

T2V: New Technology of Converting Text to CG Animation

Masaki Hayashi[†] (member), Seiki Inoue^{††} (member), Mamoru Douke^{††} (member),
 Narichika Hamaguchi^{††} (member), Hiroyuki Kaneko^{††} (member),
 Steven Bachelard^{††}, Masayuki Nakajima[†] (member)

Abstract We have developed a technology that enables the creation of TV-program-like CG animation generated automatically from text-based script. We call this technology "T2V (Text-To-Vision)" and have developed a PC application that we call the "T2V Player." The application allows users to create animations instantly by simply typing in text in a customary fashion as when writing on a word processor. The T2V Player employs a unique framework with a capability to accommodate any text processing module to realize any type of text-to-animation conversion. The T2V Player is not only an animation tool for writers creating animations from their texts, but also an application of various text-to-vision services such as automatic animation generation from Web sites and other text sources. This paper describes the concept of T2V and the technical specifications of the T2V Player and its benefits.

Keywords: Text visualization, Computer animation, Authoring tool, Text-to-animation, TVML.

1. Introduction

It is common today to see digital content made by non-professionals. This is particularly true on the Internet where numerous armature movies can be found in blogs and Social Network Systems such as Facebook, Twitter, etc. and in movie sharing services such as YouTube. However, TV-program-like film clips with high production value are found less frequently. There is still a considerable barrier to the creation of this type of content for non-professionals.

Based on this background, we have been developing a technology of producing TV-program-like CG animations (CGA) automatically from text-based script. We call this technology "T2V: Text-To-Vision" (Fig. 1). One of the significant characteristics of TV content is its large variety. The genre of TV programs includes news, drama, reality shows, sports, variety, music, education etc. The TV program production industry has accumulated production methodologies for many years. But these methodologies have not been made readily

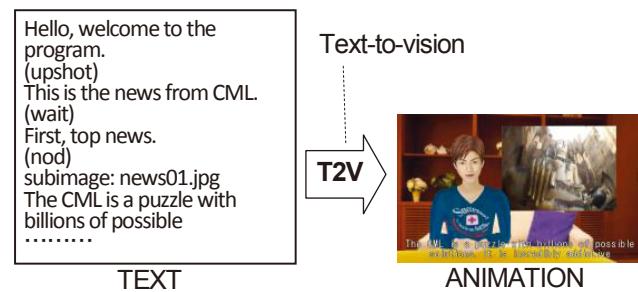


Fig.1 Concept of T2V.

available to non-industry individuals. Our aims are to open the rich TV program production methodology in the world of UGC (User-Generated-Content).

To facilitate this, we have developed a PC application that we call the "T2V Player". This application allows users to create animations instantly by simply typing in text in a customary fashion as when writing on a word processor. This is expected to lower the barrier to the creation of CGA dramatically.

The T2V technology is built with the TVML (TV program Making Language) which was proposed by NHK (Japan Broadcasting Corporation)¹⁾. TVML is designed as a special computer language that describes the content of TV program. A TVML Engine creates the complete CG-animated sequence by integrating real-time CG character

Received April 5, 2013; Revised July 29, 2013; Accepted September 27, 2013

[†] Uppsala University
 (Visby, Sweden)

^{††} NHK Science and Technology Research Laboratories
 (Tokyo, Japan)

animation, synthesized voices, superimposing, CG camera work, font display, movie file play back, still image displays, sound file play back etc²⁾⁻⁵⁾.

In the T2V Player, users begin by writing their content in a common but descriptive language. The text-processing module then converts this to TVML, which is then converted into CGA by the TVML Engine.

In order to realize the text processing, the T2V Player provides a special framework to accommodate text-processing plug-in modules. We call this module the "Flexible Interpretation Loader" or "FIL". A number of FILs can be cascaded. These FILs process the user input text in various ways ultimately converting it to TVML script.

This framework is both simple and flexible allowing developers to plug in their text processing modules to the T2V Player easily. This opens up for various ways of using the T2V Player not only for the easy-to-use CGA creation tool but also for automatic conversion of Web sites to animation as explained later in this paper.

This paper is organized as follows. Chapter 2 describes the related works. Chapter 3 is a brief introduction of our previous work on TVML. Chapter 4 describes the overall technology of T2V. Section 4.1 describes the new architecture of cascaded text processors. Section 4.2 describes the text-processing framework. Section 4.3 describes the newly developed T2V Player. Section 4.4 introduces a field test conducted with the T2V Player as distributed freeware to individuals and discusses the results. Section 4.5 describes the automatic movie generation from Web sites, which became possible by the T2V Player architecture. Finally, Chapter 5 contains the conclusion for this paper.

2. Related work

In this chapter, we will survey related work of text visualization, which is the basic concept of the T2V.

2.1 Text visualization

Technology to generate animation from text is called "Text-to-animation" and has been studied for many years. In particular, the research on analyzing text written in natural language to visualize the written text by CG has been mainly carried out in the area of Artificial Intelligence (AI.)⁶⁾ In this text-to-animation research area, many attempts have been made. Three such examples are "WordsEye"⁷⁾ which generates a static CG scene from given text written in natural language, another is "Confucius"⁸⁾ which uses 3D modeling/animation techniques with natural language-

understanding technology to achieve higher level animation generation, the last one is the automatically converting annotated fiction text into 3D animations⁹⁾. These are only relatively successful as there are many difficulties to overcome; Such as: managing the ambiguity of natural language, how to give appropriate attributes to CG objects, how to design autonomous virtual agents and so on. In fact, the trials of rebuilding appropriate animation analytically from any given arbitrary instruction written in natural language, often tend to result in incomplete visual representation confined to a very narrow application domain.

On the contrary of this straightforward AI method, there is another approach for the problem. This approach focuses on TV-program-like animation providing users with an "authoring tool" instead of the user providing arbitrary text to make animations. There are a variety of services currently available especially on the Web. These are, for example, "Xtranormal", "GoAnimate", "SitePal" etc. Among these services, the "Xtranormal" is particularly good. Users can create high-quality CG animations and publish results by pressing a few buttons followed by typing in the speech text and specifying character actions and camera work etc. by using a special GUI (Graphical User Interface) provided on the Web browser. This service is popular and many movies have been uploaded on the site. However, the service has been completed on the Web and is limited by the user creating animations through the GUI on the Web page. Therefore, it is not a simple structure to create animation from text which we intend to realize with the T2V. Other services such as "GoAnimate" and "SitePal" are similar in this regard.

2.2 Studies based on TVML

As mentioned in the introduction, T2V is based on TVML. TVML is unique as a language to describe and generate TV programs at a script level. NHK, which proposed TVML, has performed several studies based on TVML¹⁰⁾⁻¹²⁾. The studies are primarily for TV content production in the present TV industry. While the aim of the T2V is to encourage UGC through user communities on the Internet.

As an animation production tool using TVML, Digital Movie Director (DMD) has been studied for years¹³⁾. DMD is an authoring application with a dedicated GUI. A user can create CG animation on a PC with the DMD installed, and then publish it on the Web. The structure of the GUI in DMD is similar to that of "Xtranormal" mentioned in the previous section. DMD uses the TVML

internally, however, the TVML is hidden from the user. In addition, since the script is not a plain text like as is the case in the "Xtranormal", it cannot be applied to non-production purposes such as automatic movie generation, which is explained later in this paper.

Next, we will introduce TVML, which is the base technology of T2V, followed by the description of the technical details of the T2V.

3. TVML

As mentioned above, T2V technology is built on the TVML. In this chapter, we will explain the TVML to help the comprehension of the T2V technology described in the next chapter.

3.1 The language

TVML is a special computer language designed to produce TV-program-like CG animation by a computer. Unlike programming languages, the nature of the TVML is rather a "scripting language" which is similar to a script structure used in actual TV production. **Fig. 2** shows an example script of the TVML. Each line represents an event to be performed. The software called the TVML Engine produces CG animation in real-time by reading the TVML from the beginning to end, line by line.

Fig. 3 shows the format of an "event" as one line. The event consists of three parts: "Event type", "Command", and "Parameters". The event types of TVML correspond

```
camera: twoshot (target1=A , target2=B )
character: look ( name=A , target=B )
character: talk ( name=A, text="How's going?" )
character: talk ( name=B, text="Awful" )
sound: play( filename="boo.mp3" )
.....
```

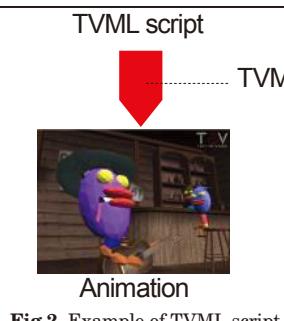


Fig.2 Example of TVML script.

character: talk (name = Bob, text = "Hello")

↓ ↓ ↓

Event type Command Parameters

Fig.3 Format of "Event".

to the basic elements used in the real TV production shown in **Table 1**. As seen in this table, the TVML covers almost all the functions needed to produce information types of TV programs. Each event provides number of commands, and each command has a number of parameters that specify the behavior of the command. In the example script shown in Fig. 2, a scene is described where "The camera creates a two-shot of character A and character B, then character A looks at character B, then character A says: 'How's it going?', then character B says: 'Awful', then the sound file named 'boo.mp3' is played back ."

3.2 TVML Engine

Fig. 4 shows the configuration of the TVML Engine. It consists of three parts: The "Sequencer", which reads the given text, line by line, and the "Parser", which parses the one TVML line and generates corresponding instructions. The "Animation engine" then generates real-time audio and video according to the instructions coming from the parser. The animation engine executes this immediately upon receiving the trigger of action

Table 1 TV production versus TVML.

Category	TV production	TVML
Studio shot	Studio set	set
	Prop	prop
	Actor	character
	Lighting	light
	Camera	camera
Footage	Video playback	movie
Telop	Text, image display	title
Superimpose	Text, image display	super
Art work	Making figure	drawing
Sound	Sound effect	sound
	Audio mixing	sound

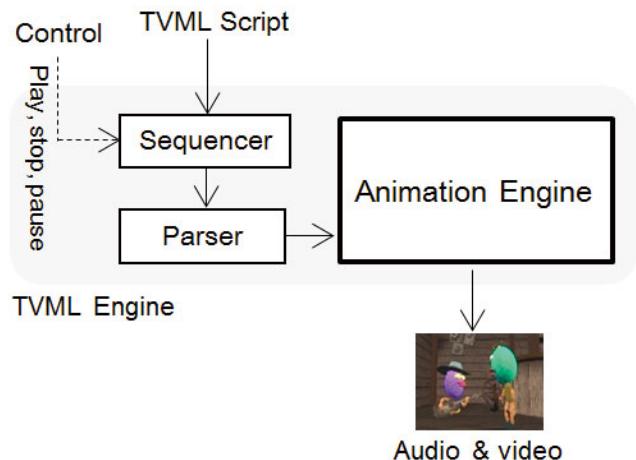


Fig.4 Configuration of TVML Engine.

from the parser. Therefore the video and audio output can be obtained immediately after receiving one or more TVML lines given from the external application. This enables the engine to work as a perfect interpreter of given TVML script.

4. T2V

4.1 Mechanism of T2V

The basic concept of T2V is "to visualize any arbitrary text". However, our approach is not to make a complete text-to-animation AI machine with the capability of interpreting any arbitrary input text given by users in natural language as explained in section 2-1. Instead of developing an "Ultimate Machine", we provide a framework enabling many types of text-to-animation engines made by developers. As a method to achieve this, we provide a plug-in socket for modules to process the given text string as an input to generate a TVML script which is then fed into the TVML Engine and played back in order to obtain the final CG animation.

Fig. 5 shows the basic configuration of T2V. Input text is fed into the text processing module called the FIL (Flexible Interpretation Loader). The processed result is output in an arbitrary text format. As shown in the figure, the text conversions are performed one after another as the FILs cascade, and then ultimately the last FIL outputs TVML script that is then visualized by the TVML Engine resulting in the final animation. Each FIL has its unique name called the "FIL name". The FIL identification format is as follows: The first FIL informs the system which FIL will follow and become the second FIL. The second FIL name is written in the first line (as indicated below) of the text output from the first FIL.

```
#!/textfilter="FIL name"
```

The system checks the "FIL name" in the line above,

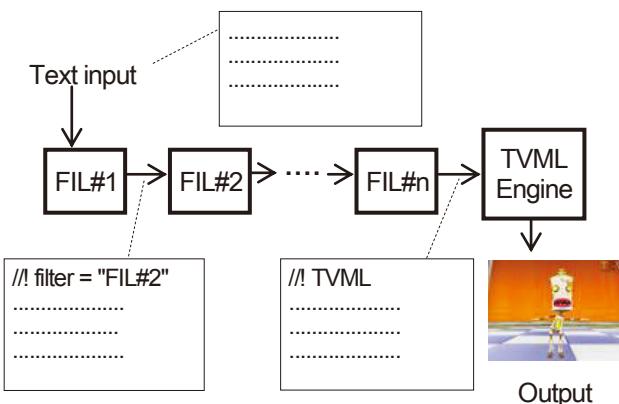


Fig.5 Configuration of T2V Player.

and retrieves corresponding FIL and input text to the FIL which then performs the process. Finally when a FIL outputs the text whose first line is as follows, then the text is passed to the TVML Engine.

```
#!/ TVML
```

Finally, the TVML Engine visualizes the processed text resulting in the CG animation. Any number of cascade FILs can be connected.

4.2 Mechanism of FIL

The details of the FIL can be described as follows: A FIL is represented by a text file in XML format named "FIL description". The FIL description is an XML file with the extension ".fil." **Fig. 6** is an example of a FIL description. As in the figure, `<name>` tag gives FIL name; `<engine>` tag specifies an entity of the text-processing module. In this example, the `<stage>` tag is specially defined for the T2V Player as described in the next section. The content of `<stage>` tag here is TVML initialization script which is referred by the text processing module.

The text processing module specified by the `<engine>` tag is a dll (Direct Link Library) in the Windows OS implementation. In this dll, only the following simple C++ function is defined.

```
char* filproc (char* intext) ;
```

The function "filproc" processes the given string data "intext" then returns the output string data. Specifications of input and output text processed by the "filproc" are left to the developer, except the first line of the text as explained in the previous section. The T2V Player provides only the framework allowing the

```

<?xml version="1.0" encoding="UTF-8"?>
<filDescription>
    <name>Test</name>
    <caption>Test Show Format</caption>
    <engine>./BeeFIL.dll</engine>
    <icon>test.jpg</icon>
    <help>test.html</help>
    <staging>
        <numofchar>1</numofchar>
        <model_for_char character='A'>
            character: openmodel( model=BUCHU,
                filename="bucchu\bucchu.bm" )
            character: casting( name=A, model=BUCHU )
            character: setvoice( name=A, voicetype="man" )
            character: position(name=A, z=1.00)
        </model_for_char>
        <setting>
            skipscript( switch=on )
            set: openmodel( name=HALL,
                filename="hall\hall.obj" )
            set: change( name=HALL )
            light: flat(name=FLAT1, vx=1.00)
            light: flat(name=FLAT2, vx=-1.00)
            camera: position(name=Acam, z=3.0, vangle=45)
            skipscript( switch=off )
        </setting>
    </staging>
</filDescription>

```

Fig.6 Example of FIL description.

developer a high degree of freedom.

A FIL is deployed in a folder on a file system. The single folder is treated as the entire module of the FIL. The folder contains a FIL description file and a text processing module (a dll in Windows OS) and other data files. The other data files include: a configuration file referred by the text-processing module, an instruction manual of the FIL and all the material data such as CG character modeling data, image data, sound data etc. referred by the TVML Engine when it is played back.

4.3 T2V Player

T2V Player is developed as one of the applications using the FIL framework as described in the previous section. The T2V Player is developed to allow users to create animations instantly by typing in text as done when writing words on an ordinary word processor. The specification of the text is designed intuitively so that non professionals and without previous knowledge can easily create animations with it.

The application runs on a Windows PC. **Fig. 7** shows the screen-shot of the application. The left side of the screen is the animation playback area and the right side is the text editing area. Users can type in script in the text editing area, and press the play button in the lower left to generate a CG animation with the CG characters appearing and acting according to the script written.

Table 2 shows an excerpt from the specification of the command and the tag used in the T2V script. Approximate sixty commands and eight tags are provided in the current version. **Fig. 8** shows an example of the script written using the commands and the tags. **Fig. 9** shows several snapshots from the resulting animation. The main part of the script is speech text. "Command" is represented by a command name in parentheses, which controls character's movement, camera, etc. "Tag" is also available to display image-files and playback sound-files, movie-files playback etc. As shown in the fig. 9, the scene (A) is the title and the sub-title display, the scene (B) is the character speech with automatically added lip-sync with superimposed subtitles, the scene (C) is the sub-image display with automatic camera panning, and the scene (D) is a full screen display of the main-image.

There is no limitation in the number of CG characters. Each character can be distinguished by giving "A:", "B:", "C:" and so forth at the head of the speech text or the command. All the commands and tags are described in the form of template in XML format in a FIL description. This allows developers to easily define new



Fig.7 T2V Player.

Table 2 Available commands and tags (excerpt from approx. 60 commands, 8 tags).

Commands	
(happy)	The character changes its facial expression 'happy'
(bow)	The character bows.
(nod)	The character nods once.
(no)	The character shakes its head once.
(look at fellow)	The character looks at the other character.
(look at camera)	The character looks at a camera which is shooting the character.
(shake)	The character shakes.
(stand)	The character stands up.
(sit)	The character sits down.
(walk n step)	The character walks <i>n</i> steps ahead. Ex. (walk 3 step)
(turn <i>n</i> degree)	The character turns <i>n</i> degrees. Ex. (turn 50 degree)
(upshot of <i>n</i>)	Display a close-up shot of character <i>n</i> . Ex. (upshot of A)
(twoshot)	Display a two-shot of the two characters.
tags	
mainimage:	An image designated by a file name written after this tag will be displayed to fill the display area.
subimage:	An image designated by a file name written after this tag will be displayed on a virtual monitor defined in a 3D-CG set.
BGM:	A sound file designated by a file name written after this tag will be played.

```
!! textfilter="HanaTalk"
title:FOOD EXPLORE ..... (A)
subtitle:Gukbap will cure your hangover<br>in Seoul
(happy, little)
In the afternoon of the second day when I stayed in Seoul,
my Korean friends took me to a Korean style cheap
restaurant.
(nod)
subimage:seoul1.jpg ..... (B)
This is it!
The restaurant's speciality is
'Gukbap', a famous Korean soup with rice.
mainimage:seoul1.jpg ..... (C)
A raw egg and dried seaweed are served in a small bowl on
the side.
You're supposed to put them in the soup, mix well and eat.
(back to studio)
This tastes very good!
```

Fig.8 Example of a T2V script.



Fig.9 Snapshots from the playback of the script of Fig. 8.

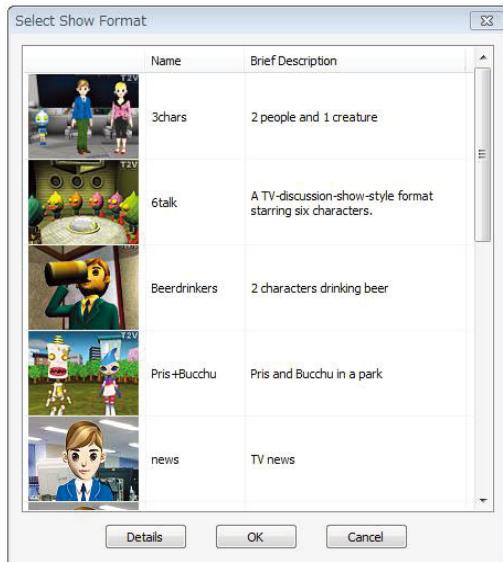


Fig.10 List of show format.

commands and tags by adding the corresponding template to the FIL description.

The T2V Player provides a list of show formats from which users can choose and apply to the text input. **Fig. 10** shows the GUI for choosing the show format from the list. The show format contains a number of instructions that direct which characters appear in which studio set, camera angle, camera work, BGM playback etc. The basic concept is to combine a "Text" and a "Show format" to generate an "Animation". Internally, one show format corresponds to one FIL. The T2V Player loads the prepared FILs at the start-up and creates a list of show formats.

4.4 Field test of T2V Player

The Japanese version of the T2V Player has been distributed as freeware¹⁴⁾ since March 2009 and the

recorded number of downloads on the download log was approximately 20000 one year after the first download started. The distribution and resulting downloads have continued since that time. The current version bundles more than twenty show formats with numerous example scripts.

The number of the active users is difficult to count. However, based on the number of downloads, we approximate that several thousand people have used the software. We investigated the number of movies made with the T2V Player on the movie sharing service "YouTube" by giving a keyword "T2V" in the search function. Approximately 350 movies were found on the site, (accessed July 2013). And several particularly interesting types of movies were uploaded. For example, an elderly woman 70-79 years of age has uploaded a series of movies. These movies are mostly inspired by events in her ordinary daily life, such as visits nearby, taking pictures, etc. and then used to create stories on T2V. The T2V Player does not have a cumbersome user interface requiring users only to type in text similar to writing E-mail. This dramatically lowers the barrier in terms of software operation. The T2V movies made by the elderly could be a good example to prove the inclusive advantage of the T2V methodology.

As we conducted several workshops using the T2V Player, one workshop will be introduced below. This workshop occurred in 2010. The participants were in their 20s and did not have any experience in creating CG animation. They received instructions for a period of one hour on how to use the T2V Player. They were then asked to create their own T2V scripts within one week. The results were positive and each of the participants successfully created their scripts. We can say that this proves its usability that enables non-professionals to create their own CG animations with the T2V Player. The evaluation of the quality of those twenty animations is inevitably subjective. However, one script out of the twenty was deemed particularly well made resulting in a high-level production. This suggests that the T2V player is not only an easy-to-use animation tool but also an effective content creation tool, which could produce high-level results by creative users. Finally, listed below are excerpts from participants' hearings.

* This is a quite easy way to make animation unlike learning computer programming.

* It might be useful for pros as well, if the quality of graphics and synthesized voice becomes better.

* It is amazing that I have made animation by myself,

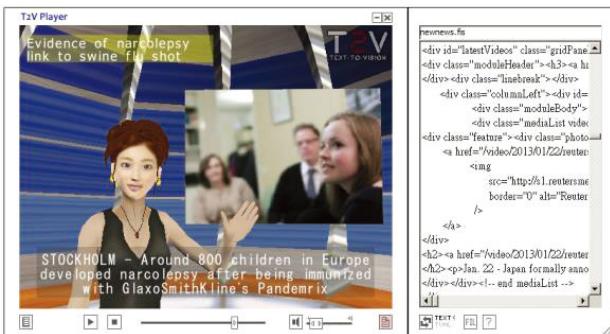


Fig.11 Automatic generation of news show.

just only after a few hours practice.

* I wanted to insert commands by icons or from pull down menus instead of writing the command words.

According to the comments, we can say that the T2V Player clears the basic functional requirement needed for making animations. However, there is still a room for improvement to enhance the usability and functionality.

4.5 Automatic movie generation

In this section we will introduce uses for the T2V Player beyond that of an animation production tool. The T2V Player can be used for automatic visualization of existing textual information. As an example, we will describe an automatic news-show generation from HTML used for News sites. This is also an example of how to use cascaded two or more FILs in the T2V Player.

Fig. 11 shows the scene of automatic generation of news-show produced from the News site "Reuters". The system generates the animation when the HTML from the site is copy and pasted into the T2V player. **Fig. 12** shows the configuration of the automatic generation of news-show. The two FILs are used to convert HTML to TVML. The operation of each FIL is as follows.

(1) FIL #1 for analyzing the HTML

The FIL analyses the tag structure of the HTML and divides it into each news article then extracts a news title, a main body of the article, JPEG images and the date and time. The FIL processes these elements and generates the T2V script as shown in Fig. 12 together with the first line specifying the next FIL name to which the script is going to pass.

(2) FIL #2 for producing news show

This FIL is one of the standard FILs prepared by the T2V Player. It has some special tags which for example superimpose the news title to be scrolled in with sound effect. The T2V Player plays back the script coming from the first FIL, and the automatic generated animation is obtained.

By applying the same framework as described above,

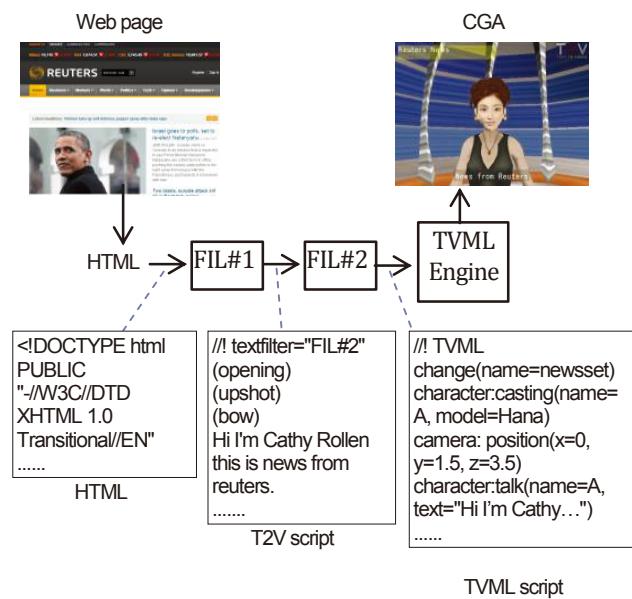


Fig.12 Configuration of the automatic generation of news show.

it is possible to visualize other types of Web sites and contents. We have prototyped a number of FILs converting Web sites such as weather forecasts, BBS (Bulletin Board Systems), E-commerce etc. Apparently, the converter would need to be re-configured if the original site's HTML configuration has changed.

This application is still in the trial phase however, it is possible to perform various enhancements in addition to the features described above. The enhancements could for example be; to provide specially compiled news shows based on the search result of RSS of the news source according to the user's keyword input. Additional features could provide native language news shows translated from other languages by inserting another FIL of machine translation to the chain of the FILs, and more.

5. Conclusion

In this paper, we described the T2V technology, which enables users to create CG animation automatically from text-based scripts. We have developed the PC application named "T2V Player" and have distributed it as freeware¹⁴⁾ encouraging UGC. The T2V Player is based on the TV program Making Language (TVML) which is designed as a computer language to describe TV-program-like animation. In order to facilitate a user-friendly and more inclusive format of text input, the T2V Player employs a special framework to accommodate text-processing modules on top of the TVML. The text processing module is called FIL. The FIL processes user input text and converts it to a TVML script for generating animation. Based on this framework, we

have also developed an automatic news show generation function in the T2V Player. This is done by converting HTML from the Internet news site to a TVML script by using two cascaded FILs.

We consider that the T2V Player is not only a useful tool for UGC but also a valuable tool for various text-visualization applications using text-processing techniques. We now aim to develop the system further to create a more interactive and intelligent television system based on the T2V technology.

[References]

- 1) M. Hayashi: "Automatic Production of TV Program from Script: A proposal of TVML", Proceedings of the 1996 ITE Annual Convention, S4-3, pp.589-592 (1996)
- 2) M. Hayashi: "TVML (TV program Making Language) -Automatic TV Program Generation from Text-based Script-", Proceedings of Imagina'99, pp.119-133 (1999)
- 3) M. Hayashi, M. Douke, K. Ariyasu, N. Hamaguchi: "TV Program Making Language -Benefits of Object-based Content-creation Method", IBC2001 (2001)
- 4) M. Douke, N. Hamaguchi, M. Hayashi, K. Ariyasu: "Production of TV Programs on a Single Desktop PC -Special scripting language, TVML, generates low-cost TV programs-", Broadcast Asia 2002 Conference (2002)
- 5) M. Hayashi: "TVML Technology and Applications", 2002 International Workshop on Broadcasting, Seoul (2002)
- 6) N. Badler, B. Webber, J. Kalita and J. Esakov: "Animation from Instructions. In Making them Move", N. Badler, B. Barsky and D. Zeltzer (Eds.), 51-93, Cambridge, MA: MIT Press (1991)
- 7) B. Coyne and R. Sproat: "WordsEye: An Automatic Text-to-Scene Conversion System", SIGGRAPH 2001 Conference Proceedings, pp.487-496 (2001)
- 8) M. Ma: "Automatic conversion of natural language to 3D animation", PhD thesis, University of Ulster, Ulster, Ireland (2006)
- 9) K. Glass, S. Bangay: "A Method for Automatically Creating 3D Animated Scenes from Annotated Fiction Text", IADIS International Journal on Computer Science and Information Systems, **4**, 2, pp.103-119 (2012)
- 10) N. Hamaguchi, M. Douke, M. Hayashi: "Internet-Based Private TV Station Enabling Individuals to Produce, Distribute and View TV Programs", IADIS2005 (2005)
- 11) H. Kaneko, N. Hamaguchi, M. Doke: "Sign language animation using TVML", VRCAI '10 Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry, pp.289-29 (2010)
- 12) M. Douke, H. Kaneko, N. Hamaguchi, S. Inoue: "Engaging Viewers Through the Connected Studio: Virtual Participation in TV Programs", Consumer Electronics Magazine, IEEE, **1**, 4, pp.30-39 (2012)
- 13) H. Yasuda: "Digital Content Creation/Distribution in a Broadband-Ubiqitous Environment", IEICE Transactions on Information and Systems, **E90-D**, 1, pp.76-80 (2007)
- 14) T2V Home Page: <http://t2vlab.jp/en/> (last visited: 2013)



Masaki Hayashi received Dr. Eng. from Tokyo Institute of Technology in 1999. He worked at NHK Science and Technology Research Laboratories to conduct research of content creation from 1986 to 2006. After the career as a director and a research fellow at several profit corporations, he began working at the Department of Game Design, Uppsala University, Campus Gotland as Associate Professor.



Seiki Inoue is a senior research engineer in NHK Science & Tech. Research Labs. He got B.S. degree in electrical engineering, M.S. and Ph.D. degrees in electronics engineering all from Univ. of Tokyo in 1978, 1980 and 1992, respectively. He joined NHK in 1980 and moved to Science & Tech. Research Labs. in 1983. From 1995 to 1998, he worked for ATR Media Integration & Communications Research Labs. From 2002 to 2005, he worked at NHK Broadcast Engineering Department. His research interests include image and video processing, CG and visual expression, and text-to-video technology (TVML: TV program Making Language).



Mamoru Doke received his Masters degree from electronics engineering from Nagoya University in 1992. He joined NHK the same year. Since 1998, he had been with NHK Science and Technology Research Laboratories. He had joined NHK Engineering Services from 2007 to 2009. He received Ph.D. from Electro-communication University in 2008. He is engaged in research on CG based video content generation and handling, human interface.



Narichika Hamaguchi had conducted research and development about content production, description, distribution, and user behavior analysis of social TV at NHK Science and Technology Research Laboratories (STR) since 2000. He had Professional Master degree of MOT (Management of Technology) and Ph.D. in Engineering degree at Tokyo Institute of Technology. Currently, he is in charge of R&D planning at STR.



Hiroyuki Kaneko joined NHK in 2002. He has been engaged in research on computer graphics based content production technology, especially on sign language animation in NHK Science and Technology Research Laboratories since 2006. Currently, he is in charge of domestic and international public relations at STRL.



Steven Bachelder holds a professorship in artistic expression in convergent media at Uppsala University, Campus Gotland where he cofounded the game design education in 2001. Today he is subject responsible for Expression in Convergent Media and head of the Convergent Media Lab. Bachelder has been teaching in the interactive arts and convergent media since 1995. He is currently conducting research on engagement in digital games.



Masayuki Nakajima received Dr. Eng. degree from the Tokyo Institute of Technology, Tokyo, Japan in 1975 and has been Professor at the Department of Computer Science, Graduate School of Information Science & Engineering, Tokyo Institute of Technology during 1997-2012 March. He began working at the Department of Game Design, Uppsala University, Campus Gotland, (Sweden) in April 2012.