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.

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.

Component Management & Configuration
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>
</Plugins>

2. Dynamic loading/unloading
<post module="system" command="load" param="SR_WinTTS"/>

3. Multiple-Engine Management

Dialog 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 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