Cooperative Assistance for Remote Robot Supervision

E. Rogers, R.R. Murphy , A. Stewart , and N. Warsi

ABSTRACT

This paper describes current work on the design of a computer system which provides cooperative assistance for the supervision of remote semi-autonomous robots. It consists of a blackboard-based framework which allows communication between the remote robot, the local human supervisor, and an intelligent mediating system, which aids interactive exception handling when the remote robot requires the assistance of the local operator.

1. INTRODUCTION

The study of vision and motion in both man and machines is of particular importance in the arena of remote robot operations. In such cases, the robot must perceive and move to perform tasks in environments where it is deemed too costly or too dangerous for actual human presence. However, since the current state of technology has not yet produced a fully autonomous robot which can be sent on such missions, there is still a strong need for human intervention. The interaction between human and robot is managed in a variety of ways collectively referred to as *telesystems.* Telesystems have long been recognized as a key technology for space exploration, and they are becoming increasingly integral to a variety of terrestrial applications including the decontamination and decommissioning of nuclear processing plants, rescue, fire-fighting, intervention operations in hazardous environments, and security. Unfortunately, telesystems, in general, have several drawbacks. First, most systems require a prohibitively high communication bandwidth in order for the human to perceive the environment and make corrections in the remote's action quickly enough. Even with adequate communication bandwidth, the operator may experience cognitive fatigue due to the repetitive nature of many tasks, poor displays, and the demands of too much data and too many simultaneous activities to monitor. Furthermore, telesystems are inefficient in that the operator generally handles only one robot and that interaction leads to reduction of work efficiency by factors of five to eight [7]. As robots use more sensors, the amount of data to be processed by the operator will increase, exacerbating the communication and fatique problems and leading to less efficiency.

The addition of artificial intelligence at the remote is one solution to these shortcomings. The intelligence involved in the operation of a mobile robot can be viewed as encompassing a continuous spectrum from master-slave teleoperation through full autonomy [4]. An important open question, therefore, is how to add intelligence so as to move the telesystem forward on this spectrum.

Semi-autonomous control schemes address this problem by increasing the artificial intelligence residing at the remote in order to reduce both the amount of communication between local and remote, and the demands on the operator. However, there is still a need for human problem solving capabilities, particularly to configure the remote for new tasks, and to respond to unanticipated situations. In order to support the interaction between the different intelligent capabilities at the remote and local, the teleoperations community is becoming increasingly interested in computerized assistance for telesystems (tele-assistance), both for the effective filtering and display of pertinent information or data, and also for the decision-making task itself (e.g., [2, 3]). The work presented in this paper addresses this problem through the paradigm of cooperative problem-solving.

2. APPROACH

The approach taken in this project is to combine the autonomous perceptual and motor control abilities of the Sensor Fusion Effects (SFX) architecture for mobile robots [5] with the intelligent operator assistance provided by the Visual Interaction Assistance (VIA) system [9]. This work is a cooperative effort between

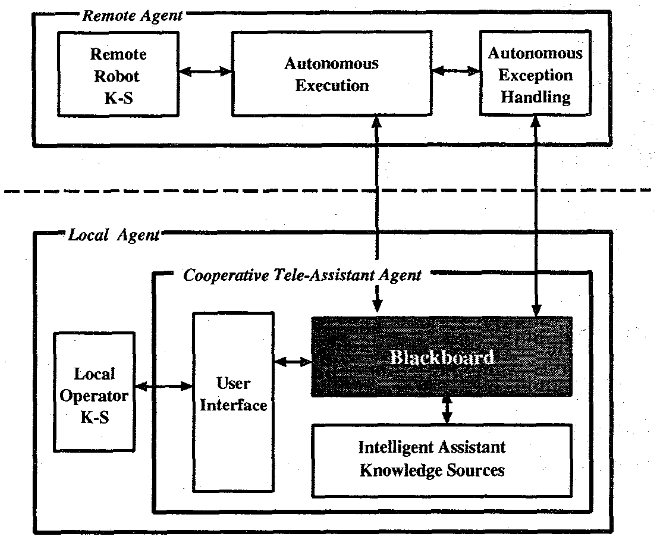

Figure 1: Overview of Agent Interaction in Tele-Assisted Robotics.

researchers at Clark Atlanta University and Colorado School of Mines. The latter houses the mobile robot laboratory which is providing the testbed for the teleassistance experiments. The intelligent sensing capabilities of the robot allow it to autonomously identify certain sensing failures, and to adapt its sensing configuration. However, if the remote system cannot resolve the difficulty, it requests assistance from the operator through the teleVIA mechanism. This cooperative computerized assistant presents the relevant sensor data, sensor information from other perceptual processes, and a log of the remote robot's hypothesis analysis to the user in a form which can lead to an efficient and viable response.

Our approach treats the remote and local as computational agents possessing unique knowledge and intelligence. The local "agent" is composed of the human operator, together with a computational agent called the intelligent assistant, which acts as an intermediary between the human and the robot. This agent doesn't move and it doesn't perceive. Rather, it supports the perception and problem solving capabilities of the human and the robot by selectively filtering and enhancing perceptual data obtained from the robot, as well as generating hypotheses about execution failures which cannot be solved by the remote.

The intelligent assistant uses a blackboard architecture to observe and manage the information posted independently by the remote and human intelligences. Blackboards have been previously used successfully for teleoperation by Edwards et al [3] in the Ground Vehicle Manager's Associate project, and by Pang and Shen [6] for high level programming and control of mobile robots involved in hazardous material spills. In our application of the blackboard, the remote, the operator, and the assistant are considered independent intelligent agents, as shown in Fig. 1. Each agent has internal routines called *knowledge sources* which read and post information to a global, asynchronous data structure called the *blackboard.* The knowledge sources at the remote post their information about the status of the robot. The operator reads the status and can use the knowledge presented by the intelligent assistant about previous or related cases to generate new directives such as task plans, sensor configurations, specification of parameters, response to anomalous situations, etc. The operator, by definition a knowledge source, communicates with the intelligent assistant and the remote via a graphical interface managed by the assistant. The interface supports learning new configurations and associates responses to extraordinary events. In an unaided system, the local task environment of the user presents numerous cognitive challenges: direct querying of the remote robot may be too slow; transmission of all related data may include unnecessary information; the sensor data itself may be in formats that are difficult for humans to understand and interpret. The display may contain different types of images obtained from various sensors involved in the failure, as well as some textual information on the hypotheses generated and tested through the robot's autonomous exception-handling mechanism. Any of this information could be faulty or misleading, and the user must quickly determine what is relevant, what it means, and what to tell the robot.

The development of our cooperative system therefore has a number of specific goals: 1) improve the speed and quality of the system's problem solving performance; 2) reduce cognitive fatigue by managing the presentation of information; 3) maintain low communication bandwidths by requesting only relevant sensory data from the remote; 4) improve efficiency by reducing the need

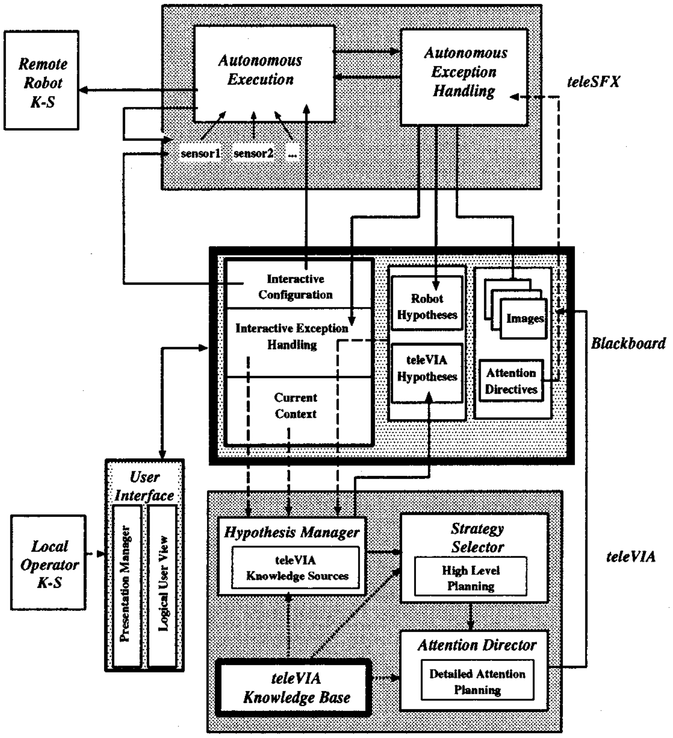

Figure 2: Cooperative Tele-Assistance System Design.

for supervision, thus allowing the operator to monitor multiple robots simultaneously; and 5) support the incremental evolution of telesystems to full autonomy. In order to achieve these goals, the intelligent capabilities of both the remote robot and the local assistant must be aligned, and this is achieved through the framework shown in Fig. 2. The components of the teleVIA part of the system are described in [8], while the details of teleSFX are presented in [5].

In the rest of this paper, we present current work on two aspects of the system design: 1) the knowledge representation used in the teleVIA knowledge base to support decision-making and image selection and enhancement heuristics, and 2) the incorporation of time into the teleSFX exception handling repertoire, and its impact on the cooperation between the two systems.

3. TELEVIA KNOWLEDGE BASE

The local intelligent assistant must maintain a repository of knowledge which can be accessed throughout the mission. The general information needed can be divided into four major categories: 1) knowledge about the robot, its capabilities and configuration; 2) knowledge about each sensor, the type of information it affords, the specifications of its data, and the type of enhancements that can be applied to that data; 3) knowledge about the current exception situation, including the type of failure, the sensors involved, the beliefs of those sensors, and the raw data used to calculate those beliefs; and 4) knowledge about the environment of operation, including its attributes and objects. The relationships of these concepts are shown in Fig. 3. Each of these concepts is formulated as a frame structure, and the general categories are linked through slots which are instantiated at the time of the mission. A suite of maintenance routines provides the ability to update information on the particular concepts needed for each new mission.

For each robot, the following knowledge is needed: robot-id, list of possible environments in which it operates, current environment, list of possible tasks it can perform, current task, list of sensors available, and current sensor list. For each sensor, the frame contains the following information in its slots: sensor-id, part-of robot-id (robot it currently belongs to), usage (type of information afforded, e.g., visible light, thermal radiation, distance, etc.), competing sensor list, complementary sensor list, horizontal and vertical field-of-view, dimensions of the data, depending on the data-type (e.g., if image data, then dimensions are height, width and depth, while numerical data just requires the number of values to be read), and a list of enhancement routines that can be applied to that particular type of data. The frame for the exception concept is the key knowledge structure that allows transfer of all the information relevant to a failure situation, and is based on the Exception Handling Knowledge Structure (EHKS) produced by the teleSFX

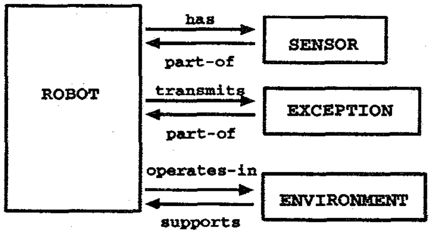

Figure 3: Knowledge Base Concept Relationships

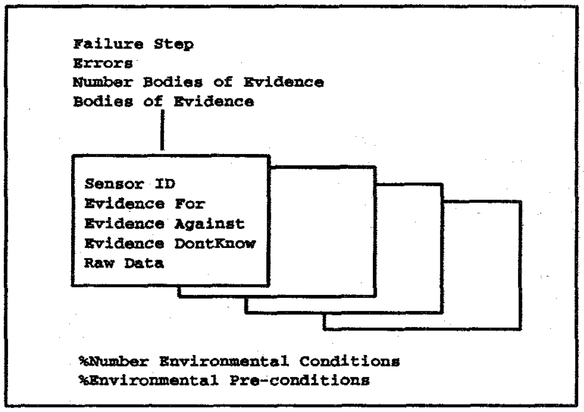

Figure 4: Exception Handling Knowledge Structure (modified from Chavez 1994)

exception handling module [1]. It contains the following information, as shown in Fig. 4: a flag that describes whether the failure occurred in the pre-proceesing step or during the fusion step, the state failure conditions, and the number of bodies of evidence together with a list of subframes describing the information about each sensor involved. The EHKS also contains slots related to environmental pre-conditions which are used by the autonomous exception handling routines of the robot. Some of this information is duplicated in the local environment frame, and therefore, at this time, this part of the original knowledge structure is not utilized by teleVIA.

The information about the environment that is represented in the frame structure includes attributes such as light intensity, ambient temperature, and a list of expected objects and dimensions (if known). Some of this information duplicates the Environmental Pre-Conditions which are checked by the robot in its exception handling activities. However, it is expected that in the case of the local assistant, more information about the environment may be stored and utilized not only in the diagnostic activity, but also for knowledge-based selection of image enhancements. It is also planned to link in encyclopaedic and cartographic knowledge for the benefit of the human operator.

4. TELESFX EXCEPTION HANDLING

In this section, a strategy is described for incorporating the role of time in constraining the exception handling activities at both the remote and the local. This is expected to allow robots to operate more effectively and reliably in domains with hard deadlines without increasing cognitive overloading of the operator. Exception handling, in this context, is defined to be the process of detecting a sensing failure, classifying the cause(s), and recovering by instantiating a new sensing plan. A *sensing failure*, or *exception*, is declared when the perceptual processing needed to support a motor behavior is not able to return a percept with a high degree of certainty. Sensing may fail for one or more of the following three reasons: a sensing malfunction has occurred (e.g., broken camera lens), the environment has changed with deteriorative effects on sensing (e.g., the lights are turned off), or the remote has errant expectations (e.g., is told to look for something that isn't there).

One objective of the teleVIA-SFX system is to allow the remote to be as self-sufficient as possible and to demand operator interaction only when there is no other safe option. The scheme described here specifies when and for how long the remote can maintain autonomous operations while attempting to identify and recover from a sensing failure. It also specifies when the remote must seek help from the operator, even though it has not necessarily exhausted its own autonomous problem solving resources.

It is posited that there are two natural deadlines in exception handling. First, there is the time that the system can afford the remote to autonomously classify and recover from the sensing failure. Second, is the time the system can devote as a whole to exception handling, either at the remote or the local, before it must abort the behavior and do something else.

In computing these deadlines, it should be noted that when a sensing failure occurs, a remote may be able to continue executing the behavior for a period of time in a "dead reckoning" mode. The period of time from the detection of a sensing failure by a remote to when it cannot safely continue executing the behavior will be designated as t_s, the time remaining until the execution of the behavior must be suspended. During this time the operator does not need to be involved while the remote is attempting to autonomously recover from a failure; this allows the operator to continue with current tasks without needless interruption. If the remote is successful, a message can be logged with the operator to immediately read or acknowledge it since the problem has been handled with almost no time delay.

If the remote does not resolve the sensing failure before t_s, then execution of the behavior is suspended. Ideally, suspension would mean that the robot would assume a fall-back or "defensive" state, allowing it to remain stationary and continue autonomous or cooperative exception handling. Unfortunately, the robot may not be able to maintain this fall-back state indefinitely; other behaviors or overarching mission parameters which are not affected may need to move the robot away from the sensing region where the failure occurred, disrupting its ability to analyze the cause of the failure. Consider the operation of a mobile robot in a highly radioactive environment. If the robot has CCD cameras, it will want to reduce unnecessary exposure to hard radiation. If the robot is not making progress on its task, it may be part of its mission to return to a shielded area.

The upper bound on how long the system can tolerate the suspension of the behavior before it has to effectively abort it is designated as t_a. If the remote is able to continue its exception handling in the interval between t_s and t_a, the operator must still be informed that the remote has entered the fall-back state. If the operator is busy and the time remaining until a behavior abort is long, the operator may choose to let the system continue to exhaust its autonomous capabilities before requiring human interaction.

Currently we are exploring the feasibility of letting t_s be the deadline for the remote exception handling activity and using t_a as a factor to influence the intelligent assistant data collection and presentation activities. This arrangement is practical and produces a reasonable overall system response, as can be seen by the following canonical cases.

$t_s = 0$. In this instance, the behavior at the remote cannot operate for any length of time in a dead-reckoning mode. The remote exception handler immediately transfers control to the local without attempting to solve it autonomously. This has the advantage of notifying the operator that a behavior has been suspended. It does not interfere with autonomous exception handling, since the remote handler is a subset of the local and the local can instruct the remote to continue classification and recovery under the intelligent assistant's supervision.

This case exemplifies what would happen if the remote perceptual process encounters a problem immediately upon instantiation, and so has no belief in the percept (and thereby no basis for dead-reckoning), or if the remote's survival depends on the behavior (for example, it might be deemed dangerous for the remote to attempt to navigate, no matter for how short a time, without sensing for obstacle avoidance).

$0 < t_s < t_a$. This is the nominal case, where the remote has some time available for exception handling without local supervision. One of three events might transpire during the time until t_s: the failure may be successfully classified and the remote recovers autonomously; the classification process may reach a point where it can go no further without human assistance and voluntarily transfers control to the local; or the classification process may still be active but the deadline is reached and control is by necessity passed to the local. In the last case, the local intelligent assistant can instruct the remote to continue its autonomous exception handling activities, but the operator is aware that the behavior has been suspended.

$t_a = 0$. This condition could arise when the remote is operating under safety-critical constraints and any change in the situation requires human intervention. Control would be passed immediately to the local, and all the exception handling would be done under the direct supervision of the operator.

$t_s \geq t_a$. In this situation, the remote is prevented from operating as long as might be theoretically possible (t_s) due to some other consideration which set $t_a < t_s$. Control must be immediately passed to the local, even though the remote could operate in dead reckoning mode for $\mid t_s - t_a \mid$, in order to insure that the local will have some time to gather and store any relevant data prior to aborting the behavior.

The interval $\mid t_a - t_s \mid$ is of particular importance to the local when the remote exception handler cannot recover. A large interval indicates that the remote can safely sit and wait for further directions from the local. A small interval serves as a warning that the remote may have to move away, that after that time, the local may not be able to request real time sensor data for help in isolating the failure(s). As a result, $\mid t_a - t_s \mid$ determines if the local requests all possible sensor data from the remote, regardless of communication bandwidth cost, in order to be sure to have it if the operator needs it. Furthermore, it can influence the choice of strategy employed by the intelligent assistant; for example, displaying sensor data at a lower resolution in order to see if the operator can immediately identify the problem. On the other hand, if $\mid t_a - t_s \mid$ is large, the intelligent assistant is under no pressure to violate its goal of minimizing communication between the systems. It can instruct the remote to continue its autonomous exception handling capabilities, or request data on demand from the operator.

Thus, the operator has three levels of supervisory involvement in exception handling in the teleVIA-SFX architecture. First, the operator does not need to participate in exception handling activities if 1) the remote is performing autonomous exception handling prior to t_s or 2) the intelligent assistant is continuing the remote's autonomous exception handling in the interval $\mid t_a - t_s \mid$. The operator is informed that exception handling has commenced but does not require the operator's attention. If the failure is resolved autonomously, the success will be posted. Again, the operator does not necessarily have to attend to that posting, and can continue to focus on other supervisory activities. Second, the operator may have cooperative supervisory duties. These would occur when 1) neither the intelligent assistant nor the remote was able to recover from the failure autonomously, or 2) a rapidly changing situation requires the operator to be aware of what is happening. In these cases, the teleVIA-SFX interface would assist the operator. Third, the operator may assume total supervisory control of the remote at any time.

5. CONCLUSIONS

It should be noted that in the initial version of the teleVIA knowledge structures, there is no accommodation for the role of time. With the development of the teleSFX time strategy described in the previous section, it is clear that some modifications are needed. In particular, the sensor frame must be expanded to include a time-out slot, which specifies time constraints related to individual sensors. For example, the inframetrics camera uses liquid nitrogen (LN2) as a reference temperature. As the LN2 evaporates, however, the camera gets warmer, and the intensity values of the image shift due to the diminishing difference between camera temperature and thermal output of the scene. Unfortunately, the period of uncompromised data collection is dependent on the exact amount of LN2 put in the camera reservoir. Thus, this type of time constraint value will have some uncertainty associated with it. Other sensors may have some sensitivity to environmental conditions. In addition, it appears that there is now a need for a task frame, which can also hold task-specific time-out information. This may include an overall deadline for the mission itself, as well as scheduling constraints for subtasks.

One of the challenges in this project is the lack of a strong domain theory, due to individual robot configurations and constraints of the applications. Often strategies must be tailored to specific instances, and it is not known in advance how the different components of the robot itself will behave under certain circumstances. In the ideal situation, the local intelligent assistant will have more knowledge from which to generate hypotheses and perform problem-solving than the robot. However, currently, the foundation of that knowledge must be based upon the robot's own intelligence.

6. ACKNOWLEDGEMENTS

This research is currently supported by the Army Center of Excellence in Information Science at Clark Atlanta University, ARO Grant # DAAL03-92-G-0377. Part of section 3 includes work performed by the third author for the Master's thesis at Clark Atlanta University.

7. REFERENCES

[1] Chavez, G.T., "Exception Handling for Intelligent Sensor Fusion". MS Thesis T-4422, Dept. of Math. & Computer Sciences, Colorado School of Mines, Golden, CO, 1994.

[2] Coiffet, P. and Gravez, P. "Man-Robot Cooperation: Towards an Advanced Teleoperation Mode". In *Intelligent Robotic Systems*, S.G. Tzafestas (Ed.). Marcel Dekker Inc.: New York, 1991, pp. 593-636.

[3] Edwards, G.R., Burnard, R.H., Bewley, W.L. and Bullock, B.L., "The Ground Vehicle Manager's Associate", *Tech Report AIAA-94-1249-CP*, 1994, pp. 520-526.

[4] Giralt, G., Chatila, R. and Alami, R. "Remote Intervention, Robot Autonomy, and Teleprogramming: Generic Concepts and Real-World Application Cases". *Proc. of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993, pp. 314-320.

[5] Murphy, R.R., "Robust Sensor Fusion for Teleoperations". *1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 2-6, 1993, vol. 2, pp. 572–577.

[6] Pang, G.K.H. and Shen, H.C., "Intelligent Control of an autonomous mobile robot in a hazardous material spill accident - a blackboard structure approach". *Robotics and Autonomous Systems 6*, 1990, pp. 351-365.

[7] Pin, F.G., Parker, L.E. and DePiero, F.W., "On the design and development of a human-robot synergistic system". *Robotics and Autonomous Systems 10*, 1992, pp. 161-184.

[8] Rogers, E. and Murphy, R.R., "Tele-Assistance for Semi-Autonomous Robots". *Proc. of AIAA Conference on Intelligent Robots in Field, Factory, Service and Space*. NASA Conference Publication 3251, 1994, pp. 500-508.

[9] Rogers, E., "Cognitive Cooperation Through Visual Interaction". *Knowledge-Based Systems*, Vol. 8, Nos. 2-3, April-June 1995, pp. 117-125.