MMDAGENT — A FULLY OPEN-SOURCE TOOLKIT FOR VOICE INTERACTION SYSTEMS

Akinobu Lee, Keiichiro Oura, and Keiichi Tokuda

Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Japan

ABSTRACT

This paper describes development of an open-source toolkit which makes it possible to explore a vast variety of aspects in speech interactions at spoken dialog systems and speech interfaces. The toolkit tightly incorporates recent speech recognition and synthesis technologies with a 3-D CG rendering module that can manipulates expressive embodied agent characters. The software design and its interfaces are carefully designed to be fully open toolkit. Ongoing demonstration experiments to public indicates that it is promoting related researches and developments of voice interaction systems in various scenes.

Index Terms— Spoken dialog system, voice interaction, open-source, embodied agent

1. INTRODUCTION

Spoken dialog systems (SDSs) and speech interfaces (SIs) are promising as intuitive man-machine interfaces. Many systems and architectures have been studied: to name a few, Galaxy Communicator [1], "How may I help you?" [2], DIPPER [3], the SEMAINE API [4] and "Let's Go" [5]. Also, not a few studies have been carried out for systems with embodied agent characters [6]. Recent speech modeling techniques have enabled wide practical use of spoken language interface such as voice-enabled web search, voice portal, voice control of car navigation systems, etc.

There are many issues to be tackled for practical SDSs: quality of speech recognition and synthesis, dialog strategies and adaptation. Also, in a real-world application, it is required to treat affective aspects such as expressive speech, embodied agents or interface design. Finding a common knowledge or methodology over those issues require a vast and diverse practical trials. Thus a highly flexible and tightly integrated platform which consists of speech processing, dialog management, and agent expressions is required.

Furthermore, to promote both commercial and academic works related to SDS, the tool should be made open. There has been many spoken dialog toolkits in both commercial ones (TellMe, BeVocal) and academic ones (Ariadne [7], SpeechBuilder [8], CSLU toolkit [9] and so on), and there are also open-source SDS toolkits such as Olympus [10] and Galatea [11]. However, the Olympus does not mainly focus on the expressive aspects, and Galatea has only a facial expression module.

We have developed an open toolkit for building various interactive speech systems for exploration of vast variety of aspects in SDSs and SIs. The software is named "MMDAgent" [12]. It incorporates low-latency fast speech recognition, HMM-based flexible speech synthesis, embodied 3-D agent rendering with simulated physics, and dialog management based on a finite state transducer (FST). The system is designed carefully to offer real-time rich interaction, light-weight high frame rate rendering, flexible control of emotional expressions. Open formats are adopted for most files to cope with other software. The system has been made public to actually push forward the field of SDS involving related researches, developments of practical applications.

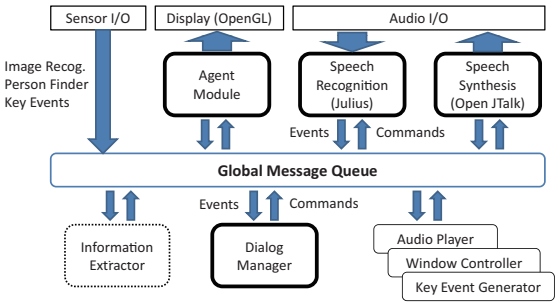

Fig. 1. System architecture

The rest of this paper is organized as follows. The overview of the toolkit and its design to allow for flexible manipulation integrating various modules is described in Section 2. Section 3 explains about compatibility with other tools and license issues. Section 4 summarizes development and demonstration experiments, and Section 5 concludes this paper.

2. SOFTWARE DESIGN FOR FLEXIBILITY

2.1. System architecture

Fig. 1 illustrates the main architecture of "MMDAgent." All the modules run in threads, sharing a global message queue. A module is activated by either I/O events (speech input, sensor signal, etc.) or command messages thrown into the queue (speech synthesis request, 3-D model handling, motion control, etc.). It can also outputs event messages to the queue (speech recognition trigger, end of speech synthesis, etc.). A dialog manager listens for the event messages and sends command messages according to a dialog scenario.

All the modules except the rendering module are implemented as pluggable dynamic libraries, and a module can easily add new events and commands to the system. This simple message-driven architecture provides good modularity and extensibility to the whole system.

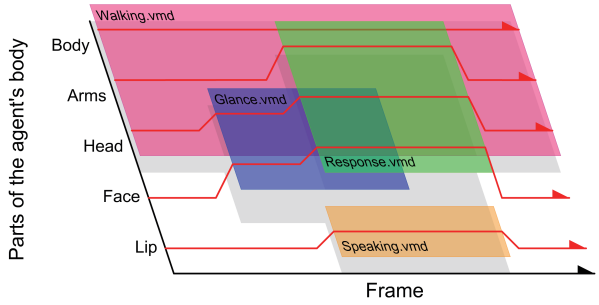

Fig. 2. Multi-track motion and motion smoothing

2.2. 3-D embodied agent

The 3-D graphics rendering and control module for embodied agents has been built on OpenGL from scratch to be tailored for voice interaction systems. Toon rendering is adopted as primary rendering scheme for its expression. Toon rendering is a method to create the look of 2D cel or cartoon animation using 3D modelings. Compared to a photo-realistic representation, the toon expression has an advantage that its abstracted expression lets people feel easy without degrading speech intelligibility [13]. Also the required rendering cost is relatively low as compared with photo-realistic rendering.

Modern 3-D object structure and movement manipulation are implemented for rich agent expression: bones, vertex morphs, inverse kinematics, skinning, alpha texture, environmental mapping, depth shadowing, and so on. Physics simulation is also incorporated to apply natural and realistic movements for objects. "MMDAgent" can manipulate the objects via pre-defined set of motions, or motion fragments as shown in the next section. It can also control a bone or a morph in a model directly by a program, which can be used for eye gazing etc.

The whole rendering process is performed on real-time: they can be rendered at very high frame rate on most PCs, and can run even on handheld devices.

2.3. Motion manipulation

To achieve flexible and interactive movement, an on-line motion composition method using motion fragments is implemented. By handling each part of a model independently and asynchronously in correspondence with user's various acts, it would be possible to realize a complex, mixed-style interaction. The motion of an agent is treated as a composition of multiple motion fragment, where each fragment corresponds to the partial movement of a model. When the motions are overlapped at a part, the last one takes the effect, while the other motions are still kept updating. A simple example in Fig. 2 shows how the composition of multiple motion fragments is executed at a situation that a walking agent is called by a user. Each motion controls only the concerning part of body: "Walking" for body, arms and head, "Glance" for head and face, "Response" for arms, head and face, and "Speaking" for lip synchronization. They are executed concurrently with a different timing. Furthermore, motion smoothing and blending are implemented to soften discontinuities and lags at the beginning and end of the motion fragment.

This motion composition enables users to build a rich, highly interactive and non-lagged continuous response.

2.4. Speech recognition

For the speech recognition engine, a general-purpose, open-source large-vocabulary continuous speech recognition engine Julius is used [14]. Julius provides fast, real-time speech decoding using a 2-pass tree-trellis algorithm with a small memory footprint. It is light-weight (requires less than two seconds to start up for 60k-word dictation on most PCs). It is equipped with functions such as low-latency audio processing, spectral subtraction, cepstral mean / variance normalization, GMM-based voice activity detection, grammar-based recognition, class N-gram, multi-model decoding, confusion network output, and so on.

Julius can handle a variety of acoustic and language models: HTK acoustic models up to triphone with arbitrary state transition, and ARPA word N-grams with arbitrary length are supported. A general Japanese acoustic / language models and dictionary are included in the package. It also supports run-time dictionary switching which can be controlled from from dialog manager.

2.5. Speech synthesis

On emotional and expressive representation in SDS, a dynamic and diverse control of the representation in a smooth and continuous way is required. Therefore, a hidden Markov model (HMM) based speech synthesis approach [15] was adopted for synthesizing speech in various styles. The HMM-based framework makes it possible to model different voice characteristics, speaking styles, or emotions without recording large speech databases. Adaptation [16], interpolation [17], and eigenvoice techniques [18] can be used to modify voice characteristics.

In this research, we apply the method of speaker adaptive training [19] to "speaking style" adaptive training. From the utterances with various speaking styles by one speaker, a model of average speaking style model is robustly estimated together with each speaking styles. Then speech with various speaking styles can be synthesized dynamically by interpolating the models between multiple speaking styles.

As speech database for the default voice, a total of 848 sentences was uttered by one female speaker, including 503 phoneme balanced sentences, 215 sentences in ordinary speaking style, and about 75 sentences for other four speaking styles (angry, bashful, happy, sad). The acoustic models are trained by the HMM-based speech synthesis system (HTS) [20] and distributed with the package of the toolkit.

The Japanese TTS system Open JTalk was adopted as the TTS system. One can express detailed emotions in the synthesized speech by using multiple acoustic models and presetting their parameters (interpolation weights, speaking speed, volume, average F0, etc.). Furthermore, it is possible to simultaneously synthesize voices for multiple characters in different speaking styles by using multiple acoustic models.

The lip motion is generated internally in real time, based on the phoneme durations used at synthesis and phoneme-to-morph mapping rule, so complete synchronization of mouth movement and synthesized speech is guaranteed.

2.6. Event-driven operation and dialog management

User's action and system output does not always happens in a sequencial manner. Thus, all modules should be run asynchronously to realize live interactions. "MMDAgent" adopts an event-driven architecture, in which all modules share a global event and command queue. Fig. 3 illustrates the time course of event queue and relevant dialog management for simple response to a user's utterance.

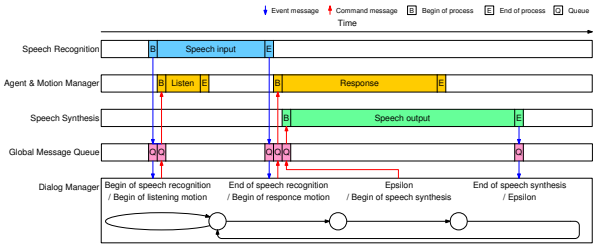

Fig. 3. Event-driven dialog management

A dialog/interaction management module will catch the events, and issue the corresponding commands to the queue. A finite state transducer (FST) based dialog/interaction discourse management is implemented as the default method. It accepts event messages as input and output command message based on the scenario.

3. SOFTWARE DESIGN FOR OPENNESS

3.1. Compatibility with other software

All the task-dependent part of SDSs and SIs from speech processing to agent expression (i.e., acoustic and language models for recognition and synthesis, 3-D object, motion, dialog script, and so on) shoule be made open. It requires a clear modulation and simple interface definitions between the modules, and a clear scheme that bridges speech activities and agent expressions is needed.

The 3-D object rendering part was made to have full compatibility with a CG software "MikuMikuDance," a free 3-D authoring tool for creating music promotion videos with animated 3-D characters. It treats 3-D model definition (.pmd) and a set of bone/morph motion (.vmd) in a separate file, assuming common bone/morph name set among them. One can design static expressions (3-D representations) and the dynamic expressions (their movements) of a 3-D object for SDSs snd SIs with the software. Also, the software already owns an active consumer generated media (CGM) scenes with thousands of users and they share models and motions to mash up their work. We expect that such CGM scene could also encourage the development of SDSs by the casual users.

Object models and motions built by several popular software such as Blender and 3ds Max can be converted to the software.

For speech recognition, Julius can handle almost any acoustic and language models in HTK and ARPA format. One can set up his own speech recognizer customized for a specific task, or can add a user dictionary for each dialog task easily.

For speech synthesis, any models trained by HTS can be used. On-line speaker interpolation is supported, so one can merge several speaker model at run time to produce speech of various speaking styles.

3.2. License

"MMDAgent" was developed as open-source free software under the modified BSD license that permits commercial use. The libraries, speech processing engines, dictionaries, and other components used in "MMDAgent" are all released under the modified BSD licenses or equivalents, or under liberal unified licenses. "MMDAgent" separates the task-dependent components from the basic part of the system. An sample set of the task-dependent components are released under Creative Commons licenses [21] because the modified BSD license is designed for only software.

Fig. 4. Exhibition of SDS system based on MMDAgent with life-sized agent at CEATEC 2010.

Table 1 shows licenses of all components used in "MMDAgent." We expect that the openness allowed by this license encourages progress in research related to spoken dialog systems in technical fields, promote the use of speech interfaces in industry, and facilitate the use of spoken dialog systems.

4. DEVELOPMENT AND DEMONSTRATION EXPERIMENT

All modules are developed from scratch and built into a single package. The main development environment is Windows, and it also runs under Mac OS and Linux. At first, it was demonstrated at CEATEC JAPAN 2010, Japan's largest IT and electronics exhibition conference, as a speech-interactive guidance system with life-sized agent on a 65-inch display device (Fig.4). The first stable version was released on May 2011. The current version is 1.3.

It has been downloaded 44,577 times since its first release. Fig. 5 shows the change of download statistics per country. The access from outside of Japan was only 12% for the first month, however the

Table 1. Licenses of all components in "MMDAgent."

Module	Component	License type
3-D rendering	BulletPhysics	zlib license
	GLee	BSD license
	GLFW	zlib license
	JPEG	BSD-style license
	libpng	zlib license
	zlib	zlib license
Speech recognition	CSRC Dictionary	BSD-style license
	Julius	BSD-style license
	PortAudio	BSD-style license
Speech synthesis	hts_engine API	BSD license
	MeCab	BSD license
	Naist Japanese Dictionary	BSD license
	Open JTalk	BSD license

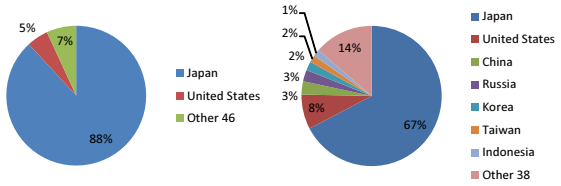

Fig. 5. Download statistics of the first month (2010/12) and recent month (2012/11)

rate growed up to 33% at 2012/11.

The unique openness and modularity, tightly combining recent up-to-date speech technologies with design works of 3-D models and character expressions, begins to motivate various people to engage in spoken dialog systems and multi-modal interaction systems [22][23]. Also, user-originated works using "MMDAgent" have arisen. To name a few:

- Speech-oriented digital signage for campus guidance [24]
- Port to Android and iPhone as applications
- Ubuntu package version [25]

5. CONCLUSION

An open-source toolkit that tightly integrates components from speech processing to 3-D rendering is developed. All the system components from acoustic model to 3-D embodied agent can be easily replaced to allow for people to explore all the aspects of practical spoken dialog systems and speech interfaces. The ongoing demonstration experiment suggests that it encourages broad range of developments of voice interaction systems in various scenes.

We hope that this system promotes various related fields: for academic, SDS and SI, spoken language processing, artificial intelligence. The systems are independent from the dialog contents: acoustic models for recognition and synthesis, dialog scenario, 3-D models and motions. We hope this toolkit invigorates researches related to voice interactions.

ACKNOWLEDGEMENT The research leading to the work was partly funded by the Core Research for Evolutional Science and Technology (CREST) from Japan Science and Technology Agency (JST).

6. REFERENCES

[1] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "Galaxy-II: A reference architecture for conversational system development," in *Proc. ICSLP*, 1998, pp. 931–934.

[2] A. Gorin, G. Riccardi, and J. Wright, "How may I help you?" *Speech Communication*, vol. 23, no. 1-2, pp. 113–127, 1997.

[3] J. Bos, E. Klein, O. Lemon, and T. Oka, "DIPPER: Description and formalisation of an information-state update dialogue system architecture," in *Proc. 4th SIGdial Workshop on Discourse and Dialogue*, 2003, pp. 115–124.

[4] M. Schröder, "The SEMAINE API: Towards a standards-based framework for building emotion-oriented systems," *Proc. Advances in Human-Computer Interaction*, vol. 2010, Article ID 319406, 2010.

[5] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi, "Let's go public! taking a spoken dialog system to the real world," in *Proc. Interspeech 2005*, 2005.

[6] E. Andre and C. Pelachaud, "Interacting with embodied conversational agents," in *Speech Technology: Theory and Applications*. Springer, 2010, ch. 8.

[7] Ariadne spoken dialog system. [Online]. Available: http://www.opendialog.org/

[8] J. Glass, E. Weinstein, S. Cyphers, J. Polifroni, G. Chung, and M. Nakano, "A framework for developing conversational user interfaces," in *Computer-Aided Design of User Interfaces IV*. Springer, 2005, pp. 349–360.

[9] S. Sutton, R. Cole, J. D. Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y. Yan, E. Kaiser, B. Rundle, K. Shobaki, P. Hosom, A. Kain, Johan, J. Wouters, D. Massaro, and M. Cohen, "Universal speech tools: The CSLU toolkit," in *Proc. ICSLP*, 1998, pp. 3221–3224.

[10] D. Bohus, A. Raux, T. K. Harris, M. Eskenazi, and A. I. Rudnicky, "Olympus: an open-source framework for conversational spoken language interface research," in *Proc. NAACL-HLT-Dialog '07*, 2007, pp. 32–39.

[11] K. Katsurada, A. Lee, T. Kawahara, T. Yotsukura, S. Morishima, T. Nishimoto, Y. Yamashita, and T. Nitta, "Development of a toolkit for spoken dialog systems with an anthropomorphic agent: Galatea," *Proc. APSIPA*, pp. 148–153, 2009.

[12] MMDAgent. [Online]. Available: http://www.mmdagent.jp/

[13] J. Beskow, "Animation of talking agents," in *Proc. AVSP'97*, 1997, pp. 149–152.

[14] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine julius," *Proc. APSIPA*, pp. 131–137, 2009.

[15] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," *Proc. Eurospeech-99*, vol. 1, pp. 2374–2350, 1999.

[16] J. Yamagishi, "Average-voice-based speech synthesis," Ph.D. dissertation, Tokyo Institute of Technology, 2006.

[17] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," *Proc. Eurospeech-97*, vol. 1, pp. 2523–2526, 1997.

[18] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," *Proc. ICSLP*, vol. 1, pp. 1269–1272, 2002.

[19] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE Transaction Information and Systems*, vol. E90-D, no. 2, pp. 533–543, 2007.

[20] H. Zen, K. Oura, T. Nose, J. Yamagishi, S. Sako, T. Toda, T. Masuko, A. W. Black, and K. Tokuda, "Recent development of the HMM-based speech synthesis system (HTS)," *Proc. APSIPA*, vol. 1, pp. 121–130, 2009.

[21] Creative Commons. [Online]. Available: http://creativecommons.org/

[22] M. Araki, "Multimodal motion learning system for traditional arts," in *Advances in Affective and Pleasurable Design*. CRC Press, 2012, ch. 15.

[23] T. Ogawa and Y. Kambayashi, "Physical instructional support system using virtual avatars," in *Proc. ACHI 2012, The Fifth Intl. Conf. on Advances in Computer-Human Interactions*, 2012, pp. 262–265.

[24] [Online]. Available: http://www.youtube.com/watch?v=B62_95BAkNQ

[25] PPA for MMDAgent. [Online]. Available: https://launchpad.net/ irie/+archive/mmdagent