. Bunschoten, "Probabilistic localization by appearance models and active vision", Proc. 1999 IEEE Int. Conf. On Robotics and Automation, pp 2255 – 2260, 1999