Multimodal Dialogue Systems with InproTK$_S$ and Venice

Casey Kennington
CITEC, Dialogue Systems Group, Bielefeld University
ckennington[1]

Spyros Kousidis
Dialogue Systems Group
Bielefeld University
spyros.kousidis[2]

David Schlangen
Dialogue Systems Group
Bielefeld University
david.schlangen[2]

[1]@cit-ec.uni-bielefeld.de
[2]@uni-bielefeld.de

Abstract

We present extensions of the incremental processing toolkit INPROTK which, together with our networking adaptors (*Venice*), make it possible to plug in sensors and to achieve situated, real-time, multimodal dialogue. We also describe a new module which enables the use in INPROTK of the Google Web Speech API, which offers speech recognition with a very large vocabulary and a wide choice of languages. We illustrate the use of these extensions with a real-time multimodal reference resolution demo, which we make freely available, together with the toolkit itself.

1 Introduction

In face-to-face conversation, interlocutors normally do more than just listen to speakers: they also observe what speakers do while they speak, for example how they move and where they look. Sensors that can make such observations are becoming ever cheaper. Integrating (i.e., fusing) the data they provide into the understanding process, however, is still a technical challenge (Atrey et al., 2010; Dumas et al., 2009; Waibel et al., 1996). We illustrate how our InproTK$_S$ suite of tools (Kennington et al., 2014) can make this process easier, by demonstrating how to plug together a little demo tool that combines instantiations for motion capture (via *Leap Motion*,[1]), eye tracking (*eyetribe*[2]) and speech (Google Web Speech).[3]

Furthermore, truly multimodal systems are more feasible today then they were 5 or 10 years ago, due to the proliferation of affordable sensors for common requirements in multimodal processing, such as motion capture, face tracking and eye tracking, among others. However, each sensor is typically constrained to specific platforms and programming language, albeit mostly the most common ones, a fact that hinders integration of such sensors into existing spoken dialogue systems. InproTK$_S$ and our Venice tools are a step towards streamlining this process.

In this paper, we will briefly describe INPROTK and the extensions in InproTK$_S$. We will then describe Venice and give a use case, which we have packaged into a real-time working demo.

2 The IU model, INPROTK

As described in (Baumann and Schlangen, 2012), INPROTK realizes the *IU*-model of incremental processing (Schlangen and Skantze, 2011; Schlangen and Skantze, 2009), where incremental systems consist of a network of processing *modules*. A typical module takes input from its *left buffer*, performs some kind of processing on that data, and places the processed result onto its *right buffer*. The data are packaged as the payload of *incremental units* (IUs) which are passed between modules.

3 Extensions of InproTK$_S$

InproTK$_S$ provides three new methods of getting information into and out of INPROTK:

- *XML-RPC*: *remote procedure call* protocol which uses XML to encode its calls, and HTTP as a transport mechanism.[4]
- *Robotics Service Bus*: (RSB), a message-passing middelware (Wienke and Wrede, 2011).[5]

[1] https://www.leapmotion.com/
[2] https://theeyetribe.com/
[3] We also have instantiations for *Microsoft Kinect* and *Seeingmachines FaceLAB*, www.seeingmachines.com/product/facelab/
[4] http://xmlrpc.scripting.com/spec.html
[5] https://code.cor-lab.de/projects/rsb

- *InstantReality*: a virtual reality framework, used for monitoring and recording data in real-time.[6]

- *Google Web Speech* has also been implemented as a module, in a similar manner to (Henderson, 2014).[7]

The first three methods have implementations of *Listeners* which can receive information on their respective protocols and package that information into IUs used by InproTK$_S$. Each method also has a corresponding *Informer* which can take information from an IU and send it via its protocol. A general example can be found in Figure 1, where information from a motion sensor is sent into InproTK$_S$ (via any of the three methods), which packages the information as an IU and sends it to the NLU module; later processed information is sent to an informer which then sends it along its protocol to an external logger.

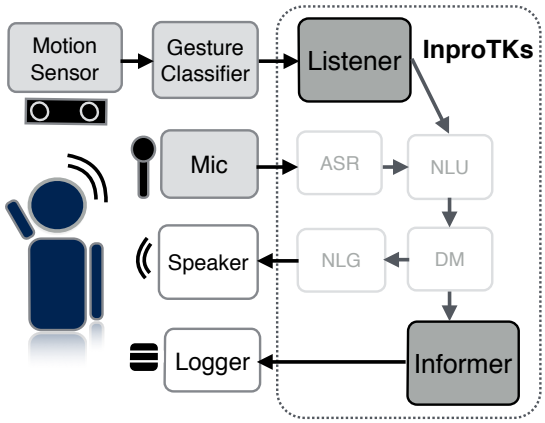

Figure 1: Example architecture using new modules: motion is captured and processed externally and class labels are sent to a listener, which adds them to the IU network. Arrows denote connections from right buffers to left buffers. Information from the DM is sent via an Informer to an external logger. External gray modules denote input, white modules denote output.

4 Bridging components together: Venice

Our venice components allows integration of any sensor software quickly and easily into InproTK$_S$ by using either of the RSB and InstantIO protocols described above as a network bus. *Venice.ipc* is a platform independent service that accepts data on a socket and pushes it to the network bus. Thus, the sensor SDK can be in any language/major OS and still be quickly integrated. *Venice.hub* is a central component that allows IO to/from any of the two protocols and disk, and is thus used for synchronous logging of all the data on the network, as well as replaying and simulating. Any number of components (sources and/or targets) can be added/removed from such a network at runtime. The Listener/Informer components of InproTK$_S$ communicate directly to this network for multimodal data I/O. Components can reside on the same computer or on dedicated workstations in a LAN.

5 Use case: The Multimodal Reference Resolution Demo

Using InproTK$_S$ we have developed a spoken dialogue system that performs online reference resolution in the Pentomino domain using three modalities: speech, gaze and deixis. We use the Leap sensor for motion capture and eyetribe for eye tracking. Both sensors are used by modifying one of their SDK examples with minimal effort, in order to send data to the venice.ipc service running on the machine. The latter sends the data using InstantIO to InproTK$_S$. The application that uses the toolkit has two InstantIO Listeners (one for each modality) and a Listener for the ASR (Google Web Speech). These are effortlessly connected to the main module (that performs the reference resolution) by means of an XML configuration file.

The main module itself performs the fusion by distributing probabilities to different candidate referents based on the input from each modality independently. If data from different modalities point to different candidates, a flat probability distribution occurs, with no candidate significantly more likely to be the referent. If more than one modalities point to the same candidate, then its probability overcomes a threshold and the reference is resolved. The confidence distribution is output by InproTK$_S$ via an Informer module back to the network and is displayed in real-time by a separate component (a Virtual Reality browser).

[6] http://www.instantreality.org/

[7] https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html

References

Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. 2010. *Multimodal fusion for multimedia analysis: a survey*, volume 16. April.

Timo Baumann and David Schlangen. 2012. The InproTK 2012 Release. In *NAACL*.

Bruno Dumas, Denis Lalanne, and Sharon Oviatt. 2009. Multimodal Interfaces : A Survey of Principles , Models and Frameworks. In *Human Machine Interaction*, pages 1–25.

Matthew Henderson. 2014. The webdialog Framework for Spoken Dialog in the Browser. Technical report, Cambridge Engineering Department.

Casey Kennington, Spyros Kousidis, and David Schlangen. 2014. InproTKs: A Toolkit for Incremental Situated Processing. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 84–88, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

David Schlangen and Gabriel Skantze. 2009. A General, Abstract Model of Incremental Dialogue Processing. In *Proceedings of the 10th EACL*, number April, pages 710–718, Athens, Greece. Association for Computational Linguistics.

David Schlangen and Gabriel Skantze. 2011. A General, Abstract Model of Incremental Dialogue Processing. *Dialoge & Discourse*, 2(1):83–111.

Alex Waibel, Minh Tue Vo, Paul Duchnowski, and Stefan Manke. 1996. Multimodal interfaces. *Artificial Intelligence Review*, 10(3-4):299–319.

Johannes Wienke and Sebastian Wrede. 2011. A middleware for collaborative research in experimental robotics. In *System Integration (SII), 2011 IEEE/SICE International Symposium on*, pages 1183–1190.