NAIST-IS-DD0261001

Doctoral Dissertation

Discourse Timing Model and International Standardization by W3C

Kazuyuki Ashimura

September 18, 2013

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Kazuyuki Ashimura

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Professor Nick Campbell	(Trinity College, Dublin, Ireland)
Associate Professor Tomoki Toda	(Co-supervisor)
Professor Masao Isshiki	(Kanagawa Institute of Technology)

Discourse Timing Model and International Standardization by W3C*

Kazuyuki Ashimura

Abstract

Recently the capability of mobile devices has been much improved, and various Input/Output modalities are available on these devices. On the other hand, ordinary GUIs including touch panels are not necessarily appropriate to use while walking, driving, etc., so the need for easier-to-use products is growing in order to help people in various situations.

To solve this issue, this dissertation tackled the research on speech interface (especially speech synthesis), because it is one of the most useful and easy-to-use interface modalities. It also analyzed utterance timing and speech rate. As a result, the functionality of utterance timing in human dialog was clarified with the basic knowledge of a novel discourse control model obtained. Also a novel network-based collaborative speech interaction framework was proposed based on the result of this research.

Following, the author concluded that a Web-based framework would be the most promising solution to achieve the objective of providing easy-to-use computer interfaces for various users in various environments based on the discussions on multimodal interaction systems with experts from all over the world. Another issue was identified on how to implement components of network-based multimodal applications and interfaces between components varied from vendor to

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0261001, September 18, 2013.

vendor, which burdened application developers to define interfaces and implement all the necessary components themselves.

Therefore the author proposed a standard library named "MMI over WebSocket (MoW)" based on the W3C MMI Architecture specification so that developers could handle a variety of Web applications and Input/Output modalities regardless of their skills. In addition, the author implemented MoW as an actual JavaScript library and evaluated its usability. MoW has made it possible to develop easier-to-use computer interfaces for various users in various situations. MoW is the first attempt in the world to provide a standard and open library for multimodal Web applications, so it is planned for public release to support application developers throughout the world.

Keywords:

Speech interface, Utterance timing, Web standardization, Multimodal interaction, MMI over WebSocket (MoW)

音声対話のタイミングモデルと W3C の国際標準化*

芦村 和幸

内容梗概

近年，スマートフォン等の各種携帯端末の性能向上を受け，ウェブ·アプリケーションに音声認識や手書き認識等の様々な入出力モダリティが利用されるようになってきている．その一方で，スマートフォン等で用いられているタッチパネルの操作は，歩行中や運転中のような「ながら利用」には不向きであり，様々な状況において適切に利用者を支援する技術が求められている．

本論文では，この課題を解決するため，音声インタフェース，特に音声合成によるインタフェースに注目し，発話タイミングと発話速度に関する研究に取り組んだ．その結果，対話における発話タイミングの機能が明らかとなり，対話制御モデルの基礎になる知見を得た．また，この結果に基づき，ネットワークベースの音声対話協調フレームワークの必要性について提案した．

その後，ウェブ·ベースのフレームワークが「多様な利用環境における，より使いやすいコンピュータ·インタフェースを実現する」という目的を達成するのに最適であるという結論を得るとともに，ネットワーク上に分散する様々な機器上の機能を統合するにあたっては，各機能コンポーネントの実装や，コンポーネント間のインタフェースがサービス提供者ごとに異なるため開発者の負担が大きいという課題があることを明らかにした．

この課題を解決するため，開発者の力量に依存することなく柔軟なシステム構築を可能とするべく，W3C の国際標準である W3C MMI アーキテクチャにもとづく標準化ライブラリである MMI over WebSocket (以下，MoW) を提案するとともに，実際に JavaScript ライブラリとして実装した上でその可用性について確認した．MoW により,「多様な利用環境における様々な利用者にとって，より使いやすいコンピュータ·インタフェース」が実現可能となった．なお，MoW は，世界でも初めてのマルチモーダル·ウェブ·アプリケーションのためのオープン

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文，NAIST-IS-DD0261001, 2013 年 9 月 18 日.

な標準ライブラリであり，世界中の開発者の一助とするべく一般公開していく予定である．

キーワード

音声インタフェース, 発話タイミング, ウェブ標準化, マルチモーダル対話, MMI over WebSocket (MoW)

Contents

1. Introduction **1**
1.1 Background and Issues 1
1.1.1 Speech Technology for Easy-to-Use Computing 1
1.1.2 Web as Application Platform 1
1.2 Goals of This Research 2
1.2.1 Discourse Timing Control 2
1.2.2 Standard Framework for Web-based Integration 3
1.3 Structure of the Dissertation 4

2. Analysis of Discourse Timing and Improving Utterance Timing of Speech Synthesis for Discourse Timing Control Framework **6**
2.1 Introduction 6
2.2 Analysis of Utterance Timing 10
2.2.1 Introduction 10
2.2.2 Dialog Speech Corpus 11
2.2.3 Devices for "Conversation Analysis" 13
2.2.4 Basic Definition of "Utterance Timing" 17
2.2.5 Analysis of Inter-Utterance Gap 20
2.2.6 Analysis of Utterance Timing based on Speech Rate 21
2.2.7 Analysis of Utterance Beginning Position based on Addressee's Utterance - using Japanese frequent expressions /hai/ and /uN/ 26
2.2.8 Conclusion 31
2.3 Estimating Speaking Rate in Spontaneous Speech from Z-scores of Pattern Durations 33
2.3.1 Introduction 33
2.3.2 Pattern Extraction Method 34
2.3.3 Experiments 36
2.3.4 Results and Discussion 39
2.3.5 Conclusion 43
2.4 Speech Synthesis using Extended CV as Speech Unit 44
2.4.1 Introduction 44

2.4.2 Extended CV 44
2.4.3 SPEAKS: GUI-based Speech Synthesizer for Template Synthesis 45
2.4.4 Generating Variations for Particles 47
2.4.5 Conclusion 49
2.5 Precise Timing Management using Real-time OS 49
2.5.1 Introduction 49
2.5.2 Experiment on Precise Timing Control using ART-Linux . 49
2.5.3 Result and Discussion 50
2.5.4 Conclusion 50
2.6 Conclusion 53

3. Web Technology and International Standardization by W3C 57
3.1 Introduction 57
3.2 Evolution of Web Technology 57
3.3 "HTML5" and the Open Web Platform 60
3.3.1 History of HTML 60
3.3.2 HTML and Plug-in's 61
3.3.3 "HTML5", a.k.a. the Open Web Platform 62
3.3.4 Latest Status of the HTML5 Specification 63
3.4 Issues on Web Application Development 64
3.4.1 Difficulties with Defining APIs for Various Devices 65
3.4.2 Difficulties with Integration of Distributed Services 66
3.4.3 Difficulties with Dynamic Selection of Multiple Modalities 66
3.5 Conclusion 66

4. Standardized Multimodal Web Application Framework 68
4.1 Introduction 68
4.2 Author's Contribution and Section Structure 69
4.3 W3C MMI Architecture as One of the Promising Solutions 70
4.3.1 Constituents of MMI Architecture 72
4.3.2 MMI Life-Cycle Events 73
4.3.3 Possible Multimodal Web Applications based on MMI Architecture 76

4.4 MoW: MMI over WebSocket . . . 77
4.4.1 The Need for Standard Library for Multimodal Web Applications . . . 77
4.4.2 MoW's Structure . . . 78
4.4.3 MoW's Merits . . . 79
4.5 Evaluation of MoW's Usability from the Viewpoint of Processing Speed and Server Load . . . 81
4.5.1 Preliminary Experiment on Improvement of Processing Speed using WebSocket . . . 81
4.5.2 Experiment on Server Load Decrease using WebSocket . . . 83
4.5.3 Discussion based on the Results . . . 86
4.6 Organizing International Standardization within W3C . . . 86
4.6.1 The Author's Role within W3C . . . 86
4.6.2 The Author's Tasks as the W3C Activity Lead . . . 88
4.6.3 W3C Workshops on Multimodal User Interfaces . . . 88
4.6.4 Interop TF of the MMI WG . . . 90
4.6.5 Japanese Chapter of the MMI WG . . . 91
4.7 Conclusion . . . 94

5. Conclusion **96**
5.1 Summary . . . 96
5.2 Future Work . . . 97
5.2.1 Remaining Issue on Discourse Timing Control . . . 98
5.2.2 Remaining Issue on Standard Framework for Web-based Integration . . . 98

Acknowledgements **99**

References **101**

List of Publications **112**

List of Figures

1 Goals of This Research . . . 3
2 Possible System Construction of Discourse Timing Control Framework . . . 7
3 Local Structure of Dialogs . . . 13
4 Model of the Next Speaker's Action Pattern (modified proposal in [17]) . . . 14
5 Utterance Timing of Turn-taking without Overlap . . . 18
6 Example Transcription of Turn-taking without Overlap (The interval, 0.786, means 0.786sec.) . . . 18
7 Utterance Timing of Turn-taking with Overlap . . . 19
8 Example Transcription of Turn-taking with Overlap . . . 19
9 Distribution of Inter-Utterance Pause Length (with No Utterance Overlaps or Internal Pauses) . . . 22
10 Distribution of Inter-Utterance Pause Length (without Utterance Overlaps but with Internal Pauses) . . . 23
11 Starting Point of Addressee's Utterance based on the Speaker's Utterance . . . 26
12 Distribution of Inter-Utterance Gaps and Its Probability Density Function . . . 28
13 Distribution of Overlap Starting Timing and Its Probability Density Function . . . 29
14 Pattern Extraction Method . . . 35
15 Duration Z-Scores of /naNka/, after Log Conversion . . . 40
16 Duration Z-Score of Patterns . . . 41
17 Correlation between Proposed Method and Conventional Method 42
18 SPEAKS's GUI Image . . . 46
19 How to Generate Particle Variations . . . 48
20 Precision of Time Control without CPU Load . . . 51
21 Precision of Time Control with CPU Load . . . 52
22 Two-Cylinder Engine Model of Discourse Timing . . . 55
23 File Number Ratio per Media Type . . . 58
24 Data Amount Ratio per Media Type . . . 58

25 Transition of Media and Devices 59
26 Open Web Platform Specifications (cited from Mozilla Japan dynamis's slides) 62
27 The Author's Contribution 69
28 The W3C MMI Architecture 72
29 MMI Architecture Components 73
30 Combination of MMI Architecture and Digital TV 77
31 MMI over WebSocket 78
32 Speed Test Environment 82
33 Speed Test Results 83
34 Life-Cycle Event Transaction during the Load Test 84
35 Load Test Environment 84
36 Load Test Results 85
37 Relationship between W3C Groups related to the Author 87
38 Activity Lead's Task 90
39 MMI Prototype (Ver. 1) by HTTP Connection 92
40 MMI Prototype (Ver. 2) by WebSocket 93

List of Tables

1 Speaker Combination and Number of Recorded Dialogs 12
2 Possible Position for Interruption (generated based on the proposal in [17] 16
3 Gap between the End of Speaker X's Utterance and the Beginning of Speaker Y's Utterance 20
4 Functions of Utterances at the Three Peaks 24
5 Functions of Utterances Which Consist the Three Peaks 25
6 Frequency of /hai/ and /uN/ 28
7 Frequency of Utterance-Contexts 30
8 Patterns of 5 Phonemes, Most Frequent 10 Patterns 39
9 Result of Listening Test 39
10 Three Groups of Correlation 43
11 Constructions of Extended CV 45

12 F_0 Pattern of /chi/ and /ka/ within Date Data 48
13 Two-Cylinder Model 55
14 Results of the Load Test 85
15 W3C Workshops on MMI Architecture 89

1. Introduction

1.1 Background and Issues

1.1.1 Speech Technology for Easy-to-Use Computing

Recently the capability of IT devices including PCs, smartphones and digital TVs is much improved thanks to high performance CPUs and fast network connection. However, the more features a specific device has, the more complicated its usage becomes. So the need for easier-to-use computer interface is getting even stronger, and it is necessary to allow all the people to select their preferred interface modalities dynamically based on their needs. For example, it is difficult for aged people to identify small characters on high-resolution displays, so they may prefer larger characters. Also, it is important for drivers to focus on driving when they want to use car navigation systems to get direction information, so they may prefer speech interface.

Among the variety of computer interface modalities, speech interface is one of the most common communication tools that we use for every day living. Hence speech interface technology has been under research for years as one of the promising candidates of easy-to-use computer interfaces, and recently speech recognition technology and speech synthesis technology have been much improved thanks to (1) recognition algorithm using Hidden Markov Model (HMM), (2) high performance CPUs and (3) large-scale speech corpora. However, existing speech systems mainly handle read speech, and it is still difficult to deal with speaker specific utterance style such as speaking rhythm and utterance timing within everyday conversations.

1.1.2 Web as Application Platform

The World Wide Web (or simply "The Web") is used as one of the "Killer Applications" of network-based computer systems by everybody throughout the world. Web browsers were developed to access the information on the Web, and Hypertext Markup Language (HTML) [1, 2] has been used as the standard way to describe Web contents so that page designers and application developers can generate Web contents easily without attention to the difference between terminal

devices or browser software.

Now that it is more than 20 years since the birth of the Web, the contents of the Web are getting more and more dynamic and interactive. In addition, the Web has become a platform for "Web Applications", which are Web contents used as applications including bookstores, interactive games and video streaming services. These days there are even Web applications in the market with natural conversational interfaces that use high performance speech recognition and speech synthesis services on the server-side, e.g., Google Search [3], Apple's Siri [4, 5] and NTT Docomo's Shabette-Concier [6, 7]. However, these systems have issues on interoperability, so (1) people cannot reuse components of these systems (e.g., speech recognition or speech synthesis) for their own systems and (2) it is not possible to add their own generated modality components like gesture recognition or 3-D face image synthesis to these systems.

1.2 Goals of This Research

As Fig. 1 shows, this research handles (1) discourse timing control and (2) standard framework for Web-based integration to solve the issues mentioned in Section 1.1 and make computer interfaces even more usable and friendly.

1.2.1 Discourse Timing Control

To solve the issue on computer interface using speech technology, this research analyzed a large amount of dialog speech data, and a basic heuristics of discourse timing management was clarified, i.e., "**In Japanese conversation, the addressee waits for 1 Mora after the utterance completion of the current speaker to start his/her own utterance.**", and it is expected that the heuristics can be used as the basis of the possible discourse timing model though the detailed model for automatic control needs further research. On the other hand, the result implied the need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of 1 Mora (around $100-200$ ms) based on the heuristics. Therefore a framework using a real-time OS for precise and stable timing control was proposed.

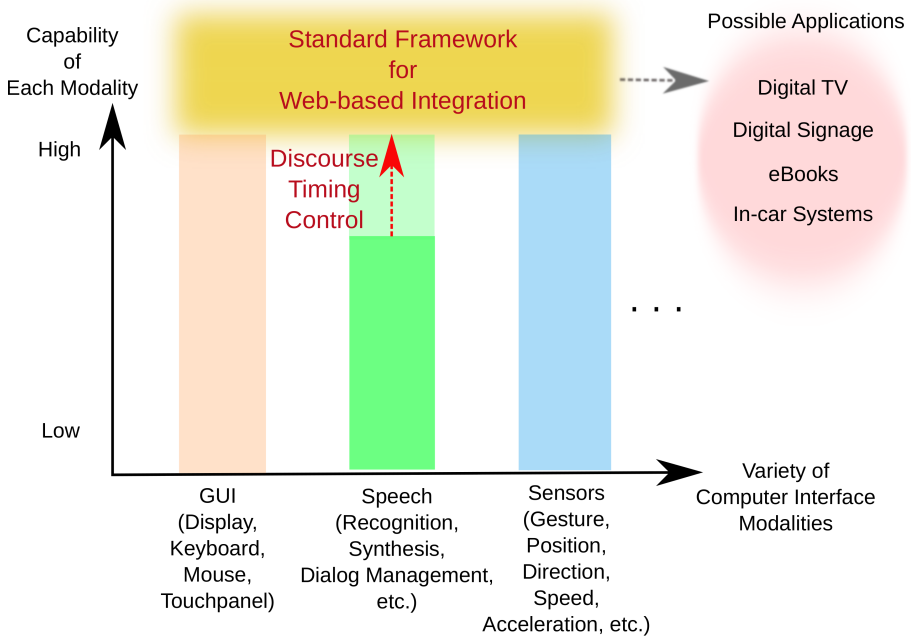

Figure 1. Goals of This Research

1.2.2 Standard Framework for Web-based Integration

To solve the issue on the development of multimodal Web applications, this research proposes a standard library named "MMI over WebSocket (MoW)" based on the W3C MMI Architecture specification [8, 9, 10] so that (1) developers can handle variety of Web contents and Input/Output modalities regardless of their skills and (2) users can choose their preferred interface modalities dynamically based on their situations. Also this research describes how to implement the library and evaluate its usability from the viewpoint of processing speed and server load. Following, this research explains the current status of standardization of the library within W3C.

Note that HTTP is usually used as the network protocol for multimodal Web applications, but it is likely that the connection is not fast enough for interactive applications. Therefore this research proposed that the standard library use WebSocket as protocol for device interaction to improve the connection speed, and confirmed that the speed of WebSocket connection could be 29 times faster

than that of HTTP. Hence the proposed library "MoW" is expected to be useful for precise and stable discourse timing control.

1.3 Structure of the Dissertation

The rest of this dissertation is organized as follows:

Chapter 2 analyzes a large amount of dialog speech data to identify the basic heuristics of discourse timing control to improve the usability of computer interfaces. Also the chapter proposes a speaking style analysis method which is independent of text information, because sometimes it is difficult and/or expensive to obtain accurate transcriptions of natural dialog data automatically (=by speech recognition) or even manually (=by human labellers) due to its great variation in articulation. The chapter proposes a new speech synthesis method to improve rhythm of synthesized speech, thus discusses a possible framework for discourse timing management using a real-time OS.

Chapter 3 summarizes an overview of Web technology and its global standardization by W3C as the background of a Web-based approach for integrating various modality components and Web services. The chapter also explains the existing issues on Web application development, especially for advanced multimodal Web applications.

Chapter 4 proposes a standard library for Web application authoring named MoW based on the W3C MMI Architecture specification to solve the issues on development of advanced multimodal Web applications described in Chapter 3. The chapter then describes how to implement MoW, and evaluates its usability from the viewpoint of processing speed and server load.

Note that HTTP is usually used as the network protocol for Web applications, but it is likely that the connection speed is not fast enough especially for interactive applications. Therefore this chapter proposes that WebSocket should be used as the network protocol for MoW to improve the

connection speed, and confirms that the speed of WebSocket connection to be 29 times faster than that of HTTP.

The chapter summarizes the role of the author within the W3C Multimodal Interaction Working Group to deploy MoW as a basic framework for multimodal Web application development, and overviews the current status and the future plan of its international standardization.

MoW is the first attempt in the world to provide a standard and open library for multimodal Web applications.

Chapter 5 finally summarizes the dissertation and describes the future direction.

2. Analysis of Discourse Timing and Improving Utterance Timing of Speech Synthesis for Discourse Timing Control Framework

2.1 Introduction

Among the variety of computer interface modalities, speech interface is one of the most common communication tools that we use in our every day lifestyles, so speech technology has been under research for years as one of the promising candidates of easy-to-use computer interfaces. There are varieties of dialog-based computer systems [12, 13, 14, 3, 4, 5, 6, 7] proposed, and recently the capability of these systems has been much improved thanks to (1) evolution of algorithm using Hidden Markov Model (HMM), etc., (2) large-scale speech corpora and (3) high performance CPUs and networks.

However, it is still difficult to handle speaker-specific utterance style such as speaking rhythm and utterance timing within everyday conversations, because speakers' utterances include not only linguistic information but also paralinguistic information like emotions, intentions and attitudes. The collaboration between the components of the existing dialog-based systems is not rich enough. For example, speech synthesis output does not necessarily use all the features extracted from the user's input by speech recognition. Therefore this chapter analyzes large amounts of natural everyday dialog speech data to clarify the relationship between the user's input speech and the computer system's output speech from the viewpoint of discourse timing, and discusses what the possible model for discourse timing management should be.

In addition, this chapter proposes a novel network-based framework (Fig. 2) for discourse timing management so that dialog-based computer systems can transfer paralinguistic meanings using utterance timing and be even more friendly to all. The basic description of the proposed framework's constituents is shown in Fig. 2.

Hard RT Manager: The **Rec** component accepts the input speech, distinguishes utterance from silence, and gets the utterance start time and the end time. It also sends the input utterance to **Speech Timing Manager**.

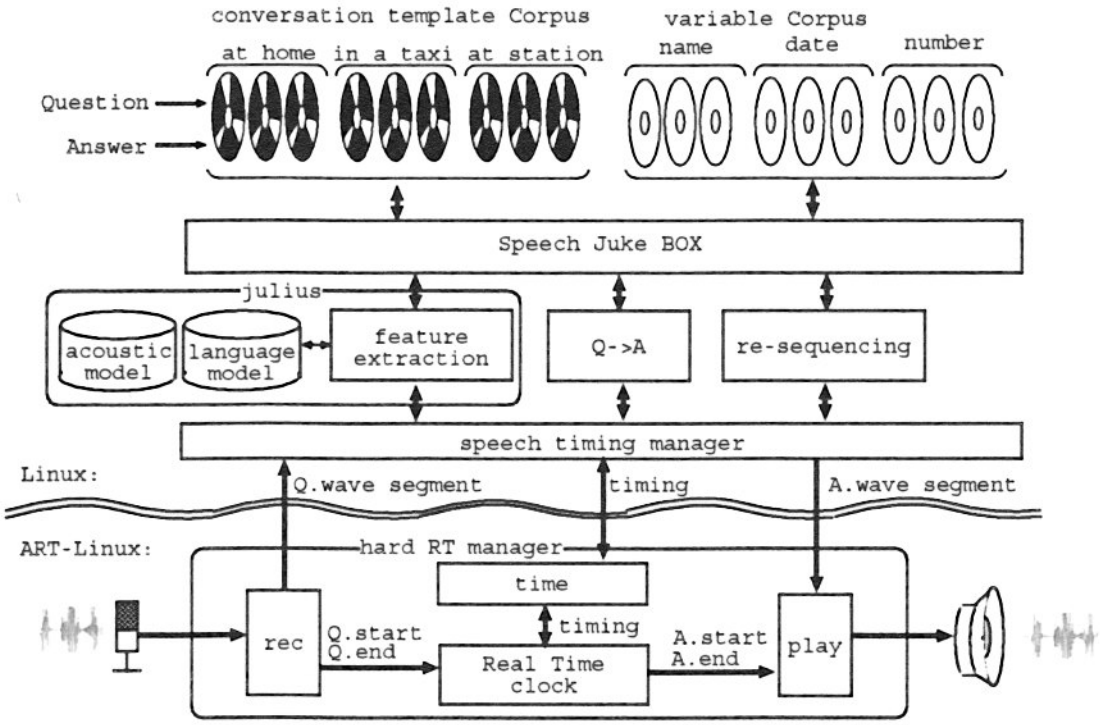

Figure 2. Possible System Construction of Discourse Timing Control Framework

The **Time** component communicates with **Speech Timing Manager** and calculates the appropriate timing for the computer system to respond to the user.

The **Real Time Clock** component ticks away milliseconds, and calculates the appropriate timing for the computer system's response based on (1) the **Rec** component's clarified start/end time of the user's utterance and (2) the **Time** component's clarified appropriate timing.

The **Play** component gets the synthesized output speech from **Speech Timing Manager** and plays it at the appropriate timing based on the timing calculated by the **Real Time Clock** component.

Speech Timing Manager: A controller for utterance timing management based on the discourse timing model. It handles (1) user's inputted utterance sent from the **Rec** component, (2) appropriate timing computed based on the discourse timing model and (3) synthesized speech to be sent to the **Play** component. It also communicates with **Feature Extraction**, **Q&A Conversion** and **Re-sequencing**.

Feature Extraction (Speech Recognition): Extracts key features for data retrieval using speech recognition technology.

Q&A Conversion: Gets answer(s) for user's asked question using **Speech Juke Box**. Speech output timing may be affected by the topics and content types of the answer(s).

Re-sequencing (Speech Synthesis): Generates a sequence of speech waveform based on the answer generated by **Q&A Conversion**.

Speech Juke Box: A data retrieval system including a large scale speech corpus which stores not only speech data but also various metadata after categorizing all the data based on topics and content types. Metadata is used for data retrieval to get the appropriate answers as quickly as possible.

To materialize the proposed framework above, we tackle the following constituents in this chapter:

- Speech Timing Manager
- Feature Extraction
- Re-sequencing
- Hard RT Manager

Note:

Q&A Conversion and **Speech Juke Box** are out of scope of this dissertation.

The rest of this Chapter has the following structure:

Section 2.2 describes the basic concepts and conditions of the data analysis, i.e., (1) the dialog speech corpus for the data analysis, (2) devices for "conversation analysis" and (3) definition of "utterance timing". The section describes the results of various conversation data from several viewpoints to see the model of timing management for **Speech Timing Manager**, and clarifies a heuristics of discourse timing.

Section 2.3 proposes a speaking style analysis method for **Feature Extraction** in addition to speech recognition, e.g., "Large Vocabulary Continuous Speech Recognition Engine Julius" [15]. The proposed method is designed to be independent of text information, as it is sometimes difficult and/or expensive to obtain accurate transcriptions based on natural dialog data automatically (=by speech recognition) or even manually (=by human labellers) due to its great variation in articulation.

Section 2.4 proposes a new speech synthesis method which uses "Extended CV" as the speech unit to improve the rhythm of synthesized speech for **Resequencing**.

Section 2.5 discusses a possible framework for **Hard RT Manager** to manage discourse timing precisely using a real-time OS.

Section 2.6 summarizes this chapter.

2.2 Analysis of Utterance Timing

2.2.1 Introduction

Speakers' utterances include not only linguistic information but also their emotions, intentions and attitudes, and transfer mental meanings beyond literal meanings of the utterances themselves. It is likely that the timing of utterances is as important as the content of the utterance especially in everyday conversations. To see the relationship between utterance timing and speaker's intention, this section does the following:

- Explaining the speech corpus for data analysis
- Describing the devices for data analysis, i.e., turn-taking and action pattern
- Defining the target of the analysis, i.e., what utterance timing is
- Analyzing 30-min dialog data spoken by two people from the viewpoint of turn-taking and action pattern
- Analyzing the current speaker's speech rate and discusses addressee's utterance timing based on the results, because utterance timing is the relationship between the two speakers' utterances and the speech rate of both speakers should be considered
- Analyzing the start position of the addressee's utterance based on the current speaker's speech rate using a large-scale spoken dialog corpus

2.2.2 Dialog Speech Corpus

One-pair Data A speech corpus, which consists of a 30-minute dialog by (1) a 22 year-old male (=Speaker X) and (2) a 42 year-old male (=Speaker Y), was used for the data analyses in Section 2.2.5 and 2.2.6. The following annotation is added to the corpus based on the result of spectrogram inspection:

- Transcribed text of the utterance
- Start time and end time of the utterance
- Pause length within the utterance
- Overlaps between utterances
- Speech act tags (Question-Answer pair, Back channel, etc.)

This data is part of the dialog speech by the two participants which is continuously recorded for 10 weeks (30 mins per week). More specifically, it is the recorded result of the second week out of the total 10 weeks, and it is likely that both participants are familiar with the recording procedure yet less so to each other.

Note:

The speech data in the corpus had the following tendency when we skimmed all the recorded data:

> Speaker X (22 year-old male) initiates the topics whereas Speaker Y (42 year-old male) gives responses.

So we analyzed the dialog data mainly from the viewpoint of turn-taking from Speaker X to Speaker Y.

Full Data A large-scale spoken dialog corpus, which consists of 144 dialogs by six speakers (=JFA, JFB, JFC, JMA, JMB, JMC) listed in Table 1, was used for the data analysis in 2.2.7. The length of each recorded data is 30 minutes. The mother tongue of all the speakers is Japanese, and all of them were strangers to each other when the recording started.

Table 1. Speaker Combination and Number of Recorded Dialogs

	Addressee						
Speaker	JFA	JFB	JFC	JMA	JMB	JMC	Total
JFA	-	10	-	10	-	-	20
JFB	10	-	10	-	-	10	30
JFC	-	10	-	-	11	-	21
JMA	10	-	-	-	11	-	21
JMB	-	-	11	11	-	10	32
JMC	-	10	-	-	10	-	20
Total	20	30	21	21	32	20	144

Note. JF* means female and JM* means male.

The following annotation is added to the corpus based on the result of spectrogram inspection:

- Transcribed text of the utterance
- Start time and end time of the utterance
- Speech act tags (Question-Answer pair, Back channel, etc.)

- Emotion tags, attitude tags

2.2.3 Devices for "Conversation Analysis"

Adjacent Pairs and Turn-taking Rules As the starting point of data analysis, we use "Adjacent Pairs" and "Turn-taking Rules", the basic devices for the "Conversation Analysis", to analyze (1) the local structure of dialogs and (2) the action pattern of the addressee based on the syntax of the speaker's uttered context.

An "Adjacent Pair" (Fig. 3) is the fundamental pair of speeches generated by the two participants which achieves the interaction such as "question and answer".

First Part≺Turn-taking≻Second Part

Figure 3. Local Structure of Dialogs

The following three constituents of an "Adjacent Pair" collaboratively construct the local structure of dialog on a specific speech activity:

- The first part of an "Adjacent Pair"
- The second part of an "Adjacent Pair"
- The "Turn-taking" between those two parts

When constructing dialogs between the user and the computer based on "Adjacent Pairs" and "Turn-taking Rules", we need to clarify the issue on "when and how utterances should be started, continued and ended". For that purpose, we analyze what happens within the dialogs between two people as the starting point.

Action Pattern of the Next Speaker based on Syntax In languages such as English which have SVO structure (="Tom eats apples."), it is possible to predict the speaker's intention based on the syntactic structure of the speaker's uttered content. However, Takanashi et al. [17] shows that speaker's intention

can be predicted even in languages such as Japanese which have SOV structure (=“トムがリンゴを食べる．”; Tom apple eat) as well. Thus a model on the next speaker’s action pattern is proposed [17] as shown in Fig. 4.

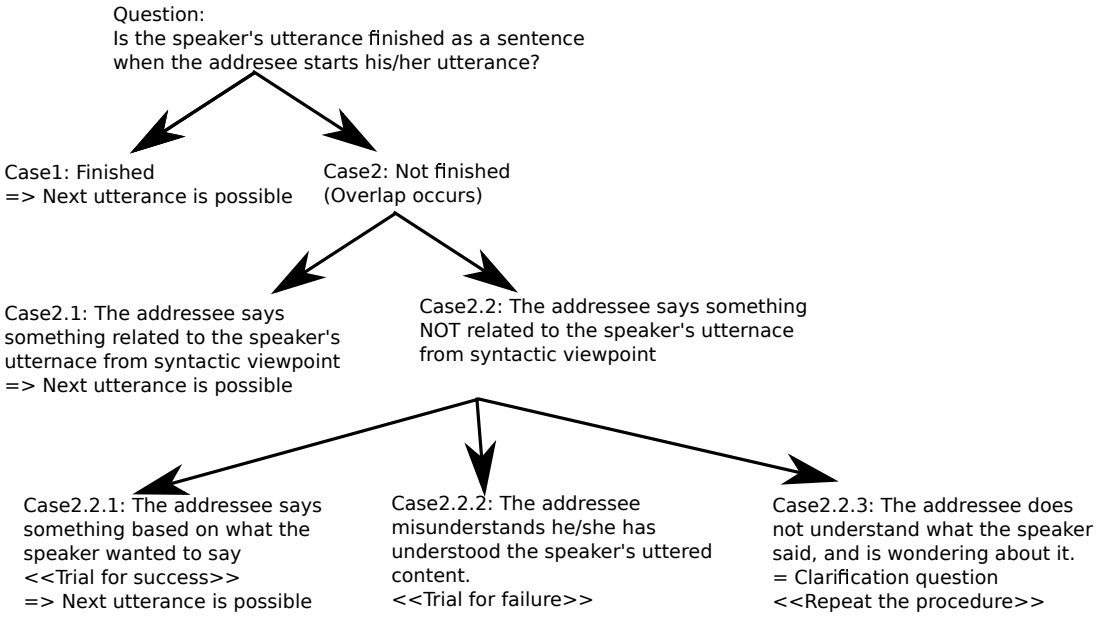

Figure 4. Model of the Next Speaker’s Action Pattern (modified proposal in [17])

Fig. 4 shows the basic model that can be summarized as below:

Question: Is the speaker's utterance finished as a sentence when the addressee starts his/hers?

Case1: If finished, next utterance by the addressee is possible.

Case2: If not finished, utterance overlap occurs and the addressee starts his/her utterance before the speaker finishes.

Case2.1: If the addressee says something related to the speaker's utterance from syntactic viewpoint, the next utterance by the addressee is possible.

Case2.2: If the addressee says something NOT related to the speaker's utterance from syntactic viewpoint, check what the addressee does.

Case2.2.1: If the addressee says something based on what the speaker wanted to say, the next utterance by the addressee is possible.

Case2.2.2: If the addressee misunderstands that he/she has understood the speaker's uttered content, the addressee's utterance is 'failure'.

Case2.2.3: If the addressee does not understand what the speaker said and is wondering about it, the addressee will make a clarification question and the procedure goes back to the "Question".

Several possible positions of interruption from syntactic viewpoint are proposed as listed in Table 2 [17].

Table 2. Possible Position for Interruption (generated based on the proposal in [17]

Possible position for interruption	Examples
Before or after conjunctions or conjunctive particle	Conjunctive particles, e.g., "～/から (kara)/～"
	Conjunctions, e.g., "～/けれど (keredo)/～"
Between the first part and the second part of the theme-rheme structure	"名前は (name)/花子だ (Hanako)" "ポチは (Pochi)/犬だ (dog)"
The end of the speaker's turn	In the middle of continuous sentence-ending particles, e.g., "です (desu)/ねえ (ne:)"
	Right before sentence-ending particles such as "です (desu)" and "ます (masu)", e.g., "描いたん (kaitaN)/です (desu)"
Between the first part and the second part of Koou expression in Japanese	Modality of recognition, e.g., "何も (nanimo)/ない (nai)"
	Modality of tense, e.g., "そのあと (sonoato)/なった (natta)"
	Predicate is predictable, e.g., "歌が (utaga)/出てくる (detekuru)"
	Syntax is completed before the predicate, e.g., "300人以上 (300 niN ijou)/いた (ita)"
Right before an inverted predicate	"復習·予習なんですよ (fukushu:·yoshu: naNdesuyo)/順番から (juNbaNkara)"

Note. "/"s in the examples are the possible interruption positions.

2.2.4 Basic Definition of "Utterance Timing"

Here we define "Utterance Timing" as the relative position within the time context between Speaker X's utterance and Speaker Y's utterance.

On the other hand, there are the following two possible cases of turn-taking as described in Fig. 4:

- Case1: Speaker X's utterance has already finished as Speaker Y's utterance starts (=turn-taking without overlap)
- Case2: Speaker X's utterance has not finished yet as Speaker Y's utterance starts (=turn-taking with overlap)

We see the relationship between SpeakerX's utterance and Speaker Y's utterance after classifying all the cases into the above two categories based on whether or not there is any overlap between those two utterances.

Turn-taking without Overlap There is no overlap observed between the current speaker's utterance and the next speaker's utterance (Fig. 5). The addressee listens to the current speaker's utterance until the end, understands the intention, then starts his/her own utterance after waiting for an appropriate timing based on the content and/or situation (Fig. 6).

In this case, utterance timing is defined as the length of the pause between Speaker X's utterance and Speaker Y's utterance. Here the timing of the next speaker's (=addressee's) utterance is based on "Case1: Finished" (=if the speaker's utterance finished as a sentence, next utterance by the addressee is possible) in Fig. 4. There were many cases observed in the dialog data within the corpus.

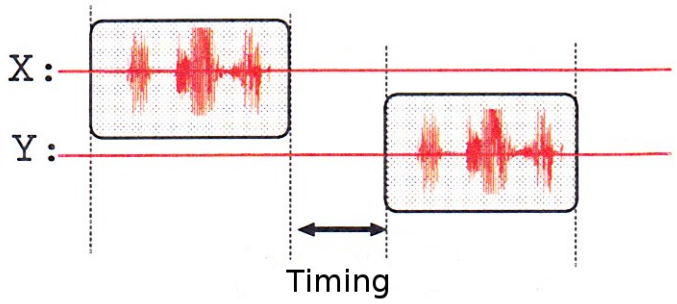

Figure 5. Utterance Timing of Turn-taking without Overlap

((Speaker X asks Speaker Y if Y likes movies when starting the topic of movies))

→ X: 映画は好きですかー? (Do you like movies?)
(0.786)
→ Y: やー，僕? (Well, me?)

Figure 6. Example Transcription of Turn-taking without Overlap (The interval, 0.786, means 0.786sec.)

Turn-taking with Overlap There is an overlap observed between the speaker's utterance and the addressee's utterance, so the addressee's utterance starts in the middle of the speaker's utterance (Fig. 7). The addressee does not wait until the end of the current speaker's utterance, but starts his/her own utterance only after waiting for a relevant timing.

In this case, the utterance timing cannot be defined by the gap between the current speaker's utterance and the addressee's utterance but should be defined as "Where in the current speaker's utterance the addressee's utterance starts?" (Fig. 8).

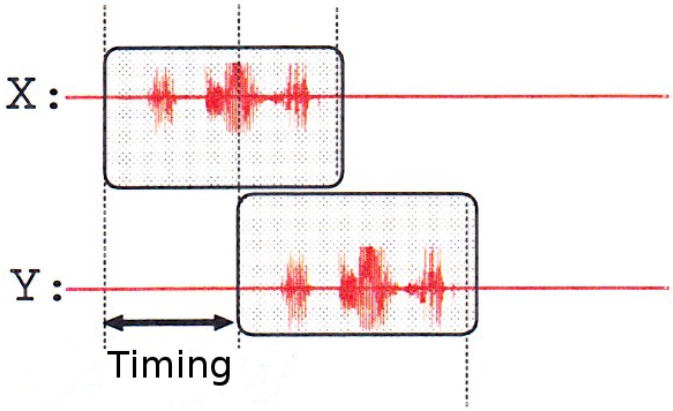

Figure 7. Utterance Timing of Turn-taking with Overlap

((X responds to Y's greeting))

→ Y: あ こんにちはー [どーも (Ah, hello. Hi.)
→ X: [ひさしぶりですー (Long time no see!)

Figure 8. Example Transcription of Turn-taking with Overlap

2.2.5 Analysis of Inter-Utterance Gap

The gaps between the end of Speaker X's utterances and the beginning of Speaker Y's utterances were measured using the "One-pair Data" described in Section 2.2.2 as Table 3 shows. Note that here the targets of the analysis were the types of Speaker Y's utterances which occurred more than twice during the turn-takings from Speaker X to Speaker Y.

Table 3. Gap between the End of Speaker X's Utterance and the Beginning of Speaker Y's Utterance

Overlaps	Types of contents	Frequency	Mean value of the gap (sec.)	Std. dev. of the gap (sec.)
Without Overlap	Back channels	75	0.329	0.504
	Continuing own speech	89	0.672	2.160
	Evaluation-Agreement pair	6	0.431	0.311
	Question-Answer pair	16	0.588	0.633
	Difficult to answer	3	0.521	0.455
	Total	189	0.519	1.532
With Overlap	Back channels	140	-0.4060	2.054
	Continuing own speech	18	-0.367	0.507
	Evaluation-Agreement pair	3	-1.549	0.899
	Question-Answer pair	8	-0.235	0.193
	Overlapping at the end	79	-0.469	0.643
	Total	191	-0.448	1.801

Note. The gap is negative if there is an overlap between Speaker X's utterance and Speaker Y's utterance.

We analyzed the details of (1) starting point of the next speaker's (=Speaker Y's) utterance and (2) utterance timing during all the turn-takings which Table 3 shows after categorizing all the data into two cases based on whether or not there was an overlap between the utterance of Speaker X and Speaker Y. As a result, almost all the turn-takings with overlaps matched "Case2.1" in Fig. 4 (=The addressee says something related to the speaker's utterance from syntactic viewpoint.) or "Case2.2.1" in Fig. 4 (=The addressee says something based on what the speaker wanted to say.), and all the transitions occurred smoothly. It seems that when one of the speakers continued to speak too long, he/she needed

to provide a queue to the addressee so that the addressee could interrupt his/her utterance smoothly.

The following are examples of the queues in the speech corpus (=One-pair Data):

- separating a sentence into short chunks
- prolonging vowels or /sU/ (from /desU/ or /masU/) at the end of a sentence

2.2.6 Analysis of Utterance Timing based on Speech Rate

We analyzed utterance timing from the viewpoint of turn-taking using 30-min dialog speech data in Section 2.2.5, and found that it seems that utterance timing is determined based on two different bases depending on whether or not there is an overlap between the current speaker's utterance and the addressee's utterance. However, utterance timing is the relationship between "the current speaker's utterance" and "the addressee's utterance", so we need to consider the speech rate of both speakers. Therefore we analyzed the addressee's utterance timing based on the current speaker's speech rate using the "One-pair Data" described in Section 2.2.2.

For that purpose, we first categorized the speech data into two categories based on the existence of overlaps, and then analyzed the distribution of inter-utterance pause after normalizing the pause length by the current speaker's speech rate in order to investigate the details of the mechanism of utterance timing in the cases without overlaps between utterances. On the other hand, regarding the cases with overlaps between utterances, we investigated where the overlaps occur and the frequency at each position.

Turn-taking without Overlap We found that there were two different distributions of inter-utterance pause length depending on whether or not there is a pause within the current speaker's utterance. So we once categorized all the dialog data based on the existence of pauses within the current speaker's utterance, then analyzed the distribution of inter-utterance pause length after normalizing by the current speaker's speech rate. More specifically, we used the following "Mora speech rate" (MSR) as the index and investigated its distribution:

$$Mora\ speech\ rate\ (MSR) = \frac{Mora\ number\ of\ the\ current\ speaker's\ utterance}{Length\ of\ the\ current\ speaker's\ utterance\ (sec.)}$$

The range of the inter-utterance pause length distribution is from 1 Mora to 12 Moras, and there is a peak at 0–1 Mora (Fig. 9). So it is likely that the addressee waits for the end of the current speaker's utterance, to start his/her own utterance as soon as possible.

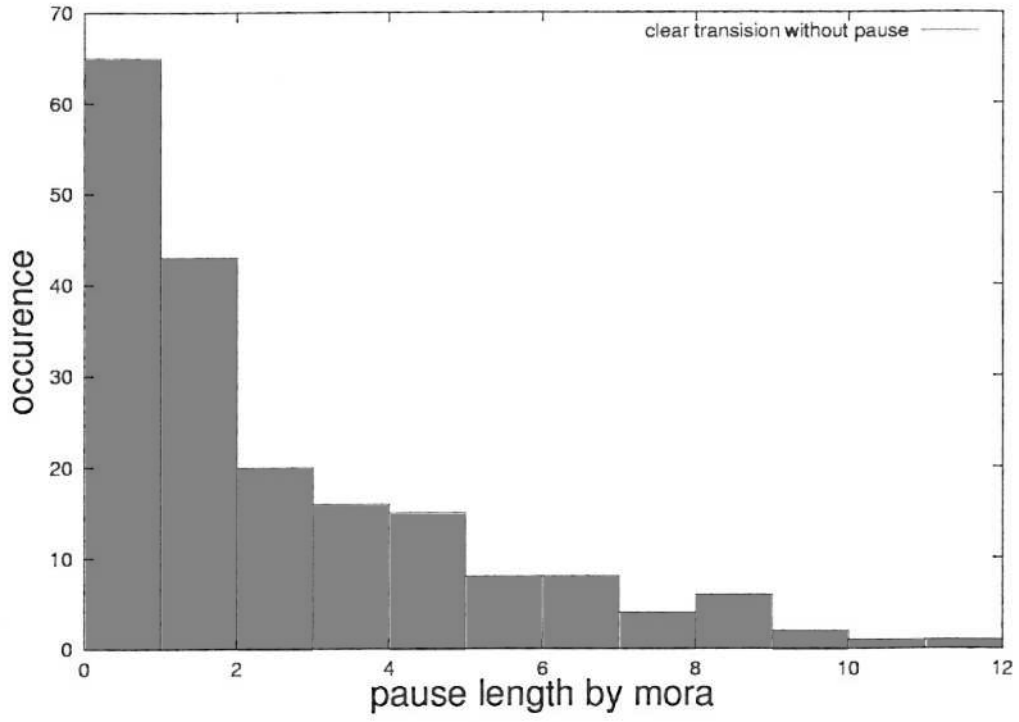

Figure 9. Distribution of Inter-Utterance Pause Length (with No Utterance Overlaps or Internal Pauses)

In the cases with pauses within the current speaker's utterance, the range of the inter-utterance pause length distribution is from 0 Mora to 9 Moras, and there are three peaks, 1–2 Moras, 3–4 Moras and 7–8 Moras (Fig. 10).

We analyzed concrete speech acts of utterances which correspond to the three peaks as Table 4 shows, and found that simple back channel at the peak of 1–2

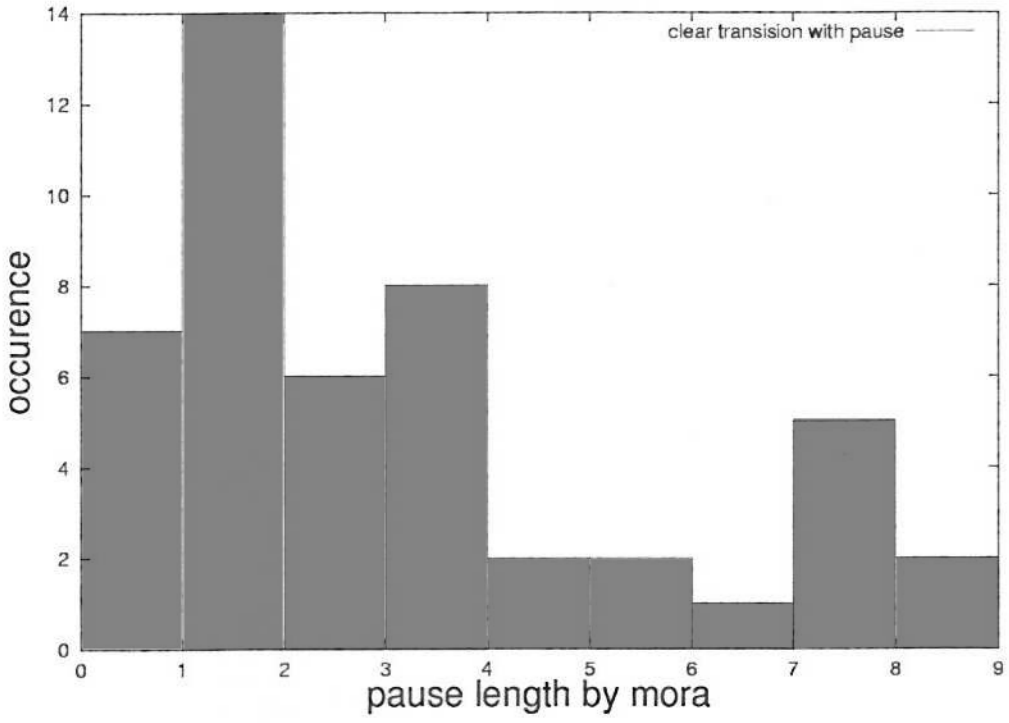

Figure 10. Distribution of Inter-Utterance Pause Length (without Utterance Overlaps but with Internal Pauses)

Moras and utterance continuation at the peak of 3–4 Moras are the distinctive features respectively. On the other hand, there was no distinctive feature at the peak of 7–8 Moras.

Table 4. Functions of Utterances at the Three Peaks

Peak of pause length distribution (Mora)	Function of addressee's utterance	Frequency
1–2	Back channel (simple back channel)	6
	Continuation of utterance	4
	Back channel (expression of reception)	3
	Back channel (response)	1
3–4	Continuation of utterance	4
	Question-Answer pair	2
	Back channel (expression of reception)	1
	Back channel (simple back channel)	1
7–8	Providing new information	1
	Question-Answer pair	1
	Question	1
	Continuation of utterance	1
	Back channel (response)	1

Turn-taking with Overlap We analyzed all the data with overlaps from the viewpoints of speech and linguistics, and found the following:

From the viewpoint of speech: Many of the overlaps occur at the following four positions from the viewpoint of speech:

1. Around the end of the utterance
2. Prolonged vowels, /N/'s and /sU/'s
3. Phrase boundary except the end or the beginning of the sentence
4. The beginning of the utterance

From the viewpoint of linguistics: Many of the overlaps occur between the two constituents of Koou ("呼応") relations in Japanese from a linguistic viewpoint (Table 5).

Table 5. Functions of Utterances Which Consist the Three Peaks

Overlap occurring position	Frequency	Percentage ($Frequency/422$)
Distinctive features from the viewpoint of speech:		
Around the end of the utterance	126	30%
Prolonged vowels, /N/'s and /s/'s	80	19%
Phrase boundaries except the end or the beginning of the sentence	63	15%
The beginning of the utterance	52	12%
Around double consonants	14	3%
Around loughs	14	3%
Around pauses within utterances	11	3%
Distinctive features from the viewpoint of linguistics:		
Between the two constituents of Koou relations in Japanese	55	13%
Before or after conjunctions or conjunctive particles	4	1%
Between the first part and the second part of the theme-rheme structure	1	0.2%
Right before an inverted predicate	3	0.7%
Total	422	100%

2.2.7 Analysis of Utterance Beginning Position based on Addressee's Utterance - using Japanese frequent expressions /hai/ and /uN/

We analyzed spoken dialog data and found there is a possibility that the starting point of the addressee's utterance depends on whether there is a pause within the current speaker's utterance, and reported what was happening in actual dialog speech data in Section 2.2.5 and Section 2.2.6. However, the data amount for those analyses was not necessarily enough for statistical analysis. We then analyzed the starting position of the addressee's utterance based on the current speaker's speech rate using the "Full Data" described in Section 2.2.2.

It is reported that pause length within read speech is affected by mora timing defined by mean length of the moras in the uttered words right before the pause [18]. To see if pause length is affected by mora timing within spontaneous dialog speech as well, the section between (1) the starting position of a specific speaker's utterance and (2) the starting position of the next utterance of the same speaker is used as the "**Utterance Unit**" (Fig. 11) to handle both the turn-takings with overlaps and the ones without overlaps using a unified mechanism.

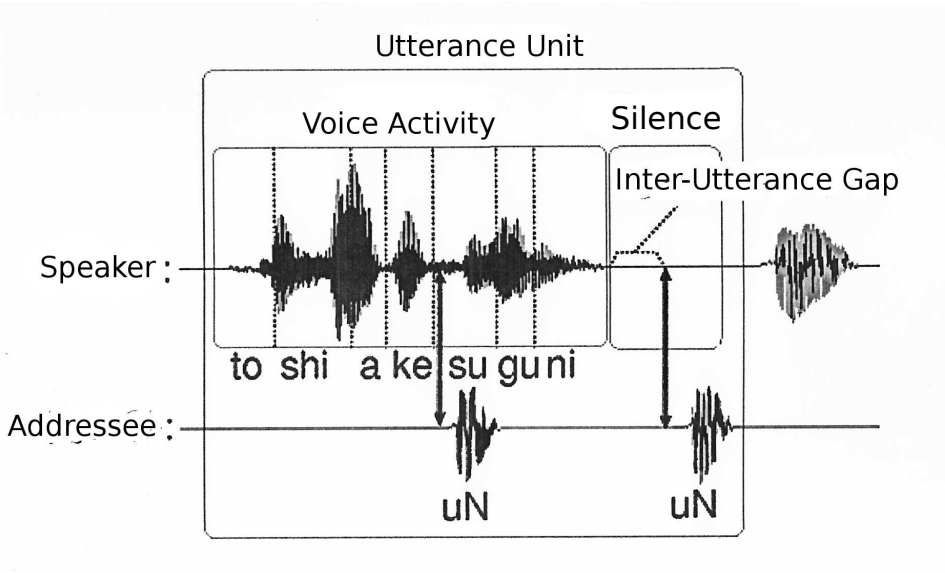

Figure 11. Starting Point of Addressee's Utterance based on the Speaker's Utterance

In the case there is overlap between the current speaker's utterance and the addressee's utterance, we can define the **start timing of the addressee's utterance** using the **mora position of the current speaker's utterance** (see the first "uN" in Fig. 11). On the other hand, if there is no overlap, the addressee's utterance starts during the silence section which we cannot identify the start timing of the addressee's utterance based on the current speaker's mora position. However, it is likely the gap between the end of the current speaker's utterance and the addressee's utterance (=**inter-utterance gap**) is affected by mora timing of the current speaker's utterance. So we use the length of **inter-utterance gap** as the index for the starting position of the addressee's utterance after normalizing by the mean mora length of the current speaker's voice activity (see the second "uN" in Fig. 11).

To get precise time information of each mora within all the speakers' utterances, 115,154 utterance segments were extracted from the speech corpus based on the speech waveforms, transcriptions and start/end times of each utterance. Then 79,010 **utterance units** which included addressee's utterances were extracted. Pronunciation information was added using a Japanese Morphological Analysis System "ChaSen" [19] to the data within the corpus, and time information of each mora was calculated based on the result of phonemic segmentation using a large vocabulary continuous speech recognition engine "Julius" [15, 16]. 68,753 from the 79,010 **utterance units** (which were able to be phonemically segmented) were then used for the analysis of the starting positions of addressee's utterances.

The target 68,753 **utterance units** included 100,012 addressee's utterances. However, the transcribed text of addressee's utterances had 58,295 variations, and it was difficult to analyze the details of each variation. So we analyzed the two most frequent patterns, /hai/ and /uN/. Table 6 shows the frequency of /hai/ and /uN/ classified based on whether or not there is an overlap between the current speaker's utterance and the addressee's utterance.

The following is the analysis result of /hai/, but the result of /uN/ was similar.

Utterance Units without Overlaps Pause length within read speech after log conversion is approximately equal to a normal probability density function

Table 6. Frequency of /hai/ and /uN/

Overlap between the current speaker's utterance and the addressee's utterance	/hai/	/uN/
Without overlap	1,700	1,295
With overlap	2,084	2,428
Total	3,784	3,723

[18]. To see if **inter-utterance gap** within spoken dialog speech also has similar distribution, we computed the probability density function of inter-utterance gap after log conversion. Note that we used the length of **inter-utterance gap** after normalizing by the mean mora length of the current speaker's voice activity.

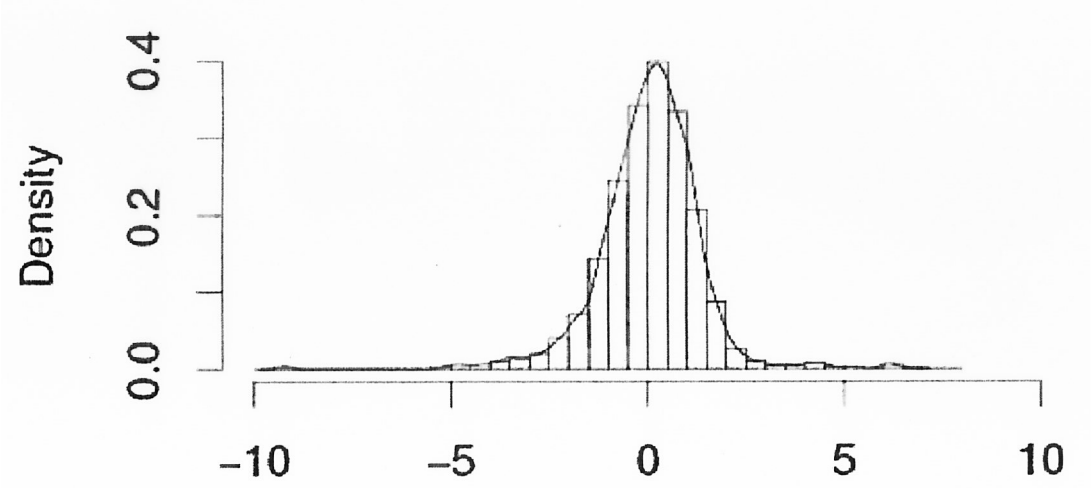

Figure 12. Distribution of Inter-Utterance Gaps and Its Probability Density Function

Based on the result of section 2.2.6 (Fig. 9), the distribution of inter-utterance gap has a peak at 0–1 Mora, and the addressee was considered to start his/her utterance right after the completion of the current speaker's utterance. However, based on the result of this section, Fig. 12 shows that the probability density function of log-converted inter-utterance gaps is approximately equal to a normal probability density function with the mean of 0.05 log Mora (=1.05 Moras) and

the standard deviation of 1.33 log Mora. So the most frequent inter-utterance gap is 1.05 Moras, and the addressee tends to start his/her utterance around 1 Mora after the completion of the current speaker's utterance.

Utterance Units with Overlaps The position of utterance overlap, i.e., the starting position of the addressee's utterance, was measured based on the number of moras from the beginning within the current speaker's utterance (Fig. 13).

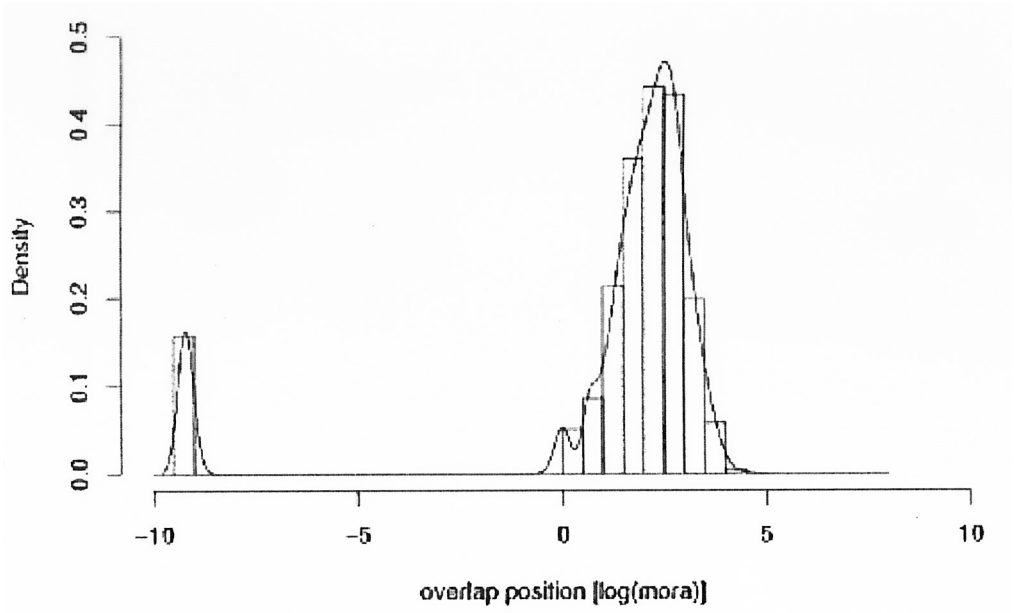

Figure 13. Distribution of Overlap Starting Timing and Its Probability Density Function

As shown in Fig. 13, the probability density function of log-converted mora-based position of overlaps is approximately equal to a normal probability density function with the mean of 2.19 log Mora (=9 Moras) and the standard deviation of 0.81 log Mora. So the most frequent position of overlap occurrence is around 9 Moras. On the other hand, the peak of around -10 log Mora in Fig. 13 corresponds to "addressee's starting his/her utterance at the same time of the current speaker's starting utterance", resulting in 64 occurrences for /hai/ and 212 occurrences for /uN/.

Based on the result of section 2.2.6 (Fig. 10), overlaps tend to occur at the end of the current speaker's utterance. However, the result from the data analysis this time using the large corpus does not explicitly show that overlaps really occur at the end of the current speaker's utterances, though we can tell (1) the mean of the position of overlap occurrence is 2.19 log Mora (=9 Moras) and (2) the mean of the length of the current speaker's utterance is 2.64 log Mora (=14 Moras). So we checked the 4 Moras (=utterance context) from the current speaker's utterances right before overlaps occurred in all the cases of overlaps with /hai/ and /uN/ by the addressee, and then classified all the contexts into the categories in the same way as Table 5 in section 2.2.6.

Table 7. Frequency of Utterance-Contexts

Utterance context of overlaps	Variations	Frequency	Percentage (Frequency/Total)
Context of addressee's /hai/:			
Around the end of the utterance	234	1,353	65%
Prolonged vowels	624	1,022	49%
The beginning of the utterance	262	458	22%
/su/	65	119	6%
Total:	-	2,084	100%
Context of addressee's /uN/:			
Around the end of the utterance	229	1,430	59%
Prolonged vowels	693	1,091	45%
The beginning of the utterance	328	587	24%
/su/	123	294	12%
Total:	-	2,428	100%

Note.

Variations: Variations of utterance context (=4 Moras within the current speaker's utterance right before the overlap)

Frequency: Frequency of each utterance context

Percentage: What percentage of the overlaps belongs to each utterance context. Ratio of overlap occurrences to the total number. A specific utterance context could be categorized into more than one category, so the sum of the total percentage is larger than 100%.

As Table 7 shows, the result of the analysis using the large corpus is similar to the one in section 2.2.6 and the three most frequent positions of overlap's occurrence are as follows:

- The end of the current speaker's utterance
- Prolonged vowels
- The beginning of the current speaker's utterance

2.2.8 Conclusion

As described in Section 2.2.5, we analyzed everyday conversations spoken by two speakers to determine the basic rules of natural discourse timing. In the analysis, the current speaker corresponded to the user in dialog-based systems, and the addressee corresponded to the computer who responds to the user's speech. The results of the analysis confirmed that the addressee basically waits for the completion of the current speaker's utterance, but in some cases the addressee starts his/her utterance before the completion of the current speaker's utterance. In the latter case (=the addressee's starting utterance before the completion of the current speaker's utterance), the addressee starts his utterance at a relevant timing, e.g., the end of current speaker's phrase and prolonging vowels. So it is likely that dialog-based computer systems should also wait for the end of the user's utterance, but can start its own response at some relevant timing.

As described in Section 2.2.6 we analyzed dialog data and clarified the basic rules of utterance timing based on the existence of overlaps between the current speaker's utterance and the addressee's (=next speaker's) utterance. However, utterance timing is the relationship between "the current speaker's utterance" and "the addressee's utterance", so we need to consider the speech rate of both speakers. Therefore we analyzed the addressee's utterance timing based on the current speaker's speech rate.

As a result, we found the following:

- When the addressee waits for the completion of the current speaker's utterance (=without overlaps between the two utterances), the addressee mostly

responds within the time of 0–1 Mora based on the current speaker's utterance rhythm.

- On the other hand, when the addressee starts his/her utterance before the completion of the current speaker's utterance (=with overlaps between the two utterances), it is confirmed his/her utterance starts at the beginning/end of the current speaker's utterance, prolonged vowels and boundaries of phrases.

As described in Section 2.2.7, we analyzed the starting position of the addressee's utterance based on the current speaker's speech rate again using a large-scale spoken dialog corpus, because the data amount used for Section 2.2.5 and Section 2.2.6 was not necessarily enough for statistical analysis.

As a result, we found the following:

- When there is no overlap between the current speaker's utterance and the addressee's utterance, the addressee (=corresponds to the computer within dialog systems) mostly starts at the point of one Mora after the utterance completion of the current speaker (=corresponds to the user within dialog systems).

- When there is an overlap between the current speaker's utterance and the addressee's utterance, the addressee mostly starts his/her utterance at (1) the end of the current speaker's utterance, (2) prolonged vowel or (3) the beginning of the next utterance of the current speaker.

So the basic rules proposed in Section 2.2.5 and Section 2.2.6 were confirmed with the large-scale corpus as well. Also the addressee's utterance start timing with no overlap was made more precise and clarified as "**1 Mora after the utterance completion of the current speaker**".

2.3 Estimating Speaking Rate in Spontaneous Speech from Z-scores of Pattern Durations

2.3.1 Introduction

People commonly express their intentions or attitudes in the form of paralinguistic information encoded in the various speaking styles of everyday conversation [20]. Such paralinguistic information is readily processed by human beings, but automatic interpretation by computer processing is still very difficult. It needs a more rigorous elucidation of the relations between (1) speaking styles and (2) intentions or attitudes. For that purpose, collection and analysis of a large-scale natural-dialog speech corpus is indispensable, but natural dialog data has great variation in articulation styles and it can be both difficult and expensive to obtain accurate transcriptions manually. Although application of speech recognition technology has improved greatly in recent years, the recognition accuracy is still inadequate for natural dialog speech data. Therefore a speaking style analysis method independent of text information is an important sub-goal of our research.

Speech rate is an important variable in the speaking styles used for everyday conversation [24, 25, 26, 27]. Although there has been considerable previous work to try and specify speech rate by e.g., combinations of syllable rate and phone rate [28], there are inconsistencies arising from both acoustic and linguistic information under the influence of e.g., vowel prolongation, devocalization, etc. Speech rate is still difficult to specify by conventional parameters such as syllable tempo or phone duration unless an accurate transcription and segmentation is available.

We therefore propose a method which makes use of sequences of speech-sound patterns which occur five or more times in a speaker's dialog data, based on a large volume of speech information but without use of text information, and we measure speaking rate by means of the z-scores [1] of these pattern durations

[1] Z-score of a raw score x is $z = \frac{x-\mu}{\sigma}$ where:

- μ is the mean of the population;
- σ is the standard deviation of the population.

The absolute value of z represents the distance between the raw score and the population mean in units of the standard deviation. z is negative when the raw score is below the mean, positive when above.

relative to the distribution of its pattern group. This method allows us to analyze changes in speech rate with respect to speaker intentions or speaker state. The method allows us to work with untranscribed speech, processing not only the speech patterns demarcated by pauses, but also the patterns embedded in a sentence utterance.

In addition, this method of pattern extraction processing is efficient as a general preprocessor for speaking style analysis of large scale speech data, since it can provide a basic framework for extraction and analysis of any variable features of dialog speech data, i.e., it is applicable not only to speech rate but also to various acoustic or prosodic features such as fundamental frequency, power, and voice quality, to enable an estimate of local settings normalized per speaker without access to a transcription or labels.

2.3.2 Pattern Extraction Method

Compared to the clean speech of laboratory or studio recordings, in dialog speech, many utterances undergo extreme phonetic modification (e.g., prolongation of vowels or gemination of consonants, elision or deletion of sounds, etc.) and transcription is an expensive and time-consuming process, particularly if it is to be done at the phonetic level. Even if it is to be used as a word-level target for automated segmentation, the written transcription itself is not necessarily suitable as a specification of the speech sounds.

For example, although speech rate in Japanese is usually specified as a period of time divided by the number of mora (or syllables) it contains, when there is insertion or prolongation of vowels or gemination of consonants, for example, it becomes difficult to specify the number of mora.

Although application of speech recognition technology to dialog speech data has improved considerably in recent years, many spontaneous speech utterances are not registered either in the lexical dictionary or the language model. Yet these non-verbal speech noises are common in dialog data, and though they signal important affective information, they also serve as a cause of recognition errors.

It is of course possible to add such non-verbal speech items to the dictionary or language model from existing transcriptions, but we note that this type of dialog speech is very dependent on speakers and speaker-hearer conditions, and

there is no guarantee that we can obtain a sufficient lexical set only by increasing the amount of transcribed data.

On the other hand, if similar types of speech patterns can be extracted or detected automatically from dialog speech, the duration of these similar patterns can be directly compared, and we can make an estimate of speech rate change by sampling the patterns at (irregular) intervals throughout the speech.

Therefore we propose a method for extracting similar speech patterns automatically, based on acoustic information, as a component technology for the analysis of speaking style and other prosodic features that slowly change throughout the course of an utterance.

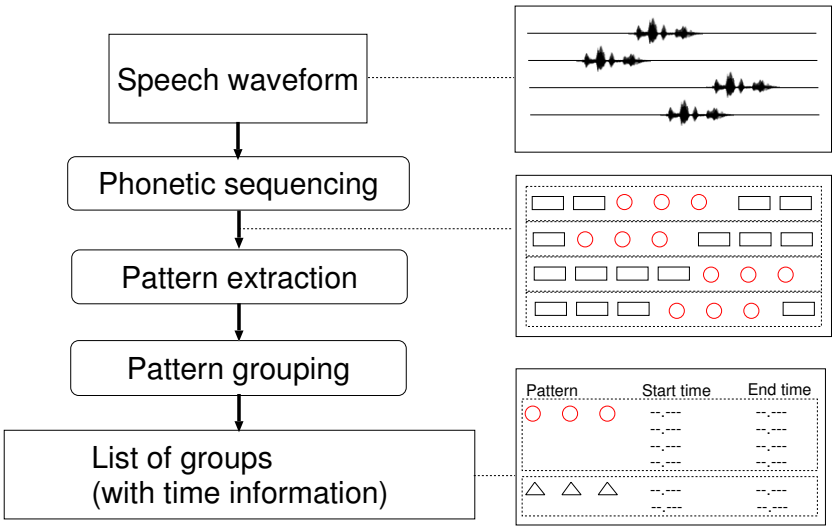

Figure 14. Pattern Extraction Method

Phonetic sequencing Fig. 14 illustrates the method. First, speech waveform information is decoded to phoneme character sequences using a speech recognizer [29, 30](based on the following acoustic features: 12 dimensional MFCC + 12 dimensional Δ MFCC + Δ log power) in the 'phonetic typewriter' mode of the speech recognition engine Julius [31]. An ergodic grammar which permits all 43 phonemes to connect freely to each other is used for the 'recognition'. In this method, the phoneme character sequence is not considered as text information

related to utterance content, but as a mere symbol sequence for carrying out an indexing of the time series patterns of the acoustic features. We do not require speech recognition accuracy based on a correspondence of the recognition result with a transcribed text.

Because previous research [16, 33] has indicated that speaker adaptation in phonemic segmentation based on transcribed text can improve accuracy better than speaker-independent HMMs, we adapted a monophone HMM using a set of phonemically balanced sentences read by the same speaker who provided the dialog speech corpus (see 2.3.3 below).

Pattern extraction Next, all patterns which appeared 50 times or more in the recognition result were extracted using the Multigram Software Package [32] which was designed for statistical natural-language processing.

The patterns were determined by dividing each label-pair sequence into pattern sequences using the maximum-likelihood-based VI-train command of the Multigram Software. The configuration was as follows: up to 10 labels can be considered as forming a pattern (or multigram dictionary entry). Each pattern must occur at least 50 times in the corpus, and the number of iterations for the likelihood estimation and pattern dictionary construction is 10. This process results in a dictionary of label-sequences that commonly co-occur, and an estimate of the occurrence likelihood for each entry.

Pattern grouping The start time and end time are assigned to each speech pattern by post-processing of the recognizer output, and the z-scores of these segment durations are calculated for all tokens in each pattern group.

2.3.3 Experiments

In order to check the validity of the proposed pattern extraction method described in Section 2.3.2 an evaluation experiment was conducted using a subset of the same dialog speech corpus. The validity was evaluated by comparing the patterns obtained by the proposed method (henceforth, 'recognition patterns') without the use of a transcription, and the equivalent patterns obtained from a phonemic segmentation based on the transcribed text (henceforth, 'transcription patterns')

which we assume to be more accurate since they are more expensive to obtain. If the same result is obtained for recognition patterns and transcription patterns, it can be said that the validity of the proposed method is high.

Corpus A section of the spontaneous dialog speech corpus from the JST/CREST-ESP project [20, 21, 22, 23] was used as the speech material .

This corpus has the following features:

- Spontaneous dialog speech over the telephone, recorded using high-quality head-mounted microphone direct to local disc at 44.1kHz.
- A young adult female speaker of Japanese
- Data recorded over period of 2 years(more than 250 hours of speech data)
- Each dialog lasts between 6 and 30 minutes.
- Includes various inter-personal relationships (parents, husband, children, relatives, friends, others)

We used 145,152 utterances from 792 dialogs for the training. Each utterance was determined from a manual transcription and defined as a 'minimal meaningful speech unit'. Processing was performed with the proposed method, based only on speech waveform information alone, and no transcribed text or other information relating to the utterance content was used except for evaluation.

Phonemic segmentation as evaluation criterion For the evaluation, we used similar patterns determined from the phoneme sequence output by a speech recognition engine (Julius [31]) used in alignment mode and fed with a hand-made transcription of each utterance in place of a grammar module.

As mentioned above, we used the VI-train command of the Multigram Package [32] to provide the dictionary of label-sequence patterns and the unit sequences. The settings of HMMs for phonemic segmentation and configuration of pattern extraction were as described in Section 2.3.2.

Because the manually transcribed text is in Japanese orthography (and therefore mixes Chinese kanji characters with the phonetic kana alphabet) we used

the public-domain Japanese morphological analyzer ChaSen [19] to produce a sequence of phonemic characters representing the text as a basis for the automatic segmentation.

Preliminary experiment The number of labels making up the multigram dictionary codebook varied in the range of $1 \leq N \leq 6$. No patterns longer than 6 occurring more than 50 times were extracted.

We found that there were many more fragmentary patterns of 3 phonemes for the recognition patterns, than for the transcription patterns. Moreover, the patterns incorporating 3 labels or fewer correspond to very many speech segments and may be greatly influenced by neighboring phonemes, so the variation of these segments will be great when compared with others of the 'same' pattern. On the other hand, patterns of 5 labels correspond to a smaller number of speech segments of a specific type, and the influence of surrounding phoneme variations can be considered to be comparatively small, so we limited our evaluation to only the changes of speech rate observed within the 5-label patterns.

Evaluation measure In this section, we verify the validity of the pattern extraction results by the proposed method and confirm the speech rate estimation (based on z-scores of pattern durations in the group distributions of each pattern) by comparison with the case of transcription patterns.

First, if there is any overlap between speech segments corresponding to the recognition pattern and segments corresponding to the transcription patterns, then we judge the recognition pattern to be sufficiently aligned or correct. Next, the validity of the proposed method is evaluated at these well-aligned points.

For the correspondingly aligned data, the agreement between the pattern duration z-scores from recognition patterns and those from transcription pattern's was checked. To the extent that both z-scores correspond, the possibility that speech rate can be appropriately measured by the proposed method is confirmed.

Furthermore, correlation between the proposed method and a conventional speech rate which used transcription was confirmed for all the extracted patterns, for there were some recognition patterns which had no transcription pattern counterparts due to mis-recognition.

2.3.4 Results and Discussion

Reliability of speech segment extraction Table 8 shows the ten most frequent patterns of multigram codewords of length five. Many include silence symbols (/sp/ or /q/) from the recognizer, but we selected the interjection /naNka/ for further manual inspection.

Table 8. Patterns of 5 Phonemes, Most Frequent 10 Patterns

Pattern	Number of tokens
<sp>ru:Nq	162
<sp>nu:Nq	161
<sp>uu:Nq	152
kaqte	134
tokoa	133
toqte	118
naNka	*116*
<sp>moNq	107
moqte	104
koqto	99

We found 116 speech segments corresponding to /naNka/ in both the transcription patterns and the recognition patterns. Of these, 81 were correctly aligned (Table 9).

Table 9. Result of Listening Test

Word including pattern	Meaning in English	Frequency
naNka	*interjection*	*96*
naNkai	how many times	16
naNka	question	2
naNka	etc.	1
naNka	something	1
total	–	116

Speech rate based on z-score of pattern duration The larger the z-score of pattern duration, the slower the speaking rate of the corresponding speech

segment can be assumed in the proposed method. Because both distributions of duration from the 116 recognition patterns and transcription patterns are similar to a log normal distribution, duration is computed as z-scores after being converted to logarithmic values. As a result, both distributions of duration patterns are close to normal distribution (Fig. 15).

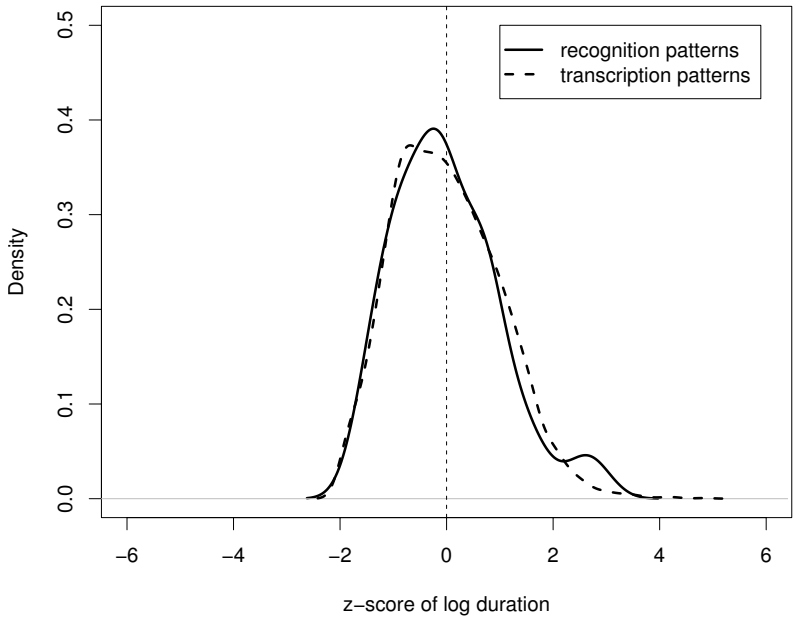

Figure 15. Duration Z-Scores of /naNka/, after Log Conversion

Also as the result of the checks on the 81 correct data in the 116 segments automatically extracted by the proposed method, the duration z-scores appear equal to the duration z-score of the transcription patterns in general. So, it is thought appropriate to predict speech rate based on z-score of recognition patterns obtained by the proposed method (Fig. 16).

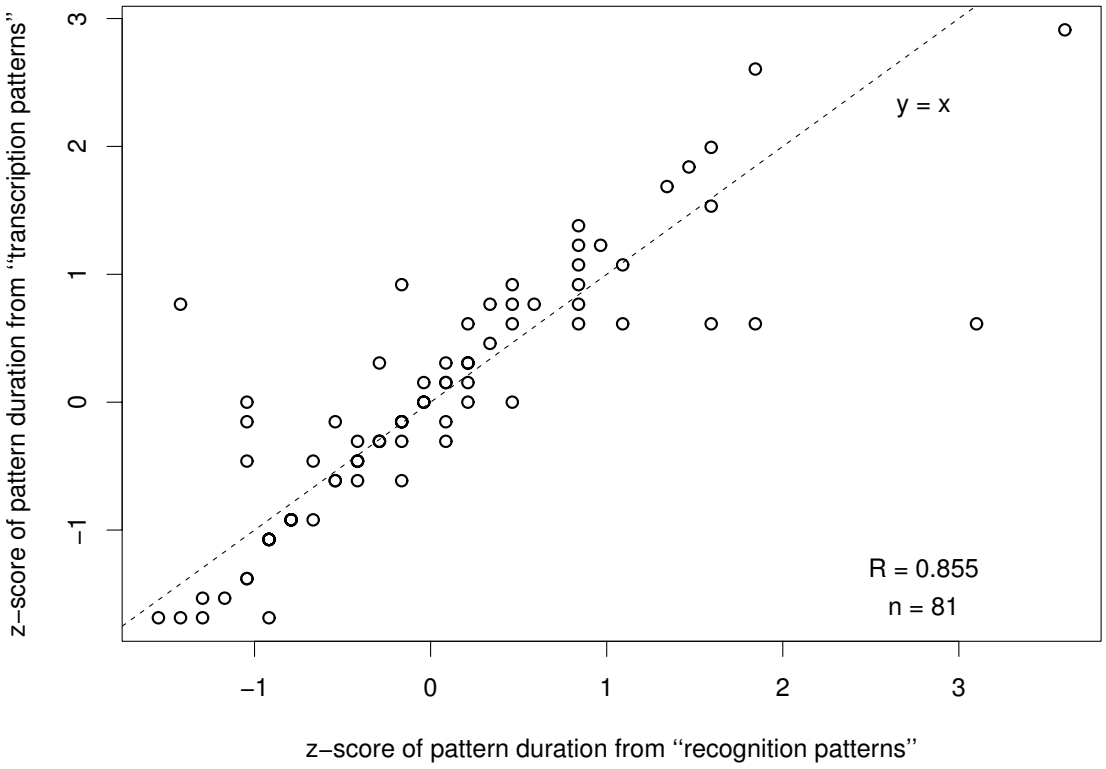

Figure 16. Duration Z-score of Patterns. The ordinate illustrates the z-score in the distribution of the transcription patterns. The abscissa illustrates the z-score in the distribution of recognition patterns.

Correlation between the proposed method and conventional vowel-per-second rate The correlation between the proposed method and vowel-per-second rate [34] is confirmed for all the extracted 1,641 pattern segments including the 116 segments of /naNka/ (Fig. 17).

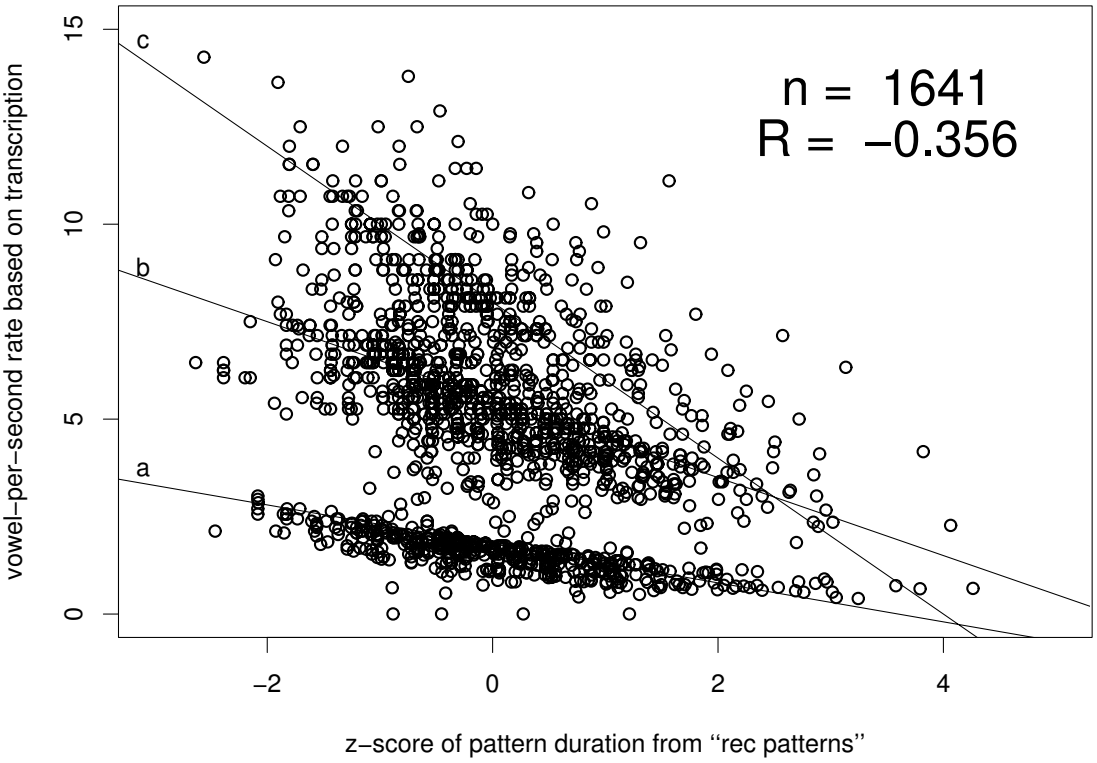

Figure 17. Correlation between Proposed Method and Conventional Method. The ordinate illustrates the z-scores of all the extracted recognition patterns based on the proposed method. The abscissa illustrates vowel-per-second rate of the speech segments corresponding the recognition patterns.

Although overall correlation is R = -0.356, there seems to be three groups of correlation. The correlation computed for each group is comparatively high and patterns in each group has characteristic structure, i.e., number of vowels included in the recognition pattern (Table 10).

Table 10. Three Groups of Correlation

Group	Correspondence formula	Correlation	Vowels
a	$y = -0.5x + 1.8$	-0.693	1
b	$y = -1.0x + 5.5$	-0.551	2
c	$y = -2.0x + 8.0$	-0.592	3

2.3.5 Conclusion

To clarify the relations between speaker's speaking styles and intentions/attitudes, analyzing large-scale natural-dialog speech corpora is indispensable. However, natural dialog data has great variation in articulation styles and it would be difficult and expensive to obtain accurate transcriptions manually. On the other hand, the accuracy of speech recognition is still not perfect for natural dialog speech data analysis though speech recognition technology itself has been improved greatly in recent years. So this section proposed a method to measure speaking rate based on the z-scores of speech-sound pattern sequences automatically extracted from a large volume of speech information.

As a result, we found that the proposed method can detect 81 out of 116 patterns of /naNka/, so it is expected that around 70% of patterns can be extracted automatically. Also the correlation between (1) the speaking rate based on the z-scores of speech-sound patterns extracted by the proposed method and (2) the actual speech duration calculated based on the transcribed data was high (R=0.855). In addition, the correlation between the proposed method and vowel-per-second rate was checked for all the extracted 1,641 pattern segments including the 116 segments of /naNka/. Even though the overall correlation was R = -0.356, it was confirmed that (1) the samples could be categorized into three groups based on "number of vowels in the recognition pattern" and (2) the correlation became much higher after being categorized into three groups based on that criteria.

So there is a possibility the proposed pattern extraction method can be used as a general preprocessor for a variety of speaking style analyses for large scale natural speech data, because this method should be useful to extract speech analysis unit, for not only speech rate but also various acoustic/prosodic features

such as fundamental frequency, power, and voice quality, to enable an estimate of local settings normalized per speaker without manually generated transcriptions. We will continue further evaluation to confirm the usefulness of the proposed method using more pattern segments with more speech features.

2.4 Speech Synthesis using Extended CV as Speech Unit

2.4.1 Introduction

The paradigm of corpus-based synthesis has divided the problems on speech synthesis into the following three separate issues:

1. Acoustic and prosodic variation of speech units within a corpus
2. Unit selection measurement to choose the best speech unit from the corpus
3. Fast and efficient search algorithm to pick the best speech unit

Non-uniform speech unit was proposed for the original Corpus-based Speech Synthesis method [35, 36], and then prosodic non-uniform speech unit was proposed for concatenative synthesis[37, 38] to improve prosody of the synthesized speech. However, there are still some remaining problems such as concatenation noise, unnatural accent, strange rhythm and mixture of needless sound.

2.4.2 Extended CV

To resolve the above problems, we propose Extended CV (CV stands for combination of Consonant and Vowel) as a speech unit in place of phonemes or moras to synthesize more human-sounding speech sound preserving natural rhythm and spectrum dynamism as in human utterances.

Here Extended CV is a sequence of sounds containing a vowel or vowels as its core, which is extracted from the recorded human utterances in order to preserve the dynamism of a vowel sequence (Table 11).

Since Extended CV keeps dynamism of spectra and fundamental frequency of sound and maintains distinctive rhythm of long vowels, diphthongs and short vowels, employment of Extended CV as a speech unit improves the naturalness at the concatenation boundaries of pieces of waveform such as in “vowel-vowel

concatenation", "vowel-semi vowel concatenation" or "a special mora", which has so far had continuity problems[39].

Table 11. Constructions of Extended CV

syllable weight	syllable weight type	construction	examples
1	light syllable	(C)(y)V	ka, sa, ta, na, ha, ma, ya, ra, ... a, i, u, e, o che, pya, ...
2	heavy syllable	(C)(y)VR (C)(y)VJ (C)(y)VN (C)(y)VQ	to:, ya:, kyu:, pyu:, ... kai, ui, pyua, ... kaN, aN, myaN, ... chuQ, ryaQ, jaQ, ...
3 or larger	super-heavy syllable	(C)(y)VRN (C)(y)VRQ (C)(y)VJN (C)(y)VJQ (C)(y)VNQ etc.	che:N, ju:N, a:N, ... u:Q, che:Q, ... saiN, pauN, ... kaiQ, daiQ, ... doNQ, chaNQ, ... etc.

Legend:

C	Consonant (except double consonant, contracted sound and syllabic 'N')
(C)	Consonant (0 or more)
Q	Double consonant
y	Palatal sound
(y)	Palatal sound (0 or more)
N	Syllabic 'N'
V	Vowel (except long bowel and diphthong)
R	chonpu (A long vowel consists of V and R.)
J	The second vowel of diphthong

2.4.3 SPEAKS: GUI-based Speech Synthesizer for Template Synthesis

We implemented a GUI-based speech synthesizer named "SPEAKS" [40, 41, 42] running on WindowsNT/2000 which uses Extended CV as speech unit. SPEAKS's aim is not Text-to-Speech but generating synthesis sound for fixed sentences, e.g., disaster prevention information for public address systems. The synthesis target

to be input consists of template sentences such as typhoon information and several variable phrases such as place name, number and date. Figure 18 shows an example GUI image of SPEAKS for typhoon information.

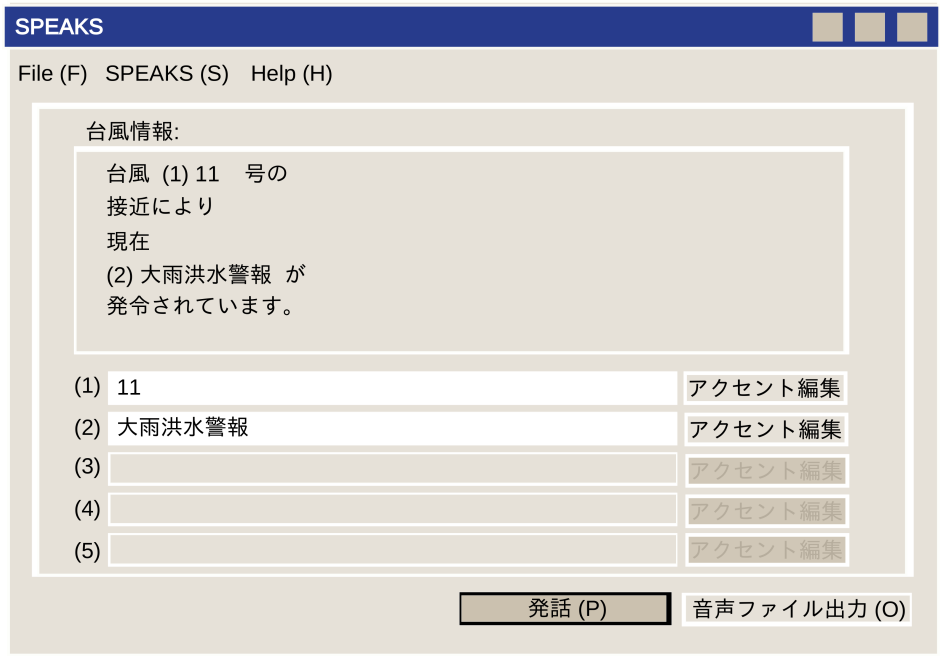

Figure 18. SPEAKS's GUI Image

Before using speech synthesis with SPEAKS, it is necessary to prepare a speech corpus. The first step is dividing prerecorded speech waveforms into Extended CVs and storing them in a corpus. And the next step is adding labels to each Extended CV with their start and end time.

In the process of speech synthesis, SPEAKS first parses the inputted Japanese to be synthesized, and obtains pronunciation, pause and accent information. Next, it selects the optimal units in the speech corpus according to the information, then extracts speech segments corresponding to the units from the corpus. It composes a target speech waveform by concatenating the extracted speech segments.

2.4.4 Generating Variations for Particles

When we need synthesized speech data in combination of large amounts of speech data, e.g., date information, and particles in Japanese, we have to (1) separate the date part and the particle part and insert unnatural short pause or (2) record all the variations with particles, e.g., 50,000 variations for the combination of 10,000 dates and five particles. However, there is a possibility we can obtain high quality synthesized speech without recording all the combinations using SPEAKS.

The procedure to generate particle variations of dates is as follows:

1. Split the synthesis target phrase (=date+article) into the following three parts:

   Head – The rest of the target date without the "Glue" part at the end.

   Glue – The data used to concatenate the date and the particle.

   Tail – The particle to be added to the date.

2. Get the "Head" part from the existing data of dates, and choose the best candidate of the "Glue" part and the "Tail" part from the newly added data.

3. Concatenate all the three parts (=Head, Glue and Tail), and generate the target speech.

In the above synthesis procedure, it is important to precisely choose the "Glue" part and determine the "Head" part based on that. For that purpose, we need to analyze minimum required data to generate the target speech sound based on the phoneme environment using the existing data of dates and the newly added particles from both acoustic and prosodic viewpoints.

Acoustic viewpoint:

Dates in Japanese end with "日", so the last Extended CV of dates is always /chi/ or /ka/. Both of those Extended CVs are voiceless fricative and voiceless plosive, and there is silence right before those sounds in their spectrograms. Therefore both of them (=/chi/ and /ka/) can be used as the "Glue" part (Fig. 19).

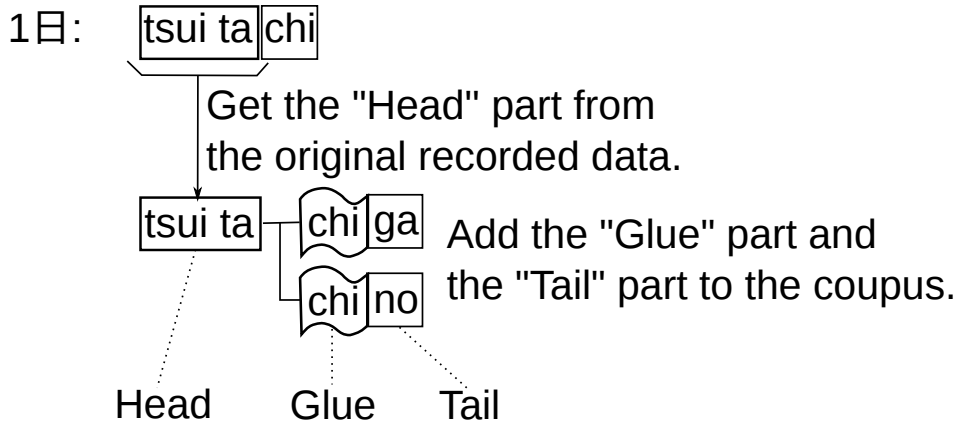

Figure 19. How to Generate Particle Variations

Prosodic viewpoint:

To obtain natural synthesized speech, we need to consider the prosodic environment as well. So the pattern of the fundamental frequency (F_0) is analyzed as listed in Table 12.

Table 12. F_0 Pattern of /chi/ and /ka/ within Date Data

Date		F_0 (*1)	Context (*2)
Glue	Kanji and /Extended CV/		
/chi/	1 日 /tsui ta chi/	H (/chi/)	HHL (/ta chi ga/
	31 日 /saN ju:i chi ni chi/	H (/chi/)	HHL (/ni chi ga/)
	16 日 /ju: ro ku ni chi/	L (/chi/)	LLL (/ni chi ga/)
	19 日 /ju: ku ni chi/	L (/chi/)	LLL (/ni chi ga/)
/ka/	2 日 /fu tsu ka/	H (/ka/)	HHH (/tsu ka ga/)
	3 日 /miq ka/	H (/ka/)	HHH (/miq ka ga/)
	20 日 /ha tsu ka/	H (/ka/)	HHH (/tsu ka ga/)

*1: Whether the F_0 of the "Glue" part is relatively high (=H) or low (=L).

*2: The pattern of the F_0 of (1) the last Extended CV of the "Head" part, (2) the "Glue" part and (3) the beginning Extended CV of the "Tail" part. Determined by whether relatively high (=H) or low (=L).

2.4.5 Conclusion

The speech sound for combination of dates and particles was generated by SPEAKS based on the procedure described in Section 2.4.4, and the quality of the synthesized speech sound reached a practicable level for actual public address systems. So it is expected SPEAKS can be used for template synthesis, i.e., synthesizing the variable phrases such as dates and embedding them into fixed sentences such as disaster prevention service.

We tried to synthesize the combination of place names and particles. However, there are so many kinds of place names and it is difficult to analyze the acoustic/prosodic variations manually. So we will continue data analysis and clarify the model to automatically calculate the required data for a specific target case, e.g., place name and company name.

2.5 Precise Timing Management using Real-time OS

2.5.1 Introduction

To implement an advanced discourse timing management system based on (1) the basic rules of utterance timing and speech rate clarified in Section 2.2 and 2.3 and (2) the new speech synthesis method proposed in Section 2.4, it is essential to control the start timing of computer's speech precisely based on user's utterance timing and speech rate. However, it is actually impossible to control utterance start timing precisely, e.g., by msec, if we use ordinary operating systems such as Linux and Windows, because these operating systems process multiple tasks at once simultaneously and the CPUs of the systems encounter constant interruptions. So this section proposes using ART-Linux[43, 44, 45, 46], an extension of Linux to make it a real-time operating system, and sees the feasibility of precise timing management.

2.5.2 Experiment on Precise Timing Control using ART-Linux

To obtain the feasibility of precise timing management, we implemented a sample program written by C language which generates 1 ms period loop and checked the actual time of one clock period generated by both ordinary Linux (Vine Linux 2.1

with Kernel 2.1.17) and ART-Linux (applied the ART-20010321 patch to Kernel 2.1.17) on a PC with a CPU whose clock frequency is 645 MHz (Intel Pentium III) and 128 MB RAM.

In order to see the effect of CPU load, we ran the sample program in two different configurations: (1) with no specific CPU load and (2) with big CPU load by kernel compilation.

Note that the clock frequency of the CPU used for the experiment is 645 MHz, so 645, 000, 000 clock cycles correspond to one second. Therefore 645, 000 clock cycles correspond to 1 ms (= 1/1000 sec).

2.5.3 Result and Discussion

The results of the experiment (Fig. 20 and Fig. 21) show that (1) it is impossible or extremely difficult to handle precise utterance timing based on 1 ms period using ordinary Linux but (2) it is probably possible using real-time OSs such as ART-Linux.

2.5.4 Conclusion

To implement an advanced discourse timing management system based on (1) the basic rules of utterance timing and speech rate clarified in Section 2.2 and (2) the new speech synthesis method proposed in Section 2.4, it is essential to control the start timing of computer's speech precisely based on user's utterance timing and speech rate. However, it is actually impossible to control utterance start timing precisely, e.g., by msec, if we use ordinary operating systems such as Linux and Windows, because these operating systems process multiple tasks at once simultaneously and the CPUs of the systems get interrupted constantly.

So this section proposed using ART-Linux, an extension of Linux to make it a real-time operating system, and saw the feasibility of precise timing management. As a result, we confirmed that it is impossible or difficult to handle precise utterance timing control based on 1 ms period using ordinary Linux but it is possible using real-time OSs such as ART-Linux.

Note that ART-Linux has the following key features, so it is easier to apply to existing computer systems than other real-time OSs:

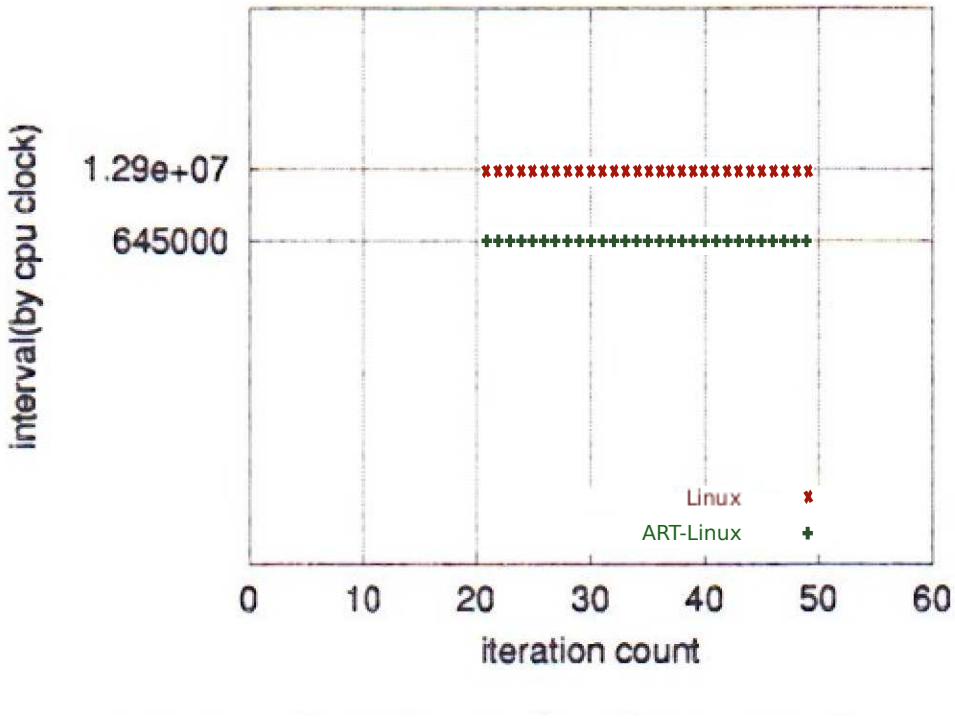

Figure 20. Precision of Time Control without CPU Load. With ordinary Linux (red 'x'), the observed interval was around 20 ms (= 12,900,000 clock cycles) though 1 ms (= 645,000 clock cycles) interval was specified. On the other hand, with ART-Linux (green '+'), the observed interval was very stable and always 1 ms (= 645,000 clock cycles) as specified.

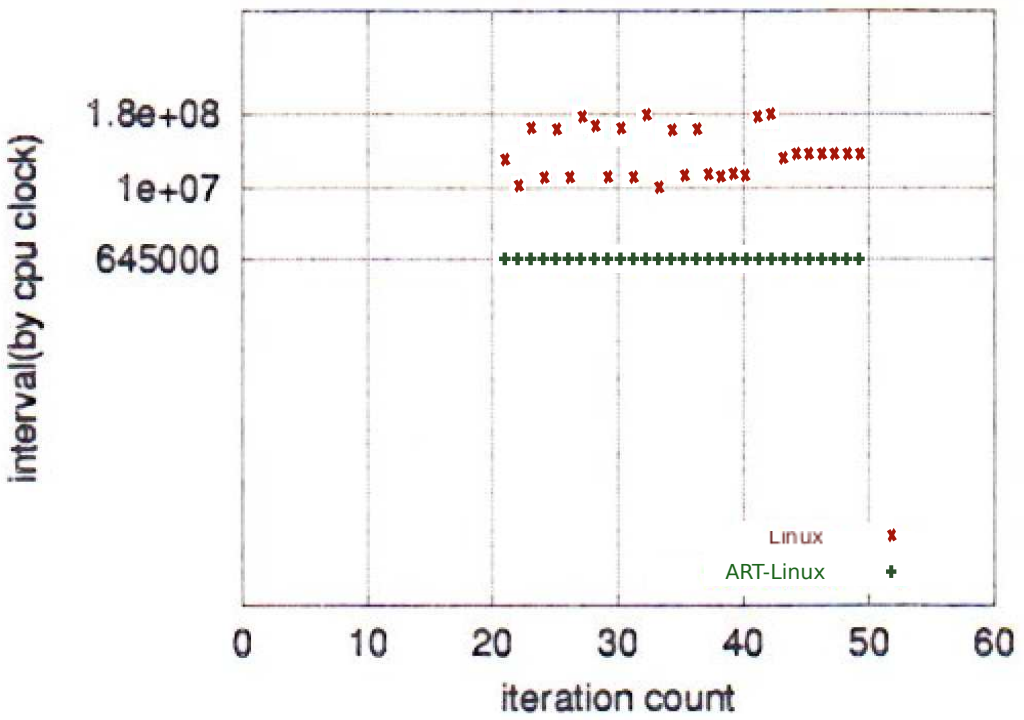

Figure 21. Precision of Time Control with CPU Load. With ordinary Linux (red 'x'), the observed interval varied between 279–16 sec (= 180, 000, 000–10, 000, 000 clock cycles) though 1 ms (= 645, 000 clock cycles) interval was specified. On the other hand, with ART-Linux (green '+'), the observed interval was very stable and 1 ms (= 645, 000 clock cycles) as specified.

- Existing drivers and applications for ordinary Linux are reusable.
- Real-time tasks can be run with regular user privileges.
- Real-time tasks can be described using three simple APIs (in case of 'ART-20010321').

2.6 Conclusion

In order to improve discourse timing management and make dialog-based computer systems even more friendly and useful, Section 2.2 analyzed utterance timing from several viewpoints and clarified a basic heuristics for discourse timing management which could be used for **Speech Timing Manager** (Fig. 2). Then Section 2.3 proposed a speaking style analysis method independent of text information for **Feature Extraction** (Fig. 2), because sometimes it is difficult and/or expensive to obtain accurate transcriptions based on natural dialog data automatically (=by speech recognition) or even manually (=by human labellers) due to its great variation in articulation. Section 2.4 proposed a new speech synthesis method which uses "Extended CV" as the speech unit to improve the rhythm of synthesized speech for **Re-sequencing** (Fig. 2). Section 2.5 discussed a possible framework to manage discourse timing precisely using a real-time OS for **Hard RT Manager** (Fig. 2).

The result of various analyses in Section 2.2 clarified the basic heuristics of discourse timing management, i.e., "**The addressee waits for 1 Mora after the utterance completion of the current speaker to start his/her own utterance.**", and it is expected that the heuristics can be used as the basis of the possible discourse timing management model. On the other hand, the result implied the need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of one Mora (around $100 - 200$ ms). However, it is impossible to control utterance timing precisely using ordinary operating systems such as Linux and Windows, because these operating systems process multiple tasks simultaneously and the CPUs of the systems encounter constant interruptions. Therefore it was proposed to use a real-time OS for discourse timing management in Section 2.5, where the feasibility of precise timing

management was confirmed. Note that it depends on the use cases and situations whether or not precise timing management is of actual need, so the point here is not "**simply responding to the user quickly**" but "**stability of response timing handling mechanism based on the possible timing control model**". A possible situation which requires this kind of precise and stable timing management is interaction with a distant device, e.g., a GPS satellite or a sensor in a South American country from Japan.

We will continue to analyze more conversation data in various situations from the viewpoint of acoustics, linguistics, utterance timing and speech rate to clarify the detail of the discourse timing model to implement an automatically controlled system shown in Fig. 2. As analyzed in Section 2.2, the addressee tends to wait for appropriate timing to start his/her own utterance and continue the communication initiated by the speaker smoothly. However, sometimes utterance overlaps occur. Therefore it is recognized that the discourse timing management model would have the following features to handle both of these cases (=smooth transition and overlapping):

- The discourse timing management model consists of two independent components for two speakers.
- The model basically allows both speakers to take their turns alternately.
- The model allows the speakers to start their utterances before the counterpart's utterance ends or even at the beginning of the counterpart's utterance in some cases.

In the previous research in 2002 [47], the author proposed a two-cylinder engine model shown in Fig. 22 as a possible model that made two independent pistons, which are loosely coupled with each other using a connection rod, repeat the procedure described in Table 13 and collaboratively maintain communication (=the rotation of the connection rod and the crank).

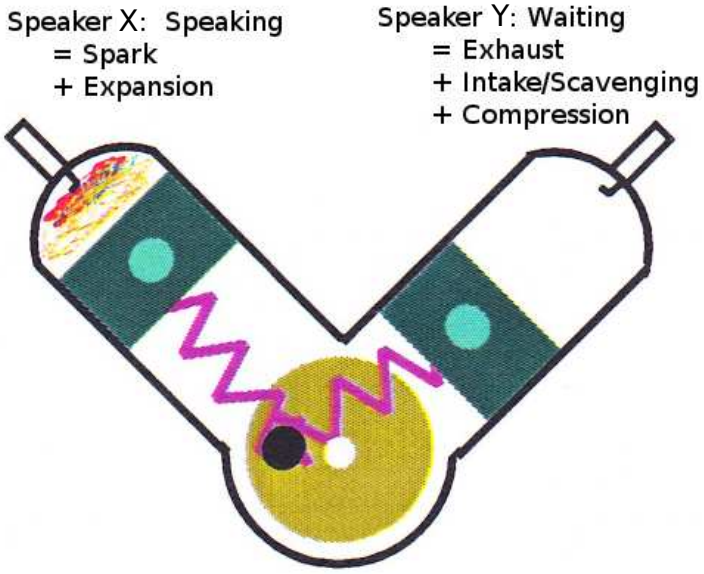

Figure 22. Two-Cylinder Engine Model of Discourse Timing

Table 13. Two-Cylinder Model

Actions	Current Cylinder/ Current Speaker (Speaker X)	Next Cylinder/ Next Speaker (Speaker Y)
Actions within the Two-cylinder Model	Spark + Expansion	Exhaust + Intake/Scavenging + Compression
Corresponding actions for Discourse Timing	Speaking	Preparing for Y's Own Utterance

Note:

- To reproduce the variations of human utterance start timing, the connection rod which connects both the cylinders (=speakers) should not be a hard one for usual car engines but more like rubber or spring so that it can express the fluctuation of discourse timing caused by various reasons.
- The dialog speech corpus for the data analysis in Section 2.2.5 ("**One-pair Data**") had the tendency of " **Speaker X (22 year-old male) initiates the topics whereas Speaker Y (42 year-old male) gives responses.**" So we can expect clarified rules of these tendencies and create a discourse timing model that reflects the roles of (1) the speaker (=the user) and (2) the addressee (=the system) as we analyze more data.

3. Web Technology and International Standardization by W3C

3.1 Introduction

The Web is used as a huge information space by all throughout the world, so interoperability, internationalization, multimodality and accessibility are its key features. Therefore global standardization is ongoing by W3C [48] and other standardization organizations.

This chapter summarizes Web technology first, then describes international standardization of Web technologies by W3C. This chapter describes "HTML5", a.k.a. "the Open Web Platform", as a promising platform for developing interactive Web applications. It also reports the latest progress of the HTML5 specification which is the core specification of the Open Web Platform. Finally this chapter describes the existing issues on Web application development.

3.2 Evolution of Web Technology

Web browsers were originally developed to access information on the Web, and Hypertext Markup Language (HTML) [1, 2] has been used as the standardized way to describe Web pages and Web applications so that page designers and application developers can generate Web pages and Web applications easily without caring of the capability of each terminal device or each browser software. However, the content of the Web has diversified much with time, and multimedia contents have increased rapidly. So it is getting more and more difficult to describe Web applications using competitive ways.

For example, according to the statistical research report on WWW contents by the Ministry of Internal Affairs and Communications [49], the number of graphics files account for 66.3% of all the files on Web servers with JP domain as Fig.23 shows. On the other hand, the data amount of multimedia contents including video (29.1%), graphics (25.4%) and audio (12.1%) accounts for 66.6% as Fig.24 shows.

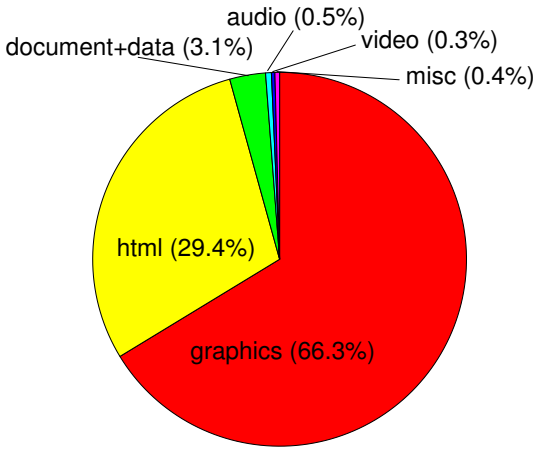

Figure 23. File Number Ratio per Media Type

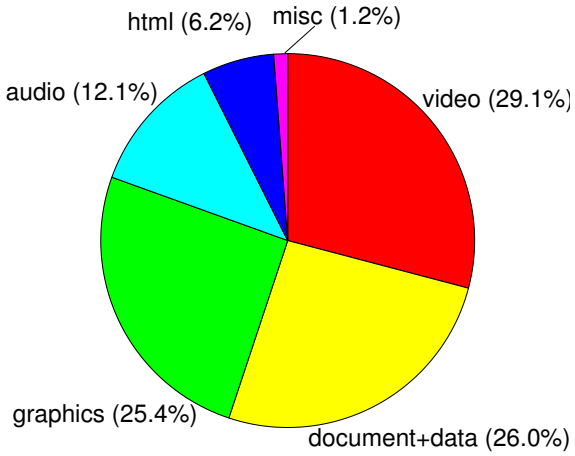

Figure 24. Data Amount Ratio per Media Type

Web contents are getting more and more dynamic and interactive as shown in Fig.25 being used for various applications, e.g., bookstores, game environments and video streaming services. So now the Web is becoming a platform for such Web Applications.

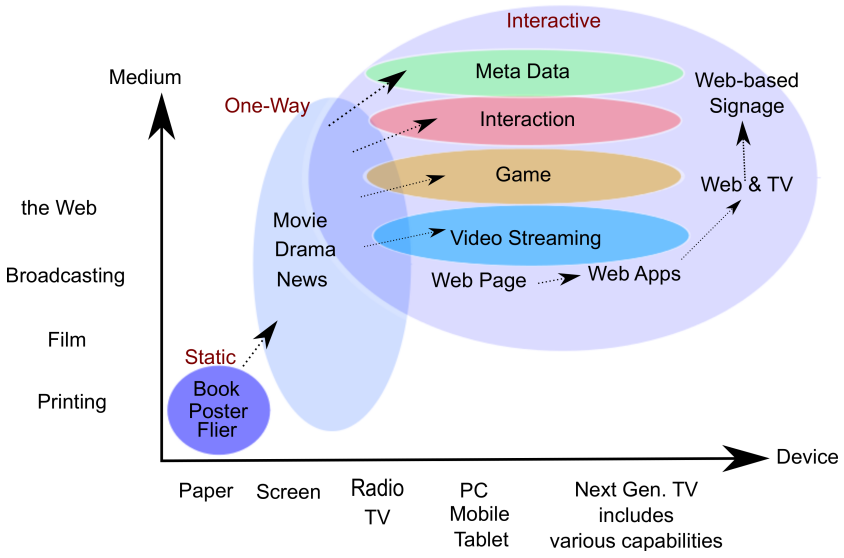

Figure 25. Transition of Media and Devices

The change of Web content usage is in accordance with the movement of the so-called "Web 2.0" which was originally mentioned by Darcy DiNucci in 1999 as follows [50]:

Web 2.0

The Web we know now, which loads into a browser window in essentially static screenfuls, is only an embryo of the Web to come. The first glimmerings of Web 2.0 are beginning to appear, and we are just starting to see how that embryo might develop. The Web will be understood not as screenfuls of text and graphics but as a transport mechanism, the ether through which interactivity happens. It will [...] appear on your computer screen, [...] on your TV set [...] your car dashboard [...] your cell phone [...] hand-held game machines [...] maybe even your microwave oven.

As Darcy predicted, the movement of "Web 2.0" changed the Web from a simple framework for one-directional content distribution to a platform for interactive applications.

On the other hand, Web content developers has been using proprietary techniques such as plug-in's in order to follow the rapid change of Web usage, but unfortunately that avoids world-wide deployment of Web contents. Therefore the standardization of HTML5 [51, 52] as a new generation Web application description language started in 2008, and now Web applications can run on not only PCs but also various devices including smartphones such as iPhone and Android. In addition, the capability of mobile devices and gaming devices is much improved recently, and various Input/Output modalities, e.g., handwriting recognition using touch panels, gesture recognition using video cameras and sensors, and speech recognition using microphones, are available now.

3.3 "HTML5" and the Open Web Platform

3.3.1 History of HTML

Tim Berners-Lee invented the World Wide Web, an internet-based hypermedia initiative for global information sharing while at CERN, the European Particle Physics Laboratory, in 1989. He wrote the first Web client and server in 1990. The first publicly available description of HTML, "HTML Tags", was published in 1991.

In 1993 Berners-Lee and Dan Connolly submitted the first HTML specification (so-called "HTML 1.0") [53] to the IETF, and in 1994 the IETF created an HTML Working Group which in 1995 completed "HTML 2.0" (RFC 1866) [54] as the first HTML specification intended to be a standard for Web browser implementation. In 1994 W3C was formed to fulfill the potential of the Web through the development of open standards [55], and in 1996 W3C formed the HTML Editorial Review Board to help with the standardization process of HTML.

Thus, the following HTML specifications were published as W3C Recommendations:

- HTML 3.2 [56] in January, 1997
- HTML 4.0 [57] in December, 1997

- HTML 4.01 [58] in December, 1999
- XHTML 1.0 [59] in January, 2000

On the other hand, in 2004 HTML5 development began within the Web Hypertext Application Technology Working Group (WHATWG). HTML5 became a joint deliverable between WHATWG and W3C and was published as a W3C Working Draft [60] in 2008. Please note that W3C maintains a document on the details of the differences between HTML5 and previous HTML versions, "HTML5 differences from HTML4" [61].

3.3.2 HTML and Plug-in's

HTML [57, 2] is used to describe documents on the Web, but the need for advanced multimedia contents such as audio and video is increasing in addition to competitive text information. Web browser extensions called "plug-in's" have been developed and used widely and commonly to handle multimedia contents.

However, implementations of plug-in software vary from vendor to vendor, and standardization of various plug-in's is very difficult. So developers and users must learn how each plug-in works causing a burden. For example, when we access a Web page which includes multimedia content such as audio and video, we need an appropriate plug-in such as Flash or Silverlight. However, such necessary plug-in's are not always installed on the user's PC or smartphone. If the necessary plug-in's are not installed, the expected multimedia contents cannot be played, with a warning like "The required plug-in is not installed!" appearing. Even if the required plug-in is installed, it might not work if the plug-in does not match with the specific version of the Operating System or the Web browser in use. In the worst case, the Web browser might suddenly abort right after the user accesses a Web page which requires the plug-in.

On the other hand, there is a difference in the CPU power and display resolution between PCs and mobile devices. So if we want to display a specific Web content appropriately on various types of devices, we actually have to create multiple Web pages for each device, e.g., using smaller images than PCs or sometimes even alternative text for small mobile devices, causing much cost for producing Web contents.

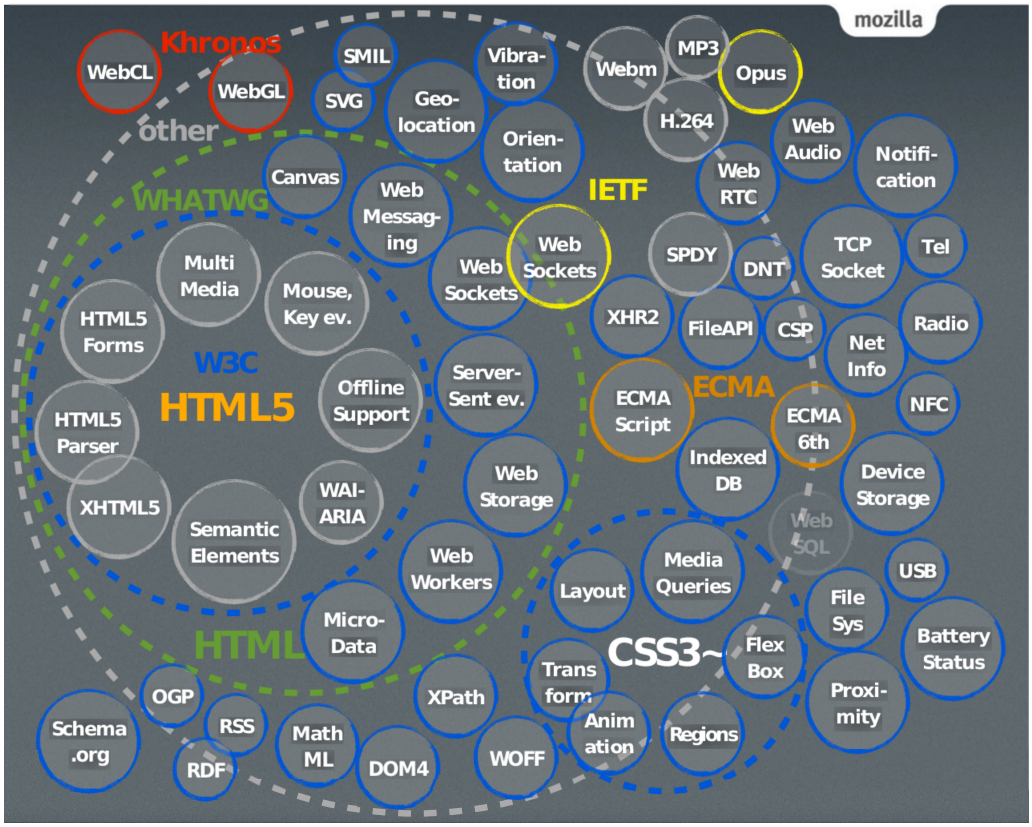

Figure 26. Open Web Platform Specifications (cited from Mozilla Japan dynamis's slides)

3.3.3 "HTML5", a.k.a. the Open Web Platform

Recently "HTML5" is used as the standard platform for richer Web applications which resolves the issues on usage of plug-in's and the burden of developers and users mentioned above. "HTML5" has strong capability of handling multimedia contents such as audio/video and interaction between the Web and the users in addition to basic GUI features of old versions of HTML, so it is used as a smarter platform for richer Web applications.

Some of the new HTML5 features are listed below:

- Video [62] and Audio[63]: playing video and audio files without any plug-in's

- Canvas [62]: 2-Dimensional image drawing
- Drag & Drop [62]: moving and editing images and text
- WebSocket [64, 65, 62]: real-time and full-duplex connection for interactive applications
- Web Storage [62]: storing data within the Web browser on the local client device
- Web Workers [62]: multi-process programming on the Web browser

Here please note that the term HTML5 is used with double quotations (“ and ”) and appears as “HTML5”, because the term “HTML5” is generally used to mean “latest Web technologies in general” and it is vague what the term really means. So W3C encourages to use term HTML5 as “the HTML5 specification” [51, 52], and use “the Open Web Platform” as the term for whole the Web technologies related to the HTML5 specification including CSS and various JavaScript APIs (Fig. 26). Therefore here the term HTML5 is used as “the HTML5 specification”, and the term “the Open Web Platform” is used as whole the Web technologies related to the HTML5 specification in this dissertation.

3.3.4 Latest Status of the HTML5 Specification

The relationship between W3C and WHATWG More than 500 experts throughout the world participate in the discussion on the HTML5 specification [51, 52] within the W3C HTML WG, collaboratively working with WHATWG [66]. In June, 2012, WHATWG made an announcement [67] on the relationship between the WHATWG HTML living standard and the W3C HTML5 specification saying:

- The WHATWG effort is focused on developing the canonical description of HTML and related technologies, meaning fixing bugs as we find them, adding new features as they become necessary and viable, and generally tracking implementations.
- The W3C effort, meanwhile, is now focused on creating a snapshot developed according to the venerable W3C process.

On the other hand, W3C also made an announcement on its official Blog [68] saying:

- First, we announced that Adobe, Google, and Microsoft have provided significant funds to sponsor more complete W3C staff coverage to achieve Recommendation Status for HTML5 for 2014.
- Second, the chairs announced people from the community chosen to participate in the editorial team to complete HTML5.

These announcements were made based on the draft plan for HTML5 stabilization [69] which was announced in late April, 2012, and there are no changes on the partnership between W3C and WHATWG on the follow-on work.

Plan 2014 In October 2012, "Plan 2014" [70], the concrete plan for making the HTML5 specification a W3C Recommendation in 2014, was announced by W3C. In this document, the following was proposed:

- All the features of HTML5 which cannot reach the Recommendation stage by the end of year 2014 will be included in "HTML 5.1" instead, and are expected to become a W3C Recommendation in 2016.
- Addition of new features will be done not by extending the HTML5 specification itself but by adding description to "the Extension Specifications" which are split from the HTML5 specification as modules.

However, the feature descriptions within "the Extension Specifications" could be merged again with the main HTML5 specification depending on the necessity and stability of the description. It was confirmed that the W3C HTML WG will continue the standardization work based on the Plan 2014 at the WG's Face-to-Face meeting during the W3C Technical Plenary and Advisory Committee Meetings 2012 (TPAC 2012) [71, 72] in Lyon, France in November, 2012.

3.4 Issues on Web Application Development

Recently the capability of mobile devices including mobile phones has been much improved. For example, CPUs of such devices have become very powerful and

various Input/Output modalities including speech and gesture are available in addition to competitive GUI along with fast network connection. Consequently, there are several Web applications with advanced natural spoken dialog interfaces emerging in the market, e.g., Google Search[3], Apple's Siri[4] and NTT Docomo's Shabette-Concier[6, 7]. Multimodal features of these applications provide a highly flexible and data-rich alternative to GUI-only or voice-only applications that preceded those multimodal ones. However, how to implement advanced multimodal applications varies from vendor to vendor, and standardization of those applications has been in demand but very difficult to apply. Even though HTML5 is adding various JavaScript APIs for device control, it is still difficult for developers to author multimodal Web applications, because HTML5 does not provide ways to extend Input/Output modalities and developers have to implement concrete modality components themselves. This in turn becomes a burden to extend applications' capability, e.g., adding Input/Output modalities and cloud services to the existing applications.

Typical difficulties for multimodal Web application development are described in the following sections.

3.4.1 Difficulties with Defining APIs for Various Devices

These days HTML5 and related Web technologies provide very powerful capability and are setting a platform for richer, dynamic and interactive Web applications. However, there are several known issues. For example, extension of the markup language's capability using XML's namespace mechanism like the previous version of HTML, i.e., XHTML 1.1, is not applicable with the HTML serialization of HTML5. There have been discussions about that and some of the existing markup languages, e.g., Scalable Vector Graphics (SVG) [73] and MathML [74], have been integrated with the HTML5 vocabularies. More discussions are being held on handling devices such as mobile phones, tablets and digital TVs, and there are plans to define additional JavaScript APIs for that purpose. However, it would be difficult and almost impossible to define specific APIs for various devices and systems used world wide. In contrast, different devices may have the same capabilities such as speech processing or image processing, so it would be redundant if APIs were defined based on devices or device types. So it

is required to consider how to classify devices and modalities, and how to define APIs.

3.4.2 Difficulties with Integration of Distributed Services

These days the capability of mobile devices is much improved thanks to high performance CPUs, high-speed network connection, etc. "Mashup" [75], a technique to create a Web application by combining existing multiple Web services such as map search and speech recognition using simple APIs, is getting more and more popular. Mashup is a quick, easy way to integrate multiple services to create a Web application, but actual service APIs vary from vendor to vendor, and standard framework to control various devices and services is needed.

3.4.3 Difficulties with Dynamic Selection of Multiple Modalities

The type of user interface modalities would be comfortable and appropriate depends on the users' characteristics and environment. For example, as a safety precaution when driving, we need to avoid the interference of driver's concentration. So a standard mechanism for modality management and data transport is required in order to select user interface modalities dynamically based on users' characteristics and environment. In addition, it would be preferable and useful if the mechanism allowed us to combine multiple modalities transparently without the care for details on which modality is implemented on which device.

3.5 Conclusion

The Web is used as a mass information space for all throughout the world, so interoperability, internationalization, multimodality and accessibility are its key features. Therefore global standardization is ongoing by W3C [48] and other standardization organizations.

This chapter summarized Web technology first, then explained international standardization of Web technologies by W3C. Following, this chapter described "HTML5", a.k.a. "the Open Web Platform", as a promising platform for developing interactive Web applications. It also reported the latest progress of the HTML5 specification which is the core specification of the Open Web Platform.

Finally this chapter described the existing issues concerning Web application development.

4. Standardized Multimodal Web Application Framework

4.1 Introduction

As described in Chapter 3, W3C works on international standardization of HTML5 [51, 52], and HTML5 is adding various JavaScript APIs for interactive Web applications. However, how to implement advanced Web applications varies from vendor to vendor, and it is still difficult for developers to author rich multimodal Web applications, because HTML5 does not provide ways to extend Input/Output modalities and developers have to implement concrete modality components themselves. This causes a burden when extending their applications' capability.

The author has been working at W3C as the W3C Activity Lead and holding various discussions on how to solve the issues on Web application development with experts from all over the world since 2005, and concluded the need for a Web-based standard framework and toolkit to develop rich multimodal Web applications. Therefore the author proposed a standard library named "MMI over WebSocket (MoW)" based on the W3C MMI Architecture specification [8] to (1) the Information Processing Society of Japan (IPSJ), (2) the Japanese industry (mainly the W3C Japanese Members) and (3) the W3C Multimodal Interaction Working Group (MMI WG) [11] so that developers could handle a variety of Web contents and Input/Output modalities regardless of their skills. The author also implemented MoW as a JavaScript library and evaluated its usability from the viewpoint of processing speed and server load. MoW has made it possible to implement easier-to-use computer interfaces for various users in various environments.

On the other hand, resulting from the various natural conversation analyses in Section 2.2, a basic heuristics of discourse timing management, i.e., "**The addressee waits for 1 Mora after the utterance completion of the current speaker to start his/her own utterance.**", was clarified. The result implied the need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of 1 Mora (around $100 - 200$ ms). So Section

2.5 proposed the use of a real-time OS for precise and stable timing control. This chapter also proposes using WebSocket as the protocol for fast connection between devices instead of HTTP, which is ordinarily used as the protocol for Web-based applications, because WebSocket connection is expected to be much faster than HTTP connection.

MoW is the first attempt in the world to provide a standard library for multimodal Web applications, so it is planned for public release to globally assist application developers.

4.2 Author's Contribution and Section Structure

Fig. 27 summarizes the author's contribution on the standardized multimodal Web application framework discussion described in this chapter. It also describes the correspondence between each contribution and the sections in this chapter.

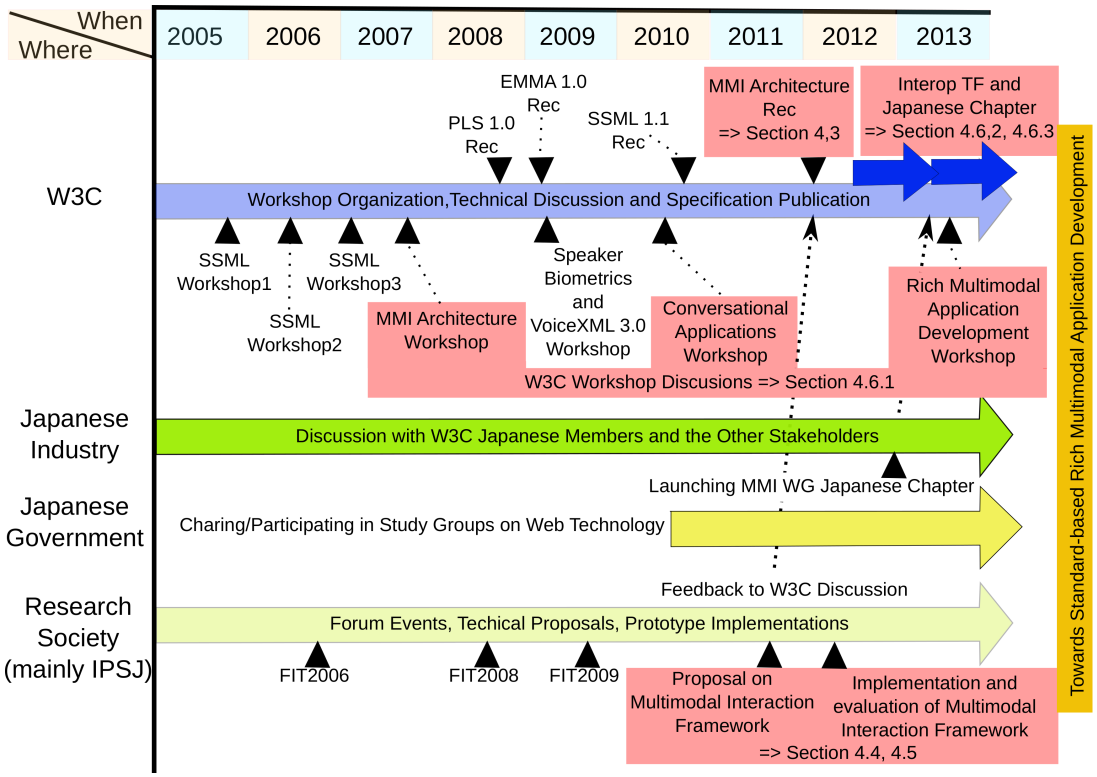

Figure 27. The Author's Contribution

The rest of this Chapter has the following structure:

Section 4.3 explains the importance of a standard framework to integrate multiple modalities and services, and illustrates the W3C MMI Architecture specification [8, 9, 10] as one of the promising candidates to solve the issues. It also mentions several existing and possible applications based on the architecture.

Section 4.4 describes a JavaScript library named "MoW" which is proposed by the author and provides a standard way for interfaces between various Input/Output modalities and Web services based on the W3C MMI Architecture specification [8, 9, 10] so that developers can handle a variety of Web contents and Input/Output modalities regardless of their skills. MoW also provides a smarter way to integrate HTML5 browsers as GUI modalities for advanced multimodal Web applications.

Section 4.5 evaluates the usability of the proposed method from the viewpoint of processing speed and server load, because there is a need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of 1 Mora (around $100-200$ ms) based on the result of Section 2.2, with the requirements of high-speed connection between devices. The proposed library might cause some latency despite the convenience.

Section 4.6 explains the role of the author within the MMI WG, and overviews the current status and the future plan of the international standardization of the proposed method.

Section 4.7 finally summarizes this chapter.

4.3 W3C MMI Architecture as One of the Promising Solutions

There have been several issues on developing multimodal Web applications which have caused a burden for developers. To solve the issues, W3C has been trying to define standard JavaScript APIs, e.g., Speech APIs and Audio APIs, for additional Input/Output modalities on the client side (=Web browser side). However,

as mentioned in Section 3.4, the issues on (1) defining APIs for various devices, (2) integration of distributed services and (3) dynamic selection of multiple modalities have not yet been resolved by the client-side API approach.

The W3C Multimodal Interaction Working Group (MMI WG) has been working on the W3C Multimodal architecture and Interfaces (MMI Architecture) specification [8, 9, 10] as the standard interface between server-side and client-side in order to integrate various Input/Output modalities and Web services. The author has been leading the standardization work of the group as the Activity Lead collaboratively with the group Chair.

At the early stage of the discussion on multimodal interfaces, a mechanism named "XHTML+VoiceXML" (X+V [76, 77]), the combination of (1) a GUI described by XHTML and (2) a speech interface described by VoiceXML [78, 79, 82, 80, 81], was considered as typical multimodal interaction. However, these days consumer electronics devices including digital TVs, gaming devices and air conditioners as well as mobile phones and tablets can be used as terminals for multimodal Web applications. So MMI Architecture has been discussed as one of the possible solutions of standardization to integrate various Input/Output modalities in a network-transparent manner.

As shown in Fig. 28, MMI Architecture handles a combination of various user interface modalities, e.g., speech interface and handwriting input in addition to ordinary GUI. Multiple modalities can be combined and usable transparently without attention to detail on how/where the modality software is implemented/installed if the system is developed based on the architecture.

When discussing the design of MMI Architecture, the MMI WG referred to several existing models for spoken dialog systems including (1) Galaxy Communicator [83, 84], an open source hub-and-spoke architecture, and (2) Model-View-Controller (MVC) model [85], a well-known design pattern for user interfaces. Because MMI Architecture allows nested structure, the architecture resolves the problem of the "hub-and-spoke model", where the hub's functionality as the controller becomes too complicated when there are too many modality components in the system. In addition, the separation of Model, View and Controller makes it much easier and flexible to add or remove modality components dynamically.

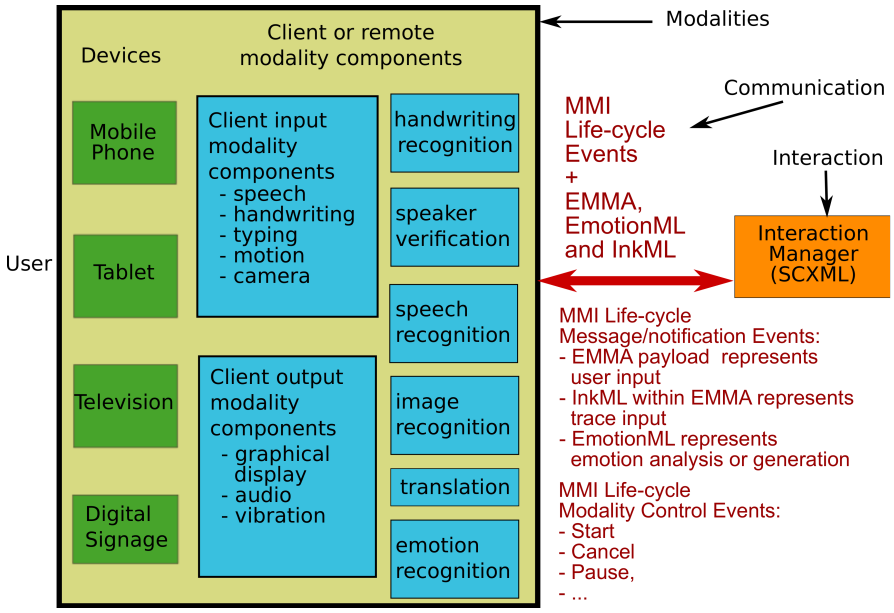

Figure 28. The W3C MMI Architecture

4.3.1 Constituents of MMI Architecture

As Fig. 29 shows, MMI Architecture has the following constituents:

Runtime Framework (RF): The working environment of MMI Architecture which provides the basic infrastructure and enables communication among the other constituents.

Modality Component (MC): Software on the client-side which controls some specific Input/Output modality (or modalities) on the client-side devices such as (1) Web browsers which processes HTML and (2) Voice browsers which processes VoiceXML. The "View" part (=Input/Output modality) in the MVC model.

Interaction Manager (IM): Software on the Server-side which manages the application logic and data for the application. The "Controller" part in the MVC model. State Chart XML (SCXML) [86] is proposed as the application logic description language for IM by the MMI WG.

Data Component (DC): Common data model for the application. The "Model" part in the MVC model. EMMA [87] is proposed as the generic data format for multimodal Web applications by the MMI WG.

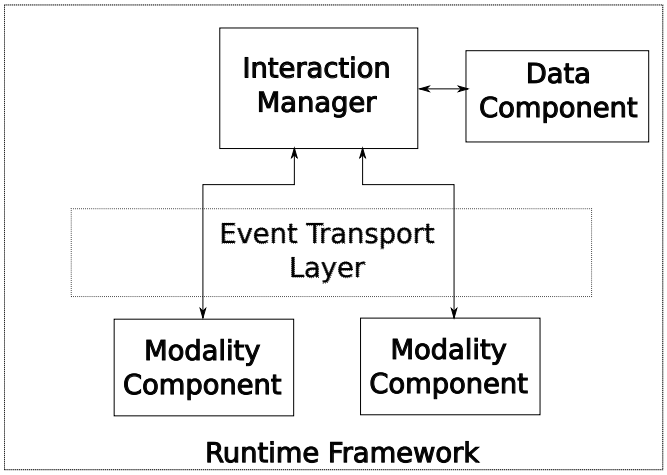

Figure 29. MMI Architecture Components

4.3.2 MMI Life-Cycle Events

MMI Architecture [8] defines the following standard application Life-Cycle Events for the interface between the IM and MCs.

Note: Here the key words *MUST*, *MUST NOT*, *REQUIRED*, *SHALL*, *SHALL NOT*, *SHOULD*, *SHOULD NOT*, *RECOMMENDED*, *MAY* and *OPTIONAL* are to be interpreted as described in "RFC 2119: Key words for use in RFCs to Indicate Requirement Levels" [88].

NewContextRequest and NewContextResponse:

MCs *MAY* send a NewContextRequest to the IM to request that a new context [2] be created. If this event is sent, the IM *MUST* respond with

[2] A URI that *MUST* be unique for the lifetime of the system. It is used to identify the current interaction. All events relating to a given interaction *MUST* use the same context URI. Events containing a different context URI *MUST* be interpreted as part of other, unrelated, interactions.

the NewContextResponse event. The NewContextResponse event *MUST* only be sent in response to the NewContextRequest event. The IM *MAY* create a new context without a previous NewContextRequest by sending a PrepareRequest or StartRequest containing a new context ID to the MCs. Furthermore the IM *MAY* respond with the same context in response to NewContextRequests from different (multiple) MCs, since the interaction can be started by different MCs independently.

PrepareRequest and PrepareResponse:

The IM *MAY* send a PrepareRequest to allow the MCs to pre-load markup and prepare to run. MCs are not required to take any particular action in response to this event, but they *MUST* return a PrepareResponse event. MCs that return a PrepareResponse event with Status of 'Success' *SHOULD* be ready to run with close to 0 delay upon receipt of the StartRequest.

The IM *MAY* send multiple PrepareRequest events to a MC for the same context before sending a StartRequest. Each request *MAY* reference a different ContentURL or contain different in-line content. When it receives multiple PrepareRequests, the MC *SHOULD* prepare to run any of the specified content.

StartRequest and StartResponse:

To invoke MCs, the IM *MUST* send a StartRequest. The MC *MUST* return a StartResponse event in response. The IM *MAY* include a value in the ContentURL or Content field of this event. In this case, the MC *MUST* use this value.

If the MC receives a new StartRequest while it is executing a previous one, it *MUST* either cease execution of the previous StartRequest and begin executing the content specified in the most recent StartRequest, or reject the new StartRequest, returning a StartResponse with status equal to 'Failure'.

DoneNotification:

If the MC reaches the end of its processing, it *MUST* return a DoneNotification to the IM.

CancelRequest and CancelResponse:

The IM *MAY* send a CancelRequest to stop processing in the MC. In this case, the MC *MUST* stop processing and then *MUST* return a CancelResponse.

PauseRequest and PauseResponse:

The IM *MAY* send a PauseRequest to suspend processing by the MC. MCs *MUST* return a PauseResponse once they have paused, or once they determine that they will be unable to pause.

ResumeRequest and ResumeResponse:

The IM *MAY* send the ResumeRequest to resume processing that was paused by a previous PauseRequest. The IM *MUST NOT* send the ResumeRequest to a context that is not paused due to a previous PauseRequest. Implementations that have paused *MUST* attempt to resume processing upon receipt of this event and *MUST* return a ResumeResponse afterwards. The 'Status' *MUST* be 'Success' if the implementation has succeeded in resuming processing and *MUST* be 'Failure' otherwise.

ExtensionNotification:

This event *MAY* be generated by the IM and *MAY* be generated by the MC. It is used to encapsulate application-specific events that are extensions to the framework defined here. For example, if an application containing a voice modality wants that MC to notify the IM when speech is detected, it will cause the voice modality to generate an ExtensionNotification event (with a 'name' of something like 'speechDetected') at the appropriate time.

ClearContextRequest and ClearContextResponse:

The IM *MAY* send a ClearContextRequest to indicate that the specified context is no longer active and that any resources associated with it may be freed. MCs are not required to take any particular action in response to this event, but *MUST* return a ClearContextResponse. Once the IM has sent a ClearContextRequest to a MC, it *MUST NOT* send the MC any more events for that context.

StatusRequest and StatusResponse:

The StatusRequest message and the corresponding StatusResponse are intended to provide keep-alive functionality. Either the IM or the MC *MAY* send the StatusRequest message. The recipient *MUST* respond with the StatusResponse message.

4.3.3 Possible Multimodal Web Applications based on MMI Architecture

These days there are several advanced multimodal Web applications available that interact with cloud services such as Google Search [3]Apple's Siri [4, 5] and NTT Docomo's Shabette-Concier [6, 7]. However, implementations of those applications are different between vendors, and cannot get connected.

On the other hand, some of the existing multimodal Web applications, e.g., AT&T's Speak4it [89, 90, 91] and Openstream's Personal Health Record Management System (PHR) [92] use MMI Architecture and EMMA as the basic framework to integrate simultaneous inputs by the user including speech, handwriting, gesture and recorded video. Both AT&T and Openstream are W3C Members and actively working on W3C standards, e.g., MMI Architecture and EMMA, so there is a possibility those applications could be integrated with the other standard-based multimodal applications in the near future. For example, there is a possibility for a simple speech-ready TV remote, which cooperates with HTML5-based GUI on a smartphone and enables us to specify channels, volume, etc., using voice as Fig. 30 shows.

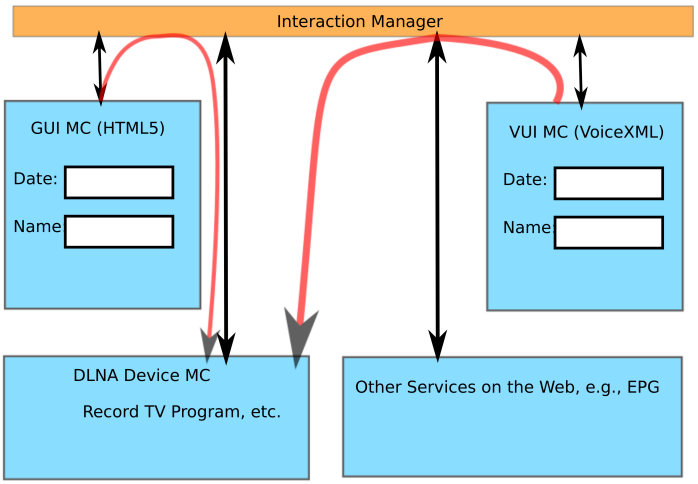

Figure 30. Combination of MMI Architecture and Digital TV

4.4 MoW: MMI over WebSocket

4.4.1 The Need for Standard Library for Multimodal Web Applications

As described in Section 3.4, how to implement advanced multimodal applications varies from vendor to vendor, and there are still several issues on multimodal Web application development. In order to solve those issues and provide a framework for application development that is independent from the ability of each application author, there is a strong need for a standardized Web application development method.

So this chapter discusses a standardized JavaScript library which provides a standard way for interfaces between various Input/Output modalities and cloud services based on the W3C MMI Architecture specification [8, 9, 10] so that developers can handle a variety of Web contents and Input/Output modalities regardless of their skills.

4.4.2 MoW's Structure

This Section proposes the "MMI over WebSocket (MoW)" library which consists of the following three modules as shown in Fig. 31 in order to solve the issues on multimodal Web applications:

- Front-end Module
- MMI Module
- Communication Module

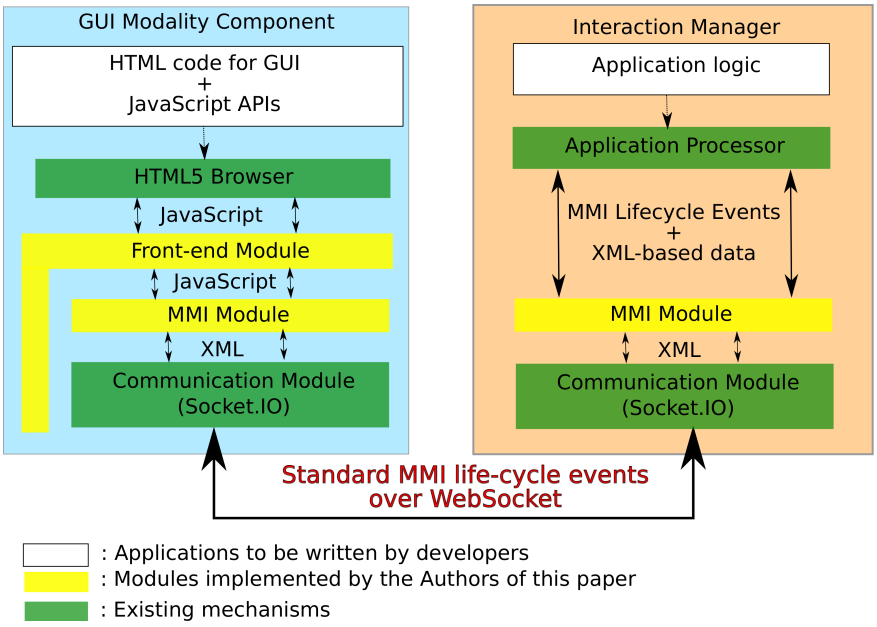

Figure 31. MMI over WebSocket

The following explains the details of the three modules:

Front-end Module The Front-end Module (Fig. 31) works on the MC side, and handles the Input/Output interfaces with HTML5-based Web browsers, and controls both the MMI Module and the Communication Module. The Front-end Module takes care of the interfaces with HTML5-based browsers and hides the

details of the MMI Module and the Communication Module, so the developers of MCs can concentrate on the description of HTML codes and JavaScript APIs without attention to MMI Life-Cycle Events or WebSocket connection.

MMI Module As Fig. 31 shows, HTML5 Browsers generally assume "HTML and JavaScript" as the Input/Output data. On the other hand, MMI Architecture uses EMMA [87], the standard data format for multimodal Web applications, and other XML-based data formats. So data conversion between JavaScript (including JSON as its data format) and XML is needed. The MMI Module (Fig. 31) generates necessary MMI Life-Cycle Events and XML data (=EMMA, etc.) based on the MMI Architecture specification [8]. It also converts the inputted XML to JavaScript and then sends it back to the Front-end Module. See Section 4.3.2 for details of the MMI Life-Cycle Events.

Communication Module The Communication Module (Fig. 31) sends the MMI Life-Cycle Events defined by the MMI Architecture specification [8] between MCs and the IM using WebSocket. To implement this module, Node.js [93, 94], a commonly used JavaScript development environment, and Socket.IO [95], a popular socket connection library for JavaScript, were used for the purpose of efficiency.

4.4.3 MoW's Merits

Because MoW uses W3C's MMI Architecture as the basis of the standardized framework to reduce application developers' workload, it has the following merits:

- Flexible and network-transparent integration of the Web and devices
- Separation of application processing logic and Input/Output modalities

The details of the above two points are described below:

Flexible and Network-Transparent Integration of the Web and Devices

W3C's MMI Architecture allows nested structure and resolves the problem of the "hub-and-spoke model", the hub's functionality as the controller becoming

too complicated when there are too many modality components in the system. The architecture enables us to do the following:

1. Combining (1) a speech input modality using a microphone with (2) speech output modality using a loud speaker on a smartphone, and make a "Voice User Interface (VUI) MC"

2. Integrating the "VUI MC" with a "Digital TV MC", an "HTML5-based GUI MC", and the IM

3. Controlling the "Digital TV MC" using either the "GUI MC" or the "VUI MC" based on the user's preference (See also Fig. 30)

Note that MoW provides a JavaScript API for HTML5 browsers to handle MMI Life-Cycle Events over WebSocket. So it can make any HTML5-based Web browser a GUI MC for MMI Architecture, and dynamically adds voice Input/Output capability to the GUI Web browser as mentioned above.

Separation of Application Processing Logic and Input/Output Modalities

MMI Architecture is designed based on the "Model-View-Controller (MVC)" model [85], a well-known design pattern for user interfaces, and consists of the following clearly separated three components (See also Fig. 29):

- Model: Data Component
- View: Input/Output MCs
- Controller: Application Logic within IM

Thanks to the architecture design, "Application logic" and "HTML code for GUI + JavaScript API for communication" can be clearly separated (Fig. 31). So developers of MCs such as GUI and speech interface can concentrate on handling user input and system output for each MC without the care of the whole application logic. Application service providers who develop the IM can concentrate on the application logic without attention to the details of each Input/Output MC.

Note:

MoW uses Node.js [93, 94] for event handling and Socket.IO [95] for socket connection to provide a way for Web application developers to handle both MMI Life-Cycle Events and bi-directional full duplex WebSocket connection easily.

4.5 Evaluation of MoW's Usability from the Viewpoint of Processing Speed and Server Load

As described in Section 4.4.3, the proposed library, MoW, uses W3C's MMI Architecture as the basic framework and has the following merits:

- Flexible and network-transparent integration of the Web and devices
- Separation of application processing logic and Input/Output modalities

However, there is a need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of 1 Mora (around $100 - 200$ ms) based on the result of Section 2.2, and fast connection between devices is required. The proposed library might cause some latency due to the additional mechanism such as the data conversion described in Section 4.4.2 despite its provided convenience. Therefore MoW uses WebSocket for network connection instead of usual HTTP to improve processing speed and server's workload, and this section evaluates the efficiency of using WebSocket as the connection protocol between MC and IM from the viewpoint of (1) processing speed improvement and (2) server load reduction.

The result of the preliminary experimentation on improvement of the processing speed using WebSocket is described below in Section 4.5.1, and then the server load is determined using our implemented MoW library in Section 4.5.2.

4.5.1 Preliminary Experiment on Improvement of Processing Speed using WebSocket

As a preliminary experimentation to see if it is possible to improve processing speed using WebSocket instead of HTTP, we measured the processing speed of a Web application [96] which analyzes some specific Japanese text (the first seven

paragraphs of a Japanese famous book, "Botchan") and separates it into morphemes.

Experimentation Environment The environment and the network structure of this preliminary experiment is shown in Fig. 32. A PC is used as the server for the IM, and another PC is used as the client for MCs.

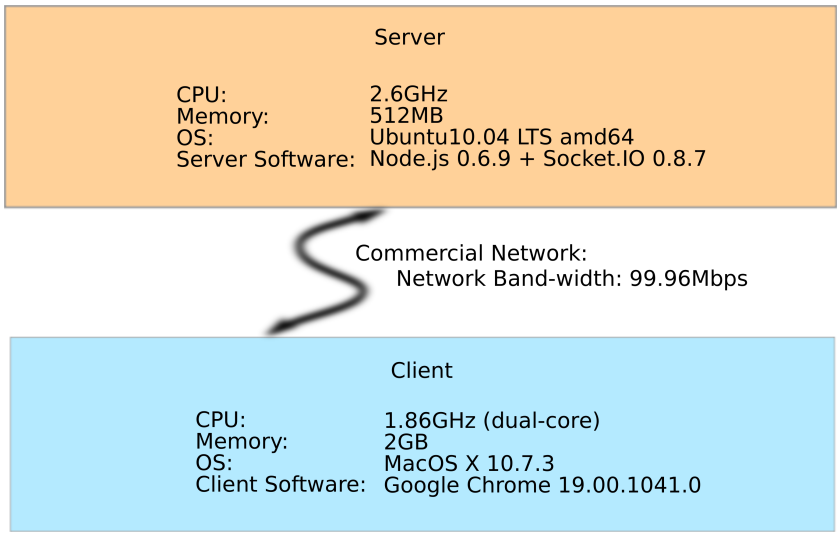

Figure 32. Speed Test Environment

Experimentation Results We measured the processing time of the Web applications using both HTTP and WebSocket as the connection protocol repeatedly 100 times, and checked the distribution of the resulted processing times by splitting the results into bins of 20ms as Fig. 33 shows.

As the results show, the highest bin of HTTP connection (=green histogram in Fig. 33) was 4,880ms-4,900ms and the median was 4,890ms. On the other hand, the highest bin of WebSocket connection (=red histogram in Fig. 33) was 160ms-180ms and the median was 170ms. So it is clarified that the processing speed can be improved by 29 times if we use WebSocket as the connection protocol.

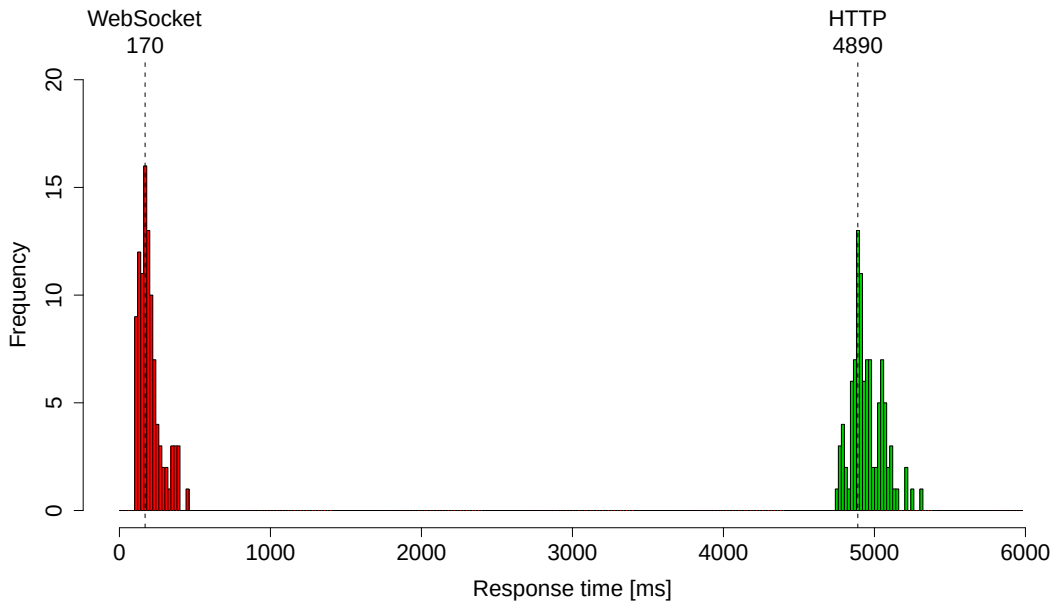

Figure 33. Speed Test Results

4.5.2 Experiment on Server Load Decrease using WebSocket

To check if it is possible to reduce the server's load using WebSocket as the connection protocol instead of HTTP, we implemented the following two components:

- **IM** installed on the server PC which handles the exchange of MMI Life-Cycle Events with the MC(s)
- **MC** installed on the client PC which sends MMI Life-Cycle Events to the IM

Then multiple (1-100) MCs were connected with a specific IM on the server PC, and the CPU load and the memory usage on the server PC were investigated.

For this experiment, the typical MMI Life-Cycle Events shown in Fig. 34 were exchanged between the IM and the MCs during an application's life-cycle, i.e., from an application's invocation to its termination.

Experimentation Environment The experimentation environment and the network structure of the experiment are shown in Fig. 35.

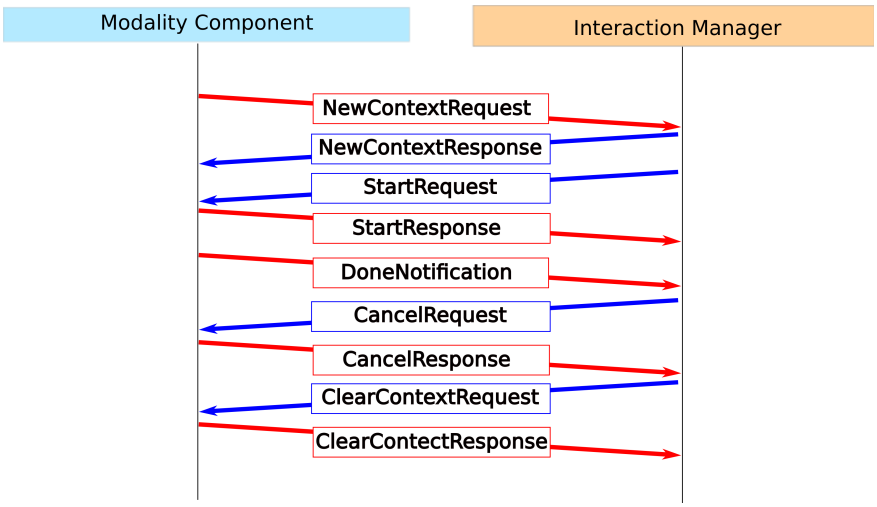

Figure 34. Life-Cycle Event Transaction during the Load Test

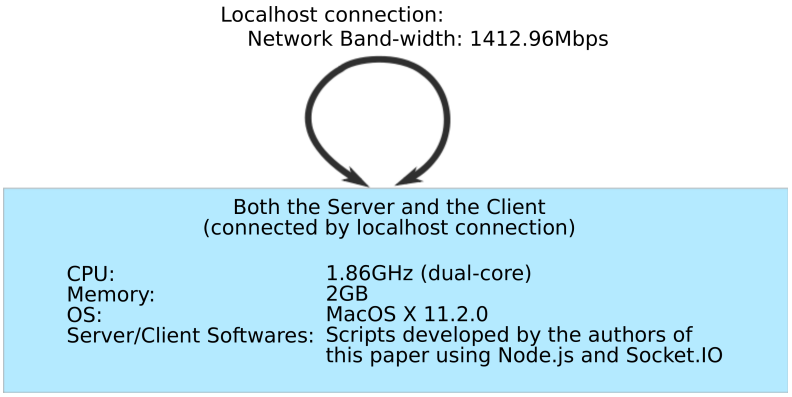

Figure 35. Load Test Environment

Experimentation Results The results of the experiment is shown as Table 14 and Fig. 36.

Table 14. Results of the Load Test

Protocol	CPU Load	Memory Usage
HTTP	The CPU load of the server was **70%** when **two MCs** connected simultaneously, and the CPU load became **100%** when **three MCs** connected. So the effect of more than two MCs could not be investigated.	**26MB** of the memory on the server was used when **two MCs** were connected simultaneously, but the effect of more than two MCs could not be investigated because the server's CPU load reached 100%.
WebSocket	As shown by the blue graph in Fig. 36, the CPU load of the server was **around 50%** when **100 MCs** were connected simultaneously.	As shown by the red graph in Fig. 36, **24MB** of the memory on the server was used when **100 MCs** were connected simultaneously.

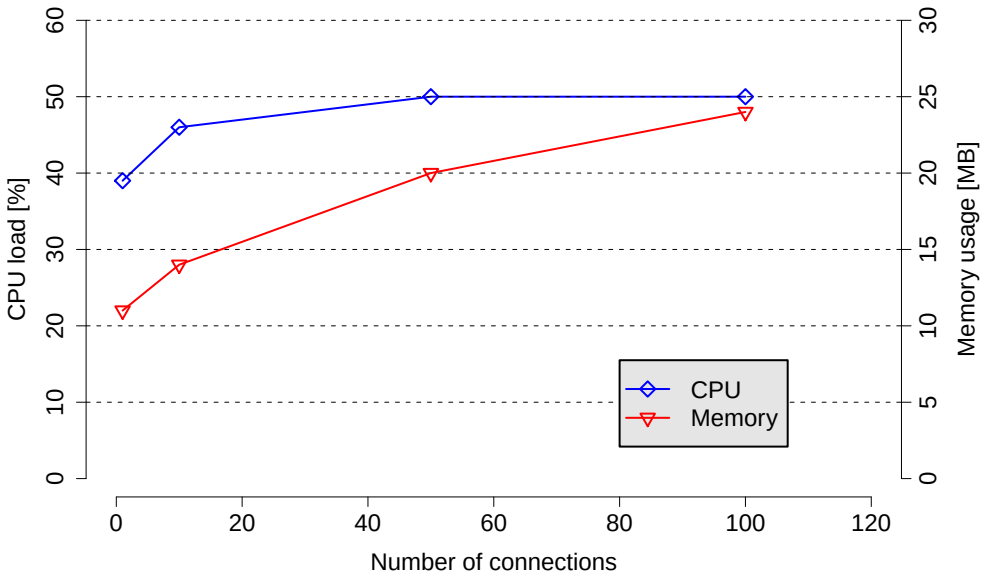

Figure 36. Load Test Results

4.5.3 Discussion based on the Results

As described in Section 4.5.1, the processing speed for Web applications can be improved by around 29 times when WebSocket is used as the connection protocol between the server and the client instead of HTTP. So we implemented the MoW library as described in Section 4.4.2, and then evaluated its usability from the view point of processing speed as described in Section 4.5.2. Based on the result of the evaluation experiment, we found that the CPU load and the memory usage on the server could be reduced using MoW as expected. So it was clarified that MoW could deal with the possible latency caused by the additional data conversion for MMI Life-Cycle Events and EMMA.

Note that when HTTP is used as the connection protocol, using more than two MCs simultaneously made the server's CPU load 100% and no more processing was available. On the other hand, WebSocket connection using MoW allowed 100 MCs to connect simultaneously and the CPU load was just 50%. So MoW would be useful when there are more than two MCs that connect with the IM simultaneously.

4.6 Organizing International Standardization within W3C

4.6.1 The Author's Role within W3C

The author started to work at W3C in 2005, and has been working on global standardization of Web technology as the Activity Lead for (1) the Voice Browser WG, (2) the Multimodal Integration WG and (3) the Web and TV IG. In addition, the author has been participating in the technical discussions within the following groups related to the above three groups and encouraging the collaboration with these groups as Fig. 37 shows:

- HTML WG
- Web Applications WG
- Device APIs WG
- Digital Publishing IG
- Web-based Signage BG
- Automotive and Web Platform BG
- Web and Broadcasting BG

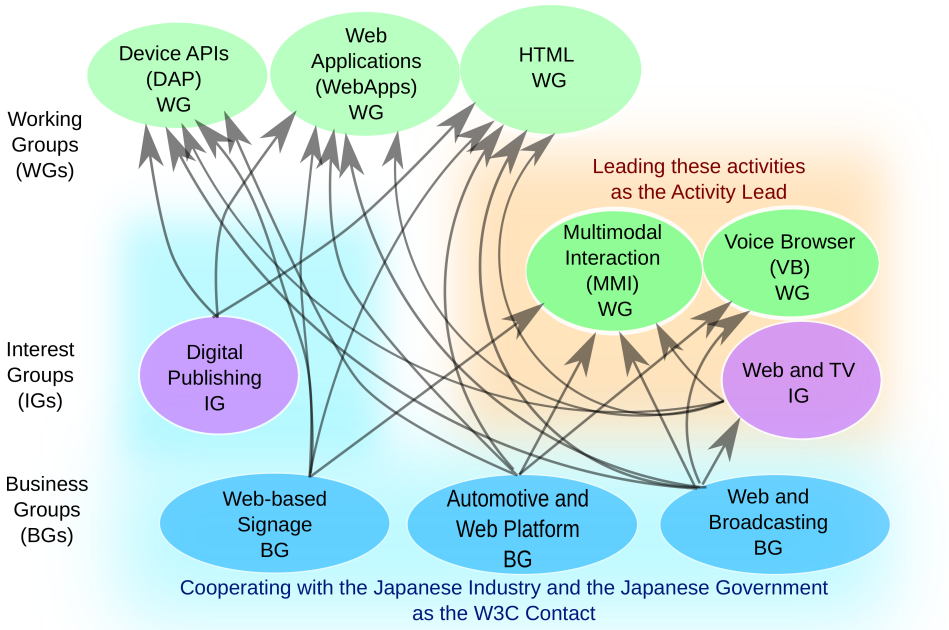

Figure 37. Relationship between W3C Groups related to the Author

4.6.2 The Author's Tasks as the W3C Activity Lead

The tasks of the author as the W3C Activity Lead includes the following as Fig. 38 shows:

- Work with the Chair to manage the group's activity:
  - Organizing W3C workshops
  - Launching new discussion groups, e.g., WGs and IGs
  - Participating in the groups' technical discussion (mailinglists, teleconferences and F2F meetings) and give advice as an expert
  - Publication of Web standards
- Interface for outside the group:
  - Establishing liaisons with the other standardization organizations
  - Discussions with non-Member public commentators
  - Collaboration with the other W3C Team Staff
  - Cooperation with the other W3C groups

4.6.3 W3C Workshops on Multimodal User Interfaces

As described in Section 4.3, the W3C Multimodal Interaction Working Group (MMI WG) works on specifications for multimodal user interfaces including MMI Architecture [8] and EMMA 1.1 [97, 98].

To solicit business use cases and identify requirements for those existing W3C specifications and make them even more useful to developers and users, the MMI WG held several W3C workshops as shown in Table 15.

Note:

The author as the Activity Lead for the MMI WG organized all the workshops listed in Table 15, and led the technical discussions during those workshops cooperatively with the other workshop co-Chairs.

Table 15. W3C Workshops on MMI Architecture

Workshop Theme → Date, Venue	Workshop Goals	Related W3C Specifications
Multimodal Architecture and Interfaces [99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114] → 16-17 Nov. 2007, Japan	To identify issues and requirements, e.g., need for modality specific grammars to improve the use of the MMI Architecture specification in order to make it more useful and popular	- MMI Architecture [8]
Conversational Applications [115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130] → 18-19 Jun. 2010, US	To identify key use cases and requirements for the existing W3C standards such as MMI Architecture [8] to support more sophisticated conversational applications	- MMI Architecture [8] - EMMA 1.1 [97] - SCXML [86]
Rich Multimodal Application Development [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151] → 22-23 Jul. 2013, US	To highlight the merits of HTML5 and MMI Architecture and to demonstrate the maturity of MMI Architecture and its suitability for developing innovative and compelling user-experiences across applications/devices	- MMI Architecture [8] - EMMA 1.1 [97]

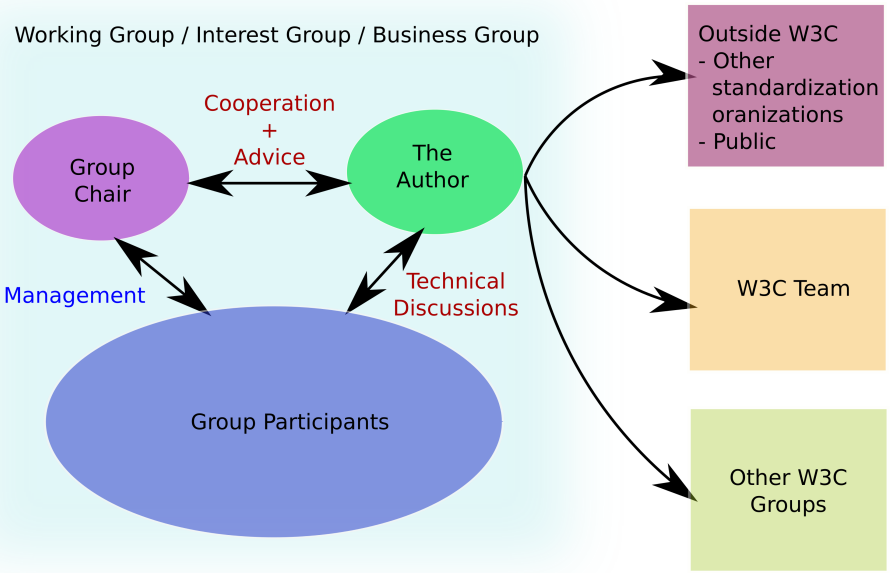

Figure 38. Activity Lead's Task

4.6.4 Interop TF of the MMI WG

As described in Section 4.6.3, the MMI WG held three international workshops [114, 130, 151] on MMI Architecture to discuss the following points:

- How to improve the use of MMI Architecture and make it even more useful
- How to extend MMI Architecture to support more sophisticated conversational applications
- How to develop innovative and compelling user-experiences across applications/devices using MMI Architecture

Resulting from the workshops and the discussions by the MMI WG after the workshops, the importance of the following two points has been recognized:

1. Integrating various devices including home appliances in a transparent manner

2. Reducing the load of developers to implement multimodal Web applications which integrate the Web and various devices

On the other hand, the author as the Activity Lead for the MMI WG has also been proposing the above two important points to the group since 2005. Consequently, the first point has already been included in the MMI WG's planning document, i.e., "the Multimodal Interaction Working Group Charter" [152], and the standardization work on extending the capability of EMMA [87] started in 2012 to define the updated version, EMMA 1.1 [97, 98]. Regarding the second point, the group has created a special Task Force named "Interoperability Testing Task Force (Interop TF)" to (1) test the interoperability between implementations developed by multiple vendors based on the MMI Architecture specification [8] and (2) generate a guideline on how to implement applications in order to reduce the developers' workload.

Interop TF implemented a prototype system which consists of the following three components shown as Fig. 39:

- Voice UI MC (VUI) developed by Openstream
- GUI MC developed by Deutsche Telekom
- SCXML-based IM developed by France Telekom

The TF published a guideline on multimodal Web application development which described the implementation details and issues clarified during the application development as an official MMI WG Note, "MMI interoperability test report" [153].

4.6.5 Japanese Chapter of the MMI WG

As shown in Fig. 27, the author proposed MoW to the Information Processing Society of Japan in 2011 [154] and 2012 [155, 156]. MoW provides a standard way for interfaces between various Input/Output modalities and Web services based on MMI Architecture [8, 9, 10] so that developers can handle a variety of Web contents and Input/Output modalities regardless of their skills. MoW also

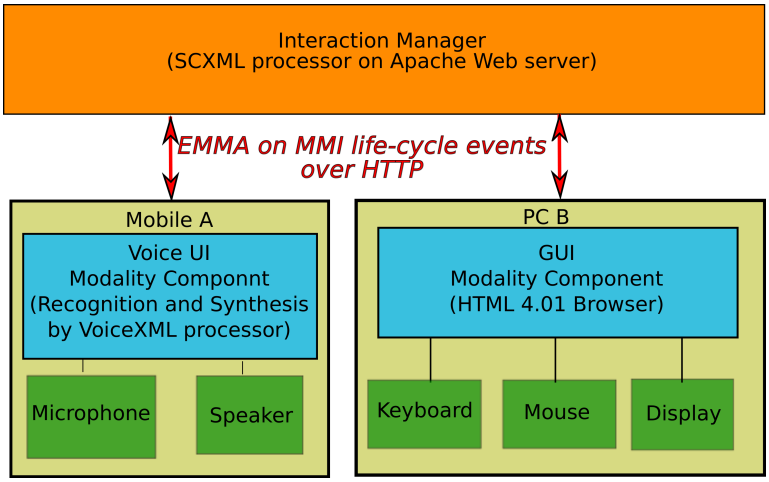

Figure 39. MMI Prototype (Ver. 1) by HTTP Connection

provides a smarter way to integrate HTML5 browsers with MMI Architecture-based systems as GUI modalities.

The author has been working with the stakeholders from the Japanese industry since 2005, and many W3C Japanese Members are now interested in using MMI Architecture as the basic framework for system integration. So the author as the Activity Lead for the MMI WG formed the Japanese Chapter of the group, and started discussions on how to apply MoW to actual multimodal system development in May, 2013.

One of the main targets of the MMI WG Japanese Chapter is to develop an updated version of multimodal Web application prototype system using MoW. The target of this second version prototype system is interaction among the following components as shown in Fig. 40:

- HTML5-based Web browsers on smartphones
- Various devices, e.g., digital TVs and air conditioners
- Various Web services, e.g., speech recognition, speech synthesis and video image analysis

The version 2 prototype is expected to be the reference implementation for integrating the Web and devices in a network-transparent manner using the stan-

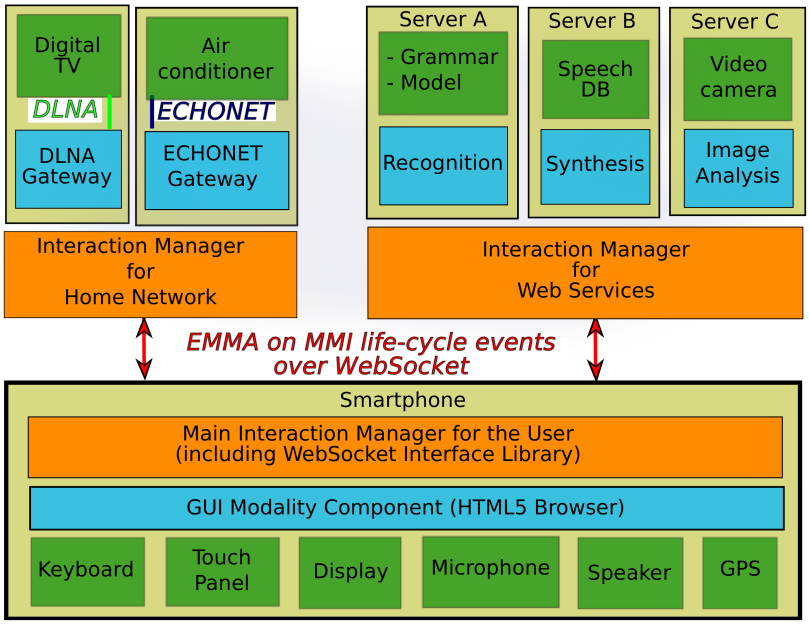

Figure 40. MMI Prototype (Ver. 2) by WebSocket

dard library, MoW. The details of the system implementation and issues clarified during the development is planned to be published as an official W3C document.

As described in Section 4.6.3, the author as the Activity Lead for the MMI WG organized the third MMI workshop on Rich Multimodal Application Development [131, 151] in July, 2013 to discuss the maturity of MMI Architecture and its suitability for developing advanced Web applications, and in order to accelerate the work of the MMI WG Japanese Chapter, the author himself presented the idea of integrating various MCs with HTML5-based Web browsers using MoW so that developers could integrate a variety of Web contents and Input/Output modalities with HTML5-based browsers easily [150]. During the talk, the author also mentioned the need for precise speech timing control using real-time OS and discourse timing model so that dialog-based computer interface could reproduce the timing and rhythm of spoken dialogs.

4.7 Conclusion

This chapter discussed the importance of a standard framework to integrate multiple user modalities and Web services, and illustrated the W3C MMI Architecture specification as one of the promising candidates to solve the issues.

Next, this chapter proposed a standard-based JavaScript library named "Mow" to solve the issues on multimodal Web application development so that developers could handle a variety of Web contents and Input/Output modalities regardless of their skills. The proposed method provides a smarter way to integrate HTML5 browsers as GUI modalities for advanced multimodal Web applications. The chapter also evaluated the usability of the proposed method from the viewpoint of processing speed and server load, because there was a possibility the proposed library might cause some latency despite the convenience. From the result of the evaluation experiment, we confirmed that various Input/Output modalities for multimodal Web applications could be handled in a dynamic and network-transparent way using our proposed method, and the possible overhead due to additional procedure for data conversion between JavaScript and XML could also be dealt with the method.

The proposed method has already been brought to the MMI WG which tackles the international standardization of MMI Architecture [8], and the author as the Activity Lead for the group will continue to work with the group to generate a reference implementation of the proposed method and confirm its feasibility to integrate HTML5 browsers, devices and Web services in a network-transparent manner. MoW, the pilot implementation of the proposed method, is planned to be published so that developers throughout the world can use it to implement their own multimodal Web applications. We would feel amply rewarded if the library proved helpful to them.

Note. As the result of various natural conversation analyses in Section 2.2, a basic heuristics of discourse timing management, i.e., "**The addressee waits for 1 Mora after the utterance completion of the current speaker to start his/her own utterance.**", was clarified. On the other hand, the result of the analyses implied the need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of one Mora (around

$100 - 200\,\mathrm{ms}$) based on the heuristics. So Section 2.5 proposed to use a real-time OS for precise and stable timing control. This chapter also proposed the use of WebSocket as the protocol for fast connection between devices, because WebSocket connection was expected to be much faster than ordinary HTTP connection. The actual connection speed of WebSocket connection was determined in this chapter, and it was 29 times faster than the ordinary HTTP connection. So the proposed library "MoW" is expected to be useful for precise and stable discourse timing control.

5. Conclusion

5.1 Summary

Recently the capability of mobile devices has been much improved, and various Input/Output modalities are available on those devices. On the other hand, ordinary GUIs including touch panels are not necessarily appropriate to use while walking, driving, etc., so the need for easier-to-use products is growing in order to help people in various situations.

To solve the issues, this dissertation tackled the research on speech interface especially speech synthesis because that is one of the most useful and easy-to-use interface modalities, and analyzed utterance timing and speech rate. As a result, a basic heuristics of discourse timing management was clarified, i.e., "**The addressee waits for 1 Mora after the utterance completion of the current speaker to start his/her own utterance.**", and it is expected the heuristics can be used as the basis of the possible discourse timing management model.

On the other hand, the result of the analyses implied the need for precise timing control of synthesized speech in millisecond resolution to handle the utterance timing of 1 Mora (around $100-200$ ms) based on the heuristics. So it was proposed to use a real-time OS for precise and stable timing management. Additionally, a possible network-based collaborative speech interaction framework was proposed based on the result of this research in 2002.

In 2005, the author started to work on international Web standardization at W3C (World Wide Web Consortium) as the Activity Lead for the W3C Multimodal Interaction Working Group, and concluded that a Web-based framework would be the most promising solution to achieve the objective of providing easy-to-use computer interfaces for various users in various environments based on the discussions in the group on multimodal interaction systems with experts throughout the world. Another issue was identified on how to implement components of network-based multimodal applications and interfaces between components that varied from vendor to vendor, which caused application developers the burden to define interfaces and implement all the necessary components themselves.

Therefore the author proposed a standard library named "MMI over Web-

Socket (MoW)" based on the W3C MMI Architecture specification so that developers could handle a variety of Web applications and Input/Output modalities regardless of their skills. The author implemented MoW as a JavaScript library, and evaluated its usability. Note that the ordinary protocol for Web-based multimodal applications is HTTP, and it is likely that the connection would not be fast enough for interactive applications. So the author proposed MoW use WebSocket as the protocol to improve the connection speed, and confirmed that the WebSocket connection was 29 times faster than the HTTP connection. Therefore MoW is expected to be useful for precise discourse timing management.

MoW has made it possible to develop easier-to-use computer interfaces for various users in various situations though it was difficult in 2002. MoW is the first attempt in the world to provide a standard and open library for multimodal Web applications, which is planned for public release to support application developers world wide.

5.2 Future Work

To solve the issues described in Section 1.1 and make computer interfaces even more usable and friendly, this dissertation discussed the following two technological areas:

- Discourse Timing Control
- Standard Framework for Web-based Integration

The combination of the results from this dissertation on those two technological areas will provide users the basic framework for more friendly and usable dialog-based computer interfaces, and make it possible to integrate the Web and various devices, e.g., digital TVs, digital signage, eBooks and In-Vehicle Infotainment (IVI) systems. However, there are still several issues to be solved in both the above technology areas as described below. We will continue to tackle these issues to define automatic discourse timing model and materialize Web-based spoken dialog system based on standard framework in order to make computer interfaces even more usable and friendly for all.

5.2.1 Remaining Issue on Discourse Timing Control

As the result of various analyses described in Section 2, the basic heuristics, which could be used as the basis of the possible discourse timing management model, has been clarified. However, further research is needed to define a detailed model of discourse timing and accomplish automatic control. So we will continue to analyze more conversation data in various situations from the viewpoint of acoustics, linguistics, utterance timing and speech rate, and clarify the detail of the discourse timing model to materialize an automatically controlled system.

5.2.2 Remaining Issue on Standard Framework for Web-based Integration

As described in Section 4, the author proposed a standard library, "MoW", based on MMI Architecture [8, 9, 10] to solve the issues on multimodal Web application development, and confirmed the usability of the pilot implementation from the viewpoint of processing speed and server load. However, further inspection is needed to see the applicability of MoW to actual multimodal Web applications. For that purpose, the author as the Activity Lead has formed a Japanese Chapter of the MMI WG and started the discussion on how to apply MoW to actual multimodal system development. The author will continue to work with the Japanese Chapter and the MMI WG to generate a reference implementation of the proposed method and confirm its feasibility to integrate MoW with HTML5 browsers, devices and Web services in a network-transparent manner. MoW is planned to be published so that developers throughout the world can use it to implement their own multimodal Web applications. We would feel amply rewarded if the library proved helpful to them.

Acknowledgements

This dissertation could not have completed without the support of a lot of people.

First of all, I really would like to thank my thesis supervisor, Professor Satoshi Nakamura, for his guidance, encouragement and patience throughout the completion of this work.

The other members of the thesis committee also provided important and thoughtful advice and encouragement, Professor Yuji Matsumoto, Professor Nick Campbell, Associate Professor Tomoki Toda and Professor Masao Isshiki. Especially I am indebted to Professor Masao Isshiki who is the manager of the W3C Keio Site and provided me the opportunity to challenge this dissertation.

I would also like to appreciate thoughtful advice and constant encouragement of my former supervisors at NAIST when I was a graduate student there in 2002-2005, Professor Kiyohiro Shikano and Associate Professor Hideki Kashioka.

I wish to thank Mr. Kensaku Komatsu from NTT Communications who helped me implement "MMI over WebSocket (MoW)", the standardized multimodal Web application framework, Mr. Tomoya Asai (dynamis) from Mozilla Japan who kindly provided a clear and easy-to-understand figure of the Open Web Platform, Dr. Shinichiro Mori from Fujitsu who gave me many useful comments and Ms. Cinthia Seino who thoroughly proofread this dissertation and fixed all the syntactical errors.

In addition, I would like to thank Dr. Parham Mokhtari and Dr. Carlos Toshinori Ishi from ATR for their useful advice, Mr. Seiichi Tenpaku and Dr. Toshio Hirai from Arcadia Inc. for their thoughtful comments, and Mr. Keisuke Nakamura from NAIST for guidance on handling HMM.

I would like to thank all the friendly participants in the W3C Multimodal Interaction Working Group (MMI WG), the W3C Voice Browser Working Group (VBWG) and the W3C Web and TV Interest Group (Web&TV IG). Especially my gratitude should go to the following people:

- Dr. Deborah Dahl, the Chair of the MMI WG
- Dr. Daniel C. Burnett, the Chair of the VBWG
- Mr. Yosuke Funahashi, the co-Chair of the Web&TV IG

- Dr. James Barnett, the editor of the MMI Architecture specification
- Dr. Michael Johnston, the editor of the EMMA specification
- Mr. Paolo Baggia, the Editor of the PLS specification
- Mr. Ingmar Kliche, Mr. Nagesh Kharidi and Mr. Piotr Wiechno, the Authors of the MMI interoperability test report Working Group Note
- Dr. B. Helena Rodriguez and Mr. Raj Tumuluri, the Authors of the Registration & Discovery of Multimodal Modality Components in Multimodal Systems Working Group Note

Finally, and always, I would like to thank my wife, Kaoru, for her continuous patience, support and encouragement.

References

[1] Wikipedia: “HTML”, Wikipedia (online), http://en.wikipedia.org/wiki/HTML.

[2] Raggett, D., Hors, A. L. and Jacobs, I.: “HTML 4.01 Specification”, W3C (online), http://www.w3.org/TR/html401/.

[3] Schalkwyk, J. et al.: “Google Search by Voice: A case study”, Google (online), http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//pubs/archive/36340.pdf.

[4] Apple: “Siri”, Apple (online), http://www.apple.com/jp/ios/siri/.

[5] Bennett, B.: “Apple Siri LIVE Demonstration”, YouTube (online), http://www.youtube.com/watch?v=l9iXUxPbDQg.

[6] Yoshimura, K.: “Shabette-Concier Service realized by Natural Language Processing”, IPSJ SIG Technical Report (2012-SLP-93-4), 2012.

[7] NTT Docomo: “Shabette-Concier Demonstration (仲里依紗 CM ドコモ DoCoMo しゃべってコンシェル)”, YouTube (online), http://www.youtube.com/watch?v=7S1_AaUvn6k.

[8] Barnett, J. et al.: “Multimodal Architecture and Interfaces (MMI Architecture)”, W3C (online), http://www.w3.org/TR/mmi-arch/.

[9] Maes, S. H. and Saraswat, V.: Multimodal Interaction Requirements, W3C (online), http://www.w3.org/TR/mmi-reqs/.

[10] Grifoni, P. et al.: “*Multimodal Human Computer Interaction and Pervasive Services*”, IGI Global, 701 E. Chocolate Ave. Hershey, PA 17033, USA (2009). (available from http://www.igi-global.com/book/multimodal-human-computer-interaction-pervasive/787).

[11] Ashimura, K.: “Multimodal Interaction Working Group”, W3C (online), http://www.w3.org/2002/mmi/.

[12] Zue, V.: “Conversational Interfaces: Advances and Challenges”, Proc. Eurospeech ’97, KN-9-KN-18, 1997.

[13] Takezawa, T. et al.: “A Japanese-to-English Speech Translation System: ATR-MATRIX”, Proc. ICSLP 1998, 1998.

[14] Sugaya, F. et al.: “Development and Evaluation of ATR-MATRIX Speech Translation System”, IPSJ Journal 43(7), pp.2230–2241, 2002-07-15, 2002.

[15] Lee, A.: “Large Vocabulary Continuous Speech Recognition Engine Julius ver. 4”, IPSJ SIG Technical Reports, (2007-SLP-69), No.53, pp.307–312, 2007.

[16] Mera, Y. et al.: “Statistical analysis of phoneme likelihoods in spontaneous speech using automatic segmentation by Julius”, Proc. ASJ Fall 2-1-5, pp.57–58, 2001.

[17] 木田敦子，乾裕子，神崎享子，高梨克也，井佐原均: “構文論から見た対話 -円滑な話者交替を可能にする構文構造-”, 人工知能学会研究会資料 (SIG-SLUD-A102-6), pp.33–38, 2001.

[18] Kaiki, N. and Sagisaka, Y.: “Study of Pause Insertion Rules Based on Local Phrase Dependency Structure”, IEICE Transaction, J79-D-II, No.9, pp.1455–1463, 1996.

[19] Matsumoto, Y.: “Morphological Analysis System ChaSen”, IPSJ Magazine, Vol.41, No.11, pp.1208–1214, 2000.

[20] Campbell, N. and Mokhtari, P.: “Voice Quality, the 4^{th} prosodic dimension”, Proc. ICPhS 2003, pp.2414–2420, 2003.

[21] Campbell, N.: “Recording techniques for capturing natural every-day speech”, Proc. LREC 2002, pp.2029–2032, 2002.

[22] Campbell, N. and Mokhtari, P.: “DAT vs. MD”, Proc. ASJ Spring 1-P-27, pp.405–406, 2002.

[23] Campbell, N.: “Labelling natural conversational speech data”, Proc. ASJ Fall 1-10-22, pp.273–274, 2002.

[24] Cedergren, H. J. and Perreault, H.: “Speech rate and syllable timing in spontaneous speech”, Proc. ICSLP 1994, Vol.3, pp.1087–1090, 1994.

[25] Crystal, T. H., House, A. S.: “Articulation rate and the duration of syllables and stress groups in connected speech”, Journal of ASA, 88, pp.101–112, 1990.

[26] Verhasselt, J. P.: “A fast and reliable rate of speech detector”, Proc. ICSLP 1996, Vol.4, pp.2258–2261, 1996.

[27] Wood, S.: “What happens to vowels and consonants when we speak faster?”, Working Papers 9, Phonetics Laboratory Lund University, pp.8–39, 1973.

[28] Pfitzinger, H.: “Local Speech Rate as a Combination of Syllable and Phone Rate”, Proc. ICSLP 1998, Vol.3, pp.1087–1090, 1998.

[29] Fukada, T. and Sagisaka, Y.: “Automatic Generation of Pronunciation Dictionary Based on Pronunciation Networks”, Proc. of IPSJ SIG-SLP 1996-SLP-14-3, pp.15–22, 1996.

[30] Ohwaki, H. et al.: “Phonetic Typewriter using Phonotactic Constraints”, Technical Report of IEICE SP93-113, pp.71–78, 1993.

[31] Kawahara, T. et al.: “Japanese Dictation ToolKit – 1999 version –”, Journal of ASJ, Vol.57, No.3, pp.210–214, 2001.

[32] Deligne, S. and Bimbot, F.: “LANGUAGE MODELING BY VARIABLE LENGTH SEQUENCES : THEORETICAL FORMULATION AND EVALUATION OF MULTI-GRAMS”, Proc. ICASSP 95, pp.169–172, 1995.

[33] Ashimura, K. and Campbell, N.: “Speech Rate of Similar Speech Pattern in Dialogue Speech”, 1-7-9, pp.229-230, Proc. ASJ 2004 Spring, 2004.

[34] Pellegrino, F., Farinas, J., and Rouas, J.-L.: “Automatic Estimation of Speaking Rate in Multilingual Spontaneous Speech”, Proc. Speech Prosody 2004, pp.517–520, 2004.

[35] Sagisaka, Y.: “Speech Synthesis of Japanese Using Non-Uniform Phoneme Sequence Units”, Technical Report of IEICE, SP87-136, pp.47–52 (1988.3).

[36] Sagisaka, Y.: “Corpus Based Speech Synthesis”, J. Signal processing., Vol.2 No.6, pp.407–414 (1998.2).

[37] Campbell, N. and Black, A. W.: “CHATR: a multi-lingual speech re-sequencing synthesis system”, Technical Report of IEICE, SP96-7, pp.45–52 (1996,5)

[38] Tanaka, K. et al.: “A text-to-speech system using multi-form unit”, Proc. ASJ Fall, pp.211–212 (1999).

[39] Kubozono, H. and Ohta, S.: “*Phoneme construction and accent*”, Kenkyusha, Tokyo, Japan (1998).

[40] Ashimura, K. and Tenpaku, S.: “A New Corpus-based Speech Synthesis Method using Extended CV as Speech Unit”, 2-1-1, pp.175–176, ASJ 2000 Autumn, 2000.9.

[41] Ashimura, K. et al.: “SPEAKS: a new speech synthesizer -using extended CV as speech unit-”, 2-1-2, pp.177–178, ASJ 2000 Autumn, 2000.9.

[42] Ashimura, K. et al.: “SPEAKS: A New Speech Synthesizer -using Extended CV as Speech Unit-”, Proceedings of WESTPRAC VII, pp.417-1–417-2, 2000.10.

[43] Ishiwata, Y.: “Advanced Real-Time Linux”, SoourceForge (online), `http://sourceforge.net/projects/art-linux/`.

[44] Ishiwata, Y.: “Details of ART-Linux”, SoourceForge (online), `http://sourceforge.net/projects/art-linux/`.

[45] AIST Digital Human Research Center: “ART-Linux”, AIST (online), http://www.dh.aist.go.jp/en/research/assist/ART-Linux/.

[46] 石綿陽一: “*Art-Linux* **設計の経緯と使い方**”, インタフェース 1999 年 11 月号, pp.109–118, CQ 出版 (1998.6).

[47] Ashimura, K., Campbell, N. and Takeda, K.: “Analysis of utterance timing in everyday conversation”, IPSJ Technical Report (2002-SLP-40-19), pp.109–114, 2002.

[48] W3C: “W3C Top Page”, W3C (online), http://www.w3.org/.

[49] 佐伯千種，島田博也，田畑伸哉：“WWW コンテンツ統計調査報告書　～ 我が国の Web 上のコンテンツ情報量から見たインターネットの発展 ～ (No.2004-I-02)”，総務省（オンライン），http://www.soumu.go.jp/iicp/chousakenkyu/data/research/survey/telecom/2004/2004-1-02-1.pdf.

[50] Wikipedia: “ Web 2.0”, Wikipedia (online), http://en.wikipedia.org/wiki/Web_2.0.

[51] Wikipedia: “HTML5”, Wikipedia (online), http://en.wikipedia.org/wiki/HTML5.

[52] Hickson, I.: “HTML 5 A vocabulary and associated APIs for HTML and XHTML”, W3C (online), http://www.w3.org/TR/html5/.

[53] Berners-Lee, T. and Connolly, D.: “Hypertext Markup Language (HTML)”, W3C (online), http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt.

[54] Berners-Lee, T. and Connolly, D.: “Hypertext Markup Language - 2.0”, IETF (online), http://www.rfc-editor.org/rfc/rfc1866.txt.

[55] Raggett, D.: “A history of HTML”, W3C (online), http://www.w3.org/People/Raggett/book4/ch02.html.

[56] Raggett, D.: “HTML 3.2 Reference Specification”, W3C (online), http://www.w3.org/TR/REC-html32.

[57] Raggett, D., Le ,Hors, A. and Jacobs, I.: “HTML 4.0 Specification”, W3C (online), http://www.w3.org/TR/REC-html40-971218/.

[58] Raggett, D., Le ,Hors, A. and Jacobs, I.: “HTML 4.01 Specification”, W3C (online), http://www.w3.org/TR/html401/.

[59] Pemberton, S. et al.: ”XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)”, W3C (online), http://www.w3.org/TR/xhtml1/.

[60] Hickson, I.: “HTML 5 A vocabulary and associated APIs for HTML and XHTML”, W3C (online), http://www.w3.org/TR/2008/WD-html5-20080122/.

[61] Kesteren, A. v. and Pieters, S.: "HTML5 differences from HTML4", W3C (online), http://www.w3.org/TR/html5-diff/.

[62] Google: "HTML5 ROCS", Google (online), http://www.html5rocks.com.

[63] Humphrey, D. et al.: "Audio Data API", Mozilla (online), https://wiki.mozilla.org/Audio_Data_API.

[64] Wikipedia: "WebSocket", Wikipedia (online), http://en.wikipedia.org/wiki/WebSocket.

[65] Hickson, I.: "The WebSocket API", W3C (online), http://www.w3.org/TR/websockets/.

[66] Hickson, I.: "HTML Living Standard", WHATWG (online), http://whatwg.org/html.

[67] Hickson, I.: "Administrivia: Update on the relationship between the WHATWG HTML living standard and the W3C HTML5 specification", W3C (online), http://lists.w3.org/Archives/Public/public-whatwg-archive/2012Jul/0119.html.

[68] Jaffe, J.: "HTML5 and HTML.next", W3C (online), http://www.w3.org/QA/2012/07/html5_and_htmlnext.html.

[69] Cotton, P.: "Draft HTML5 Stabilization Plan", W3C (online), http://lists.w3.org/Archives/Public/public-html/2012Apr/0205.html.

[70] HTML WG: "Plan 2014", W3C (online), http://dev.w3.org/html5/decision-policy/html5-2014-plan.html.

[71] W3C: "W3C Technical Plenary / Advisory Committee Meetings Week 2012", W3C (online), http://www.w3.org/2012/10/TPAC/.

[72] W3C: "W3C TPAC 2012 Minutes", W3C (online), http://www.w3.org/2012/10/29-webtv-minutes.html.

[73] Andersson, O. et al.: "Scalable Vector Graphics (SVG) Tiny 1.2 Specification", W3C (online), http://www.w3.org/TR/SVGTiny12/.

[74] Carlisle, D. et al.: "Mathematical Markup Language (MathML) Version 3.0", W3C (online), http://www.w3.org/TR/MathML3/.

[75] Wikipedia: "Mashup", Wikipedia (online), http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid).

[76] Axelsson, J. et. al: "XHTML+Voice Profile 1.0 Note", http://www.w3.org/TR/xhtml+voice/.

[77] IBM and Opera Software: "X+V 1.1", http://www.ibm.com/developerworks/library/specification/x-v11spec/index.html.

[78] Sharma, C. and Kunins, J.: "*VoiceXML*", John Wiley and Sons, Inc., New York, USA (2002).

[79] McGlashan, S. et al.: "Voice Extensible Markup Language (VoiceXML 2.0)", W3C (online), http://www.w3.org/TR/voicexml20/.

[80] Oshry, M. et al.: "Voice Extensible Markup Language (VoiceXML) 2.1", W3C (online), http://www.w3.org/TR/voicexml21/.

[81] McGlashan, S. et al.: "Voice Extensible Markup Language (VoiceXML) 3.0", W3C (online), http://www.w3.org/TR/voicexml30/.

[82] Hocek, A. and Cuddihy, D.: "*Definitive VoiceXML*", Prentice-Hall, New Jersey, USA (2002).

[83] Bayer, S.: "*Building a Standards and Research Community with the Galaxy Communicator Software Infrastructure*", Springer, Berlin, Germany (2005). (in D. A. Dahl (Ed.), Practical Spoken Dialog Systems (Vol. 26, pp. 166–196). Dordrecht: Kluwer Academic Publishers).

[84] MIT: "Galaxy Communicator Documentation", MIT (online), http://communicator.sourceforge.net/sites/MITRE/distributions/GalaxyCommunicator/docs/manual/.

[85] Wikipedia: "Model-view-controller", Wikipedia (online), http://en.wikipedia.org/wiki/Model-view-controller.

[86] Barnett, J. et al.: "State Chart XML (SCXML): State Machine Notation for Control Abstraction", W3C (online), http://www.w3.org/TR/scxml/.

[87] Johnston, M. et al.: "EMMA: Extensible MultiModal Annotation markup language", W3C (online), http://www.w3.org/TR/emma/.

[88] IETF: "RFC 2119: Key words for use in RFCs to Indicate Requirement Levels", IETF (online), http://www.ietf.org/rfc/rfc2119.txt.

[89] Johnston, M. and Ehlen, P.: "Speak4it: Multimodal Interaction in the Wild", Spoken Language Technology Workshop (SLT), 2010 IEEE, http://www.research.att.com/export/sites/att_labs/people/Johnston_Michael_J/library/publications/speak4it_slt_2010.pdf, 2010.

[90] Johnston, M. and Ehlen, P.: "Speak4it and the Multimodal Semantic Interpretation System", Proc. INTERSPEECH 2011, pp.3333–3334, http://www.research.att.com/export/sites/att_labs/people/Johnston_Michael_J/library/publications/johnstonehlen_speak4it2011interspeech.pdf, 2011.

[91] Yellowpages.com: "Speak4it", AT&T (online), http://www.speak4it.com/.

[92] Openstream: "Openstream Multimodal PHR mixed initiative mixed mode", YouTube (online), http://www.youtube.com/watch?v=uTJ22AysBYM.

[93] Wikipedia: "Node.js", Wikipedia (online), http://en.wikipedia.org/wiki/Node.js.

[94] Joyent: "Node.js official site", Joyent, Inc. (online), http://nodejs.org/.

[95] Google: "Socket.IO: the cross-browser WebSocket for realtime apps.", Google (online), http://socket.io/.

[96] Komatsu, K.: "Wakachi demo (demonstration of Japanese morphological analysis using WebSocket)", Komasshu (online), http://wakachi.komasshu.info/.

[97] Johnston, M. et al.: "EMMA: Extensible MultiModal Annotation markup language Version 1.1", W3C (online), http://www.w3.org/TR/emma11/.

[98] Dahl, D. A. et al.: "Use Cases for Possible Future EMMA Features", W3C (online), http://www.w3.org/TR/emma-usecases/.

[99] Ashimura, K. et al.: "Workshop on W3C's Multimodal Architecture and Interfaces - Agenda", W3C (online), http://www.w3.org/2007/08/mmi-arch/agenda.html, 2007.

[100] Ashimura, K.: "Introduction to W3C & MMIWG", W3C (online), http://www.w3.org/2007/Talks/1116-w3c-mmi-ka/, 2007.

[101] Dahl, D.: "W3C's Multimodal Architecture and Interfaces", W3C (online), http://www.w3.org/2007/08/mmi-arch/slides/Overview.pdf, 2007.

[102] Ashimura, K.: "InkML: Digital Ink specification at W3C", W3C (online), http://www.w3.org/2007/Talks/1116-ink-ka/, 2007.

[103] Wilcock, G. and Jokinen, K.: "SCXML, Multimodal Dialogue Systems and MMI Architecture", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/w3positionpaper.pdf, 2007.

[104] Cave, E. K.: "Multimodal Framework Proposal", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/Japan_MMI_Workshop_Proposal_Nov_2007.pdf, 2007.

[105] Araki, M.: "Proposal of a Hierarchical Architecture for Multimodal Interactive Systems", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/position.pdf, 2007.

[106] Schepers, D.: "W3C Technologies as Consumers of Multimodal Interfaces", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/workshop-position-paper.xhtml, 2007.

[107] Tumuluri, R.: "Multimodality in Mobile Force Applications", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/MMI-WS2007-PositionPaper-Openstream.pdf, 2007.

[108] Yamakami, T.: "Challenges in Mobile Multimodal Application Architecture", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/a-draft07w3c-multimodal.pdf, 2007.

[109] Takagi, S.: "KDDI's position - Workshop on W3C's Multimodal Architecture and Interfaces (SVG Map service on mobile phones)", W3C (online), http://www.w3.org/2007/08/mmi-arch/slides/MMIpositionPaperTakagi.pdf, 2007.

[110] Madhvanath, S.: "Use cases of gesture and handwriting recognition", W3C (online), http://www.w3.org/2007/08/mmi-arch/slides/PenInput_MMIArchNov07_SriG.pdf, 2007.

[111] Metze, F. and Kliche, I.: "Kinesthetic input modalities for the W3C Multimodal Architecture", W3C (online), http://www.w3.org/2007/08/mmi-arch/papers/071009_Position_Paper_W3C_Workshop_v05.pdf, 2007.

[112] Shioya, M.: "IBM's position - Workshop on W3C's Multimodal Architecture and Interfaces", W3C (online), http://www.w3.org/2007/08/mmi-arch/slides/Proposal_paper_for_W3C_MMI_3.2.pdf, 2007.

[113] Ohno, K.: "Polytechnic University - Workshop on W3C's Multimodal Architecture and Interfaces", 2007.

[114] Ashimura, K. et al.: "Workshop on W3C's Multimodal Architecture and Interfaces - Summary", W3C (online), http://www.w3.org/2007/08/mmi-arch/summary.html, 2007.

[115] Ashimura, K. et al.: "Workshop on Conversational Applications - Agenda", W3C (online), http://www.w3.org/2010/02/convapps/agenda.html, 2010.

[116] Ashimura, K.: "Welcome from the W3C", W3C (online), http://www.w3.org/2010/Talks/0618-convapps-ka/, 2010.

[117] Patch, K.: "The User Context: Aligning System and User Behavior", W3C (online), http://www.w3.org/2010/02/convapps/Papers/Position_Paper_-_The_User_Context_2010-04-30.pdf, 2010.

[118] Burnett, D. C. and Bagshaw, P.: "Information Transfer from Dialogue Response Generation to Speech Synthesis", W3C (online), http://www.w3.org/2010/02/convapps/Papers/ConvApps_Workshop_v2.pdf, 2010.

[119] Fuqua, K.: "Conversational Lexical Standards", W3C (online), http://www.w3.org/2010/02/convapps/Papers/ConvLexicalStds.pdf, 2010.

[120] Chandra, S.: "Use of Part of Speech and morphological information for resolving multiple pronunciations in Pronunciation Lexicon Specification (PLS) for Indian Languages", W3C (online), http://www.w3.org/2010/02/convapps/Papers/Position-Paper_-India-W3C_Workshop-PLS-final.pdf, 2010.

[121] Baggia, P., Dahl, D and Carter, J.: "Extending SRGS to Support More Powerful and Expressive Grammars", W3C (online), http://www.w3.org/2010/02/convapps/Papers/Dahl.txt, 2010.

[122] Fuqua, K.: "Conversational Syntax Requirements", W3C (online), http://www.w3.org/2010/02/convapps/Papers/ConvSyntaxStd.pdf, 2010.

[123] Gandrabur, S.: "The Future of Advanced Dialogue Applications", W3C (online), http://www.w3.org/2010/02/convapps/Papers/The_Future_of_Advanced_Dialogue_Applications.pdf, 2010.

[124] Tumuluri, R.: "Extension of SRGS and SISR", W3C (online), http://www.w3.org/2010/Talks/0618-convapps-ka/, 2010.

[125] Hori, C.: "Use Cases and Requirements for New Models of Human Language to Support Mobile Conversational Systems", W3C (online), http://www.w3.org/2010/02/convapps/Papers/PositionPaper2NICTpdf.pdf, 2010.

[126] Tumuluri, R.: "Extension of SRGS and SISR", Welcome from the W3C, W3C (online), http://www.w3.org/2010/Talks/0618-convapps-ka/, 2010.

[127] Fuqua, K.: "Conversational Architecture Requirements", W3C (online), http://www.w3.org/2010/02/convapps/Papers/ConvArchReqs.pdf, 2010.

[128] Akolkar, R.: "Beyond the Form Interpretation Algorithm", W3C (online), http://www.w3.org/2010/02/convapps/Papers/ConvApps10.pdf, 2010.

[129] Salsman, J.: "Asynchronous Microphone Upload", W3C (online), http://www.w3.org/2010/02/convapps/Papers/asynchMicUpload.pdf, 2010.

[130] Ashimura, K. et al.: "Workshop on Conversational Applications - Summary", W3C (online), http://www.w3.org/2010/02/convapps/summary.html, 2010.

[131] Ashimura, K. et al.: "Workshop on Rich Multimodal Application Development - Agenda", W3C (online), https://www.w3.org/2013/07/mmi/agenda.php, 2013.

[132] Dahl, D.: "Natural Language Processing for Sentiment Analysis Using the MMI Architecture", W3C (online), https://www.w3.org/2013/07/mmi/slides/Dahl1.pdf, 2013.

[133] Teixeira, A. et al.: "W3C MMI Architecture as a Basis for Enhanced Interaction for Ambient Assisted Living", W3C (online), http://www.w3.org/2013/07/mmi/papers/Teixeira.pdf, 2013.

[134] Dahl, D.: "MMI Architecture", W3C (online), https://www.w3.org/2013/07/mmi/slides/Dahl2.pdf, 2013.

[135] Barnett, J.: "SCXML", W3C (online), https://www.w3.org/2013/07/mmi/slides/Barnett.pdf, 2013.

[136] Johnston, M.: "EMMA", W3C (online), https://www.w3.org/2013/07/mmi/slides/Johnston.pdf, 2013.

[137] Ashimura, K.: "Related standards and technologies", W3C (online), http://www.w3.org/2013/Talks/0722-web+device-ka/, 2013.

[138] Umejima, M. and Isshiki, M.: "Smart house strategy empowered by ECHONET Lite", W3C (online), http://www.w3.org/2013/07/mmi/slides/Umejima.pdf, 2013.

[139] Campbell, C.: "Interchangeable Modalities", W3C (online), http://www.w3.org/2013/07/mmi/papers/Campbell.pdf, 2013.

[140] Banski, H.: "W3C Workshop on Rich Multimodal Application Development", W3C (online), http://www.w3.org/2013/07/mmi/slides/Banski.pdf, 2013.

[141] Whysel, N. and Corwin, B.: "W3C Rich Multimodal Workshop", W3C (online), http://www.w3.org/2013/07/mmi/slides/Whysel.pdf, 2013.

[142] Aoki, R.: "Trimming and structuring information on a Web browser", W3C (online), http://www.w3.org/2013/07/mmi/slides/Aoki.pdf, 2013.

[143] Dooner, B.: "iVVi Reader Mobile App", W3C (online), http://www.w3.org/2013/07/mmi/slides/Dooner.pdf, 2013.

[144] Einstein, M. and Ng, S.: "sTVe: Synchronized TV Ecosystem", W3C (online), http://www.w3.org/2013/07/mmi/slides/Ng.pdf, 2013.

[145] Yau, W. and Ng, S.: "Multimodal Interaction and TV", W3C (online), http://www.w3.org/2013/07/mmi/slides/Yau.pdf, 2013.

[146] Kozaki, Y.: "Large-scale system of persistently connected devices using WebSocket", W3C (online), http://www.w3.org/2013/07/mmi/slides/Kozaki.pdf, 2013.

[147] Tumuluri, R: "Discovery and registration of components", W3C (online), https://www.w3.org/2013/07/mmi/slides/Rodriguez.pdf, 2013.

[148] Johnston, M.: "Future EMMA use cases", W3C (online), https://www.w3.org/2013/07/mmi/slides/Johnston2.pdf, 2013.

[149] Ashimura, K.: "MMI.next", W3C (online), http://www.w3.org/2013/Talks/0723-mmi-next-ka/, 2013.

[150] Rosenberg, P.: "TV Everywhere", W3C (online), https://www.w3.org/2013/07/mmi/slides/Rosenberg.pdf, 2013.

[151] Ashimura, K. et al.: "Workshop on Rich Multimodal Application Development - Summary", W3C (online), https://www.w3.org/2013/07/mmi/summary.php, 2013.

[152] Ashimura, K.: "Multimodal Interaction Working Group Charter", W3C (online), http://www.w3.org/2011/03/mmi-charter.html.

[153] Kliche, I., Kharidi, N. and Wiechno, P.: "MMI interoperability test report", W3C (online), http://www.w3.org/TR/mmi-interop/.

[154] Ashimura, K. et al.: "Multimodal Interaction Framework Towards Transparent and Smarter Integration of the Web and CE Devices", IPSJ SIG Technical Report (2011-CDS-2 No.17), 2011.9.

[155] Ashimura, K., Komatsu, K. and Isshiki, M.: "Implementing Multimodal Interaction Framework for Transparent and Smarter Integration of the Web and CE Devices", IPSJ SIG Technical Report (2012-CDS-3 No.22), 2012.1.

[156] Ashimura, K., Komatsu, K. and Isshiki M.: "Implementing Multimodal Interaction Framework for Transparent and Smarter Integration of the Web and CE Devices", IPSJ Transaction Vol.2, No.2 (CDS), pp.19–28, July 2012.

List of Publications

Journal Paper

1. Ashimura, K., Komatsu, K. and Isshiki., M.: Implementing Multimodal Interaction Framework for Transparent and Smarter Integration of the Web and CE Devices, IPSJ Transaction on Consumer Devices and Systems Vol.2 No.2 19-28 (July 2012), 2012.7.

Invited Journal Article

1. Ashimura, K.: The Open Web Platform: Impacts of Web Technologies including HTML5 on Smart TVs, ITE Journal Vol.67 No.2 pp.92-97, 2013.2.

International Conferences

1. Ashimura, K., Hirai, T., Sobajima, Y., Fukuroya, T. and Tenpaku, S.: SPEAKS: A NEW SPEECH SYNTHESIZER –USING EXTENDED CV AS SPEECH UNIT–, Proc. WESTPRAC VII, pp.417-1–417-2, 2000.10

2. Ashimura, K., Kashioka, H. and Campbell, N.: Estimating speaking rate in spontaneous speech from Z-scores of pattern distributions, Proc. INTERSPEECH, Jeju Island, South Korea, 1433-1436, 2004.10

Domestic Conferences/Meetings

1. Ashimura, K. and Tenpaku, S.: A New Corpus-based Speech Synthesis Method using Extended CV as Speech Unit, 2-1-1, pp.175-176, ASJ 2000 Autumn, 2000.9.

2. Ashimura, K., Hirai, T., Sobajima, Y., Fukuroya, T. and Tenpaku, S.: SPEAKS: A New Speech Synthesizer - using Extended CV as Speech Unit -, IPSJ SIG Technical Report (2002-SLP-40-19), pp.109-114, 2002.2.

3. Ashimura, K., Campbell, N. and Takeda, K.: Analysis of Utterance Timing in Everyday Conversation, IPSJ SIG Technical Report (2002-SLP-40-19), pp.109-114, 2002.2.

4. Ashimura, K. and Campbell, N.: Telephone Dialogue Data Base of JST/CREST Expressive Speech Processing Project, 3C5-11, JSAI 2002, 2002.5.

5. Ashimura, K. and Campbell, N.: Analysis of Utterance Timing based on Speech Rate, 3-10-10, pp.347-348, ASJ 2002 Autumn, 2002.9.

6. Ashimura, K. and Campbell, N.: Analysis of Utterance Beginning Position based on Addressee's Utterance - Using Japanese Frequent Expressions /hai/ and /uN/ - 3-8-4, pp.275-276, ASJ 2003 Autumn, 2003.9.

7. Ashimura, K. and Campbell, N.: Speech Rate of Similar Speech Pattern in Dialogue Speech, 1-7-9, pp.229-230, ASJ 2004 Spring, 2004.3.

8. Ashimura, K., Isshiki, M., Nakata, J., Nakajima, H. and Komatsu, K.: Multimodal Interaction Framework Towards Transparent and Smarter Integration of the Web and CE Devices, IPSJ SIG Technical Report (2011-CDS-2 No.17), 2011.9

9. Ashimura, K., Komatsu, K. and Isshiki, M.: Implementing Multimodal Interaction Framework for Transparent and Smarter Integration of the Web and CE Devices, IPSJ SIG Technical Report (2012-CDS-3 No.22), 2012.1

Events at FIT Forum

1. 新田恒雄, 芦村和幸, 甘粕哲郎, 荒木雅弘, 桂田浩一, 西本卓也: FIT2006 イベント企画「音声マルチモーダル対話記述とその標準化」, 情報科学技術フォーラム 2006, http://www.ipsj.or.jp/10jigyo/fit/fit2006/fit2006program/html/event/event.html#16, 2006.9

2. 芦村和幸, 荒木雅弘, 桂田浩一, 西本卓也, Felix Sasaki, Michal Smith: FIT2008 イベント企画「ユビキタス Web ーこれからの Web のために必要な技術は何かー」, 情報科学技術フォーラム 2008, http://www.ipsj.or.jp/10jigyo/fit/fit2008/fit2008program/html/event/event.html#19, 2008.9

3. 芦村和幸, 荒木雅弘, 桂田浩一, 五寶匡郎, 高木悟, 藤沢淳: FIT2009 イベント企画「マルチモーダル Web ―いつでも，どこでも，そして誰もが透過的に Web 上の情報にアクセスするために」, 情報科学技術フォーラム 2009, http://www.ipsj.or.jp/10jigyo/fit/fit2009/fit2009program/html/event/event.html#13, 2009.9

Contribution to W3C's International Standardization as the Activity Lead for Web&TV, MMI and Voice

W3C Recommendation Track Documents

1. EMMA: Extensible MultiModal Annotation markup language:
   http://www.w3.org/TR/emma/
2. Ink Markup Language (InkML):
   http://www.w3.org/TR/InkML/
3. MMI Authoring:
   http://www.w3.org/TR/mmi-auth/
4. Semantic Interpretation for Speech Recognition (SISR) Version 1.0:
   http://www.w3.org/TR/semantic-interpretation/
5. Voice Extensible Markup Language (VoiceXML) 2.1:
   http://www.w3.org/TR/voicexml21/
6. Pronunciation Lexicon Specification (PLS) Version 1.0:
   http://www.w3.org/TR/pronunciation-lexicon/
7. Speech Synthesis Markup Language (SSML) Version 1.1:
   http://www.w3.org/TR/speech-synthesis11/
8. Voice Extensible Markup Language (VoiceXML) 3.0:
   http://www.w3.org/TR/voicexml30/
9. Voice Browser Call Control: CCXML Version 1.0:
   http://www.w3.org/TR/ccxml/
10. Multimodal Architecture and Interfaces:
    http://www.w3.org/TR/mmi-arch/
11. EMMA: Extensible MultiModal Annotation markup language Version 1.1:
    http://www.w3.org/TR/emma11/

12. Emotion Markup Language (EmotionML) 1.0:
    http://www.w3.org/TR/emotionml/

13. State Chart XML (SCXML): State Machine Notation for Control Abstraction:
    http://www.w3.org/TR/scxml/

Group Notes

1. Common Sense Suggestions for Developing Multimodal User Interfaces:
   http://www.w3.org/TR/mmi-suggestions/

2. Authoring Applications for the Multimodal Architecture:
   http://www.w3.org/TR/mmi-auth/

3. Use Cases for Possible Future EMMA Features:
   http://www.w3.org/TR/emma-usecases/

4. Best practices for creating MMI Modality Components:
   http://www.w3.org/TR/mmi-mcbp/

5. MMI interoperability test report:
   http://www.w3.org/TR/mmi-interop/

6. Vocabularies for EmotionML:
   http://www.w3.org/TR/emotion-voc/

7. Registration & Discovery of Multimodal Modality Components in Multimodal Systems:
   http://www.w3.org/TR/mmi-discovery/

8. Requirements for Home Networking Scenarios:
   http://www.w3.org/TR/hnreq/

Editor's Drafts

1. MPTF Requirements for Adaptive Bit Rate Streaming:
   http://dvcs.w3.org/hg/webtv/raw-file/tip/mpreq/adbreq.html

2. MPTF Requirements for Content Protection:
   http://dvcs.w3.org/hg/webtv/raw-file/tip/mpreq/cpreq.html

Chairing W3C Workshops

1. Workshop on Internationalizing the Speech Synthesis Markup Language (SSML) in Beijing (2005.11):
   http://www.w3.org/2005/08/SSML/ssml-workshop-agenda.html

2. Workshop on Internationalizing the Speech Synthesis Markup Language (SSML) in Crete, Greece (2006.5):
   http://lists.w3.org/Archives/Public/www-voice/2006JulSep/0000.html

3. Workshop on Internationalizing the Speech Synthesis Markup Language (SSML) in Hyderabad, India (2007.1):
   http://www.w3.org/2006/10/SSML/minutes.html

4. Workshop on W3C's Multimodal Architecture and Interfaces in Fujisawa (2007.11):
   http://www.w3.org/2007/08/mmi-arch/summary.html

5. Workshop on Speaker biometrics and VoiceXML 3.0 in Menlo Park (2009.3):
   http://www.w3.org/2008/08/siv/summary.html

6. Workshop on Conversational Applications in New Jersey (2010.6):
   http://www.w3.org/2010/02/convapps/summary.html

7. Web on TV Workshop in Tokyo (2010.9):
   http://www.w3.org/2010/09/web-on-tv/summary.html

8. Second W3C Web and TV Workshop in Berlin (2011.2):
   http://www.w3.org/2010/11/web-and-tv/summary.html

9. Third W3C Web and TV Workshop in Hollywood (2011.9):
   http://www.w3.org/2011/09/webtv/summary.html

10. Workshop on Emotion Markup Language in Paris (2010.10):
    http://www.w3.org/2010/10/emotionml/summary.html

11. Workshop on Web-based Signage in Makuhari (2012.6):
    http://www.w3.org/2012/06/signage/summary.html

12. W3C Workshop on Rich Multimodal Application Development in New Jersey (2013.7):
    http://www.w3.org/2013/07/mmi/summary.php

Organizing the Other W3C Workshops as a Member of the Organizing Committee

1. W3C Workshop on Electronic Books and the Open Web Platform in NY (2013.2):
   http://www.w3.org/2012/08/electronic-books/rapportebook.html

2. Second W3C Workshop on Electronic Books and the Open Web Platform in Tokyo (2013.6):
   https://www.w3.org/2013/06/ebooks/report.php

Generating W3C Group Charter Documents and Group Home Pages to Manage the Standardization Activity

1. Multimodal Interaction Working Group Charter:
   http://www.w3.org/2011/03/mmi-charter.html

2. Multimodal Interaction Working Group Home Page:
   http://www.w3.org/2002/mmi/

3. Voice Browser Working Group Charter:
   http://www.w3.org/2012/01/voice-charter.html

4. Voice Browser Working Group Home Page:
   http://www.w3.org/Voice/

5. Web and TV Interest Group Charter:
   http://www.w3.org/2010/09/webTVIGcharter.html

6. Web and TV Interest Group Home Page:
   http://www.w3.org/2011/webtv/

Misc

Japanese Government's Study Groups

1. 総務省 次世代ブラウザ Web and TV に関する検討会 オブザーバ, 2010 年 11 月 16 日-

2. 総務省 次世代 Web ブラウザのテキストレイアウトに関する検討会 代理参加, 2010 年 11 月 16 日-

3. 総務省 次世代ブラウザ Web and TV に関する検討会 作業部会 オブザーバ, 2011 年 1 月 7 日-

4. 経済産業省 調査委託事業: 平成 24 年度 我が国情報経済社会における基盤整備 (テレビのネットワーク化に研究), 次世代テレビに関する検討会 委員長, 2012 年 9 月 6 日-2013 年 2 月 28 日

5. 総務省 次世代 Web ブラウザのテキストレイアウトに関する検討会 電子書籍関連分科会 構成員, 2012 年 10 月 16 日-

6. 総務省 Web と車に関する検討会 構成員, 2012 年 12 月 3 日-

7. 総務省 Web と車に関する検討会 作業部会 主査, 2013 年 4 月 11 日-

Patents

1. "Speech synthesizer", Japanese patent publication 2001-100776, March 2001.

2. "Speech waveform analyzer and pre-processor", Japanese patent registered 4213608, Nov. 2008.

ATR Technical Reports

1. “Extension of CHATR using Perl”, TR-IT-0299, March 1999.

2. “Interface Specifications of ATR-MATRIX”, March 1999.

Commercial Magazine

1. 芦村和幸: マルチメディア時代の音声合成-多言語音声合成システム CHATR, インタフェース 1998 年 8 月号, pp.109–118, CQ 出版, 1998.6