Name of Workshop: Multimedia

Title of the Paper: SMIL-Based Wireless Multimedia Gateway Architecture And Platform

Authors:

Sheng-Tzong Cheng and Kuan-Hung Lu,

Wireless Communications and Mobile Networks Laboratory,
Department of Computer Science and Information Engineering,
National Cheng Kung University,
Tainan, Taiwan.
Email: stcheng@mail.ncku.edu.tw
Tel: 06-2757575 ext. 62529
Fax: 06-2747076

Contact Author: **Prof. Sheng-Tzong Cheng**

Abstract: In June 1998 the W3C released the SMIL that enables simple authoring of interactive audiovisual presentation with difference types of media combined together. However this is based on a wired wide-band network environment. Even version 2.0 of SMIL in August 2001 that provides limited functionality on a resource-constrained device named SMIL2.0 Basic, but it still has some limits by our investigation. So that, we propose WSMIL-based Wireless Multimedia Access that includes client, gateway and server sides. We wish to integrate those frameworks to provide friendlier interface for browsing multimedia presentation on resource-constrained devices.

SMIL documents can be translated to WSMIL documents by the proposed gateway architecture with the CC/PP framework. By the WSMIL Filter in the gateway, Web-content providers only needs to provide one single SMIL document. Powerful devices (such as PC)as well as the resource-constrained devices can access and browse the contents adaptively.

Keywords: SMIL, WSMIL, WSMIL Filter, CC/PP, and Multimedia Platform.

SMIL-Based Wireless Multimedia Gateway Architecture And Platform[1]

Sheng-Tzong Cheng and Kuan-Hung Lu,

Wireless Communications and Mobile Networks Laboratory,
Department of Computer Science And Information Engineering,
National Cheng Kung University,
Tainan, Taiwan, R.O.C.
E-mail: stcheng@mail.ncku.edu.tw
Tel: 06-2757575 ext. 62529
Fax: 06-2747076

Abstract

In June 1998 the W3C released the SMIL that enables simple authoring of interactive audiovisual presentation with difference types of media combined together. However this is based on a wired wide-band network environment. Even version 2.0 of SMIL in August 2001 that provides limited functionality on a resource-constrained device named SMIL2.0 Basic, but it still has some limits by our investigation. So that, we propose WSMIL-based Wireless Multimedia Access that includes client, gateway and server sides. We wish to integrate those frameworks to provide friendlier interface for browsing multimedia presentation on resource-constrained devices.

SMIL documents can be translated to WSMIL documents by the proposed gateway architecture with the CC/PP framework. By the WSMIL Filter in the gateway, Web-content providers only needs to provide one single SMIL document. Powerful devices (such as PC)as well as the resource-constrained devices can access and browse the contents adaptively.

Keywords: SMIL, WSMIL, WSMIL Filter, CC/PP, Multimedia Platform

[1] The work is supported by the Institute of Information Industry under the contract of 91-C-42.

1. Introduction

SMIL [1] is a W3C [2] standard language that enables simple authoring of interactive audiovisual presentation with difference types of media combined together. However this is based on a wide band of wired and powerful devices. Even version 2.0 of SMIL in August 2001 that provides limited functionality on a resource-constrained device named SMIL2.0 Basic Profile (or SMIL2.0 Basic) [4], but it still has some limited by our research, that is the presentation of parallel group.

By the DTD of SMIL, the children of the par element can overlap in time. There is a semantics error in this par element for resource-constrained device. Because author may put more than one continuous media object in the par element for presentation. Although this SMIL document maybe well-formed, validity, and can be presentation in the powerful devices. But, can it be presented in resource-constrained devices?

So that, we propose Wireless Synchronized Multimedia Integration Language (WSMIL) [17] based on XML for resource-constrained devices. And propose a Wireless Multimedia Access Platform includes client, gateway, and server sides. The markup language, WSMIL, is designed according to the properties of resource-constrained devices for multimedia presentation. Author could control temporal behavior of the parallel presentation by some limit to avoid the semantics error.

To integrate this Multimedia Platform is that we wish to provide friendlier interface for multimedia presentation of resource-constrained devices. By the platform, multimedia content provider could just preparation one SMIL document and the multimedia that reference to it for any kinds of client devices. In this paper, we will focus on Gateway Architecture. Propose a par Converting Algorithm and Card-Based Synchronization for automatic SMIL to WSMIL translates. And how WSMIL Filter could work with Internet, and how to used CC/PP [11, 12] to select the appropriate items.

The remainder of this paper has the following structure: In chapter 2, describes the background and related work, include XML, SMIL and WSMIL. In chapter 3, we will explain the current technology of multimedia platform. That was a three-tiered platform with server, gateway and client sides. Chapter 4, what are the problems at SMIL for presentation in resource-constrained device? Use par Converting Algorithm and Card-Based Synchronization for translate SMIL to WSMIL and provide a friendlier interface, CGI, for client device. And how CC/PP could work with Gateway Architecture. Finally, chapter 5 presents the conclusions, as well as future ongoing work.

2. Related Work

2.1 XML

The eXtensible Markup Language (XML) [5] is a kind of markup language and a subset of SGML (Standard Generalized Markup Language). It is used to describe a class of data objects called XML documents and partially describes the behavior of computer programs which process them. Users must define all of their own tags, attributes, values, and etc. to be used in an XML document. The first recommendation version of XML is XML1.0, which was being proposed on the 10th of February in 1998.

The logical and physical structures of an XML document must nest properly. In other words, it must be *well-formed* and *validity*. First, an XML document is well-formed, if there is exactly one element, called the root or document element, no part of which appears in the content of any other element. For all other elements, if they are no empty elements, the start-tags and end-tags nest properly within each other.

In the next place, an XML document is validity, if it has an associated *Document Type Declaration* (DTD) or *XML Schema* [6], and if the document complies with the constraints expressed in it. It meaning, an element is valid if there is a declaration matching the element type declaration where the name matches the element type. The XML1.0 specification defines the concepts of well-formedness and validity; it is very simple to check a document for well-formedness, while validation requires more work but allows the user to define more powerful constraints on document structure. So that, an XML document may be both DTDless and valid, but it must be well-formedness. We also called that is a complete XML document:

For processing the XML documents, we will probably make a choice between standard tools or custom tools. As time goes on, more and more standard tools will become available in processing XML documents. There are at least two kinds of generally XML parser can be used to parse an XML document, *Tree-based parser* and *Event-based parser*. One is reading the document and creating a tree structure that represents the document. And the processing algorithm is then applied to the tree structure. For example: Document Object Model, or *DOM* [7] for short.

Another way to parse a document is to analyze the XML document as a stream of text, the processing algorithm examine the sequence of characters that comprise the XML text and raise events, such as the start and end of elements, as the components in the document are encountered. For example: Simple API for XML, or *SAX* [8] for short. There are advantages and disadvantages to each approach, depending on your overall objective.

2.2 SMIL

The Synchronized Multimedia Integration Language (SMIL [1], pronounced "smile") is a new markup language based on XML1.0 and being developed by World Wide Web Consortium (W3C) [2]. The first specification of SMIL1.0 [3] was released in June 1998 by W3C, and several implementations are available. SMIL is an easy-to-lean HTML-like language, allows the use of a text editor to write interactive multimedia presentations. It can be said to enable simple authoring of interactive audiovisual presentations, such as TV-like multimedia presentations, PowerPoint-like presentations, E-learning, Informative newscast-style presentations, and etc. to specify what should be presented when and where. Using SMIL, authors can describe the temporal behavior of the presentation, describe the layout of the presentation on a screen, and associate hyperlinks with media objects.

SMIL has a simple tag structure, and using tags to point to external media objects. Rather than defining the actual formats used to represent multimedia data, SMIL defines commands to specify whether various multimedia components should be played in parallel or in sequence. This multimedia objects can be classified into two classes: *Continuous Media*, media objects with an intrinsic duration such as video, audio; *Discrete Media*, media objects without intrinsic duration such as text, image. Each of media object will be accessed with a unique Uniform Resource Identifiers (URI). Anchors and links can be attached to visual media objects, i.e. media objects rendered on a visual abstract rendering surface.

On the 7th of August in 2001, SMIL2.0 [4] was been recommendation by W3C. The SMIL1.0 specification contains 40 pages, size A4; but the SMIL2.0 contains more than 443 pages. Since the publication of SMIL1.0, interest in the integration of SMIL concepts with the HTML and other XML languages has grown. The strategy considered for integrating respective functionality with other XML-based languages is based on the concepts of modularization and profiling. Modularization is an approach in which markup functionality is specified as a set of modules that contain semantically-related XML elements, attributes, and attribute values. And the profiling is the creation of an XML-based language through combining these modules, in order to provide the functionality required by a particular application.

SMIL functionality is partitioned into ten functional areas. Each of them consists of one or more modules. They are *Timing Modules*, *Time Manipulations Module*, *Animation Modules*, *Content Control Modules*, *Layout Modules*, *Linking Modules*, *Media Object Modules*, *Metainformation Module*, *Structure Module*, and *Transitions Modules*. These modules may be independent or complementary. W3C also provides three language profiles by using SMIL2.0 Modules: the *SMIL2.0 Language Profile*, the *SMIL2.0 Basic Language Profile*, and the *XHTML+SMIL2.0 Language Profile*.

SMIL2.0 Language Profile includes all the major modules of SMIL2.0. SMIL2.0 Basic Profile also called SMIL2.0 Basic or SMIL Basic. It is clear that not all of the SMIL2.0 features will be required on all platforms. So, SMIL2.0 Basic provides limited functionality on the resource-constrained device while allowing richer capabilities on the more capable devices. XHTML+SMIL is SMIL integration set conformant that includes the following modules: *BasicContentControl, Basic-InlineTiming, BasicMedia, BasicTimeContainers, MinMax-Timing, RepeatTiming,* and *SyncbaseTiming*. For more information about SMIL, see references [1, 3, 4, 18].

2.4 WSMIL

Wireless Synchronized Multimedia Integration Language, WSMIL [17], is an XML-based markup language that is intended for use in specifying contents and control of multimedia components for narrowband devices. WSMIL includes four major functional areas: 1) Multimedia presentation and layout. WSMIL includes video, audio, image and text support, including a variety of formatting and layout commands. 2) Deck/card organisational metaphor. Cards specify one or more multimedia components. They are grouped together into decks, and are the unit of content transmission. A WSMIL deck is similar to an SMIL page. 3) Inter-card navigation and linking. WSMIL supports navigation between cards and decks, and also supports anchored links, similar to those found in SMIL. 4) String parameterisation and multimedia state management. All WSMIL decks can be parameterised using a state model. This parameterisation allows for efficient use of network resources.

WSMIL rooted at <wsmil> element, which may contain one or zero <head> element and more than one <card> elements to be its children. Card is a unit of the display and has "class" attribute to define which cards belong to the same group and relate to each other. The card elements contain information for presentation. It implicitly defines a seq element. And the head element contains information for the properties of the document.

In the synchronization of WSMIL, it is limited to contain one video object and one audio object in <par> element. Because processing capabilities of the resource-constrained device is poor and wireless transmission may be narrowband.

To allow client devices to understand a WSMIL presentation even if some of the media objects aren' t accessible. The media object elements of WSMIL used an "altsrc" attribute specifies a link URI to a long description for the media in presentation. It is strongly recommended that all media object elements have an alt or altsrc attribute with a meaningful description especially when client device was resources-constrained device.

The main different between SMIL and WSMIL are structure module, synchronization constraint and presentation unit. These features all are designed to implement the synchronization presentation of multimedia components on the resources-constrained devices. For more information about WSMIL, see reference [17].

3. Multimedia Platform Technology

3.1 Server Side

We propose a Wireless Multimedia Access Platform include client, gateway and server sides. In our multimedia platform, Web content providers can only provide one SMIL document and it's own media objects. By gateway architecture, we can support both powerful and resource-constrained devices to get the files for multimedia presentation. Figure 3.1 shows the architecture of Wireless Multimedia Access Platform.

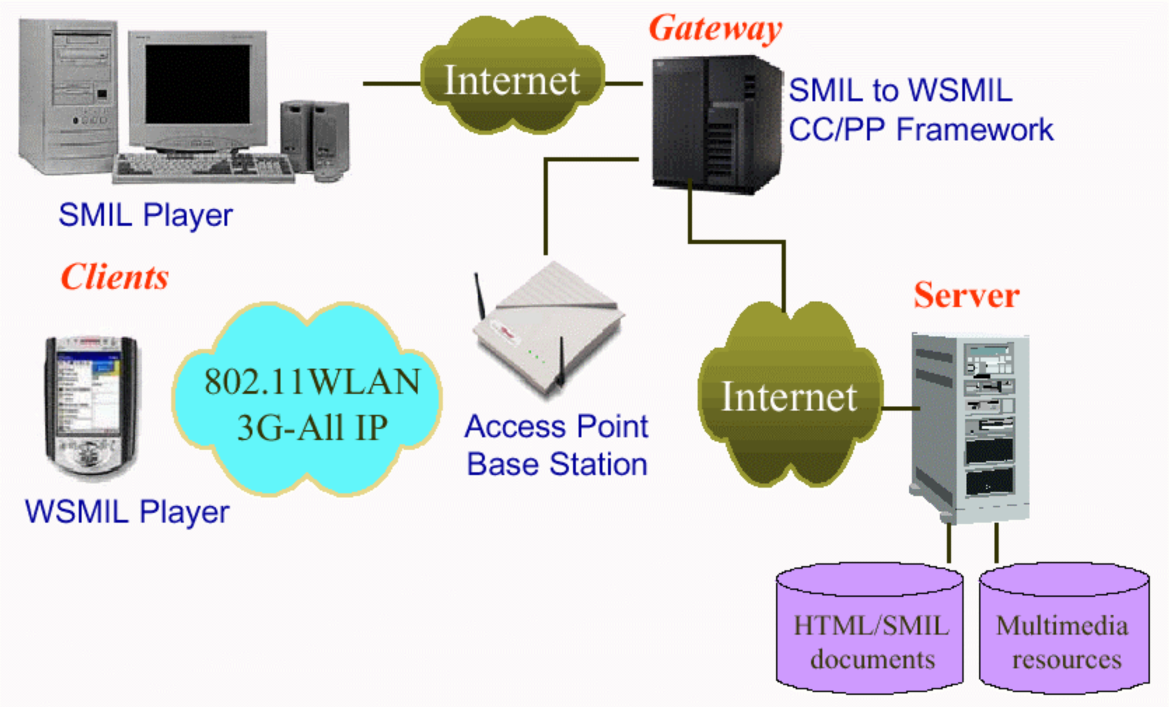


Figure 3.1: Wireless Multimedia Access Platform

In the server side, the servers can be a standard HTTP Web Server to provide SMIL documents, discrete media, and etc. or use separate Streaming Server [13] to provide continuous media for real-time presentation.

There were many kinds of SMIL authoring tools in W3C Synchronized Multimedia Home page [1]. To make a SMIL presentation we can first create the areas for the media, then fill in those areas with media objects, and finally determine the order in which to play them; in sequence, parallel or a combination of both. It is also possible to associate hyperlinks to each media object, and so on. Current SMIL authoring tools are: *GRiNS Editor by Oratrix*, *SMILGen by RealNetworks*, *Fluition by Confluent Technologies*, and etc.

3.2 Gateway Side

In wireless environment, wireless channels are less reliability and channel bandwidth varies with time. Furthermore, the heterogeneous clients also make it difficult to achieve efficiency and flexibility especially for video multicast. Although we can use Gateway Architecture to receive user capabilities, CC/PP [11, 12], and maybe dynamic video transcoding [19, 20] for the presentation of client, the available bandwidth of wireless still exist non scalable on quality of service when more and more users send the requests.

Another technology called non-real-time delivery system, Multimedia Messaging Service (MMS) [9, 15], was been proposed. It provides a rich set of content to subscribers in a messaging content and supports both sending, receiving by client devices in wireless. However, real-time messaging also exists in various forms

As given above, we wish to propose a concept called Personal Gateway Unified Notification for wireless mobile client that enable user has a personalization virtual environment [21] in Internet. In the mobile environment, figuring out how to combine different sources such as text, video, audio, and etc. to presentation is an important goal for getting personal. Mobile devices make user unlimited demand for anytime anywhere access to the content, information, and applications from Internet and Wireless. As the number and variety of devices grows, there is a corresponding increase in the need to deliver content that is tailored to the capabilities of heterogeneous mobile devices.

Here, W3C propose Composite Capability Preference Profiles (CC/PP) to describe the device capabilities and user preferences that can be used to guide the adaptation of content presented to that device. For getting personal, our gateway supports CC/PP framework that can receives the profile from client, selects the appropriate content and then sends to client. And it is also possible to use this information for multimedia transcoding to support the scalable video transport over wireless networks.

As we can see, the short textual messages (e.g., SMS messages in GSM networks) was very efficient to use and user friendly. This is because WAP makes it possible to implement services similar to the World Wide Web. Today, Multimedia Messaging Service that is a system application allow WAP client is able to provide a messaging operation with a variety of media types are growing. Think about the limited and the character of wireless, non-real-time multimedia presentation is also an interesting research area.

Use Gateway Architecture to integrate current technologies such as WAP gateway, MMS Proxy Relay, Push Proxy Gateway, user agent, notification system, and etc. to provide a virtual environment make user the intelligent unified telecommunication will be the next interesting future ongoing work. Figure 3.2 below shows an illustration of Personal Gateway Unified Notification.

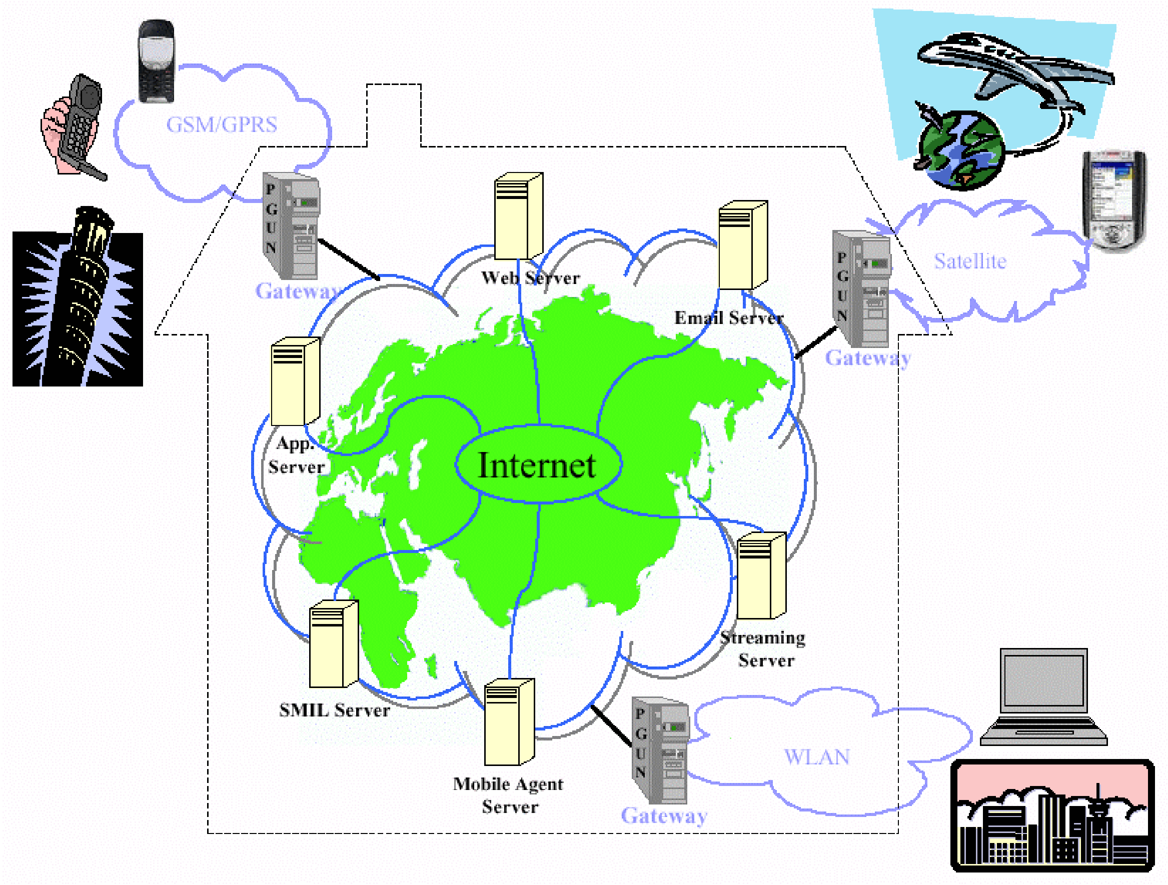


Figure 3.2: An illustration of Personal Gateway Unified Notification

3.3 Client Side

For the sake of play a SMIL or WSMIL document, client-side device needs some kind of player or browser that supports the SMIL or WSMIL format and functionality. There have been many kinds of players support SMIL documents in W3C [1]. The first commercial SMIL player is RealPlayer G2 by RealNetworks, and its latest version is RealOne that full support for the SMIL2.0 Language Profile. In addition, common players that support the presentation of SMIL document are QuickTime by Apple, Internet Explorer by Microsoft, GRiNS by Oratrix, X-Smiles by TML laboratory, and etc.

Besides we had used the Microsoft eMbedded Visual C++ 3.0, which is the Windows CE Development Tools for Pocket PC, Palm-size PC, and Handheld PC, to implement a WSMIL Player. And used the Qt/Embedded Free Edition to implement a WSMIL Player. Qt [10] is a cross-platform C++ GUI application framework. It provides application developers with all the functionality needed to build state-of-the-art Graphical User Interfaces. And it is fully object-oriented, easily extensible, and allows true component programming. The embedded version is that supported on the embedded Linux platforms with framebuffer.

The Qt XML Module provides a well-formed XML parser using the SAX2 interface and DOM Level 2. Qt DOM class provides an interface to access and change the content and structure of an XML document. It makes a hierarchical view of the document, DOM Tree, available with the root element of the XML document serving as its root. Here, we used the Qt DOM class to parse the WSMIL documents for WSMIL Player. And also used Qt/X11 for WSMIL Filter on Gateway Architecture.

4. Gateway Architecture

4.1 Problems and Solutions

As we had discussed above, there are some of problems at SMIL for presentation in resource-constrained device. The main problem is the presentation of parallel group. Besides, define root-layout for display may also have the problem. So that, we proposal WSMIL [17] for resource-constrained devices. Moreover, the document of 3GPP TS 26.234 V5.0.0 [14] also provides seven guidelines for SMIL content authors. They are:

1 General — number of media to be played simultaneous should be limited. Small devices will not be able to handle more than one video sequence at the time.

2 BasicLinking — mobile terminals will not be able to handle area elements that include multiple selectable regions within an area element. Author should not create presentations that rely on that the player can handle area elements.

3 BasicLayout — the "fit" attribute defines how different media should be fitted into their respective display regions. Here, "fit=scroll" should not be used, it is recommended to use "fit=hidden".

4 EventTiming — this module was appended over SMIL2.0 Basic by 3GPP. If the player cannot receive explicit timing value, it will have problems to initiate the event. Besides, not all terminals will support the appropriate user interactive.

5 MetaInformation — this module was appended over SMIL2.0 Basic by 3GPP. The excessive meta data will increase the file size that needs to be transferred to the mobile terminal.

6 XML entities — entities are a mechanism to insert XML fragments inside an XML document. Use of XML entities in SMIL presentation is not recommended.

7 XHTML Mobile Profile — when referring to an XHTML Mobile Profile document from a SMIL document, images and other non-text contents should be included in the SMIL document.

Note these are just the guidelines by 3GPP for SMIL content authors when they want to create SMIL documents for resource-constrained devices. That is to say, the problems may still exist.

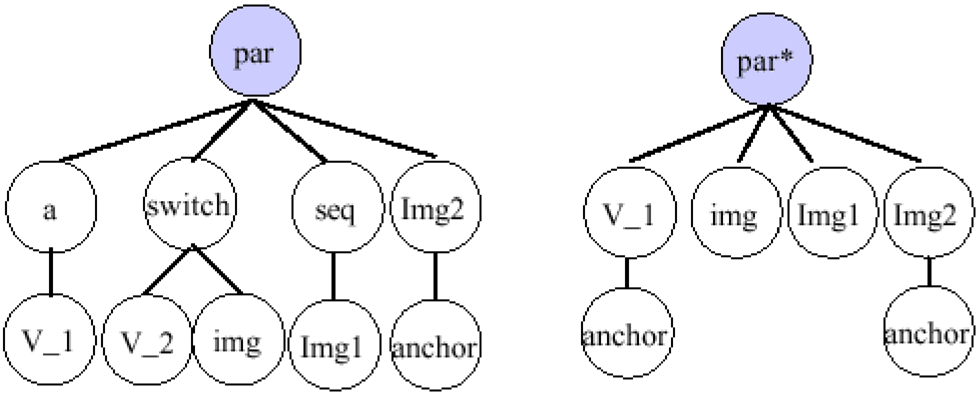
So that, we used Gateway Architecture with WSMIL Filter to automatic translation SMIL documents to WSMIL documents to handle those problems. Here, we propose a par Converting Algorithm to check the semantic errors at the par elements in SMIL documents. And used a new concept, Card-Based Synchronization that is a kind of cutting mechanism, to handle the temporal behavior in WSMIL documents by filtering. These are for *General* problem and the details will be discussed in the flowing paragraph. In the case of *BasicLinking*, we also can use a new card element to spread up all linking elements, something like each link with one image button, to provide the appropriate user interface, if mobile clients can't handle BasicLinking problem. *BasicLayout* and *MetaInformation*: it is also possible to change the attribute value and delete the meta information by WSMIL Filter. *EventTiming*: our filter was based on SMIL2.0 Basic Profile. In this assumption, we can help mobile clients to handle all of the event timing. *XML entities* and *XHTML Mobile Profile*: both of these were for hybrid documents. However if mobile clients can't support them, we also can use filter to skip them. To help all heterogeneous clients could getting themselves, here we used CC/PP framework.

4.1.1 par Converting Algorithm

In SMIL documents, the par element can contain the following four functions: Content Control with switch element, Linking with a and anchor (or area) elements, Media Object with media object elements, and Timing with par and seq elements. Because switch, par, and seq elements may appear multiple times as direct children of a par element. So, we propose par Converting Algorithm. After converting, the children of par element will only the media object elements. To find out how many continuous media object elements are there in the par element will be easily. As well as use a timestamp to find out does these continuous media objects overlap in timing to do the semantic validation.

An example for par Converting Algorithm is illustrated in figure 4.1 and 4.2 below. Figure 4.1 (a) shows a tree structure of par node before converting. And figure 4.2 (a) illustrated the detail attribute values of each element in figure 4.1 (a). Figure 4.1 (b) and figure 4.2 (b) show the results after par Converting Algorithm.

For the sake of this par Converting Algorithm we classify into four cases: ***hyperlink elements***, ***switch element***, ***synchronization elements***, and ***media object elements*** to help us converting. These four cases were described at the following.



(a): the original par node **(b): after converting**

Figure 4.1: Tree structures of par node

```
<par>
  <a href="next.smil">
    <video id="V_1" dur="10s" src=…  />
  </a>
  <switch>
     <video id="V_2" dur="10s"
            systemCPU="Pentium Ⅳ" src=… />
     <img id="img" dur="10s" src=… />
  </switch>
  <seq dur="5s">
    <img id="Img1" src=… />
  </seq>
  <img id="Img2" dur="5s" src=… >
    <anchor href="next.smil"/>
  </img>
</par>
```

```
<par>
  <video id="V_1" dur="10s" src=…  >
     <anchor href="next.smil"/>
  </video>
  <img id="img" dur="10s"  src=… />
  <img id="Img1" src=… dur="5s"/>
  <img id="Img2" dur="5s" src=… >
     <anchor href="next.smil"/>
  </img>
</par>
```

(a): the original par element **(b): after converting**

Figure 4.2: The examples of par element

- **Hyperlink Elements**

There have existed two kinds of hyperlink elements in SMIL, a and anchor (or area) elements. The a element is restricted in that it only allows associating links with a complete media object. But the anchor element allows associating links destination to spatial and temporal subparts of a media object. So that, we can say *a* is a subset of *anchor*.

Because the a element could contain par, seq, switch, and all media objects to be its children. However, the anchor element is an empty element. It has no child element and its parent may only the media object elements. As a result of this, if there existed the a element under the par element in SMIL documents, we can change the a element become the anchor element and then append to its direct children, media object elements.

- **Switch Element**

By the definition, there is only one acceptable child should be chosen at a switch element. If there isn't existed acceptable child, implementations should not arbitrarily pick an object within a switch element. Authors should order the alternatives from the most desirable to the least desirable. Furthermore, authors should place a relatively fail-safe alternative as the last item in the switch element so that at least one item within the switch is chosen. Unless; this switch element is explicitly not desired.

It is difficult to automatically detect how many media object elements, under the switch element, will be chosen to presentation, especially we have no idea what kinds of system test attributes does client device has. Since we don't know what system test attributes does client device keep own. So, we will choose a default child element. We mean it will select the fail-safe alternative element under a switch element to be our default element. If there has no fail-safe alternative element, no one will be choosing.

Additionally, to allow more flexibility in switch element selection, the system test attributes can be processed by the gateway using CC/PP negotiation. So that, gateway could choose the appropriate child element from the switch element. The CC/PP framework will be discussed at next paragraph.

- **Synchronization Elements**

SMIL2.0 Basic specifies two types of time containers, parallel and sequence. The par element allows played at the same time. And the seq element allows played from a temporal sequence. The par element may contain the following time containers: par, seq, or both to be its children. If this was happened, parent element may also determining effective time values of its children for control the temporal presentation. For example, if the effective end of the parent element is earlier than the begin time of its child, in this case the child element is not shown at all.

The par Converting Algorithm will translate the time model values from par or seq element to their direct children: media object elements. Here, we support following timing attribute value grammars: clock-values, offset-values, smil-1-syncbase-values, syncbase-values, which are defined in SMIL2.0 Timing and Synchronization modules. The clock-values include Full-clock-value, Partial-clock-value and Timecount-value.

- **Media object Elements**

For the sake for handles all hyperlink elements and synchronization elements, we used a data structure: doubly linked list. The program will preorder traversal the par node in SMIL DOM Tree. If the current node is *par*, *seq*, or *a* node we will insert a new item at the end of the linked list. And we will remove the last item from linked list automatic when the traverse leaves this node. By the way, if the current node is switch node we will choose the default child node.

When the current node is media object node. First, use CC/PP to check does system-Attribute value correct or not and then if it is continuous media node we can find out its intrinsic duration otherwise compute its simple duration. After this, we will check our linked list from last to first. Converting the values of global times to this media object node or append a new anchor child to it.

After the preorder traversal of the par node was finished, we will get the par* node which is illustrated in figure 4.1 (b). The children of par* element were been media object elements and all the timing values of them will be the explicit timing values. It is possible more easy to used this par* element to do the semantics validation than original par element. That is to say, are there more then one video object elements overlap in timing under the par* element? As the result of we used WSMIL to be the output for resource-constrained device, after the par Converting we will use Card-Based Synchronization to avoid this General problem.

4.2 SMIL to WSMIL

4.2.1 Compare WSMIL with SMIL

There have existed many kinds of SMIL authoring tools on W3C Synchronized Multimedia Home page [1]. Since our WSMIL is not yet a standard and there are some of problems at SMIL for presentation in resource-constrained device. Why not design a filter to automatic help users make WSMIL documents from SMIL documents?

Both these markup language were based on XML and conforms to the DOM Core [7]. It is friendlier to use DOM to dynamically access and create the content, structure, and style of documents. Our SMIL to WSMIL Filter will traverse and translate the DOM Tree from a SMIL document into a WSMIL document. Via using DOM to create WSMIL documents, unquestionably the WSMIL documents will be well-formed. When filter was working, we also needed to take notice of another question: validity. The main differences between WSMIL and SMIL show as following table.

Table 4.1: The main differences between WSMIL and SMIL

Differences	WSMIL	SMIL
Browsing Devices	resource-constrained devices	powerful devices
Presentation Contents	all cards	body
Presentation Unit	card	body
Optional Unit Presentation	Yes	No
Alternative Presentation	Multimedia Objects	Texts
Synchronization Presentation	constrained	unconstrained
Transmission Bandwidth	low	high

The constraint of synchronization presentation in WSMIL is par element. Both WSMIL and SMIL, the children of the par can overlap in time. In a SMIL document, the par element can contain many media objects for presentation. But in a WSMIL document, the par element only can contain one video, one audio, and other discrete media objects for presentation.

Besides, the main structure of SMIL and WSMIL was different. The main structure of SMIL is composed of the smil, head, and body elements, and is compatible with SMIL1.0 and SMIL2.0. However, WSMIL use a deck that is a collection of cards for synchronized multimedia presentation. The main structure of WSMIL is composed of the wsmil, head, and cards elements. To translate SMIL documents into WSMIL documents, we can use new card elements to maintain this functionality. Furthermore, there are many different attributes between WSMIL and SMIL. The filter also needs to consider these different attributes.

4.2.2 Card-Based Synchronization

For translate SMIL document to WSMIL document and keep up the presentation of author's definition. We propose another concept, Card-Based Synchronization that is a kind of cutting mechanism, based on the presentation of client device and how to cut down a segment from SMIL documents.

SMIL is design for integration multimedia presentation includes temporal behavior, spatial behavior, and can associate hyperlinks with media objects. However, here exist the General problem for resource-constrained devices. Card-Based Synchronization was our proposal solution. It is a kind of cutting mechanism; each par element has only one continuous media object and all of the discrete media objects will be kept back. After par Converting Algorithm, we will have the element par*. Use new card elements to handle the synchronization presentation from the original par element. Note the first child of card element must be par element; it mean we can't use the implicitly define, seq, for card element. Any other elements must append under this par element in the card element. Reserve all discrete media object elements into every new card elements and then put the redundant continuous media object elements to each new card elements. After this, we also need to append hyperlink elements for navigation.

After par Converting Algorithm, only media object elements will be reserved. Sum up these media object elements, we will know how many videos, audios, and discrete media elements do we have. Consequently, we will know how many new card elements do we need. And which discrete media object elements shall be reserved at each new card elements, which continuous media object elements shall be distributed into every new card elements by Card-Based Synchronization.

Besides, for the sake of navigation with the original par and every new card elements and can do synchronization presentation. We also needed to append hyperlink elements and "cardSync" attribute with them. More details will be explained at the next subpart.

The figures (Figure 4.3 through Figure 4.5) below show examples. Figure 4.3 shows the SMIL DOM Tree by using DOM parser. Filter will using par Converting Algorithm and Card-Based Synchronization for traverse and translate this SMIL DOM Tree and then create a new WSMIL DOM Tree.

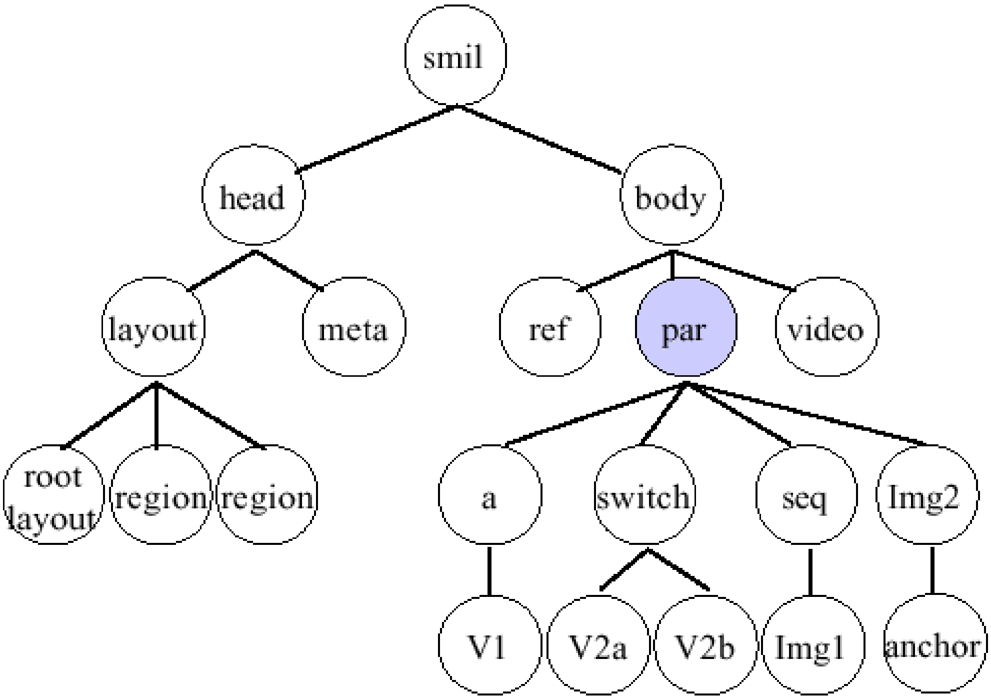


Figure 4.3: SMIL DOM Tree

If there have par nodes under the body in SMIL DOM Tree, first of all we will call par Converting Algorithm to processing. Figure 4.4 shows a tree after processing. Only media object nodes will be reserved. The a node will be changed to the anchor node. The default child node of switch node will be reserved. And the timing values of seq node will be appended to its direct child.

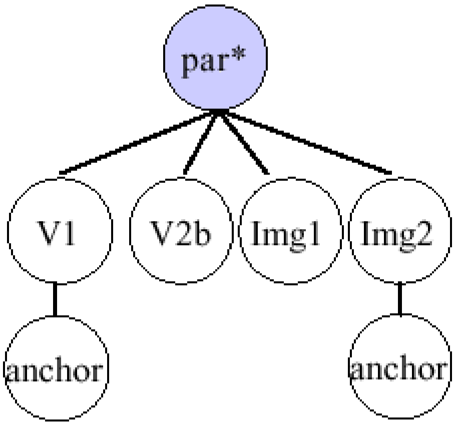


Figure 4.4: After the par Converting Algorithm

Figure 4.5 shows a WSMIL DOM Tree after filtering. We created a new card node for V2b and other discrete media nodes. The hyperlink nodes and region nodes were been appended for navigation. Each of them will have a unique identify value. Final, converts this WSMIL DOM Tree back to its textual WSMIL documents. The WSMIL Filter was finished.

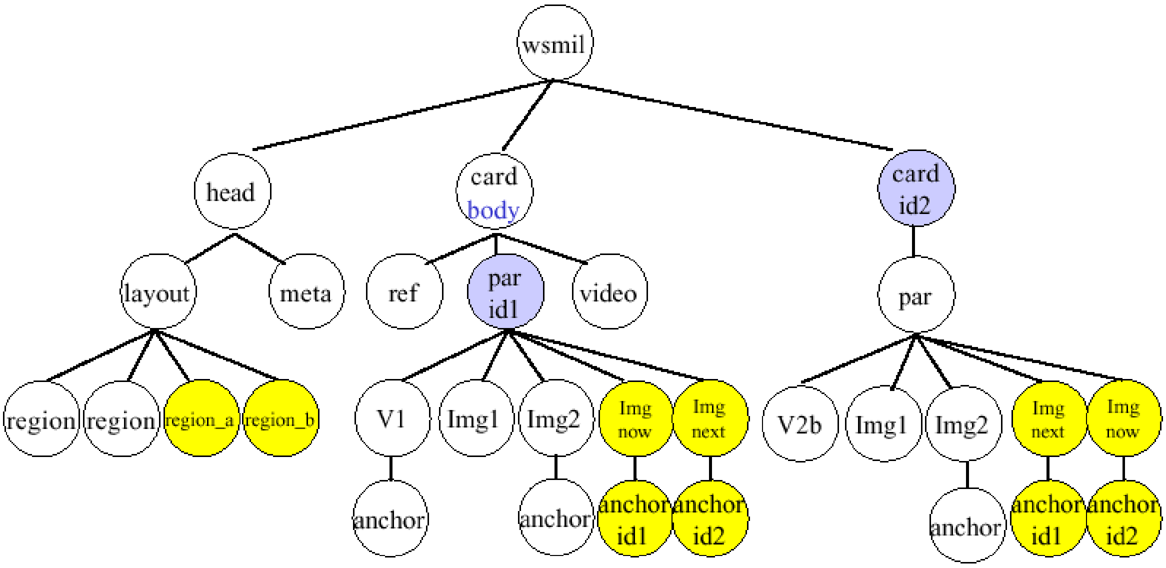


Figure 4.5: WSMIL DOM Tree

4.2.3 Navigation with par and card elements

For the sake of Card-Based Synchronization can do the synchronization presentation, we need to appended hyperlink elements under the original par element and every new card elements. All of them were needed a unique ID to identifies an element within WSMIL document. Each hyperlink activation of media objects were also needed a region attribute to specify their position.

As illustration on figure 4.5, after the traversed of the par node we needed one new card node. Furthermore, we also needed two hyperlinks activation of media object nodes and two region nodes for specify their position, size and scaling. How to append navigation links object between them? It can be described at the following two cases.

- **Hyperlink Element**

In WSMIL, we may only using anchor element under the discrete media object element for navigation. To start filtering, our gateway must provide two images in the Internet beforehand: now-image and next-image. For example above in the figure 4.5, after the traversing we needed two hyperlinks activation of media object nodes for navigation. So, we must give each par or card node a unique ID so that the hyperlink activation could linked to it for navigation.

For more friendly with end-user browsing, we used the now-image to point out the position for current presentation node. Because each of them had a unique ID, the program could know what was the current node and current hyperlink activation of media object node by this unique ID. So, we could give a correct point to it.

- **Region Element**

Our program will put every discrete media object nodes below the maximum presentation position in a WSMIL document, which have a hyperlink associate with them. When filter finish the translation of SMIL DOM Tree, we had to sum up the maximum position size from all region nodes under the head node. Note since the now-image and next-image was provided by us, the image size of them were clear. When we get the maximum size of position, we could calculate the first position for discrete media object that have hyperlink associate with it. And then we could easily calculate the next and next positions for others. All region elements were also needed a unique ID value so that the region attribute of media object elements could reference to it.

The region elements must have their immediate parent element: layout element. We must put our new region elements under the layout element in WSMIL documents. However, switch element may be used within the head element in WSMIL documents. In this case, switch element can contain layout elements to be its children. We shall put all new region elements under every layout elements of switch element. Lastly the head element may needless in WSMIL documents. In this case, if filter create new region elements it also need to create head element, layout element to been their parent element.

4.3 WSMIL Filter with Internet

4.3.1 WSMIL Filter with CGI

For the sake of provide friendlier interface for end-users browsing. WSMIL Filter worked with a Common Gateway Interface (CGI) program. So that end-users could get WSMIL documents from every SMIL documents in the Internet.

When WSMIL Filter works with a CGI program, we could help end-users translation SMIL documents to WSMIL documents from any Web content providers of Internet. If the WSMIL Player wants to see a SMIL document via gateway, just send a URL as an HTTP [16] request to the gateway. Then gateway will response an appropriate WSMIL document for browsing. Here is a sample URL as an HTTP request example:

http://Gateway.host/cgi-bin/cgi.qt?http://SMIL.host/test.smil

The first part of the URL was the absolute URL of gateway, up to and including the executable file name (cgi.qt) specifies the program to be run on the gateway side. The next part, the character "?" specifies this is a HTTP GET request method. Finally, the value following the "?" are supplied to the URL of SMIL document in an environment variable (QUERY_STRING). Only our WSMIL Filter with CGI program will try to parse it, it can be any information we like. We define this QUERY_STRING as an absolute URL to specify the SMIL documents in the Internet.

After received this QUERY_STRING, the CGI program must checked the host, port, abs_path, and fragment. For example, the host may be an IP address or a Domain Name address. The port may not the default value (port 80). The QUERY_STRING may have a fragment value, and so forth. We must parse these messages and then sending a HTTP request to multimedia server to get the SMIL document. When CGI program received the HTTP response form media server, checked the status-code so that we could decision should we translation SMIL document to WSMIL document or not. Lastly CGI program must send WSMIL document to client.

4.3.2 The Processing of Hyperlink Activation

When WSMIL Filter works with a CGI program, there are two kinds of elements we shall take care in SMIL documents. Those are media object elements and hyperlink elements. The value of their attribute "src" and "href" is a URI. It may be an absolute URI address or a relative URI address. The different URI address may not have the influence for presentation when client direct accesses WSMIL documents. However if client was getting WSMIL documents from gateway, the relative URI address will be difference. For maintain the real URI address of multimedia server, gateway must help client to translate the relative URI address to absolute URI address.

Here, we will explain what should we take care by following two cases. The first, in media object elements the attribute "src" might be a relative URI. If this happened, we must change this relative URI to absolute URI.

Table 4.2: The examples of processing src attribute for Media Object Elements

Relation URI of src	Absolute URI of src
<img src=“image.jpg”/>	<img src =“http://SMIL.host/image.jpg”/>
<video src =“./media/video.mpg”/>	<video src =“http://SMIL.host/media/video.mpg”/>

The second are hyperlink elements and their attribute "href". The attribute href contains the URI of the link’s destination. This destination may be a SMIL document, a non-SMIL document, or just a fragment identifier. If the link was followed to other SMIL document in this current document, not only change the relation URI to absolute URI but also need appending the address of gateway to it. So that, gateway can help client to get other SMIL documents and then translation to WSMIL documents for presentation.

Table 4.3: The examples of processing href attribute for Hyperlink Elements

Original URI of href	After translate of href
<a href=“smil1.smil”> <…> </a>	<a href=“http://Gateway.host/cgi-bin/cgi.qt? http://SMIL.host/smil1.smil”> <…> </a>
<a href=“non-smil.html”> <…> </a>	<a href=“http://SMIL.host/non-smil.html”> <…> </a>
<anchor href = “http://SMIL.host/smi2.smil”>	<anchor href=“http://Gateway.host/cgi-bin/cgi.qt? http://SMIL.host/smil2.smil”/>
<anchor href =“#ID1”/>	<anchor href =“#ID1”/>
<anchor href=“smil3.smil#ID2”/>	<anchor href=“http://Gateway.host/cgi-bin/cgi.qt? http://SMIL.host/smil3.smil#ID2”/>

4.3.3 WSMIL Filter with CC/PP

The Composite Capability/Preference Profiles (CC/PP) framework [11, 12] is a mechanism for describing the capabilities and preferences associated with users and user agents accessing the WWW. Information about user agents includes the hardware platform, system software, applications and user preferences. Those are intended to provide information necessary to adapt the content and the content delivery mechanisms to best fit the capabilities and preferences of the user and user agents.

Here, we will use CC/PP to provide the user information of WSMIL Player from client to gateway by HTTP protocol. So that, WSMIL Filter could choose the appropriate child element under the switch element and then returns the appropriate WSMIL document to heterogeneous client devices. Our strategy is to send a request with the absolute URI for SMIL document and profile information on the message body of POST method in HTTP protocol.

Our POST method was based on the CC/PP exchange protocol [12], which is based on the HTTP Extension Framework. An example is shows as figure 4.6. Used the "Profile" variable to describe the absolute URI of SMIL document and "ProfileDiff" variable that contains the CC/PP description of WSMIL Player. When clients send a request with the Profile and ProfileDiff on the message body in POST method, the gateway will get the SMIL document and using CC/PP descriptions to select the best appropriate items into WSMIL document and then response to clients.

Figure 4.7 below shows the process diagram of WSMIL Filter with Internet. And figure 4.8 shows more detail for the process diagram of SMIL to WSMIL. There are two swimlanes in figure 4.7 *process of Web Server* and *process of CGI*. The process of Web Server represents responsibility for client's request and response methods of Web Server. After receive client's request, Web Server will call our CGI program to do the process. Here, we support both GET and POST request methods to help client to get the SMIL document and then translate to WSMIL document. Final, the CGI program sends the WSMIL document to standard output and it will be response to client by Web Server. In general, our client is a WSMIL Player. If there is any error, we will response a WSMIL document that association with an error image to show the error was happened.

Figure 4.8 shows the sub diagram of figure 4.7 for SMIL to WSMIL. At first, we use Qt DOM parser to check and create a SMIL DOM Tree and also create a WSMIL DOM Tree of empty. Here, we use preorder traversal to search and translate SMIL to WSMIL. Final, converts the WSMIL DOM Tree to WSMIL textual document and then return to CGI program.

```
POST /cgi-bin/s2w.cgi HTTP/1.0
Content-Length: 559

Profile=http://SMIL.host/smil.smil&
ProfileDiff=<rdf>
 <Description about="HardwarePlatform">
   <systemScreenSize>320x240</systemScreenSize>
   <systemScreenDepth>32</systemScreenDepth>
   <systemCPU>ARM SA1110</systemCPU>
 </Description>
 <Description about="SoftwarePlatform">
   <systemBitrate>10000</systemBitrate>
   <systemCaptions>on</systemCaptions>
   <systemLanguage>en</systemLanguage>
 </Description>
 <Description about="BrowserUA">
   <name>WSMILBrowser</name>
   <version>CJVersion1.0</version>
   <vendor>NCKU</vendor>
 </Description>
</rdf>
```

Figure 4.6: An example of POST method with CC/PP descriptions

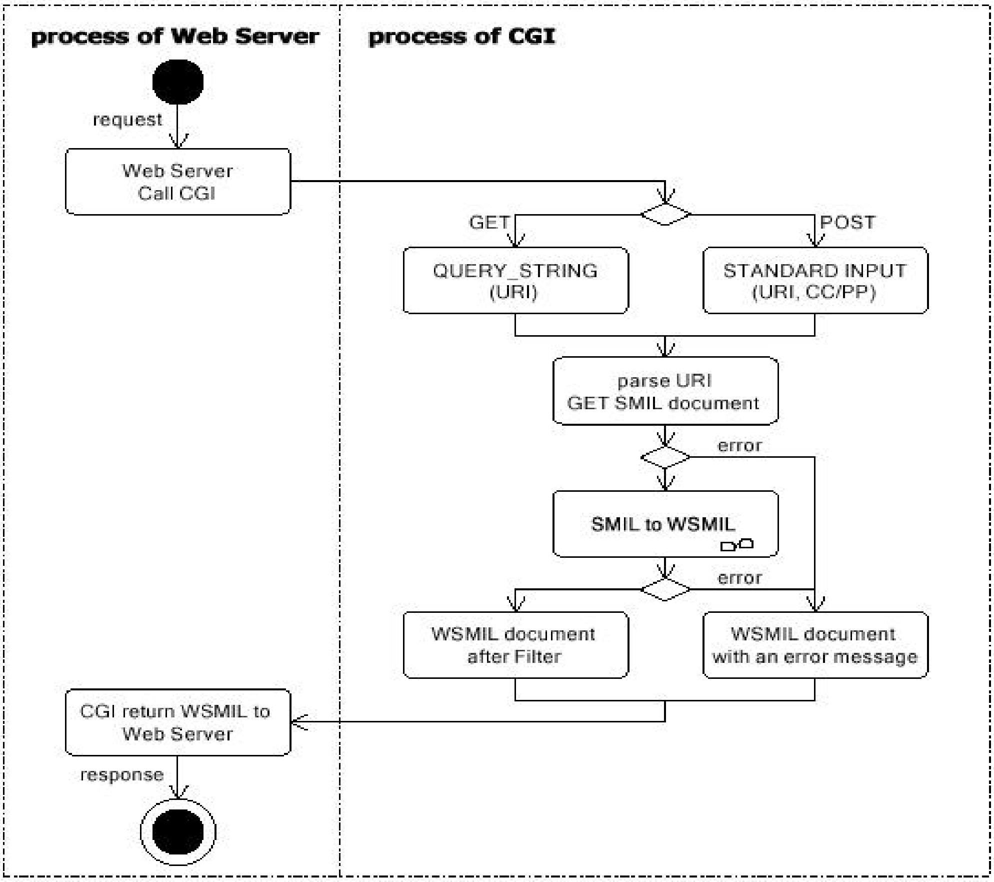

process of Web Server
process of CGI
request
Web Server Call CGI
GET
POST
QUERY_STRING (URI)
STANDARD INPUT (URI, CC/PP)
parse URI GET SMIL document
error
SMIL to WSMIL
error
WSMIL document after Filter
WSMIL document with an error message
CGI return WSMIL to Web Server
response

Figure 4.7: Process Diagram of WSMIL Filter with Internet

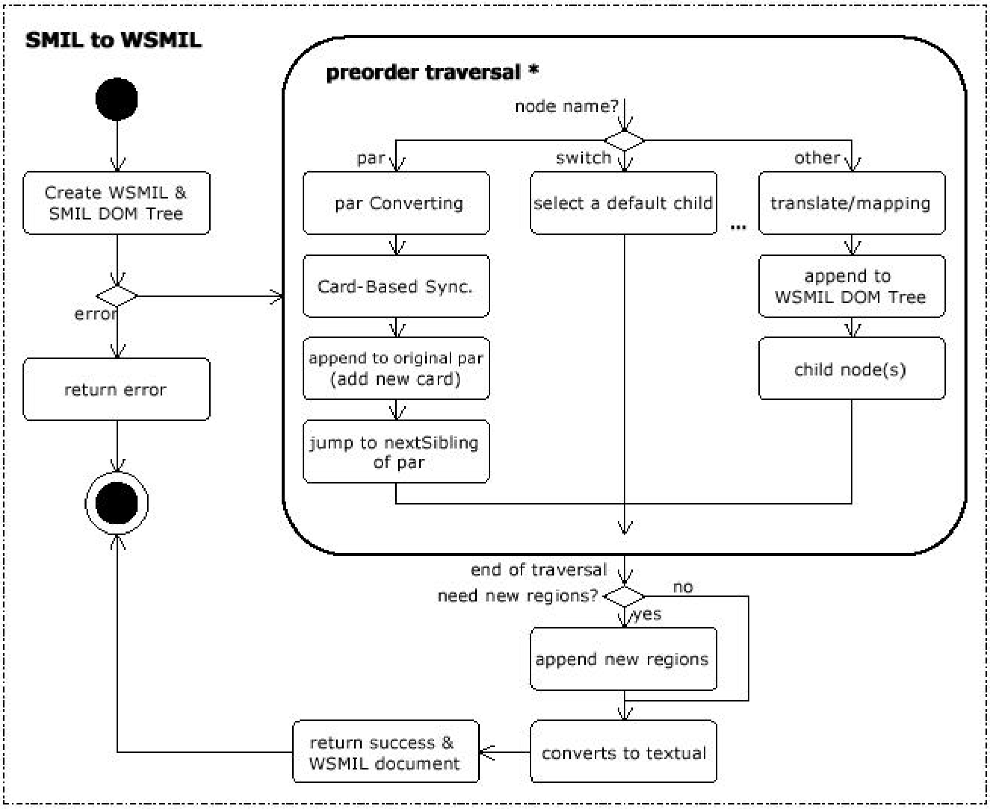


Figure 4.8: Process Diagram of SMIL to WSMIL

5. Conclusions

In this paper, we propose a Wireless Multimedia Access Platform includes client, gateway, and server sides for multimedia presentation of heterogeneous resource-constrained devices. Here, we use SMIL1.0 (or SMIL2.0 Basic) as format of the scene description, which can specify what should be presented *when* and enable control the timing with the display. SMIL is a W3C standard language and become the recommendations by 3GPP for mobile terminals. However, there are some semantic errors in SMIL for such devices. Even today's 3GPP only provides informative guidelines for SMIL content authors. But, we had already proposed WSMIL to guarantee the General problem, which limit more than one video sequence at the time.

By our Gateway Architecture, we had provided a CGI program and used WSMIL Filter, which can translate SMIL document to WSMIL document and support CC/PP, to help resource-constrained devices for multimedia presentation. Here, we propose par Converting Algorithm for semantic validation and Card-Based Synchronization that is a kind of cutting mechanism for WSMIL Filter. For heterogeneous clients, we also support CC/PP framework that enable client device to send capability profiles to gateway and then gateway will selects the appropriate item and return to it.

SMIL2.0 used modularization and profiling so that we can reuse the element packets and can combine modules into usable languages. For example, SMIL2.0 Basic Profile is a basic subset of SMIL for resource-constrained devices. Append new modules to provide other functionality for presentation such as PSS SMIL Profile of 3GPP has anticipated. To support more and more functionality will be the future works. Besides, the propagating changes to times make we difficulty to calculate the time values. It is also a future study of us.

References

【1】 W3C Synchronized Multimedia Working Group, http://www.w3.org/AudioVideo/.

【2】 The Word Wide Web Consortium (W3C), http://www.w3.org/.

【3】 W3C Recommendation 15-June-1998: Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, http://www.w3.org/TR/REC-smil.

【4】 W3C Recommendation 07-August-2001: Synchronized Multimedia Integration Language (SMIL 2.0), http://www.w3.org/TR/smil20/.

【5】 W3C Recommendation 6-October 2000: Extensible Markup Language (XML) 1.0 Second Edition, http://www.w3.org/TR/REC-xml.

【6】 XML Schema was approved as a W3C Recommendation on 2-May-2001, http://www.w3.org/XML/Schema.

【7】 Document Object Model (DOM), http://www.w3.org/DOM/.

【8】 Simple API for XML (SAX), http://www.saxproject.org/.

【9】 WAP Forum Specifications, http://www.wapforum.org/what/technical.htm.

【10】 Qt, a product of Trolltech, http://www.trolltech.com/.

【11】 Composite Capabilities/Preferences Profile Working Group Public Home Page, http://www.w3.org/Mobile/CCPP/.

【12】 W3C Note 24 June 1999, CC/PP exchange protocol based on HTTP Extension Framework, http://www.w3.org/TR/NOTE-CCPPexchange/.

【13】 3GPP TS 26.233: "Transparent end-to-end packet switched streaming service (PSS); General description".

【14】 3GPP TS 26.234: "Transparent end-to-end packet switched streaming service (PSS); Protocol and codecs".

【15】 3GPP TS 23.140: "Multimedia Messaging Service (MMS); Functional description stage 2/3".

【16】 IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1", Fielding R. et al., June 1999.

【17】 Ting-Ching Hsu "Design and Implementation of an Universal Multimedia Access Platform" Master of Computer Science And Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C., July 2001.

【18】 Åsa Viken. "Streaming: Past, Present and Future – An Investigation into the Synchronized Multimedia Integration Language 2.0 (SMIL2.0)" M. Sc. Thesis in Engineering Physics performed at Atello AB, Department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm May 2001.

【19】 Takayuki Warabino, Shinji Ota, Daisuke Morikawa, Masayoshi Ohashi, Hajime Nakamura, Hideaki Iwashita and Fumio Watanabe, "Video Transcoding Proxy for 3Gwireless Mobile Internet Access", IEEE Communications, Oct. 2000.

【20】 Dapeng Wu, Yiwei Thonas Hou, Ya-Qin Zhang, "Scalable Video Coding and Transport over Broad-Band Wireless Networks", Proc. Of IEEE, January 2001.

【21】 Gero Offer, "The New Age of IN: The Intelligent Server Farm" IEEE 2000.