ROS and Rosbridge

Roboticists out of the loop

Christopher Crick
Brown University
Providence RI
chriscrick@cs.brown.edu

Graylin Jay
Brown University
Providence RI
tjay@cs.brown.edu

Sarah Osentoski
Bosch Research and
Technology Center
Palo Alto CA
sarah.osentoski@us.bosch.com

Odest Chadwicke Jenkins
Brown University
Providence RI
cjenkins@cs.brown.edu

ABSTRACT

The advent of ROS, the Robot Operating System, has finally made it possible to implement and use state-of-the-art navigation and manipulation algorithms on widely-available, inexpensive standard robot platforms. With the addition of the Rosbridge application programming interface, interface designers and applications programmers can create robot interfaces and behaviors without venturing into the specialized world of robotics engineers. This tutorial introduces ROS and Rosbridge, and shows how quickly and easily these tools can be used to design and conduct large-scale online HRI experiments, access algorithms for autonomous robot behavior, and leverage the huge ecosystem of general-purpose web-based and application-oriented software engineering for robotics and HRI research. Tutorial attendees will learn the basics of autonomous and teleoperated navigation and manipulation, as well as interface design for online interaction with robots. During the tutorial they will design and write their own remote presence application, as well as develop strategies for incorporating autonomy and dealing with data collection.

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Artificial Intelligence—*Robotics*

General Terms

Design, Human Factors

Keywords

Robot middleware, protocols, HRI

1. SPEAKERS

- **Christopher Crick** is a postdoctoral research associate at Brown University. He has contributed ROS drivers for visual recognition and the AR Drone flying robot, as well as Rosbridge support for multi-robot systems. He has also developed and published Rosbridge-based HRI studies in human perception and robot learning.

- **Graylin Jay** is a researcher at Brown University and developer specializing in language tools. He is the lead developer for the base Rosbridge framework.

- **Sarah Osentoski** is a Research Engineer at the Bosch Research and Technology Center. She contributes to the PR2 Remote Lab project and the remote_lab stack that enables web visualization of ROS robots.

- **Odest Chadwicke Jenkins** is Associate Professor of Computer Science at Brown University. His recent work focuses on web-scale robot learning and casting robots as web services to broaden public accessibility to state-of-the-art robots.

2. MOTIVATION

Since its inception, robotics has been plagued by the fact that every lab had its own robot, unlike any other, which required the care and feeding of a platoon of graduate students well-versed in the intricacies and idiosyncracies of that machine's low-level control systems. HRI only intensifies this problem, owing to the fact that successful HRI research requires not only extensive expertise in robot programming and robotics algorithms, but in interface design and experimental procedures as well. This has made HRI an intensely collaborative field, with progress dependent upon close cooperation among people who contribute different skillsets.

It has lately, at last, become possible to decouple some of the elements of the long and intricate development cycle which has always been the rule in HRI, resulting in multiple points of failure or delay and difficulty in generalizing results beyond singleton experimental studies. ROS, Rosbridge and rosjs make it much easier to port studies from one robot to another, to design and instantly test robot interfaces, to leverage algorithms developed elsewhere, to engage large numbers of experimental subjects over standard web interfaces, and to apply arbitrary programming languages, tools and libraries to problems in robotics and HRI.

HRI'12, March 5–8, 2012, Boston, Massachusetts, USA.
ACM 978-1-4503-1063-5/12/03.

This tutorial will demonstrate how much more easily and quickly HRI experiments can be designed, tested and fielded on a large scale by using the tools we are developing.

3. AUDIENCE

This tutorial is designed to appeal to researchers in HRI who occupy several different niches. For the roboticists who program robot interactions and behaviors, ROS provides a huge array of pre-existing code implementing most common algorithms in robotics, and running on a large variety of platforms. Rosbridge makes it possible to pull that functionality into a user's existing software and systems, seamlessly and easily. However, the tutorial is also explicitly intended for HRI researchers who are not themselves roboticists. Rosbridge and rosjs make it easy to design interfaces and implement robot behaviors without bringing robotics experts into the loop.

A passing familiarity with web technologies (HTML and Javascript) and a desire to work with robots are the only prerequisites for the tutorial. The program will be of interest to the following:

- Anyone looking for an introduction to ROS
- Roboticists interested in using Rosbridge to apply ROS's algorithms and tools to their own preexisting software, robots, and development methodologies
- HRI researchers looking to shorten the development cycle between experimental design, implementation and subject testing
- Researchers interested in collecting large-scale HRI data over the internet
- Designers interested in producing functional robotic interfaces quickly and without requiring constant input from roboticist programmers

4. OVERVIEW

1. Introduction to robot middleware (approx 45 minutes)
   (a) Message protocols
   (b) ROS messages and services
   (c) Using and understanding messages
2. What ROS offers (approx 100 minutes, plus break)
   (a) Platform independence
   (b) Tools and algorithms
      i. Navigation
         A. Localization, path planning and AMCL
         B. Mapping and SLAM
      ii. Manipulation
         A. Object recognition and the Point Cloud Library
         B. Action servers
      iii. Control and state machines
   (c) Visualization and RVIZ
3. Afternoon: Rosbridge (approx 45 minutes) *Note: attendees who are already familiar with ROS can attend only the afternoon session, starting here*
   (a) Messaging protocol and JSON
   (b) Connecting to ROS from anywhere
   (c) Managing a robot through a Rosbridge connection
   (d) Authentication
   (e) Data collection
4. Building online interaction tools (approx 90 minutes, plus break)
   (a) Javascript, rosjs and wviz
   (b) Serving video with mjpeg_server
   (c) Rosjs and wviz features
      i. WebGL visualization
      ii. Teleoperation
      iii. Pick and place
      iv. Interactive markers
   (d) Teleoperation in fifteen lines
5. Designing large-scale experiments in HRI (approx 30 minutes)

5. REFERENCES AND LINKS

[4] lays out the design philosophy behind ROS. A huge amount of information, including tutorials, installation tools for various robots, and an exhaustive listing of available tools can be found at http://www.ros.org. Rosjs is first described in [3] and is described at http://www.ros.org/wiki/rosjs. Rosbridge and the philosophy behind its creation are found in [1]. Both Rosbridge and rosjs are available for download at http://code.google.com/p/brown-ros-pkg, and can also be used as part of a standard ROS installation. [2] describes HRI experiments conducted using the infrastructure and methodologies to be covered in the tutorial.

6. REFERENCES

[1] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins. Rosbridge: Ros for non-ros users. In *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[2] C. Crick, S. Osentoski, G. Jay, and O. C. Jenkins. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th ACM/IEEE Conference on Human-Robot Interaction*, 2011.

[3] S. Osentoski, G. Jay, C. Crick, B. Pitzer, C. DuHadaway, and O. C. Jenkins. Robots as web services: reproducible experimentation and application development using rosjs. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, 2011.

[4] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *Proceedings of the Open-Source Software Workshop of the International Conference on Robotics and Automation*, 2009.