HRItk: The Human-Robot Interaction ToolKit Rapid Development of Speech-Centric Interactive Systems in ROS

Ian Lane[1], Vinay Prasad[1], Gaurav Sinha[1], Arlette Umuhoza[1], Shangyu Luo[1], Akshay Chandrashekaran[1] and Antoine Raux[2]

[1] Carnegie Mellon University, NASA Ames Research Park, Moffett Field, California, USA
[2] Honda Research Institute, Mountain View, California, USA
lane@cs.cmu.edu, araux@honda-ri.com

Abstract

Developing interactive robots is an extremely challenging task which requires a broad range of expertise across diverse disciplines, including, robotic planning, spoken language understanding, belief tracking and action management. While there has been a boom in recent years in the development of reusable components for robotic systems within common architectures, such as the Robot Operating System (ROS), little emphasis has been placed on developing components for Human-Robot-Interaction. In this paper we introduce HRItk (the Human-Robot-Interaction toolkit), a framework, consisting of messaging protocols, core-components, and development tools for rapidly building speech-centric interactive systems within the ROS environment. The proposed toolkit was specifically designed for extensibility, ease of use, and rapid development, allowing developers to quickly incorporate speech interaction into existing projects.

1 Introduction

Robots that operate along and with humans in settings such as a home or office are on the verge of becoming a natural part of our daily environment (Bohren et al., 2011, Rosenthal and Veloso 2010, Kanda et al., 2009, Srinivasa et al., 2009). To work cooperatively in these environments, however, they need the ability to interact with people, both known and unknown to them. Natural interaction through speech and gestures is a prime candidate for such interaction, however, the combination of communicative and physical actions, as well as the uncertainty inherent in audio and visual sensing make such systems extremely challenging to create.

Developing speech and gesture-based interactive robots requires a broad range of expertise, including, robotic planning, computer vision, acoustic processing, speech recognition, natural language understanding, belief tracking, as well as dialog management and action selection, among others. This complexity makes it difficult for all but very large research groups to develop complete systems. While there has been a boom in recent years in the development and sharing of reusable components, such as path planning, SLAM and object recognition, within common architectures, such as the Robot Operating System (ROS) (Quigley, 2009), little emphasis has been placed on the development of components for Human-Robot Interaction although despite the growing need for research in this area.

Prior work in Human-Robot Interaction has generally resulted in solutions for specific robotic platforms (Clodic et al., 2008) or standalone frameworks (Fong et al., 2006) that cannot be easily combined with standard architectures used by robotics researchers. Earlier work (Kanda et al., 2009, Fong et al., 2006) has demonstrated the possibilities of multimodal and multiparty interaction on robotic platforms, however, the tasks and interactions explored until now have been extremely limited, due to the complexity of infrastructure required to support such interactions and the expertise required to effectively implement and optimize individual components. To make significant progress, we believe that a common, easy to use, and easily extensible infrastructure, similar to that supported by ROS, is required for multi-modal human-robot interaction. Such a framework will allow researchers to rapidly develop initial speech and gesture-based interactive systems, enabling them to rapidly deploy systems, observe and collect interactions in the field and iteratively improve system components based on observed deficiencies. By using a common architecture and messaging framework, components and component models can easily be upgraded and extended by a community of researchers, while not affecting other components.

Towards this goal we have developed HRItk[1] (Human-Robot-Interaction toolkit), an infrastructure and set of components for developing speech-centric interactive systems within the ROS environment. The proposed toolkit provides the core components required for speech interaction, including, speech recognition, natural language understanding and belief tracking. Additionally it provides basic components for gesture recognition and gaze tracking.

[1] HRItk is available for download at: http://speech.sv.cmu.edu/HRItk

NAACL-HLT 2012 Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data, pages 41–44,
Montréal, Canada, June 7, 2012.

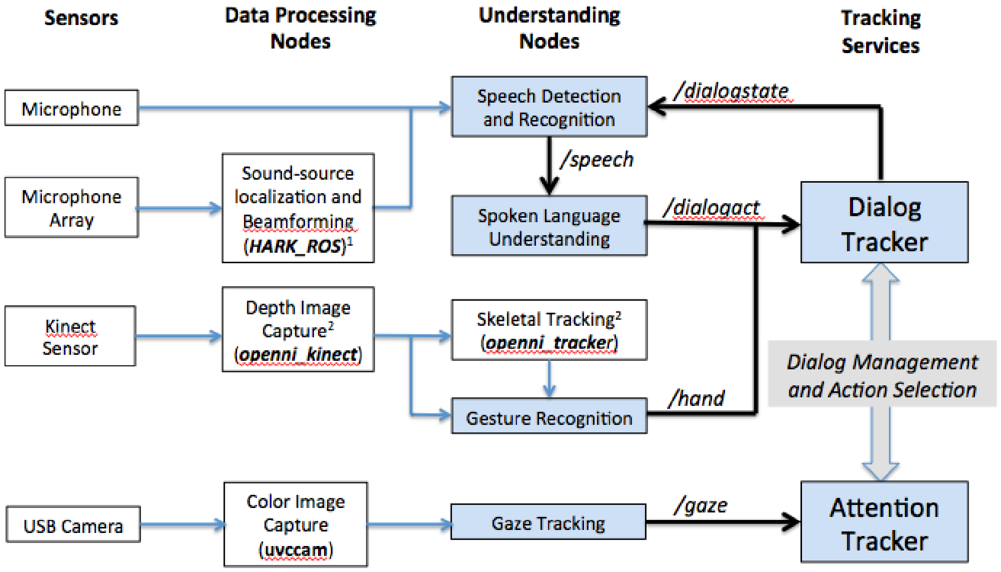
Sensors
Data Processing Nodes
Understanding Nodes
Tracking Services
Microphone
Microphone Array
Kinect Sensor
USB Camera
Sound-source localization and Beamforming (HARK_ROS)^{1}
Depth Image Capture^{2} (openni_kinect)
Color Image Capture (uvccam)
Speech Detection and Recognition
Spoken Language Understanding
Skeletal Tracking^{2} (openni_tracker)
Gesture Recognition
Gaze Tracking
/dialogstate
/speech
/dialogact
/hand
/gaze
Dialog Tracker
Dialog Management and Action Selection
Attention Tracker

Figure 1: Overview of core understanding and tracking components within HRItk

2 Framework Overview

An overview of the core components in the toolkit are highlighted in Figure 1. We introduce two classes of components required for speech and multimodal interaction into the ROS framework, *understanding nodes* and *tracking services*. *Understanding nodes* are perceptual components that recognize and understand interaction events. Using input from sensors, intermediate processing nodes or other understanding components, these nodes generate hypotheses about current user input. *Tracking services* monitor the long term and continuous aspects of interaction, including user dialog goals and the user's focus of attention. These services are leveraged by components including Dialog Management and Action Selection to perform interaction. Additionally, these services provide context to *understanding nodes* enabling them to apply context-specific processing during the understanding phase.

2.1 Data Processing Nodes

The understanding components implemented in this work heavily leverage existing components developed in ROS (Quigley et al., 2009). These include the "*openni_kinect*" node, which processes depth-images from the Microsoft Kinect sensor, the "*openni_tracker*", which performs skeletal tracking, and "*uvccam*" node, which processes color images from external USB cameras. In the near future we also plan to support far-field speech recognition using the HARK_ROS toolkit (Nakadai et al., 2010).

2.2 Understanding Nodes

Understanding nodes recognize and understand events observed during interaction. As input they use either data obtained directly from sensors, preprocessed data from intermediate processing nodes or output from other understanding components. They either perform processing on explicit interaction events, such as speech or gesture input, or process continuous input such as joint position or gaze direction. The current *understanding nodes* implemented within HRItk are listed in Table 1 along with the ROS topics on which they publish.

Understanding nodes publish two forms of messages, "**state**" messages {`READY`, `START` and `STOP`}, indicating the state of the node and whether an interaction event has been detected, and "**hypothesis**" messages which enumerate the most likely observed events along with a likelihood measure for each. The specific structure of the "**hypothesis**" message is dependent on the event being observed.

2.3 State Tracking Services

In addition to understanding specific events such as utterances or gestures, an interactive system needs to track longer term and/or continuous aspects of interaction. Such aspects include user goals, which can span several utterances in a dialog, and the user's focus of attention (using, e.g., gaze and posture information). These can be defined as characterizing the *state* of the world (i.e. the user, the interaction, or the environment) at a given time, with possible reference to history.

Table 1: ROS nodes, Topics, Services and Messages implemented within HRItk

ROS Node	Topic / Service (*)	Description of Messages
Speech Detection and Recognition	speech/state speech/hypothesis speech/hypothesis/best speech/hypothesis/final speech/context	State identifying interaction event, each with a unique eventID Partial and final hypotheses generated during speech recognition. Outputs include 1-best, N-best hypotheses and confusion networks. All output contains confidence or component model scores Context indicating dialog-state, domain, task of current interaction
Natural Language Understanding	dialogact/hypothesis dialogact/context	Hypotheses of Concept/Value-pairs generated during NLU Context indicating dialog-state, domain, task of current interaction
Gesture Recognition	hand/hypothesis hand/context	Hypothesis set of Gesture-Actions with confidence measure Context indicating domain or task of current interaction
Gaze Tracking	gaze/hypothesis hand/context	Estimate of gaze direction Context listing visually salient objects within users field of view
Dialog State Tracking	dialogstate/state belief * dialogstate/context	Receives an `UPDATED` message when the belief changes Belief over the concept set specified in the service request Context indicating system actions potentially affecting belief

In addition, states can be significantly larger objects than individual event understanding results, which could unnecessarily consume significant bandwidth if constantly broadcast. Therefore, state tracking modules use ROS *services* rather than topics to communicate their output to other modules. Any module can send a message to the tracking service containing a specific query and will receive in response the matching state or belief over states.

In order to allow components to react to changes in the state, each state-tracking module publishes an `UPDATED` message to its **state** topic whenever a new state is computed.

2.4 Component Implementations

Speech Detection and Recognition is performed using a ROS node developed around the Julius Speech Recognition Engine (Lee and Kawahara, 2009). We selected this engine for its compatibility with HARK (Nakadai et al, 2010), and its support of common model formats. A wrapper for Julius was implemented in C++ to support the ROS messaging architecture listed in Table 1. Partial hypotheses are output during decoding, and final hypotheses are provided in 1-best, N-best and Confusion Network formats. Context is supported via language model switching.

In order to develop a Speech Recognition component for a new task at minimum two component models are required, a pronunciation dictionary, and a language model (or recognition grammar). Within HRItk we provide the tools required to generate these models from a set of labeled example utterances. We describe the rapid model building procedure in Section 4.

Natural Language Understanding is implemented using Conditional Random Fields (Lafferty et al. 2001) similar to the approach described in (Cohn, 2007). For example, given the input utterance: "*Take this tray to the kitchen*" listed in Table 3, three concept/value pairs are extracted: `Action{Carry}, Object{tray}, Room{kitchen}.` Similar to the speech recognition component, the NLU component can be rapidly retrained using a set of tagged example sentences.

Gesture Recognition of simple hand positions is implemented using a Kinect depth sensor and previous work by Fujimura and Xu (2007) for palm/finger segmentation. Currently, the module publishes a hypothesis for the number of fingers raised by the user, though more complex gestures can be implemented based on this model.

Gaze Tracking is implemented using ASEF filters (Bolme et al., 2009) and geometric projection. Separate ASEF filters were training to locate the pupils of the left and right eye as well as their inner and outer corners. Filters were trained on hand-labeled images we collected in-house.

Dialog State Tracking is in charge of monitoring aspects of dialog that span multiple turns such as user goal. Our implementation is based on the Hound dialog belief tracking library developed at Honda Research Institute USA. Currently, our belief tracking model is Dynamic Probabilistic Ontology Trees (Raux and Ma 2011), which capture the hidden user goal in the form of a tree-shaped Bayesian Network. Each node in the Goal Network represents a concept that can appear in language and gesture understanding results. The structure of the network indicates (assumed) conditional independence between concepts. With each new input, the network is extended with evidence nodes according to the final understanding hypotheses and the system belief is estimated as the posterior probability of user goal nodes given the evidence so far.

A request to the dialog state tracking service takes the form of a set of concept names, to which the service responds with an m-best list of concept value assignments along with the joint posterior probability.

Examples.txt

```
<Tagged example sentence>                 <Action>
@Room{kitchen}                            None
on the @Floor{fifth} floor                None
take this @Object{package}
to @Room{room 123}                        Carry
```

Structure.txt

```
<Node>                                    <Parent>
Room                                      ROOT
Floor                                     Room
Object                                    Room
```

Figure 2: Training examples for robot navigation task

3 Rapid System Build Environment

The models required for the core interaction components in the system can be build from a single set of labeled examples ("*Examples.txt*"), along with a concept structure file ("*Structure.txt*") used by the Dialog State Tracker as shown in Figure 2. Running the automatic build procedure on these two files will generate 3 new models,

The data in the "Examples.txt" file is used to train the language model and pronunciation dictionary used by the Speech Detection and Understanding Node and the statistical CRF-parser applied in the Natural Language Understanding component. Given a set of labeled examples, the three models listed above are trained automatically without any intervention required from the user. Once a system has been deployed, speech input is logged, and can be transcribed and labeled with semantic concepts to improve the effectiveness of these component models.

As explained in section 3.5, our dialog state tracker organizes concepts in a tree structure. For a given domain, we specify that structure in a simple text file where each line contains a concept followed by the name of the parent concept or the keyword ROOT for the root of the tree. Based on this file and on the SLU data file, the resource building process generates the files required by the Hound belief tracker at runtime. This "off-the-shelf" structure assumes at each node a uniform conditional distribution of children values given the parent value. These distributions are stored in a human-readable text file and can thus be manually updated to more informative values.

Using the above tools, we have developed a sample using the proposed framework for robot navigation task. The entire system can be build from a single set of labeled examples as shown in Figure 3 used to train the language model and a component to perform actions on the SLU output.

4 Conclusions

In this paper we introduce HRItk (the Human-Robot-Interaction toolkit), a framework, consisting of messaging protocols, components, and development tools for rapidly building speech-centric interactive systems within the ROS environment. The proposed toolkit provides all the core components required for speech interaction, including, speech recognition, natural language understanding and belief tracking and initial implementations for gesture recognition and gaze tracking. The toolkit is specifically designed for extensibility, ease of use, and rapid development, allowing developers to quickly incorporate speech interaction into existing ROS projects.

References

Bohren J., Rusu R., Jones E., Marder-Eppstein E., Pantofaru C., Wise M., Mosenlechner L., Meeussen W., and Holzer S. 2011. *Towards autonomous robotic butlers: Lessons learned with the PR2*, Proc. ICRA 2011

Bolme, S., Draper, B., and Beveridge, J. 2009. *Average of Synthetic Exact Filters*, Proc. CVPR 2009.

Clodic, A., Cao, H., Alili, S., Montreuil, V., Alami, R. and-Chatila, R. *2008. Shary: A Supervision System Adapted to-Human-Robot Interaction*. In Proc. ISER 2008.

Cohn, T. 2007. *Scaling conditional random fields for natural language processing*. University of Melbourne.

Fong T., Kunz C., Hiatt L. and Bugajska M. 2006. *The Human-Robot Interaction Operating System*. Proc. HRI 2006.

Fujimura, K. and Xu, L. 2007. *Sign recognition using constrained optimization*. Proc. ACCV 2007.

Kanda, T., Shiomi M., Miyashita Z., Ishiguro H., and Hagita N. 2009. *An affective guide robot in a shopping mall*. In Proc. HRI 2009

Lafferty J., McCallum A., and Pereira F.. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In Intl. Conf. on Machine Learning, 2001.

Lee, A. and Kawahara, T. 2009. *Recent Development of Open-Source Speech Recognition Engine Julius*. Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2009.

Nakadai, K., Takahashi, T., Okuno, H.G., Nakajima, H., Hasegawa, Y., and Tsujino, H. 2010. *Design and Implementation of Robot Audition System "HARK"*.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T. Leibs, J., Berger, E., Wheeler, R. and Ng, A. 2009. *ROS: an open-source robot operating system*. Proc. Open-source Software Workshop, ICRA 2009.

Raux, A. and Ma, Y. 2011. *Efficient Probabilistic Tracking of User Goal and Dialog History for Spoken Dialog Systems*. Proc. Interspeech 2011.

Rosenthal S., Veloso M. 2010. *Using Symbiotic Relationships with Humans to Help Robots Overcome Limitations*. In Workshop for Collaborative Human/AI Control for Interactive Experiences 2010.

Srinivasa S., Ferguson D., Helfrich C., Berenson D., Collet A., Diankov R., Gallagher G., Hollinger G., Kuffner J., VandeWeghe M. 2009. *Herb: A Home Exploring Robotic Butler*. Autonomous Robots, 2009