Component Pluggable Dialogue Framework And Its Application to Social Robots

Ridong Jiang, Yeow Kee Tan, Dilip Kumar Limbu, Tran Anh Dung & Haizhou Li

Institute for Infocomm Research, Singapore 138632

Abstract

This paper is concerned with the design and development of a component pluggable event driven dialogue framework for service robots. We abstract standard dialogue functions and encapsulate them into different types of components or plug-ins. A component can be a hardware device, a software module, an algorithm or a database connection. The framework is empowered by a multi-purpose XML-based dialogue engine which is capable for pipeline information flow construction, event mediation, multi-topic dialogue modelling and different types of knowledge representation. The framework is domain independent, cross-platform and multilingual.

Aim

Provide a configurable, scalable, extensible and component reusable spoken dialogue framework to ultimately reduce the efforts for building up a new spoken dialogue applications.

System Architecture

In order to make the dialogue framework reusable, we employ object oriented approach, loose component coupling, event driven paradigm and plug-and-play strategy to design and develop the proposed dialogue framework. The overall system architecture is shown in Fig. 1.

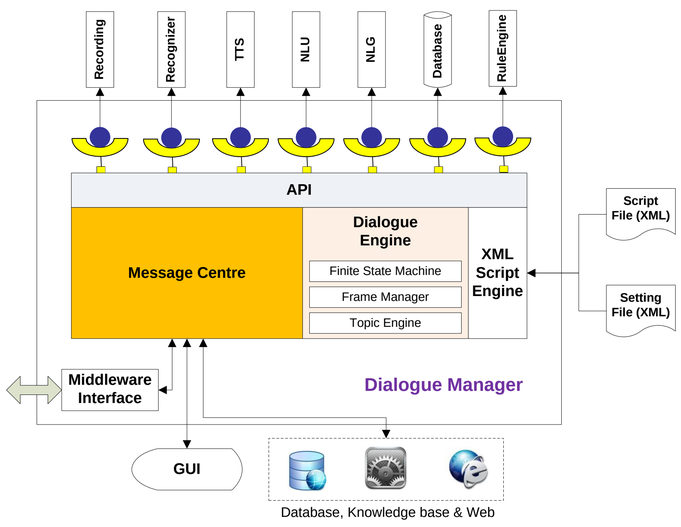

Fig. 1 System architecture of component-based spoken dialogue framework

Interface Design - Generic Component

- **OnMessage –** Handler of all incoming messages
- **FireEvent –** Interface to fire an event by a particular component
- **SendMessage – I**nterface to send a message to remote agents
- **Print –** Show message in main console

API & Interface Hierarchy

The framework provides C++ application programming interface (API) for both Generic Component and Standard Dialogue Component development. With the API provided, new algorithms and dialogue components can be quickly integrated into the framework. Fig.2 illustrates the generic interface and standard dialogue interface as well as their inheritance hierarchy with some of the developed plug-ins.

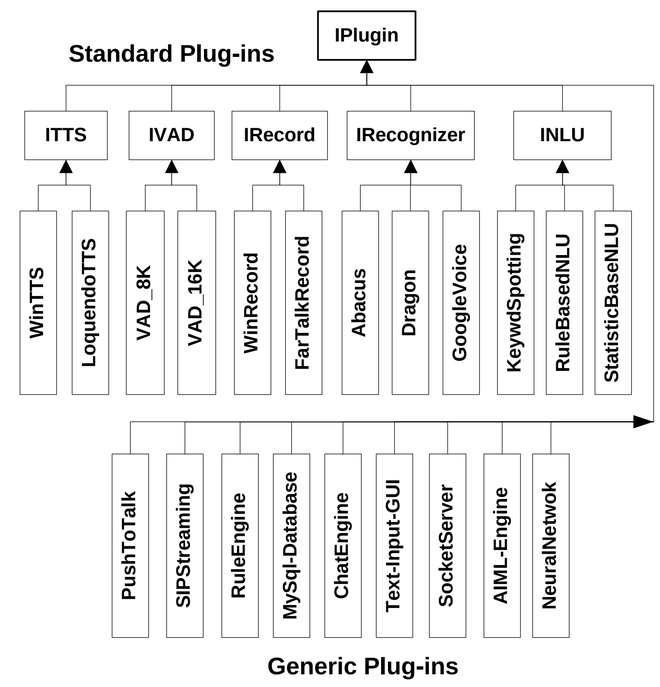

Fig. 2 Object-oriented interface and its hierarchical relationship with plug-ins

Component Management & Communication

1.Component Reuse and Configuration

```
<Plugins>
   <Plugin>SR_WinTTS</Plugin>
   <Plugin>SR_SpeechEnhancement</Plugin>
   <Plugin>SR_Dragon</Plugin>
   <Plugin>SR_StatisticNLU</Plugin>
   <Plugin>SR_NLG</Plugin>
   <Plugin>MySQLDatabase</Plugin>
</Plugins>
```

2.Dynamic loading/unloading

```
<post module="system" command="load"
param="SR_WinTTS"/>
```

3.Multiple-Engine Management

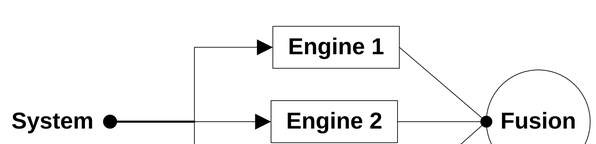

Fig. 3 Diagram of multiple engines (the same type) work in parallel for improved results

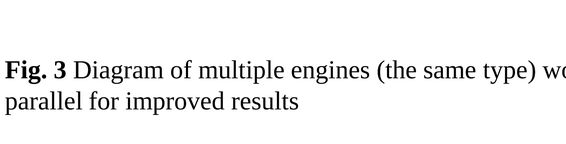

Fig. 4 Diagram of multiple engines (the same type) – dynamically switch based on scenarios

Dialogue Modelling in XML

1.Finite State Machine

2.Frame-based Representation

3.Rule-based Knowledge Representation

4.Hybrid Task Representation

5.XML Procedural Programming

- **Variables**
- **Expression**
- **Function**
- **Control statement: if/else, for loop, etc.**
- **Timer**
- **Network: send message thru TCP/IP**
- **File IO & string handling**
- **Built-in Support of Array**
- **Built-in Support of List**

Conclusion

- Light weight component pluggable spoken dialogue framework
- Configurable architecture
- Component reusable
- Extensible through plug-ins
- Fully driven by XML script in both dialogue logic and low level flow control
- Full-fledge rule engine and MySql database support
- Deployed on a number of Robots & dialogue applications

References

1. MICHAEL F. MCTEAR, "Spoken Dialogue Technology: Enabling the Conversational User Interface", *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 90–169.
2. R. Pieraccini, and J. Huerta, "Where do we go from here? Research and commercial spoken dialog systems." *6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, Portugal, September 2-3, 2005.
3. Galaxy Communicator Documentation: http://communicator.sourceforge.net/sites/MITRE/distributions/GalaxyCommunicator/docs/manual/index.html.
4. Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P. and Zue, V, "Galaxy-II: A Reference Architecture for Conversational System Development," *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP '98)*, pp. 931-934, Sydney, Australia, December, 1998.
5. Dan Bohus, Antoine Raux, Thomas K. Harris, Maxine Eskenazi, Alexander I. Rudnicky, "Olympus: an open-source framework for conversational spoken language interface research." *Proceedings of HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology (2007)*.
6. Levin, E., Narayanan, S., Pieraccini, R., Biatov, K., Bocchieri, E., Di Fabbrizio, G., Eckert, W., Lee, S., Pokrovsky, A., Rahim, M., Ruscitti, P., and Walker, M The AT&T-DARPA "Communicator mixed-initiative spoken dialog system", *Proceedings of the Sixth International Conference on Spoken Language Processing* (ICSLP'2000), Beijing, China, pp. 122-125, 2000.
7. Staffan Larsson, David R Traum, "Information state and dialogue management in the TRINDI dialogue move engine toolkit", *Natural Language Engineering*, Volum 6, Issue 3-4, 323 – 340, 2000.
8. J. Bos, E. Klein, O. Lemon, T. Oka, "DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture", *Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue*, 2003.
9. Peter Ljunglöf, "Trindikit.py: An open-source Python library for developing ISU-based dialogue systems." In *Proceedings IWSDS'09, 1st International Workshop on Spoken Dialogue Systems Technology Workshop*, Kloster Irsee, Germany, 2009.
10. S.W. Hamerich, Y.H. Wang, V. Schubert, V. Schless and S. Igel, "XML-Based Dialogue Descriptions in the GEMINI Project". *Proceedings of the "Berliner XML-Tage 2003*, Germany, pp.404-412.
11. T. Heinroth and D. Denich, "Spoken Interaction within the Computed World: Evaluation of a Multitasking Adaptive Spoken Dialogue System", 35th Annual IEEE International Computer Software and Applications Conference (COMPSAC 2011), IEEE, 2011