VIRTUAL ENVIRONMENT FOR COOPERATIVE ASSISTANCE IN TELEOPERATION

Olivier Heguy, Nancy Rodriguez, Hervé Luga
Jean-Pierre Jessel, Yves Duthen
University of Toulouse,
IRIT, 118 route de Narbonne, 31062 Toulouse
FRANCE
[heguy, rodri, jessel, luga, duthen] @irit.fr www.irit.fr

ABSTRACT
- In order to help the user to accomplish a task, teleoperation systems have to integrate different tools such as visualization, divers interaction devices, planning tools, etc.… The interface must be able to give complete information of the real world and the user can, using a distributed platform, be helped by others users as well as by autonomous robots. The objective of our project is to combine teleoperation, virtual reality and adaptive systems to improve the control of teleoperation missions.

Keywords: teleoperation, adaptive systems, virtual reality, cooperative work.

1. INTRODUCTION

Teleoperation platform allows a user to execute or to control remote tasks, i.e. without being where the action takes place. Thanks to this, teleoperation trys to minimize risks during dangerous works: spatial exploration, toxic device to operate… it could also allow scientists to go where a human cannot go: volcanoes, smart caves… To help the user to accomplish such tasks, other users or an autonomous or teleautonomous robot could be used. Assistance robots complete human faculties and allow the system to take advantage of computer capacities to realize repetitive tasks, physically hard work, and to use as best the expert dexterity to look, fall and react at the right time [Arai00].
In our project, we are studying man-machine cooperation to carry out teleoperated missions in a system using virtual reality and adaptive tools. The goal for the human users and the autonomous robots is to achieve a global task in virtual environment.
Our project uses both technologies: virtual reality and behavior simulation. Thanks to virtual reality, we can have natural and intuitive interface and to mix different information to increase user perception. Behavior simulation tools are used to help a human user by means of autonomous robots.

VIRTUAL REALITY

Virtual reality is a combination of technologies offering natural interactions with a simulated world created by a computer.. That is why it could be very useful for training and teleoperation applications. Senses such as the touch thanks to force feedback devices are essential in teleoperation to realize tasks requiring a precise manipulation of objects. [Burde96]. The DIVE system (Distributed Interactive Virtual Environment) is a virtual reality system allowing many users to explore the 3D space and to interact with each other [Hagsa96]. VIPER, a system developed in our team [Torgu00] is also a generic, multi-user distributed virtual reality platform able to run on heterogeneous physical architectures.

BEHAVIORAL SIMULATION

Behavioral simulation is a part of the animation techniques that try to reproduce real world by giving to the actors or animated entities some independent behaviors. By the way those entities are not directed by a global controller but rather by a decision center set in each individual. Each character of the simulation will decide his behavior about its motion at the next step, using its internal state and all the information about the simulation at the current step. Therefore behavior simulation helps actors to interact naturally in an environment using its own characteristics [Terzo94],[Blumb95]. Each individual modifies his behavior using his sensors to filter the information from its environment. Then the decision center chooses the behavior to use and the actuators act in the environment.

2. PROJECT EVIPRO

Crucial aspects for the human-robot cooperation include simulation, distribution, robots autonomy, behavior descriptions and natural human-machine communication. We have developed an experimental environment called EVIPRO (VIrtual Environment for Prototyping and Robotic) that addresses these issues, allowing the assistance of autonomous robots during the realization of a teleoperation mission. The overall system consists of a reactive system (ASSET) that understand the events of the dynamic environment, and a system of behavioral simulation on the basis of autonomous robots (A^3) whose mission is to help the user

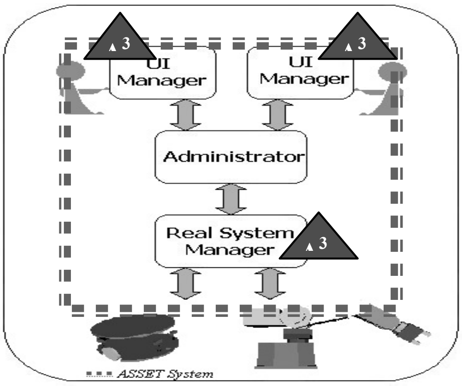

Figure 1. EVIPRO System

Figure 1 shows the EVIPRO system, with two users who cooperate between them and with an autonomous robot. Users addresses commands to the robots by means of interaction devices and the user interface (UI Manager). These commands are sent first to the local simulators to produce the feedback (visual, audio, force) to the user. If the result of that action triggers the simulation to another valid state, then commands are sent to the Administrator. By replicating simulation and models in every host participant, we have rapid feedback and “filtration” of commands. After validation, commands are communicated to the Administrator that is responsible for coordinating the actions of different users. Result is finally sent to control module of real system that addresses it to the simulation. After simulation update, the behavioral unit can know the state of each component of simulation, and can thus react as well as possible. This unit filters information to obtain the only needed data for making a choice of behavior and update his state. Finally, the acting devices execute commands. It is important to note that the autonomous robot is controlled only by the behavioral unit associated with the simulator of the Real System Manager, to allow system A3 to work only with valid global data.

EVIPRO can be installed on a variety of platforms because it uses only portable standard technologies – i.e. Java and VRML2.

3. ASSET AND A^3 SYSTEMS

We are currently working on the EVIPRO project: the aim of this work is to build a platform where drawbacks are compensated with an autonomous robotic system. Our study is then separated in two parts and their integration.

THE A^3 SYSTEM

Whereas teleoperation deals with sometimes-big delays it must be important to act without them. That’s one reason we try to make Adaptive Autonomous Agents (A^3). The other reason to make such a system is to help, if necessary, users who are resolving a task. But, moving and acting in a dynamic environment is not easy. As a matter of fact, sensors as well as effectors are limited and coarse. The environment is noisy and full of potentially useful information.

In the one hand, purely deliberative systems are considered the classical control architecture. But such an approach has many drawbacks: Generating a plan could be slow and it could take a lot of space (memory) to represent and manipulate the robot's state space representation.

In another hand we have Reactive control. it is based on a tight loop connecting the robot's sensors with its effectors. Purely reactive systems do not use any internal representations of the environment, and do not look ahead.

What we want is the best of both. This implies combining the different time-scales and representations. It is called hybrid control [Gat98]. It has to compensate for the limitations of both the planner and the reactive system. Then it has to reconcile any contradictory commands between.

The reactive layer: Sensors in this layer consist of numerical sensors without interpretation. This layer contains a library of primitives functions such as move_forward, turn_left,. By coupling sensors, effectors with combinations of primitive we create reactive behaviors. To react upon unpredictable event due to noisy sensors or other some primitives are added to stop if a collision not expected is detected or during long delays of planning.

The planning layer: Planning [SIM99] could be interpreted as looking ahead, or search. Planning is used to find a solution to a problem where the beginning as well as the goal is a state. Push planning is used to compensate actions of the users when trying to push an object. By applying basic rules of the dynamic and reinforcement learning the current approach angle could be adjusted to match the correct approach angle. Path planning is used to find a way the robot or the object can go. We use Genetic algorithm to find a sequence of instructions to go from start until goal. Here the individual consist of a sequence of move. The genetic algorithm works on a population of individuals initially randomized. Then thanks to genetic operators such as selection, crossover and mutation, the algorithm will converge on a solution. The fitness function uses only the distance to the goal. Each time a move is impossible (collision with an object of the scene) it is removed from the individual and the process continue until the end of the sequence.To speed up the process, another operator is added. It is used to optimize the sequence by applying loop truncation at each generation.

The intermediate layer: A typical hybrid controller employs a planner "at the top", a reactive system "at the bottom", and some means of combining/reconciling the two in between. it consists of a finite state machine coupled with a buffer of messages. Then any "important" changes discovered by the low-level controller are passed back to the planner in a way that the planner can use to re-plan. The low-level controller (and thus the robot) is stopped if it must wait for the high level planner to tell it where to go.

ASSET SYSTEM

ASSET (Architecture for Systems of Simulation and Drive in Téléopération) is a tool specialized in the development of new systems of teleoperation. Our motivation for the construction of this system was to provide a testbed for experimenting with behaviors, communication protocols, simulation models and interaction and acting devices. ASSET is a set of reusable Java components and mechanisms that could be used in the fast development of applications prototypes and in making of tests for new simulations models or new devices.

The architecture showed in figure 2 could be separated into 3 modules with subparts representing java components:

User Interface Manager: This module is responsible for communication between the system and the user, it handles the local simulation and the interaction devices. Communication between this module and the others modules of system is managed by the Communications and Events component.

Real System Manager: It is the module of control of the real system. It executes the commands and manages coherence between the real state and the simulated stated to update the simulations. It has too, a Communications and Events component.

Administrator: The administrator coordinates the interactions between the participating entities, users and robots. It has a Communications and Events component to transmit the commands to effectors and the information from the real system to the users.

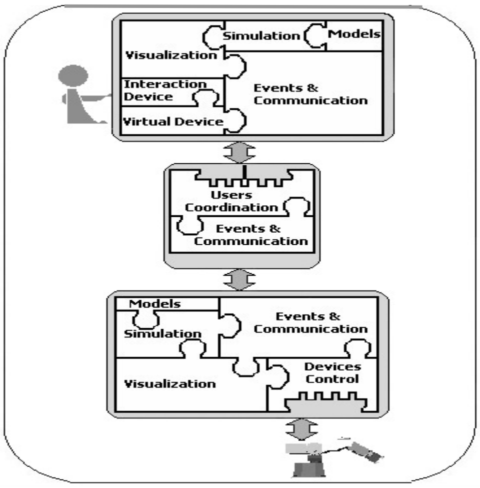

Figure 2. ASSET System

ASSET defines the mechanisms needed for the execution of the applications. They allow the interaction between the different components of each module and between the different modules.

Space data and Event Handling:. The data space notifies the occurrence of the events when a component adds a message and, in this way, the devices and the objects can recover them. All the communication between simulation, the communications manager and the devices is realized through the data space and the events that this one generates.

Simulation: In ASSET, there is a simulation in the UI Manager and another in the Real System Manager. The simulator in the UI Manager makes it possible to give feedback to the user without delay. On another side, the simulator in the Manager Real System avoids the transmission of the state at the end of each interval of simulation. The Manager updates the simulation if the state of the real system is very different from the state of simulation. The simulation component in ASSET offers the services of 3D visualization, collision detection and reading of Java3D and VRML2.0 models. One feature that is very interesting in ASSET simulation is the possibility to define the behaviour for each simulation object. This allows having entities with different degrees of autonomy in the simulation.

State: To know if the real state and the simulated state are different, ASSET uses the conditions defined by the user for his application. The user defines the set of variables that constitute the state of the system and, for each variable, he defines the maximum error value. If there are one or more variables that reach their maximum error value, simulation must be updated. The type of the variables and the concept of distance can be modified by the user because the system instantiates dynamically the classes developed by the user to manage his variables.

Devices management: In ASSET we have defined virtual devices. These devices offer a set of services that can be implemented with various physical devices. The virtual device is a mediator between the real device and the manager of events who allows ensuring the independence between the application and specific devices.

With the architecture and mechanisms defined in the ASSET system, we have achieved important requirements: extensibility, adaptability and a system highly configurable that can integrate available resources. For example, for testing of behaviors, the developer only needs to provide a geometric model and a control class for the entity. Same way, for testing a new device, the developer only needs to implement the basic services defined in the virtual device. This feature allows reducing the time of development and letting the developer to focus in his work to optimize it.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have defined EVIPRO, a system for the development of teleoperation applications that uses as essential technologies, virtual reality, teleoperation and adaptive systems.

In this experimental environment, the developer can test new simulation models, behaviors and devices. One key feature in EVIPRO, is the management abstract of the application's basic components (i.e. devices, communications, state). This feature facilitates the utilization of EVIPRO in the creation of new systems and in the making of rapid tests and prototypes. Furthermore, to achieve platform independence, the implementation of EVIPRO is based on Java and Java3D. We have used EVIPRO in the building of a sample application, and we have been able to demonstrate the benefits of EVIPRO flexibility, in particular in the dynamic integration of behaviors in the simulation and in the construction of application independent of specific devices.

Future work will cover several aspects. First, we work in the development of control methods for the autonomous and adaptive robots. Second, we will work in distributed simulation aspects. This concerns the interaction and synchronization between entities and management of time, with the integration of techniques known from standard distributed simulation management such as HLA, RMI and CORBA. Finally, improvements of the collision detection engine and application performance are pursued. Another valuable feature to be integrated will be the task planning, to allow the operator to use the result of his experiences in simulation to execute a task in the real system without permanent control.

REFERENCES

[Arai00] Arai Hirohiko, Takubo Tomohito, Hayashibara Yasuo and Tanie Kazuo: Human - Robot Cooperative Manipulation Using a Virtual Nonholomic Constraint. *Proceedings 2000 IEEE International Conference on Robotics and Automation*. 2000

[Burde96] Burdea Grigore : C.. Force and Touch Feedback for Virtual Reality. *John Wiley & Sons*. 1996

[Blumb95] Blumberg, B.; Galyean, T. : Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments". *SIGGRAPH - Computer Graphics Proceedings*, pp 47-54. Los Angeles, 1995.

[Gat98] Gat E.: On Three-Layer Architectures D. *Kortenkamp et al. eds. AI and Mobile Robots. AAAI Press*, 1998

[Hagsa96] Hagsan O.: Interactive Multiuser VEs in the DIVE System. *IEEE Multimedia Magazine*. Vol. 3, No 1. Printemps 1996

[Sanza99] Sanza C., Panathier C., Luga H., Duthen Y.: Adaptive Behavior for Cooperation : a virtual Reality Application *RO-MAN, 8th IEEE International Workshop on Robot and Human Interaction.Sep*. 1999 Pise, Italia.

[Simeo99] Simeon C. : Algorithmique du mouvement et planification de tâches en robotique *Habilitation à diriger les recherches du LAAS de Toulouse*, sept 99

[Terzo94] Terzopulos D., Xiaoyuan T., Grzeszczuk R..:Artificial Fishes with Autonomous Locomotion, Perception, and Learning in a Simulated Physical World. *Artificial Life IV*, R. Brooks et P. Maes Editors, MIT Press, pp 17-27, 1994

[Torgu00] Torguet P, Ballet O, Jessel JP, Gobetti E, Duchon J, Bouvier E.: CAVALCADE a system for collaborative virtual prototyping *Journal of Design and Innovation Research*, volume 2 number 1 special "virtual prototyping", Jan-Feb 2000