Restoring Semantics to BML Content for Data Broadcasting Accessibility

Kinji Matsumura, Yasuaki Kanatsugu, Takuya Handa, and Tadahiro Sakai

NHK Science & Technical Research Laboratories,
1-10-11 Kinuta, Setagaya-ku, Tokyo, Japan
{matsumura.k-ke,kanatsugu.y-ki,handa.t-es,
sakai.t-jk}@nhk.or.jp

Abstract. This paper proposes a method to enable visually impaired viewers to access data broadcasting content by using various accessible devices. Data broadcasting, coded in Broadcast Markup Language, has been almost inaccessible because of its usage of dynamic HTML features and lack of semantics. This method uses a template rule to transcode BML content into an accessible static description with structure and semantics. An implementation of a trial receiver system and an experiment on transcoding are described.

Keywords: Data Broadcasting, BML, Accessibility, Transcoding.

1 Introduction

Digital data broadcasting started in Japan in 2000. Although its content is coded in Broadcast Markup Language (BML)[1], no accessibility supports have been implemented so far. To improve the accessibility of digital broadcasting, we developed a method to transcode BML content into a static and structural form with sufficient semantics to make it accessible to persons with visual impairments.

The goal of this research is to develop a unified accessible platform for presenting various content types on different kinds of devices for visually impaired users, by defining and utilizing a common set of APIs and metadata. In line with this concept, this paper explains our development of a transcoding method to make BML accessible on the unified platform.

2 Background

2.1 Research Framework

The goal of our development is to realize a unified platform that provides visually impaired people with accessibility to various multimedia contents such as HTML, Flash, and data broadcasting, by using user interface methods such as voice browsers, tactile displays, and Braille displays. As the key feature, we aim to define a common set of APIs and metadata. Fig. 1 shows an overview of the unified platform and an

C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 88–97, 2007.

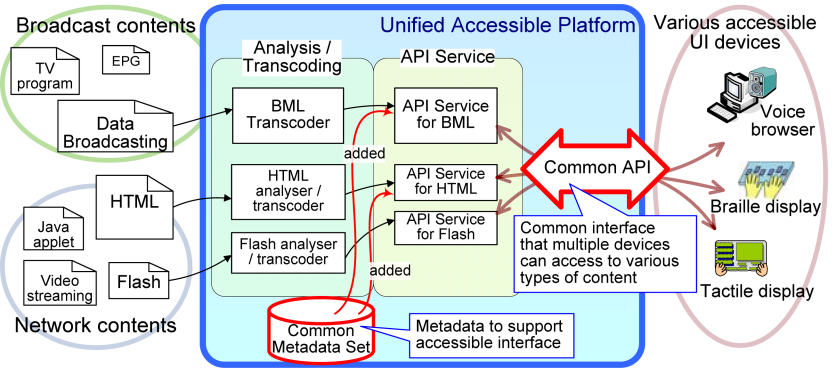

Fig. 1. Overview of unified platform

example of how different kinds of contents are provided through the common API. By using a device or software that complies with the common API, users can access different kinds of multimedia content through a uniform procedure.

2.2 Present Situation of Accessibility in Data Broadcasting Services

Data broadcasting provides a variety of information by using BML. Although several accessibility supports for TV programs such as closed captions, sub-channel commentary and speech rate conversion have been developed so far, no such support has been implemented in data broadcasting, which means that visually impaired people do not have access to data broadcasting content.

Table 1. Composition of BML specifications

XHTML 1.0	Basic structure and declaration of text and visual objects[1]
CSS level 1/2	Graphical presentation design[2]
ECMAScript	Data processing and dynamic interaction
DOM Level 1/2	Dynamic interaction
Enhancements in ARIB STD B24	Broadcast specific functionalities: Navigation with remote controller, AV stream/event handling, and receiver functionalities such as Rec. reservation, persistent memory read/write, etc.

As shown in Table 1, BML is based on Dynamic HTML technology, and it has enhanced functionalities for broadcast use, which is prescribed in ARIB STD-B24 [1]. In an ongoing service, though, the user interface design is restricted under the premise that content is presented only on a TV screen and users interact with the content by using a remote equipped with four-way arrow keys and an OK button. For this reason, the available XHTML tags and CSS properties are limited in actual operations.

[1] Available tags are strictly limited in actual operations.
[2] A fixed layout using absolute positioning is required in actual operations.

3 Requirements Analysis

3.1 Requirements of User Interface and Basic Concept of Common API

As the first step of our study, we derived the fundamental functionalities that the platform should provide via the common API. We assessed the requirements of visually impaired users in regard to accessing general multimedia content with various user interface methods, particularly considering devices with different characteristics such as Braille displays [2] and 2D tactile displays [3] as well as ordinary voice browsers. The requirements are as follows:

- *Content should be fragmented into information units, and the units should form a tree or a table structure reflecting the content's essential structure.* Content should be provided in a versatile form in order to allow various devices to present content in an appropriate style utilizing the device's characteristics and abilities. Hence, the common API should be designed to provide information in a form whereby content is fragmented into information units and is structured according to the content's original semantic structure or navigational structure. Also, the structure should be static with no dynamic alterations, to ensure that the user understands it.
- *Semantics should be attached to each content fragment.* Semantics such as meaning and role should be added to let devices use them in order to behave correctly. Existing metadata such as [4] should be utilized.
- *Alternative or descriptive annotations should be added.* An alternative textual description should be added to visual content or to long and complicated content.

Based on these fundamental requirements, we defined the basic concept of a common API. The API is a simple set of commands to provide content according to the content's tree/table structure with semantic and alternative descriptions. The user's devices call these APIs to navigate through the content, acquire information fragment, and then present the content in a manner appropriate for the user agent's characteristics. Note that we do not describe the detailed specifications of the API in this paper.

On the other hand, this concept becomes a functional requirement for the transcoding process (see Fig. 1 – analysis / transcoding layer) that makes multimedia content accessible via the common API. Such a process depends on each content format. Many studies have been done on such technologies for WWW content formats.

3.2 Technical Issues of BML Content

We analyzed the characteristics of actual broadcast BML contents to determine issues affecting accessibility. We found the following problems specific to BML:

A. *Very poor structural and semantic information due to tag usage restriction.* As mentioned above, BML content has very little structural and semantic information. Tags are mostly used for declaration and layout of visual objects. Therefore it is difficult to extract structural and semantic information from tags.

B. *Most content is transmitted in an external data file.* Most of the essential information is stored and transmitted in an external data file named Binary Table. A BML document file only has a description of how it is presented, such as CSS layouting and ECMAScript code for dynamic UI behavior and data processing. Also, in many cases, the data in the Binary Table are processed in ECMAScript before being presented. As shown in Fig. 2, for example, in order to present personalized weather forecast, the script code searches a Binary Table by referring to the user's preset postal code, and then the retrieved value is used as a reference to display an image that represents the corresponding forecast.

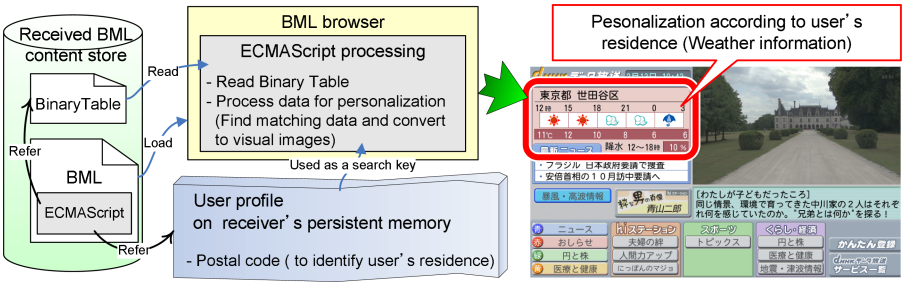

Fig. 2. Example of Binary Table data processing

C. *Input event handling specifically designed for remote controller UI.* In DHTML and Flash, user-input events are usually linked to corresponding visual objects on the screen, so the target of the user's operation can be identified by observing these objects. But in several BML contents, user inputs from a remote are directly processed with a script code. In such cases, an object that looks visually focused does not correspond to the browser's internal focused state. For example, in a content shown in Fig. 2, all 'onkeydown' events are obtained by only one div element and script processing handles visual focus-moving presentation.

D. *Heavy use of custom GUI – especially for page-flip GUI.* Custom GUIs such as tab and pop-up menu are used. Notably, page-flip GUIs (Fig. 3) are used in many cases. In a page-flip GUI, temporal fragmentation of content on the same DOM tree node prevents content analysis and structuring.

These characteristics also lead to an increase in the size of script code in BML. According to our research, in more than 130 of the BML documents we looked at, 66.6% of the lines are ECMAScript code. This also shows that most BML documents have characteristics that make it harder to improve accessibility.

Our analysis indicated that BML has characteristics similar to dynamic content format such as DHTML, Flash and Java applets, despite it being based on XHTML. Moreover, it is sometimes much more complex because its design is specific to a remote controller UI.

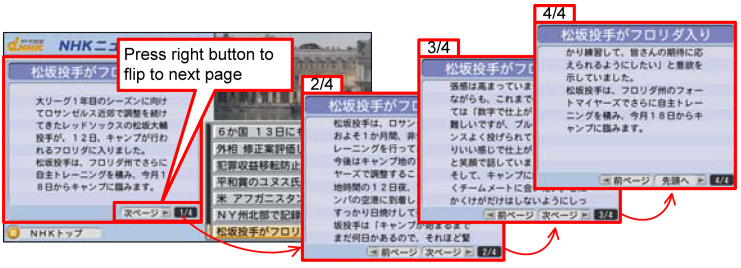

Fig. 3. Example screenshot of page-flip GUI

3.3 Broadcast Service Requirements

This section outlines the requirements specific to the broadcast service.

- *Information must be correct.* A broadcast service must provide accurate information. The possibility of transcoding errors should thus be minimized.
- *The impact on existing receivers should be avoided.* Broadcast systems are huge systems with millions of anonymous receivers. Therefore, when a new additional service is implemented, the compatibility with existing receivers must be maintained.

4 Study on Possible Solutions

We used the results of the aforementioned study to determine a feasible method to transcode BML content. We compared five possible methodologies by using the following evaluation criteria: a) quality of an expected service, b) cost for additional authoring, and c) impact on existing receivers. Criteria a) and c) are derived from broadcast service requirements and we also considered criterion b) because it is an important factor in operating actual services. We evaluated the five methodologies as follows:

1. *Static analysis and annotation for BML document.* Many of existing accessibility technologies adopt this methodology. Tags or the visual layout of content are analyzed statically and additional metadata annotations such as described in [4] are used to add semantics.
   Evaluation: a) Additional annotations can help to alleviate the problem of BML's lack of semantics. However, most BML content will not be accessible since the static analysis does not support dynamic content. b) Cost is relatively small, although it depends on the amount of additional annotations that must be prepared for each content. c) The impact is avoidable if annotations are separated from the BML content to avoid causing malfunctions in existing receivers. We also must consider how to distribute annotations.
2. *Observing the state of the browser's internal state.* In addition to the techniques described in i, the content player software (i.e. BML browser) sends its internal DOM tree state so that analyzer/transcoder software can obtain information on

dynamic content changes. The accessibility API of MSAA[5] adopts this methodology.
Evaluation: a) Observing changes in the DOM tree can solve some of the problems caused by dynamic content. However, BML has characteristics that prevent this methodology from working. In cases like 3.2 C, the user's target can not be obtained by checking the DOM tree, so the analyzer/transcoder software cannot identify which object is being manipulated. Also, in cases like 3.2 D, the acquirable data is fragmented and temporally separated. Hence, it is hard to determine the order of the fragments, and thus, content is difficult to structure. b) Almost the same as i. c) The same as i.

3. *Embedding script code for accessibility purposes.* In this methodology, a content author embeds script code for accessibility purposes. The embedded code describes the procedures so that the accessible devices can behave in a similar fashion as designed for the normal presentation devices. This methodology is generally used in combination with the techniques described in ii (see[6]).
   Evaluation: a) If all the necessary descriptions are properly coded, users can interact in almost the same manner as the content is designed for general viewers. b) The amount of code tends to increase accordingly. Because the BML document, by its nature, has complex script code, the cost for authoring and maintenance may be a serious problem. c) It is difficult to separate the original code from the additional code for accessibility. To avoid conflicts with existing receivers, conditional statements to determine the receiver's supported functionality must be written everywhere. This also leads to a cost problem.
4. *Static analysis and annotation of Binary Table.* As a new approach, we considered a methodology that applies external static annotations to a Binary Table to make it structured. Data in the Binary Table are directly used for the presentation without using a script code in the BML.
   Evaluation: a) Static annotation does not work when Binary Table data are processed before presentation. b) Binary Table has no structural information, so the annotations must describe the structure from scratch. c) Similar to i.
5. *Applying an external template with data processing to the Binary Table.* We also considered the methodology that extracts data from the Binary Table and processes it before running it down to the structured template. This does not use the original BML document either, but it is different from methodology iv in that data processing is possible.
   Evaluation: a) By using a properly authored template, it can produce similar content to the original BML because data processings such as search and text

Table 2. Comparison of possible solutions

	a) Quality	*b) Cost*	*c) Impact*
i. Static annotation to BML document	Bad	Good	Fair
ii. Dynamic observation	Fair	Good	Fair
iii. Embedded accessibility script code	Good	Bad	Bad
iv. Static annotation to BinaryTable	Bad	Fair	Fair
v. Process and structure BinaryTable	Good	Poor	Fair

processing are possible. b) Relatively high because template author must code both the structure and data processing from scratch. However, it does not have a complexity problem like that of iii. c) Almost the same as methodology i.

Table 2 summarizes the above analysis. To give a quality of service close to that of existing broadcast services, we decided that criterion a) is the most important. As a result, we chose methodology **v** for our system because only **v** meets the required quality level and has sufficient feasibility. We believe that the cost problem can be handled because the amount of broadcast content is much smaller than on the Internet.

5 Design of Transcoding Method

5.1 Template-Engine Mechanism

We designed a concrete BML transcoding scheme by adopting methodology **v**. Fig. 4 shows the sequence of transcoding. In this model, documents are transcoded according to template rules that have descriptions for data extraction, processing, and structuring. The transcoder finds and applies a matching template rule to the original BML document, and generates similar content to the original one but it is static and structured. We provisionally named the coding scheme of trancoded content Simplified BML (SBML). The SBML processor processes transcoded documents, then provides information to devices via the common API.

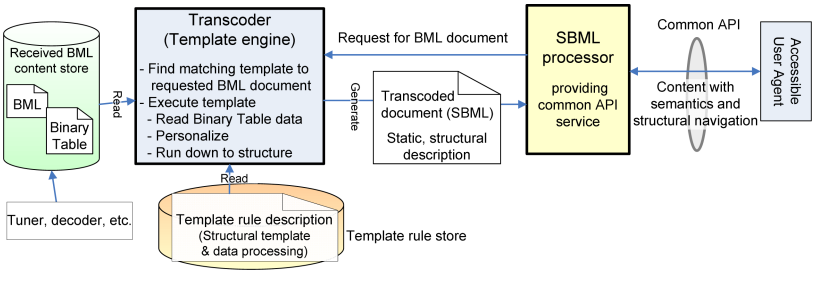

Fig. 4. Accessible BML browsing system with transcoding mechanism

The transcoding process of this model is similar to existing technologies of template engines such as PHP and JSP, although it is different in that it runs on the receiver side. Thus, in the prototyping, we used Ruby scripting language (erb) for the template description and processing.

5.2 Consideration of Authoring and Distribution of Template Rules

As with existing annotation-based technologies, these template rules depend on the original content and they must be prepared for each original BML content. To be more precise, template rules depend on the information structure of the original content, i.e., the Binary Table's data scheme and the usage of its data.

On the other hand, most BML content is automatically updated using authoring templates. Therefore, template rules can be defined by making use of the design pattern of the original authoring templates. Fig. 5 shows the proposed distribution system. By distributing these rules to receivers beforehand, receivers can transcode updating content at any time by receiving only broadcast waves. A matching template rule can be identified by the received BML's broadcast URL. When the authoring template is replaced, however, the corresponding new template rule must be redistributed by using the network or other means. The other advantage is that the existing data broadcasting system is not affected at all.

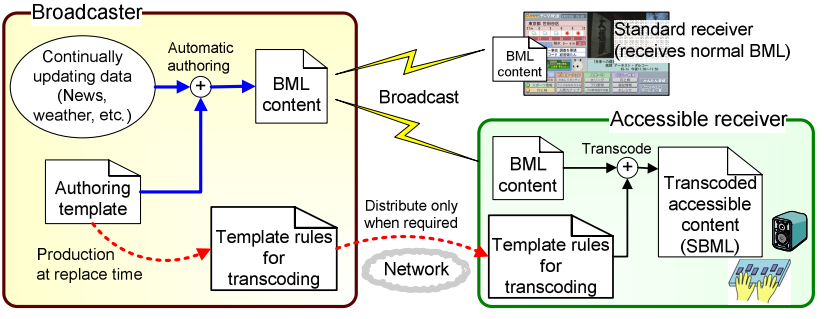

Fig. 5. Broadcast system with template rule distribution for content transcoding

5.3 Description Scheme of Transcoded Content

We also defined the description scheme of the transcoded content – SBML. The scheme must be able to describe structure and semantics. On the other hand, enhanced functionalities defined in the BML specification (such as access to persistent memory and broadcast event handling) should be available as long as the script code does not dynamically alter the DOM tree structure of the content.

For these reasons, we determined that the description scheme should be based on XHTML (including structuring tag sets) with BML enhancements, but without

Table 3. Differences between BML and SBML

	BML (currently operated)	*SBML (provisional) (for accessibility use)*
Document Structure	Mostly only p, div, object	Same as on the left. Plus, h1-h6, tables, forms, etc.
DHTML feature	Used	Not used
Presentation design	Strictly specified in content	Not specified in content
Broadcast specific APIs & tags	Available	Available
Metadata for accessible user agent	Not available	Included as compound document

DHTML features. In addition, metadata for accessibility is available to form a compound document. This can be regarded as another profile of BML, in contrast with the one used in actual operations. Table 3 shows the differences between SBML and current BML.

6 System Prototyping

We designed and implemented a prototype receiver system with the functionalities listed below:

- Receives BML contents from broadcast waves and transcodes them in real time.
- Applies corresponding template rules to each BML content.
- Provides common API service processing transcoded SBML. (The provisional specification was used for the common API to meet the requirements described in 3.1.)

We used the system to confirm the feasibility of the transcoding method. The experiment used NHK's actual broadcast content and we authored template rules for 31 BML documents, all of which generated SBML documents as we expected. Below is example code for a template rule. Codes between <% and %> are descriptions of data processing, and the rest are templates that describe structure and semantics.

```
<?xml version="1.0" encoding="euc-jp"?>
<?bml bml-version="_._"?>
<html>
<head><title>NHK Anytime News</title></head>
<body style="clut:url(news.clt)">
<%  f = BinaryTable.open("~/0011/news.btb","1,S:1V")
    rowData = f.getRowData()
    /(....-..-..-)(..)\.(..)\...\...../ =~ rowData[0]  %>
<object type="image/png" data="NHK.png" alt="anytime news" />
<%= $2 %>:<%= $3 %> updated
<%  while rowData = f.getRowData()
      if (rowData[0]=="newsreportstartmarker") %>
      <h1 id="news<%= I %>"><%= f.getRowData() %></h1>
<%  else %><%= rowData %><%
   end
end %>
<a accesskey="Y" href="/40/0000/top.bml">NHK Top Page</a>
</body>
<html>
```

We confirmed that this system can transcode received BML content into an accessible form and provide the content to devices via the common API. Also, by receiving BML contents at different times, we confirmed that a single template rule can be applied to different versions of content that are generated from the same authoring template.

7 Conclusion

We described the development of a method to make BML content accessible to visually impaired people. We analyzed the requirements and found that BML has specific problems affecting accessibility. We proposed a static pre-transcoding methodology for BML and implemented a trial system to confirm its feasibility.

We will evaluate the quality and precision of the proposed transcoding method in the near future, as well as provide a justification for the common API design. The BML transcoding method proposed in this paper can be utilized for other purposes other than improving accessibility. By transcoding BML into static and device-independent desciptions, the possibilities for versatile usage of data broadcast content will be greatly enlarged.

As a system functionality, we are planning to enhance the system so it will be able to handle broadcast events triggered from broadcasters at an asynchronous timing.

Acknowledgements. Part of this research was carried out with the support of the National Institute of Information and Communications Technology.

References

1. ARIB. ARIB STD-B24, Data Coding And Transmission Specification For Digital Broadcasting (Version 5.0) (2006) http://www.dibeg.org/aribstd/ARIBSTD.htm
2. Sakai, T., et al.: Development of an Integrated Information Receiving Terminal for Visually Impaired Persons. ICCC 2,I-2-06 172–179 (1999)
3. Sakai, T. et al.: Improving Access to Digital Broadcasting for Visually Impaired Users, HCI International (2005)
4. W3C: Role Taxonomy for Accessible Adaptable Applications Working Draft http://www.w3.org/WAI/PF/GUI/roleTaxonomy-20060508.html
5. Microsoft: Microsoft Accessibility http://www.microsoft.com/enable
6. W3C: Roadmap for Accessible Rich Internet Applications (Working Draft) (2006) http://www.w3.org/TR/aria-roadmap/