Developing Open Source Software for Human Robot Interaction

○Yosuke Matsusaka (AIST)

1. Introduction

Recently, there are increasing demands for the robots which has an ability to communicate with human.

OpenHRI[1] is an open source software components for human robot interaction developed as part of NEDO intelligent robot technology software project. The components wrap Julius speech recognizer[2] for Japanese, English and German voice recognition and OpenJTalk[3], Festival[4] and MARY[5] for Japansese, English and German voice synthesis respectively and also implements auditory filters and extensible dialogue manager components all available for free and open source. Thanks to uniform interface defined in RT-Component specification[6], the developer can easily integrate these components into their complete robotic systems.

In this abstract, the design and implementation as well as the practical development techniques of OpenHRI is introduced.

2. Architecture of OpenHRI

The components are classified into 5 groups. Audio I/O components control audio I/O hardwares and convert the data to/from RT-Component data streams. Audio filter components applies signal processing to improve the clearness of the input signals. Voice recognition and voice synthesis components will convert from/to voice to/from text. Dialog manager components will try to understand each voice input and the other informations to decide how to respond to the user.

The system can be elastically reconfigured to adapt to the environments or to the tasks by cascading or replacing the components.

3. Development Techniques

3.1 Documentation

SPHINX[7] is used for generate documents for OpenHRI components and systems. Half of the documents are written by hand by the developer. The other half of the documents are automatically generated from the texts embedded in the component itself by using the "rtdoc" tool developed by ourselves[8]. Component diagrams used in examples documentations are also automatically generated by "rtstodot" tool combined with GraphViz visualization tool[9].

3.2 Testing

Testing of the components configured in typical system compositions are done automatically by running the UNIX shell scripts. One of the difficulty of testing HRI systems are: we have to make an actual pronunciation to test the voice recognizer and listen to the actual voice to confirm correct output of the voice synthesizer. For this problem, we have enabled automated test by reconstructing the components to form an inverse system that recognize the voice synthesized by the system itself. This technique is especially useful when testing multilingual system without hireing native speakers.

3.3 Building Installer and Package

Building of installer and package is the another important step that should be automated to enable fast iteration of deployment and incorporation of user feedbacks. We use launchpad[10] to automate the creation of Linux binary packages and NSIS script[11] to automate the creation of Windows installer. Update of the user side package has also been done automatically by apt-get system for Linux platform and "rtsetup" utility[12] developed by our selves for Windows platform.

4. Summary

In this abstract, we have briefly introduced practical techniques we have developed to maintain our softwares and documents. We hope these techniques are shared among the community to realize effective development of open source robotic softwares.

References:

[1] http://openhri.net/
[2] http://julius.sourceforge.jp/
[3] http://open-jtalk.sourceforge.net/
[4] http://www.cstr.ed.ac.uk/projects/festival/
[5] http://mary.dfki.de/
[6] http://openrtm.org/
[7] http://sphinx.pocoo.org/
[8] https://github.com/gbiggs/rtshell
[9] http://www.graphviz.org/
[10] https://launchpad.net/
[11] http://nsis.sourceforge.net/
[12] http://code.google.com/p/rtsetup/