A SMIL Editor and Rendering Tool for Multimedia Synchronization and Integration

Stephen J.H. Yang[1], Norman W.Y. Shao[2], Kevin C.Y. Kuo[3]
National Central University[1] National Kaohsiung First University of Science and Technology[2, 3]
jhyang@src.ncu.edu.tw[1] u9115904@cc.nkfust.edu.tw[2] u9111805@cc.nkfust.edu.tw[3]

Abstract

Along with the broadband technology matured, more and more information can be found in the Internet. To use rich multimedia to express content will be an invertible trend. How to efficient and fast integrate multimedia has been a hot research topic. In this paper, we propose a fast and efficient working environment which integrating variety of multimedia to provide an organized media content. Currently, there are many type of digital media exist on the internet. Each of them has specific format to contribute their features. To sharing and reusing rich type of resource, we use W3C recommended Synchronized Multimedia Integration Language (SMIL) to integrate these variety. Petri nets technology is introduced to enhance the temporal attributes of SMIL. The proposed multimedia rendering environment is shown as a fast generating and intuitive method.

Keywords: synchronized multimedia integration language, Petri Net, streaming media

1. Introduction

Petri nets is a systematic analysis method, it use graph, mathematic, and module to simulate and represent the system. Since Carl Carl Adam Petri proposed [1], there are many extended researches, likes Timed Petri Nets [2], Coloured Petri Nets [3], Object Composition Petri Nets [4], Multimedia, and Object Petri Nets [5]. These extended Petri nets has been applied to different application and correspondence domain to provide a completed systematic analysis.

SMIL is an extension from eXtensible Markup Language (XML) which is recommended by World Wide Web Consortium (W3C). Since 1998 the initial version 1.0 was proposed, now the current version is 2.0 in which the basic syntax has been defined. [6] The most benefit which SMIL contributed is the capability to integrate variety multimedia sources. Through SMIL player, a unified and outstanding multimedia content will be presented. SMIL syntax is similar with HyperText Markup Language (HTML) and easily for editing. Using the temporal syntax, it can provide the solution for ordering and synchronization among media which still an open issue in general homepages.

From the above description of Petri nets and SMIL, it is obviously found that these two subjects have their specific contributions, but applied to e-learning still has something need to be figured out. For example, the organization of content involved both temporal and spatial ordering and synchronization, especially in a complicated multimedia objects presentation. In our research, it is shown that the former is emphasized on workflow and analysis. It can provide a good temporal arrangement, but without detailed spatial attributes. The later is focus on integrating task among different types of multimedia, including simple layout allocation and objects sequencing. On the designing phase of learning content, it can not provide a good method to help author to control and analysis relationships among these medias, especially when a complicated sequencing and temporal scenario involved. In this paper, we will show the essentiality to integrate Petri nets and SMIL. Based on this, a new rendering tool is proposed to help author to generate a complicated content in a fast and intuitive way. This tool has been used in our content management system and facilitate on digital learning environment.

2. Related research

Before further exploit the integration between Petri nets and SMIL, we will briefly discuss the basis of Petri nets and SMIL.

2.1. Petri nets

The most benefit to using Petri nets is providing a mathematical model to represent system. It needs only few symbols to depict workflow about a system. It also can provide a visualized method to simulate the system dynamically. In this way, Petri nets provide a better representation and acknowledgement.
The definition about Petri nets is shown on Table 1.[7]

Table 1. Formal Definition of a Petri Net

A Petri nets is a 5-tuple, $PN = (P, T, F, W, M_0)$ where:
$P = \{p_1, p_2, \cdots, p_m\}$ is a finite set of places,
$T = \{t_1, t_2, \cdots, t_n\}$ is a finite set of transitions,
$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation)
$W : F \rightarrow \{1,2,3,\cdots\}$ is a weight function,
$M_0 : P \rightarrow \{0,1,2,3,\cdots\}$ is the initial marking,
$P \cap T = \varnothing$ 且 $P \cup T \neq \varnothing$
A Petri nets structure $N = (P, T, F, W)$ without any specific initial marking is denoted by *N*.
A Petri nets with the given initial marking is denoted by (N, M_0).

The basic concept of Petri nets is shown on Figure 1. The place is represented with a white circle in which marking can be stored; transition is represented with a box which indicating status shifts. The arc indicates the direction of flow. The arc is often labeled with weight, in general case with '1'. The black circle in place is the value of initial marking. It is used a number to represent the black circle more than three. For example, place may be the resource, likes video chip, text, image, or audio. The marking can be treated as a time marking.

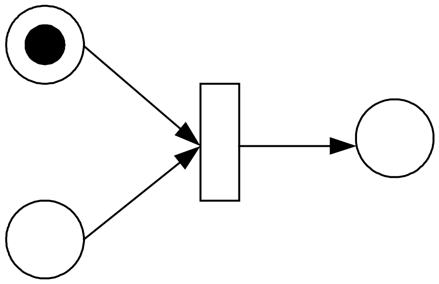

Figure 1. Petri nets Graph

There are some other transition rules (system simulation) and behavior features (system analysis), which will be studied in next phase will not be addressed in this paper.

2.2. Synchronized Multimedia Integration Language (SMIL)

SMIL is an extension markup language from XML. It is initiated from W3C. The currently version is 2.0. It provides a continuous multimedia presentation characterized with spatial and temporal synchronization of multiple integrated media which HTML limited.

In SMIL, every resource can be treated as object. Each object has its own attributes or properties and described in metadata format, likes identification, title, description, format, and etc. SMIL can be considered to provide a integrated markup platform to organize objects. The user or teacher can use SMIL to arrange the appearing order and position among objects to express the learning objectives.

In summary, SMIL has five major features, as follows: [8]

a. Media Content

SMIL is not used to create multimedia content, but to integrate existing variety of multimedia. It can transfer these multimedia content into a unitary file which can be played. We can use search engine to find out the Uniform Resource Locator (URL) of multimedia object and stored related information into multimedia databases. Through a well defined knowledge management, we can simply reuse these media objects.

b. Layout

Layout is used to coordinate the scope of content displaying. Each object can be annotated with corresponding location information either in absolute coordinate or relative coordinate with others in tag's attributes. Author can use layout to coordinate each media on the screen to achieve the overall presentation.

In our study, layout is a key factor which effects the final representation of content. This function will be designed in an intuitive mode to help author allocating each media object shown on the screen.

c. Timing

Synchronization control among media is the most important contribution for SMIL. There are two basic elements which are "sequence" and "parallel" in temporal domain. The corresponding attributes are "begin", "end", and "duration".

d. Linking

The hyperlink function is similar with HTML, SMIL can provide linkage in document and linkage among document triggered by user interaction or other triggering events.

e. Adaptively

Web content will be retrieved by people who live around the world. SMIL also provide adaptive function to let people tailor content according to characteristics such as language, preference, and device capability.

In Figure 2 shown an example of SMIL file played. The corresponding SMIL file is shown on Figure 3. It can be easily found that it has a similar syntax and structure with XML file.

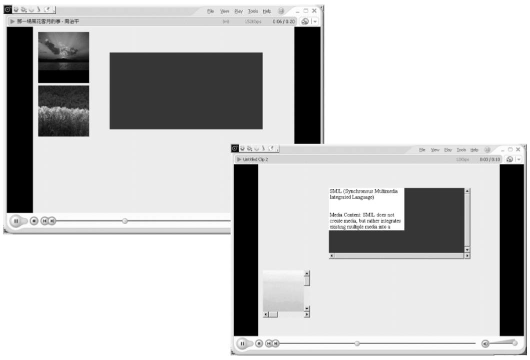

Figure 2. Example of playing SMIL file.

```
<smil>
 <head>
  <layout>
   <root-layout width="1024" height="728" background-color="yellow" />
   <region id="Sunset" left="20" top="20" width="200" height="200" fit="meet"/>
   <region id="Winter" left="20" top="230" width="200" height="200" fit="fill"/>
   <region id="Blue hills" left="20" top="450" width="200" height="200" fit="scroll"/>
   <region id="Text region" left="300" top="100" width="600" height="300"
           fit="scroll" background-color="red" />
  </layout>
  <transition id= "fade1s" type = "fade" dur="1s"/>
 </head>
 <body>

  <seq>
    <par>
      <audio src="03_那一場風花雪月的事.mp3" dur="10s" />
      <img src="Sunset.jpg" alt="The Sunset icon" region="Sunset" dur="10s"
           transIn="fade1s" transOut="fade1s"/>
      <img src="Winter.jpg" alt="The Winter icon" region="Winter" begin ="2s" dur="18s"/>
    </par>
    <par>
    <img src="Blue hills.jpg" alt="The Blue hills icon" region="Blue hills" dur="10s"/>
    <text src="test_english.txt" region="Text region" alt="Text region" dur="10s"/>
    </par>
</seq>
 </body>
</smil>
```

Figure 3. SMIL code.

In fact, SMIL can not perfectly support temporal relationships and attributes especially a complicated multimedia system involved. During rendering process, author can not have a completely picture what he/her really wanted. Author must wait for playing SMIL file then know exactly content is. It is the reason that the Petri nets is introduced in our system to proving a intuitive object management.

3. Petri Nets based Synchronized Multimedia Rendering Environment

In this section, the proposed system will be introduced. This system is built under the following environment.

3.1. Architecture

The architecture of proposed SMIL editor and rendering tool is shown on Figure 4. This system is composed of "rendering center" and "server management". In this study, we focus on "Manifestation Plan" and "Toolkit" in the rendering center. User can use Petri nets to visualize the rendering plan in both temporal and spatial domain. The system will automatically transfer the visualized content information into SMIL format. The generated SMIL file will be transmitted to Media Sever for further process. Then using the player embedded in client device to play the SMIL file.

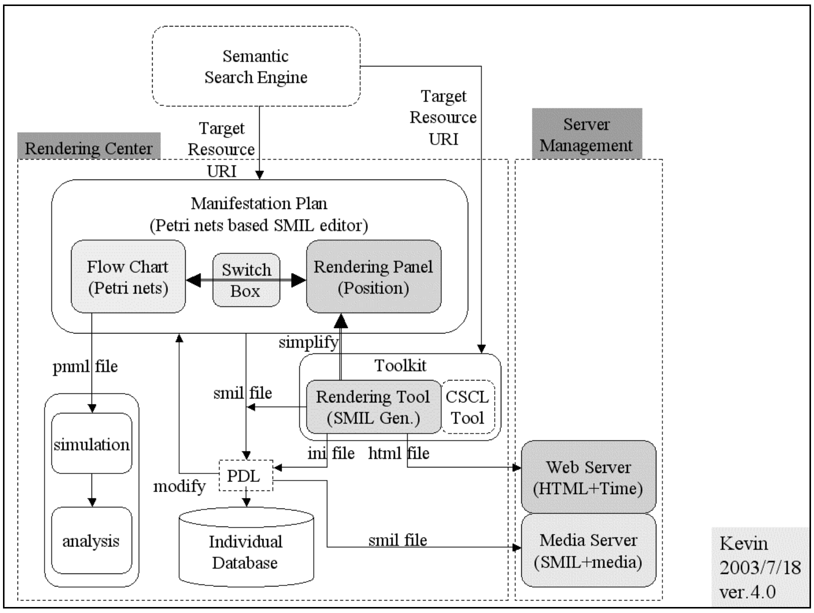

Figure 4. Petri Nets based Synchronized Multimedia Rendering Environment architecture

In Figure 4, Manifestation Plan is composed of three modules, as follows:

a. Flow chart (Petri nets)
b. Switching box
c. Rendering panel (position)

In flow chart module, it will focus on the workflow designing for content representation. Switch box is used to annotate layout, synchronization, and workflows transition of object. User can use Rendering Panel coordinate multimedia object arrangement in spatial domain.

Toolkit is composed of Rendering Tool and Computer-Support-Collaborative-Learning

(CSCL) Tool. The rendering tool will be used to integrate multimedia objects and generate unitary SMIL file. The CSCL tool will provide sharing function for multi-users on the Internet which not addressed in this paper.

3.2. Petri nets and SMIL

The fundamental of Petri nets design is shown on Figure 5. The major function include create, move, link, property and save. Simulation and Analysis will be incorporated on next phase.

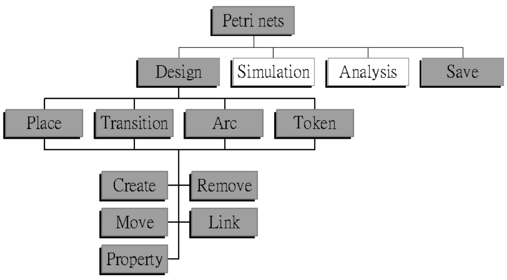

Figure 5. Petri nets design concept

From Figure 5, it is found that there are four basic elements used in Petri nets. They are the key points to establish the relationship with SMIL. So in Table 2, the feature comparison is listed.

Table 2. Summary between Petri nets and SMIL

Petri nets	Place	Transition	Arc	Token
SMIL	Object property	Scene transition	Flow path	Play status

From Table 2, Place will be extended to define the attributes of multimedia objects, likes spatial and temporal information. Transition is responsible for the switching control among workflows. Arc represents the media workflow. Token will be the status of playing which represent as a black circle.

3.3 Rendering panel

In proposed system, we focus on the layout allocation, media appear sequence, and the corresponding temporal information. In stead of inputting accurate digital values about object, an intuitive visualized object allocation will be more acceptable by user oriented. In the same time, the sequence of object and related synchronous information can be added to desired object. In this way, the time for rendering SMIL file will be reduced significantly. In Figure 6, it is shown that using SMIL editor to integrate variety of multimedia in a layout.

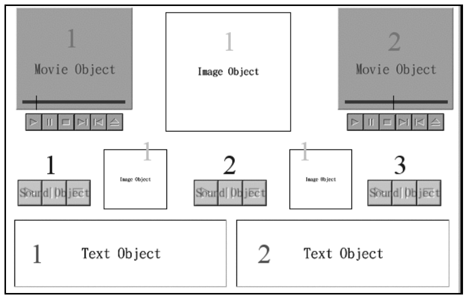

Figure 6. Example of using SMIL editor

3.4. Manifestation Plan

When we try to combine Petri nets and SMIL to form an integrated rendering environment, we need to decide which one is the dominant and coordinating mechanism. The last issue need to be resolved is how to make multiple objects which SMIL editor can render onto a single object/Place mechanism which Petri nets operated. In fact, we have successful extended this single object mechanism to a complex objects mechanism.

The rendering system architecture is shown on Figure 7 which is extracted from Figure 4. We will use Figure 7 to explain the relationship between Petri Nets based SMIL editor and Rendering Tool.

The proposed Petri nets based SMIL editor is composed of three modules. Petri nets (flow chart) will communicate with Rendering Panel through Switch Box. The Rendering Panel is a simplified single object module compared with Rendering Tool.

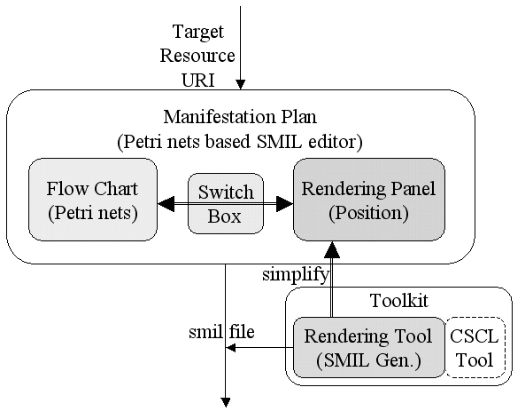

Figure 7. System architecture of Petri nets based SMIL editor

4. Implementation

In this section, Rendering Tool and Petri nets based SMIL editor will be demonstrated. The generated SMIL file will be delivered to streaming media server and played.

4.1. Rendering tool

This tool mainly provides a fast method to help user designing desired layout, then export in SMIL files. The user interface of SMIL editor is shown on Figure 8. Through this interface, user can import multimedia objects, likes video, text, image, and audio. Our system can support the file format which SMIL player supported. It also provides a real time pre-view function to let user check preliminary content status.

In this interface, we provide a intuitive method to help user operating. That is all the multimedia objects can be put on any position in the layout by drag-and-pull.

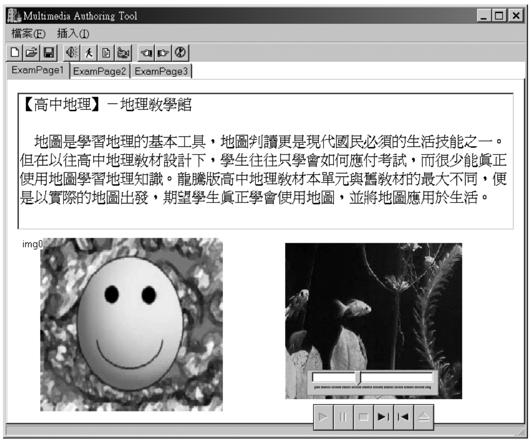

Figure 8. Interface of SMIL Editor

After finished layout design, the whole content information will be stored to object files shown on Figure 9. Then system will transform these object files to a SMIL file. In the same time, user can use pre-view function to view the result shown on Figure 10.

```
[Picture1]
URI=C:\mediareport\smile.jpg
Format=.jpg
Left=44
Top=219
Width=255
Height=242
[Document1]
URI=C:\mediareport\start.txt
Format=.txt
Left=16
Top=71
Width=694
Height=192
[Movie1]
URI=C:\mediareport\FishTank.avi
Format=.avi
Left=390
Top=281
Width=247
Height=218
```

Figure 9. Example of object file

When SMIL file generated, it will be delivered to multimedia sever. Multimedia sever will use streaming technology to play desired multimedia based on the SMIL file.

Figure 10. Example of result to Figure 9's SMIL file

4.2. Integrated Rendering Environment

In section 4.1, it is shown that using Rendering Tool to generate SMIL file. This intuitive interface only can handle simple scenario likes multimedia object in a single layout. In fact, it is not enough for user to design content to express a whole concept. It is the reason that Petri nets will be used to resolve this problem.

From above mentioned and Figure 7, Petri nets SMIL editor has three major components. In this section, the interface and function for each component will be demonstrated. From Figure 11, it is found clearly that the workflow which contains sequence and transition information of objects in content. The white circle is used to as a multimedia container. The transition effect of object is defined in the box.

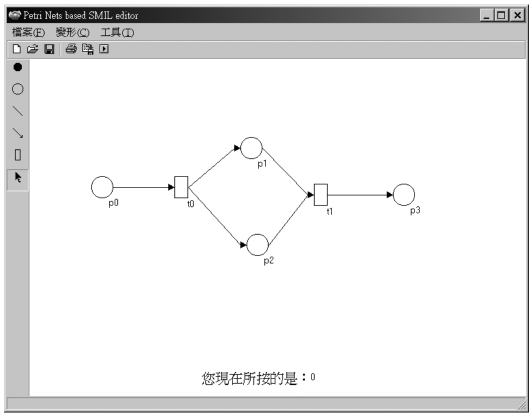

Figure 11. Petri nets graph

The switch box is used as a bridge which connects Petri nets and Rendering Panel shown on Figure 12. It will contain source, position, time, sequence, and transition information of object. In this way, the Petri nets can only focus on the workflow design. The Rendering Panel also only needs to focus on layout designing. This approach will combine the advantage from

Petri nets and Rendering Panel to achieve a perfect SMIL rendering environment.

Figure 12. Switch Box

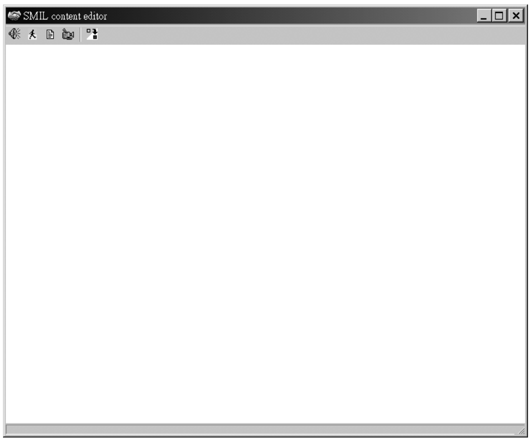

Figure 13. Rendering Panel

When completed Rendering Panel designing, the corresponding information can be shown on the Place in the Petri nets, likes type of Multimedia and temporal information, shown on Figure 14. In same way, the transition effect information will also be shown on transition box. Using Petri nets, user can easily arrange the workflow and scenario of the content.

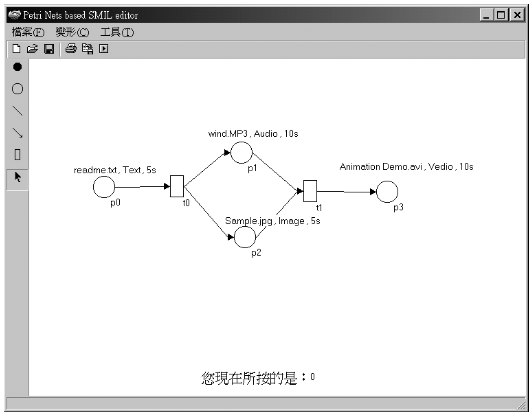

Figure 14. A rendered content shown on Petri nets module

The rendered file will be stored in SMIL format. The played result of SMIL is shown on Figure 15.

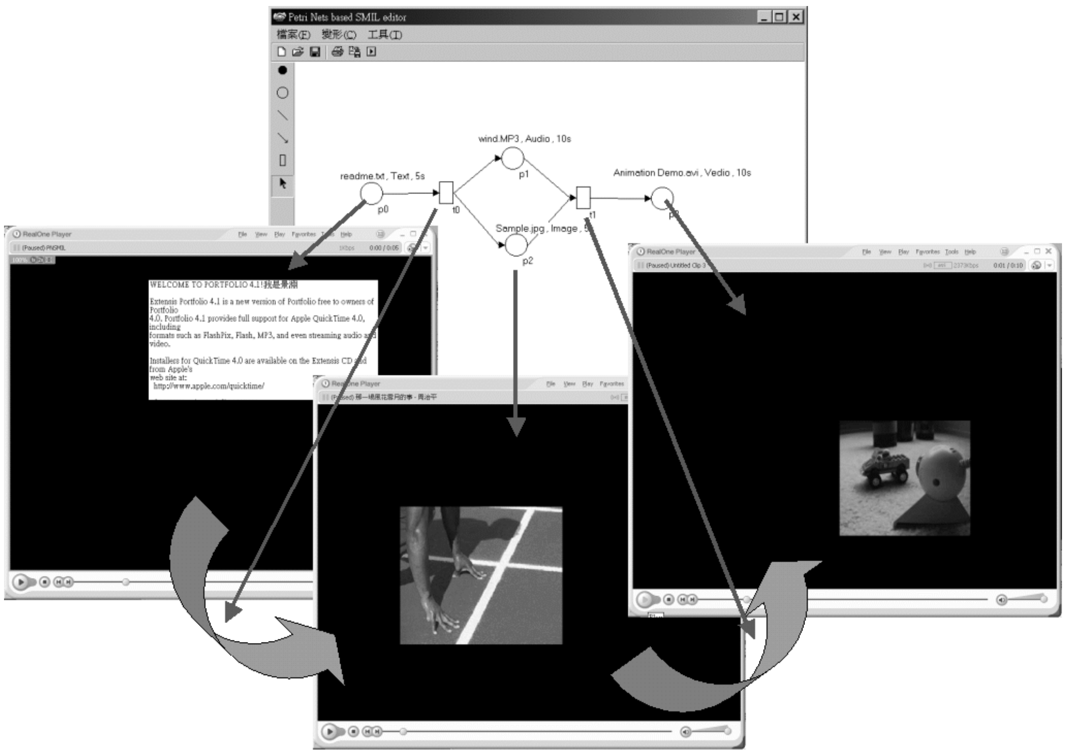

Figure 15. The result of SMIL

4.3. Streaming media server

In our study, there are many streaming server system can be used to support multimedia streaming. Due to RealNetworks Helix Universal Server can support SMIL file, so we used it to be our media server. Helix Universal Server also can support the following type of multimedia:

a. RealNetworks: RealAudio (.rm), RealVideo (.rm, .rmvb), RealPix (.rp), RealText (.rt)
b. Macromedia: Flash (.swf)
c. Microsoft: Windows Media (.asf, .wma, .wmv)
d. Apple: QuickTime (.mov)
e. Standards-Based:MPEG-1, MPEG-2, MPEG-4, MP3
f. Image Formats: GIF (.gif), JPEG (.jpg, .jpeg), PNG (.png)
g. Other:AU (.au), AIFF (.aif, .ief), WAV (.wav), SMIL (.smi, .smil)…

and the following protocols:

a. Real Time Streaming Protocol (RTSP)
b. Progressive Networks Audio (PNA)
c. Microsoft Media Services (MMS)
d. HyperText Transfer Protocol (HTTP)

Helix Universal Server provide web interface to operate. We only need to set the IP and port of RTSP, and put the related files into defined directory. User can input the url of SMIL file on the SMIL player, then the file will be played in streaming way from the server.

5. Conclusion

In this study, we have proposed SMIL editor and rendering tool for Multimedia Synchronization and Integration can provide a fast and intuitive method to help user generate multimedia content. The mentioned problems between Petri nets and SMIL editor have been resolved in the proposed system. There are some topics still in developing, likes the analysis and simulation modules on Petri nets to enhance the SMIL rendering Environment.

Reference

[1.] C. A. Petri, “Kommunikation mit Automaten,” PhD thesis, 1962

[2.] P. Merlin and D. Farber. "Recoverability of communication protocols – implications of a theoretical study,” IEEE Transactions on Communications, 1976.

[3.] K. Jensen. “Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Basic Concepts.” EATCS Monographs on Theoretical Computer Science, pp 1-234. Berlin: Springer-Verlag, 1992.

[4.] T. D. C. Little and A. Ghafoor, “Synchronization and Storage Models for Multimedia Objects,” IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, 1990, pp. 413-427.

[5.] Nabil R. Adam et al. “A Dynamic Manifestation Approach for Providing Universal Access to Digital Library Objects” IEEE Transactions On Knowledge And Data Engineering, Vol. 13, No. 4, pp.705-716 July/Aug. 2001

[6.] W3C Recommendation, “Synchronized Multimedia Integration Language (SMIL 2.0),” Aug. 2001, http://www.w3.org/TR/smil20.

[7.] Tadao Murata, "Petri nets: Properties, analysis and applications", Proceedings of the IEEE, Vol. 77, NO. 4, pp.541-580, Apr. 1989

[8.] L. Rutledge,"SMIL 2.0: XML for Web Multimedia" IEEE Internet Computing, pp.78-84, Sep.-Oct. 2001

[9.] RealNetworks Production Guide, http://service.real.com/help/library/guides/realone/ProductionGuide/HTML/realpgd.htm