

Robopedia: Leveraging Sensorpedia for Web-Enabled Robot Control

Richard Edwards, Lynne E. Parker
Electrical Engineering and Computer Science Dept.
University of Tennessee, Knoxville TN, USA
{redwar15.parker}@eecs.utk.edu

David R. Resseguie
Computational Sciences and Engineering Division
Oak Ridge National Lab, Oak Ridge TN, USA
resseguiedr@eecs.utk.edu

Abstract—There is a growing interest in building Internet-scale sensor networks that integrate sensors from around the world into a single unified system. In contrast, robotics application development has primarily focused on building specialized systems. These specialized systems take scalability and reliability into consideration, but generally neglect exploring the key components required to build a large scale system. Integrating robotic applications with Internet-scale sensor networks will unify specialized robotics applications and provide answers to large scale implementation concerns. We focus on utilizing Internet-scale sensor network technology to construct a framework for unifying robotic systems. Our framework web-enables a surveillance robot's sensor observations and provides a web-interface to the robot's actuators. This lets robots seamlessly integrate into web applications. In addition, the framework eliminates most prerequisite robotics knowledge, allowing for the creation of general web-based robotics applications. The framework also provides mechanisms to create applications that can interface with any robot. Frameworks such as this one are key to solving large scale mobile robotics implementation problems. We provide an overview of previous Internet-scale sensor networks, Sensorpedia (an ad-hoc Internet-scale sensor network), our framework for integrating robots with Sensorpedia, two applications which illustrate our frameworks ability to support general web-based robotic control, and offer experimental results that illustrate our framework's scalability, feasibility, and resource requirements.

I. INTRODUCTION

People are becoming more and more connected every day through various Internet technologies, such as blogs, wikis, smart phones, etc., while, sensor systems that provide people with vital information remain disjoint and disconnected from each other. For example, weather sensors and river sensors do not interact within the same system and require a person to monitor two separate systems to detect flooding from heavy rains. In addition, unifying sensors and sensor networks can aid in search and rescue, persistent surveillance, quality of life, etc.

These overwhelming benefits have led to an increased interest in Internet-scale sensor networks, i.e. sensor networks that consists of heterogeneous sensors located throughout the world integrated into a single system or application. However, most of the proposed or built systems do not support data openness. We define a system's data to be open if and only if its data is freely accessible and does not require

detailed knowledge of the system's internal protocols. Therefore, under this criteria, there are three possible methods for making a system support data openness: 1) support a wide range of data formats, 2) follow a set of accepted standards, or 3) use external services to disseminate data on the system's behalf.

There have been a large number of contributions by OGC, Open Geospatial Consortium, for the second option for sensor networks and the sensor web enablement effort [1]. Several deployed sensor network systems have adopted the standards set forth by OGC, but standards are not always adopted quickly, unless it has sufficient momentum within the community as a whole. While we fully support the OGC interface standards, it would be best to balance all three criteria in such a way that any sensor or sensor networks can be integrated into a unified system.

Sensorpedia – an Internet-scale sensor integration platform, developed at Oak Ridge National Lab (ORNL) (<http://www.sensorpedia.com>), is able to present any number of heterogeneous sensors on a Google map mashup and present each individual sensor's data to a user through a single unified application [2]. Sensorpedia's key feature is that it supports data openness, by leveraging Web 2.0 technology to facilitate sensor integration into itself or other applications built external to Sensorpedia. In addition to external application support, Sensorpedia can be extended to support dynamic sensors, i.e. sensors that move throughout their environment. For example, a camera mounted to a robot can be considered a dynamic sensor.

Supporting mobile sensors can lead to a multitude of possibilities, such as a cost effective system for integrating multiple robot platforms into a single application. Since Sensorpedia supports data openness, once the robots are integrated into Sensorpedia, any number of applications could be built to utilize them. For example, consider a set of robots autonomously patrolling a perimeter and monitoring for intruders; how does one oversee the system as a whole and how can an individual instruct certain robots manually for exploration not defined by the perimeter?

Supervising the patrolling robots requires that each robot have an interface for communicating its observations to an external source, such as a monitoring station or central control unit. However, the perimeter robots create a more robust system if they are under decentralized control [3]. Thus, one must integrate each individual robot into mon-

Much of this research was performed at Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285. It was supported in part by U.S. Department of Energy under DOE Project No. 2367-T103-06, in part by SERRI, and in part by the EECS Department of the University of Tennessee.

itoring stations and then link each monitoring station into a unified application. Integrating every robot into a single unified application or multiple applications requires explicit knowledge of robotics, architecture development for supporting all the robots, and supporting multiple communication protocols if the robots are heterogeneous. Therefore, we propose Robopedia – a general Robot integration framework which leverages Sensorpedia’s data openness to abstract away a priori robot knowledge for developing general web-based robotics applications. A general web-based robotics application includes status monitoring, specifying goals for the robot, and teleoperating actuators or sensors, such as a pan-tilt-zoom camera.

Therefore, after giving an overview of previous Internet-scale sensor networks (Section II) and an in-depth overview of the Sensorpedia architecture (Section III), we present Robopedia’s architecture for incorporating mobile robots into Sensorpedia (Section IV). We then present two different Internet based robot applications that demonstrate Sensorpedia’s data openness and how Internet-scale sensor network platforms can be applied towards building general web-based robotics applications. (Section V). Finally, we close with a discussion on the benefits of applying Internet-scale sensor networks to robotics and explore potential future research areas (Section VII).

II. RELATED WORK

A. Internet Scale Sensor Networks

The increased interest in creating a world wide sensor network has lead to the proposal for several architectures, proto-type construction, and designs of standardized communications [4] [2] [5] [1]. However, most previous work has focused on providing the necessary architecture for a tightly coupled system which operates within a well-defined framework [4] [5]. For example, IrisNet focuses on data acquisition and the issues associated with querying sensor data in a distributed environment, such as cache and data consistency, and overall system performance [4]. IrisNet presents reasonable solutions for these problems, but solutions for sensor integration and application development are not specified [4]. Thus, even though an application is presented in [4], the processes for reusing the same sensor information for a new application or incorporating new sensors into the existing application are unclear. Sensorpedia addresses these concerns by introducing a loosely coupled system, which pushes data consistency and performance concerns off to the external user, in exchange for making application development and sensor integration transparent [2].

The approach of [5] focuses on building an Internet-scale sensor system with clearly defined sensor communication protocols using W3C standards. This makes application development and sensor integration easier, but each sensor and application can have its own data format; that is, even though all systems use the same communication protocol there are no restrictions on the transmitted data.

OGC’s sensor web enablement movement has been addressing the shortcomings of not having standardized data formats, in addition to defining standardized interfaces for Internet based sensors [1]. Also, a recent addition to the sensor web enablement framework extends the framework to make compliant sensors discoverable [6]. However, much like [5], the system is restricted to working with systems that can currently use these standardized interfaces and communication protocols. Therefore, Sensorpedia aims to combine the two principles by defining a single communication protocol for the entire system, while supporting data transmission in standard formats, as well as, proprietary or legacy formats [2].

B. Applications

Robopedia is motivated by previous work in mobile sensor networks and previous work in Internet controlled automation. CarTel, a notable mobile sensor network, demonstrates the necessary components for integrating a mobile sensor with an Internet application [7]. CarTel has a central application for interfacing with the mobile remote sensors and supports time-delayed messages through muling [7]. However, CarTel communicates via continuous SQL queries rather than using a standard sensor network communication protocol; that is, all transmitted data is stored locally in a small database and queries run continuously across those databases to extract information [7]. Robopedia aims to integrate CarTel’s successful concepts with standardized communication protocols and data formats for autonomous robot control.

In addition to notable mobile sensor networks, Telegarden [8], the first robot operated over the Internet, allowed users to remotely construct and maintain community gardens. Telegarden was built using a client-server architecture. The clients were people connecting to the system via the mosaic web browser and the server managed user session information, robot data transmission, and client robot access time. Telegarden proved that it is possible to construct a system for operating a robot over the Internet that can be utilized in accomplishing tasks, such as gardening. Robopedia looks to expand on these early key fundamental ideas by building a more loosely coupled system which can support multiple robots, various sensors, and actuators.

A more recent step towards constructing a networked robotics application is the Distributed Garden [9]. The Distributed Garden consists of sensor enabled pots that monitor soil moisture and a set of iCreates that use the sensor enabled pots to maintain the garden. Robopedia provides an ideal framework for scaling the system to an Internet-scale system and constructing external monitoring systems for supervising the garden robots.

Additional insight comes from [10] which focuses on studying the need for incorporating autonomous control into web based teleoperated robots. The work in [10] elaborates on issues with manually operating a robot over the Internet, such as, sudden increases in latency and packet loss. It ultimately concludes that there must be sufficient

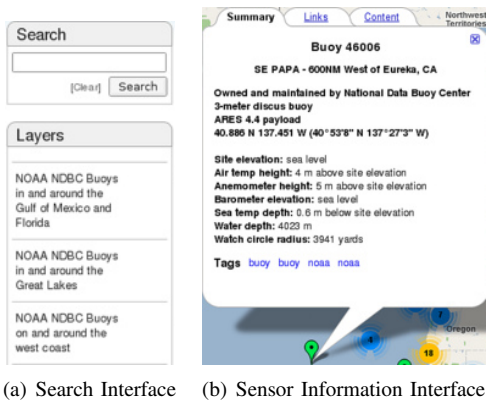


Fig. 1. Subfigure (a) shows how the Sensorpedia Web Application provides query results in an easy to read list. Subfigure (b) illustrates the Sensorpedia Web Application displaying sensor information and sensor observations when an individual sensor is selected on the map.

autonomous behaviors incorporated into the robot system, such as, obstacle avoidance and motion planning to handle sudden Internet delays or service interruptions. Robopedia addresses the network issues by modeling human driven decisions as high level goals for system-critical applications, and allowing full teleoperation control for non-system critical situations.

Further motivation is obtained by a fully automated CNC machine which allows Internet-based product fabrication [11]. However, the key insight from [11] is that the entire manufacturing environment is web-enabled and integrated into a single application. Therefore it is possible to scale Robopedia applications from operating a single robot, to operating teams or multiple sets of teams. Robopedia aims to integrate previous existing ground work, present Internet-scale sensor networks, and Web 2.0 technology to establish a standardized web interface for building Internet-based robot control applications.

III. SENSORPEDIA

Sensorpedia, as mentioned previously, is an Internet scale sensor network integration platform that utilizes Web 2.0 Technology to facilitate sensor sharing. Sensorpedia is best viewed in two parts – the web application interface and the architecture that supports it. The web application allows a user to query the system for sensors (Fig. 1(a)), view Google map mashups (Fig. 2), view general sensor information and observations (Fig. 1(b)), and register sensors with Sensorpedia.

Sensorpedia’s architecture (Fig. 3) creates a loosely coupled system supporting a multitude of sensor data formats, by using the Atom Syndication [12] format for all communication; i.e, sensors are registered via Atom documents, sensor information is presented in the form of an Atom document, etc. Atom is an XML format for notifying users about updates to web content; for example, a person could subscribe to a news website’s Atom service and be alerted each time there is an updated news article. However, Sensorpedia uses Atom because it can physically encapsulate or link remotely



Fig. 2. The Sensorpedia Web Application’s Google map mashup illustrating Sensorpedia’s ability to cluster sensors according to a user’s current level of magnification. [2]

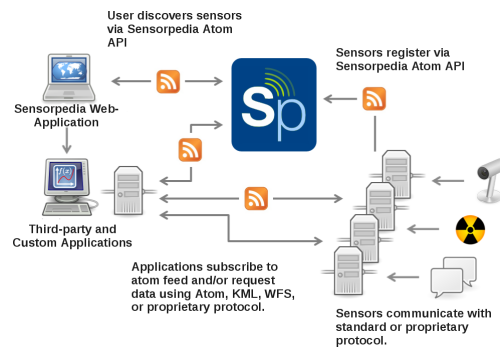


Fig. 3. Sensorpedia Software Architecture [2]

to any sensor observation data format. This includes data formats such as SOS (Sensor Observation Service), existing Atom services, HTML, and many other proprietary formats.

Sensorpedia’s architecture combines Atom with a simple lightweight RESTful [13] web service API for registering, deleting, updating, and querying sensors. This combination produces a loosely coupled system that can easily integrate existing web-enabled sensors and leaves two possible options for integrating non-web-enabled sensors. The first option is building an external service that will periodically update Sensorpedia with sensors’ current readings. However, when using a high bandwidth sensor, such as a camera, that samples the environment very frequently, option one becomes infeasible, since the Internet standard for storing binary data in a document requires that it is encoded in base 64. The second option, used by Robopedia, is to locally web-enable the sensor or sensors and integrating the web enabled system into Sensorpedia.

IV. ROBOPEDIA

A. Framework

Constructing a foundation for an Internet-scale robot system provides several technical challenges, such as, network reliability, network scalability, and providing a general interface that facilitates application development. Robopedia addresses network reliability and network scalability by separating the system into two parts, as seen in Fig. 4. The first part is the back-end robot communication server

which uses Player to receive robot sensor observations and to issue robot commands. The second part is a web-server which provides RESTful web-services for interfacing with the system. Dividing Robopedia into two components allows the two separate systems to leverage Sensorpedia for implicit communication; that is to say, the components do not have explicit knowledge about other pieces in the system. For example, the robot communication server reports observations directly to Sensorpedia, and the web-interface accesses the same observations from Sensorpedia. However, as mentioned in the previous section, the reports to Sensorpedia contain only meta-information for retrieving a set of robot sensor observations. This meta-information enables the web-interface to establish a direct connection with the robot communication server and retrieve the observations described by Sensorpedia. The implicit communication between the web-interface and the robot communication server provides Robopedia with the ability to inherently support any multitude of robot servers and web servers.

In addition to Robopedia offering a scalable and reliable robot integration platform, it provides standardized methods for integrating additional robots, additional applications, and the ability to automate connections between new robot servers and new robot web-interface. Robopedia achieves this objective currently by enforcing that all communication uses HTTP (Hyper Text Transfer Protocol). This restriction narrows the possible communicated data types to all known Internet MIME types making it easier for users to build and maintain web-based robotic applications. However, it is possible within the existing Robopedia framework to construct direct connections via a socket to facilitate communication, but this is intended only for robot commands and not for retrieving sensor observations.

B. Web-Interface

Robopedia's RESTful web-interface facilitates generic web-based application development for Internet based robot control, i.e, it provides general reusable interfaces for controlling a robot or robots via the web. However, achieving that objective requires solving numerous technical challenges, such as, robot web-enablement, data consistency, reducing latency, and designing a general robot web interface.

Web enabling a robot or robots requires a single server running a web server, a small robot communication server, and a process per sensor updating Sensorpedia with its latest web address pointing to the most recent reading. A process per sensor is reasonable, since each sensor's information arrives asynchronously, and only in the worst case will all sensor processes attempt to upload simultaneously. However, since a computer does not have an infinite amount of space, the data hosted by the local web server has a limited time window for past observations. Limiting the past number of observations may conserve hard disk space, but it creates data consistency problems with people asynchronously accessing web content.

Data consistency problems are generally solved by using time stamps for gaging the staleness of data. However, an

Internet address to an image will not provide time stamp information, and since each image is being overwritten periodically there is no method to determine if that address still points to a consistent image. For example, if someone were to store addresses for a sequence of frames from a video camera, these addresses may no longer point to frames in the original sequence. This problem can be solved by forcing an external application to download and store all data locally or have the application use only the most current readings provided by Sensorpedia. Ideally, in the future, we will implement an address validation system, but currently the external Robopedia applications use the second option; that is, to use only the most current readings provided by Sensorpedia.

Sensorpedia provides a partial solution for a general web interface, given that robot sensor information can be accessed directly from it without explicit robot knowledge. This leaves determining a suitable data format for each sensor's data type, and determining the robot command interface. Instead of defining a candidate data representation for every possible sensor type, we limited robot support to the Pioneer 3DX equipped with a camera, SICK laser range finder, odometry encoders, and sonar. All the data formats of the sensors, except the camera, were defined as comma separated files; however, there is currently no available meta-data indicating a relationship between each value in the file and its field. The camera's data format does not require meta-data since it is using jpeg images, which contain header information describing the image's basic properties.

C. Command-Interface

The Robopedia command interface is an additional set of RESTful web services that allow applications to issue commands to a program running on a robot or directly to a robot's actuators. Robopedia's command web-interface issues commands in the proprietary Player server data format to the robot communication server. Subsequently the communication server issues the instruction to the robot. Currently all robot commands are executed in FIFO (First in First Out) order, but Robopedia's command interface does not currently provide special consideration to the fact that multiple users could be trying to control the same robot. However, previous work in Internet controlled robotics has addressed this issue in great detail and our implementation can be extended easily to support such resource conflicts. The command interface can operate the Pioneer's Canon PTZ camera, as well as, issue motion commands, such as, velocity and specify goal points to the robot's onboard planner. However, the connection between the communication server and the robot web services is predefined; that is, Sensorpedia does not currently provide the required meta-information to establish the connection automatically. While the interface requires minimal robot knowledge, running the robot API on a remote web server requires the connection information to be explicitly provided.

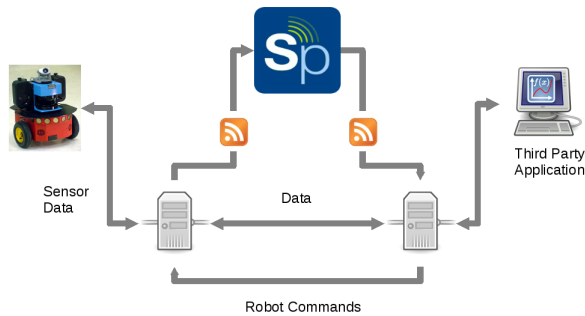


Fig. 4. Robopedia Architecture

V. ROBOPEDIA APPLICATIONS

Robopedia was tested with two different tasks – manual surveillance and autonomous navigation. The manual surveillance task involves a person teleoperating a camera via a Java web application as seen in Fig. 5. However, initial testing of the application showed that transferring a single frame over HTTP (Hyper Text Transfer Protocol) was slow, and users experienced an extremely low frame rate. Transferring frames is slow due to the HTTP connection overhead that is created from each frame request, since each request opens a connection and once the data is transferred the connection is closed. In order to maximize throughput, several frames are placed into a single image, which greatly increases the application’s frame rate. However, this requires the application to know a priori the number of frames stored per image. The number of frames in each image can be described by meta-data, but currently the system lacks the ability to describe meta-data in a non-proprietary manner. In addition, the exact number of frames per image was determined based on the total time required to receive and process a single frame. Even though this technique increases system performance, it most likely will not generalize to other systems.

The autonomous navigation application requires a user to specify a point of interest and the robot to navigate to that point. As the robot navigates, the user receives visual environmental feedback from the camera application, as seen in Fig. 5. The application is very general and loads all robot information through Robopedia; however, it suffers from deficiencies similar to the camera application. Converting an image point in the map to a physical global coordinate requires information about the number of meters per pixel. Therefore, even though the application is general, it is not completely free of prior knowledge.

VI. SYSTEM ANALYSIS

Robopedia presents an ideal system framework for constructing Internet-scale robotics applications. However, our applications, presented in the previous section, are currently capable of operating a single robot. While we were not able to evaluate Robopedia’s performance at operating multiple robots, we were able to measure the system’s ability

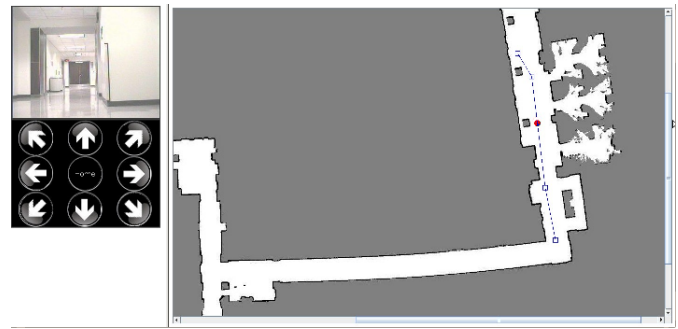


Fig. 5. (Left) The Canon PTZ (Pan,Tilt,Zoom) Camera application, shown above, interfaces with the observation web-interface to retrieve camera images and uses the arrow buttons to issue camera poses to the camera command interface. (Right) The Navigation application allows a user to command the robot to move to any valid point in the map. A valid point in the map is one that is reachable by staying within the lightly shaded regions. Darkly shaded regions are considered to be walls or obstacles.

to support multiple users accessing a single color camera sensor. The color camera sensor is the most bandwidth intensive component available within Robopedia. The camera currently produces frames at 30FPS (Frames Per Second) and each frame is compressed to jpeg with a resolution of 320x240 (Width by Height).

In addition to evaluating the systems capability to support multiple users accessing a high bandwidth component, we also wanted to explore Robopedia’s performance with the web-server connected to a standard at home low bandwidth Internet connection. Sufficient performance on a low bandwidth connection would imply that it is possible for Internet-Scale robotics systems to integrate into users’ homes, as well as, provide a means for hobbyists to integrate their personal robots into Robopedia.

The two experiment configurations are as follows: 1) Robopedia connected to a high-bandwidth Internet connection capable of reaching and sustaining a 4 MiBs (Million Bytes Per Second) upload rate and up to twenty local network computers attempting to access the same camera sensor repeatedly, 2) Robopedia connected to a low-bandwidth at home Internet connection that is capable of sustaining a 1 Mbps (Million bits Per Second) upload rate and up to ten computers attempting to access the same camera.

Initial results for low-bandwidth (Fig. 6(a)) and high-bandwidth (Fig. 6(b)) connections show that the system is indeed capable of supporting a large number of users. However, the high-bandwidth connection results imply that the current implementation is not stable when scaling to a large number of users; that is to say, as the number of users increases a user’s experienced frame rate degrades and can become erratic.

We addressed this issue by implementing a multi-user caching scheme. The multi-user caching scheme places a user’s most recently retrieved observation aside for other users within the web-server. For example, when a user requests the most current observation from the web-server, the web-server will download that observation and then

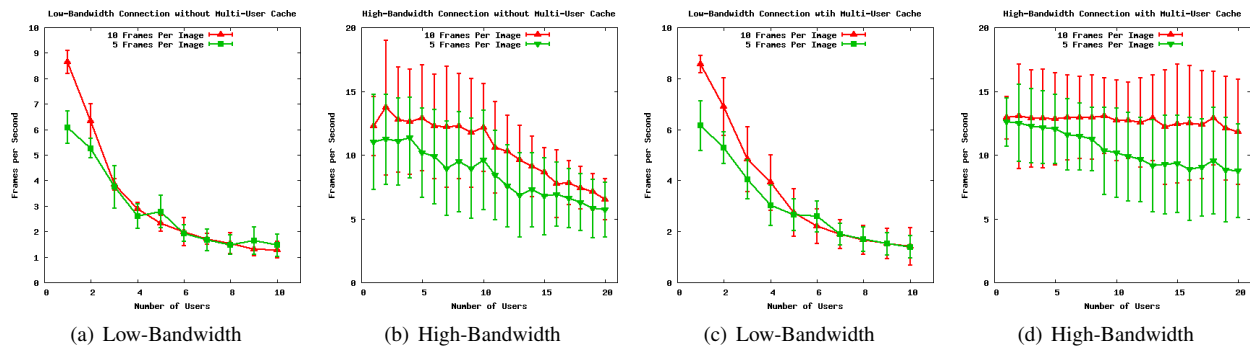


Fig. 6. These graphs measure the ability for a low-bandwidth and high-bandwidth connection to support multiple users accessing a single camera sensor without (Fig. 6(a) and 6(b)) and with (Fig. 6(c) and 6(d)) caching. Graphs display the average frame rate for 10 frames per image and 5 frames per image.

transmit it to the user. However, if the web-server stores the retrieved information locally it can instantly transmit it to the next user that makes a request. In Section IV-B we mentioned that caching HTTP links was difficult and required a link validation scheme, but this method avoids the need for a link validation scheme by downloading and storing the content locally. Additionally, the web-server can implement a local cache data staleness policy independent of external users, and independent of the content that is being stored.

Our results for low-bandwidth (Fig. 6(c)) and high-bandwidth (Fig. 6(d)) connections with the new caching scheme show great promise. While the low-bandwidth connection experienced very minor improvement, the high-bandwidth connection shows greater stability under additional users. In addition, the high-bandwidth connection results strongly suggest that the current implementation of Robopedia can support more than twenty users accessing a bandwidth intensive sensor simultaneously. Therefore, Robopedia is practically feasible for multiple users operating a single robot, and requires additional analysis to determine the feasibility of multiple users operating multiple robots simultaneously.

VII. CONCLUSIONS AND FUTURE WORK

Robopedia presents a scalable, reliable, and feasible framework for creating Internet-scale robotics applications. In addition, Robopedia establishes flexible methods for implicit communication among its components by utilizing Sensorpedia to facilitate information discovery. Robopedia's system analysis demonstrates that Robopedia can manage a large number of external users, as well as, support robot hobbyist with low-bandwidth connections who wish to integrate their robots into Robopedia. Lastly, Robopedia has expanded on current Internet-scale sensor network research by integrating autonomous mobile sensor systems and by providing an interface for Internet-scale mobile robotics application development.

Robopedia is capable of managing multiple users and one robot; there is need for further investigation into how well it supports multiple users operating multiple robots. Once Robopedia's applications are extended to support multiple robots, one must determine how Robopedia should manage

resources within the system. The exploration into resource management will answer several key questions, such as, the ultimate long term feasibility of constructing Internet-scale robotic systems, and how to manage those systems.

REFERENCES

- [1] M. Botts, G. Percivall, C. Reed, and J. Davidson, "Ogc sensor web enablement: Overview and high level architecture." OGC, Tech. Rep., December 2007.
- [2] B. L. Gorman, D. Resseguie, and C. Tomkins-Tinch, "Sensorpedia: Information sharing across incompatible sensor systems," *Collaborative Technologies and Systems, International Symposium on*, vol. 0, pp. 448–454, 2009.
- [3] A. Marino, F. Caccavale, L. E. Parker, and G. Antonelli, "Fuzzy behavioral control for multi-robot border patrol," *Mediterranean Conference on Control and Automation*, vol. 0, pp. 246–251, 2009.
- [4] R. Peng, K. A. Hua, and G. L. Hamza-Lup, "A web services environment for internet-scale sensor computing," *Services Computing, IEEE International Conference on*, pp. 101–108, 2004.
- [5] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan, "Irisnet: An architecture for a worldwide sensor web," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 22–33, 2003.
- [6] S. Jirka, A. Brring, and C. Stasch, "Discovery mechanisms for the sensor web," *Sensors*, vol. 9, no. 4, pp. 2661–2681, 2009. [Online]. Available: <http://www.mdpi.com/1424-8220/9/4/2661>
- [7] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *4th ACM SenSys*, 2006, pp. 125–138.
- [8] K. Goldberg, H. Dreyfus, A. Goldman, O. Grau, M. Gržinić, B. Hannaford, M. Idinopulos, M. Jay, E. Kac, and M. Kusahara, Eds., *The robot in the garden: telerobotics and telepresence in the age of the Internet*. Cambridge, MA, USA: MIT Press, 2000.
- [9] N. Correll, A. Bolger, M. Bollini, B. Charrow, A. Clayton, F. Dominguez, K. Donahue, S. Dyar, L. Johnson, H. Liu, A. Patrikalakis, J. Smith, M. Tanner, L. White, and D. Rus., "Building a distributed robot garden," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.
- [10] R. Luo and T. M. Chen, "Development of a multi-behavior based mobile robot for remote supervisory control through the internet," *Mechatronics, IEEE/ASME Transactions on*, vol. 5, no. 4, pp. 376–385, Dec 2000.
- [11] L. Wang, P. Orban, A. Cunningham, and S. Lang, "Remote real-time cnc machining for web-based manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 6, pp. 563 – 571, 2004, 13th International Conference on Flexible Automation and Intelligent Manufacturing. [Online]. Available: <http://www.sciencedirect.com/science/article/B6V4P-4D34JJ7-1/2/25c2edf457645cca7b0b0eac1c14e9c6>
- [12] IETF, "Atom syndication format," Tech. Rep., 2007. [Online]. Available: <http://tools.ietf.org/html/rfc4287>
- [13] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Department of Information and Computer Science, University of California, 2000. [Online]. Available: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>