Design Requirements and Framework for a Robotic Infrastructure System

JOSHUA D. JACKSON, DALE W. CALLAHAN, DONALD S. APPLEBY, and LEA B. CALLAHAN
Department of Electrical and Computer Engineering
University of Alabama at Birmingham
Birmingham, AL 35294
USA
www.uab.edu

Abstract: Robotic Infrastructure Systems will soon become an integral part of industry, offices, and our homes. They will be able to be controlled remotely from the Internet, and have interfaces to many systems and databases. However, current robots are mostly used for stand-alone manufacturing operations, toys, and scientific research. In order to exist as part of the critical infrastructure, robots must be able to interact with multiple users, authenticate users, track usage and users, and provide feedback to the user. The critical element that will allow such functionality is the communications channel – which will most likely be the Internet. To make proper use of the Internet, system architecture and protocols must be developed to support robots and the unique functions they will provide to the users. This paper addresses the issues that must be accounted for in developing architecture for robotics within the infrastructure. These issues are the basis for design decisions in developing robotic infrastructure systems.

Key-words: Protocol, network, robotics, Internet, TCP/IP, telemedicine, sensing

1. Introduction

Robots are poised to become more than specialized applications with limited access and control. Robotic systems have great promise as becoming part of the infrastructure of many systems. Such systems might include robotics that can be used in medicine for remote access to patients, doctors, and specialized test equipment. Such applications are often referred to as telemedicine. Robotically controlled laboratories will allow scientists from all over the world to take part in experiments remotely using very specialized equipment. Such ability will allow a scientist to link to a Mars rover and initiate experiments from his office. Later the results of such experiments would be available via some electronic access or even an email message from the rover. Homes will largely be treated as robotic systems, with connections to the many robotics components in the home, such as appliances, heating and cooling systems, and security systems.

Anytime users need to be able to remotely control machines or systems to collect data – a robotic system in the infrastructure would provide the tools. However, before such systems can be put in place, several things will be required. First is obviously the communication architecture. However, with the prominence of the Internet this problem may be solved, at least in part. Once communication is established a system must be developed which allows a distributed set of users to have access to the robots and understand their capabilities and limitations. Security will always be an issue, and different robotic applications will have differing levels of security concerns.

The contribution of this paper is to provide a critical analysis of the design issues involved in developing robotic infrastructure systems (RIS).

2. Existing Scenarios

There have been (and exist) several systems that allow users to remotely control robotics using the Internet. Details of many of these applications are in the book Beyond Webcams by Goldberg and Siegwart [1]. Of particular interest is the early work (1994) in the Mercury Project [2], which used a robot and a camera to allow a remote user to dig through sand to find an artifact. The Mercury project was a first attempt to make use of the HTTP (Hypertext Transfer Protocol) Internet protocol to control a robot. The Mars Pathfinder project came in 1998, which allowed scientists to remotely work on the Pathfinder project via the Internet making use of JAVA scripts [3]. The PUMA paint robot allowed the remote user to be an artist from afar [1]. Many other Internet controlled robotics systems have been and continue to be developed, but mostly as scientific experiments. It is based on these experiments and the scientists' conclusions that the authors have developed the set of design issues that will need to be addressed when developing architecture and protocols for robotic systems.

These systems have made use of technologies, protocols, and standards as they have become available - including HTTP, CGI (common gateway interface), JAVA, and XML (Extensible Markup Language), VRML (Virtual Reality Markup Language), and CORBA (Common Object Request Broker Architecture). The development of these systems has helped to identify two critical issues involved in Internet controlled robotics.

Security – Being on the Internet allows everyone access to the communications channel. Robotic systems for experimentation and science (such as some listed above) might allow everyone unlimited access. But, robotics systems in the infrastructure may have moving actuators that can be dangerous in the wrong hands. Also, like other Internet systems, some robots may deal with sensitive data, such as a medical diagnostics robot. And since robots may be designed for remote maintenance, the use of different security privileges will become an issue.

Time Delay – Depending on the system, round trip delays can be in the order of seconds for standard Internet systems to perhaps hours for space based robots. This time delay may put the user in a "move-and-wait" strategy. Further complications with the Internet as the medium are the uncertainty of the delay or even the uncertainty of packet arrival at all. Therefore robotic systems may have similar issues to VoIP (Voice over Internet Protocol) or Video over the web, in which the QoS (quality of service) becomes a key issue in the protocols. Like VoIP, robotic protocols will not be able to rely on the standard TCP/IP protocols alone to provide this QoS. Furthermore, the delay based QoS parameters for robots will likely be different than those for voice or video.

When robots move to widely deployed infrastructure systems, other issues arise. Some of these will be developed in section 5.

3. Characteristics of Robotic Systems

Before getting into the characteristics of robots themselves, the definition of an RIS must be given. RIS will consist of all the elements required to allow users to physically manipulate in the remote environment and collect data from a variety of sensors. Sensors would include devices such as cameras, thermometers, location detectors (of the entire robot or the robotic manipulators), and many other types. In order to provide these functions to the user, RIS will include communications, databases, actuators, and sensors. These may all be self-contained on a robot, or they may be distributed amongst many elements that makeup the system.

With this working definition, the following characteristics are listed for RIS. Many of these are the basic characteristics of individual robots.

3.1 Functions

Functions will include manipulation and movement via actuators, feedback to users via force feedback or haptic controls, data collection or sensing, and communications (to the controller, peers, or a slave).

3.2 Control Architecture [4]

- One to One – One user controlling one robot.
- One to Many – One user controlling more than one robot.
- Many to One – Many users controlling one robot.
- Many to Many – Many users controlling more than one robot.

3.3 Control Modes [4]

- Direct Control - User controls every single action of the robot through primitive commands.
- Supervisory Control – Robot operates in an autonomous mode and interacts with the user when the robot encounters a situation it cannot handle.
- Learning Control – The robotic systems develop effectiveness through learning either with the help of the user or through preprogrammed learning algorithms. Therefore, the robot might begin in the direct control mode and move over time to an autonomous system.

3.4 Administrative Features

- Security – Authenticate users. Access privileges. Allowing different levels of security to different users. Encryption algorithms.
- Billing – Since the RIS will be infrastructure systems, usage based billing may be required. The RIS will have billing parameters and protocols.
- Data collection and data queries – Data collected for one user could be retrieved when another user makes the same request at a later time. The system will need to be able to store and query data. This storage of data could also be a security

issue since one user might not want another using their data.

4. Future Infrastructure Robotics

The infrastructures of existing systems yield some important questions for robotics engineers. For instance, the medical field is based on medicine and medical practitioners. But in addition to the skilled workforce, a medical infrastructure containing MRI, CAT scans and other diagnostic and medical facilities has become standard to the practice of medicine. Similar situations can be seen in many fields such as the transportation, communication, and power industries. Industries have become highly dependent on their infrastructures. The business arrangements between the owners of the infrastructures and the customers of the businesses are critical to the existence of the relationships. Doctors do not typically own the advanced diagnostic tools; they use the tools of service providers. These service providers in turn bill the doctors or the patient directly. Likewise, the airlines do not own the terminals and often do not own many of the services needed to keep the airlines running.

Remote controlled robotics will soon be playing a larger role as a type of infrastructure. Consider for example a Mars rover robot. If scientists could connect to this robot through the Internet and perform their own experiments, NASA could bill the scientists for time used. While sitting at his desk a scientist could connect to the rover, learn the billing structure for time and functions, and then remotely dig into the dirt on Mars and do his own chemical analysis on the soil. The sensors on the robot would be used to collect and send the resulting data back to the scientist, along with billing information. To perform such a scenario, several things must happen. The scientist would have to be able to establish a connection and be authenticated. Then, the scientist would have to learn about the capabilities of the robot through some sort of service discovery protocol, such as exists in Bluetooth [5]. The current state of the robot, the position of all of the robotic movements, the range of motion, the sensors on board and their range and sensitivities would all have to be communicated to the user.

While this interplanetary robotic control over the Internet might be a ways away (Jet Propulsion Laboratory does have in the works an InterPlaNet plan [6]), many other situations might call for such functionality. The University of Alabama at Birmingham was involved in a Telepath [7] project in which a pathologist would remotely access a microscope and view slides of patients and make a diagnosis. This system allows a single pathologist to cover several smaller cities without having to drive from hospital to hospital.

5. Design Issues for Infrastructure Robotic Systems

The list of design issues below is based on the previous research on Internet controlled robots and experiences from other infrastructure systems. Some of theses issues have been discovered by researchers, while other issues are new contributions. They have been compiled as a list of design issues.

5.1 Establishing a communications channel

It is expected the connections to RIS will be accomplished via the Internet, and therefore via the TCP/IP protocol. The initial connections and responses might be no different than typical Internet connections, but other protocols will be needed immediately after the connections are established.

5.2 Security

Users must be validated. Also, different levels of permission will need to be established. These levels of security are no different from typical Internet connections to databases and servers, but with robotics new issues of security do arise. One issue is related to having a robot perform a test and analyze results. The system may need to store results for future use. Unlike a database system, where query results can all be dumped and additional duplicate queries can be run with minimal overhead, robotic systems might have physically moved to another position and therefore cannot easily reconfigure itself to run a test again. However, one user might not want their tests results shared with anyone. Rules must be in place to determine the security issues of sharing test results.

5.3 Service discovery

Once connected to a RIS, the user will need to be able to determine the types of services offered by the RIS. What sensors are available? What actuators can be used? What are the ranges of motion? What are the sensitivities of the sensors? Is force or haptic (sense of touch) feedback available? What are the initial positions or conditions of the actuators, sensors, and the entire robot itself? Billing information might also be provided during service discovery.

5.4 User interface

The user interfaces connected to RISs should automatically be rendered based on the type of interface the user has (PC with a web browser, PDA, cell phone, other). During service discovery, the user might have the option of choosing of with which sensors and actuator he wants to interact. The user interface could then be automatically rendered indicating the sensors and actuators and the initial positions.

5.5 Multiple users queue

The RIS would need to be able to interface with multiple users. Similar to databases and other systems, but now the functions are not logical data but physical systems. The physical nature of the RIS can create a great deal of latency since physical operations will likely take longer than queries. A session might also need to be 100% complete with the previous user's session first before moving to the next. Alternatively, some robotic functions would be mutually exclusive – such as a single arm to dig in the dirt on Mars and a grasping of the hand on the arm. If user A is using the arm, then user B could not use the tools to grasp. Other functions might be independent. For example, user A might have used the robotic arm to pull a dirt sample, which might have been given to another internal actuator to do analysis. Now the robotic arm is free, but not the whole system. Scheduling multiple users will be more complex in this system than databases. Simple FIFO or LIFO algorithms will not work. Additional complications arise since the entire robot might move, such as in the Mars rovers. If three requests (x, y, z) are in queue to perform a soil analysis, but request x and z are in the same physical location of the robot, then the system would want to perform these first before performing request y. System specific optimization and resource allocations must be developed, yet common protocols must exist to interface these parameters to users as needed.

5.6 Other Concerns

- Completion of session. How do you know when a session is complete? Will the user need to request the time slots, or detail all the operations before the first operation is performed?
- Billing. Perhaps the systems will have several billing policies. Billing per time used or based on types of functions performed. One thing is clear; an RIS will have monetary value.
- Learning algorithms. Since latency will always be an issue, can learning algorithms be used to determine frequent patterns? Do users always to a chemical analysis after taking a soil sample? Should the RIS decide to do it while waiting on that command? Or does it simply hold the soil until told differently, therefore blocking other requests?
- System diagnostics and self-healing. The systems should be designed with failure in mind, but at different levels depending on how easy it will be to obtain access and the cost of down time. Clearly space travel robots must have a great deal of thought into using self-healing and remote diagnostics and repair. But all robots in the infrastructure will also need some level of remote maintenance to maximize availability.
- Maintenance. How will software updates and other administrative activities be performed?
- Virus Control. What new types of viruses will exploit RIS?
- Batch processing and user scheduling. How will the robot schedule users? Will it allow batch processing and notification of results as mentioned previously? Or will the system notify users of a scheduled time slot to access the system? How will these notifications take place? Email or other TCP/IP based protocols?

6. Requirements Framework

Using the design issues and the potential future of robotic systems, a framework of requirements can be developed.

- IP based – The main communications model for a new protocol would have to be IP based.
- Standards – Since robots will be prevalent in networks, a standard protocol model must be built and adhered to by developers. This standard would likely come from a group such as IEEE or the IETF.
- Protocol Suite – Since many elements have to be taken into consideration (as covered in the paper) the protocol will likely be a suite of robotic protocols. The individual protocols will address the various issues. One protocol may be required for delay control while another access to common sets of databases and yet another for service discovery.
- Platform independence – The protocols must be able to work under a variety of hardware/software/operating system platforms.
- Safety considerations – Safety concerns must be built into the protocol which would allow the communication of safety concerns (mostly due to movement) over the protocol messages. Safety

measures built into the hardware of the systems must be available to the clients.

- Service discovery – Given the multitude of available services, a method of discovering services and functions is required.
- Rendering – Methods to render the browser over many types of clients will be required.
- Multiple users and queuing – Methods will be needed to accommodate and schedule multiple users or manage single user connections.
- Security – Security could take place via an authorization module or protocol and could possibly use existing protocols or methods.

7. Conclusion

This paper has attempted to examine design issues that will have to be addressed as robots move from stand-alone system to become a part of an infrastructure. Many of the design issues are similar to those found in the development of other infrastructure systems, such as databases and networks. Yet the RIS brings new design challenges along with their many opportunities.

References

[1] Goldberg, K., Siegwart, R., Beyond Webcams: An Introduction to Online Robots, MIT Press; 1st edition, 2001.

[2] Goldberg, K., Gentner, S. et. al., The Mecury Project: A Feasibility Study for Internet Robots. IEEE Robotics & Automation Magazine, pgs. 35-40, March 2000.

[3] Backes, P. G., Tso, K. S. et. al., Mars Pathfinder Mission Internet-Based Operations Using WITS, Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pgs. 284-291, 1998.

[4] Luo, R.C., Su, K.L. et. al., Networked Intelligent Robots Through the Internet: Issues and Opportunities, Proceedings of the IEEE, Vol. 91, No. 3, pgs.371-382, March 2003.

[5] Bluetooth Special Interest Group (SIG), www.bluetooth.org, Last accessed March 1004.

[6] InterPlaNetary Internet Project, www.ipnsig.org, last accessed March 2004.

[7] Grimes, G. J. et. al., Remote Microscopy for High-Resolution Real-Time Interactive Pathology, Advanced Imaging, p13ff, July 1997.