DYNAMIC WEB MARKUP LANGUAGE (DWML) FOR GENERATING ANIMATED WEB PAGES WITH CHARACTER AGENT AND TIME-CONTROL FUNCTION

Du Peng   Mitsuru Ishizuka

Department of Electronic Engineering
School of Engineering, University of Tokyo

ABSTRACT

Rapid growth of competition on the electronic market place calls for innovative communication modes with web users. Considering the emergence of dynamic media objects containing temporal infromation such as animated character agents, web page scripts nowadays cannot satisfy users' increasing requirements well. We will develop a Dynamic Web Markup Language (DWML) as a software platform for authoring animated web pages with character agents. We assume that both primary media objects (text, image, etc.) and animated characters can be dynamic objects carrying time-based information, and that a character agent is a special dynamic medium as part of a multimedia interface. DWML, having a distributed structure and timeline flow, is used to generate both dynamic and static media objects, as well as characters' verbal and non-verbal behaviors.

1. INTRODUCTION AND MOTIVATION

Computers are now getting released from the times of WIMP (windows, icons, menus, point-and-click devices) graphical user interface. Post-WIMP will be increasingly based on the way we naturally interact with our environment and with other humans [1]. During the recent half century, skilled artists have achieved surprisingly life-like behaviors in animated characters. In the last few years, character agents, based on real video, cartoon-style drawings, or model-based 3D graphics, have become increasingly popular in user interfaces. For applications, they are a promising option since they make the application interface more lively and more appealing. When applied to the WWW, they bring more time-based information to web pages since they have verbal and non-verbal behaviors containing complex temporal information. However, present web scripts still have little functionality for describing temporal information of media objects. Existing hypertext models are also insufficient to describe the complexity of web contents which contain both dynamic and static information [5].

Based on these facts, we want to develop a Dynamic Web Markup Language (DWML) as a software platform for authoring dynamic web pages carrying time-based information. DWML describes behaviors of both primary media objects and character agents, generates messages of both audio and video channels, and deals with spatial and temporal relations for web pages. This work is also an extension to MPML (Multimodal Presentation Markup Language) [9, 10], which is designed mainly for simplicity in controlling character agents.

2. FEATURES OF DWML

2.1. Character Agent

In DWML, character agents play a main role. Among media objects, character agents carry the most complex information as they deliver messages over both verbal and non-verbal channels. They can be given the form of humans, animals, animate objects, or fantasy creatures [4]. Compared with other media, they can be employed to attract users' focus of attention, guide the users regarding web contents, teach new ways of presentation and convey additional conversational and emotional signals through their behaviors. We distinguish three types of character agents' behaviors:

- Speech: The verbal channel message through which character agents convey information mainly. While character agents are speaking, text balloons are shown to visualize the contents of utterance. The pitch, speed, volume of voice can be changed, and some words or sentences can be emphasized.

- Gesture: Occasional gestures make character agents more natural and more attractive [2, 3]. To keep character agents' simple to use, we use predefined gestures for every character. Users can define and create higher-level or intended behaviors by easily combining basic gestures and locomotions sequentially.

- Locomotion: In some cases, a character agent has to move its body to a suitable place in order to gesture at something or not to occlude other things.

2.2. Multimodal Presentation Mode

Figure 1 suggests the alternation of web content expression mode from HTML to DWML through giving dynamic information over time axis in several channels.

In HTML, texts and figures are mostly shown at the time when a web page is accessed and typically do not change over time[1]. Video and audio files are executed in fixed periods. Whereas in DWML, media objects have behaviors. Their behaviors occupy time spans and are lined in arrays and executed in predefined order. Through this alternation, web pages are no longer plain information boards, but web-based multimodal presentations whose contents are conveyed by character agents' behaviors supplemented with audio and visual aids.

We endow static media (text, image, etc.) behaviors over time such as movement, flash and enlargement. Their

---

[1]Occasionally animations created by Java, Javascript and Informix are used.
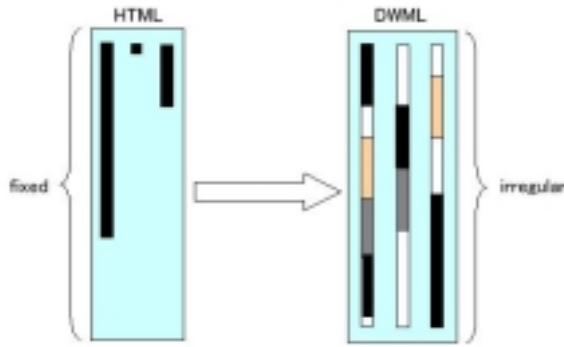
Figure 1: Alternation of Expression Mode

emergence and disappearance are defined as behaviors of "show" and "hide". Moreover, as we define the idle time behavior as media objects' behavior of "idle", they possess continuous behaviors so that web pages have multiple modalities to convey their contents.

2.3. Distributed Structure

We endow web pages authored by DWML the function of interaction with users through typing commands, clicking mouse-sensitive devices and even speech recognition. Then web pages are not just played linearly, but have branching structures which allow users to choose among different possibilities of navigation.

Inspired by the hypermedia model of PPP persona [2, 3], we represent information of web pages by a collection of interactive units and define anchors and links specifying how to get from one unit to another. Due to the fact that DWML documents contain time-based information, the links must unambiguously refer to an associated time point in the destination documents. In DWML, therefore, the same destination document can be different destination anchors of links. Links can also be pointed at current documents themselves to destruct the timeline flow. The following types of links are used in DWML:

- Hard link: Browser abandons current document, refreshes window, then loads the destination document.

- Soft link: Browser stops current timeline flow, goes on to execute scripts of the destination document. By using soft links instead of branched scripts, a document containing branches can be separated into multiple sets of scripts, thus showing those branches.

2.4. Time-Control Function

Every unit is a set of scripts generating a sequence of behaviors of character agents and other media. These behaviors are lined in arrays and executed either sequentially or in parallel. Timeline structures enable us to represent the temporal behaviors in an intuitive manner; however, they provide no means of describing the control flow of interactions. That is the reason why we keep every unit as one timeline for describing the temporal behaviors and use an additional distributed structure serving to describe the navigation between web pages.

We have developed a time-control interface for DWML documents. By using the control interface similar to that of an audiocassette or compact disc player, users can stop, start, fast-forward, rewind (and sometimes search through) the presentations. It brings great convenience for users to debug their works.

3. SPECIFICATION

DWML documents respects the XML and SMIL specifications [14]. The advantage of a markup language is that it is human-legible, reasonably clear, and independent of platforms and systems. We chose the style of well-formed document for the reason that this style reduces users' load to comply with DTD (Document Type Definition). DWML meets all the well-formedness constraints given in the XML specification. Generally, we divided DWML into ⟨HEAD⟩, which provides properties of documents and predefined information, and ⟨BODY⟩, which comprises the content of web pages to be presented. This structure has already been familiar to users.

3.1. Document Information

Document information is collected in the document head. Following HTML, we also use ⟨TITLE⟩ to record document titles, ⟨META/⟩[2] to define properties of documents (e.g., author, expiration data, key words, etc.) and assign values to them. Besides, we add ⟨LAYOUT/⟩ to describe documents' layouts, ⟨REGION/⟩ to predefine the regions of posterior objects. Regions are defined by a set of values of "left", "top", "width", and "height".

3.2. Media Object

DWML generates these kinds of media objects: text, image, table, form, audio and video files, as well as character agent, defined by ⟨TXT⟩ ⟨IMG/⟩ ⟨TABLE⟩ ⟨FORM⟩ ⟨AUDIO/⟩ ⟨VIDEO/⟩ and ⟨AGENT/⟩ respectively. Properties of objects are described in the attributes of tags. For instance:

```
<TXT id="title" size="8" region="100,100,300,200">
        Welcome to our site!
</TXT>
<AUDIO id="welcome" src="welcome.wav"/>
<AGENT id="peedy" char="peedy"
                region="100,100,x,y"/>
```

The attribute "region" can be given the value of "id" predefined by ⟨REGION/⟩ in the document head. "x,y" means original sizes of media objects. In terms of table and form, we try to keep the concordance with HTML. Furthermore, multiple objects can be put into ⟨GROUP⟩ to compose a group object so that they are easy to be handled together.

3.3. Media Behavior

Media objects delimited by ⟨SHOW⟩ will appear and will be cleaned off by ⟨HIDE/⟩ at the time points in some way (e.g., zoom, slide-out, etc.). For instance:

---

[2] A tag followed by a slash is an empty-element tag, otherwise it is a pair of start-tag and end-tag.

```
<SHOW way="zoom">
   <IMG id="photo" region="50,50,300,300"
                  src="photo.jpg"/>
</SHOW>
<HIDE obj="photo" way="slide-out-left" />
```

Behaviors of media objects are described respectively by ⟨MOVE/⟩ ⟨PLAY/⟩ and ⟨SPEAK⟩. For instance:

```
<MOVE obj="photo" des="0,100,x,y"/>
<PLAY id="start" obj="peedy" act="greet"/>
<SPEAK obj="peedy" volume="50">Welcome!</SPEAK>
```

The attribute "obj" should be given the "id" value of predefined media object.

3.4. Synchronization

Similar to SMIL, DWML use ⟨PAR⟩ and ⟨SEQ⟩ to describe synchronization of behaviors. The children of ⟨PAR⟩ can overlap in time. The textual order of appearance of children has no significance for the timing of their presentation. Whereas the children of ⟨SEQ⟩ will be executed sequentially according to the listed order. By default, the document body is delimited by ⟨SEQ⟩.

3.5. Interaction

We define three tags to detect users' input and give corresponding feedback. ⟨COMMAND/⟩ processes commands input from command window (this window is motivated and cleaned off by hot keys); ⟨CLICK/⟩ detects users' mouse-click; ⟨VOICE/⟩ is used to discriminate users' voice input. The attribute "href" indicates the destination file. It contains not only file name, but also the id value of the time point from which presentation should start. For instance:

```
<VOICE msg="go next" type="soft"
            href="next.dwml(start)"/>
```

To respect the timeline structure of document body, we consider these definitions as document information and put them into document head.

3.6. Others

Users can insert ⟨PAUSE/⟩ in scripts to stop the execution of the presentation unit. This suspension lasts indefinitely until users push the [play] button in the time-control interface. Users can arbitrarily change the timeline flow through inserting ⟨LINK/⟩ in timeline. When the processor meets it, the current document will be stopped and the destination document will begin to be executed. As far as comment description, we use the same style as HTML.

Additionally, in DWML, every tag can hold an "id" attribute to specify individual media objects (for media object) or time points (for behavior), an "abstract" attribute to give a brief description of the content contained in the element, a "begin" attribute to specify the time for the explicit begin of an element.

4. AN EXAMPLE

Here we give an example to explain the timeline flow of DWML documents. Script 1 shows a unit of DWML documents describing the well-known story of the famous mathematician J.C.F. Gauss. Scripts are executed sequentially and output is played as a presentation. Figure 2 shows the scenes at three time points of the timeline. On the top of every window is the time-control interface which contains a slide bar with five buttons. Names of scenes are recorded in scripts. Notice the change of arithmetic formulas and the slide bar.

```
<SPEAK obj ="peedy">One day, Gauss's teacher let
         the students solve this problem.</SPEAK>
<PLAY obj="peedy" act="GestureLeft"/>
<SHOW><TXT id="line1" region="350,140,100,10">
               1+2+3+4+.....+99+100=?</TXT></SHOW>
<PLAY obj="peedy" act="Confused"/>
<PAR><PLAY obj="peedy" act="Uncertain"/>
    <SEQ><SPEAK obj="peedy">My god, how to do?
             Like this? 1 plus 2 equal...</SPEAK>
         <SHOW>
          <TXT id="xline1" region="340,170,100,10">
               =3+3+4+5.....+99+100</TXT></SHOW>
         <SPEAK obj="peedy">
             And this? 3 plus 3 equal...</SPEAK>
         <SHOW>
          <TXT id="xline2" region="340,200,100,10">
               =6+4+5+6.....+99+100</TXT></SHOW>
    </SEQ>
</PAR>
<PAR id="scene1" obj="peedy">   <PLAY act="Dislike"/>
                  <SPEAK>Oh, NO,no,no!</SPEAK>
</PAR>
<HIDE obj="xline1;xline2" way="slide-out-right"/>
<PLAY obj="peedy" act="Suggest"/>
<SPEAK obj="peedy">
     Ok, Let's see what Gauss is doing.</SPEAK>
<PAR obj="peedy"> <PLAY act="Think"/>
                  <SPEAK>The equation can be
                     transformed into this.</SPEAK>
</PAR>
<SHOW id="scene2">
    <TXT id="line2" region="340,170,100,10"">
            =1+100+2+99+3+98....+50+51</TXT></SHOW>
<PLAY obj="peedy" act="Write"/>
<SPEAK obj="peedy">And, it equals...</SPEAK>
<SHOW><TXT id="line3" region="340,200,100,10">
               =101+101+101+....+101</TXT></SHOW>
<SPEAK obj="peedy">And, it equals...</SPEAK>
<SHOW><TXT id="line4" region="340,230,100,10">
                         =101*50</TXT></SHOW>
<PLAY obj="peedy" act="Pleased"/>
<SHOW><TXT id="line5" region="340,260,100,10">
                         =5050</TXT></SHOW>
<SPEAK obj="peedy">Yeah, the result is 5050!</SPEAK>
<PAR id="scene3" obj="peedy">
    <PLAY act="Congratulate"/>
    <SPEAK>Great!
    Gauss is really a mathematical Genius!</SPEAK>
</PAR>
```

Script 1: A Unit of DWML

From this example, one can see that the merits of DWML include:

1. Users can understand the content easily through reading DWML scripts only.

2. The utilization of character agents makes this presentation style more attractive and more lively.
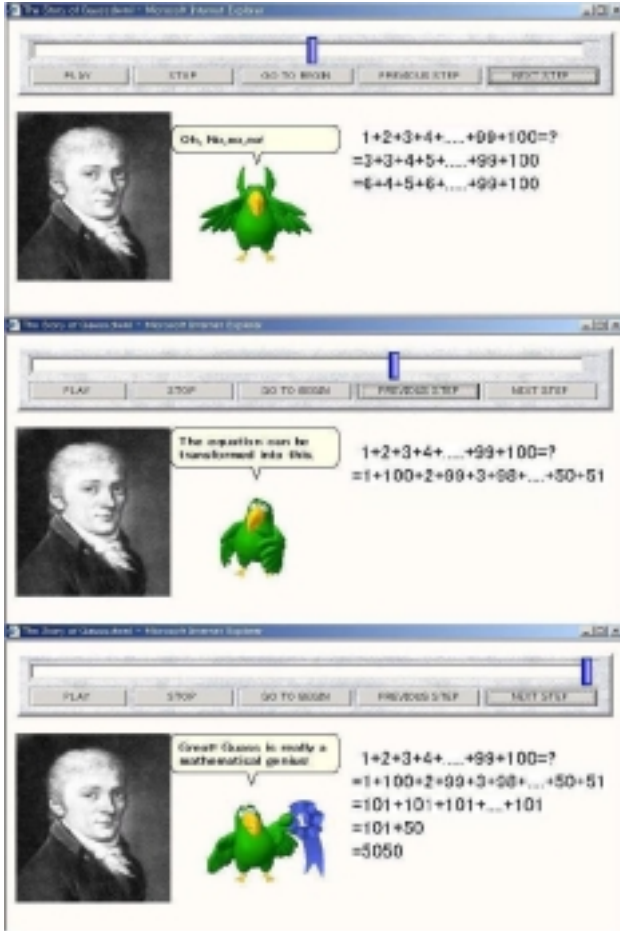
Figure 2: Three Scenes at Different Time Points

3. Text and image can be shown and cleaned off timely, which helps users understand the contents better.

4. DWML is useful for modeling dynamic multimodal educational contents.

5. IMPLEMENTATION

Owing to the fact that current web browsers (I.E., Netscape etc.) are ubiquitous and popular, we implement this language by existing tools. We have produced a DWML processor using the technique of XML and XSL. The eXtensible Markup Language, abbreviated XML, is a subset of SGML. The eXtensible Stylesheet Language, abbreviated XSL, is a language for transforming XML documents and specifying formatting semantics. At present, we are using the Microsoft Agent package [6, 13] to realize speech recognition and generate character agents and their behaviors for DWML documents.

6. CONCLUSION

Although currently dynamic web pages can be authored by some visual applications (e.g., Turquoise [7], Dynamic Web Page Builder [12], Astound [11], etc.), and some presentation agents have been developed for multimodal presentations [2, 4, 8, 9], there are still few systems able to control primary media objects and character agents together with time-control function. DWML is suitable for creating dynamic web pages in which an animated character plays a main role and other multimedia objects such as audio, video and text are synchronized with the behavior of the characters.

DWML is a markup language conforming to the XML specification. It transforms traditional representation mode of web information into timelined presentation style. At the same time it supports time-control function and media synchronization. So far, DWML only contains basic tags and functions to complete simple tasks. Future work will be focused on enriching functions of DWML for more complex web information. We are also considering to develop an interactive model instead of anchors and links for describing the transitions between presentation units.

7. REFERENCES

[1] Andy van Dam. Beyond WIMP. IEEE Computer Graphics and Applications 20(1), Jan/Feb. 2000.

[2] Elisabeth André, Thomas Rist, Jochen Müller. WebPersona: a lifelike presentation agent for the World-Wide Web. Knowledge-Based Systems II, 1998.

[3] Elisabeth André, Thomas Rist, Jochen Müller. Integrating Reactive and Scripted Behaviors in a Life-like Presentation Agent. Autonomous Agent 1998.

[4] Ken Perlin and Athomas Goldberg. Improv: A System for Scritping Interactive Actors in Virtual Worlds. Computer Graphics/Vol. 24, No. 3, 1996.

[5] Lynda Hardman, Dick C.A. Bulterman, Guido van Rossum. The Amsterdam hypermedia model. Communications of the ACM 37(2), 1994.

[6] Microsoft. Developing for Microsoft Agent. Microsoft Press, 1998.

[7] Robert C. Miller, Brad A. Myers. Creating Dynamic World Wide Web Pages By Demonstration. Carnegie Mellon University School of Computer Science Tech Report CMU-CS-97-131 (and CMU-HCII-97-101), May 1997.

[8] Tsukasa Noma, Norman I. Badler. A Virtual Human Presenter. Workshop on Animated Interface Agents, IJCAI-97.

[9] Takayuki Tsutsui, Santi Saeyor, Mitsuru Ishizuka. MPML: A Multimodal Presentation Markup Language with Character Agent Control Functions. Proc. Web-Net 2000 World Conference on the WWW and Internet, San Antonio, Texas, USA, 2000.

[10] Yuan Zong, Hiroshi Dohi, Helmut Prendinger, Mitsuru Ishizuka. Emotion Expression Function in Multimodal Presentation. Advances in Multimodal Interfaces—ICMI2000, Beijing, China.

[11] http://www.astound.com/

[12] http://www.bigplanet.com/dwp/

[13] http://www.microsoft.com/msagent/

[14] http://www.w3c.org/