Open-source Software for Developing Anthropomorphic Spoken Dialog Agents

Shin-ichi Kawamoto[*1] Hiroshi Shimodaira[*1] Tsuneo Nitta[*3] Takuya Nishimoto[*4]
Satoshi Nakamura[*5] Katsunobu Itou[*6] Shigeo Morishima[*7] Tatsuo Yotsukura[*7]
Atsuhiko Kai[*8] Akinobu Lee[*9] Yoichi Yamashita[*10] Takao Kobayashi[*11]
Keiichi Tokuda[*12] Keikichi Hirose[*2] Nobuaki Minematsu[*2] Atsushi Yamada[*13]
Yasuharu Den[*14] Takehito Utsuro[*3] Shigeki Sagayama[*2]

∗1 Japan Advanced Institute of Science and Technology, ∗2 The University of Tokyo, ∗3 Toyohashi University of Technology, ∗4 Kyoto Institute of Technology, ∗5 Advanced Telecommunications Research Institute International, ∗6 National Institute of Advanced Industrial Science and Technology, ∗7 Seikei University, ∗8 Shizuoka University, ∗9 Nara Institute of Science and Technology, ∗10 Ritsumeikan University, ∗11 Tokyo Institute of Technology, ∗12 Nagoya Institute of Technology, ∗13 The Advanced Software Technology and Mechatronics Research Institute of Kyoto, ∗14 Chiba University

Abstract

An architecture for highly-interactive human-like spoken-dialog agent is discussed in this paper. In order to easily integrate the modules of different characteristics including speech recognizer, speech synthesizer, facial-image synthesizer and dialog controller, each module is modeled as a virtual machine that has a simple common interface and is connected to each other through a broker (communication manager). The agent system under development is supported by the IPA and it will be publicly available as a software toolkit this year.

1. Introduction

Anthropomorphic spoken dialog agent (ASDA), behaving like humans with facial animation and gesture, and making speech conversations with humans, is one of the next-generation human-interface. Although a number of ASDA systems (Gustafson et al., 1999; Julia and Cheyer, 1999; Dohi and Ishizuka, 1997; Ushida et al., 1998; Sakamoto et al., 1997; Cassell et al., 1999) have been developed, communication between the ASDA system and humans is far from being natural, and developing high quality ASDA system is still challenging. In order to activate and progress the researches in this field, we believe that easy-to-use, easy-to-customize, and free software toolkit for building ASDA systems is indispensable.

We have been developing such an ASDA software toolkit since 2000, aiming to provide a platform to build next generation ASDA systems. The features of the toolkit are as follows: (1) basic functions to achieve incremental (on-the-fly) speech recognition, (2) mechanism for "lip synchronization"; synchronization between audio speech and lip image motion, (3) high customizability in text-to-speech synthesis, realistic face animation synthesis, and speech recognition, (4) "virtual machine" architecture to achieve transparency in module to module communication.

If compared to the related works such as CSLU toolkit (Sutton and Cole, 1998) and DARPA Communicator Program (DARPA, 1998), our toolkit is still germinal. However, it is compact, simple, easy-to-understand and thus suitable for developing ASDA systems for research purposes. At present, simple ASDA systems have been successfully build with the toolkit under UNIX/Linux operating systems, and the subset of the toolkit will be publicly available in the middle of the year 2002.

This paper is divided into six sections. Requirements for the ASDA software toolkit are discussed in section 2 followed by the discussion of system design in section 3. Implementation issue and evaluation are described in section 4. Finally the last section is devoted to conclusions.

2. Requirements for the toolkit

In this section, we discuss the requirements for the software toolkit to build ASDA systems which speak, listen, and behave like humans.

2.1. Key techniques for achieving natural spoken dialog

If compared to the keyboard-based conversation, typical phenomena are observed in speech-based conversation. These include the case that human listeners nod their heads or say "yes"during a conversation, and the case that the speakers control the prosody to indicate types of utterances such as questions, statements, and emotions. We regard it important for the toolkit to be a platform for human-like speech-based conversation for providing basic functions to achieve those phenomena.

In addition, speed, quality and balance are also important factors for the toolkit. For example, if the system fails to respond to the user quickly, it loses the naturalness and efficiency of conversation. If the agent's face, voice and behavior are artificial and far from natural, or if the agent looks very similar to humans apart from the point that the voice is synthesized, then the users feel something strange and it prevents them to communicate with the system naturally.

2.2. Configuration for the easy-to-customize

As a common basic toolkit for research and development, the toolkit should not be designed for a specific purpose, but it should be used for multi-purpose. The agent's face, voice, and tasks must be customizable so that the users

This work is supported by IPA (Information-technology Promotion Agency, Japan).

of the toolkit can customize the agents easily depending on the purposes and applications. The customizability includes that the agent characters should be replaced easily by changing the face and voice of a person to those of an another person.

2.3. Modularity of functional units

In some situations, system creators or toolkit users will not be satisfied with the performance of the original modules in the toolkit and they would like to replace them with the new ones or add new ones to the system. In such cases, it would be desired that each functional unit (module) is well modularized so that the users can develop, improve, debug and use each unit independently from the other modules. This would help to improve the efficiency of software development.

Moreover, modularizing the functional units enables the system to work in parallel,

2.4. Open-source free software

The technology used for creating the toolkit is still not enough to achieve human-like conversation. Therefore it is desired that not only the creators of the toolkit but also the researchers and developers who use the toolkit would contribute to improve the toolkit further. In that sense, the toolkit should be released as a free software along with the program source codes.

There have been no existing ASDA softwares so far satisfying all of the requirements described above.

3. Toolkit design and outline

In this section, we discuss the design of the toolkit and its module functionality to achieve the requirements given in the previous section.

First of all, to fulfill the requirements of modularity and customizability, the toolkit must have at least three functional units (speech recognition, speech synthesis, and facial animation synthesis) for task customization, and a unit for integrating those units, which we name as “agent manager”.

3.1. Speech recognition module (SRM)

The authors have been developing the Japanese large vocabulary continuous speech recognition (LVCSR) engines, Julius (Kawahara et al., 1998; Lee et al., 2001) and SPOJUS (Nakagawa and Kai, 1994). Julius employs N-grams as a statistical language model (LM), though, as a toolkit for various tasks, grammar-based LM is suitable for small tasks, where easy-to-use and easy-to-customize LMs are preferable. In order to provide such a grammar-based recognition engine as a functional module of the toolkit, “Julian” (Fig. 2) has been developed. Julian can change more than one grammar sets on the instant, and it can output incremental speech recognition results.

3.2. Speech synthesis module (SSM)

To achieve customizable speech synthesis module (SSM), the module has to accept arbitrary Japanese texts including both of “Kanji” (Chinese) and “Kana” characters, and synthesize speech with a human voice clearly in a specified style. For this purpose, HMM-based speech synthesis method is employed in which spectrum, pitch and duration are modeled simultaneously in a unified framework of HMM (Yoshimura et al., 1999). Lexical and syntactic analyzer is developed as well.

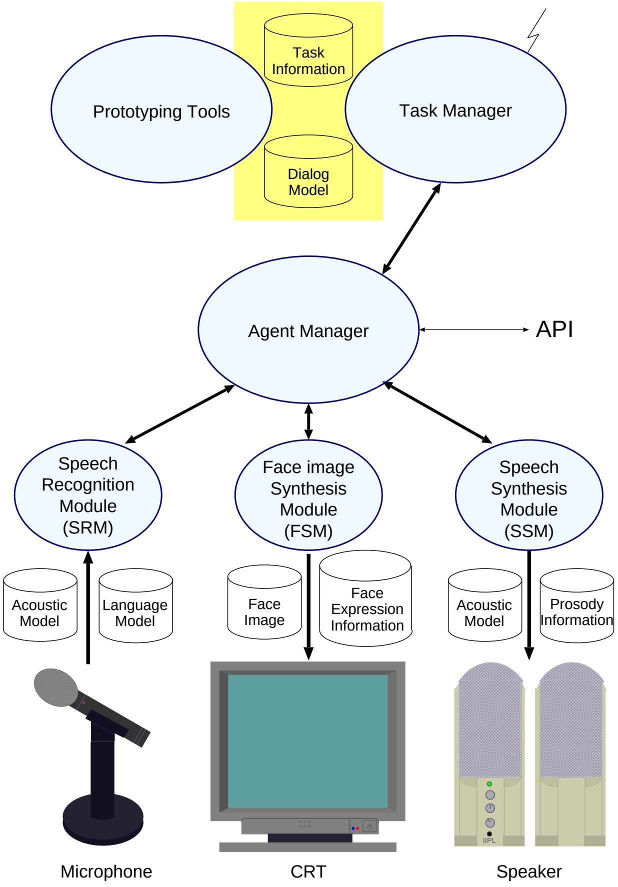


Figure 1: ASDA platform

Another important function of this module is to implement a mechanism for synchronizing the lip movement with speech, which is called “lip-sync”. The employed mechanism is based on the sharing of each timing and duration information of phoneme in the speech that is going to be uttered between the SSM and the FSM (facial image synthesis module).

3.3. Facial image synthesis module (FSM)

The basic software of synthesizing human facial images can synthesize human facial animations of any existing person if a single photo image of the person is given and the image is fitted manually to a standard 3D wire-frame model (Morishima et al., 1995). The software including a model fitting tool is publicly available as a result of the former IPA project (facetool, 1998). Under the current ASDA toolkit project, we are enhancing the former software package to support higher quality and controllability of agent facial image, and precise lip-sync with synthetic speech. Fig. 3 shows the process of fitting a 3D wire-frame model to a real human face, and the examples of the synthesized human facial images.

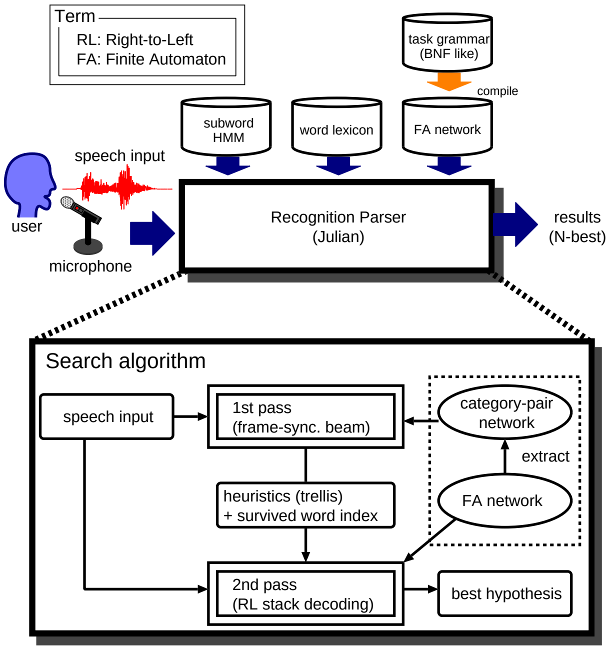


Figure 2: Speech recognition module

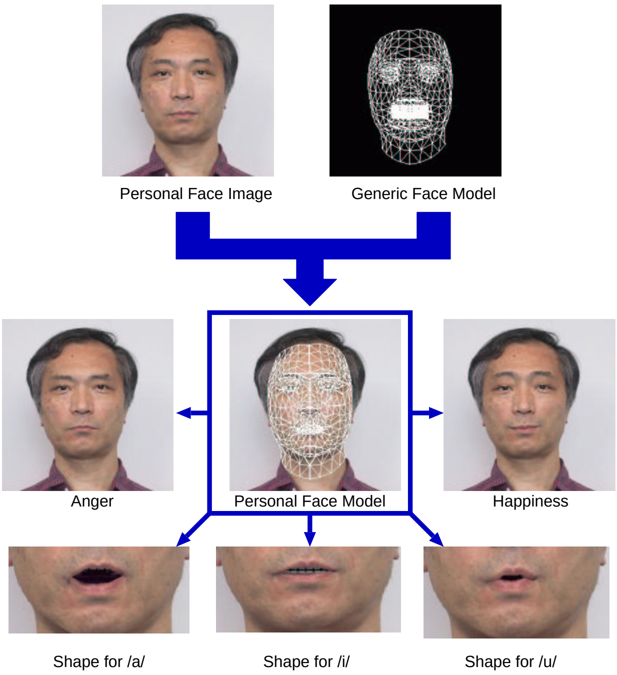


Figure 3: Facial image synthesis module

3.4. Module integration and customization tools

3.4.1. Agent manager

The Agent Manager (AM) serves as an integrator of all the modules of the ASDA system. One of its main functions is to play a central role of communication where every message from a module is sent to another module with the help of the AM. Here, the AM works like a hub in the Galaxy-II system (Seneff et al., 1998). Another essential function of the AM is to work as a synchronization manager between speech synthesis and facial image animation to achieve the precise lip-sync.

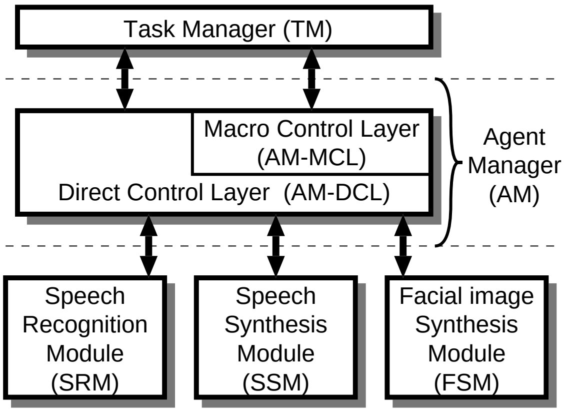

Task Manager (TM)
Macro Control Layer (AM-MCL)
Direct Control Layer (AM-DCL)
Agent Manager (AM)
Speech Recognition Module (SRM)
Speech Synthesis Module (SSM)
Facial image Synthesis Module (FSM)

Figure 4: Basic configuration of the AM and Modules

The AM consists of two functional layers: the Direct Control Layer (AM-DCL) and the Macro Control Layer (AM-MCL). Fig. 4 illustrates a schematic representation of the relationship between the AM and the various modules. The AM-DCL works as a dispatcher receiving commands from a module and forwarding them to the designated module. On the other hand the AM-MCL is a macro-command interpreter processing the macro commands mainly issued by the Task Manager (TM). There are mainly two functions for the AM-MCL. The first one is to simply expand each received macro-command in a sequence of commands and send them sequentially to the designated module. The second function is to process macro-commands that require more complicated processing than just expanding the commands. This happens in the case where more than one modules are involved. Currently, the lip synchronization process is realized by a macro command and an example is given in section 4.

3.4.2. Virtual Machine model

As is previously described, the AM works as a hub through which every module communicates with each other. It is desired that every module has a common communication interface so that the AM can make connection with each module regardless of the interface used in the module. Furthermore, having a common interface reduces the effort of understanding and developing module dependent interfaces. For this purpose a virtual machine (VM) model is employed, where module interface is modeled as a machine with slots, each of which has a value and attribute controlled by a common command set. Each slot can be regarded as a switch or dial to control the operation or a meter to indicate machine status. Fig. 5 illustrates the communication between the AM and a virtual machine model. Changing the slot values by a command corresponds to check or control the running status of the module or the function. For example, following command to the speech synthesis module means starting voice synthesis of a given text right now.

```
set Speak = Now
```

3.4.3. Task manager (TM)

As a software toolkit, it is crucial to define the complexity of tasks that the software can deal with. In that sense, VoiceXML (VoiceXML, 2000) has been employed

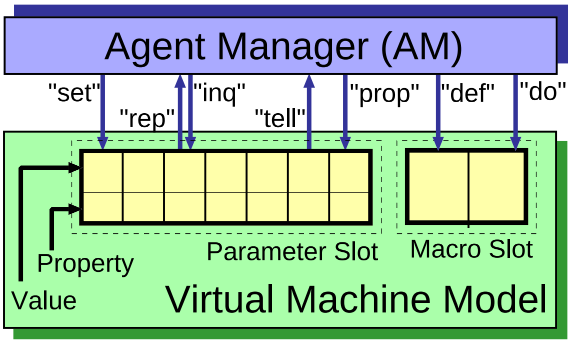
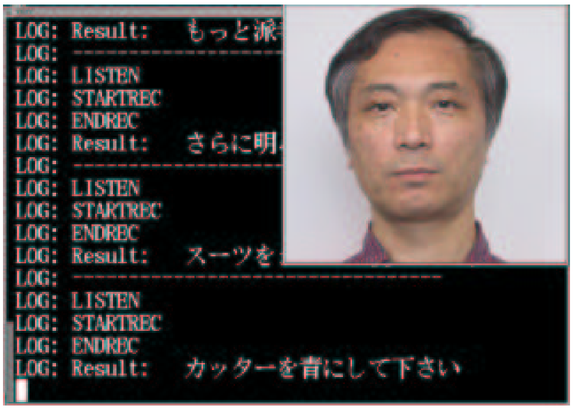


Figure 5: Relationship between the AM and a virtual machine model



```
LOG: Result:   もっと派
LOG: ------------------------
LOG: LISTEN
LOG: STARTREC
LOG: ENDREC
LOG: Result:   さらに明
LOG: ------------------------
LOG: LISTEN
LOG: STARTREC
LOG: ENDREC
LOG: Result:   スーツを
LOG: ----------------------------------------
LOG: LISTEN
LOG: STARTREC
LOG: ENDREC
LOG: Result:   カッターを背にして下さい
```

Figure 6: Screenshot of ASDA

as a basic description language to describe the tasks. Since VoiceXML is originally designed for voice communication over the telephone, and difficulties arise when it is applied to other applications such as anthropomorphic dialogue agents, extensions to its specification are being made in this project. For example, the original specification of VoiceXML does not include any functions to control facial expressions of anthropomorphic dialogue agents.

3.4.4. Prototyping tools

For achieving an easy to customize toolkit, we have a plan to provide prototyping tools. These tools manage some agent customization features. For example, dialog scenario, and related parameters.

4. Experimental Systems

Using the software toolkit, we have built several experimental ASDA systems to evaluate the toolkit. A screenshot of the system and an example of a user-system interaction are shown in Fig. 6 and Fig. 7 respectively.

All the tasks employed were very basic, small vocabulary where the number of uttered word is less than 100 and the perplexity is less than 10. The tasks include (1) echo-back task which repeats what it heard using speech recognition and synthesis, (2) simple appoint-arranging task which changes facial expressions as the conversation goes on, (3) fresh food ordering task that takes orders from customers and responses with “yes” and nodding on the fly.

Those systems consist of the SRM (Kawahara et al., 1998), the SSM (Yoshimura et al., 1999), the FSM (Morishima, 2001), the AM, and a simple task-specific TM which was programmed directly with the command set of the toolkit. We implemented the systems on several platforms with different configurations. Fig. 8 shows the hardware configurations. Some of the demonstration movies (in Japanese, unfortunately) are available in our web site (`http://iipl.jaist.ac.jp/IPA/`).

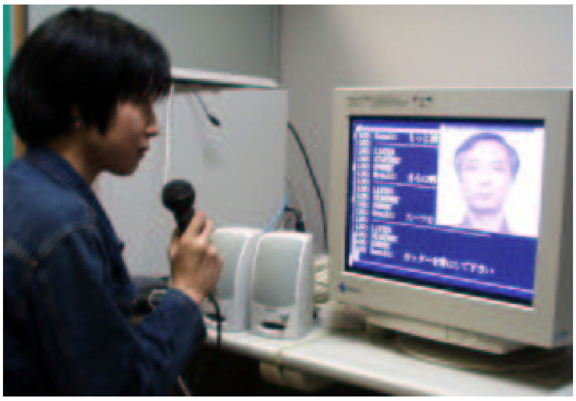
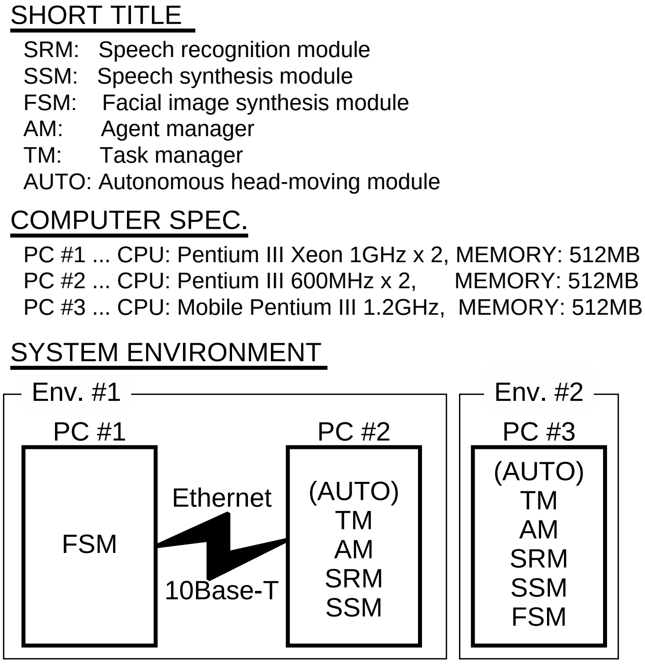


Figure 7: An example of user-system interaction



SHORT TITLE

SRM: Speech recognition module
SSM: Speech synthesis module
FSM: Facial image synthesis module
AM: Agent manager
TM: Task manager
AUTO: Autonomous head-moving module

COMPUTER SPEC.

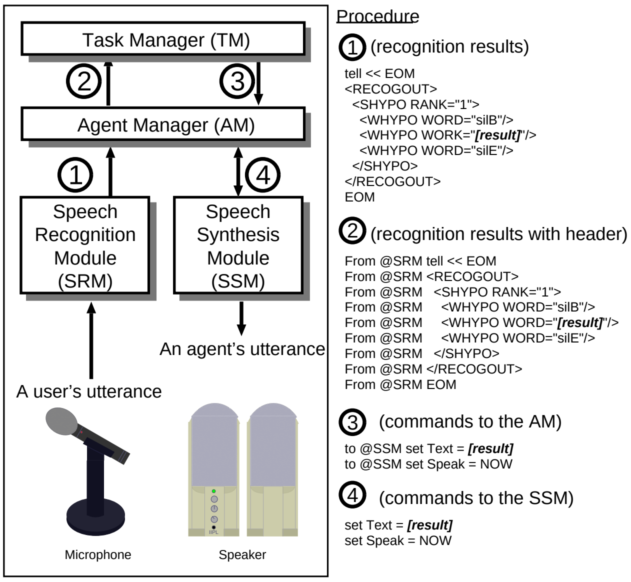
PC #1 ... CPU: Pentium III Xeon 1GHz x 2, MEMORY: 512MB
PC #2 ... CPU: Pentium III 600MHz x 2, MEMORY: 512MB
PC #3 ... CPU: Mobile Pentium III 1.2GHz, MEMORY: 512MB

SYSTEM ENVIRONMENT

Figure 8: Hardware configuration of the ASDA

Fig. 9 shows an example of how the AM and related modules work in the echo-back task. However, the FSM and lip-synchronization mechanism have been omitted in the figure for brevity. Here, the macro commands, which is introduced in 3.4.1., are used in the procedure 3 and 4 to achieve lip-synchronization between the speech and animation. Fig. 10 shows the sequence of commands involved in this lip-synchronization process.

Note that the modules operate in parallel and thus the speech recognition process is active while the agent is speaking. As a result, we confirmed that the system responded to the users quickly, at the same time face animation and synthesized voice were synchronized. However, in this case, we assumed that the ideal environment that the results of speech recognition are not influenced by the output



Procedure

① (recognition results)

```
tell << EOM
<RECOGOUT>
  <SHYPO RANK="1">
    <WHYPO WORD="silB"/>
    <WHYPO WORK="[result]"/>
    <WHYPO WORD="silE"/>
  </SHYPO>
</RECOGOUT>
EOM
```

② (recognition results with header)

```
From @SRM tell << EOM
From @SRM <RECOGOUT>
From @SRM   <SHYPO RANK="1">
From @SRM     <WHYPO WORD="silB"/>
From @SRM     <WHYPO WORD="[result]"/>
From @SRM     <WHYPO WORD="silE"/>
From @SRM   </SHYPO>
From @SRM </RECOGOUT>
From @SRM EOM
```

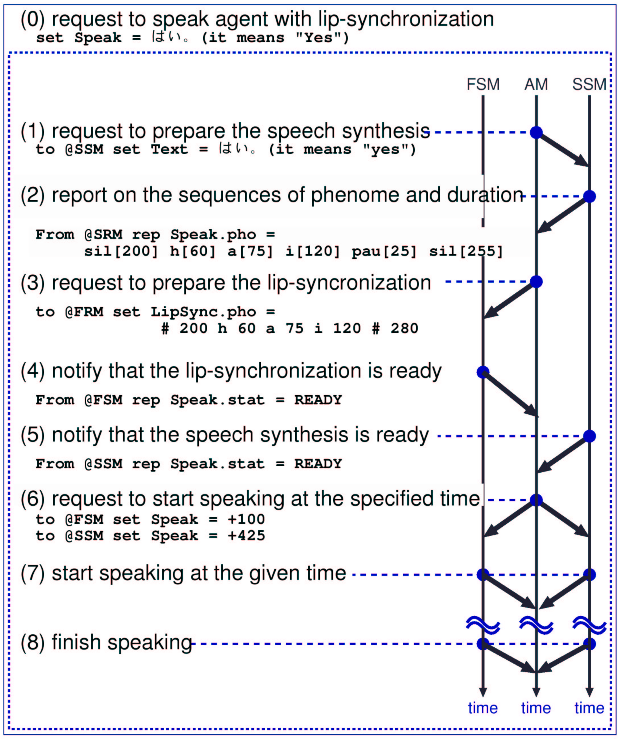
③ (commands to the AM)

```
to @SSM set Text = [result]
to @SSM set Speak = NOW
```

④ (commands to the SSM)

```
set Text = [result]
set Speak = NOW
```

Figure 9: An example of echo-back processing task



(0) request to speak agent with lip-synchronization

```
set Speak = はい。 (it means "Yes")
```

(1) request to prepare the speech synthesis

```
to @SSM set Text = はい。 (it means "yes")
```

(2) report on the sequences of phenome and duration

```
From @SRM rep Speak.pho =
      sil[200] h[60] a[75] i[120] pau[25] sil[255]
```

(3) request to prepare the lip-syncronization

```
to @FRM set LipSync.pho =
                # 200 h 60 a 75 i 120 # 280
```

(4) notify that the lip-synchronization is ready

```
From @FSM rep Speak.stat = READY
```

(5) notify that the speech synthesis is ready

```
From @SSM rep Speak.stat = READY
```

(6) request to start speaking at the specified time

```
to @FSM set Speak = +100
to @SSM set Speak = +425
```

(7) start speaking at the given time

(8) finish speaking

Figure 10: Processing flow among the AM, the SSM, and the FSM when agent speaks (an example of processing in the AM)

of speech synthesis.

5. Discussion

This section describes the current developing status of the software toolkit and discusses further improvement.

5.1. Customization features

In SRM, multi-grammar supporting has been realized where grammars can be changed instantly, and those grammars are easy to customize by means of a supporting software tool.

The SSM can synthesize speech from arbitrary text sentences of mixed kanji and kana (Chinese characters and phonetic script), with customizable prosody. Though speaker adaptation has not been implemented, the employed HMM-based approach is promising in case of speaker adaptation (Tamura et al., 2001b; Tamura et al., 2001a).

The FSM synthesizes 3D realistic facial animations from a single snapshot of a person's face by fitting a wire-frame model to a 2D picture. A software tool is provided to help fitting a standard wire-frame model to the input picture, whose manually fitting operation takes normally 10 minutes. Once the fitting is completed, one can get realistic 3D facial animation of the person whose motion, including blinking and facial expression, is easily and precisely controllable by commands in real time. Comparing to the cartoon based existing approaches where the number of characters is very limited, the proposed framework enables to generate facial animations of almost unlimited number of characters as far as facial pictures are provided.

5.2. Software Modularity of functional units

As is described in the previous section, the virtual machine model enables highly modularity of each functional units such as SRM, SSM and FSM. Furthermore, communication interface based on the UNIX standard I/O stream helps to develop and debug software modules easily.

5.3. Achievement of natural spoken dialog

Although the implemented mechanism for lip-sync contributes to enhance the naturalness of the synthetic facial animation, number of issues are yet to be implemented to make the agent behave like humans. For example, humans move their heads while they are speaking. Besides the facial animation, realtimeness of conversation is another crucial factor for the agent's naturalness as is described in section 2.1. A simple mechanism for incremental speech recognition has been implemented in the SRM. The mechanism provides frame-synchronous temporal candidates giving maximum scores at the moment before observing the end of utterance. These incremental recognition results will help to achieve interactive spoken dialog including nodding.

6. Related Works

Several attempts have been made to develop ASDA toolkits. Among them, the CSLU toolkit (Sutton and Cole, 1998) is similar to our toolkit. The CSLU toolkit provides a modular, open architecture supporting distributed, cross-platform, client/server-based networking. It includes interfaces for standard telephony, audio devices, and software interfaces for speech recognition. It also includes text-to-speech and animation components. This flexible environment makes it possible to easily integrate new components and to develop scalable, portable speech-related applications. Although the target of both of the toolkits are similar, function wise and implementation wise they are different. Compared to the speech recognizer and speech synthesizer of the CSLU toolkit that support several European languages, our toolkit supports Japanese language. The TTS in the CSLU toolkit is based on “unit selection and concatenation synthesis” from natural speech. It is a data-driven and *non* model-based approach. However, the TTS in our toolkit employs the HMM-based synthesis that is a data-driven and model-based approach. The different approaches give different characteristics to TTS. Generally speaking, the model-based TTS requires less training samples and it can control speech more easily than the non model-based TTS at the expense of speech quality.

Similar system architectures for distributed computing environment are employed in the Galaxy-II (Seneff et al., 1998) of DARPA Communicator (DARPA, 1998), the SRI Open Agent Architecture (OAA) (OAA, 2001), and our toolkit. Each of them have a central module called “Hub”, “facilitator” and Agent Manager (AM) respectively. If compared to the existing systems which employs a large number of commands, our toolkit is more compact and simpler and it has only 8 commands and 2 identifiers so that the programmers can understand and use the toolkit easily.

7. Conclusions

The design and architecture of a software toolkit for building an easy to customize anthropomorphic spoken dialog agent (ASDA) has be presented in this paper. Human-like spoken dialog agent is one of the promising man-machine interfaces for the next generation. The beta-version of the software toolkit described in this paper will be released publicly in the middle of 2002. However, a number of factors are to be improved. Because of the high modularity and simple communication architecture employed in the toolkit, we hope that it would speed up the researches and application development based on ASDA, and as a result the toolkit would be upgraded.

8. References

J. Cassell, T. Bickmore, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. 1999. Requirements for an architecture for embodied conversational characters. In D. Thalmann and N. Thalmann, editors, *Proceedings of Computer Animation and Simulation ’99 (Eurographics Series)*, pages 109–122.

DARPA. 1998. DARPA Communicator Program. http://fofoca.mitre.org/.

Hiroshi Dohi and Mitsuru Ishizuka. 1997. Visual Software Agent: A Realistic Face-to-Face Style Interface connected with WWW/Netscape. In *IJCAI Workshop on Intelligent Multimodal Systems*, pages 17–22.

facetool. 1998. Facial Image Processing System for Human-like ”Kansei” Agent. http://www.tokyo.image-lab.or.jp/aa/ipa/.

Joakim Gustafson, Nikolaj Lindberg, and Magnus Lundeberg. 1999. The August Spoken Dialogue System. In *EuroSpeech*, pages 1151–1154.

Luc Julia and Adam Cheyer. 1999. Is Talking To Virtual More Realistic? In *EuroSpeech*, pages 1719–1722.

T. Kawahara, T. Kobayashi, T. Takeda, N. Minematsu, K. Itou, M. Yamamoto, T. Utsuro, and K. Shikano. 1998. Sharable software repository for Japanese large vocabulary continuous speech recognition. In *ICSLP-98*, pages 3257–3260.

A. Lee, T. Kawahara, and K. Shikano. 2001. Julius — an open source real-time large vocabulary recognition engine. In *European Conf. on Speech Communication and Technology*, pages 1691–1694.

S. Morishima, S. Iwasawa, T. Sakaguchi, F. Kawakami, and M. Ando. 1995. Better Face Communication. In *Visual Proceedings of ACM SIGGRAPH’95*, page 117.

Shigeo Morishima. 2001. Face Analysis and Synthesis. *IEEE Siginal Processing Magizine*, 18(3):26–34, may.

Seiichi Nakagawa and Atsuhiko Kai. 1994. A Context-Free Grammar-Driven, One-Pass HMM-Based Continuous Speech Recognition Method. *Systems and Computers in Japan*, 25(4):92–102, September.

2001. OAA (The Open Agent Architecture). http://www.ai.sri.com/˜oaa/.

Kenji Sakamoto, Haruo Hinode, Keiko Watanuki, Susumu Seki, Jiro Kiyama, and Fumio Togawa. 1997. A Responce Model for a CG Character Based on Timing of Interactions in a Multimodal Human Interface. In *IUI-97*, pages 257–260.

Stephenie Seneff, Ed Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue. 1998. GALAXY-II: A Referece Architecture for Conversational System Development. In *ICSLP-1998*, pages 931–934.

S. Sutton and R. Cole. 1998. Universal speech tools: the cslu toolkit. In *Proceedings of the International Conference on Spoken Language Processing(ICSLP)*, pages 3221–3224.

Masatsune Tamura, Takashi Masuko, Keiichi Tokuda, and Takao Kobayashi. 2001a. Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 805–808, May.

Masatsune Tamura, Takashi Masuko, Keiichi Tokuda, and Takao Kobayashi. 2001b. Text-to-speech synthesis with arbitrary speaker’s voice from average voice. In *Proceedings of European Conference on Speech Communication and Technology*, volume 1, pages 345–348, September.

H. Ushida, Y. Hirayama, and H. Nakajima. 1998. Emotion Model for Life-like Agent and its Evaluation. In *AAAI-98*, pages 62–69.

VoiceXML. 2000. Voice eXtensible Markup Language VoiceXML Ver1.0. http://www.voicexml.org.

T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. 1999. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *EuroSpeech*, volume 5, pages 2347–2350.