Oksana Arnold
Wolfgang Spickermann
Nicolas Spyratos
Yuzuru Tanaka (Eds.)

Communications in Computer and Information Science 372

# Webble Technology

First Webble World Summit, WWS 2013
Erfurt, Germany, June 2013
Proceedings

Springer

Oksana Arnold   Wolfgang Spickermann
Nicolas Spyratos   Yuzuru Tanaka (Eds.)

# Webble Technology

First Webble World Summit, WWS 2013
Erfurt, Germany, June 3-5, 2013
Proceedings

Springer

Volume Editors

Oksana Arnold
Erfurt University of Applied Sciences, Germany
E-mail: oksana.arnold@fh-erfurt.de

Wolfgang Spickermann
University of Erfurt, Germany
E-mail: wolfgang.spickermann@uni-erfurt.de

Nicolas Spyratos
University of Paris South, Orsay, France
E-mail: nicolas.spyratos@lri.fr

Yuzuru Tanaka
Hokkaido University, Sapporo, Japan
E-mail: tanaka@meme.hokudai.ac.jp

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-38835-4 e-ISBN 978-3-642-38836-1
DOI 10.1007/978-3-642-38836-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013939514

CR Subject Classification (1998): H.5.1, K.3, H.3, H.5, H.4

*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

The Webble World Summit 2013 held in Erfurt, Germany, in June 2013 was set up as the first world-wide meeting for potentially all scientists and engineers working on Webble technology and its applications. From the very beginning, the event was declared open to all participants from other disciplines interested in the technology and potential applications in their particular domains. In response, the conference's preparation has been accompanied by intense communication, for instance, with interested businessmen from several enterprises in Thuringia and with some historians working in several universities. Whereas businessmen ask themselves for potentials of Webble technology to establish business cases, the historians are interested in using Webble-based infrastructures to open new horizons of research based on huge databases, large corpora of historical maps, and a growing number of geo-referenced assets.

The first day of the Webble World Summit was dedicated to the participants' visit to the Thuringian Creative Innovation Summit in Weimar, where science meets industries, industries meets business, and business meets science. Some panels aimed at informing a wider audience about the essentials of Webble technology. This opening day was followed by two full days of conventional conference program.

Richard Dawkins' seminal book *The Selfish Gene*, 1976, has spread the insight that our human life is bringing with it as a side-effect nonbiological evolution in areas such as architecture and fashion, for instance. The units of nonbiological evolution are called memes. Accordingly, Yuzuru Tanaka's attempts to promote cultural evolution by computer technologies, to systematically investigate and explore the digital encapsulation and evolution of human knowledge, led to the introduction of meme media ideas and implementations. The most recent forms of meme media are called Webbles–the focus of this conference.

This volume comprises the contributions to the Webble World Summit 2013. There were four distinguished speeches–three keynotes and one dinner speech–and ten regular papers grouped in sections. The keynotes and the dinner speech together are intended to provide a rather comprehensive approach to the field going into depth, on the one hand, and providing a bird's eye perspective, on the other hand.

The conference program was opened by a series of papers intended to challenge contemporary Webble technology and applications. Mina Akaishi, Yoshihiro Okada, and Masahiko Itoh presented their own research work and their own applications implemented by means of IntelligentPad and IntelligentBox technologies, respectively. Those achievements may be seen as challenges to contemporary Webble technology based on Silverlight or HTML 5.

The next session was opened by a keynote delivered by Micke Kuwahara. Emphasis was put on the details of Webble technology. This session was meant to inform a wider audience about the essentials of the technology to allow for informed decisions about potential engagements in research, development, applications, and business cases. The keynote was complemented by three contributions presented by Micke Kuwahara and Yuzuru Tanaka, Nicolas Spyratos and Tsuyoshi Sugibuchi, and Jun Fujima.

The second keynote by Klaus P. Jantke addressed recent applications in technology-enhanced learning with emphasis on the feature of direct execution and particularly aiming at approaches to game-based learning. Two papers by Oksana Arnold and Vincent Reichelt and by Jonas Sjöbergh and Yuzuru Tanaka completed the session on applications.

The next session addressed a variety of visions about the future development and exploitations of Webble technology. It was opened by the third keynote delivered by Nigel M. Waters on geographic information systems (GIS) knowledge and the future federation of GIS knowledge. A few further visions were contributed by Klaus P. Jantke and by Sebastian Arnold and Jun Fujima.

The dinner speech by Yuzuru Tanaka concluded the conference program and wrapped up three days of communication and interaction. Besides informing on the history of theory and technology evolution, the dinner speech aims to familiarize the participants with the rich and exciting field of meme media technology making them members of a world-wide community.

The editors gratefully acknowledge the remarkable financial support by the Thuringian Ministry of Economics, Labor and Technology (Thüringer Ministerium für Wirtschaft, Arbeit und Technologie, TMWAT) given to the University of Erfurt for preparing and running the conference.

April 2013

Oksana Arnold
Wolfgang Spickermann
Nicolas Spyratos
Yuzuru Tanaka

# Table of Contents

# A Meme Media Based Application Framework for Context-Driven Information Access

Mina Akaishi

Faculty of Computer and Information Sciences, Hosei University
3-7-2 Kajino-cho Koganei-shi Tokyo 184-8584, Japan
mina@hosei.ac.jp

**Abstract.** This paper introduces several information access methods constructed by Meme Media systems. A context-based information access space provides users with a component-based interactive 3D virtual environment for the information access based on the context model. A chronicle analysis tool provides several views of records from different viewpoints. To construct such systems, Meme Media system provides powerful environment. This paper shows several application frameworks and implementation examples.

## 1 Introduction

Currently, a vast amount of information is being accumulated at an accelerating pace in many fields. As a result, piles of records are stored. However it is difficult to understand all of their content within the limit of time. Moreover, it is harder to create new ideas or to discover new knowledge from the chaos of information. To solve this problem, this paper shows some information access methods based on Meme Media systems, such as IntelligentPad/IntelligentBox [5].

One is a context-based information access space, which is a 3D virtual environment to support users' information access activities on the basis of a context model. Another is a chronicle analysis tool which provides several views of records from different viewpoints.

The implementation has been carried out in IntelligentBox, a constructive visual software development system for interactive 3D graphic applications. It can be freely combine with any 3D components in IntelligentBox architecture. This means that the user can extend the functionality of her system by adding existing or future implemented components. Moreover, users can reuse arbitrary parts of system as other application components.

The remainder of this paper is organized as follows. Section 2 explains a context-based information access space. Section 3 describes a chronicle data analysis. Finally, Section 4 concludes the paper.

## 2 A Context-Based Information Access Space (CIAS)

A CIAS is a 3D virtual environment that allows access to information based on context models [1] and path-language [2]. This mechanism can be a basis of flexible structured virtual museum or virtual shopping mall based on knowledge in the information base. In this section, we explain these concepts and their component based implementation framework.

## 2.1 The Notion of Context

A context consists of an identifier *c* plus *acontent*. The content of c is a set of triplets of the form
$< names, objectidentifier, reference >$
where *names* is a set of descriptions for the object and $reference$ is a context identifier or Nil. The context referenced by an object contains information relevant to that object. It is important to note that *names* and $references$ are context dependent. An object can belong to different contexts and may have different names and/or different references in each context. This feature is useful when we want to view an object from different perspectives

Accessing information in a contextualized information base often involves navigating from one object to another by following references. From an object within a given context, we can reach any object that belongs to its reference and, recursively, any object that lies on a path. Navigation is based on the notion of path.

## 2.2 Overview of CIAS

In order to create a CIAS, we map the components of a context into individual virtual spaces. A context identifier is mapped into an individual virtual space, called a *context space*. Each triplet in the content of the context is mapped into an individual virtual space, called an *object space*. Therefore, an object space is a 3D visualization of a triplet, a context space includes a set of object spaces, and a CIAS is a set of context spaces.

A CIAS supports context traversal as multiple virtual space traversals: accessing a context, and accessing an object within the context. When a user is inside a context space, the user will see a set of Object Space Ports (OSPs), which is pointers to object spaces. The user may focus on one of the OSPs in the context. Then the user is navigated to an object space by choosing one of the OSPs. If a user is interested in the current object, then the user can go to a relevant context space through a Context Space Port (CSP) in a current object space. In its context space, there are OSPs of relevant information again. Users can choose the context space, enter it, and follow the links depending on their interests.

## 2.3 The Mechanism for Creating a CIAS

In this section, we focus on the development of generating Information Access Space from existing Contextualized Information Bases. It is impossible to materialize all spaces and objects corresponding to all data in a large information base. We propose a mechanism for dynamic creation of a CIAS according to a user's current position. Users' information access activities form the virtual information access spaces.

To construct a CIAS, the necessary functions are divided into the following points. The first is a space management function to create multiple virtual spaces, make connections between spaces, and traverses a space to reach other spaces. The second is an access to an information base to obtain context data that satisfy the query condition. The third function is a database record reification mechanism to instantiate the data in a context. After necessary data are obtained, the data should be materialized by appropriate media objects.

We have been implementing these functions as software components, using the IntelligentBox(IB) system [5] as the basis for constructing CIAS. The IB is a constructive visual software development system for interactive 3D graphic applications. The IB represents objects as interactive 3D visual objects, called Boxes, which can be manually combined with other Boxes. It provides a uniform framework for the concurrent definition of both geometrical compound structures among Boxes and their mutually interactive functional linkages. Each Box has its own state values that are stored in variables, called slots. Each Box also has a unique function associated with its slot value. We use the IB system as the basis for constructing CIASs, since it allows the integration and collaboration of existing and future developing tools. It allows expansion of the application fields of this system.

Figure 1 shows the functional linkage of components to construct CIAS based on the context data in the contextualized information base. This mechanism provides the dynamic creation of necessary context spaces and object spaces by following the user's information space traversal. In the following sections, we describe each mechanism and process to materialize a CIAS.

**Space Ports.** To construct a CIAS, we need a mechanism that addresses efficient access to information objects within a restricted 3D display space. Space ports enable us to embed multiple 3D spaces in a single virtual environment and navigate through these different spaces. An embedded space can be represented by a 3D media component. The user can see the contents of the embedded space from outside, and jump into the embedded space to change the current working place. Such space port functions are provided as OSP/CSP Boxes in IB [6].

The right-hand side of Figure 1 shows a prototype system displaying a CIAS. In this example, the context space contains four OSPs that are connected with corresponding object spaces. When a user approaches one OSP, the user will be navigated to the object space that includes instantiations of an object. The reference is a CSP that also contains OSPs. When a user approaches that reference CSP, the user can enter it. In the same way, users can continuously explore the information spaces via the paths.

**Access to Information Base.** A Query Definition Box generates query statements by a predefined template query and arguments. A Query Definition Box

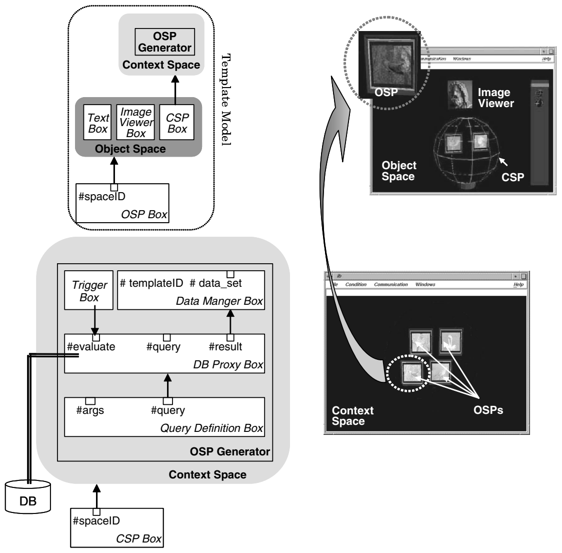


**Fig. 1.** The components structure for generating the Object and Context Spaces

receives a context ID as an input, and then generates a query statement to obtain contents of the specified context.

A Database Proxy Box works as an interface between a database system and the IB system. The DB Proxy Box is connected to the contextualized information base. Through a DB Proxy Box, users can use a set of database functions. The DB Proxy Box receives a query and sends the evaluation of the query to a Data Manager Box.

We control the timing of a query evaluation according to the user's current working space. We combine a trigger control component with a DB Proxy Box. When a DB Proxy Box receives a trigger, it starts a query evaluation. A Trigger Box issues a signal only when a condition is satisfied. In the case of our implementation, when a user enters the space that includes a CSP, then the Trigger Box in the connected context space issues a trigger signal. In this way, the necessary space is created dynamically as the user explores data. The timing of triggering can be changed by changing a condition definition.

**Reification of the Contents of a Context.** A Data Manager Box reifies database records as Boxes. The Data Manager Box stores an identification of the template Box(es) to materialize the database record. Arbitrary Boxes can be pre-registered in the Template Box Table, which is a global variable in the IB system. When the Data Manager Box receives the result of the query evaluation from the DB Proxy Box, the Data Manager Box makes copies of pre-registered template Boxes. Each copy of the template Box is then instantiated with the value of the corresponding element of the collection.

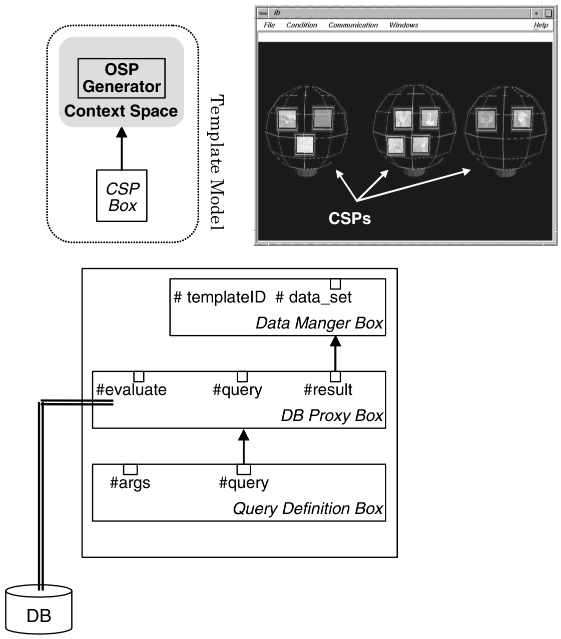


**Fig. 2.** The component structure for generating the Context Spaces

A Data Manager Box materializes the content of context corresponding to the given data as 3D components depending on predefined template models. The upper left part of Figure 1 shows the composite structure of the template Boxes in order to materialize an OSP and an object space.

**The Design of an Object Viewer.** It is up to a system designer how object viewers are designed. In the case of Figure 1, an Image Viewer Box is used as an object viewer. It has an image file name specified by an object identifier and visualizes it. In our system, all functions are designed as 3D components, allowing free combination with other components. This means that the user can extend the functions by adding existing or future implemented components.

**Context Space Generation Mechanism.** Figure 2 shows the component structure for obtaining a collection of context spaces that satisfy a given query. The Data Manager Box stores a CSP (a CSP Box that is connected with a context space) as a template Box. It then gives the set of context spaces that satisfy the query condition. For example, let us consider a query for selecting all contexts that include a specified object *o*. We call such a set of contexts the *facets* of *o*. It gives users an overview of different aspects of the object *o* in different context spaces. After query evaluation, the DB Proxy Box receives all context data that includes the object $o$, that is, its *facets* of $o$. The result data are sent to the Data Manager Box. The Data Manager Box then makes copies of the template Box, the detailed structure of which is shown in the upper-left of Figure 2. Each CSP Box materializes the context space by following the beforementioned procedures. Users can choose the context space, enter it, and follow the links depending on their interests.

## 3 Information Access for Chronicles

Vast amounts of records human experience are stored all over the world. To understand and use such chronicles effectively, several relations embedded in records must be found. This paper describes a chronicle analysis tool, which provides several views of records from different viewpoints.

### 3.1 A Framework for Chronicle Analysis

A framework fr analyzing chronicle data consists of three layers. The first layer provides a function for filtering data by specifying conditions. A parameter *scope* consists of two kinds of queries: a period and a viewpoint. A period is a query concerning the date attribute, and a viewpoint is a query concerning the other attributes.

The second layer visualizes the distribution of filtered data as a combination of matrix and bar chart graphs. If users can find interesting patterns on the distribution map in the second layer, the third layer represents transitions of relations among focal attribute values by using the *topic sequence* technique [4]. The focal attribute is specified by a parameter focus. A *topic sequence* is regarded as a graphical plot of a series of text chunks. Each chunk is visualized by a *word colony* [3], which visualizes the dependency relationships among terms in a text as a graph. A *word colony* is regarded as a visual abstract of a relation among terms in a text. A *topic sequence* is formed by the connections of *word colonies.* A *topic sequence* provides overviews of the topical change of text chunk content over time.

At the beginning, users see the patterns of distribution for whole data. These patterns give an overview of a target data set. If users can find interesting patterns on the distribution map by selecting an attribute for the vertical axis, then they can narrow their interest down by filtering data or they can see more detailed information by visualizing a transition of interesting attribute values. The distribution map or transitions of attribute values suggest users' next keywords for specifying another viewpoint. A reiteration of this process corresponds to a story/hypothesis/interpretation generation cycle. If users can find interesting patterns of transitions of attribute values, then these patterns form the foundations of new stories.

## 3.2 Chronicle Analysis Tool

**Data Filtering.** By filtering data, characteristic patterns will occur on a distribution map. However, it is difficult to specify a keyword that represents a user's interest. Even if there is no query, a distribution map representing an overview of whole data will suggest candidate keywords to specify a user's interest.

Let us assume that a user sees the distribution map of whole data in the Dai-Nihon Shiryo database[1] and finds an interesting keyword, "gift." Then, those records that include the keyword "gift" and contain information on an event that occurred between 1550 and 1650 are selected from the database.

**Distribution Map.** Figure 3 shows a distribution map of events related with each person. The vertical axis represents the names of people in the order of frequency, and the bar below a name shows the number of records that include the specified person's name.

The distribution map represents personal activities concerning "gift." It is easy to see the co-occurrence of Japanese emperors and men of power, e.g. Emperor Ogimachi – Nobunaga Oda, Emperor Goyozei – Ieyasu Tokugawa, etc. This distribution suggests that relationships based on "gift" indicate the relation of authorities and powers.

**Transition of Attribute Values.** A *topic sequence* [4] represents the content of a document as a graphical plot of a document. It supports users' efforts to guess the context (plot) of the document and to find desired parts of documents.

The records in the same cell of a distribution map are grouped together into a document. Then, each document is visualized by a *word colony* [3], which is a directed graph representing dependency relationships among terms in a document. The *topic sequence* is formed by the connections of such *word colonies.*

Figure 4 shows four *topic sequences* representing human relations of two people (Nobunaga Oda and Ieyasu Tokugawa) on the basis of "gift" and "battle." The major people in Nobunaga's human relations concerning "gift" are authorities at that time, such as Ashikaga Shogun and Emperor Ogimachi. In the case of Ieyasu's human relations concerning "gift," the major people are authorities only earlier in his life, after which Ieyasu's son took prominence for most of his remaining years. Ieyasu was the first shogun in the Tokugawa shogunate. He consolidated his son's position as his successor.

*Topic sequences* concerning "battle" show that Nobunaga is related with many daimyos (territorial lords in pre-modern Japan), while Ieyasu is related with a few daimyos. In light of knowledge about the territory these daimyos had control over, the visualization gives rise to the interpretation that Nobunaga fought various enemies all over Japan, while Ieyasu fought a big enemy next to his territory, which actually corresponds to history as we know it. These patterns help users create new interpretations of history by exploring chronological records.

[1] The most comprehensive collection of pre-modern Japanese historical materials.

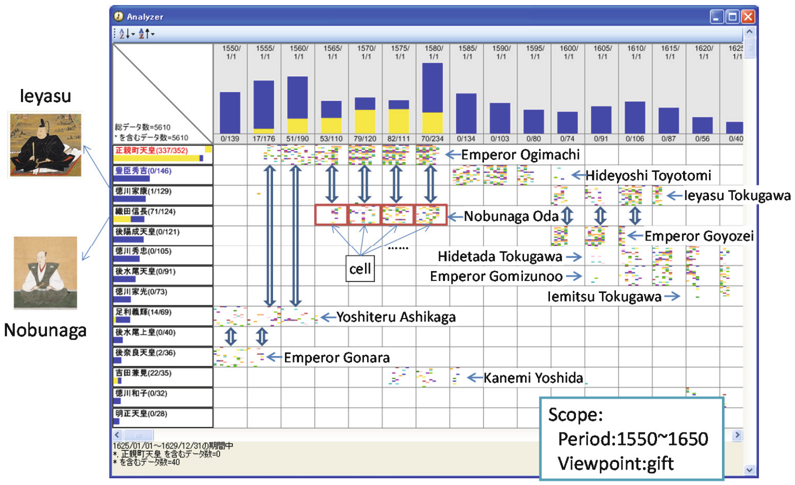


**Fig. 3.** Distribution map of people's relations filtered by the *scope*

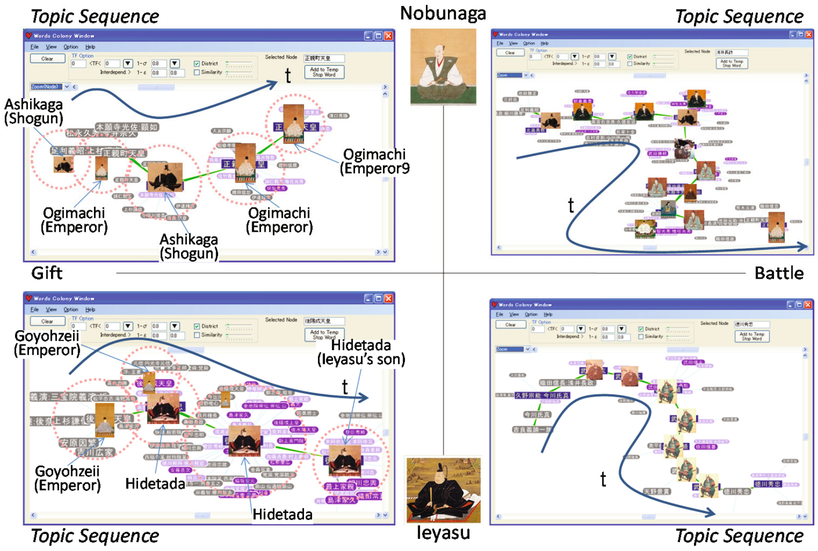


**Fig. 4.** Comparison of human relations of two persons based on "gift" and "battle"

Users can find several relationships among people on the basis of various viewpoints. This is helpful for considering the stories or inspiration for new ideas on the basis of records of historical events. Not only each *topic sequence* but also a combination of *topic sequences* enables users to compose a more complicated story.

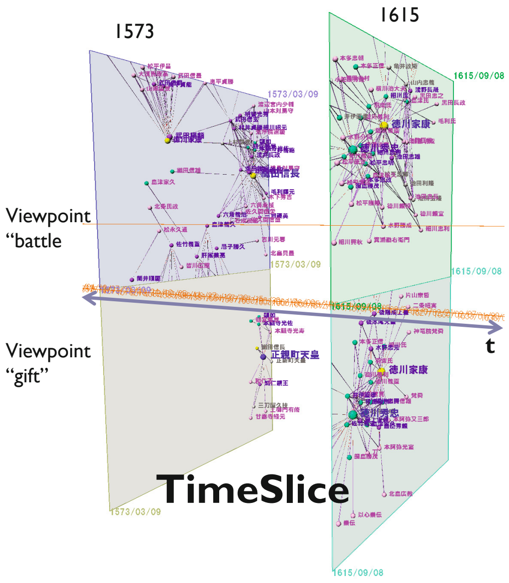


**Fig. 5.** A Chronicle Data Analysis by TimeSlice

### 3.3 3D Extension for Analyzed Chronicle Data

Time-tunnel [8] and TimeSlice [9] are visualization tools for time-series data. Those systems are constructed on the basis of a Meme Media Environment. [10] visualized human relations by using TimeSlice as a plat-form for visualizing temporal changes in the structure of a human network in a 3D space. A TimeSlice is a 2D plane for visualizing a snapshot of a network with a specified timing. A temporal change of network is visualized by dragging a TimeSlice along the timeline. Multiple TimeSlices visualize human relations from different viewpoints and with different timings. These mechanisms help users to compare multiple situations easily. Figure 5 shows TimeSlices which represent human networks in 1573 and 1615 with different viewpoints; "battle" and "gift".

## 4 Conclusions

This paper shows the concept of context-based information access space and its implementation framework. Through the OSPs in the context spaces, users can follow the reference links and obtain the information object and paths in which they are interested. The system dynamically creates the information spaces as the need arises.

Moreover the paper introduced a framework for visual chronological analysis to support finding interesting relations embedded in chronicles. On the basis of the proposed framework, we implemented a tool called "CAT", which provides a functionality for representing latent patterns in chronicles on the basis of dynamics of *scope*, *axis*, and *focus*. It enables users to interpret new histories from various viewpoints.

The integration and collaboration of existing and future tools over a hypermedia environment would contribute to the efficient usage of the human knowledge stored in computer systems and would enhance productivity and collaboration. In the frameworks represented in this paper, some components are represented by components of the Meme Media system such as IB. In the Meme Media system, all functions are provided as Meme Media components. Therefore, any component can be combined with other functional components such as 3D animation tools, CSCW support tools, and the scientific visualization tools. This might expand the application fields.

## References

1. Teodorakis, M., Analyti, A., Constantopoulos, P., Spyratos, N.: Context in Information Bases. In: Proc. of the 3rd Int. Conference on Cooperative Information Systems (coopIS 1998), pp. 260–270 (1998)
2. Theodorakis, M., Analyti, A., Constantopoulos, P.: Querying Contextualized Information Bases. In: Proc. of the 24th Intern. Conference on Information and Communication Technologies and Programming (ICT&P 1999), pp. 260–270 (1999)
3. Akaishi, M., Satoh, K., Tanaka, Y.: An associative information retrieval based on the dependency of term co-occurrence. In: Suzuki, E., Arikawa, S. (eds.) DS 2004. LNCS (LNAI), vol. 3245, pp. 195–206. Springer, Heidelberg (2004)
4. Akaishi, M.: Narrative based Topic Visualization for Chronological Data. In: Proc. IV 2007, pp. 139–144 (2007)
5. Okada, Y., Tanaka, Y.: IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications. In: Proc. Of Computer Animation 1995, pp. 114–125 (1994)
6. Itoh, M., Tanaka, Y.: World Mirror and World Bottle: Components for Embedding Multiple Spaces in a 3D Virtual Environment. IPSJ 42(10), 2403–2414 (2001)
7. Ohigashi, M., Tanaka, Y.: A Framework for the Virtual Reification of Database Records. IPSJ 42(SIG 1 (TOD8)), 80–91 (2001)
8. Akaishi, M., Okada, Y.: Time-Tunnel: Visual Analysis Tool for Time-Series Numerical Data and its Aspects as Multimedia Presentation Tool. In: Proc. IV 04, pp. 456–464 (2004)
9. Itoh, M., Toyoda, M., Kitsuregawa, M.: An Interactive Framework for Visualizing Time-series of Web Graphs in a 3D Environment. In: Proc.IV 2010, pp. 54–60 (2010)
10. Itoh, M., Akaishi, M.: Visualization for Changes in Relationships between Historical Figures in Chronicles. In: Proc. of IV 2012, pp. 283–290 (2012)

# Web Version of *IntelligentBox* (*WebIB*) and Its Integration with Webble World

Yoshihiro Okada

Innovation Center for Educational Resource (ICER), Kyushu University Library, Motooka 744, Nishi-ku, Fukuoka, 819-0395, Japan
okada@inf.kyushu-u.ac.jp

**Abstract.** This paper treats a 3D graphics software development system called *IntelligentBox* known as a 3D meme media system and its web version (*WebIB*). *IntelligentBox* provides various 3D software components called *boxes* each of which has a unique functionality and a 3D visible shape. *IntelligentBox* also provides a dynamic data linkage mechanism that allows users to develop interactive 3D graphics applications only by combining already existing *boxes* through direct manipulations on a computer screen. Recently, the author extended *IntelligentBox* system to make it possible to develop interactive web 3D contents. This is called the web version of *IntelligentBox* (*WebIB*). This extended mechanism is also regarded to be available for the integration of *IntelligentBox* with Webble World, an open web-based meme media environment. This paper explains essential mechanisms of *IntelligentBox* and *WebIB*, introduces some interactive web 3D contents realized using *WebIB*, and also discusses about the possibility of the integration of *IntelligentBox* with Webble World.

**Keywords:** 3D graphics, Toolkit system, Component ware, Web contents, 3D Meme Media.

## 1 Introduction

Advances in recent computer hardware technology have made possible 3D rendering images in real time. Consequently, 3D software has become in great demand although its development is more laborious work than 2D software development. For this reason, we proposed a 3D graphics software development system called *IntelligentBox* known as a 3D meme media system [1, 2]. Application fields of *IntelligentBox* include 3D-CG animation creation, virtual reality software development, interactive simulator development, and so on. However, the developed 3D graphics applications could not be available on the web. If they were available on the web, the usefulness of *IntelligentBox* would become higher. Therefore, we extended *IntelligentBox* system to make it possible to develop interactive web 3D contents [3, 4]. This is called *WebIB* as the web version of *IntelligentBox*. In this paper, we explain essential mechanisms of *IntelligentBox* and *WebIB*, and introduce some interactive web 3D contents including educational contents and information visualization tools.

On the other hand, recently, an open web-based meme media environment called Webble World was proposed [5] and its components are called webbles. If *IntelligentBox* system is possible to integrate with Webble World, its usefulness would become higher than ever. Fortunately, the same mechanism as that of *WebIB* seems available for the integration. So, in this paper, we also discuss about the possibility of the integration of *IntelligentBox* system with Webble World.

The remainder of this paper is organized as follows: Section 2 introduces related work. In Section 3, we explain essential mechanisms of *IntelligentBox* and its extended mechanisms for *WebIB*. Section 4 introduces several interactive web 3D contents realized using *WebIB*. In Section 5, we discuss about the possibility of the integration of *IntelligentBox* system with Webble World. Furthermore, in Section 6, we discuss about development costs of web 3D contents, their performances and significance of *IntelligentBox* system and *WebIB* as development environments for web 3D contents. Finally, we conclude the paper in Section 7.

## 2 Related Work

Our research purpose is to propose a software architecture that makes it easier to develop 3D graphics applications including interactive web 3D contents. Its related systems are 3D graphics toolkit systems and programming libraries like Open Inventor [6], Coin3D [7] and 3D Widget [8]. Open Inventor is an OpenGL based object oriented programming library. Coin3D is also library very similar to Open Inventor. 3D Widget is a Widget-based toolkit system for the 3D GUI development. Some of them provide an authoring tool that enables to design 3D graphics contents. Even using such authoring tools, it is not easy to develop 3D graphics applications because developers have to write text-based programs for that. As for development tools for interactive web 3D contents, there are library systems like Java3D [9], Jogl [10], Papervision3D [11] and WebGL [12]. Java3D and Jogl is Java-based 3D graphics library that works as a plug-in virtual machine running on a web browser for 3D graphics contents. Papervision3D is Flash-based 3D graphics library that also enables to develop web-based 3D graphics contents. WebGL (Web Graphics Library) is JavaScript API for rendering interactive 3D graphics and 2D graphics within any compatible web browser without the use of plug-ins. These are library systems so that the user has to write text-based programs for developing web-based 3D graphics contents.

Our research system *IntelligentBox* and its web version (*WebIB*) provide various 3D software components called *boxes* represented as visible, manually operable, and reusable functional objects. Furthermore, they provide a dynamic data linkage mechanism called slot connection. These features make it easier for even end-users to develop 3D graphics applications including web 3D contents. This is the main difference of *IntelligentBox* and *WebIB* from others. We also have a plan to integrate *IntelligentBox* system with Webble World using the same mechanism as that of *WebIB*. This is new feature mainly shown in this paper.

## 3 Essential Mechanisms of IntelligentBox and WebIB

*WebIB* employs the same essential mechanisms of *IntelligentBox*. This section explains those mechanisms in the following two subsections. They are very simple but very useful. After that, we describe extended mechanisms for *WebIB*.

### 3.1 Model-Display Object (MD) Structure

As shown in Figure 1, each *box* consists of two objects, a model and a display object. This structure is called MD (Model-Display object) structure. A model holds state values of a *box*. They are stored in variables called slots. A display object defines how the *box* appears on a computer screen and also defines how the *box* reacts to user operations. An example of *RotationBox* in Figure 1 has a slot named 'ratio' that holds a double precision number used as a rotation angle. Through direct manipulation on the *box* using a mouse device, its slot value is changed. Furthermore, its visual image changes simultaneously with the change of the slot value. In this way, a *box* reacts to the user's manipulation according to its functionality.

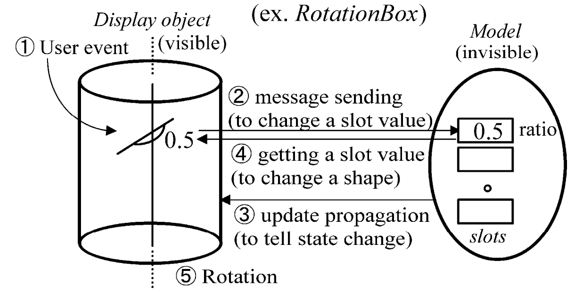


**Fig. 1.** An *MD* structure of a *box* and its internal messages

### 3.2 Dynamic Data-Linkage Mechanism Called Slot Connection

Figure 2 illustrates a data linkage concept among *boxes*. As shown in the figure, each *box* has multiple slots. Its one slot can be connected to one of the slots of other *box*. This connection is called a slot connection. The slot connection is carried out by three standard messages, i.e., a set message, a gimme (give me) message and an update message, when there is a parent-child relationship between two *boxes*. These messages have the following formats:

(1) Parent *box* set <slotname> <value>.
(2) Parent *box* gimme <slotname>.
(3) Child *box* update.

A <value> in a format (1) represents any value, and a <slotname> in formats (1) and (2) represents a user-selected slot of the parent *box* that receives these two messages. A set message writes a child *box* slot value into its parent *box* slot. A gimme message reads a parent *box* slot value and sets it into its child *box* slot. Update messages are issued from a parent *box* to all of its child *boxes* to tell them that the parent *box* slot value has changed. By these three messages, the two slots of a child *box* and its parent *box* are connected and their two functionalities are combined.

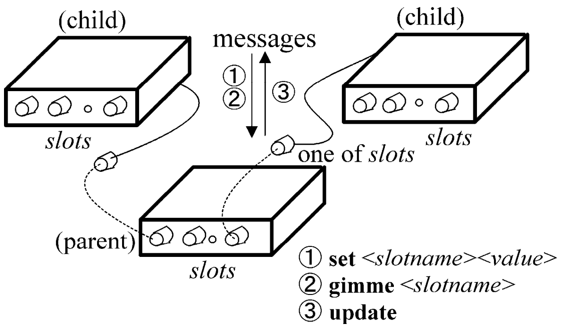


**Fig. 2.** Standard messages between *boxes*

### 3.3 Extended Mechanisms for *WebIB*

*IntelligentBox* system uses OpenGL 3D graphics library which provides an off-screen rendering functionality. As shown in Figure 3 and 4, using this functionality, a rendered image of a 3D scene can be generated on a web-server and transferred to a web-browser through the Internet. On the web-browser, besides a HTML program, a JavaScript program runs to manage user operation events, i.e., a mouse move, a mouse button click and so on. Such user operation events will be transferred to the web-server using XMLHTTPrequest message through a CGI-program of the web-server. The CGI-program (Perl script) once receives the user operation events and applies them to *IntelligentBox* system running on the web-server. And then, *IntelligentBox* system will generate next off-screen rendering image of the 3D scene to be updated by the received operation event. In this way, the user can interactively manipulate 3D contents of *IntelligentBox* that runs on the web-server through his/her web-browser. Since *WebIB* supports most web-browsers, e.g., Internet Explorer (IE), Mozilla Firefox, Opera, Google Chrome and Safari, *WebIB* is available on any mobile device like iOS device and Android OS device. However, when using iOS device or Android OS device, the touch interface should be used. Therefore, *WebIB* also supports the touch interface for iOS devices and Android OS devices.

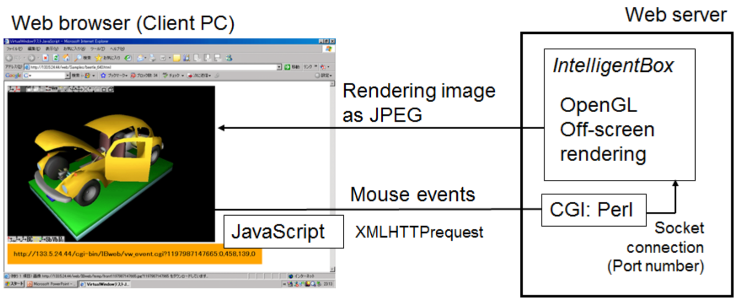


**Fig. 3.** Extended mechanisms of *WebIB*

### 3.4 Mechanisms of *WebIB* for Multiple Users

As shown in the left figure of Figure 4, *WebIB* can provide multiple users with a web-based collaborative virtual environment based on the same mechanism of Figure 3. *WebIB* has a System ID No. (SID No.), and by specifying it, each client user can access its corresponding *WebIB* that has the same SID No. through his/her web-browser. Using this data-linkage, communications between a teacher and a learner become possible in the cases of educational 3D contents. The right figure of Figure 4 shows another case that each of multiple users individually uses his/her own *WebIB*. As shown in the figure, each of multiple *WebIB*s can run on the same web server, and using different System ID No.(SID No.), each user can access his/her own *WebIB* through his/her web-browser.

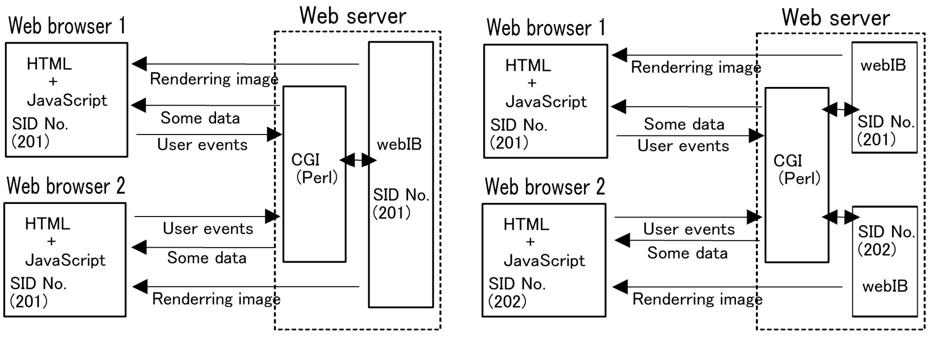


**Fig. 4.** Data-linkage among one *WebIB* and its two clients (Left), and data-linkage among two individual *WebIBs* and two dedicated clients (Right)

## 4 Web Contents Realized Using *WebIB*

This section presents actual examples of interactive web contents realized using *WebIB*. In this paper, we especially introduce educational contents because they are regarded to be practical contents. We also introduce information visualization contents related to our one project about cyber physical systems.

### 4.1 Educational Contents

As visual contents for educations, there are two types of contents, i.e., 3D contents and 2D contents. *WebIB* is used as a 3D model viewer for 3D contents and as an image viewer for 2D contents. In the following, we introduce these examples.

**1) 3D Model Viewer**

The left figure of Figure 5 shows an example 3D model for students to learn a brain and its some parts represented as web 3D contents using *WebIB* on IE browser. A student can operate interactively for the change of his/her viewpoint, i.e., rotation, zoom and pan, and can point out any parts by a mouse device click to display their part names of the brain. As described in the previous section, using the same SID No., these operations are shared among other users. If one user is a teacher and other users are students, the all students can see the teacher's operations and they can understand part names pointed out by the teacher. The right figure of Figure 5 is another example, a hurt model. Similarly to the left figure, each student can learn the structure of a hurt and its part names individually on the web at any time and at any location.

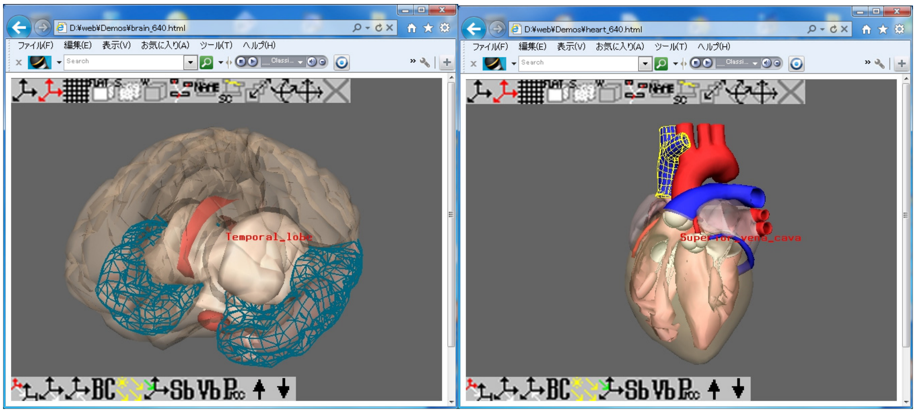


**Fig. 5.** Web 3D content examples of medical education

### 2) Image Viewer

As educational contents, images are common and often used. In this case, *WebIB* can be used as an image viewer. Although *WebIB* contents are 3D graphics contents, using the texture mapping mechanism, any image data can be displayed in a 3D space. The left figure of Figure 6 shows an image of a certain typhoon [13] provided by NASA and the right figure of Figure 6 shows the same typhoon image looks from the very far viewpoint. It is possible to manipulate the image for the pan and the zoom like Google Maps operations. If a teacher have high resolution images as educational materials, he/she can provides such images with his/her students on the web like Google Maps service. Students can interactively make a pan or a zoom towards any part of an image that they want to look. Figure 7 shows another examples, CT images as educational contents of medicine.

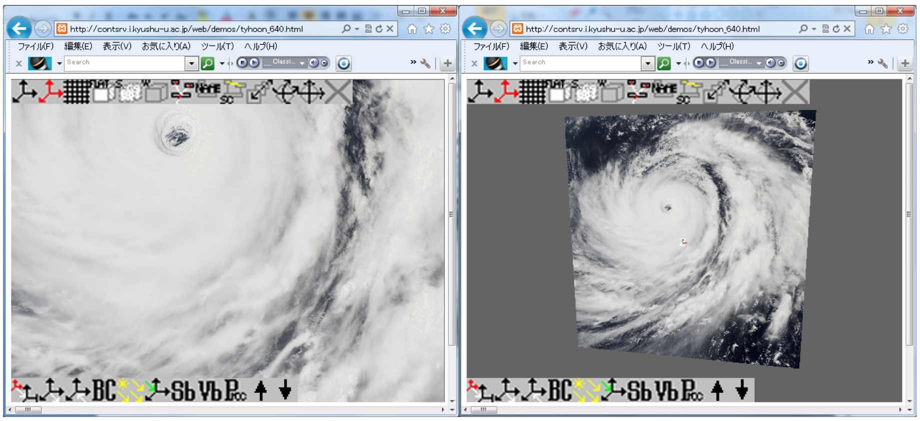

http://contsrv.i.kyushu-u.ac.jp/web/demos/tyhoon_640.html
http://contsrv.i.kyushu-u.ac.jp/web/demos/tyhoon_640.html

**Fig. 6.** Typhoon images as educational content examples

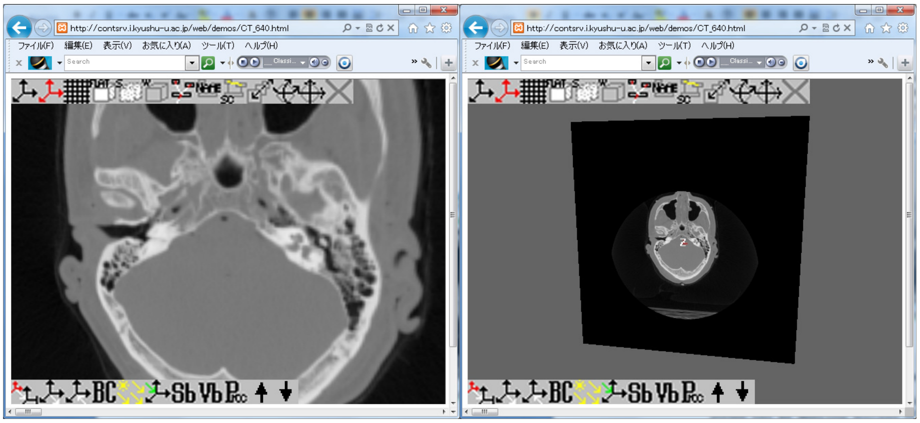

http://contsrv.i.kyushu-u.ac.jp/web/demos/CT_640.html
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Search

**Fig. 7.** CT images as educational content examples

### 4.2 Information Visualization Contents

We have one project about cyber-physical systems. One of the important topics in researches on cyber-physical systems is the analysis of big data to be collected from the physical world. As one of the analysis methods, information visualization is useful. Currently, we have been developing two types of visualization tools using *IntelligentBox* system for our cyber-physical system project. The first one is for the analysis of human movements because human movements are very important cues for the analysis of human activities. The left figure of Figure 8 shows a screen image of such web content. This is the building of our graduate school consisting of several floors. To simplify a 3D model for the building, we employ 2D floor map images and use the texture mapping mechanism. It is very easy to visualize the building and display human movements. In this case, glyphs have a different color, a different size and a different shape to specify several attribute values of their corresponding persons and move on a floor. So, we can understand persons' activities from glyphs' movements.

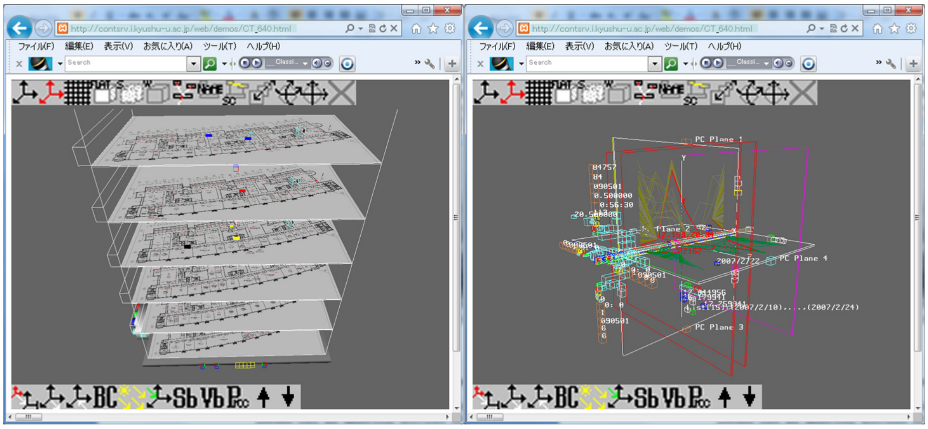

http://contsrv.i.kyushu-u.ac.jp/web/demos/CT_640.html
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Search
PC Plane 1
PC Plane 2
PC Plane 3
PC Plane 4
2007/2/22
(2007/2/10),...,(2007/2/24)

**Fig. 8.** Information visualization contents as web services

The other one is for the analysis of multi-dimensional/multi-attributes data because data to be collected from the physical world are various types and have various attributes. We have already proposed one information visualization tool called *Time-tunnel* for the analysis of time-series numerical data and extended it by adding a visualization functionality similar to Parallel Coordinates. The right figure of Figure 8 shows a screen image of such web content. See the papers [14, 15] for its detail.

## 5 Integration of *IntelligentBox* with Webble World

Webble World is an open web-based meme media environment and its components are webbles developed using Microsoft Silverlight technology. Originally, Silverlight supports XMLHTTPrequest communication mechanism. Therefore, as shown in the upper figure of Figure 9, a communication between a webble existing on a web browser and *IntelligentBox* system runs on a web server is possible through the same mechanism shown in Figure 3. In addition, the latest version of Silverlight supports the standard socket communication mechanism. Therefore, a webble also can communicate with *IntelligentBox* system runs on any server without any CGI-program as shown in the lower figure of Figure 9. In this way, *IntelligentBox* system can be integrated with Webble World technologically although we have not yet implemented such webbles. In the near future, we will develop such wobbles and their applications.

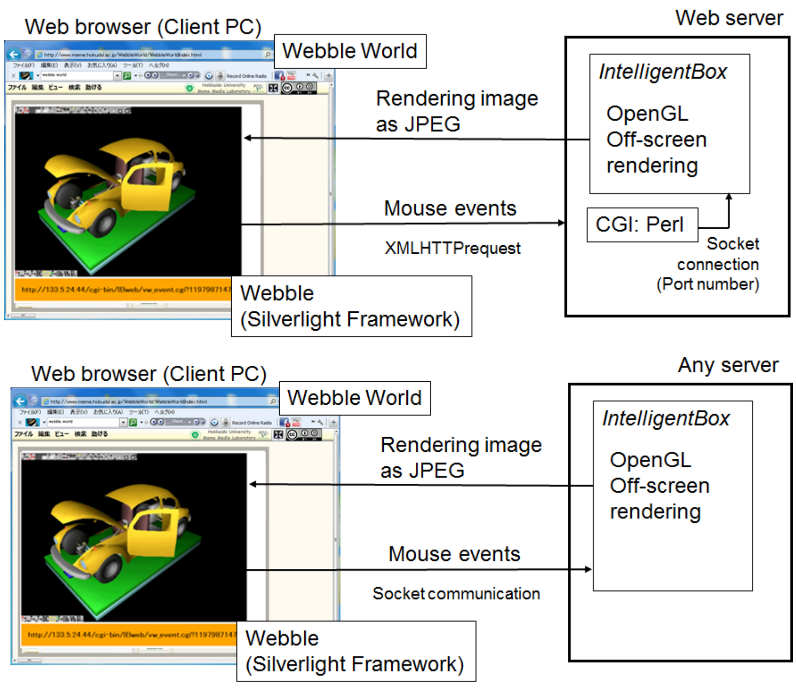


**Fig. 9.** Integration of *IntelligentBox* with Webble World

## 6 Discussion

### 6.1 Development Costs

For preparing the example models of Figure 5, 6 and 7, it was not necessary to write any text-based program. They were constructed only through direct manipulations on a computer screen using a mouse device. Therefore, even teachers who do not have any programming knowledge can construct web 3D contents including 3D educational contents by using *WebIB*.

### 6.2 Performances

The most important factor should be considered as for the performance of *WebIB* is update rate (refresh rate or frames per second) of off-screen rendering images on a web-browser transferred from a web-server. This value depends on a network environment. However, there is no problem because the system employs JPEG compression and can appropriately reduce the network bandwidth required to transfer an off-screen rendering image.

### 6.3 Significance

There are many services on the web. However, there are very few services of 3D graphics contents on the web. One of its reasons is that currently most web-browsers natively do not support the 3D graphics rendering. If we want to receive 3D graphics content services on the web, we have to install any plug-in software like Java3D, Jogl [11], etc. This is inconvenient especially for end-users who are not accustomed to doing so. Even if this is not inconvenient, there is another reason that 3D graphics contents need 3D polygonal model data but they are usually not allowed to be distributed through the Internet due to their user-licenses. Even if the distribution of 3D polygonal model data is allowed, there is a problem that the transmission time of them become very long. On the other hand, our web version of *IntelligentBox* works as a SaaS application does not have the above problems because only off-screen rendering images of 3D scenes are transferred from a web-server to a client web-browser as explained in Sec. 3. So, interactive 3D contents already developed as applications of *IntelligentBox* can be reused as web 3D contents those work as SaaS applications. Its MashUp with other SaaS applications are also possible. As a result, the usefulness of *IntelligentBox* has become higher than ever. The integration of *IntelligentBox* with Webble World mainly proposed as new feature in this paper is also regarded as one of such MashUp examples.

## 7 Conclusion

This paper introduced software architecture that makes it easier to develop 3D graphics applications including interactive web 3D contents. Our research system *IntelligentBox* provides various software components each of which corresponds to each of required functionalities. Combining these software components by direct manipulations on a computer screen enables users to develop 3D graphics applications without writing any text-based program. This feature is important for end-users who do not have any programming knowledge. Furthermore, we enhanced the usefulness of *IntelligentBox* by extending it to make it work as a development system for web 3D graphics applications like web 3D educational contents. This is called *WebIB*, the web version of *IntelligentBox*. In this paper, we showed some interactive web 3D contents to clarify the availability of *WebIB* and also indicated the possibility of the integration between *IntelligentBox* system and Webble World, an open web-based meme media environment.

As future works, we will improve the transfer mechanism of an off-screen rendering image to reduce the required bandwidth and to improve the performance. Also, we will develop more practical web 3D contents to clarify the usefulness of *WebIB*. Moreover, we will develop webbles that can communicate with *IntelligentBox* system runs on a web server and their applications to show the usefulness of the integration of *IntelligentBox* with Webble World.

**Acknowledgment.** This work was partially supported by FOST(Foundation for the Fusion Of Science and Technology), Japan.

## References

1. Okada, Y., Tanaka, Y.: IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications. In: Proc. of Computer Animation 1995, pp. 114–125 (1995)
2. Okada, Y., Tanaka, Y.: Collaborative Environments in IntelligentBox for Distributed 3D Graphic Applications. The Visual Computer 14(4), 140–152 (1998)
3. Okada, Y.: IntelligentBox as Development System for SaaS Applications Including Web-based 3D Games. In: Proc. of the 9th Annual European GAMEON Conf., pp. 22–26 (2008)
4. Okada, Y.: Web version of IntelligentBox (WebIB) for Development of Web 3D Educational Contents. In: Proc. of IADIS Int. Conf. Mobile Learning 2011, pp. 251–255 (2011)
5. Webble World, `http://cow.meme.hokudai.ac.jp/WebbleWorldPortal/`
6. Open Inventor, `http://oss.sgi.com/projects/inventor/`
7. Coin3D, `http://www.coin3d.org/`
8. 3D Widget, `http://www.viewpoint.com/widgets/`
9. Jave3D, `https://java3d.dev.java.net/`
10. Jogl, `https://jogl.dev.java.net/`
11. Papevision3D, `http://blog.papervision3d.org/`
12. WebGL, `http://www.khronos.org/webgl/`
13. `http://eoimages.gsfc.nasa.gov/images/imagerecords/15000/15467/nabi_amo_02sep05_250m.jpg`
14. Akaishi, M., Okada, Y.: Time-tunnel: Visual Analysis Tool for Time-series Numerical Data and Its Aspects as Multimedia Presentation Tool. In: Proc. of 8th International Conference on Information Visualization (IV 2004), pp. 456–461. IEEE CS Press (2004)
15. Notsu, H., Okada, Y., Akaishi, M., Niijima, K.: Time-tunnel: Visual Analysis Tool for Time-series Numerical Data and Its Extension toward Parallel Coordinates. In: Proc. of CGIV 2005, pp. 167–172. IEEE CS Press (2005)

# IntelligentBox Based Interactive 3D Information Visualization Frameworks for Comparison and Exploration: An Overview

Masahiko Itoh

Institute of Industrial Science, University of Tokyo
imash@tkl.iis.u-tokyo.ac.jp

**Abstract.** This paper introduces an overview of IntelligentBox based interactive 3D visualization frameworks for exploring large information space, and shows example applications using such frameworks. We first introduce WorldMirror and WorldBottle that embed 3D information spaces as 3D components into a 3D environment to navigate information in the 3D space. Second, we introduce spreadsheet based coordinated multiple 3D information visualization framework. Third, we show TimeSlices that are 2D planes in the 3D space for exploring changes in time sequential data with structures.

**Keywords:** Visualization, CMV, 3D system, Interactive System, IntelligentBox, TimeSlice.

## 1 Introduction

Explorative information visualization systems have been considered as an important layer of researches in the era of Big Data to find out new knowledge from huge information space. Information visualization is not static representation of information such as InfoGraphics. Information visualization requires interactive seeking processes to navigate new knowledge, such processes are mentioned as Shneiderman's mantra [8]; "Overview first, zoom and filter, details-on-demand".

IntelligentBox [10] is a component-based visual software development system for interactive 3D graphics applications. Each 3D component in IntelligentBox environment called box is connected each other through slot connection to construct 3D applications such as 3D visualization systems. The constructed 3D applications as composite boxes are editable and reusable. Users can copy composite boxes, and change attribute values for comparing data. IntelligentBox therefore match for base infrastructure for constructing interactive visualization system in a 3D environment.

Much work related to IntelligentBox based information visualization researches have been introduced [12,3,15,11] after Ohigashi et al. have introduced first system at WISS 1997 (Japanese domestic workshop on interactive systems and software).

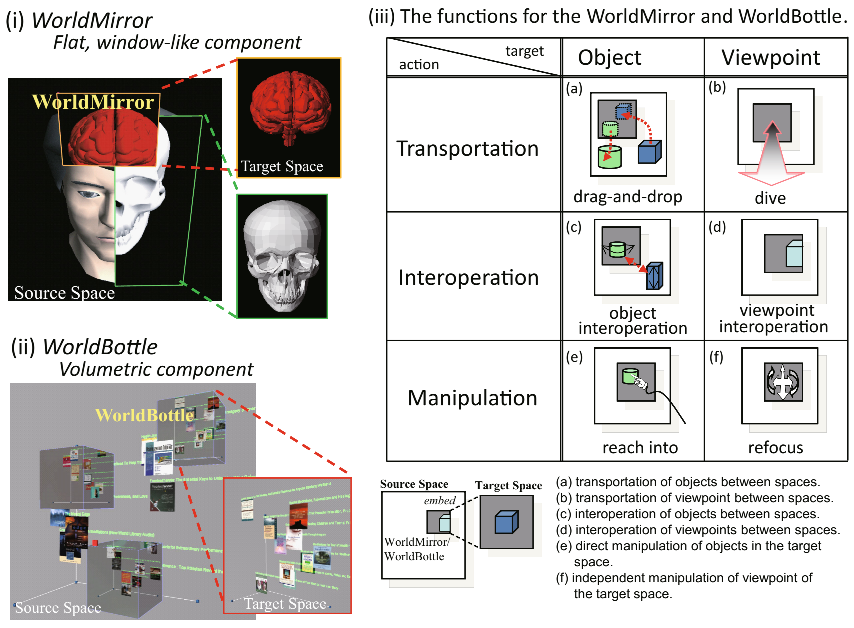

(i) WorldMirror
Flat, window-like component
WorldMirror
Target Space
Source Space
(ii) WorldBottle
Volumetric component
WorldBottle
Source Space
Target Space
(iii) The functions for the WorldMirror and WorldBottle.
target
action
Object
Viewpoint
Transportation
(a) drag-and-drop
(b) dive
Interoperation
(c) object interoperation
(d) viewpoint interoperation
Manipulation
(e) reach into
(f) refocus
Source Space
Target Space
embed
WorldMirror/ WorldBottle
(a) transportation of objects between spaces.
(b) transportation of viewpoint between spaces.
(c) interoperation of objects between spaces.
(d) interoperation of viewpoints between spaces.
(e) direct manipulation of objects in the target space.
(f) independent manipulation of viewpoint of the target space.

**Fig. 1.** The WorldMirror, the WorldBottle, and their functions

This paper introduces our previous work on coordinated multiple visualization frameworks for exploring information in the large information space based on the IntelligentBox architecture. Section 2 presents WorldMirror and WorldBottle that embed 3D information spaces as 3D components into a 3D space to navigate information in the 3D space through doors in 3D space. We next introduce spreadsheet based 3D information visualization framework to compare and explore various types of information and representations in Section 3. We then provide TimeSlice that is a visualization component for exploring temporal changes in structures and values such as changes in hyperlink structures on the Web in Section 4.

## 2 WorldMirror and WorldBottle

We have introduces two types of visual components; a flat, window-like component WorldMirror (Fig. 1 (i)) and a volumetric, arbitrary shape WorldBottle (Fig. 1 (ii)) [1]. They enable users to construct a hyperlinked space in the 3D virtual environment, and to navigate through such a 3D hyperlinked space. By using WorldMirror/Bottle, any arbitrary 3D space can be embedded into any arbitrary 3D component which renders the space on their surface, so that users can look into the contents of other spaces from their current working space, and enter them. They also allow users to manipulate contents in embedded spaces from their current space, and to move an object from an embedded space to

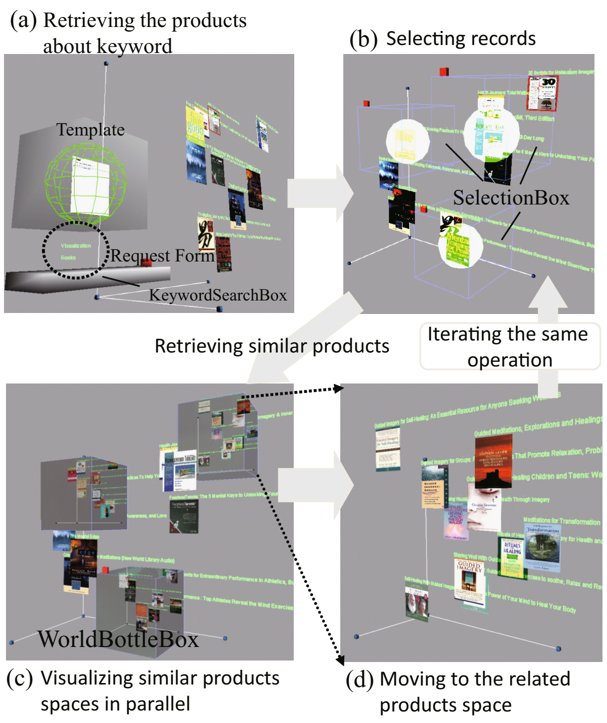

(a) Retrieving the products about keyword
Template
Request Form
KeywordSearchBox
(b) Selecting records
SelectionBox
Retrieving similar products
Iterating the same operation
WorldBottleBox
(c) Visualizing similar products spaces in parallel
(d) Moving to the related products space

**Fig. 2.** Successive navigation through WorldBottle using the Similar Products Search

the current working space, vice versa. To construct the navigatable structure through a large information space, WorldMirror and WorldBottle provide the functions described in Fig. 1 (iii).

Example in Fig. 2 shows an application for exploring similar products in a 3D environment using the Amazon Web service (AWS).First, the user can examine products from keywords using the KeywordSearch. The records retrieved from a KeywordSearch are visualized in the current space (Fig. 2 (a)) using a mechanism described in [3]. If the user is interested in the related products information about a visualized prouct, the user can select a product dynamically and search the related products about the selected product from the SimilaritySearch (Fig. 2 (b)). The records retrieved from the SimilaritySearch are visualized in another space. When the user approaches the selected product, WorldBottles appear, and the user can observe the related products about any selected one from the current space (Fig. 2 (c)). If the user wants to compare related products about several products, the user can select more than one product and visualize in multiple spaces (Fig. 2 (b)(c)). The user can enter the related product space, in which there are many products that the user prefers (Fig. 2 (d)). The user can explore these related products recursively (Fig. 2 (d)-(b)).

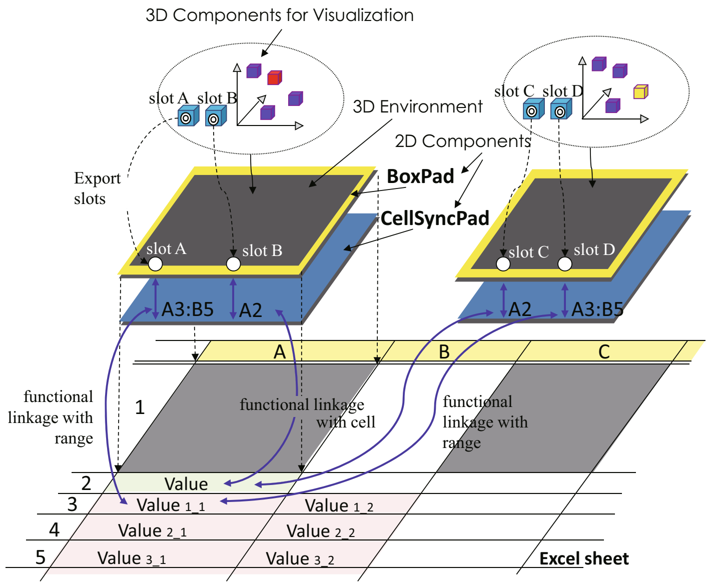


**Fig. 3.** Overview of architecture for spreadsheet-based coordinated parallel visualization

## 3 Coordinated Multiple 3D Visualizations on Spreadsheet

We have proposed a spreadsheet-based visualization framework for end-users to generate and modify multiple 3D visualizations of data-sets from various Web resources [2]. Our framework enables users to use various values for different parameters and/or different 3D components in parallel to enable exploratory searches for solutions by making intercomparisons of multiple results.

Our approach enables users to embed 3D visualization environments into spreadsheet cells (Fig. 3). We utilize an Excel spreadsheet environment to define multiple coordinated visualizations. A 3D visualization environment is an IntelligentBox environment itself. We can construct 3D visualizations in this environment. Our approach also allows users to only export the necessary functions of 3D visualization mechanisms into the embedded environments to input and output parameter values to and from spreadsheet cells. By using this facility, users can share the slot values of boxes in different visualization environments in different cells through spreadsheet cells. To define coordination, we provide interactive 2D/3D components and coordination rules that enable users to use operations for coordination such as brushing and linking, retrieving related or detailed information one after another, synchronizing viewpoints between 3D visualizations, and sharing table values between 3D components for visualizations and cells on Excel.

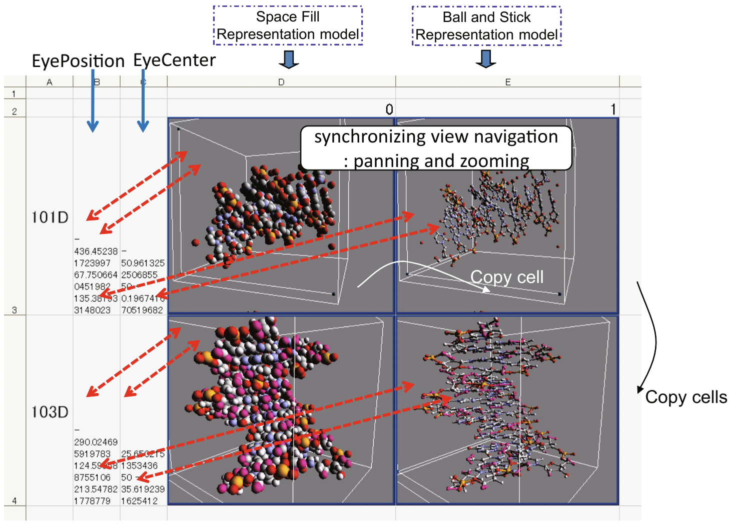

Space Fill Representation model
Ball and Stick Representation model
EyePosition
EyeCenter
A
B
C
D
E
1
2
0
1
synchronizing view navigation : panning and zooming
101D
Copy cell
3
103D
Copy cells
4

**Fig. 4.** Multiple visualizations of 3D structure of protein and/or DNA

Fig. 4 shows an example of visualizing 3D structures of protein/DNA using multiple data-sets, visual representations and angles. We can get the 3D structure data-sets for protein and DNA through Web service. In this example, we first input a PDB ID to visualizer components [3] in 3D environments through a cell. We can copy visualizations just through copying the cells to compare various data-sets with different representations. Visualizations of each row can have the different data-sets. We can change the 3D representation of protein structure just by inputting a template ID into the range D1:E1. We can also synchronize angle of view rendering among different cells through sharing parameters.

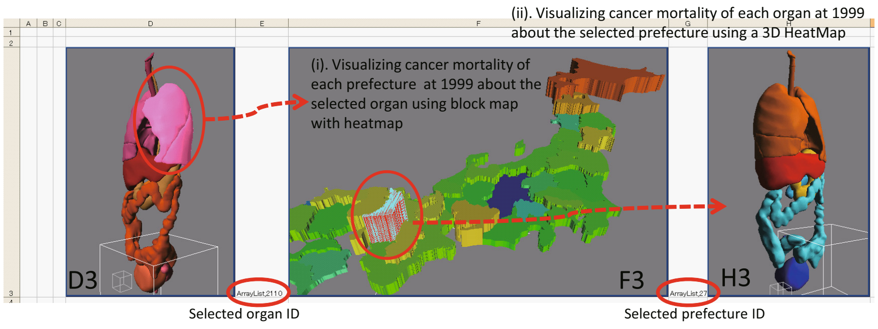

(ii). Visualizing cancer mortality of each organ at 1999 about the selected prefecture using a 3D HeatMap
(i). Visualizing cancer mortality of each prefecture at 1999 about the selected organ using block map with heatmap
D3
F3
H3
ArrayList,2110
ArrayList,27
Selected organ ID
Selected prefecture ID

**Fig. 5.** An example of visualizing cancer mortality of each prefecture and of each organ

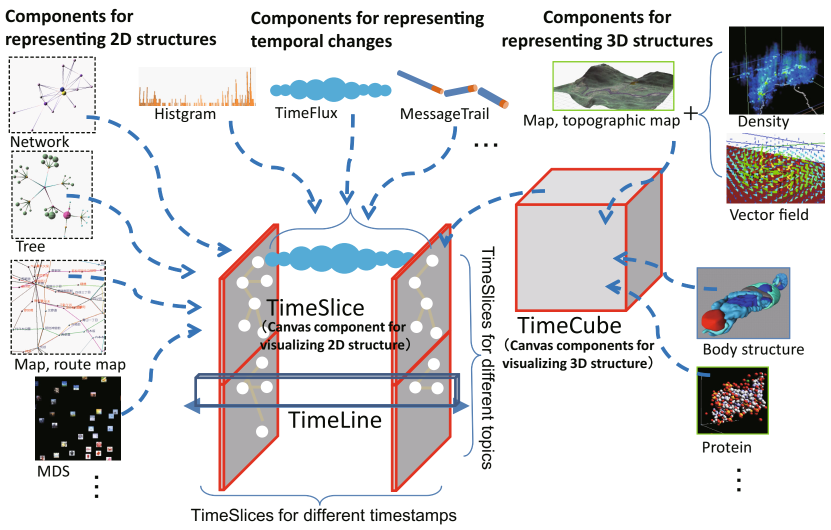

Components for representing 2D structures
Components for representing temporal changes
Components for representing 3D structures
Network
Histgram
TimeFlux
MessageTrail
Map, topographic map
Density
Vector field
Tree
Map, route map
MDS
TimeSlice
(Canvas component for visualizing 2D structure)
TimeLine
TimeSlices for different topics
TimeCube
(Canvas components for visualizing 3D structure)
Body structure
Protein
TimeSlices for different timestamps

**Fig. 6.** 3D visualization framework for for temporal changes in structures and values

Fig. 5 shows an example of visualizing the cancer mortality in Japan by prefecture and organ. Cell D3 shows an organ map. Users can select one organ to see the mortality from cancer for the organ they selected. In this example, cell F3 shows the cancer mortality for all prefectures in 1999 for the selected organ using a block map and a HeatMap. These results indicate that Osaka prefecture has the highest rate of mortality from lung cancer. Users can also select one prefecture in cell F3. Cell H3 visualizes the cancer mortality for all organs in 1999 for selected prefectures using the 3D HeatMap. Here, the mortality from liver cancer is higher than that from lung cancer in Osaka prefecture in 1999.

## 4 TimeSlice

We have introduced visualization framework for temporal changes in 2D/3D structures and values (Fig. 6). In this framework, we can construct 3D visualization applications through combining interactive Timeline component, 2D or 3D canvas components called TimeSlice [7] and TimeCube respectively. 2D canvas component can use components for representing 2D structures such as network, tree, and map. 3D canvas component can use component for representing 3D structures such as map, density, and vector field. Structures on canvas components change along position on the TimeLine. We can also add components for representing changes in values along the timeline such as histogram, and TimeFlux [4]. Moreover, we can copy canvas components to compare different timings and multiple topics.

Fig. 7 shows a system for analyzing temporal changes in the activities and interests of bloggers through a 3D visualization of phrase dependency structures

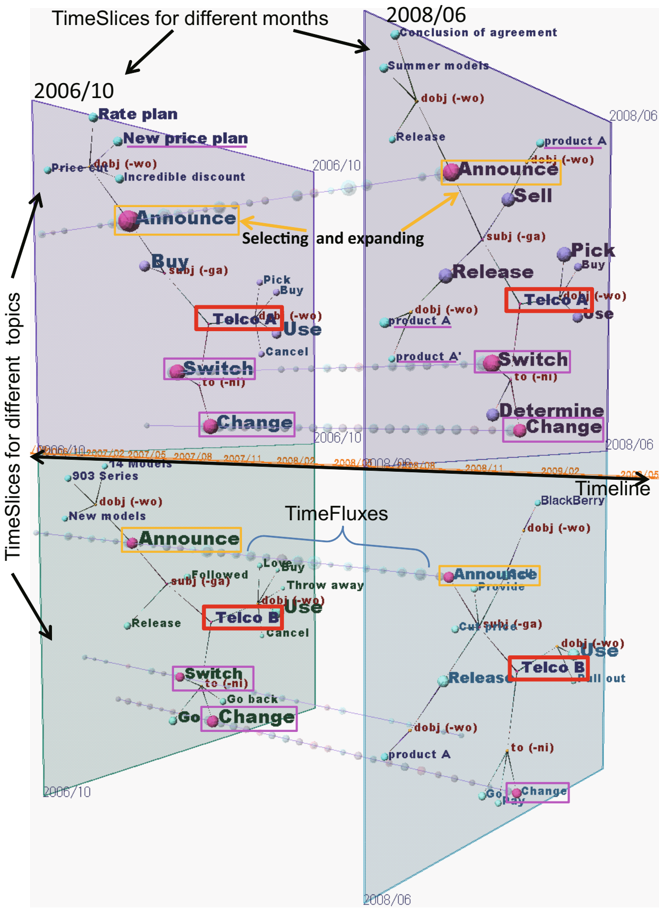
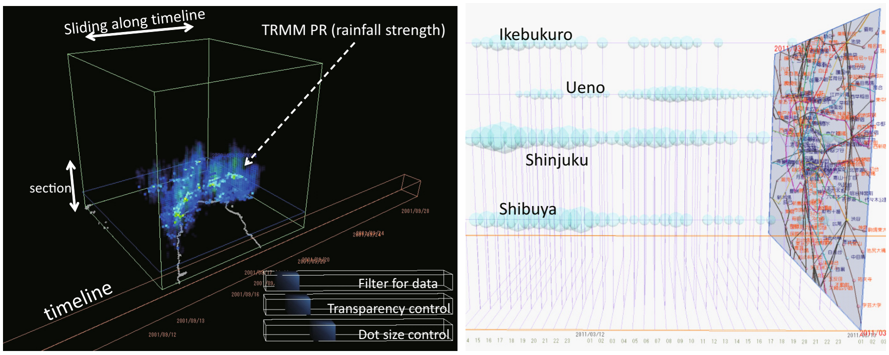


**Fig. 7.** Comparing Marketing Effect of two Telcos

in sentences [6]. The upper TimeSlice shows a topic for 'Telco A', while the lower one shows a topic for 'Tepco B'[1]. We can recognize events related to 'change/switch to Telco A' are more popular than 'change/switch to Telco B' in most months by observing changes in the structure and frequencies for events. To start exploring possible reasons for such actions and plans, we first find times when events "change/switch to Telco A" are popular, and then observe events around them on the TimeSlice in detail. We can see that there are some peaks in the events 'switch to Telco A' by observing the size of the spheres on the TimeFluxes. We place TimeSlices at the left on the position of timing when the first peak is observed (October 2006), and TimeSlices at the right for the second peak (June 2008). We find that 'Telco A' announced or released something because the size of the 'announce' and 'release' nodes increased. We next expand

[1] Companies' names have been anonymized.



(a) Visualizing temporal changes in rainfall strength over Taiwan

(b) Visualizing temporal changes in amount of tweets in Tokyo after Earthquake

**Fig. 8.** Example of visualizing temporal changes in values on 3D/2D geographical space

these nodes to find details on announcements and products that were released, and we then find that they announced a 'new price plan' in the first peak, and released 'product A' in the second peak. However, although 'Telco B' also announced new products very frequently, there are few peaks related to events 'switch' for 'Telco B'. These results mean that 'Telco A's' marketing activities had more impact in Japan than those by 'Telco B'.

Fig. 8 shows examples of visualizing temporal changes in values on 3D/2D geographical space. Fig. 8 (a) visualizes temporal changes in rainfall strength over Taiwan around typhoon season in which data is collected through DIAS project[2]. We can recognize rain concentrates along shoreline of Taiwan. Fig. 8 (b) visualizes temporal changes in amount of tweets in Tokyo after Earthquake on March 11, 2011. We can recognize that many people tweets whole through night around main stations in Tokyo because transportation systems stopped. We can also find out each station such as Shibuya, Shinjuku, Ueno, and Ikebukuro shows different trends in frequencies of tweets.

We have proposed an interactive visualization system to extract networks of historical figures from historical data and to show time-varying changes in their relationships [5] (Fig. 9). We use red for clusters related to "adversarial relationship" or "battle", blue for clusters related to friendship, green for others, and gray for keywords that are not categorized in any clusters. Fig. 9 shows temporal changes in characteristics of selected relationships between historical figures[3]. Fig. 9 (b) shows red edges between TOKUGAWA Ieyasu and TAKEDA Katsuyori. These represent their relationships in continual battles. Fig. 9 (c) shows changes in the relationship between ODA Nobunaga and UESUGI Kenshin for each year. Fig. 9 (d) visualizes complex relationships between Nobunaga and ASHIKAGA Yoshiaki.

[2] `http://www.editoria.u-tokyo.ac.jp/dias/index.html`

[3] They are powerful territorial lords in pre-modern Japan.

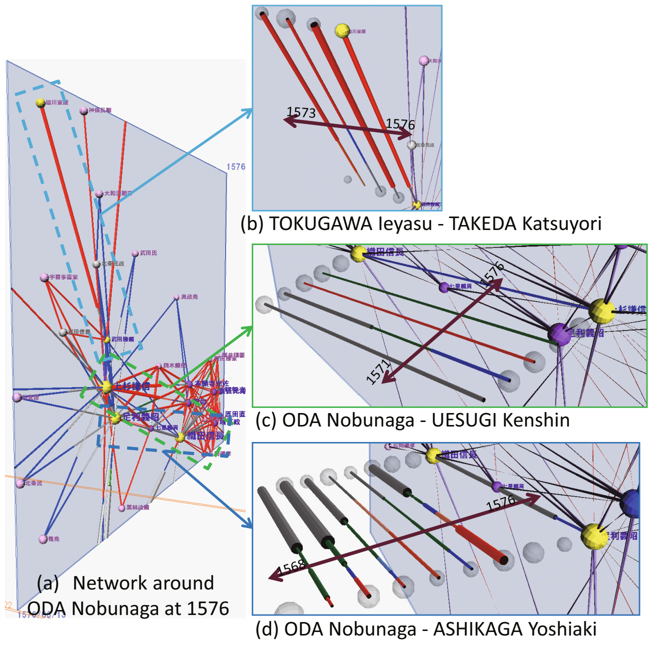

1573
1576
(b) TOKUGAWA Ieyasu - TAKEDA Katsuyori
1576
1571
(c) ODA Nobunaga - UESUGI Kenshin
1576
1568
(d) ODA Nobunaga - ASHIKAGA Yoshiaki
1576
(a) Network around ODA Nobunaga at 1576

**Fig. 9.** Changes in relationships between historical figures around ODA Nobunaga

## 5 Conclusion

We have introduced our interactive information visualization framework on 3D focusing on comparison and exploration using coordinated multiple views techniques based on the IntelligentBox architecture. The frameworks introduced in this paper are generic frameworks. They can derive for many explorative applications. WorldMirror and WorldBottle shown in Section 2 are applied to various types researches on exploring information spaces [13,14]. TimeSlices are also applied to various kinds of information with temporal changes such as relationships between peoples in social media [4] or historical data [5], extracted activities in social media by using NLP techniques [6], and hyperlink structures on the Web [7].

The challenges left are applying these frameworks to big data; how to visualize overview of absolutely huge amount of data, and seamlessly move on to zoom and filter mentioned in Shneiderman's mantra, or how to combine other analysis processes using such as machine learning to show important part of dataset mentioned in Keim's visual analytics mantra [9];"Analyze first - show the important - zoom, filter and analyses further - details on demand.".

## References

1. Itoh, M., Ohigashi, M., Tanaka, Y.: WorldMirror and WorldBottle: Components for Interaction between Multiple Spaces in a 3D Virtual Environment. In: Proc. IV 2006, pp. 53–61 (2006)
2. Itoh, M., Tanaka, Y.: A Framework for Constructing Coordinated Multiple 3D Visualizations on Excel. In: Proc. IV 2009, pp. 162–170 (2009)
3. Itoh, M., Tanaka, Y.: 3D Component-Based Visualization Framework for Generating Simple 3D Applications Using Web Services. In: Proc of WI 2006, pp. 823–830 (2006)
4. Itoh, M.: 3D Techniques for Visualizing Users' Activities on Microblogs. In: Proc of the IET International Conference on Frontier Computing - Theory, Technologies and Applications, pp. 384–389 (2010)
5. Itoh, M., Akaishi, M.: Visualization for Changes in Relationships between Historical Figures in Chronicles. In: Proc. of IV 2012, pp. 283–290 (2012)
6. Itoh, M., Yoshinaga, N., Toyoda, M., Kitsuregawa, M.: Analysis and Visualization of Temporal Changes in Bloggers' Activities and Interests. In: Proc. PVis 2012, pp. 57–64 (2012)
7. Itoh, M., Toyoda, M., Kitsuregawa, M.: An Interactive Framework for Visualizing Time-series of Web Graphs in a 3D Environment. In: Proc. IV 2010, pp. 54–60 (2010)
8. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proc. of VL 1996, pp. 336–343 (1996)
9. Keim, D.: Scaling Visual Analytics to Very Large Data Sets. In: Workshop on Visual Analytics, Darmstadt (2005)
10. Okada, Y., Tanaka, Y.: IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications. In: Proc. of CA 1995, pp. 114–125 (1994)
11. Notsu, H., Okada, Y., Akaishi, M., Niijima, K.: Time-tunnel: Visual Analysis Tool for Time-series Numerical data and its Extension toward Parallel Coordinates. In: Proc. CGIV 2005, pp. 167–172 (2005)
12. Ohigashi, M., Tanaka, Y.: A Framework for the Virtual Reification of Database Records. Trans. of Information Processing Society of Japan 42(sig1(20010115)), 80–91 (2001)
13. Ohigashi, M., Tanaka, Y.: 3D Space Framework for the Multi-facet Accessing of Database Records. In: Grieser, G., Tanaka, Y. (eds.) Dagstuhl Seminar 2004. LNCS (LNAI), vol. 3359, pp. 142–158. Springer, Heidelberg (2005)
14. Akaishi, M., Ohigashi, M., Spyratos, N., Tanaka, Y.: Information Access Space Framework over Contextualized Information Base. In: Proc. IV 2003, pp. 414–420 (2003)
15. Sugibuchi, T., Tanaka, Y.: Integrated Framework for the Visualization of Relational Databases and Related Web Content. In: Proc. of the 2003 Joint Conference of the Fourth International Conference, pp. 197–201 (2003)

# The Power of Webble World and How to Utilize It

Micke Kuwahara

Meme Media Laboratory, Hokkaido University, Sapporo, Japan
mkuwahara@meme.hokudai.ac.jp

**Abstract.** Webble World is a web-based environment where everything are Webble objects. Stand-alone objects that may be organized and manipulated freely into simple gadgets and/or complex applications, either by attaching to or detaching from, completely inside the browser with mainly standard mouse operations. All that is required by the user is a pursue of a dream and a bucket of creativity with a pinch of a mindset that views the environment as merely parts of the whole, and these parts are themselves just a bunch of smaller parts loosely connected. With that in the travel bag the journey to become a Webble application developer is afoot. Digging deep into the vast repository of Webble Objects, grab the ones that seem to fit your inner picture and start building anything from games, data analyzer systems, map visualizations, web tools, web sites, data managers and much, much more. This paper will demonstrate the structure of Webble World environment, how to access and enjoy the rich library of Webbles and applications produced during the last years and how to learn to self become a Webble World contributor.

**Keywords:** Meme Media Objects, Customize, Configure, Web, Share, Distribute, Resource, Interact, Participate, Federation, e-learning.

## 1 Introduction

This paper will put its main focus on thoroughly explaining what Webble World and a Webble really is, mainly on a theoretical and technical level but to put things in focus and perspective we will start by giving a background story to its birth and there also include some of the more philosophical and non-technical angles.

### 1.1 The Philosophy and Goal

Webbles are based on the philosophy of memes, a word which is coined by Richard Dawkins [9], that every thought and all knowledge which humans shares and which flows between us, may collide with other thoughts and knowledge and then reproduce or mutate, all in favor of survival and adaptation. The meme is a paraphrase which is supposed to make us see that human knowledge and cultural expressions are like genes in the way they evolve.

The idea of the meme has stimulated the research on how to improve the digital form of human knowledge to better fit the meme description so that knowledge may be easier spread, evolved and enhanced in a globally shared environment. One of these attempts of aligning humans natural ways of exchanging memes to work seamless within a partly non-human environment has been conducted by Yuzuru Tanaka and his group at Meme Media Laboratory in Hokkaido University, Japan. Through their work we were given the theoretical as well as numerous versions of the practical creation of the meme media concept known by the name IntelligentPad [1] [2] [5] [6]. These were developed both by the meme media laboratory on their own but also in cooperation with several partner companies of various size and status, like for example Hitachi, Fujitsu and the latest implementation PlexWare from 2003 by K-Plex in USA and Japan. No matter version or implementer the purpose of meme media was always to work as smart collaborative containers of all forms of digital knowledge.

From that philosophy and research has now risen an evolved and extended web based framework that allows every user to bring their ideas and knowledge to the table in order to create a true federation of humanity's collected knowledge, that is in constant progress and evolution. We have chosen to call this framework Webble World [3] [7] [8].

The compound documents accumulated in such a world-wide repository may work as genes to evolve their variety. They can be easily copied, which corresponds to the self-replication of genes. Their reediting by people corresponds to the recombination of genes. Some people may replace a component of a compound document with another to find out an unexpected functional effect. This may correspond to a mutation of a gene. Finally, Webbles are subject to people's evaluation; Some of them are frequently copied and reused, while others are gradually forgotten. This corresponds to the natural selection of genes.

A Webble as a Meme Media object, may contain more or less anything digitally available. It can be developed by a programmer to look and behave in any way imaginable within ordinary software development restrictions. When it has been deployed to the web it can be downloaded by any Internet user together with other Webbles directly in the browser and combined together to form new compound Webbles, also known as Webble Widgets, which the user can configure so that they may solve some tasks, or present some content the user wants to share. Further on, such compound Webbles can be additionally combined as a set of tools and/or interactive content presentation devices in order to form Webble applications. When satisfied the user can republish the creation to the web and then be found by other users who may use the Webbles either as they were intended or further improve and reedit with new features and new content. Through its reuse by other users the original Webble evolves step by step every time it gets republished, though previous versions originally developed by someone else are still accessible and are not removed or altered. This does not protect the Webbles of the past from the laws of evolution though, since Webbles with high usability rating and popularity will be found more easily and in greater numbers than those that are no longer in favor of the users and which will sink down into the layers of human productivity and finally be no more than an historical fact. It makes the Internet a true living and organic community where everyone participates and quite complex software application development can be done online directly in the browser together with the entire world.

As our understanding of culture and knowledge sharing gains, along with the transition to true interactivity with Web 2.0 and the steady evolution towards the semantic Web 3.0, the need for new, more open and more powerful tools emerges, tools that fill our need to build and communicate, to be inspired and further inspire others, to develop and change, edit and contribute. Such a tool needs to allow all types of users, from any background and with any sets of skills. It needs to be open and free so it may evolve alongside the rich source of knowledge it is carrying. It should be easy to access from anywhere in the world and it should be fully adaptable.

We believe Webble World could be such a tool.

## 1.2 The Birth of Webble World

Current version of Webble World has submerged over a period of 5 years, and has now reached an acceptable level of maturity to be engaged as a major component in several larger software projects around the world. This does not mean that we are at some final stage of development, for there are still many road blocks to force through before Webble World will be truly at the hands of every willing human being. Some of the later iterations of re-factoring the system though has allowed a much larger possibility to take full advantage of its core technology and to stretch its meme media capabilities even further into its outer realms. The online available running version is fully functional and in most aspects, from a users point of view, already displays most of the features a meme media, knowledge federation framework is imagined to display but there are specific issues, like mobile device availability and the more fuzzy, open source vs. proprietary issues of the tool and development base, that clouds the sky rather heavily, and which cannot be addressed within the current frame of technical solutions and therefore has forced the Webble World team to prepare the transition to a new technical core platform which will level up the system immensely. This will be further explained more in detail at a later chapter of this paper.

The existence of Webble World came to be as a result of wanting to prove the IntelligentPad concept as a recommended approach in knowledge sharing and system development, while participating in an EU funded project for the integrated IT support of clinical trials on cancer (ACGT - Advancing Clinico-Genomic Trials [10]), by developing a web based and highly graphical clinical trial management application for preparing, conducting and analyzing cancer research trials using the principles of editable and compoundable modules like IntelligentPads. Though earlier implementations of the IntelligentPad concept fully existed, as mentioned earlier, even the most recent one, in this case PlexWare, was already several years old, and though that maybe was not a major problem in itself, it also lacked many of the new requirements and suggestions formulated by the EU project, like free open source, complex graphics, full web compatibility, efficient community accessibility, web browser integration and independence of OS and browser applications, just to mention a few. But it was the fact that PlexWare was a commercial product that finally sealed its fate and made it impossible to use within this new project.

The year was 2007, and it was clear that a new and evolved version of these theories in system engineering had to be developed, where the major differences were enhanced object communication that allowed slightly more advanced forms of object manipulations between objects, full web integration to run the Webble framework in any browser, free online, easy access to achieve an expansion of the community of object developers and users, open source support in order to work along other open source solutions within the EU project, full compatibility with the existing web in order to be able to communicate with any other Web technology, no limitations in appearances and graphics so that complex visual objects could be designed and used, re-usability of other web resources so that Webble development may benefit from work already done in order to save time and money, and finally large scale independence from domain and content source restrictions so that Webbles and its content can be found and combined from any available online server. We jumped on the task to achieve that.

## 2 History of Webble World

Before we get to the mission at hand to explain how to use and take part in the current version of Webble World we feel it is beneficial to first let you in on how we got to this point and what problems we had to deal with.

### 2.1 The Way to Get There

It was not obvious to start with in what direction this development should go. Many technologies available out there contained promising issues that would make it possible for us to implement the platform we were imagine. But did any of them contain all of them? Back then, Flash was still a pretty strong contestant for creating rich web based systems. While HTML and JavaScript was, as always, the master of web approaches, their gigantic problems of browser implementation support made it fairly easy to see that using that approach while implementing Webble World, which at that moment of time went by another work name, would be an head ache too hard to bear. Also we had to consider the need for stronger powers and freedom regarding graphics in this new version. Vector graphic seemed like a good idea, but the support for SVG and similar was weak at best and definitely not the same between browsers and it also often required plug-ins, so if we needed plug-ins anyway maybe Flash was a better option. Java and applet concepts was considered but that never left the brainstorming meeting for various reasons, one being as simple as the fact that Java as a programming language was not well looked upon by the team, nor its tools and its way of inserting itself by brute force into a website (partly due to the way HTML 4 dealt with plug-in objects). Many APIs by both Google, Yahoo and other player's was considered and even mockup tested to see whether they could bring what we were looking for and let us avoid Flash. So why was Flash not the obvious choice? Simply because it was not a programming tool, it was a design tool for web designers with the light support of simple scripting. It was used for simple interactions, in simple relational situations in order to enhance the graphical appearance and user experience. It was not a tool any 'real' programmer would like to implement a platform of above described caliber in, and then also automatically force the rest of the world to use as their tool of choice as well. Flash just did not do it and none of the other plug-ins was either mature or promising enough to make us invest any serious time in them.

Then came Silverlight. It was a browser plug-in developed by Microsoft which used JavaScript and that promised full OS and browser support. It included its own implementation of vector graphics support, namely XAML, which was already known and popular through WPF (Windows Presentation Foundation). MS was giving this new technology its full attention and being a large, rather secure corporation made us believe that this could be it. We started creating the first draft of our Platform and came to realize that this had been what we were looking for. The tool base for Silverlight development was strong and became stronger, as well as its angle to target programmers at least as strong as web designers, which it really did and its rumors of being the Flash killer felt believable.

The first version of our new web based IntelligentPad platform looked amazing in comparison of previous desktop version, though of course lacking in number of Pads, but that was just a matter of time and effort to us. With Silverlight 2.0 came C# and VB. JavaScript was pushed back and though the underlying use of it still existed, C# or VB were the languages to use with the Silverlight Plug-in. And naturally followed the king of all IDE, Microsoft Visual Studio. This was of course extremely beneficial and welcomed to us in the lab, but we would later realize that it came with a high price, the lack of access to this tool base by ordinary home users. With Silverlight 2 Webble World 0.9 was developed and with SL3 and SL4 everything went from clarity to clarity and the strengths of the plug-in just kept getting stronger. Silverlight 5 was incorporated nicely and Webble World 1.5 was now online since early 2012. During the later part of 2012 Webble World went through another re-factoring of somewhat major impact within several fields of the platform and this is where we are today with Webble World 2.0

## 2.2 What Is Different

Though there are many sharing communities, blogs and interactive web sites available today on Internet with sometimes quite satisfying possibilities for at least ordinary communicating participation, most web sites are rather static, being completely non-reusable or editable by visitors and also one-directional, treating the visitor mainly as a silent consumer. The interactivity that do exist are either too complicated for ordinary Internet users to use or very non generic, only allowing one or two things, like adding comments, in a much closed environment.

A click-able button or a forum does not make the web user a true participant or collaborator. We need the power to take what exists from several scattered sources and recombine, add, remove and reedit and then finally republish some of them back to the web so that other users can process them even further. So the task is to make the web fully re-editable for every Internet visitor so that its contents, including both static and functional ones, can constantly evolve. There is only one main goal; to share the collected minds of us all.

The base of the requirements is that these meme media object have to be available online and located from an arbitrary browser running on an arbitrary OS, and there, in the browser they should be fully functional, editable and re- distributional. The object should also be fairly easy to develop, preferably in a well-known programming language and have extended and rich graphical possibilities which should be easily implemented by either programmers or designers. Finally these objects should not be bound by commercial products or demands, but instead mainly rest on open source, or at least freeware, foundation.

Today we are, if not entirely, still very close to that vision. Therefore the limitations of the Webble, the Webble widget and the Webble application lies more or less completely within the limitations of the imagination of the human resources, except within the fields we will explain in the next section. The environment is free and open and the pieces comes in all shapes and forms and can even be reconfigured into even more still. This 'meme Lego' which even takes the original concept of IntelligentPad even further, these Webbles, may be the next stepping stone toward the true idea of a shared mind.

## 2.3 Obstacles and Problems

This might sound like a system designers dream, not to mention the dream of numerous creative Internet users who today lacks appropriate tools for their creativity. But in practice there have been many obstacles and problems, even in the end, regular showstoppers (missing essential features or embedded restrictions). It has gone so far that Webble World 3.0 which has just begun being developed will not use Silverlight anymore.

Every implementation of IntelligentPad has carried with them various issues that hindered or damaged parts of the promised experience or possibilities. Though we will not explain much further what those issues were before Webble World (like for example difficulties ranging from shape restrictions, problems of pad distributions across users, copyrights issues, development restrictions to preserve generic usage and the one still lingering, sandbox and cross-domain restrictions), we would like to involve you more thoroughly in the problems with this latest version.

Silverlight, like every other common web technology, is heavily sand-boxed, maybe even more so than most. The reason for this is of course security and to keep malicious websites from being able to do nasty things to the user's computer, but the result is also that a lot of cool and important ways to connect servers and tools with each other gets severely limited. There are solutions and hacks to get around it, but they do not always suffice and they are never pretty.

Silverlight does not work well with HTML either. Every attempt to blend them in a website is riddled with problems and ugly hacks that in the end is not even close to what the developer intended to do. Pure Silverlight sights, or small independent islands of Silverlight inside a HTML page works kind of okay, but anything beyond that is a nightmare, which of course is extremely problematic for Webble World which tries to connect all sort of web sites and web solutions in a seamless way.

Next issue on the list is the fact that plug-ins in general are not supported by mobile devices. A trend that first began with Apples iPhone and iPad and has now been adopted by both Google's and Android Microsoft's Windows Phone. This means that Webble World is completely shut out from these devices. Back in 2007 when development began and Silverlight was new, there basically were no smart phones, so it was hard to foresee this problem, but now it is obvious that it is there and it is not going away. It actually means that plug-ins in general will probably fade away as a valid use of web solution since the mobile market is basically dominating the sales nowadays.

Last but definitely not least is the strong feeling and persistent rumors that Microsoft will turn their back to their own creation for greener pastures, namely HTML5, and in the future only use Silverlight as framework for creating user interfaces for their Windows OS technologies. (Both Windows 8 and Windows-phone use Silverlight for their interfaces, but that is completely disconnected from the web browser plug-in solution).

So what conclusion does this daunting list of problems emerge to? Only the most obvious one, That the next generation of Webble World will not use Silverlight as its technology base. But the current do, and it does so extremely well, both on Mac OS and on Windows inside most common browsers.

Webble World, as of today, early 2013, rests on Silverlight, but the concepts, ideas and solutions of this system and the theories behind regarding meme media does not. One can fully enjoy Webble World already today and create multiple forms of rich Internet applications, and when the day comes and the Silverlight plug in is no longer required, those same Webbles will in most cases load as good as ever, and in some rare cases still work fine if Silverlight is available on the machine running this new version.

Webble World may come in many shapes and forms until the end of time, and this platform will evolve and mutate alongside the Webbles it is hosting, but still they will be Webbles and as Webbles they will prevail.

## 3 Webble World

Finally we have arrived to what matters, the actual platform environment Webble World and its inhabitants, the Webbles. It is time to understand their greatness and how to utilize and harvest their fruits both as a hard core programmer and primitive Webble developer but even more important as an ordinary non-programming web user that just have a need or an idea waiting to be realized.

### 3.1 A Webble in Theory

The concept of the Webble is simple enough where keywords like availability, customizable, editable, redistribution, combinable and configurable are supposed to describe its existence. A Webble can, depending on its developers intentions carry any kind of digital data, and support any kind of operation and behavior programmable today. It can be stored anywhere on the Internet and then reloaded from most browsers anywhere. Webbles can, no matter of its intended purpose be customized in many ways by users and combined with other Webbles in order to create compound Webbles or Webble applications, where the primitive building blocks together form a complex tool. A single Webble or a set of many different ones can be used to build simple websites holding some plain text, images or other media, or they can be constructed into interactive tools and complex applications like virtual labs, media content editors, games and much more.

Webble is the short name of the acronym Web-Pebble, which in turn means 'Pad Enhanced Building Block Lifelike Entity' (on the World Wide Web). It aims to tell you that a Webble is a digital object which can be loaded into a web browser (and in a future version maybe also the desktop), an object that contains Enhanced IntelligentPad qualities that make them like stand alone building blocks, that are often powerful and useful on their own, but when combined with other Webbles become even more useful. They are building blocks because they can fit together with any other Webble no matter what purpose or design that Webble have. And not only do they fit together, they also have the power to communicate with other Webbles and exchange data in order to make the data evolve inside the Webbles in order to solve a specific task. Some Webbles may even have very complex functions as AI, physics, 3D or real life simulations.

The Webble is, as explained before, constructed based on the IntelligentPad system with some enhancements, which in reality means, that a Webble is structured a certain way, has a wrapping or coating that encapsulate the internal unique code and implementation of a specific object, so that it will think of itself as a Webble and be able to do Webble things.

In details, the current Webble adopts a simplification of the MVC (Model-View-Controller) concept which means that a Webble is divided in two parts, a Model and a combined View/Controller part, from here on referred to as the View (also known as the display object).

The Model is considered to handle all internal matters that do not require any external interface, also known as the business logic while the View deals mainly with all interaction with the user and holds all visual parts of a Webble. Any Model can be combined with any View, all depending on the task the Webble is being designed to fulfill. One good reason for this separation is the possibility for new Webbles with different views to easily inherit Model behaviors from an old Webble.

Furthermore, within the core of Webble and IntelligentPad design one will find the concept of slots, each which is defined inside the Model and the View. A Slot is an externally available property parameter or method controller whose values may be viewed, exchanged, communicated and modified between present Webbles and also by users. The name slot tells us that we can see it as a hole or a plug where one may connect a contact in order to create a stream channel or path between two slots in two separate Webbles. This channel can be configured as a unilateral or a bilateral one.

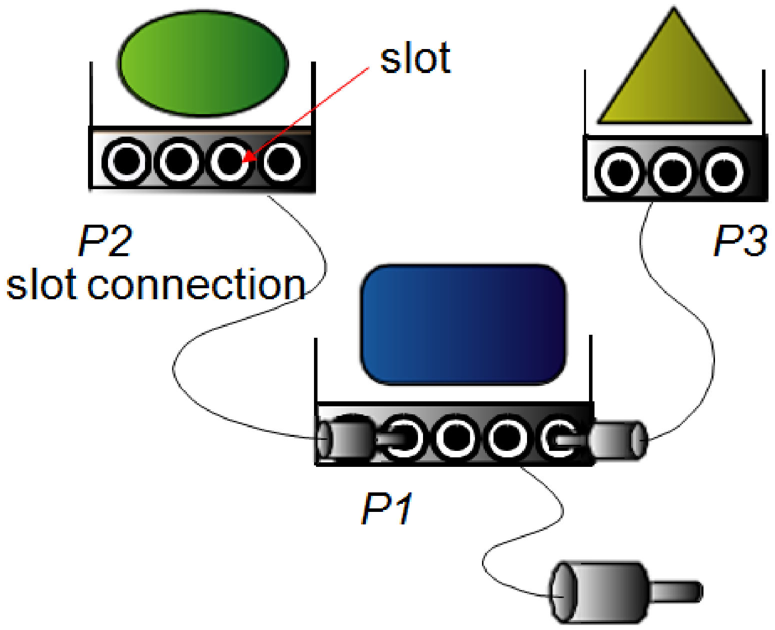


**Fig. 1.** Slot connections for property value sharing between meme media objects

Slot values can in theory be of any type, especially true if the Webble World System is built upon a non-typed language like for example JavaScript but even for a type-dependant language like the current used C#, the idea of a Webble slot is to always be able to handle all forms of data that is thrown at it. In realty though when dealing with strictly typed languages the support will only cover a set of the most common and used ones but that will in most cases suffice. Independent stand alone Webbles can of course contain Slots of any type imaginable, but weather other Webbles understand that type correctly if it is out of the ordinary is another question. If a slot is given data of a type it does not understand it will treat it as text in external communication and ignore it internally. No data type will ever make a Webble choke. Even in a system where slots do not require to be type-dependant it may sometimes be necessary for a Webble to know the data before it can do something constructive with it, so Webble World built on top of any environment will always try to keep a library of typical behaviors for certain forms of data. Is it an number or a string? Sometimes it does not matter, but sometimes it does and the task for a Webble slots is to know when and act accordingly. In the current version there are numerous known types available, from the classical numerals and strings, to more detailed ones as colors and points and also complex types as xml documents and object dictionaries.

Slots can also be bounded to methods where the value of the slot can serve as method parameters. Slots can also be generated and bound to attributes of the visual objects in order to directly control the appearance of a Webble via the slot.

The internal slot values of a Webble are in most cases configurable by the user, unless the creator of the primitive Webble have made any value non-editable for some reason, which can be that either the value is used as a trigger and therefore should only be set as a result of slot communication between other Webbles, or the value is of a very complex type and therefore are either set internally or by a custom made configuration tool provided with the Webble itself. But besides those rare cases Webble slots can all be configured via platform provided configuration tools or forms and have their values changed by the user. There are no limit to how many slots a specific Webble may have or, as explained, of which type its values can be.

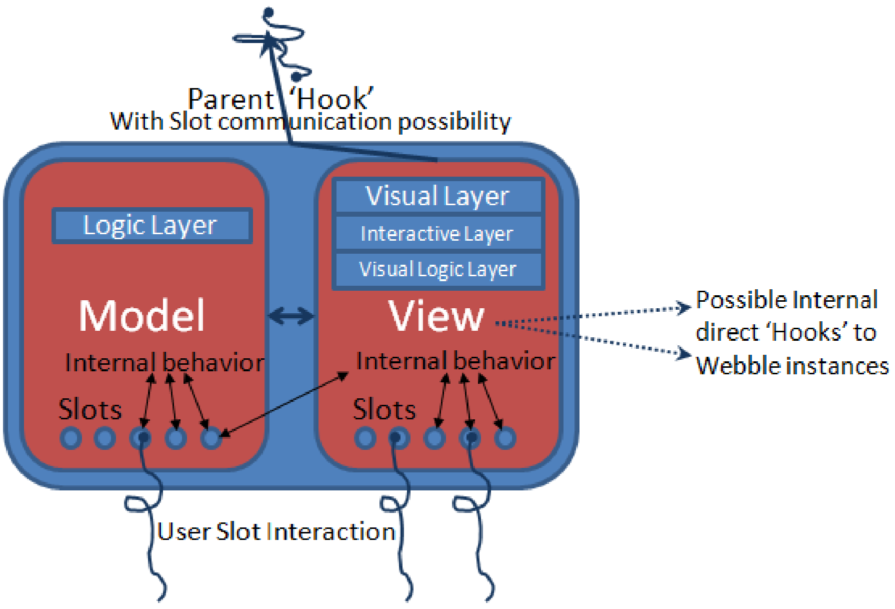


**Fig. 2.** The inside of a Webble and its different parts needed to enable the right to be called a Webble

Slot communication between Webbles requires that the Webbles are closely related, in order to make them aware of each other's existence. This is achieved by structuring Webbles in hierarchical parent/child relationships. Webbles can have any number of children, but every child can only have one parent. The reason for this is to make the hierarchal overview very simple to follow and understand for human users and developers.

A Webble may then create slot connections with its parent (single) and its children (multiple), and those parent and children may in turn connect further in the relational chain. The slot communication is handled by the child in the relation, meaning that it is the child who knows which slot in the child and which slot in the parent that are supposed to share values and also in which directions this communication should be flowing. In most cases it is the task of the user to choose which slot in the parent and which slot in the child should communicate with each other, but there are cases where advanced Webbles have the ability to handle such connecting on its own.

As soon as such a communication channel is configured, the Webble will handle the communication on its own by three ways of control methods. Whenever the parent have any value change in any slot it will fire the 'Update' message, informing every child that something have changed, it is then the task of the child to use the 'Gimme' message in order to retrieve the value of any specific slot from the parent to see if the update concerns them and the value of the slot they care for has changed. After that, it is then within the scope of the child to react upon the value collected. If instead it is the child slot that is altered it will transfer that alteration over to the parent with the 'Set' message to the connected parent slot, which in turn may make the parent react on the value change. Internally, both in the View and the Model, all slot changes fire a slot reaction method which sometimes does nothing and in other cases do a lot, maybe even start changing other slot values. In 'normal' slot communication a Webble can only have one slot channel open with each relative (parent or children) at one time. But there are of course ways to set up multiple channels by using for example some specialized Webbles that were built to handle that. A similar structure of communication is going on between the Model and the View within the Webble itself.

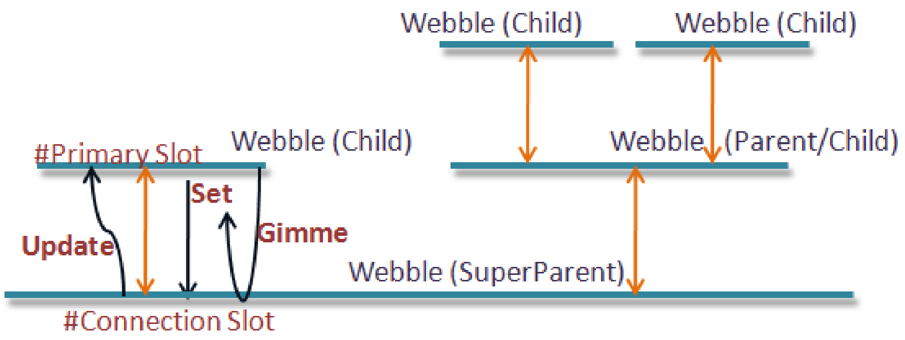
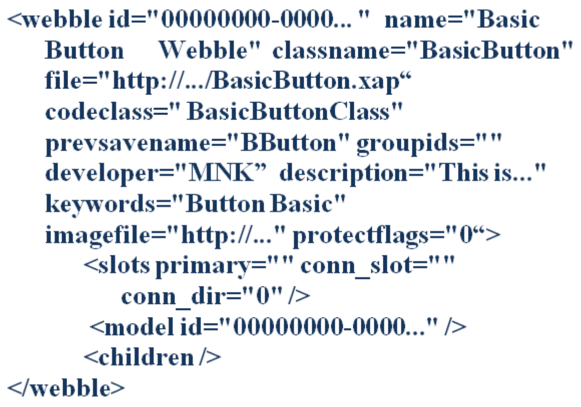


**Fig. 3.** Webble structure hierarchy and slot communication

A Webble can be duplicated freely, either as a separate entity or by sharing the Model with its original. In the latter case, a duplicate is called a shared copy. This is another form of internal communication between Webbles, which does not require any parent/child relationship, though it only affects the Model part of the Webble and then not just slot values but the Model as a whole.

The Model works on its own and does not care about the View, but it is aware of the Views existence in order to call the Update method for all Views connected. The View on the other hand can take the model into as much consideration as it wants, since it has complete access to its reference, this is configured mainly by the code developer of the Webble and does not concern ordinary Internet users.

A Webble is not only defined as described above with programming code. Actually an even more important part of the Webble is the Webble definition or configuration file which describes each part in each particular individual Webble. Such a Webble definition is described in XML and hosts all internal values and properties of a particular Webble, like slot values, children, connections, model as well as telling any Webble Platform where to find the Webble code package and much more depending on Webble class. It is this XML file that separates primitive Webbles (code generated only) from compound ones.



```
<webble id="00000000-0000... "  name="Basic
    Button    Webble" classname="BasicButton"
    file="http://.../BasicButton.xap“
    codeclass=" BasicButtonClass"
    prevsavename="BButton" groupids=""
    developer="MNK”  description="This is..."
    keywords="Button Basic"
    imagefile="http://..." protectflags="0“>
        <slots primary="" conn_slot=""
            conn_dir="0" />
         <model id="00000000-0000..." />
        <children />
</webble>
```

**Fig. 4.** A Webble definition XML file in its simplest form

This XML is completely independent from technology used to implement the platform. The capability of a Webble to read and write XML that describes its inner states and relationships is another crucial part of the Webble concept. One that makes it possible to share Webbles across platforms and implementations. These XML files can be stored anywhere online and will be reachable from any server as long as the server containing the Webble Definition files is configured properly with Cross Domain settings. Internet Domains are often limited and sandboxed, not allowing two domains to communicate. This can be regulated with Cross Domain files and safe domain settings which Webbles exploit to the limit. The current Webble World platform uses a SOAP Web service in order to save Webble XML definition files on online servers and make them available for others to find and to download, but future Webble platforms may use completely other techniques in order to store and find available Webbles. As long as they are available from anywhere in any browser the technical solution matters very little.

So what constitutes a Webble is an independent self reliable digital object that can describe its inner states via parameters called slots which is also its way of communicating externally both with other Webbles but also the user and the platform, and which forms child parent relationships with other Webbles in order to enable above mentioned communication. That's it. Because of these simple characteristics Webbles can be joined together in order to solve all forms of tasks in various ways and fashions.

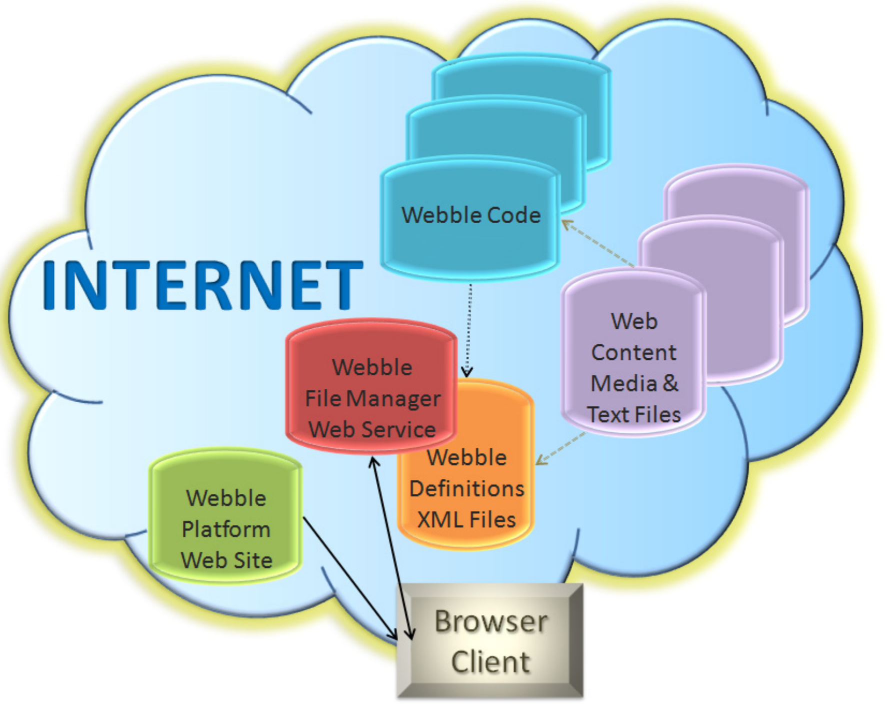

INTERNET
Webble Code
Web Content Media & Text Files
Webble File Manager Web Service
Webble Definitions XML Files
Webble Platform Web Site
Browser Client

**Fig. 5.** The relation between Webble parts, the platform and data content as found on Internet

## 3.2 Creating a Webble

There are two levels of Webble construction. Let's look at them both in turn.

The first one requires programming skills and is therefore aimed towards programmers. This is conducted outside the Webble World environment on a local machine using a basic editor or common IDE which can load the Webble template code prepared by the Webble World team. This is the creation of primitive Webbles, so called because they are generated from code only and does not require the previous mentioned configuration XML file. The goal of such developer is to wrap a certain feature or content inside the Webble shell and make its parts available and editable through slots. This is how every single Webble starts out and though the programming language may differ, depending on which platform they are created to run on, the idea is the same, they are described in code alone. A Webble developer coding a primitive Webble can of course extend, and in some cases even alter, these default behaviors. A Webble can for example search for Webbles on its own internally and create relationships without the user interaction and also be able to interact with related Webbles in a much more complex way than what is possible in a default user interaction situation. Webbles are structured a certain way for interconnectivity and easy maintenance but they are no way restricted from, alongside the basic Webble pattern, implementing other software patterns and complex internal solutions or sophisticated means of communications beyond the default possibilities. That is of course one of the major strengths of the Webble and meme media object theories, that though recognizable in structure and design as well as human interaction interfaces, they are never limited to these only. A meme media object can look, feel and behave any possible way, if it can be done with code at all, it can be done by a Webble, only limited by the imagination of the developer, but it will at least always be what we expect it to be; a meme media object; a Webble.

The second approach requires no programming and is therefore aimed to the general public. This is happening inside the Webble World environment, in the browser online and is the manipulation and configuration of available primitive Webble's and/or previously constructed Webble objects and applications. This is simply done by loading wanted, appropriate Webbles into the platform and begin manipulate slot-property values in combination with forming Webble relationships and enabling slot communications, all by using simple mouse operations and straight forward keyboard operations. The result is stored as XML but to the user, who do not care how the creation is saved, this is all under the hood. By creating parent/child relationships between Webbles, adding slot value communications between them, and using Webbles with certain features, one can form a vast range of thinkable behavior in a Webble gadget or application. If a specific feature is missing for a certain task, it is easy enough to create a primitive Webble in code and publish it as your own generic Webble and use it in your compilation, but in most future cases the probable vast available number of primitive Webbles will allow most users to create gadgets and application without writing a single line of code.

In both described cases of development the goal is always to make the creations available for everyone and anyone to use and enjoy.

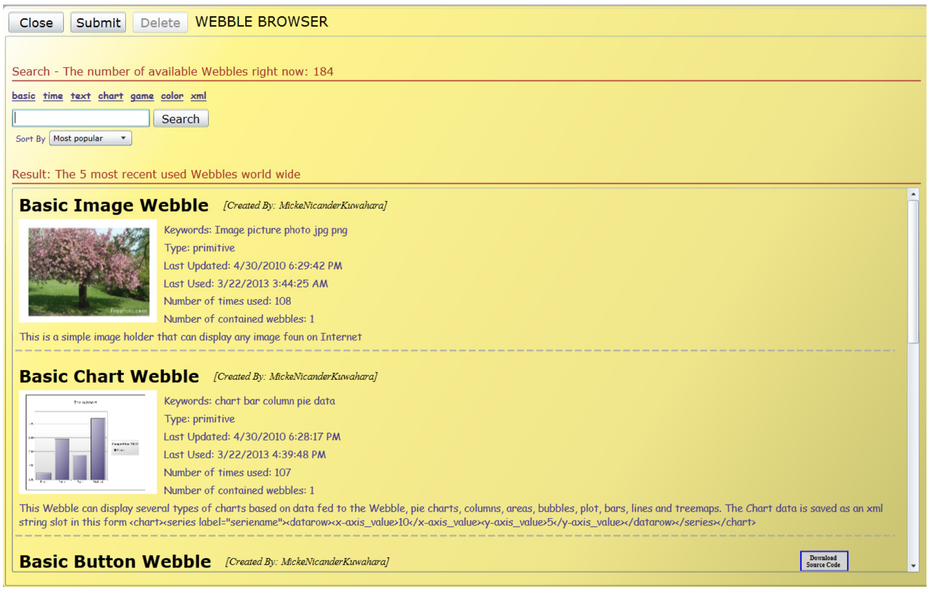
## 3.3 Webble Development in Practice

We will not describe in any further details what it means to code a primitive Webble since it is just simply classic programming using the Webble API provided for download via the Webble World website and to follow the guidelines thoroughly explained in the included developers manual on how to create models, views, connect them and finally publish to the online repository.

What we will take a much closer look at though is how we in practice may create something useful and appealing directly in the browser without writing code by just using the building blocks available.

In reality the first thing a Webble user must do is to visit a Webble World Platform Web site, where many options are available depending on the users need and intentions. All published and properly registered Webbles can be detected by the Platform via a Webble management Web service, which keeps track of the XML definition files available and when requested by a user, by using provided platform Webble tools, like a search engine for Webbles, that allows the user to search, filter and sort Webbles in order to find the Webbles that fits the users need, may be downloaded.

As mentioned above, all data regarding description, and use and ownership of a Webble is stored inside the Webbles XML definition file. When a user submit a load



Search - The number of available Webbles right now: 184

Result: The 5 most recent used Webbles world wide

**Basic Image Webble** *[Created By: MickeNicanderKuwahara]*

Keywords: Image picture photo jpg png
Type: primitive
Last Updated: 4/30/2010 6:29:42 PM
Last Used: 3/22/2013 3:44:25 AM
Number of times used: 108
Number of contained webbles: 1

This is a simple image holder that can display any image foun on Internet

**Basic Chart Webble** *[Created By: MickeNicanderKuwahara]*

Keywords: chart bar column pie data
Type: primitive
Last Updated: 4/30/2010 6:28:17 PM
Last Used: 3/22/2013 4:39:48 PM
Number of times used: 107
Number of contained webbles: 1

This Webble can display several types of charts based on data fed to the Webble, pie charts, columns, areas, bubbles, plot, bars, lines and treemaps. The Chart data is saved as an xml string slot in this form <chart><series label="seriename"><datarow><x-axis_value>10</x-axis_value><y-axis_value>5</y-axis_value></datarow></series></chart>

**Basic Button Webble** *[Created By: MickeNicanderKuwahara]*

**Fig. 6.** How searching for Webbles looks to the Webble World visitor

Webble request, the web service will return the XML for the Webble in question and the Platform will then within its meme media structure framework, by reading the XML, locate the needed code files, which the platform will download and load into memory before it calls the initiation method within the Webble downloaded.

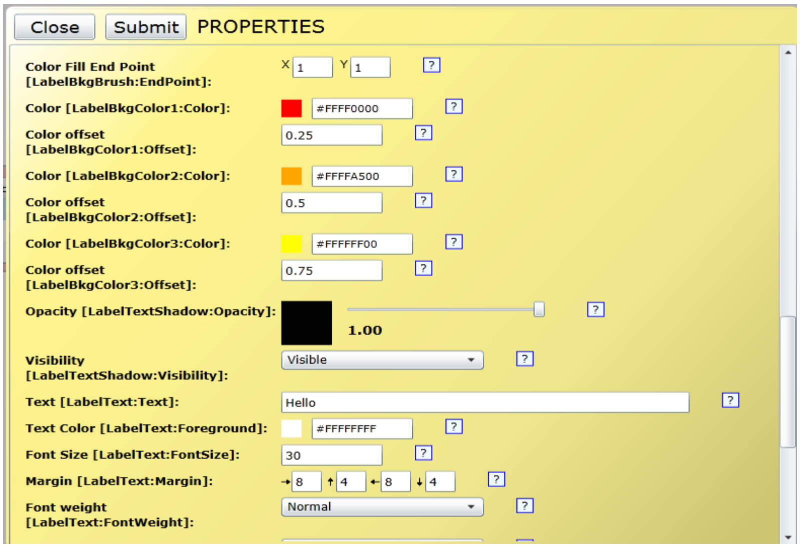
The initiation call includes the XML as one of the parameters and the responsibility to setup and render the Webble is by that call more or less handed over to the Webble itself.

The work of the platform is merely to provide an environment, a pond, where all the Webbles can swim freely, and to keep track of these Webbles existence. Though it is not the case of the current version of Webble World it is likely that future versions and iterations will more and more strive towards making the main part of the platform itself into just more and specialized Webbles, where the framework will be slimmed down to a minimum of necessities, just enough to allow a Webble to load itself. Next generation of Webbles will then be solely responsible for its existence, and it is up to the Webble loaded weather more Webbles will be able to coexist with the first one, therefore the term "Platform Webble".

After the user have located the Webbles he wants and download them into the Webble World Framework inside the browser he or she can then start the manipulation and configuration of welding them together so they can perform the task in mind.

The configuration includes just a few default steps that applies to all Webbles, but every individual Webble may also include some additional configuration possibilities which applies only to that specific Webble.

The default steps includes first and foremost the editing of slot values, which of course may have many different effects depending on the slot. Some common slot changes is of course those that in some way alter the visual appearance of a Webble,



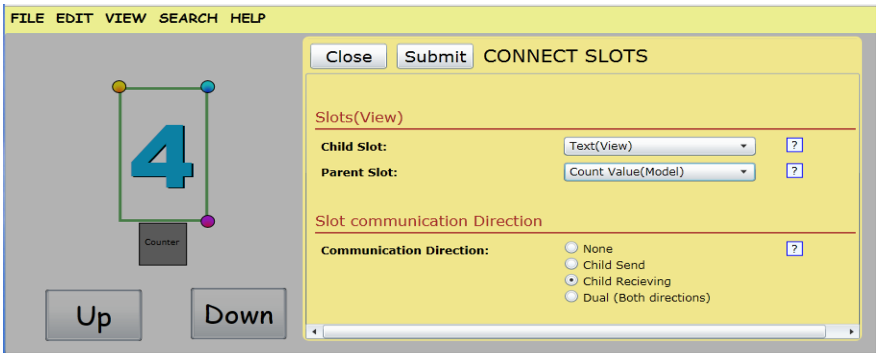
| | |
|---|---|
| Color Fill End Point [LabelBkgBrush:EndPoint]: | X 1 Y 1 |
| Color [LabelBkgColor1:Color]: | 🟥 #FFFF0000 |
| Color offset [LabelBkgColor1:Offset]: | 0.25 |
| Color [LabelBkgColor2:Color]: | 🟧 #FFFFA500 |
| Color offset [LabelBkgColor2:Offset]: | 0.5 |
| Color [LabelBkgColor3:Color]: | 🟨 #FFFFFF00 |
| Color offset [LabelBkgColor3:Offset]: | 0.75 |
| Opacity [LabelTextShadow:Opacity]: | 1.00 |
| Visibility [LabelTextShadow:Visibility]: | Visible |
| Text [LabelText:Text]: | Hello |
| Text Color [LabelText:Foreground]: | #FFFFFFFF |
| Font Size [LabelText:FontSize]: | 30 |
| Margin [LabelText:Margin]: | → 8 ↑ 4 ← 8 ↓ 4 |
| Font weight [LabelText:FontWeight]: | Normal |

**Fig. 7.** Slot property configuration of a text label Webble

but other internal logic slots may also be of importance when configuring a Webble for a specific use or task.

The second default configuration is that of pasting a Webble onto another to form a child-parent relationship. The action is simple, by after selecting the chosen Webble then pick the 'assign parent' option in one of the available interfaces or menus and then finish by selecting the target. The two Webbles will then be automatic related and it is most visually shown by moving the parent and see how all children are moving too.

The third and final default configuration is to form slot communication between related Webbles which in Webble world is done via a special tool provided by the Webble World platform.



Slots(View)

| | |
|---|---|
| Child Slot: | Text(View) |
| Parent Slot: | Count Value(Model) |

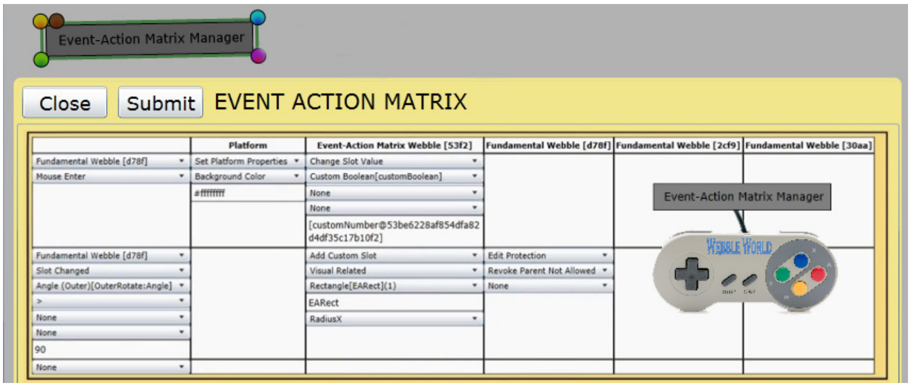
Slot communication Direction

Communication Direction: None [✓ Child Send] Child Recieving Dual (Both directions)

**Fig. 8.** Setting up slot communication between two related Webbles

There the user can select which slot in the child and which slot in the parent should exchange values and also in which direction or directions this communication should apply. When a slot connection is submitted the values will immediately synchronize with each other, and any effect that may have on the compound Webble will activate.

In addition to these configurations, specific Webbles my offer several more tools and input forms or interaction objects to even further configure the compound Webble which the user is creating. For example the Event-Action Manager Webble which can listen to events triggered by the platform, itself or, most useful, other Webbles and based on the result of these events activate certain actions, has its own interface form for setting up such event-actions.



| | Platform | Event-Action Matrix Webble [53f2] | Fundamental Webble [d78f] | Fundamental Webble [2cf9] | Fundamental Webble [30aa] |
|---|---|---|---|---|---|
| Fundamental Webble [d78f]<br>Mouse Enter | Set Platform Properties<br>Background Color<br>#ffffffff | Change Slot Value<br>Custom Boolean[customBoolean]<br>None<br>None<br>[customNumber@53be6228af854dfa82d4df35c17b10f2] | | | |
| Fundamental Webble [d78f]<br>Slot Changed<br>Angle (Outer)[OuterRotate:Angle]<br>><br>None<br>None<br>90 | | Add Custom Slot<br>Visual Related<br>Rectangle[EARect](1)<br>EARect<br>RadiusX | Edit Protection<br>Revoke Parent Not Allowed<br>None | | |
| None | | | | | |

**Fig. 9.** The specialized interface for the Event Action Manager Webble takes the form of a matrix

So in only a few minutes and with just a few Webbles, a simple gadget as an analog clock or a calculator may be created, but with more work and complexity even a whole RIA-style Webble application may be built that serves as a powerful tool or a game for the users that interact with it. When this compound Webble, or Webble application, has been finished, the Webble designer can save the creation, once again simply as an XML Webble definition file, either locally or preferably at any online Webble server. As soon as it is saved, it is now available for use by any Internet visitor who may use the Webble World platform's Search tool to find Webbles online, by name, description, developer or popularity.

Once loaded, the Webble will be working as the designer and the developer intended, and the user can do what the Webble promise to achieve. The users are always free to participate in further development of the Webble in use. It can be as simple as text editing, or as advanced as adding or removing features to a Webble cluster by customizing the present Webbles, by adding new ones, or by removing some of the already existing ones. This new evolved Webble is now easily resaved back online for further access and remakes. The original version remains unchanged and is available as well.

Webble World is one platform and one framework in one single portal but the number of interaction levels which the users and visitor may engage in are fully individual and are ranging from devoted hardcore developing and compound Webble designing to the plain everyday Internet user who simply visit Websites of certain interests, that contains information, multimedia or entertainment which the user enjoys to consume, which just happens to be made by Webbles.

## 3.4 Some Words on the Interface and Major Features

We do not wish to go into major detail on the actual interface of the current Webble World, partly because interfaces are constantly being improved and in new versions of the platform they may change entirely, but to just clarify to any possible Webble user what to expect in the current system we mention lightly some parts here.

Webble World has a main menu, either available at the top of the screen or by right click on the desktop, from there a user can reach all types of interactions and features the system provide. A short time experimenting should clear out most question marks regarding what is available. Worth mentioning though is that Loading a Webble from the menu requires the knowledge of the exact location (URI or local file system) but Search for Webbles needs only adequate search terms.

Each Webble loaded on top of the desktop have two major interfaces. First the Webble menu which contain all default Webble options but also Webble-class specifics. This menu can be reached by right click on the Webble or by selecting the Webble and therefore activating the second interface, the interaction balls, which is some small balls attached to the Webbles selection border and which contains some common operations like, resizing and parent assignment. These are the major ways to interact with the Webbles and the system.

Most of the features available is self explanatory or already described above, like for example slot value configuration, but some are not yet mentioned but still very much deserve to be acknowledged.

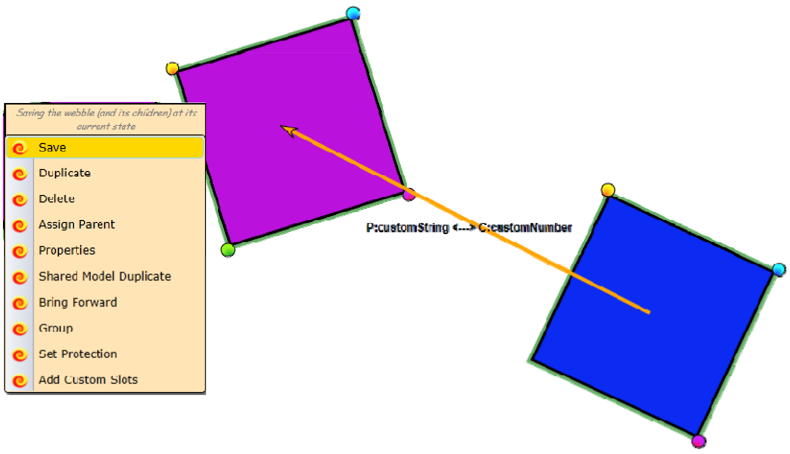
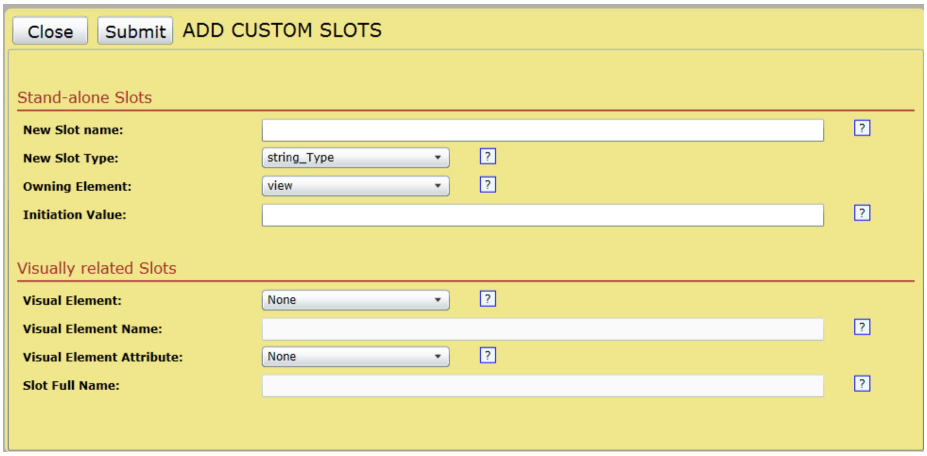


**Fig. 10.** The Webble and Webble World interfaces tries to be simple and highly accessible

On the platform system side there is for example the multi-Webble real time property manipulator that allows the user to change many Webbles slot values at the same time and instantly see the result. This has been found very useful by seasoned Webble developers. Under the View section is a feature that toggles the visibility of current slot connections in form of labeled arrows. When working with complex systems this is very useful in order to keep track of relationships and slot value flow. Finally is the registration and the login parts under the Help section which is nothing really to explain, but just be aware of, that it is required to be a registered user of the system before it is possible to save any Webbles on the online servers.

There are two Webble specific features that needs to be brought to attention; the first is the protection form that is used to limit or lock parts of a Webbles behavior in order to avoid unwanted effects of a certain Webble application, like for example blocking a Webble from being moved or duplicated. The second is the add custom slot feature. A very powerful feature that allows the user to add a wide range of new slots to any Webble, slots of many simple types as well as slots that are bound directly to previously non-exploited parts of the visual tree of the Webble, like for example colors, rounded corners, transformations and projections. These new slots are stored and behaves like any other default slot and therefore makes the need for creating new primitive Webbles in code much less needed.



ADD CUSTOM SLOTS

Stand-alone Slots

|  |  |
|---|---|
| New Slot name: | |
| New Slot Type: | string_Type |
| Owning Element: | view |
| Initiation Value: | |

Visually related Slots

|  |  |
|---|---|
| Visual Element: | None |
| Visual Element Name: | |
| Visual Element Attribute: | None |
| Slot Full Name: | |

**Fig. 11.** The Add Custom Slot interface of Webble World, simple and straightforward

## 3.5 Some Webble Examples

Which Webbles to use and what is available is one part of the challenge when developing a Webble app. The best way to get an overview of, at least the availability, is to learn how to use the Webble search tool inside the Webble World platform. At the time of this writing there are almost 200 Webbles available on meme media laboratory's online server (and probably around double that spread globally in various internal networks). Of those 200, about 25% is of the primitive kind, the simplest building block, while the rest is compound constructions and applications. A new user should definitely

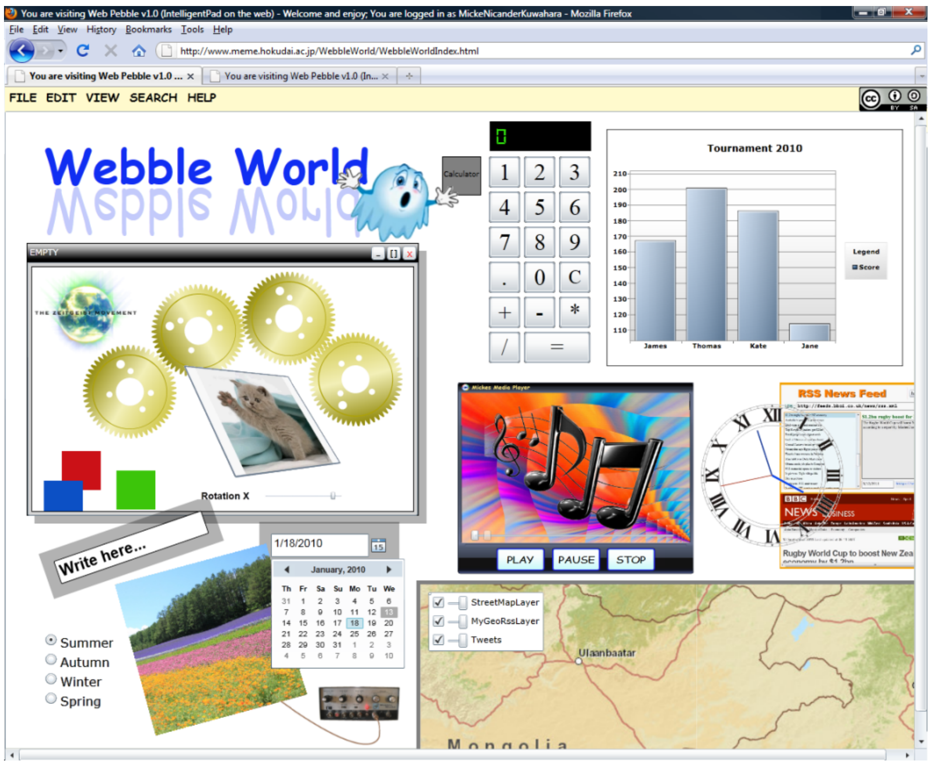
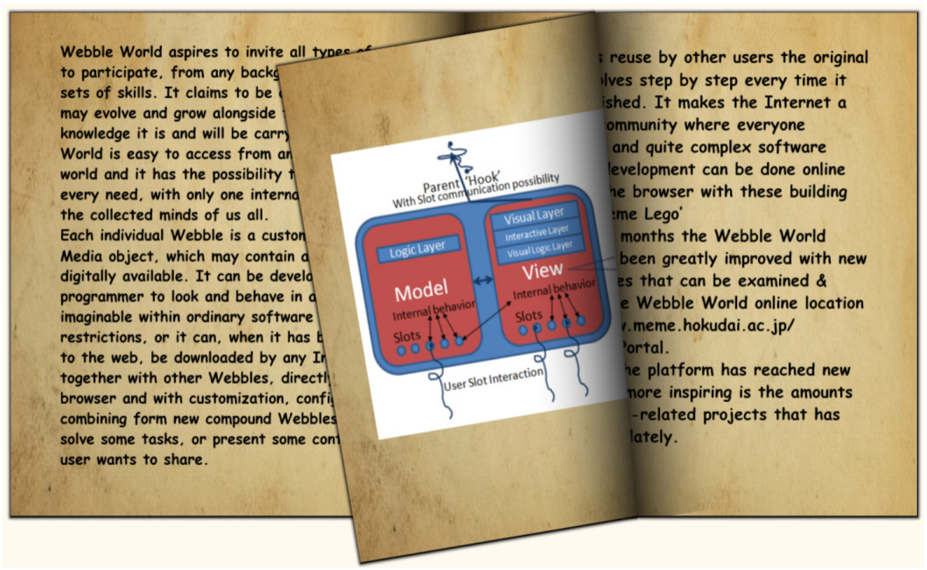

You are visiting Web Pebble v1.0 (IntelligentPad on the web) - Welcome and enjoy; You are logged in as MickeNicanderKuwahara - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.meme.hokudai.ac.jp/WebbleWorld/WebbleWorldIndex.html
FILE EDIT VIEW SEARCH HELP
Webble World
Calculator
EMPTY
Rotation X
Write here...
Summer
Autumn
Winter
Spring
1/18/2010
January, 2010
Tournament 2010
Legend
Score
James
Thomas
Kate
Jane
Mickes Media Player
PLAY
PAUSE
STOP
RSS News Feed
StreetMapLayer
MyGeoRssLayer
Tweets
Ulaanbaatar
Mongolia

Fig. 12. Just a minor set of Webbles available in Webble World

begin the Webble World exploration by free search and download of any Webble that seems interesting followed by examining these Webbles a bit closely, like their Slots and how they interact with other Webbles connected. After some time doing that the understanding of Webble World and its possibilities will be much stronger.

Though, in order to give you a hint of what's out there, we will shortly present a few common primitive Webbles and applications here and now.

Besides the basic form of Webbles that do the most rudimentary tasks like displaying images, display texts, text input, buttons, time and date trackers etc, one of the most useful primitive Webble is the basic window Webble. It looks like a classic window in its default shape but when its slot property form is opened it is obvious to the user that thus is mere the beginning of this Webble. The basic concept of it is to be a Webble child container. All children to this Webble gets wrapped inside no matter how the window is displayed or behave. There are both a long list of behavioral altering slots as well as appearance tweaking slots the user can work with. Behaviors ranging from duplicating children clicked or catching any Webbles dropped on top of it as a new child. These behaviors are extremely useful when creating toolboxes or work areas, where the children Webbles work as specialized building blocks of some sort. The appearances ranges from classic window and speech balloons to tab fields and even as a virtual book.



Webble World aspires to invite all types
to participate, from any backg
sets of skills. It claims to be
may evolve and grow alongside
knowledge it is and will be carry
World is easy to access from an
world and it has the possibility t
every need, with only one interna
the collected minds of us all.
Each individual Webble is a custon
Media object, which may contain a
digitally available. It can be develo
programmer to look and behave in a
imaginable within ordinary software
restrictions, or it can, when it has b
to the web, be downloaded by any In
together with other Webbles, directl
browser and with customization, confi
combining form new compound Webbles
solve some tasks, or present some con
user wants to share.

; reuse by other users the original
lves step by step every time it
ished. It makes the Internet a
ommunity where everyone
and quite complex software
levelopment can be done online
he browser with these building
me Lego'
months the Webble World
been greatly improved with new
es that can be examined &
e Webble World online location
.meme.hokudai.ac.jp/
ortal.
he platform has reached new
nore inspiring is the amounts
-related projects that has
lately.

**Fig. 13.** The virtual book setting of the basic Window Webble enabled

Of the more complex Webbles like the chart Webble that display data in numerous eye-pleasing ways, the interaction tweaker Webble that manipulates other Webbles interfaces, The map Webble that displays and interacts with Bink maps, the web browser Webble that can display HTML content and the select list Webble that can display various forms of lists and list related interfaces, one of the most interesting one is the XAML customization Webble. XAML is as mentioned before the vector graphic implementation of Silverlight which is the engine in the current Webble World, but one can imagine similar Webbles in any other Webble World implementation, using for example SVG or HTML with CSS. The concept of this Webble is to allow the user to paste (or write if the user is familiar with the scripting language of XAML) any piece of XAML into the appointed slot and by that alter the Webbles appearance accordingly. This Webble is extremely useful and powerful and even further limits the need to create new primitive Webbles.

The list of cool and powerful applications grow constantly, like the Trial Outline Builder for cancer trial research mentioned before or virtual lab environments for e-learning settings developed by Fraunhofer Institute in Germany or the Data Analyzer Dashboard currently under development at Meme media Laboratory.

Though some of these are not available to the public there have also been some Webble application development lately which tries to appeal directly to the main stream audience, even outside the academic world, mainly some games. One of these games recently published is a remake of the classic ball and paddle game Arkanoid which can be played by any Webble World visitor. Besides enjoying playing the game, any curious user can load it into the Webble World desktop in order to take it apart and examine how it was made. Many more game related demos and mini apps are available which, if nothing else, can be very useful study objects in order to better understand the potential of what Webbles can do.

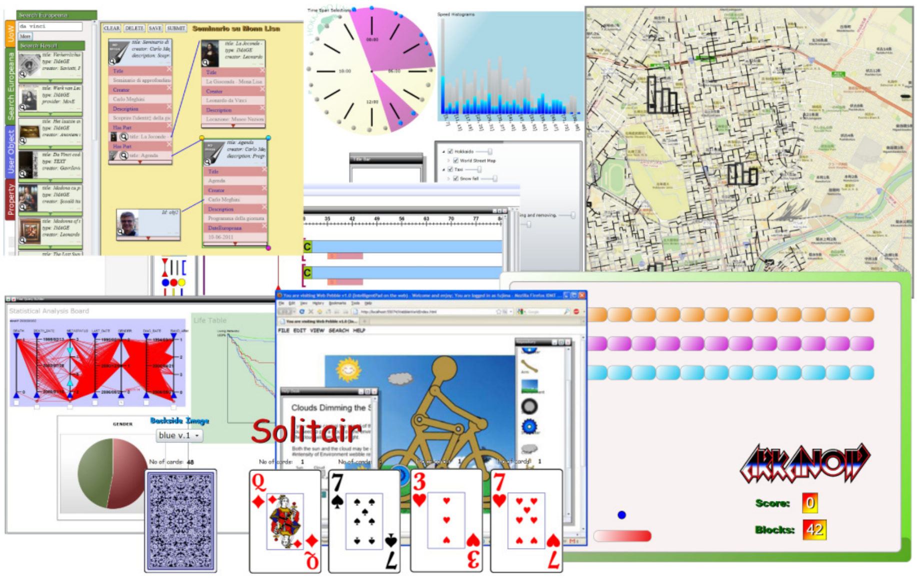


**Fig. 14.** A small subset of Webble applications crammed together in one image

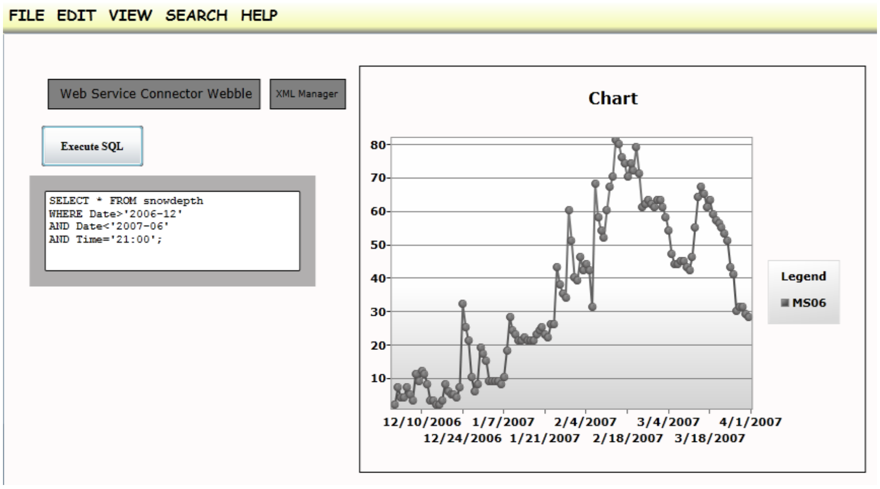
## 3.6 A Case Study

In order to finally clarify any lingering questions, we wish to describe a simple case study of a possible Webble development situation.

Imagine that a person have access to a database with some data and on top of that a web service that can retrieve data from that database via SQL queries and return the result as an XML. This person now wish to access this database, query it and then display its result in a graph and he (or she) have decided to use Webbles for this task. This is our suggestion on how to go about in doing so.

The Web Service connector Webble seems like a good start. It is loaded into Webble World and the slot property form is opened. One slot wants the URI of the service, this is given and saved. The Webble then automatic contact the service and gets in return a list of available service methods, also available as a slot. The user selects the one wanted which results in another set of slots that appears which represents the needed parameters required by the service. In this case a database name a password and the SQL query. The user fills in the first 2 and wait with the query for later and instead save and close the slot property form. Next the user wants to improve the user interface of his creating, for this is needed one text input Webble and a button Webble. The Text input Webble is also configured slightly to look better for its use and then by right clicking it and selecting assign parent, followed by a click on the web service Webble create a child parent relationship. Next the user opens the slot connection form in the text input Webble and connect the text slot in the child with the SQL parameter slot in the parent. Now the user can feed queries more easily to the service. Then the user attach the button Webble as a child to the web service Webble and configure it appropriately before connecting the buttons click slot to the web service Webble's service call activation slot, in order to easily fire new web service calls. If an SQL query was typed into the text box and the button was clicked, the web service Webble would then do its thing and the response from the service call would end up in the result slot of the Webble. But since the response would be an XML which is not the requested end result there are still some work to do. When loading the Chart Webble and investigating the slots available and the additional chart Webble interface the user realize that even though the chart Webble is happy to receive data in form of XML, this XML needs to be formatted a certain way, which is not the way the service will respond. So what is needed is an XML converter Webble which luckily seem to exist in the Webble repository. It seems such a Webble have been requested and used before. After the XML manager Webble has been loaded and connected as a child to the web service Webble and its input XML slot has been connected to the parents Service response slot the user needs to define to the XML manager the way the XML from the service should transform to become the format requested by the chart Webble. This is done in the XML template slot which also contains instructions on how to do so. The user types in what is needed and saves and close the slot property form. Finally the chart Webble is connected as a child to the XML manager Webble and the data slot in the chart Webble is connected to the output XML slot in the parenting XML manager Webble. After saving the construction on the local machine for safe keeping the small Webble app is tested. A query is typed in and the button is clicked and a second later a chart is nicely drawn and displayed on the screen. This little app is now ready to be saved online.

One can imagine more Webbles in this construction, for example a SQL Query generator Webble that removes the need to understand SQL syntax, or a table Webble to display the data in additional forms, but for this simple example we feel we are done.



```
SELECT * FROM snowdepth
WHERE Date>'2006-12'
AND Date<'2007-06'
AND Time='21:00';
```

**Fig. 15.** The final result of our little case study app

## 4 Future Possibilities

What follows is the next iteration of Webble World in order to mainly solve those issues mentioned in section 2 with device compatibility. The new version will be developed in HTML5, CSS3 and JavaScript, the later supported by some carefully selected libraries like the newly Google developed MVC structured AngularJS and the very helpful and popular JQuery. The HTML5 package has a very wide support on both mobile devices as well as laptops and desktop computers and it seems that from most angles this solution will solve all our current needs.

The plan is to have a new running version of Webble World by the end of this year, 2013. The goal also include to make the transition from the Silverlight version to the HTML5 version as smooth as possible, both by being able to read the same XML configuration files as the current Webble World version but also by including a new Webble that can wrap Silverlight content when Silverlight is available.

### 4.1 New Features and Old Concerns

During this new task of reinventing Webble World a revisit to the ever lingering problem of simplicity vs. complexity will be made. The concept is easily explained, it should be very simple to work with Webble World in order to build very complex things. But to realize that is a completely different matter. IntelligentPad and Webble World users have discovered for years that it is hard to make complex systems when the tool only allows simple connections. Usually this has been solved by creating extremely complex Pads or Webbles that under the hood does what is wanted, but the question is if that is the right way to always go. How much damaging would it be to the structure and design if Webbles could communicate freely with any Webble beyond children and parents, and what if every Webble could create as many slot connections it wanted. Would it not be possible to maintain clarity and viewability by a higher level of platform design instead, that helped the user to see the paths and flows of connections both within and between Webbles. If these questions will be answered we cannot know, but the questions will be asked.

Another field of research that should be addressed is the creation of a Webble ontology. Today Webbles are named and described freely without any second thought and that is obvious to anyone not a good way to go in a world where Webbles are shared and distributed all over, cross systems and platforms. How this will be solved is not cleared, but surveys and input is being collected to find a solution that secure the longevity of the Webbles. What also follows within the same realm is versioning of Webbles. Since Webbles are reloaded and resaved during its entire lifetime there must be a system in place that makes sure users have access to the latest version of a Webble, but with the choice to continue using the last unmodified one.

Exactly what new features comes with the new version is not all clear yet, but the new version is on its way and you will know it by the name Webble World 3.0.

## 5 Summary

By now you should have a very good understanding of what Webble World and its inhabitants the Webbles are. How to find them and how to use them and more importantly how to build them and create them. You are fully aware of the greatness and potential of meme media objects, like Webbles, but also informed of the problems the concept face. You understand the technical reasons why the current version of Webble World is not the final step and the reasoning behind the decisions to take the next step in a certain direction. You should be eager to dig deeper in the practical use of Webbles and become a Webble World contributor yourself but also look forward in eager anticipation to the next version arriving to your browser within a year.

## References

1. Tanaka, Y., Imataki, T.: IntelligentPad: A Hypermedia System allowing Functional Composition of Active Media Objects through Direct Manipulations. In: Proceedings of the IFIP 11th World Computer Congress, San Francisco, USA, pp. 541–546 (1989)
2. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)
3. Kuwahara, M., Tanaka, Y.: Webble World, a web-based knowledge federation framework for programmable and customizable meme media objects. In: IET International Conf. on Frontier Computing, vol. CP568, pp. 372–377 (August 2010)
4. Tanaka, Y.: Knowledge Federation over the Web Based on Meme Media Technologies. In: Jantke, K.P., Lunzer, A., Spyratos, N., Tanaka, Y. (eds.) Federation over the Web. LNCS (LNAI), vol. 3847, pp. 159–182. Springer, Heidelberg (2006)
5. Tanaka, Y.: Meme media and a world-wide meme pool. In: Proceedings of the fourth ACM International Conference on Multimedia, pp. 175–186. ACM (1996)
6. Kuwahara, M., Tanaka, Y.: Advanced "Webble" Application Development Directly in the Browser by Utilizing the Full Power of Meme Media Customization and Event Management Capabilities. In: IEEE International Conference on Multimedia And Expo, Tempeku Workshop: Tangible Edutainment Media for Playful Evolution of Knowledge and Understanding, ICME 2012, Melbourne, Australia, pp. 211–216 (July 2012)
7. Kuwahara, M., Tanaka, Y.: Webble World - A Web Based Knowledge Federation Framework for Programmable and Customizable Meme Media Objects. In: Proc. of the IET International Conference on Frontier Computing, Taichung, Taiwan, pp. 372–377 (2010)
8. Dawkins, R.: The Selfish Gene. Oxford University Press, New York (1976)
9. Weiler, G., Graf, N., Schera, F., Hoppe, A.: Ontology based data management systems for post-genomic clinical trials within a European Grid Infrastructure for Cancer Research. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 6434–6437 (2007)

# The Mindset of a Webble World Citizen: Developing Applications in a Meme Media Environment

Micke Kuwahara and Yuzuru Tanaka

Meme Media Laboratory, Hokkaido University, Sapporo, Japan
{mkuwahara,tanaka}@meme.hokudai.ac.jp

**Abstract.** Webble World is the latest generation of IntelligentPad implementations that is fully available online for any individual or organization interested in using it. It is a meme media objects environment that allows users to develop rich software applications for the web, even without any skills of programming, but instead with simple basic mouse operations which is used to configure and arrange these meme media objects, known as Webbles, in ways that may create various types of complex behaviors and thus form full-fledged applications. It is intended to be inviting and used by the general public as a whole, skilled programmers and their grandmas alike, as well as academics and corporations. It is a framework for learning and improving, a knowledge federation where humanity can collect their shared skills in software application development. But with this power comes the responsibility to build and develop, not just for you and a few chosen, but for the entire world at all times. In order to support a useful and open federation of meme media objects the developer must try to get into the head of every possible future user of his component, and the users need to see that Webble systems is a network of objects with little or no central control but instead a cooperation of  independent parts. To fully appreciate Webble World the user need to adapt a certain mindset, and it is that mindset this paper will try to explore.

**Keywords:** Meme Media Objects, Customize, Configure, Web, Share, Distribute, Resource, Interact, Participate, Federation.

## 1 Introduction

The mission statement goes as follow; A platform available online via any common browser running on more or less any common operative system. This platform allows the user to do multiple form of interactions, ranging from search and use applications and/or widgets that perform anything software applications usually and less usually do, as well as to develop and build the very same with only the most basic of computer operations. In this environment everything is an independent object that the user can configure, assemble and manipulate at free will. Every application is built up by these objects, arranged in a simple, yet effective hierarchic structure and ordered to communicate in certain ways in certain directions. When done correctly, no classic programming is needed and full-fledged software applications may be constructed in a fraction of usually needed time and mainly with mouse operations only. These applications are then just a mouse click away from being saved and published online and be available to the worldwide Internet users.

Any application found, no matter its intended use, are still just a composition of smaller and more primitive objects and thus can be taken apart and re-arranged at each users will. Software is no longer bound to be only what its original developer decided, but is instead just a source of inspiration for future users to create and improve. It is an environment of software evolution that invites even the most basic Internet user to participate, by just tweaking and re-configure a constantly growing federation of objects and compound objects available.

The more advanced and classical programmers can use their skills to either create the smallest and most primitive of these building block objects, both in classic code, from easy accessible templates or by, inside the environment, adding more complex coded behaviors to existing objects via available scripting interfaces.

With the use of wrapping objects, more or less anything outside this environment can be drawn in and encapsulated and work like any other object of the framework even though its content and base behavior originates from other web and desktop sources, not created within the platform. It is a framework designed to help human beings save time, share knowledge and participate globally despite background and platform.

Such a mission statement might sound daunting enough, but after years of research and development it has come to be the less complex of two parts in order to introduce an environment as explained above. The more difficult part has shown to be to tweak the mindset of those who think they already know how it is done and those who think they do not have what it takes to do it. We believe that is where the challenge really lies.

## 2 Background

We will not give you the full history of meme media and knowledge federation within this paper. That can be better found in our earlier work listed in the references [1] [2] [6] [7] ,but briefly just give you a glimpse of its origin and major milestones.

The concept and idea of the meme media object was born about 20 years ago, and its inspiration as the name suggests comes from Richard Dawkins [4] lectures on the meme, the gene counterpart of the human mind and cultural knowledge. The suggestion that our thoughts evolve and mutate like genes in a human interactive environment. From that was theoretically designed the 'meme media object', a digital, relatively small and simple carrier of a specific part of knowledge and or behavior that like both memes and genes may interact and connect with any other meme media object and form more complex 'life forms'.

A decade later and several test implementations into this train of thoughts (where many more sources of inspiration have been involved [11] [12]) was developed the 'IntelligentPad' [5], a meme media object framework application for windows PC. Several IntelligentPad-like systems was developed in the early part of this century both by venture companies as well as well established enterprises like Fujitsu and Hitachi. All based on our design. These products worked very well for their time, but had problem reaching a broad audience. This was because of multiple reasons, but to mention just a few, these systems were not free, or they were only used internally and only the final products were then released, basically locked down for change, which of course removed some major selling points of being easy accessed and available as well as reconfigurable. The implementations were still good but the benefits of IntelligentPad were mainly aimed for a selected few to reap its benefits. Those who had access to it saw the great potential, but most did not have that access.

But it was not that simple either. There was another factor that crippled the global whirlwind of IntelligentPad and that was as mentioned before, the mindset of its users. In those days it was mainly aimed for skilled programmers in order to simplify their work, but they had been developing code for years so the somewhat different approach of thinking that came with IntelligentPad did not seem to penetrate the average developer so well. Instead of creating a large library of small reusable objects, a gigantic library of larger one-purpose objects were formed, where many IP (IntelligentPads) became bulky and super specialized and more or less impossible for use outside its original settings. Of course there were simple blocks like buttons and textboxes made, but often more complex needs was implemented, not as more building blocks, but instead as additional code in already very specialized objects. Programmers tend to develop a form of tunnel view when writing code, and though it was a bit more natural to create smarter classes at some point it never became a rule to break down IP into smaller and smarter components for greater re-usability, this is of course also due to the fact that making one complex IP takes less time to make than ten well designed parts that would form the complexity needed. Time that would of course be regained the next moment similar needs were raised, but that was not often enough considered.

In 2007 a request was made to implement a smart reconfigurable tool for cancer research trial management [8] using the concept of IntelligentPad but the current available implementation was not able to fulfill every requirement, instead a new version of IntelligentPad framework would be developed. During a number of years this new platform was developed in parallel with world-wide projects that would benefit from this type of framework which also worked as a great stress tester of the underlying system. In 2010 that framework went online for limited use outside its academic confinements and in 2012 its stability and feature base was considered strong enough for the general public. This framework is called Webble World.

## 3 Webble World

For those of you that are familiar with meme media objects and IntelligentPad. A Webble is all that, and a bit more [3] [10]. Webble World is the browser based framework where Webbles live. When visiting a Webble World web site the user get access to a federation of Webble objects published online together with an even larger set of widgets and applications constructed and built with these Webbles.

## 3.1 The Inside of a Webble

As seen in figure 1, a Webble is defined by two parts, the model and the view. The model is the business logic part of the Webble, that handles matters unrelated to how the Webble is displayed or how the user interacts with it. The view on the other hand is all related to display and user interaction, which should only contain interface related logic.

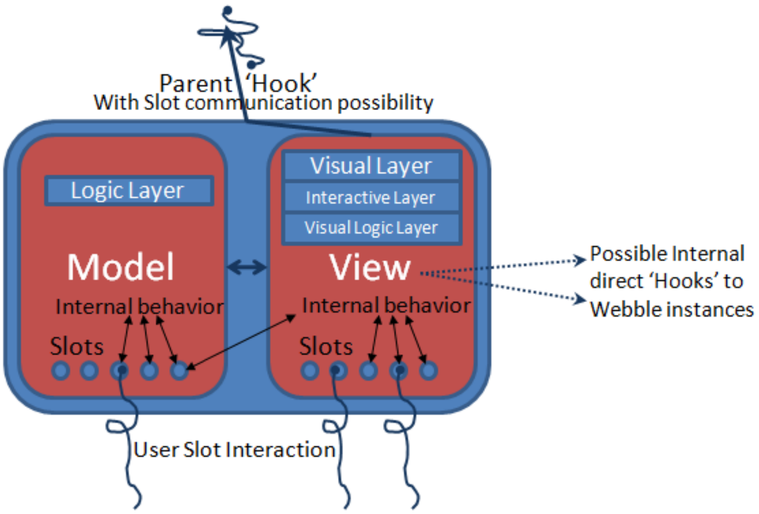


**Fig. 1.** A virtual cross section of a primitive Webble as described below

These two parts are coded the traditional way in a classic programming language and when combined and published to the platform becomes a Webble, also known as a primitive Webble. Being a Webble means that some specific characteristics are in place; The possibility to connect to other Webbles in a hierarchical structure, most commonly known as a parent – child relationship. A Webble can have unlimited number of children but only one parent. This allows the user to easier organize the Webbles and bind Webbles that work together in intuitive groups and sub groups. Each Webble has slots, both within the model as well is in the view. Slots are configurable properties of a Webble but they are also the basic communication channels both internally and externally as is demonstrated in figure 2 where the user can connect. Beyond these few specific trademarks a Webble may look in any way imaginable or contain any type of features.

Every Webble can also be duplicated, deleted and saved, as well as having its generic Webble behavior configured in order to protect it from unwanted, unintentional (ab)use when being used for a specific purpose.

When a Webble is saved, online or locally, it is stored plainly as an XML file, that includes all valuable meta data, slot values, child-parent relations and needed URI links. Rather small files that can easily be shared with the world.

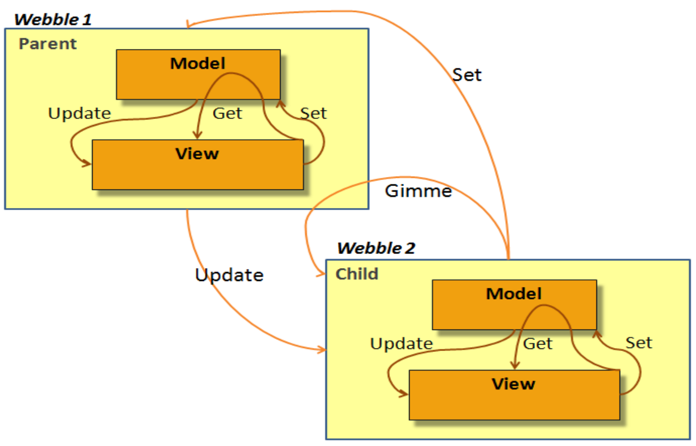


**Fig. 2.** The inner workings of slot communication of Webbles uses three methods, Gimme(<slotname>), Set(<slotname>, <value>) and Update()

## 3.2 Using Webbles

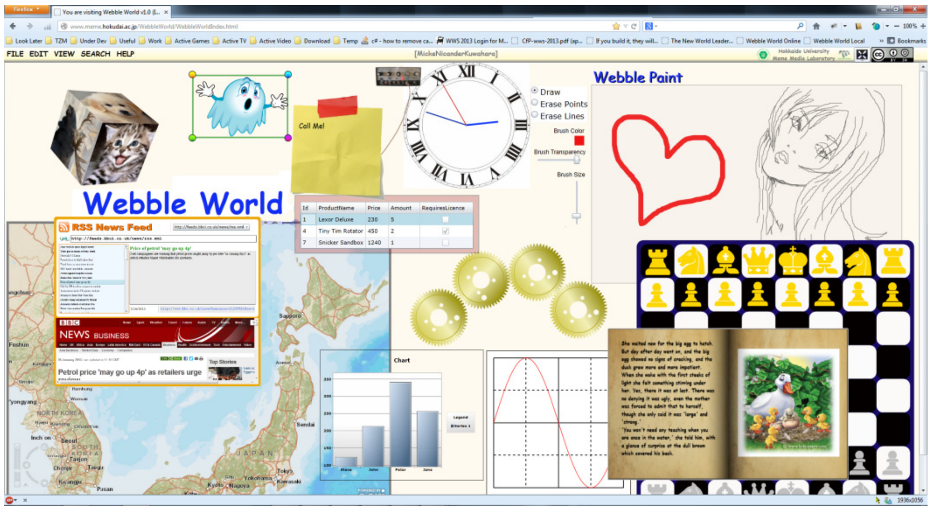
Using the Webble World interface for searching and locating available Webbles online and then load them into the environment for use and interaction is mere only the beginning. What follows is the configurations in order to create something new.

Two common initial approaches are often used. Either load something already made, an application that partly do what is wanted and then either add those Webbles that are missing or detach those Webbles that are wanted and dispose of those that are not, or completely start from scratch and load only the most basic primitive Webbles one needs. Either way, the primary work is to choose the most suitable Webbles for the task and when those are loaded into the Webble World desktop, begin arranging them into logical parent child relationships with each other. Webbles or Webble groups does only need to form relationships when they are directly communicating, but sometimes such connections are also good for better overview. Next is a combination of configuring slot properties via the provided interfaces and setting up slot connections with related Webbles so that values that need to be altered and shared during the lifetime of the application do so appropriately. And that is all there is to it. Form external object structure, configure internal object slots, connect slots between Webbles.

When done correctly, and this is the key, basically any software application can be made like this, but, as should now be very apparent, it does require a bit of a different mindset from traditional software development. Instead of thinking in terms of for-loops and if-statements a Webble Application developer needs to think like a communicator where each object must be allowed to speak in turn and when finished allow all other objects that so wishes to reply. Webble Applications work more like divisive individual friends working together to solve a problem by communicating each current state, than one centralized person (traditional software) that work alone doing every step on its own and boss external resources around. The later is sometimes maybe faster in a one purpose scenario, but it will be more or less completely lost in a new setting and environment and reusability in future scenarios will be severely crippled.

### 3.3 Available Webbles

Webbles comes in many shapes and forms. The traditional objects as buttons, text-boxes, labels and images etc exists, but also more complex ones like windows, charts, lists and customizable XAML (vector graphic script language) Webbles, just to mention a few. Also a lot of specialized Webbles for generating random numbers, transforming XML, keeping time and generate maps is included in the repository with a much smaller but very powerful set of Webble Control Webbles like the often used Event-Action Manager [10] that allows the Webble application designer to catch events and react accordingly.



| Id | ProductName | Price | Amount | RequiresLicence |
|---|---|---|---|---|
| 1 | Lexor Deluxe | 230 | 5 | ☐ |
| 4 | Tiny Tim Rotator | 450 | 2 | ✓ |
| 7 | Snicker Sandbox | 1240 | 1 | ☐ |

**Fig. 3.** A small set of available Webble Widgets built by another set of primitive Webbles. How these Webbles are connected and work together is best discovered by visiting Webble World, where these Webbles can be found and examined and experimented with.

Some of the most advanced applications [8] [9] developed in Webble World was made for specific environments and users dealing with highly secured data, and are therefore not available to the general public, but there are still a few applications and games to test, and also our, still under development, data analyzing dash board.

## 4 Current State

The current version of Webble World, as of the time of this writing, was built on top of Microsoft Silverlight technology, a browser plug-in with rich state of the art capabilities for XAML vector graphic design and a powerful C# SDK. More powerful than Flash regarding its coding possibilities, which was mainly made for web designers and artists with rather weak scripting tools.

Back then, in 2007, plug-ins in general was considered the primary choice for creating web software that required that extra, and though many different approaches was considered, choosing Silverlight at that time, five years ago, was a decision made without any later regret.

## 4.1 Strengths

Today's version is the third iteration of the framework since it has been 'forced' to closely follow the level of progress done by Microsoft in regard of Silverlight. For each iteration the system has become stronger and more pleasing, in regard of stability, feature richness, interface appeal and user friendliness. It takes only a few hours to create a classic ball and paddle game like Arkanoid, and only a few minutes to create a common style interactive website.

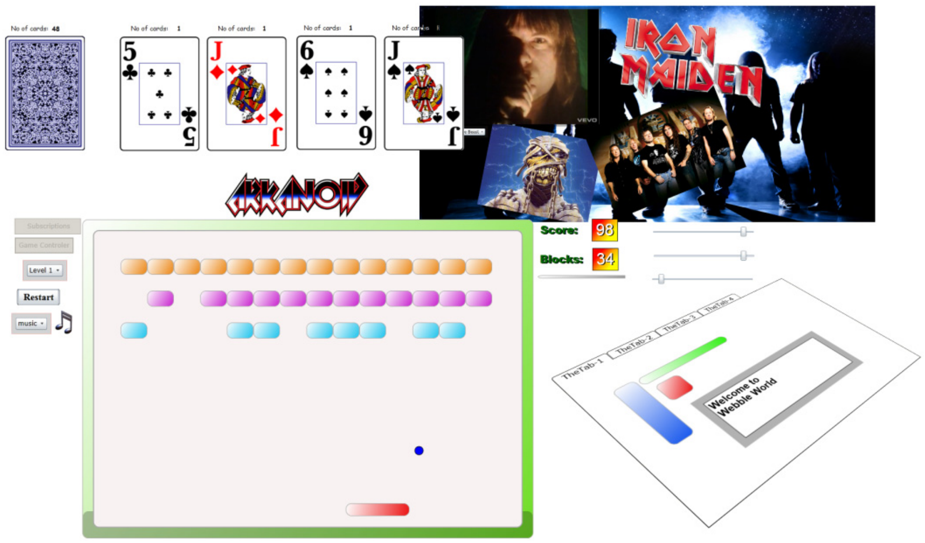


**Fig. 4.** The 'Dashboard' gets a whole research paper of its own, so here we only show apps that entertain without any attempt to educate, except future Webble application developers that is

At the beginning of Webble Worlds Life cycle, each new application idea also required one or more primitive Webbles to be built and coded as well, but today the repository is getting rich enough to allow most current Webble application developers to build numerous types of applications without the need to look for building blocks outside the repository. This is partly due to the latest increase of primitive Webbles but also the strengthened capabilities of Webble World that allows the user to configure and control Webbles beyond their original intended use. Two Webbles of the same primitive class can sometimes look so different in appearance and behavior due to slot configuration, custom slot creation, and increased vector graphic tree control, that it would not be possible to realize their common origin without a much more detailed examination of their internal meta data.

### 4.2 Weaknesses

But it is not all smell of roses in the Webble garden, there are problems. One that has already been mentioned is the mindset of most visitors, a mindset that is not necessarily wrong but does not apply well with Webble World interface it seems. Those that already has a set of tools and skills to create web applications, does not find the motivation to do it in an unknown environment as often as one would wish, and those who lack both skills and the tools, does not seem to see how any tool or framework could ever give them the power to develop software applications, and in both cases the visit becomes too short and unsatisfying. Equally so for the Webble World development team who is not given enough feedback in order to find ways to lower the initial threshold. The search for that missing piece, here described as an inviting mindset, that would turn Facebook users into Webble developers are still yet to be found.

The other problem is not less serious, and maybe even somewhat related, but it is a problem with a much clearer image of how to solve. This problem is called Silverlight. Silverlight promised a browser independent and OS independent solution, and for some time it seemed to be just that, even though Linux never really reached the finishing line. But as the years passed, reality became different then the promise, and Apple computer users did not receive the same Silverlight experience as Windows users did. But this is only the technical issues, the emotional ones are bigger still, so ones again a mindset issue. The world is full of people that shuns anything Microsoft creates, and Silverlight was no exception, unfortunately those were often the very same people that might love Webble World, the open source developers. And finally came, if not the nail in the coffin or a killing blow, at least a strong indication that it is time to move on. Tablets and phones, even those made by others than Apple, does not, and as far as is known today, will not ever support plug-ins. Even so much so that Microsoft more or less quietly seems to indicate the termination of future development of Silverlight as a browser plug-in, but instead seems to focus it only towards its use to develop device interfaces.

But this is of no real surprise, Internet moves in mysterious ways, and one who dwells in the land of the unknown, in uncharted waters, must always be prepared to change course. Webble World can only become stronger.

## 5 Next Generation

A code base that more or less runs on every device, OS or browser thinkable, without any requirement of plug ins, on technology with decades of quality assurance and a huge world-wide base of already existing skilled developers. A coding that has not stagnated but is still under constant improvement, that carries all that Silverlight offered and delivered in form of vector graphics powers, alongside with an open source, free access mindset. That is all it takes, and look and behold, thus enter HTML5 the stage and our problem is solved.

Webble World HTML5 version has just entered its initial developing stage and is planned to see the first light of day within the boundaries of this year. It will perform at least as good as the current Silverlight version, and it will also allow and support different approaches for integrating legacy Webbles originally created in Silverlight in order to make the transition as smooth as possible, mainly be supporting the same xml formats, or by adding a Silverlight wrapper Webble (the later would still contain Silverlight shortcomings though).

This new version will run Webbles inside windows tablets, iPads and android phones as well as desktop computers with Linux, Apple OS and Windows. Everything that runs HTML5 that is. To create primitive Webbles will require a basic skills of HTML5, CSS3 and mainly JavaScript, but for the larger part of web users the Webble will be what it always was, a meme media object ready to configure, organize and use directly in your favorite browser.

## 6 Summary and Discussion

This paper is trying to investigate the mismatch between the mindset of a possible Webble World user and the development interface approach offered by Webble World. When it comes to skilled programmers a few problems seems to surface; They're familiar with working in code of their chosen language and feels out of water when asked to develop in a code-free WYSIWYG environment, they also believe that development goes faster the traditional way, not considering reuse or meme object separation, which would be true if you do not consider the time you save the following times you need similar components. The non-programming user on the other hand still seem to perceive Webble World to be too complex, at least at the first look. Concepts as slot, slot connections and parent-child relationships sounds daunting to regular Internet users who therefore avoid investing any further time into understanding it. It could also be, though this is mere speculations, that the internal image of an object differs from what these users see on the screen presented as a Webble, which then in turn stifle creativity. The same creativity we originally wished to inspire.

The need for feedback is constant and can never be too much so the entire world is welcome to join the Webble World community[1].

We claim that the world entire would benefit immensely if software development was conducted in environments like Webble World. The benefits would be, after the initial creation of major useful meme media objects, much time saving whenever a new application needs to be built, mainly since most parts have already been developed. It would also be good for the world that previous non-programmers suddenly was given an opportunity to participate in the development of software. Our understanding, especially within

[1] `http://www.meme.hokudai.ac.jp/WebbleWorld/WebbleWorldIndex.html`. (login info if requested; username: 'webble', password: 'webble')

the field of user interaction, would explode just by study how users organize their own personal software. Also in times like this when file sharing is under constant debate, the idea of sharing, not only content and context, but also usefulness, behavior and entire customizable software solutions would definitely increase the power of the people and challenge the established ideas of open source vs. proprietary software.

We believe the Webble World platform will never be a finished system, more likely it will slowly turn into meme media object on its own, and will come in various shapes and forms, built up within, by Webbles itself. The shared base would only be the power to understand how to read a Webble file and turn it into something tangible. We believe one of the first systems like this exists today, all that it takes is a little bit of a new mindset, or for us to better understand the current one.

## References

1. Tanaka, Y., Imataki, T.: IntelligentPad: A Hypermedia System allowing Functional Composition of Active Media Objects through Direct Manipulations. In: Proceedings of the IFIP 11th World Computer Congress, San Francisco, USA, pp. 541–546 (1989)
2. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)
3. Kuwahara, M., Tanaka, Y.: Webble World, a web-based knowledge federation framework for programmable and customizable meme media objects. IET International Conf. on Frontier Computing CP568, 372–377 (2010)
4. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)
5. Fujima, J.: A Unified Framework for Organizing, Accessing, and Federating Web Resources. Hokkaido University (2006)
6. Tanaka, Y.: Knowledge federation over the web based on meme media technologies. In: Jantke, K.P., Lunzer, A., Spyratos, N., Tanaka, Y. (eds.) Federation over the Web. LNCS (LNAI), vol. 3847, pp. 159–182. Springer, Heidelberg (2006)
7. Tanaka, Y.: Meme media and a world-wide meme pool. In: Proceedings of the fourth ACM International Conference on Multimedia, pp. 175–186. ACM (1996)
8. Weiler, G., Graf, N., Schera, F., Hoppe, A.: Ontology based data management systems for post-genomic clinical trials within a European Grid Infrastructure for Cancer Research. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 6434–6437 (2007)
9. Aloia, N., Concordia, C., van Gerwen, A.M., Hansen, P., Kuwahara, M., Tuan Ly, A., Meghini, C., Spyratos, N., Sugibuchi, T., Tanaka, Y., Yang, J., Zeni, N.: Design, implementation and evaluation of a user generated content service for Europeana. In: Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries: Research and Advanced Technology for Digital Libraries, TPDL 2011, pp. 477–482 (2011)
10. Kuwahara, M., Tanaka, Y.: Advanced "Webble" Application Development Directly in the Browser by Utilizing the Full Power of Meme Media Customization and Event Management Capabilities. In: IEEE International Conference on Multimedia And Expo, Tempeku Workshop: Tangible Edutainment Media for Playful Evolution of Knowledge and Understanding, ICME 2012, Melbourne, Australia, pp. 211–216 (July 2012)
11. Stefik, M.: Introduction to Knowledge System. Morgan Kaufmann (1995)
12. Huberman, B.A.: The Ecology of Computation. North-Holland (1988)

# Linked Open Webble: Connecting Webbles to the World Wide Web

Nicolas Spyratos and Tsuyoshi Sugibuchi

Laboratoire de Recherche en Informatique, Université Paris-Sud 11, France
{Nicolas.Spyratos,Tsuyoshi.Sugibuchi}@lri.fr

**Abstract.** Webble technology is a media technology allowing media generation by direct manipulation. Webble users can build systems and interfaces by interactively combining visual components, called webbles, and dynamically connecting functions of components to create new, composite components. A casual user can simply place one webble on another and plug it in, and the combination will be able to run immediately *assuming* the connected functions are compatible. However, the current webble technology provides virtually no means for (a) searching a potentially huge pool of webbles to find those needed for a specific application and (b) knowing whether the functions of two webbles are compatible (and therefore whether the webbles can be connected). In this paper, we propose an approach to a solution for this problem.

## 1 Introduction

Webble technology is a media technology allowing media generation by direct manipulation [1]. Webble users can build systems and interfaces by interactively combining visual components, called *webbles*, and dynamically connecting functions of components to create new, composite components. Each function of a webble is physically represented as a webble slot. Casual users can simply place one webble over another (by drag-and-drop), plug it in, and the combination will be able to run immediately, *assuming* that the connected functions are compatible.

Therefore, when a webble is dropped on another, one of two things can happen: if the relevant slot types of the two webbles are compatible then the webbles are effectively connected and the new, composite webble is available for further use; otherwise there is no connection established and therefore no creation of a new, composite webble. In other words, placing one webble on another and plugging it in is *not* necessarily creating a connection, unless there is slot compatibility.

The long term vision of webble technology is to start with an open pool of atomic webbles created by an initial group of technology oriented users, and grow it to a world wide webble pool that can be accessed freely by casual or professional users. The webble pool users will either use existing components for their applications, or create new components from old and contribute to growing the pool by making their webbles available to the webble community.

However, in the current version of webble technology, there is little or no help provided to the casual user in order to (a) search a potentially huge pool of webbles to find those needed for a specific application and (b) know whether the functions of two webbles are compatible (and therefore whether the webbles can be connected). As a consequence the user must try to connect webbles by trial and error - a formidable task, as the number of slot combinations is huge even in a pool of a few tens of webbles.

In this paper, we propose to solve this problem by introducing annotations at two levels: at the webble level and at the slot level. Our claim is that if these annotations are based on standard, controlled vocabularies then one can define search mechanisms (based on those vocabularies) allowing to access webbles of interest, and to have information about compatibility of their slots. Additionally, the world of webbles can then communicate with other worlds using the same vocabularies, and in particular it can communicate with the world wide web.

Today, there are several standard vocabularies, such as Dublin Core[2], IEEE's LOM [15], CIDOC-CRM[3], ACM's CCS [16] and others. These vocabularies are rather widely used today in several areas, including digital libraries, archives [4], or distant learning systems, to mention just a few areas. Annotation framework using such standard vocabularies or more sophisticated ontology representing domain knowledge [5][6][7] is continuously very hot research topic and already in use in various application domains. One of most successful examples is semantic annotation of biomedical data. Nowadays various biomedical data are annotated by using several standard vocabularies [8][9] and those annotations are widely used for biomedical data analysis [10] and information retrieval[11].

Our proposal is also motivated by another simple observation regarding webble technology. To begin with, from a programming point of view, webbles can be regarded as a visual, dynamically typed programming language. Compared to static typing, dynamic typing has the advantage of enabling the connection of arbitrary pairs of slots during runtime. However, such over-flexibility entails the risk of misconnections, as well as difficulties in choosing an appropriate combination of slots from a huge number of possible combinations.

One possible solution is to enhance dynamic programming with *optional type annotation*[12][13][14]. It is the kind of annotation that programmers optionally embed into source code for indicating types of variables or function signatures. Optional type annotation has no effect on the run-time semantics of the programming language. However, it can be used for improving productivity and runtime performance of dynamic programming languages by static type checking before execution, JIT (just-in-time) compiler optimization and content suggestion in IDEs (integrated development environments). Today, optional typing is a very popular approach for introducing more stability and strictness to dynamic programming languages while preserving their flexibility.

The idea of optional type annotation is not limited to the programming world. In the world wide web, we also find various *microformats* for annotating pieces of HTML documents by using predefined, controlled vocabularies. For instance, the *hCard* standard defines a set of class names for indicating contact information. By using hCard, we can explicitly indicate telephone numbers like `<span class="tel">0678901234</span>`. Microformats also enable us to preserve expressiveness of HTML itself and to *optionally* introduce semantics of data in HTML documents.

From these observations, we believe that there can be various useful applications of optional type annotation for webbles as well. However, we have to keep in mind that webbles is *not* a RAD (Rapid Application Development) tool. It is a media architecture. Therefore, we should not limit possible annotation types for webbles only to those used for data types in programming.

For instance, if you make a dialog webble for entering book reference data, it is a good idea to annotate both the webble and its slots, using the Dublin Core vocabulary. In this way, if later on you develop a new application, say for EPUB books, you can easily find that dialog from a webble repository and almost automatically connect it to your new application. This is possible because Dublin Core is not only a programming technology but also a common vocabulary used for annotating files of various media formats, including EPUB. We can also imagine similar examples not only for Dublin Core, but also DBpedia, YAGO, AAT, MeSH and any other controlled vocabularies used in the Web. By annotating webbles with such widely used vocabularies, webbles can gradually become citizens of the Linked Open Data (LOD) world.

In this paper, we introduce a generic and formal framework for annotating webbles. The kind of vocabulary that we envisage can be a pre-defined vocabulary specific for the webble framework, or the LOD world. The framework that we propose here works with any kind of controlled vocabulary. Moreover, our framework doesn't touch the webble architecture itself - it just overlays an additional, semantic layer on top of the naked webble architecture.

## 2 Webble Annotations

The current webble technology offers some means to attach annotations both at webble level and at slot level. At webble level we find several kinds of annotations in the current webble implementation: name, class name, developper, group identifiers, description keywords and “metadata”. However, the terms in these annotations do not come from any standard vocabulary and the mechanism for creating metadata is rarely used (i.e. most webbles have empty metadata).

At slot level, we find two kinds of annotation: (a) a slot name and (b) a slot type. However, none of these information items is exploited from the user's point of view. Moreover, there is no mechanism available to express attributes (or properties) of a slot other than its name and type.

Another problem is that the names in these annotations do not come from any standard vocabulary either. For instance, if someone wants to search webbles by name, he has to read a document about webble file format to identify the name of the annotation which represents webble names. This lack of relationship with widely-used controlled vocabularies prevents visibility, searchability, and reusability of webbles in the context of the world-wide-web.

To solve this problem, we propose to enrich the existing annotations of a webble at both levels (i.e. at webble level and at slot level), and moreover use annotations for designing search mechanisms and for reasoning about webbles. The annotation framework that we propose in this paper aims at the following goals:

**Easy to define.** To encourage webble developers and users to start annotating webbles, we should keep the annotation model as simple as possible. From user's point of view, a webble annotation in our framework is just a set of attribute-value pairs. Our framework also provides a controlled vocabulary for webble annotation: it's a small set of terms borrowed from RDF together with a few webble-specific terms. Reasoning about annotations is based on a set of entailment rules defined by the framework. However, as far as users are concerned, reasoning is transparent: it is basically a kind of detail users do not need to care about.

**Annotation of both media containers and media contents.** Webble is not only a GUI toolkit, but also a media architecture. A webble can be a container of media contents. When we annotate webbles, we need to distinguish between media containers and media contents. For example, suppose an image webble has an annotation "`dc:creator` is Hokkaido university". In this case, Hokkaido university is a creator of what? The webble itself, or the image displayed by the webble? Actually, we need to handle both cases. Out framework allows users to annotate both webbles and contents represented by webbles.

**Cooperation with webble architecture.** Our framework can cooperate with the basic mechanisms of webbles, in particular, slot connection and webble composition. It can map slot values to annotations, and it can reify annotations as webbles. These features allow us to interactively annotate webbles and media contents by using slot connections and webble compositions.

**Easy to map to RDF.** After annotating webbles, the next step is to make webbles visible in the LOD world. A straightforward way to do this is to represent webble annotations by using the standard metadata format in LOD world, namely RDF. Our framework provides a simple mechanism to map webble annotations to RDF graphs. From webble annotations, our framework produces RDF graphs representing annotations of webbles, annotations of contents, and the relationship between a webble and its contents.

**Enhance the behavior of webbles.** If the webbles are annotated, then the webble runtime environment can use annotations for various use. In particular, RDF mapping of webble annotations enables us to use the powerful reasoning system of RDF in order to control the behavior of webbles. In this paper, we demonstrate rule-based restrictions of webble composition and slot connection by using webble annotations.

## 2.1 Annotating Webbles and Their Slots

Figure 1 shows the work flow in an example that we shall use as a running example throughout this paper. In this example, a user wants to create an image

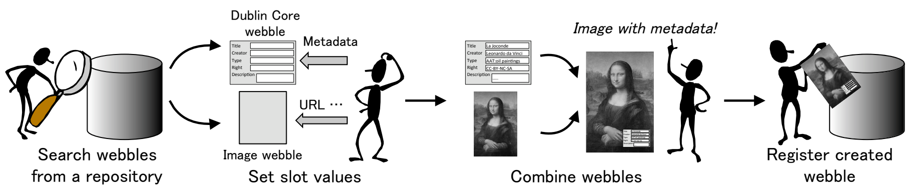


**Fig. 1.** A example work flow

webble whose content (i.e. the image contained in the webble) is annotated by the content of a metadata webble. To achieve this goal, the user has to do the following: (1) search for webbles in repositories for creating the composite webble satisfying his goal; (2) set slot values, define webble compositions and slot connections to create the intended webble; and (3) possibly store the created webble into a repository for future reuse. Figure 2 shows the structure of the composite webble.

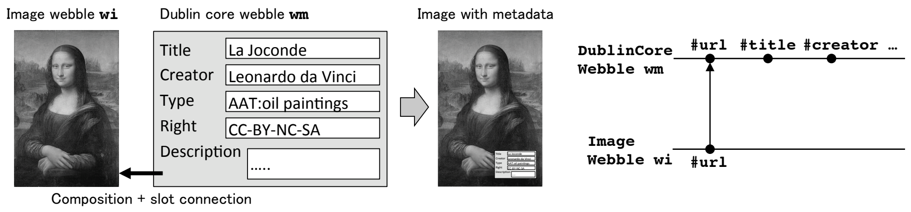


**Fig. 2.** Structure of the example webbles

In the composite webble, we can see an image webble `wi` which can contain and display an image, and a metadata webble `wm` which can contain metadata defined in Dublin Core (hence its name "DublinCoreWebble"). The user wants to compose these two webbles so that the metadata contained in `wm` is added to the image displayed by `wi`.

This work flow can be done perfectly without any annotations at all. However, from the practical point of view there are several problems.

1. How can the user efficiently search for webbles in a webble repository?
2. How can the user make this composition without tedious try-end-error repetition?
3. How can the user expose results of the webble composition in a searchable form for future reuse?

To solve these problems, we need (1) well-organized webble annotations on which to base search mechanisms (2) a query language for searching webble repositiories (3) a mechanism preventing users from performing forbidden operations, and (4) a mapping rule from annotations of composite webbles to a searchable form.

In this paper, we first discuss webble annotations (point (1) above); then a mapping rule from webble annotations to RDF (point (4) above); and finally we demonstrate a plug-in type system to restrict user's operations in order to avoid errors (point (3) above). We do not address the issue of defining a query language (point (2) above). To begin with, let's define formally what an annotation is.

**Definition 1 (Webble annotation and slot annotation).** *Let $i$ be the identifier of a webble or of a slot. The annotation of $i$, denoted An(i), is defined to be a set of attribute-value pairs, namely:* $An(i) = \{(A_1, v_1), ..., (A_n, v_n)\}$.

There are several important points to mention regarding the definition of a webble annotation. Firstly, we assume that both webbles and slots are associated with identifiers, and moreover that we can tell whether an identifier is a webble identifier or a slot identifier (e.g. from its syntax). In the rest of this paper, we use the following conventions: (1) if $w$ denotes a webble then `w` denotes the identifier of $w$, and (2) if $s$ is a slot of webble $w$ then `w:#s` denotes the identifier of slot $s$ of $w$.

Although we do not discuss how webble annotations are defined, we assume two types of annotations: *system annotations*, defined by webble developers and *user annotations* defined by users. Attributes and values of system annotations are pre-defined by developers or dynamically assigned by webble runtime environments, while user annotations are defined by users who wish to append application-specific inforation to webbles.

Finally, we assume that all attribute names $A_1 \ldots A_n$ come from some controlled vocabulary. In particular, in order to be able to map webble annotations to RDF, we will introduce a restriction such that every attribute name must be the URL of an RDF property. Therefore in the rest of the paper we use only URLs of RDF properties as attribute names.

Figure 3 shows an example annotation of an image webble $wi$. Basically all annotations in this example are system annotations.

```
     An(wi) = {(dc:title, "Image Webble"), (rdf:type, ImageWebble),
               (dc:creator: "Hokkaido Univ."), (wb:hasSlot, wi:#url)}
An(wi:#url) = {(dc:title, "URL"), (rdf:type, TextSlot),
               (wb:hasValue, http://a/b/c.jpeg)}
```

**Fig. 3.** Webble annotation example

In this example, some basic information (name, title, ...) are described by using terms from Dublin Core, whose name space is denoted as `dc`. The example annotation also describes ownership and value of `url` slot by using `wb:hasSlot` and `wb:hasValue` that come from a controlled vocabulary that we have specifically designed for webble annotations. We call this vocabulary *webble vocabulary* and in the rest of the paper its name space is denoted as `wb`. Note that `wb:hasValue` is a typical dynamic annotation. Indeed, the value of this annotation is dynamically assigned by a webble runtime environment to indicate the slot value of an annotation.

An important point to notice here is that webble annotations describe only webbles and their slots. Indeed, the example annotation of Figure 3 does not say anything about the image `http://a/b/c.jpeg` displayed by the image webble `wi`. In other words, webble annotations in this figure annotate only media *containers* and do not annotate media *contents*.

## 2.2 Annotating Slot Values

To annotate media contents, our approach uses an entailment rule to *infer* annotations of media contents from annotations of media containers and their state. In other words, our framework assumes that metadata of a media content is represented as state of a media container. Regarding the webble architecture, the state of a webble is externalized as a set of slot values.

```
        An(wm) = {(dc:title, "Metadata"), (rdf:type, DublinCoreWebble),
                  (wb:hasSlot, wm:#url), (wb:hasSlot, wm:#title), ...}
   An(wm:#url) = {(dc:title, "URL"), (rdf:type, TextSlot),
                  (wb:hasValue, "") }
 An(wm:#title) = {(dc:title, "Title"), (rdf:type, TextSlot)}
                  (wb:represents, dc:title), (wb:annotatesValueOf, wm:#url),
                  (wb:hasValue, "La Joconde") }
           ... ... ...
```

**Fig. 4.** Slot value annotation example

Figure 4 shows an example annotation of DublinCoreWebble `wm` and its slots (only annotations of `url` slot and `title` slot are shown).

The properties `wb:represents` and `wb:annotatesValueOf` in the annotation of `title` slot are terms from the webble vocabulary used for annotating *values* of slots. Intuitively, the annotation of `title` slot says that “the value of `title` slot represents `dc:title` of something specified by `url` slot”.

This statement is represented by three terms of webble vocabulary and one entailment rule. Figure 5 illustrates the mechanism of slot value annotation. In this figure, `wb:represents` associates slot $t$ with an attribute name $p$.

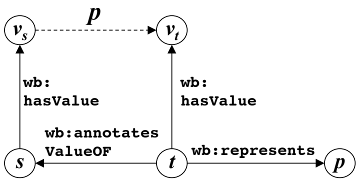


**Fig. 5.** `wb:represents` and `wb:annotatesValueOf`

wb:annotatesValueOf associates slot $t$ with slot $s$. The slots $s$ and $t$ have values $v_s$ and $v_t$. For these annotations, webble vocabulary defines the following entailment rule $R_1$.

$$\frac{(\mathtt{wb:hasValue}, v_t) \in \mathtt{An}(t) \wedge (\mathtt{wb:represents}, p) \in \mathtt{An}(t) \wedge \\ (\mathtt{wb:annotatesValueOf}, s) \in \mathtt{An}(t) \wedge (\mathtt{wb:hasValue}, v_s) \in \mathtt{An}(s)}{(p, v_t) \in \mathtt{An}(v_s)}$$

**$R_1$: Slot value annotation**

By $R_1$, the annotations in Figure 5 entails annotation $(p, v_t) \in \mathtt{An}(v_s)$, which is illustrated as the dashed line arrow in the figure. The role of this entailment rule $R_1$ is to *lift* slot annotations to slot *value* annotations. By using this mechanism, we can annotate media *contents* indicated by slot values.

Now we are ready to annotate contents specified by url slot. However, at present, url slot of DublinCoreWebble wm has no value. To get the url of the image displayed by the image webble wi, we can use webble composition and the slot connection mechanism. By combining wm with wi as a child, and connecting both url slots, the url slot of wm gets a url from url slot of wi. As a result, the annotation of wm's url slot gets the following attribute-value pair.

– An(wm:#url) = {..., (wb:hasValue, http://a/b/c.jpeg)}

Then, we apply entailment rule $R_1$ to associate slot values with annotations of slot values. In this example, rule $R_1$ matches with $s$ =wi:#url, $t$ =wm:#url, $v_s$ = http://a/b/c.jpeg, $v_t$ ="La Joconde", $p$ =dc:title. As a result, this example annotation entails (dc:title, "La Joconde") $\in$ An(http://a/b/c.jpeg). It is exactly an annotation of the image http://a/b/c.jpeg.

As we demonstrated here, our framework uses wb:hasValue, wb:represents, wb:annotatesValueOf, and entailment rule $R_1$ to associate slot values to annotations. These three terms and one rule form the core of the webble vocabulary. The full definition of the webble annotation core vocabulary can be found in the appendix.

## 2.3 Mapping Annotations to RDF Graphs

Conceptually a webble annotation is just a set of attribute-value pairs. However, by restricting attribute names to be only RDF properties, we can easily map webble annotations to RDF graphs. The basic idea is that we can represent an attribute-value pair $(a, v)$ in annotation $An(i)$ as an RDF statement "i a v" if a is an RDF property.

Table 1 shows the mapping from webbles and webble annotations to RDF graphs used in our framework. In our approach, each webble and each of its slots is mapped to an RDF resource. Static structure (ownership of slots, etc.), dynamic state (slot values) and annotations are mapped to RDF statements.

The graph in figure 6 shows an RDF representation of the example that we use in this section. In this graph, statements mapped from webble annotations are

**Table 1.** Mapping from webbles and webble annotations to RDF graphs

| Webbles and annotations | RDF |
|---|---|
| webble $w$ | `w` |
| slot $s$ of webble $w$ | `w wb:hasSlot w:#s` |
| value $v$ of slot $s$ of webble $w$ | `w:#s wb:hasValue v` |
| pair $(a, v)$ in annotation $An(i)$ | `i a v` |
| webbles and annotations | an RDF graph |

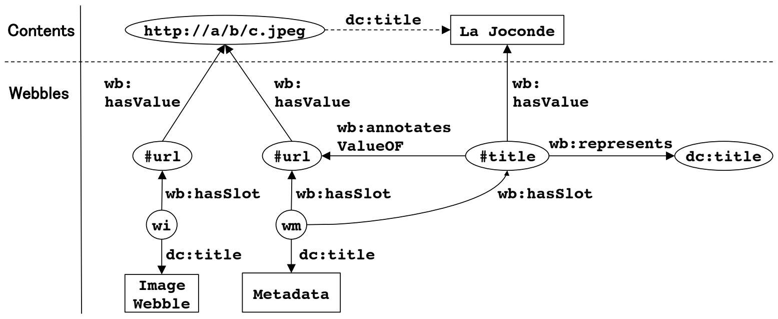


**Fig. 6.** RDF representation of webble annotation

represented as solid line arrows, and statements inferred by the entailment rule are represented as dashed line arrows. An RDF representation of webble annotations consists of graphs representing annotations of webbles and slots (webble graph), and graphs representing annotations of contents (content graph). As we can see from this example, our framework can produce "clean" content graphs, that is, content metadata is represented by using only standard controlled vocabularies without webble specific vocabularies. This is an advantage for interoperability of webble annotations with existing metadata standards. By storing RDF representations of webbles into webble repositories, we can search webbles by using SPARQL which is a query language for retrieveing information from RDF documents.

## 2.4 Enhancing Webble Behavior Based on User Annotations

The webble architecture provides a webble composition mechanism and a slot connection mechanism. The design of these mechanisms are intended to combine arbitrary pairs of webbles and slots. As we mentioned in the introduction, such over-flexibility entails the risk of misconnections. Regarding webble composition, the current webble architecture does not provide any information for restricting the combination of webbles to be combined. For slot connection, one possible solution is type-checking of slots. However, basically, webbles are designed and should be designed for preserving interoperability of webbles by choosing their slot types from a small set of common slot types (string, integer, real, etc.). Additionally, automatic data conversion is implemented between many combinations of common slot types. As a result, selectiveness of slot types is usually too low for helping users to avoid misconnections of slots (static type checking in statically typed programming languages helps us in finding some errors but it is still far from eliminating all errors).

As a consequence, if we want to introduce a mechanism which helps users in reducing miscombination and misconnection of webbles, it has to be application specific. For each application, we need to "plug-in" an additional type system of webbles and slots specific for the application. For this purpose, we can use *user* annotation of webbles. RDF mapping of webble annotations enable us to define custom type systems by using RDF schema (RDFS) or more advanced RDF vocabularies like OWL. Then we can annotate types of webbles and slots used in applications.

By using the example illustrated in figure 2, suppose a user intends to combine the image webble `wi` only with child webbles which describe metadata. In this case, the user can introduce a custom type system of webbles, then declare webbles by using this type system and check whether a webble can combine with another.

Firstly, the custom type system has to be declared as a set of RDF triples.

- `DublinCoreWebble rdfs:subclassOf wb:Webble`
- `DublinCoreWebble rdfs:subclassOf MetadataWebble`
- `CIDOCWebble rdfs:subclassOf MetadataWebble`
- ...

Then, the type of webbles acceptable as children by the image webble `wi` has to be declared as part of `wi`'s annotation. In this example, the acceptable type is declared by using `wb:acceptsAsChild` which is a term of webble vocabulary:

- `An(wi) = {..., (wb:acceptsAsChild, MetadataWebble) }`

Finally, we can check whether a webble can be combined with `wi` or not by evaluating the following rule.

$$\frac{\texttt{x wb:acceptsAsChild t} \wedge \texttt{y rdfs:subclassOf t}}{\texttt{y} \text{ can be combined with } \texttt{x} \text{ as a child}}$$

In the same manner, we can also restrict possible combinations of slots to connect by using annotations and rules. For instance, both image webble and Dublin Core webble have `url` slot whose type is TextSlot. TextSlots can accept texts in general as values. However, `url` slot of image webble has a URL of an image as its value, and `url` slot of Dublin Core webble has a URL of a resource to annotate in general as its value. We can declare such detailed information of slot value domains as annotations.

- `An(wi:#url) = {..., (wb:valueDomain, vra:Image)}`
- `An(wm:#url) = {..., (wb:valueDomain, rdfs:Resource)}`

In the above example, `vra:Image` (which is a class defined by Visual Resource Association [17]) is used to fix the domain of `wi:#url` slot to image URLs. The domain of `wm:#url` slot is also fixed to URLs of resources in general by using `rdfs:Resource`. By using this additional information about slot value domains, we can check whether values of one slot can be set to another slot by using the following rule:

$$\frac{\texttt{x wb:valueDomain s} \wedge \texttt{y wb:valueDomain t} \wedge \texttt{s rdfs:subclassOf t}}{\text{values of slot } \texttt{x} \text{ can be set to slot } \texttt{y}}$$

In this paper, we do not discuss details about webble vocabulary and rules for controlling behavior of webbles. The main point we would like to emphasise here is that by introducing application specific type systems and rules, we can carefully restrict the operations that users can perform. We believe that such restrictions on user's operations for avoiding misconnections of webbles is useful and somehow indispensable when we provide webbles for *casual* users.

## 3 Conclusion

In this paper we have argued that if the webble technology is to be used in its full strength it should be enhanced in two ways: (a) providing means for searching a potentially huge pool of webbles and (b) controlling webble's behavior by introducing application specific type systems and rules. We have proposed webble annotations as an unobtrusive way to remedy these two deficiencies of the current webble technology. Additionally, by studying an important use case (annotation of digital media contents) we have demonstrated the benefits of using standard, controlled vocabularies in the definition of webble annotations.

We are currently researching several aspects of composite webbles. The main concern is query formulation and evaluation for searching webbles in the webble repository. In this paper, we have demonstrated how webble annotations can be directly mapped to RDF graphs and how our mapping method produces "clean" RDF graphs of content metadata. Therefore we can search media *contents* in the webble repository by using the same manner for querying content metadata. However, the question is how to search media *containers* in the same repository. Webbles may have many slots and sometimes they are composite. Annotations over such complex webbles are represented as complex RDF graphs which is difficult to query. To allow typical users to search complex webbles from the repository, we need to introduce a mechanism for "summarizing" webble annotations into simpler forms to search. We have developed a theoretical framework of metadata summarization in the context of composite document repositories [18]. This framework works with metadata comprising a set of terms from a controlled vocabulary. In future work, we will extend this framework to handle metadata represented as RDF graphs.

## References

1. Webble portal, `http://cow.meme.hokudai.ac.jp/WebbleWorldPortal/`
2. Dublin Core Metadata Initiative: `http://dublincore.org/`
3. The CIDOC CRM: `http://www.cidoc-crm.org/`
4. Gill, T.: Building semantic bridges between mquseums, libraries and archives: The CIDOC Conceptual Reference Model. First Monday, vol. 9(5) (2004)
5. Handschuh, S., Staab, S.,, M.: CREAM: creating relational metadata with a component-based, ontology-driven annotation framework. In: Proc. of the 1st International Conference on Knowledge Capture, pp. 76–83 (2001)
6. Knight, C., Gasevic, D., Richards, G.: An ontology-based framework for bridging learning design and learning content. Journal of Educational Technology & Society 9(1), 23–37 (2006)
7. Kitamura, Y., Washio, N., Koji, Y., Sasajima, M., Takafuji, S., Mizoguchi, R. An ontology-based annotation framework for representing the functionality of engineering devices. In: Proc. of Asme IDETC/CIE 2006 (2006)
8. Gene Ontology Consortium.Gene Ontology: tool for the unification of biology. Nature genetics 25(1), 25–29 (2000)
9. Lipscomb, C. E. Medical subject headings (MeSH). Bulletin of the Medical Library Association, 88(3), 265 (2000)
10. Al-Shahrour, F., Daz-Uriarte, R., Dopazo, J.: FatiGO: a web tool for fnding signifcant associations of Gene Ontology terms with groups of genes. Bioinformatics 20(4), 578–580 (2004)
11. Lu, Z.: PubMed and beyond: a survey of web tools for searching biomedical literature. Database: The Journal of Biological Databases and Curation (2011)
12. Bracha, G., Griswold, D.: Typechecking Smalltalk in a Production Environment. In: Proc. of the OOPSLA 1993 Conference on Object-oriented Programming Systems, Languages and Applications, pp. 215–230 (1993)
13. PEP (Python Enhancement Proposals) 3107 Function Annotations: `http://www.python.org/dev/peps/pep-3107/`
14. Bracha, G.: Optional Types in Dart., `http://www.dartlang.org/articles/optional-types/`
15. Learning Object Metadata, IEEE Standard 1484.12.1–2000 (2002)
16. The ACM Computing Classification System (2012), `http://www.acm.org/about/class/2012`
17. Assem, M.V.: RDF/OWL Representation of VRA., `http://www.w3.org/2001/sw/BestPractices/MM/vra-conversion.html`
18. Sugibuchi, T., Tuan, L.A., Spyratos, N.: Metadata Inference for Description Authoring in a Document Composition Environment. In: Agosti, M., Esposito, F., Ferilli, S., Ferro, N. (eds.) IRCDL 2012. CCIS, vol. 354, pp. 69–80. Springer, Heidelberg (2013)

## Webble Core Annotation Vocabulary and Semantics

```
wb:Webble           rdf:type    rdfs:Class
wb:Slot             rdf:type    rdfs:Class

wb:hasSlot          rdf:type    rdfs:Property
wb:hasSlot          rdfs:domain wb:Webble
wb:hasSlot          rdfs:range  wb:Slot
wb:hasValue         rdf:type    rdfs:Property
wb:hasValue         rdfs:domain wb:Slot

wb:represents       rdf:type    rdfs:Property
wb:represents       rdfs:domain wb:Slot
wb:represents       rdfs:range  rdfs:Property
wb:annotates        rdf:type    rdfs:Property
wb:annotates        rdfs:domain wb:Slot
wb:annotates        rdfs:range  rdfs:Resource
wb:annotatesValueOf rdf:type    rdfs:Property
wb:annotatesValueOf rdfs:domain wb:Slot
wb:annotatesValueOf rdfs:range  wb:Slot
```

**$R_1$: Slot value annotation**

$$\frac{\begin{array}{c}(\texttt{wb}:\texttt{hasValue}, v_t) \in \texttt{An}(t) \wedge \\ (\texttt{wb}:\texttt{represents}, p) \in \texttt{An}(t) \wedge \\ (\texttt{wb}:\texttt{annotatesValueOf}, s) \in \texttt{An}(t) \wedge \\ (\texttt{wb}:\texttt{hasValue}, v_s) \in \texttt{An}(s)\end{array}}{(p, v_t) \in \texttt{An}(v_s)}$$

**$R_2$: Webble or slot annotation**

$$\frac{\begin{array}{c}(\texttt{wb}:\texttt{hasValue}, v_t) \in \texttt{An}(t) \wedge \\ (\texttt{wb}:\texttt{represents}, p) \in \texttt{An}(t) \wedge \\ (\texttt{wb}:\texttt{annotates}, s) \in \texttt{An}(t) \wedge\end{array}}{(p, v_t) \in \texttt{An}(s)}$$

# Building a Meme Media Platform with a JavaScript MVC Framework and HTML5

Jun Fujima

Fraunhofer IDMT, Children's Media Department, 99094 Erfurt, Germany
jun.fujima@idmt.fraunhofer.de

**Abstract.** Meme media technology is developed for provoking and supporting the evolution of knowledge on computers and the Internet. With the development of World Wide Web, the Web comes to have elemental functionalities for realizing meme media-based knowledge evolution. Webble World, the latest meme media platform, was implemented for seamless integration of meme media technology into the modern Web technologies. However, because it works on a browser plugin, it is slightly getting detached from other Web-related technologies. To achieve more seamless integration, we explore the possibility of developing a meme media platform with pure Web technologies. First, we look back on the original meme media architecture to understand what is necessary to implement the platform. Then, we pick up a JavaScript framework for building structured Web applications, called AngularJS. Consequently, we develop a prototypical implementation of a meme media platform and some example meme media objects on the platform. Because the platform is implemented only with HTML and pure JavaScript, it works on modern Web browsers that support the latest HTML5 and JavaScript specifications.

**Keywords:** meme media, HTML5, MVC framework, web application platform, web-based middleware.

## 1 Introduction

World Wide Web (WWW) is a powerful platform for recent computer applications. In December 2012, the specification of the next generation of HTML, known as HTML5, became a candidate recommendation of World Wide Web Consortium (W3C). Thereby, separated Web-based state-of-the-art technologies for building rich Web application, especially in the development of the interactive user interface, has proceeded to be unified. Only with those standard technologies, developers can implement modern Web applications.

In past days, such a rich Web application was developed with browser plugins like Adobe Flash or Microsoft Silverlight. However, in most case, those plugins do not work on recent mobile devices such as mobile phones or tablets. Owning to HTML5 and its related technologies such as Cascading Style Sheet (CSS), Scalable Vector Graphics (SVG), HTML Canvas, or WebGL, the cross-platform support in Web application development become possible.

Meme media technology [1] has been developed for provoking and supporting the evolution of knowledge on computers and the Internet. It provides those media objects which work as 'memes' coined by Dawkins [2]. It allows users to re-edit and redistribute different knowledge resources wrapped by media objects. Users can combine media objects through direct manipulation of them such as 'drag' or 'paste' to compose new composite media objects without any programming. One user uploads a new media object on the computer network. Then, another user can download and reuse it. Furthermore, another user can combine the object with other objects to compose a new composite object and publish it on the network. Through this cyclic process of users' interactions with meme media objects, wide variety of media objects are accumulated on the network, and the evolution of media objects is emulated.

With the development of World Wide Web, the Web has elemental functionalities for realizing the knowledge evolution based on meme media. Thinking of an HTML document as such a media object, it is possible for people to create a document (as a new media object), to publish it on a Web server. Another can browse it with a Web browser. In addition, regarding reediting and republishing media objects, people can use copy-and-paste operation to reuse some contents of existing Web documents. In this sense, the Web realizes a part of functionalities that meme media try to provide, but for the interactive Web application or tools, it is still difficult for the general public. People cannot extract parts of a Web application with direct manipulation of them and reuse them in other Web applications. To do this, people still need to do some programming with some expert knowledge.

The latest meme media platform, called Webble World[1] [3–5], was implemented with the aim of seamless integration of meme media technology into the modern Web technologies. However, because it works on a browser plugin, Microsoft Silverlight, the Webble World is slightly getting detached from other modern Web-related technologies. Silverlight provides a powerful framework and an execution environment for client-side rich Internet applications, but, unfortunately, it is not supported on many mobile devises which are recently one of the main targets in various application domains. Furthermore, because those plugins are not the first class citizen on the Web, casual Web users and Web developers tend to divert their attention away form the Webble World. To make the meme media technology available to casual users on multiple devises, it is necessary to implement a meme media platform based only with standard Web technologies, that is, with HTML and JavaScript. Now it has become possible.

In this paper, we explore the possibility of developing a meme media platform with pure Web technologies for more seamless integration with the Web and for availabilities on more different types of devices. We focuses on the implementation of the client-side media object functionalities such as basic media object architecture and combine operation of media objects. In recent JavaScript development, there exist some problems like the compatibility among different browsers and over flexibility. To avoid those problems we pick up a base framework for

[1] `http://www.meme.hokudai.ac.jp/WebbleWorldPortal/`

building structured Web applications, AngularJS[2]. Using AngularJS as the base framework, we made a prototypical implementation of a meme media platform. In addition, we developed some example media objects on the platform and tested operation of them on multiple Web browsers. Because we used only HTML and pure JavaScript to implement the platform, the developed platform works on modern Web browsers that support the latest HTML5 and JavaScript specifications.

## 2 Meme Media Platform and Its Architecture

In this section, we look back on the development of the meme media technologies for those who are not familiar with it. Then, we describe the original meme media architecture to understand what is essential to implement a meme media platform.

### 2.1 History of Meme Media Platform Development

The first version of the meme media platform, IntelligentPad, was originally developed on Smalltalk-80 in 1987 [6]. In IntelligentPad, each media object has a 2-dimensional view called 'pad'. Users can paste a pad onto another pad to set up a child-parent relationship to combine pads. In 1995, some major IT companies in Japan such as Fujitsu and Hitachi Software released a commercial version of IntelligentPad developed with C++. For the 3-dimensional environment, IntelligentBox [7] was developed in 1995. In IntelligentBox, each media object is represented as an object which has a 3-dimensional shape called 'box'.

All those platforms were 'standalone' local applications. However, another company (K-Plex) released Web-top version of IntelligentPad called PlexWare[3]. This platform is based on Microsoft ActiveX technology, therefor the platform environment can be embedded not only in Web sites for Internet Exproler but also in other Microsoft products such as Word, or Excel.

Kimihito Ito developed a distributed system version of IntelligentPad called CHIP (Collaborating Host-Independent Pads) [8] both with Micirosft .Net framework and with Java. In the CHIP, each media object is an independent local application that communicates with other objects via http messages over the Internet. In particular, it has a powerful Web browser object that can wrap an existing Web application as a reusable component.

The most recent version of meme media platform is extended as a complete Web-top system called Webble World [3–5]. Because the Webble system is implemented by using Microsoft Silverlight, it is available in major Web browsers only by installing the Silverlight plugin. The Webble World allows us to publish and combine media objects directly on the Web. The Webble is a short form of the Web Pebbles, where 'Pebble' is the abbreviation for 'Pad Enhanced Building Block Lifelike Entity'. In the Webble World, each media object has

[2] `http://angularjs.org/`

[3] `http://www.kplex.com/products/plexware.html`

a 2D vector graphic as its view, called ‘webble’. It is possible to create Web-based compound documents including dynamic graphic components represented as composite webbles. The Webble World brings in flexible representations of media objects.

## 2.2 Meme Media Architecture

Meme media objects wraps each knowledge resource in computers, such as text, image, tool, or services, as a two dimensional visual object. In IntelligentPad system, such a media object is called a ‘pad’. Each pad has a card-like view on the screen and a standard set of operations like ‘move’, ‘resize’, ‘copy’, ‘paste’, and ‘peel’. Users can easily copy a pad, paste a pad onto another, and peel a pad from a composite pad through mouse operations. A pad can be pasted onto another pad to define both a physical containment relationship and a functional linkage between them.

Figure 1 shows the user interface of a composite pad and its functional linkage structure. When pad $P3$ is pasted onto another pad $P1$, the pad $P3$ becomes a child of $P1$, and $P1$ becomes the parent of $P3$. No pad may have more than one parent pad. Pads can be pasted together to define various types of multimedia documents and application tools. Each pad provides a list of interfaces that maintain its state data, called ‘slots’. From users’ point of view, a slot works as a connection jack of a pad. Users can functionally connect each child pad to one of the slots of its parent pad. Only a single connection can be defined between a parent and its child.

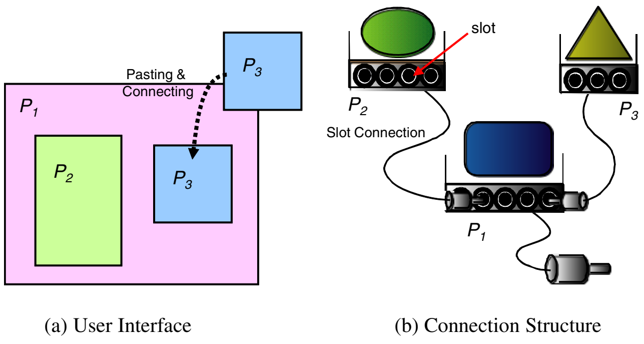


**Fig. 1.** A composite pad and its slot connection structure in IntelligentPad

In IntelligentPad, each pad consists of three basic components: model, view, and controller. Model-View-Controller pattern is originally from the Smalltalk-80’s GUI component architecture [9]. In the IntelligentPad, the structure of MVC is slightly different from the original structure (Fig. 2). There is no direct connection between the model and the controller. Here, a model is responsible for managing data and domain logics for operating the data. A view is working as output interface to the users. It shows users the representation of the data from the associated model. A controller is responsible for managing input from users. It detects users' input events (e.g. key typing or mouse button clicks) and sends a message to its view for changing the display or for modifying the date in the corresponding model. In modern GUI toolkits, controllers are often integrated into view parts (which is called visual objects). There, developers can assign event handlers to events published by visual objects for defining behaviors for responding users' inputs.

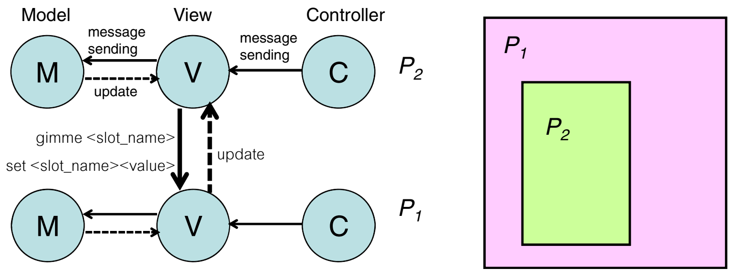


**Fig. 2.** Internal component architecture and message exchange between a parent and the child

### 2.3 Communication between Parent and Child

For communication between pads, IntelligentPad uses three standard messages: 'set', 'gimme', and 'update' (Fig. 2). In their default definitions, a 'set' message sends its parameter <value> to its recipient slot specified by the parameter <slot_name>, whereas a 'gimme' message requests a value from its recipient slot named by <slot_name>. An 'update' message notifies its state change to its child pads. Typically, the pad, which works as an input component to its parent, sends a 'set' message to its parent. When the message brings the state change of its parent, the parent pad sends 'update' messages to all of its children. Each child pad that has received an 'update' message sends a 'gimme' message to update its internal state using the slot value returned by the message. The functional linkage between pads is achieved by the message passing using above mentioned three standard messages and the update propagation of internal states according to the value transferred by the messages. These messages can be enabled or disabled by each pad.

## 3 MVC Frameworks in JavaScript

### 3.1 Problems in JavaScript Development

Mainly, to build a meme media platform, we have to implement functionalities described in section 2 by coding with JavaScript from scratch. Of course, it is possible. However, there are some common problematic points in JavaScript development.

For example, there is the compatibility problem among different Web browsers. The standardization process of HTML5 is not completed yet. In addition, Web browser vendors compete each other for getting more shares by implementing attractive new functionalities. As a result, depending on browsers, some functionalities are implemented in some browser, but others are not. HTML5 specification is changing this situation, but some differences among browsers still remain.

Moreover, the language specification of JavaScript itself has some difficulties for large-scale software development. JavaScript is a flexible, powerful programming language and easy to learn. People can quickly write a few JavaScript codes to set up simple interactivity on their Web sites. Nevertheless, because it is a weakly typed and interpreted language, its development process is difficult to be supported by Integrated Development Environments (IDEs). Nowadays, some powerful IDEs are produced for JavaScript development. However, it is still not so matured as on other strongly typed, compiled languages like Java or C++. Furthermore, there is different syntax to do the same things because of its flexibility.

### 3.2 An MVC Framework: AngularJS

To solve those problems, some software companies and JavaScript developers produce JavaScript libraries or frameworks. jQuery[4] is one of the most popular and powerful JavaScript libraries. It provides a uniform way and syntax to manipulate DOM structure and CSS properties. It is often used for building interactive user interface on Web sites. A lot of developers provide plugins of jQuery; therefor extensively its functionalities are expanded. However, it does not provide a uniform way to structure software like MVC, leaving it entirely to developers.

Nowadays, there exist many JavaScript MVC (or MV*) frameworks [10] such as Backbone[5], Ember.js[6], and Knockout[7]. We selected AngularJS as the basic framework of the meme media platform out of those many frameworks. AngularJS is developed by Google and has a large developer community. It supports composite views, data-binding between model and view, and custom vocabulary definition by detectives.

Figure 3 shows an overview of AngularJS's architecture and its composition structure of composite views. It looks totally different from the original MVC structure[8], but it has clear separation of roles of components. Here, models can be any JavaScript objects. Views are DOM fragments rendered in a browser window.

[4] `http://jquery.com/`
[5] `http://backbonejs.org/`
[6] `http://emberjs.com/`
[7] `http://knockoutjs.com/`
[8] Some people call it MVW (Model-View-Whatever) framework.

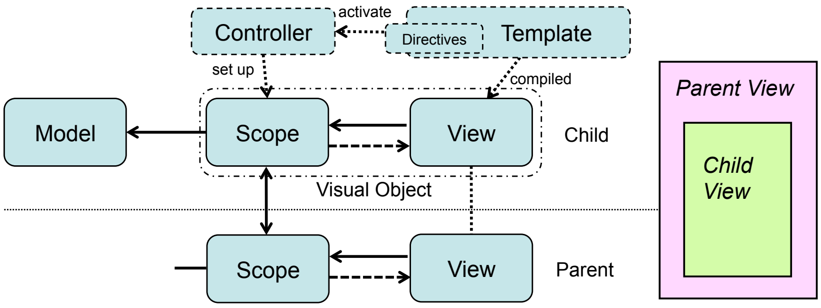


**Fig. 3.** The composition structure in AngularJS

AngularJS has a template system for reusing the same types of views. With directives, developers can declaratively specify data-bindings between views and scopes and set up controller association to views. Controllers are totally different from the original definition of the MVC. They work as glue to set up a linkage between models and scopes. Scopes manage the state of views, expose data form models to the corresponding view, and operate models according to the message form the view. Similar to the IntelligentPad, views and scopes have parent-child relationships corresponding to the containment structure of the views. When a child scope is created, it inherits properties from the parent scope. Through this mechanism, a child scope and its parent can access each other for some communication. Developers can define custom directives as element names, CSS classes, or element attributes to reuse user-defined functionalities or behaviors in a declarative way.

## 4 Implementation

We developed a prototypical meme media platform based on the AngularJS. We needed to implement the following functionalities:

1. views working as a wrapper of different contents,
2. the view composition mechanism by defining parent-child relationships,
3. the functional linkage through slot connection, and
4. the uniform user interface and operations of views.

We implemented each functionality by following the AngularJS's fashion. It makes it possible to integrate a lot of existing third-party components or directives developed with AngularJS seamlessly. The wrapper functionality was implemented by defining a special directive to specify the media object wrapper (1). Attaching this directive to an HTML element, the element works as a wrapper of its contents. For (2), we reused the Angular's parent-child relationships of scopes. The functional linkage between a parent and a child was implemented in the controllers and scopes (3). We defined a basic controller for defining common settings of a meme media object. The controller creates some properties and methods in the corresponding scope such as slot definitions, the data holder for visual properties, methods for managing standard three messages for communication, and behaviors for slot value changes. By defining a JavaScript object that is derived from the basic controller, developers can define a new media object that has extended functionalities and slots. For (4) we used jQuery UI[9] to implement move and paste by mouse operations. This was coded in a detective definition of the wrapper (1). The directive also contains other functions opening common dialogs for changing properties of a media object or setting up slot connection.

## 5 Application Example

Using the implemented platform, we created four media objects: text label object, text input object, image object, and image search object (Fig. 4). Each object is very simple and has only minimum functionalities.

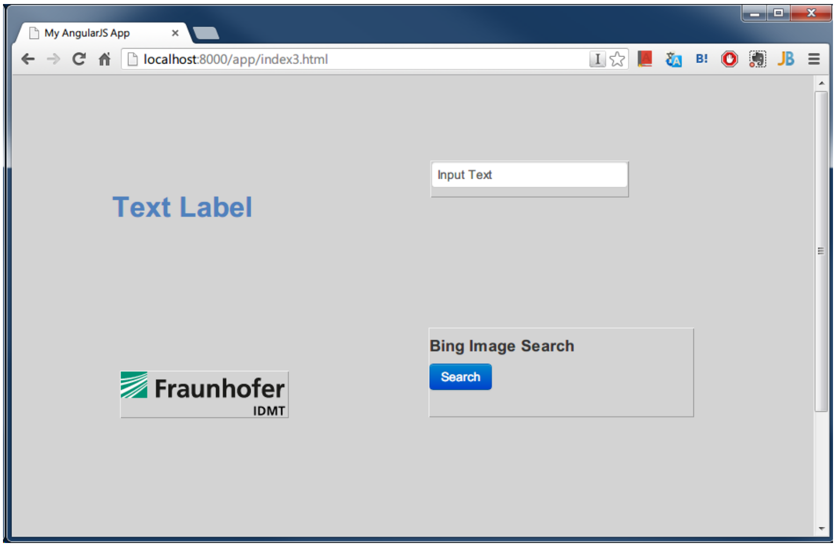


**Fig. 4.** Example meme media platform where three object are placed

The text label object and the text input object have only one slot `#value` (we represent a slot named 'slotname' as `#slotname`) which holds a text string displayed on each object. The image object has slot `#url` to hold an URL of a picture displayed on the object surface. The image search object is a wrapper component of the Bing Search API[10]. It has three slots: `#query`, `#title`, and

[9] `http://jqueryui.com/`

[10] `https://datamarket.azure.com/dataset/5BA839F1-12CE-4CCE-BF57-A49D98D29A44`

`#imageUrl`. When a user inputs some keywords to slot `#query`, the image search object retrieves the corresponding pictures using the Bing Search API and sets the title and the url of the first candidate of retrieved results to slot `#title` and slot `#imageUrl` respectively.

Each object is defined straightforwardly just by adding a few code for extended functionalities. For example, to define the image object, we needed the following JavaScript code to define new slot `#url`:

```
function ImageController($scope, $injector)
{
    $injector.invoke(WebbleController, this, {$scope: $scope});
    $scope.publicSlotList.push('url');
    $scope.slots.url = 'http://www.idmt.fraunhofer.de/.../idmt-logo.gif';
};
ImageController.prototype = Object.create(WebbleController.prototype);
```

The `ImageController` derives the basic setup from the `WebbleController`, sets `#url` as a public slot (which means an accessible slot form other objects), and defines the default value of the slot. The view of the image object can be defined declaratively:

```
<div class="webble" ng-controller="ImageController">
  <img ng-src="{{slots.url}}">
</div>
```

Here, the directive 'webble' is used as the CSS class of the `<div>` element, and the attribute directive `ng-controller` associates the controller of this media object with the `<div>`. Inside of the `<div>`, there is an `<img>` element used to display a picture. It has a directive `ng-src` to bind the image source to the value of slot `#url`.

Figure 5 shows an example composition of above three objects. With the drag-and-drop operation of objects, users can paste an object into another object. Here, the text input object, the label object, and the image object are pasted on the image object as its children. Moreover, these objects are connected to slot `#url`, `#title`, and `#imageUrl` of the image search object, respectively. The slot connections are customizable through the basic connection-setting dialog on demand. When a user enters a keyword into the text box, the image search object retrieves the corresponding picture and updates the value of slot `#title` and `#imageUrl`, finally the label object and the image object display the corresponding title and the picture.

We tested the developed platform with multiple Web browsers on multiple devices on demand. We used the latest stable version of Google Chrome, Firefox, Safari, and Opera both on Windows and Mac OS. As mobile devices, we used Google Chrome and Firefox both on Android and iOS. On all the Web browsers on Windows and Mac OS, the platform worked as intended. However, on the mobile devices, although the functionality of each object worked correctly, the drag-and-drop operation of object did not work. The reason is that, for

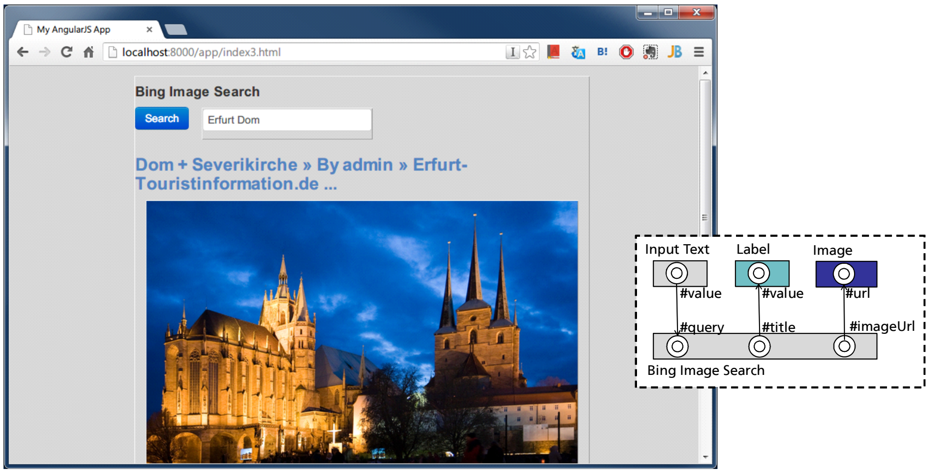


**Fig. 5.** A composite example and its slot connection structure

implementing the drag-and-drop operation, we used jQuery UI that does not support the mobile 'touch' events. We need to another library or another UI design to support in editing on mobile devices.

## 6 Conclusion and Future Work

In this paper, we described a prototypical implementation of a meme media platform with a modern JavaScript framework. AngularJS was used as the basic framework of the implementation. The framework supports in the development effectively. As a result, we could define new meme media objects with a few codes and in declarative manner. In the most of resent Web browser without mobile Web browsers, the platform works as intended. Users can combine the visual object to create new composite objects through direct manipulation.

This paper focused on the client-side editing functionality of the meme media. Therefore the platform is still missing lots of functionalities for exchanging composite objects among different browser windows over the Web. Such functionalities include save and load of composite objects, server-side repository of primitive and composite objects.

As the next step, we would like to consider how composite objects could be saved and loaded by specifying necessary information for replicating composite objects. In addition, we also improve user interface for better support of mobile devices.

Although the implementation uses AngularJS, other JavaScript MVC frameworks are also enough powerful and flexible for implementing a meme media platform. We would also like to investigate other JavaScript MVC frameworks and evaluate their pros and cons for meme media platform implementation.

## References

1. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press (2003)
2. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)
3. Kuwahara, M.N., Tanaka, Y.: Webble world – a web-based knowledge federation framework for programmable and customizable meme media objects. In: IET International Conference on Frontier Computing. Theory, Technologies and Applications, IET, pp. 372–377 (2010)
4. Kuwahara, M.N., Tanaka, Y.: Webbles: Programmable and customizable meme media objects in a knowledge federation framework environment on the web. In: Karabeg, D., Park, J. (eds.) Second International Workshop on Knowledge Federation, Dubrovnik, Croatia, October 3-6, vol. 822 (2010)
5. Kuwahara, M., Tanaka, Y.: Advanced "webble" application development directly in the browser by utilizing the full power of meme media customization and event management capabilities. In: Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2012, pp. 211–216. IEEE Computer Society, Washington, DC (2012)
6. Tanaka, Y., Imataki, T.: IntelligentPad: A hypermedia system allowing functional compositions of active media objects through direct manipulations. In: Proceedings of the IFIP 11th World Computer Congress, pp. 541–546 (1989)
7. Okada, Y., Tanaka, Y.: IntelligentBox: a constructive visual software development system for interactive 3d graphic applications. In: Proceedings of the Computer Animation, CA 1995, pp. 114–125. IEEE Computer Society, Washington, DC (1995)
8. Ito, K., Tanaka, Y.: A visual environment for dynamic web application composition. In: Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia, HYPERTEXT 2003, pp. 184–193. ACM, New York (2003)
9. Goldberg, A., Robson, D.: Smalltalk-80: the language and its implementation. Addison-Wesley Longman Publishing Co., Inc., Boston (1983)
10. Osmani, A.: Learning JavaScript Design Patterns. O'Reilly (2012)

# Direct Execution Learning Technology

Klaus P. Jantke

Fraunhofer IDMT, Children's Media Department, 99094 Erfurt, Germany
klaus.jantke@idmt.fraunhofer.de

**Abstract.** Technologies are pervading the daily life and are changing the way in which humans learn from school through higher education and vocational training to life-long learning. Peculiarities of the technology determine the potential advantages of technology enhanced learning and characterize the difficulties, pitfalls, and risks of particular approaches. Direct execution is a new paradigm emerging from the combination of the direct manipulation idea with contemporary meme media technology. Direct execution denotes the feature of media objects to be able to run automatically when manipulated by humans and when sticked together. Direct execution turns out to be an effective approach to exploratory learning even in abstract domains such as, for instance, recursion theory.

**Keywords:** e-learning, exploratory learning, meme media technology, direct manipulation, direct execution, instructional design, didactics.

## 1 From Memes by Meme Media to Direct Execution

Richard Dawkins has coined the term *meme* in his seminal book [8] to denote units of non-biological evolution. Susan Blackmore [7] is providing some folklike introduction into Dawkins' theory and its potential relevance to many domains. Yuzuru Tanaka [30] has taken the initiative to carry over Dawkins' approach to software technology where memes may be seen embedded in certain digital representations. Tanaka has coined the term *meme media*. Meme media objects considered to be digital containers, so to speak, of ideas when being in use may be subject to replication, to mutation, to cross-over and, finally, to fitness criteria.

This contribution deals with memes in technology enhanced learning and with the exploitation of meme media technology for technology enhanced learning. Emphasis is put on the way in which memes or, more precisely, meme media are manipulated within learning processes.

The paradigm of *direct manipulation* due to Ben Shneiderman [27] is known to be advantageous when dealing with involved problems digitally (also [28,11]). Direct manipulation means to treat objects on a computer screen just as these objects were what they are intended to represent. Some objects are operational. With an appropriate media technology at your fingertips, you may even make those objects run. For this refinement (or specialization) of direct manipulation, Jun Fujima and Klaus P. Jantke [9] have coined the term *direct execution*.

## 2 Towards Educational Memes

The present contribution deals with meme media technology, in general, for the purpose of computer supported education or, as some briefly put it, e-learning. Webble technology is of a particular interest, where the question for the underlying technological basis–be it Silverlight, HTML 5, or whatsoever–is left open. The crucial requirement is to have Webbles which allow for direct execution as introduced and exemplified in [9].

The driving motivation of the present work is an attempt to exploit the potential of memetics à la Dawkins with a certain focus on educational memes. Some way in which educational ideas or memes, so to speak, may be represented and encapsulated digitally is discussed in [12]. The key idea is that, anyway, didactic design needs some representation, preferably in a certain digital format. The technology of choice is storyboarding introduced by Jantke & Knauf in [15]. Storyboarding is the process of systematical, tool-supported anticipation of the human learners' relevant experiences. The technology allows for an incremental design process in which top-down and bottom-up development may be dovetailed according to the designers needs and, possibly, in dependence on other parties' contributions. The resulting storyboards may be represented in XML format. There is no need for any storyboard implementation–known to be an error-prone process like any other implementation of high level specifications–if there is a way of computational interpretation of storyboards as demonstrated in [6].

Conceptually, storyboards are hierarchically structured graphs adopted and adapted from dynamic planning (see [3] for one of the sources of inspiration).

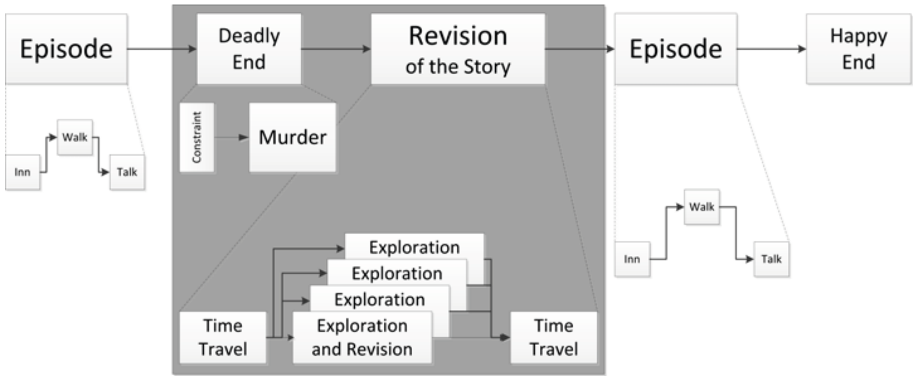


**Fig. 1.** Cutout of some storyboard for the game Shadows of Destiny (Konami, 2001)

The sample storyboard on display in figure 1 shows some very small part of a high level storyboard. The nodes are composite and may be refined by sub-graph replacements to specify their respective meanings in more detail as exemplified.

When playing the game, you frequently fall victim to a murderer, virtually. What follows is your chance to revise the story by virtually traveling backwards in time. In the virtual past, players have opportunities to change conditions such that the constraints of being murdered in the virtual present are no longer valid. The part of the storyboard highlighted by the underlying box indicates the key idea of the game which occurs frequently when playing.

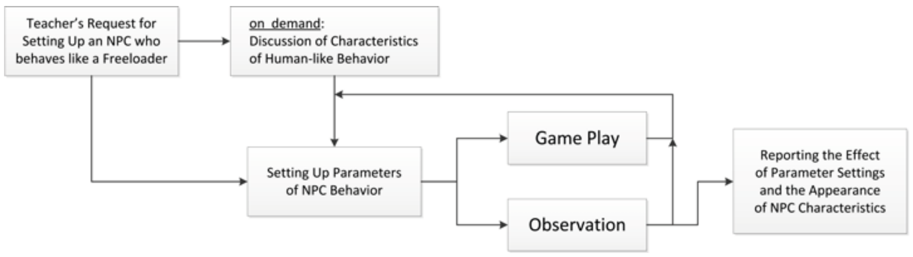

Teacher's Request for Setting Up an NPC who behaves like a Freeloader
on demand: Discussion of Characteristics of Human-like Behavior
Setting Up Parameters of NPC Behavior
Game Play
Observation
Reporting the Effect of Parameter Settings and the Appearance of NPC Characteristics

**Fig. 2.** Cutout of some storyboard for exploratory learning by playing a digital game

The sample storyboard on display in figure 2 is adapted from [16]. As in the example before, the nodes are composite and may be refined by sub-graph replacements. Designers may also decide to consider some node as atomic, i.e., not being subject to further refinement. In the example above, one might decide that 'game play' does not need any further specification.

There are branching points and loops. The first branching point indicates alternative offers, whereas the second one explicates some parallelism of actions.

Interested readers are invited to consult [15] for the basics of the present storyboarding approach and [6], [12], [16], [18], and [21], for instance, for varying application studies. The graph-theoretic background of the formalisms has been developed in much detail in [4] and [13] using the concept of pin graphs according to [10]. Hierarchically structured graphs define families of potential expansions which may be seen as formal languages of graphs. Involved results about the complexity of those languages [19] are not seen as negative. In contrast, they characterize the enormous expressiveness of the approach.

This paper does not aim at a comprehensive introduction into storyboarding. Instead, chunks of storyboards are used to wrap, so to speak, certain memes of game play, of didactics, of interaction design, and the like as proposed in [12].

For illustration, one may see the highlighted chunk of the storyboard of figure 1 as the representation of some meme of game design and game playing.

So, are digitally represented storyboards and constituents of storyboards meme media? Maybe they are. But in contrast to IntelligentPad, to IntelligentBox (see [30] for both), and to the contemporary Webble technology [22], they are much less operational. Those storyboards allow for direct manipulation [27], but are still far from direct execution [9].

This changes considerably, if storyboards are interpreted operationally [6]. Figure 3 illustrates some storyboard in use for training teams of executive staff in civil protection and disaster management (see [6] for background and details). The Excel file next to the storyboard contains text components assigned to particular nodes of the storyboard.

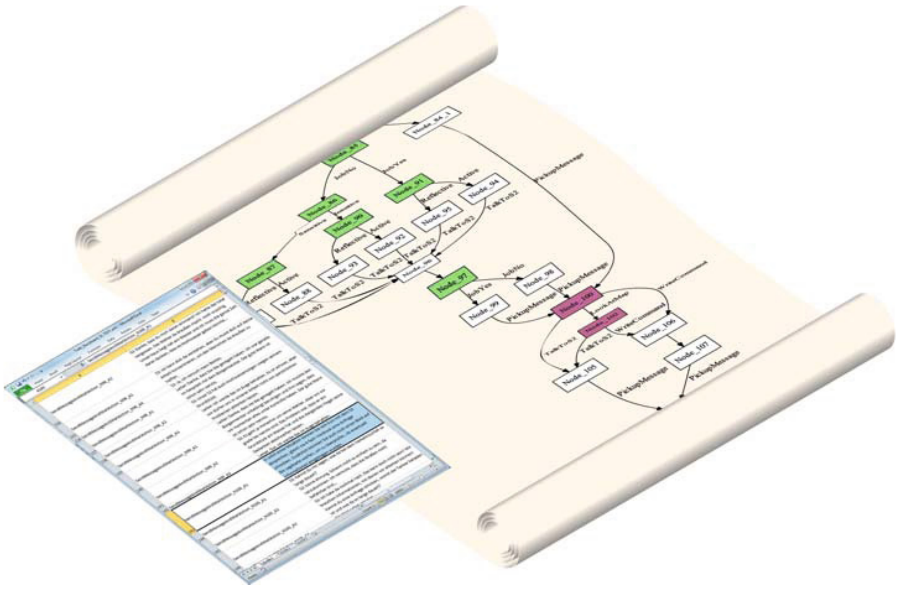


**Fig. 3.** Cutout of some application storyboard and related text in a relational database

One may imagine the work of the digital system based on this storyboard roughly as follows. The system is running on top of the storyboard beginning on some initial node. When the system enters some node, it is reading the semantics of the node and performing an action accordingly. If the node has a pointer into the relational database on display next to the storyboard, the system is reading the content unit to present it to the human user. If the node is a branching point, the system presents options of choice such as different buttons, e.g., to the user. According to the storyboard structure and to the user interaction, the system traverses the storyboard from one node to the other.

In some sense, the underlying storyboard is directly executed by the system–be it a digital game, an e-learning system, or whatever–running on top.

As demonstrated by Knauf et al. [21], substructures of such a storyboard may contain or represent didactic knowledge. Having the storyboard separated, to some extent, from the digital system interpreting it, allows for modifications. Running on some modification means the direct execution of educational memes.

## 3 Direct Execution for Making the Abstract Concrete

There are paramount potentials of the deployment of modern technologies for new ways of learning. Nowadays, information and communication technologies are often seen as a basis for replicating concrete learning targets in virtual worlds to allow for risk-free frequently repeated inspections and explorations. In a sense, the concrete real is represented by means of the concrete virtual. Manipulations of the virtual are expected to contribute to a better understanding of the real, perhaps, even to mastering the real.

By means of digital media technologies, humans are able to learn and/or train concrete knowledge and skills, respectively, if the digital environment allows for really doing it. You may really learn manage scarce resources by doing so in a virtual world, you may also train your language reading and writing skills by doing so digitally, and with some more advanced technologies you may also train your pronunciation by really talking to the digital system. But you will never learn to swim, to ride a bike, or to sail a boat by manipulating a digital system. There is the key question for the relation between the real and the virtual [17].

But what about the abstract? Every direct manipulation, in general, and every direct execution, in particular, is concrete.

It is an indication of the power and the reach of Webble technology for technology enhanced learning that even extremely abstract learning contents may be mastered by means of some appropriate memetic implementation [5].

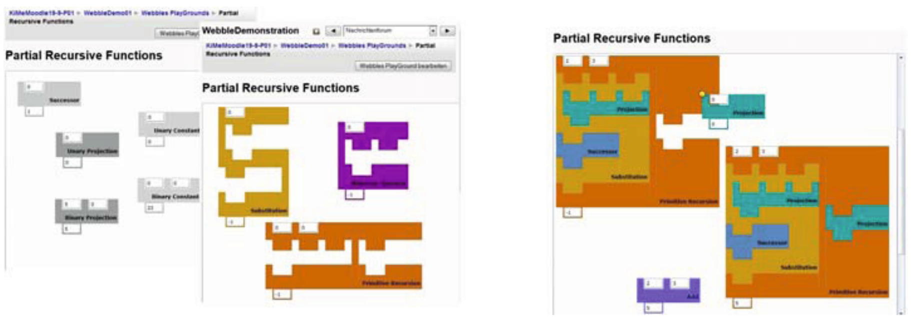


**Fig. 4.** Basic functions and operations of the theory of effective computability (left) and some primitive recursive definition immediately before and after completion (right)

One of the difficulties of studying highly abstract topics is that pondering involved problems usually requires lengthy and intriguing thought operations. There is not much you can touch play with. The theory of effective computability is such a domain [20], [26].

As illustrated by means of figure 4, Webbles allow for playful exploration of computability. In the screenshot on the right, there is some construction not yet completed. As soon as the missing Webble is inserted, it does some calculation.

## 4 Direct Execution for Exploratory Games

In contrast to the other sections of this paper, the present section goes into some more detail of, so to speak, foreign content. In practice it is known that it is difficult to discuss details of didactics without some understanding of the learning contents and goals. For the particular idea of learning certain algorithms and variations of them in a playful manner, having some clue of the underlying algorithmic ideas is inevitable for pondering didactic concepts.

### 4.1 The Problem Domain of Pattern Inference

Dana Angluin [2] introduced her seminal pattern concept concisely, formalized an unprecedentedly clear concept of learning patterns, and developed learning algorithms including deeper insights into the algorithmic complexity of learning patterns. Because patterns in many application domains such as research into media impact, in general, and into the impact of game playing, in particular, are more expressive and of higher complexity than Dana Angluin's concepts, her results may serve, so to speak, as lower estimates of the problems one is facing in those application domains. Saying it laxly, what is complex in Dana Angluin's setting is at least of the same complexity in those applications, and what does not work in Dana Angluin's setting will never work in those applications.

Assume any finite, not empty alphabet $M$ and any set of variables $X$, where it is practically relevant that both are disjoint, i.e. $M \cap X = \emptyset$. For readability, we choose $X = \{x_1, x_2, x_3, \ldots\}$. For any set $A$, $A^+$ denotes the set of all finite, but not empty strings f symbols from $A$. $P = (M \cup X)^+$ is the set of patterns under consideration. Every pattern $p \in P$ allows for generating string of $M^+$ by substituting for each variable in $p$ some string from $M^+$. Seen in its right perspective, every pattern $p$ is a grammar of some formal language named $L(p)$.

For investigations into learning patterns from examples, some notions and notations are usefull.

If $s$ and $t$ are two strings or patterns, the notation $s \sqsubseteq t$ indicates the $s$ is an initial segment of $t$, not necessarily a proper one. The notation $s \subseteq t$ means that $s$ occurs as a substring of $t$. More formally, $s \sqsubseteq t$ means $\exists s' \in M^* : s\, s' = t$. Similarly, $s \subseteq t$ means $\exists s', s'' \in M^* : s' s\, s'' = t$.

For two given patterns $p$ and $q$, it is helpful to be able to describe their comparable expressiveness. $p \preceq q$ means that $q$ is similarly expressive or more expressive than $p$ which means in formal terms $L(p) \subseteq L(q)$. Accordingly, $p \prec q$ means $L(p) \subset L(q)$.

The latter terminology is useful, for instance, to specify what it means that some pattern $p$ is a best fit to some sample set $S \subseteq M^+$ formalized as follows: $S \subseteq L(p) \;\wedge\; \neg\exists q : S \subseteq L(q) \subset L(p)$. Dana Angluin [2] calls a pattern which fits some sample best being *descriptive of the sample.* Note that patterns may be seen as generalizations of sets of strings. Thus, for every sample $S \subseteq M^+$, any pattern $p$ with $S \subseteq L(p)$ is a generalization. If $p$ is descriptive of $S$, it turns out to be some *least general generalization.*

## 4.2 Pattern Inference: The Problem and Its Solutions

Patterns are usually understood as descriptions of something of general interest, some structure, some behavior, or the like. Those patterns do not show directly. What is perceived are the instances of some pattern. The key application problem is to reconstruct some unknown pattern from the instances observed. In formal terms, some pattern $p$ shall be learned from the sequence of all strings in $L(p)$. Even more formally, assume $s_1, s_2, s_3, s_4, \ldots$ to be any ordering of the elements of some formal language $L(p)$. There is the question for an algorithm powerful enough to solve this learning problem for any target pattern–one that learns all. Learning is some process which expands over time. Any learning algorithm $\mathcal{A}$, in processing some sample $S_n = \{s_1, \ldots, s_n\}$ of observed instances, generates a certain hypothetical pattern $\mathcal{A}(S_n)$ shortly named $p_n$. With information growing over time, the learning algorithm generates some sequence $p_1$, $p_2$, $p_3$, $p_4$ $\ldots$ of pattern hypotheses. It is said that $\mathcal{A}$ does solve the current learning problem successfully, exactly if at some point $k$ in time, the hypothesis $p_k$ is correct, i.e. $L(p) = L(p_k)$, and this hypothesis is kept even if further instances are observed.

**Dana Angluin's Classical Algorithm.** Dana Angluin [2] has provided the first solution of the pattern inference problem under consideration. Her approach is intuitive and easy to understand. Even to prove its correctness is quite easy.

It is helpful to introduce the notion of consistence. Given any sample set $S \subseteq M^+$, a pattern $p$ is said to be consistent with $S$, if and only if $p$ is able to generate all strings of $S$, i.e. $S \subseteq L(p)$. Recall that any consistent pattern which fits best is said to be descriptive.

Dana Angluin's idea summarized in common speech is roughly as follows. Given some current pattern hypothesis $p_n$ and some current instance $s_{n+1}$, check whether or not the hypothesis generated before is consistent with the new data. If so, keep the hypothesis. Otherwise, compute any pattern descriptive of all instances seen so far. Based in this astonishingly clear idea, one is able to learn any pattern from its instances.

In more formal terms, the sequence of hypotheses generated by $\mathcal{A}$ on some sequence of instances $s_1, s_2, \ldots$ is defined as follows, where $S_n$ means $\{s_1, \ldots, s_n\}$.

$$p_1 = s_1$$

$$p_{n+1} = \begin{cases} p_n & \text{if} \quad s_{n+1} \in L(p_n) & \text{[DA i]} \\ descr(S_{n+1}) & \text{if} \quad s_{n+1} \notin L(p_n) & \text{[DA ii]} \end{cases}$$

The term *descr* denotes some procedure computing a descriptive pattern of the argument sample. The paper [2] contains a proof of the algorithm's correctness. Dana Angluin has also provided an algorithmic principle for implementing *descr*.

$$D = \{\, p \mid S \subseteq L(p) \,\}$$

The set of all patterns consistent with some sample is finite up to renaming of variables. With the requirement that variables are canonically numbered from left to right, $D$ becomes finite.

$$D = \{\, p \mid S \subseteq L(p) \ \wedge \ p \text{ canonical} \ \wedge \ p \text{ of maximal length} \,\}$$

Every pattern in $D$ which is minimal w.r.t. $\preceq$ is descriptive of $S$. In this way, the procedure *descr* is computable. The algorithmic learning problem is solved.

Unfortunately, both the problem of computing *descr* and the problem of deciding consistency are NP-hard [2]. The crux is that the decision problem "$s \in L(p)$ ?" for any string $s$ and any pattern $p$ is even NP-complete. There is an ingenious proof of NP-completeness by reduction of the SAT problem in [2].

**The Algorithm of Lange and Wiehagen.** Rolf Wiehagen when pondering the question for *best examples* when learning from positive data has pointed out that for learning patterns it is sufficient to process only the shortest examples. This insight lead to some algorithm published in [23] which will be subsequently briefly called the Lange/Wiehagen algorithm.

The Lange/Wiehagen algorithm solves the problem of learning patterns in polynomial time. This surprising result is possible by giving up consistency– not every intermediate hypothesis is able to generate the data on which it has been generated. Although some of the intermediate hypotheses output by the Lange/Wiehagen algorithm appear useless and do not reflect much of the process of learning so far, the process as a whole always converges to a correct solution.

$\mathcal{A}'$ is used to denote the Lange/Wiehagen algorithm. Its subsequent outputs on some input sequence $s_1, s_2, \ldots$ are named $p_1, p_2, \ldots$ as before.

$$p_1 = s_1$$

$$p_{n+1} = \begin{cases} p_n & \textit{if} \quad |p_n| < |s_{n+1}| & \text{[L/W i]} \\ s_{n+1} & \textit{if} \quad |p_n| > |s_{n+1}| & \text{[L/W ii]} \\ lgg(p_n, s_{n+1}) & \textit{if} \quad |p_n| = |s_{n+1}| & \text{[L/W iii]} \end{cases}$$

The procedure $lgg$ computes some least general generalization of the current hypothesis $p_n$ and the presented new instance $s_{n+1}$.

This computation is quite easy, because both inputs are of the same length. For specifying $lgg$, it is helpful to assume two local variables, i.e. internal short term memories. These are some counter $c$ to hold the value of the currently used largest variable index and some storage of temporary variable assignments *VA*. The initial settings are $c = 0$ and $VA = \emptyset$.

If $l$ denotes the length of $p_n$ and $s_{n+1}$, one may write both strings symbol by symbol as $p_n = p_n(1) \ldots p_n(l)$ and $s_{n+1} = s_{n+1}(1) \ldots s_{n+1}(l)$, respectively. There will be constructed some output string $q$ symbol by symbol, accordingly.

```
(1) for $i = 1$ to $i = l$ do
  (1.1) if $p_n(i) = s_{n+1}(i) \in M$ then $q(i) := p_n(i)$;
        if $i = l$ then goto (2)
  (1.2) if $p_n(i) \neq s_{n+1}(i)$ then
    (1.2.1) if $\exists j : [p_n(i), s_{n+1}(i), j] \in VA$ then $q(i) := x_j$
    (1.2.2) else $c := c + 1$; $q(i) := x_c$ $VA := VA \cup \{ [p_n(i), s_{n+1}(i), c] \}$;
        if $i = l$ then goto (2)
(2) return $q = q(1) \ldots q(l)$
```

In contrast to the NP-hardness of the computations in [DA i] and [DA ii], the test [L/W i] and [L/W ii] are of linear complexity, and the computation of [L/W iii] is of quadratic complexity. Consistence is lost in [L/W i] and [L/W ii].

**Some Refinement by Jantke and Grieser.** The Lange/Wiehagen algorithm has been discussed from the viewpoint of knowledge efficiency in [14]. Within the process of learning, there are two out of three cases in which the algorithm is loosing information. First, when some string longer than the current hypothesis does occur–case [L/W i] in the description of the algorithm above–it is ignored. Second, when some string shorter than the current hypothesis does occur–case [L/W ii] indicated above–the hypothesis kept so far is given up and the shorter string is adopted to form the new hypothesis. Certain refinements seem possible.

In the first case [L/W i], checking whether or not the longer string carries useful information is costly, because it means checking membership. Does the current pattern allow for generating this string or not? This decision problem is known to be NP-hard and, thus, the task of refinement is not taken into account.

In the second case [L/W ii], all information collected so far in the latest hypothesis is thrown away when the newly appearing shorter string takes the place of the hypothesis. Isn't there any way to preserve at least some of the information in the pattern? A refinement proposed by Jantke and Grieser [14] is considered for the purpose of the present investigation.

Assume that the Lange/Wiehagen algorithm as described above is currently keeping some hypothetical pattern $p$ of length $\mid p \mid = k$. Furthermore, assume that some new string $s$ of length $\mid s \mid = l$ with $l < k$ is presented to the learning algorithm. Instead of throwing $p$ away and adopting $s$ as it is, the refinement is to calculate the new hypothesis as some least general generalization of $p$ and $s$.

$$\begin{array}{lcl} p_1 & = & s_1 \\ p_{n+1} & = & \begin{cases} p_n & \textit{if} \quad |p_n| < |s_{n+1}| \qquad\qquad [\text{J/G i}] \\ lgg(p_n, s_{n+1}) & \textit{if} \quad |p_n| \geq |s_{n+1}| \qquad\qquad [\text{J/G ii}] \end{cases} \end{array}$$

At a first glance, this description seems to be more rough–instead of being more fine–than the algorithm $\mathcal{A}'$ above, because it has only two many cases, whereas the other one has three. However, case [L/W ii] is really refined.

Conceptually, one may think of this refinement as follows. First, the pattern $x_1 \ldots x_l$ is taken as a possible generalization, because it definitely generalizes both $p_n$ and $s_{n+1}$. Next, traversing the pattern $x_1 \ldots x_l$ symbol by symbol from the left to the right it is specialized.

The algorithm is not of high computational complexity, but a bit cumbersome due to a larger number of distinct cases and their dovetailing.

To warm up, some intuitive example is presented in which the references on the right refer to the algorithm to be presented below. The following figure is illustrating the annotation process for the pattern $p = ax_1x_2x_3bx_4ccax_1x_2x_3x_4c$ of length $k = 14$ and the string $abacac$ with $l = 6$. The goal is to generalize the pattern $p$ by some other pattern of length $l = 6$. The algorithm runs in turns. In every turn, certain identical substrings are found and indexed.

As a result of the procedure illustrated by means of figure 5, the pattern is partitioned into $l = 6$ substrings. The partitioning represented by the last line of the figure above is interpreted as the intermediate pattern $x_1x_2x_3x_4x_1x_5$ which generalizes $p$.

| pattern $p$ | $a$ | $x_1$ | $x_2$ | $x_3$ | $b$ | $x_4$ | $c$ | $c$ | $a$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $c$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1.1 | 1.1 | 1.1 | 1.1 | | | | | 1.2 | 1.2 | 1.2 | 1.2 | | | by rule 3b |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | | | | 1.2 | 1.2 | 1.2 | 1.2 | | | by rule 3a |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | 3.1 | 3.1 | | 1.2 | 1.2 | 1.2 | 1.2 | 3.2 | 3.2 | by rule 3b |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | 3.1 | 3.1 | 4.1 | 1.2 | 1.2 | 1.2 | 1.2 | 3.2 | 3.2 | by rule 1 |

**Fig. 5.** Illustration of Jantke/Grieser annotations for the Lange/Wiehagen algorithm

Subsequently, the pattern is put into relation to the string $s$, because until this points only the string's length has been taken into account.

In every row of figure 5 there are introduced some annotations. The process terminates as soon as every symbol received an annotation. The particular annotation $u.v$ means that some variable $x_u$ is introduced and the symbols annotated by $u.v$ form some string for the $v$th occurring substitution of $x_u$.

The example shown in figure 5 above results in the pattern $x_1x_2x_3x_4x_1x_3$. Annotations introduced on some line above are dimmed out to stress the recently introduced annotations.

Every turn of the annotation algorithm is defining some partition of the original pattern. Each partition is a refinement of the preceding one by increasing the number of sections or be annotating at least one previously not annotated section. At the end of the run, the total number of sections equals $l$, the length of the given input string, and all sections are annotated.

In figure 5 above, the first row is showing 4 sections after the first turn, two of them with annotations. The second row shows 5 sections, 3 with annotations. The third row shows 6 sections, 5 with annotations. Finally in row 4, there are 6 sections, all with annotations.

The following figure 6 for $p = ax_1bx_2bax_2ax_1bx_2ax_1a$ and $s = aaaaaaa$ with $l = 7$ is used to illustrate some different phenomenon.

| pattern $p$ | $a$ | $x_1$ | $b$ | $x_2$ | $b$ | $a$ | $x_2$ | $a$ | $x_1$ | $b$ | $x_2$ | $a$ | $x_1$ | $a$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1.1 | 1.1 | 1.1 | 1.1 | | | | 1.2 | 1.2 | 1.2 | 1.2 | | | | by rule 3b |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | | | 1.2 | 1.2 | 1.2 | 1.2 | | | | by rule 3a |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | 3.1 | | 1.2 | 1.2 | 1.2 | 1.2 | 3.2 | | | by rule 3(b)i |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | 3.1 | 4.1 | 1.2 | 1.2 | 1.2 | 1.2 | 3.2 | | | by rule 2a |
| | 1.1 | 1.1 | 1.1 | 1.1 | 2.1 | 3.1 | 4.1 | 1.2 | 1.2 | 1.2 | 1.2 | 3.2 | 5.1 | 5.1 | by rule 1 |

**Fig. 6.** Illustration of Jantke/Grieser annotations for the Lange/Wiehagen algorithm showing some particular phenomenon in stage 3 of the turn-based annotation process

In the third round of the annotation process, the one-symbol substring $p(6)$ of the pattern $p$ is selected for annotation because it occurs repeatedly at later positions of $p$. Although it holds $p(6) = p(12)$ and $p(6) = p(14)$, only substring $p(12)$ gets an annotation, but not $p(14)$. The reason is that an annotation of $p(14)$ by 3.3 would separate the pattern into 8 sections exceeding the limit $l = 7$.

It follows a description of the annotation algorithm which works stage by stage beginning on stage 1. The input parameters are any $p \in P$ with $|p| = k$ and any $s \in M^+$ with $|s| = l$ and $l < k$.

There are at most $l$ turns. The current stage number is denoted by $n$. At the beginning of turn $n$, there is a certain number $an_n$ of already annotated sections of the pattern $p$ and a certain number $un_n$ of not yet annotated sections with $an_n + un_n \leq l$.

1. If $an_n = l - 1$ and $un_n = 1$, then $p$ has the structure $p = p_1^a p^u p_2^a$ where $p_1^a, p_2^a \in (M \cup X)^*$, $p^u \in (M \cup X)^+$, and exactly the symbols of $p^u$ do not yet carry any annotation.
   Annotate exactly all symbols of $p^u$ by $n.1$; $an_{n+1} := l$; $un_{n+1} := 0$; exit the procedure.
2. If $an_n = l - 2$ and $un_n = 2$, then $p$ has the structure $p = p_1^a p_1^u p_2^a p_2^u p_3^a$ where $p_1^a, p_2^a, p_3^a \in (M \cup X)^*$, $p_1^u, p_2^u \in (M \cup X)^+$, and exactly the symbols of $p_1^u$ and $p_2^u$ do not yet carry any annotation.
   (a) If $p_1^u = p_2^u$, then annotate exactly all symbols of $p_1^u$ by $n.1$ and all symbols of $p_2^u$ by $n.2$; $an_{n+1} := l$; $un_{n+1} := 0$; exit the procedure.
   (b) If $p_1^u \neq p_2^u$, then annotate exactly all symbols of $p_1^u$ by $n.1$; $an_{n+1} := l-1$; $un_{n+1} := 1$.
3. Otherwise, $an_n + un_n < l$ and $un_n > 1$. In this case, the pattern $p$ has an initial segment $p_1^a p^u p_2^a \prec p$ where $p_1^a, p_2^a \in (M \cup X)^*$, $p^u \in (M \cup X)^+$, and exactly the symbols of $p^u$ do not yet carry any annotation.
   Look for some longest initial segment $r \preceq p^u$ which occurs repeatedly in $p^u$ or in any other following not yet annotated sections (occurrences should be non-overlapping).
   (a) If there is no such $r$ which occurs at least twice, take the first letter $z$ of $p^u$, i.e., $z \in M \cup X$ and $z \preceq p^u$, and annotate it by $n.1$;
   $an_{n+1} := an_n + 1$; if $z = p^u$, then $un_{n+1} := un_n - 1$ else $un_{n+1} := un_n$.
   (b) If $r$ is found, look for a maximal number $f$ of non-overlapping occurrences in not yet annotated parts such that the following conditions are met, where $f$ denotes the resulting number of newly annotated sections and $g$ denotes the resulting number of not yet annotated sections left over.
      i. The value $g$ does not exceed the value $l - an_n - f$.
      ii. The value $l - an_n - f$ does not exceed the remaining number of symbols without any annotation.

   If the conditions are valid, annotate these $f$ sections by $n.1$ to $n.f$, resp., and set $an_{n+1} := an_n + f$ and $un_{n+1} := g$, accordingly.

As a result, the whole pattern $p$ is separated into sections being annotated by terms of the form $u.v$, where $u$ is not exceeding $l$. The ultimately resulting number of sections is exactly $l$.

The resulting annotation is interpreted as some pattern, where each section annotated by some $u.v$ is understood to represent the variable $x_u$. For instance, in fig. 5 this pattern is $x_1x_2x_3x_4x_1x_3$. In figure 6, the pattern is $x_1x_2x_3x_4x_1x_3x_5$. This pattern does always generalize $p$, but does not necessarily generalize $s$.

To see this more explicitly, let us have a closer look at the example of figure 5 where the string of length $l = 6$ is *abacac* and the pattern constructed so far is $x_1x_2x_3x_4x_1x_3$. The string differs in its 3rd and its 6th position, but the pattern does not. Hence, $abacac \notin L(x_1x_2x_3x_4x_1x_3)$.

The variable $x_3$ appearing in two different positions has to be split into $x_3$ and $x_5$ to match $s$. The resulting pattern is $x_1x_2x_3x_4x_1x_5$. To fit as close as possible, this pattern is specialized to $x_1cx_3bx_1x_5$ and finally normalized to $x_1cx_2bx_1x_3$. Missing algorithmic details are described below.

To sum up, the Lange/Wiehagen algorithm with Jantke/Grieser refinement shortly named $\mathcal{A}''$ returns on the input pattern $p = ax_1x_2x_3cx_4bbax_1x_2x_3x_4b$ and the input string *acabac* the new hypothesis $x_1cx_2bx_1x_3$.

The procedure performed after annotation of the input pattern is as follows. Note that there are exactly $l$ substrings, where $l$ is the length of the input string $s$. Consequently, every substring corresponds to a uniquely defined position in $s$.

1. Take the pattern $p$ and replace every string annotated by some number $i$ simply by the one variable $x_i$.
2. For every $x_i$ representing a substring of length 1, i.e. some letter $z$, check whether at all the corresponding position in $s$, the same letter $z$ does occur. If so, substitute $x_i$ by $z$.
3. For every $x_i$, look at all occurrences and check whether or not all correspondig letters in $s$ are identical. If not, split the variable until all occurrences of the same variable relate to identical letters in $s$.
4. Rename variables from left to right to get a canonical pattern.

Step 2 is specialization and step 3 is generalization. The order of both may be changed, i.e. there is some inherent nondeterminism. This completes $\mathcal{A}''$.

Seen from a bird's eye view, the refined algorithm $\mathcal{A}''$ in case [J/Gii] does the following: (i) generalizing $p$ by some pattern of the length of $s$, (ii) generalizing to match string $s$ as well, (iii) specializing to fit string $s$ as close as possible, and (iv) normalizing.

At this point of discussion, the reader knows a bit about patterns à la Angluin, about the pattern learning problem, about its computational complexity, and about the algorithms and algorithmic variants, respectively, named $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{A}''$.

There is no doubt about the relevance of formulas, flowcharts and other formal descriptions for specifying algorithms in depth and detail. In particular, when variants of algorithms are distinguished by subtle details, nothing competes formalisms. Nevertheless, the mastery of algorithms means more than knowing about details. It covers some feeling for the algorithms' behavior, some experience, and some familiarity as well. To gain this requires much doing.

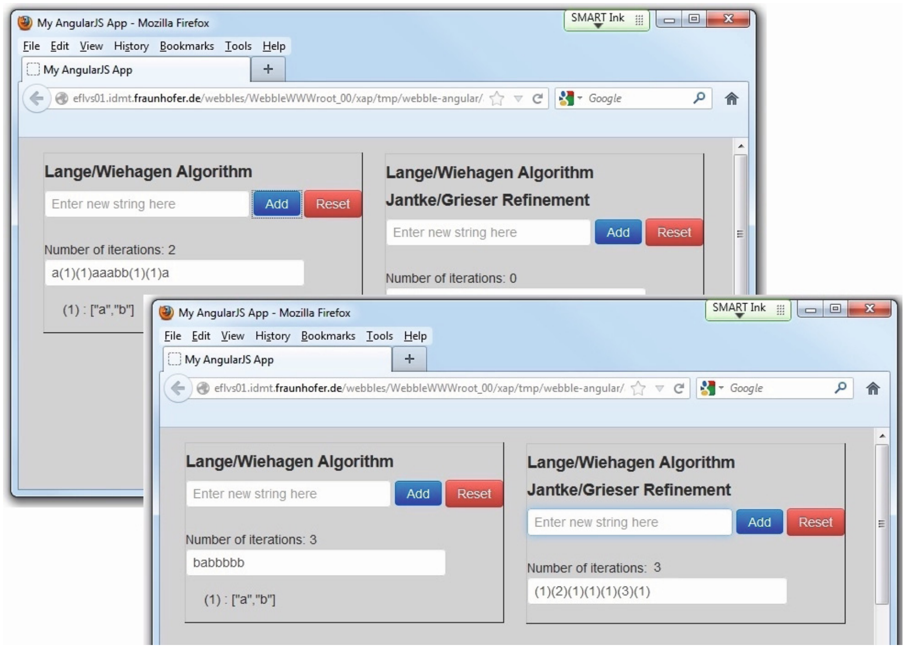
By means of the following subsection 4.3, we return to the issue of technology enhanced learning. The exclusive purpose of the lengthy and rather detailed investigation above into some particular learning domain is to firmly lay some cornerstone for the discussion of learning with emphasis on direct execution and on ideas for playful exploration.

### 4.3 Exploratory Learning by Playing with Meme Media

When the target of learning is not just one algorithm, but several alternatives, like in fields such as searching and sorting (think only of the fine difference between *Heapsort* and *Bottom-Up Heapsort*), there are varying opportunities for didactic concepts based on the competitiveness of the algorithms.

For teaching pattern inference, in general, and aiming at an understanding of the Lange/Wiehagen algorithm $\mathcal{A}'$ and variations such as $\mathcal{A}''$, in particular, there are many ways of making the studies exciting.

Here is one task teachers may assign to their students: *Find some sequence of strings as short and simple as possible such that after processing this sequence the Lange/Wiehagen algorithm returns a string without variables, whereas the Jantke/Grieser refinement leads to some hypothesis which consists of variables, exclusively.*


My AngularJS App - Mozilla Firefox
SMART Ink
File Edit View History Bookmarks Tools Help
My AngularJS App
eflvs01.idmt.fraunhofer.de/webbles/WebbleWWWroot_00/xap/tmp/webble-angular/
Google
Lange/Wiehagen Algorithm
Enter new string here
Add
Reset
Lange/Wiehagen Algorithm
Jantke/Grieser Refinement
Enter new string here
Add
Reset

Number of iterations: 2
a(1)(1)aaabb(1)(1)a
(1) : ["a","b"]

Number of iterations: 0

Number of iterations: 3
babbbbb
(1) : ["a","b"]

Number of iterations: 3
(1)(2)(1)(1)(1)(3)(1)

**Fig. 7.** Webbles implemented by Jun Fujima (Fraunhofer IDMT, Erfurt) in HTML 5 for the playful exploration of the Lange/Wiehagen algorithm and some of its variations

This example problem is not that easy to solve. Some students need several approaches before they arrive at a solution. The screenshots of figure 7 show the results of processing the three strings *abbaaabbbba*, *aaaaaabbaaa*, and *babbbbb*. The two alternative results are *babbbbb* and $x_1x_2x_1x_1x_1x_3x_1$.

There is a large variety of similar competitive exercises. Just for illustration, ask for input data on which $\mathcal{A}'$ returns some constant string hypothesis from $M^+$, whereas $\mathcal{A}''$ returns some pattern with exactly one variable. This exercise may be easily modified by asking for two, three, or more variables and, perhaps, by putting constraints on the number of repeated occurrences of one variable. Note that all the examples mentioned so far refer just to the comparison of two results generated by $\mathcal{A}'$ and by $\mathcal{A}''$, respectively. One extension may take $\mathcal{A}$ into account asking for input data on which $\mathcal{A}$ and $\mathcal{A}''$ generate identical results or, alternatively, return different hypotheses. Extra constraints as above may be put on the number of variables and on the frequency of those variable occurrences. Other extensions result from considering sequences of hypotheses, possibly until reaching the final learning goal. For instance, one may ask for some target pattern $p$ and investigate sequences of strings from $L(p)$ such that certain phenomena may be recognized such as, e.g., $\mathcal{A}''$ converging much faster than $\mathcal{A}$. Such a difference in behavior may be expressed quantitatively.

One of the highlights of didactics would surely be to encourage students to design attractive competitive exercises to their fellow students. Those exercises should not just be tasks assigned to other students, but should be investigated and discussed for gaining insights into the key ideas behind and for their didactic potentials.

Direct execution is essential to those exercises. Learners should be encouraged to try out larger numbers of data sets.

Skeptical readers might counter that any collection of programs implementing the algorithms under consideration would do as well. Although this is basically true, it is missing the point. Meme media implementations allow for a much larger variation of experiments including the decomposition of algorithms under consideration. This aspect has not been demonstrated by means of the pattern inference algorithms, but may be clearly seen in figure 4, right screenshot, where some algorithm showing some hole, so to speak, may be completed by fixing the hole in the one or the other way.

To play with the Lange/Wiehagen algorithm and some of its variations has been chosen just as a simple case study of exploratory learning about algorithms using Webbles implementations which allow for direct execution. Even within the limits of this elementary case study, there are overwhelmingly many options of formulating playful exercises.

Just for illustration, one may think of short multi-player games as follows. Some game master is secretly choosing a pattern $p$ and is presenting a few longer inputs (two, at least) from $L(p)$ to the Lange/Wiehagen algorithm which, in turn, immediately responds with a certain hypothesis. Now, it is the players' turn simultaneously. Everyone has the right to guess further string from $L(p)$. For correct guesses of strings that really belong to $L(p)$, they score some points. Turn by turn, the players proceed toward identifying the hidden target pattern. Feeding the right string into the algoritm, be it $\mathcal{A}'$, $\mathcal{A}''$, or any other variant, such that the target pattern is found is rewarded especially and the game ends. Note that, naturally, the game master may be computerized.

## 5 Direct Execution in Educational Patterns

Since the early days of didactics, engaged human teachers are striving hard to distill essential ideas frequently reusable for efficient and effective teaching–educational memes as discussed in [12].

In scientific discourse, following the early work of Christopher Alexander [1], repeatedly occurring structures are called patterns. To be more precise, one should the concrete phenomena that occur *instances* and reserve the word *pattern* to the general some instances have in common. According to the scientific background of scientists involved and corresponding to the traditions and to the methodologies of pedagogy, educational patterns are usually very vague and uncertain [25]. The meme media idea may help to encapsulate and, thus, to clarify several patterns of didactics.

Within the present contribution, emphasis is put on those patterns which relate to the paradigm of direct execution. Let us begin with a pattern named *challenging the target* in [12]. This educational pattern is applicable to some settings of exploratory learning where the learning target is something being operational such as, e.g., an algorithm. Those settings appear overwhelmingly in studies of computer science, for instance.

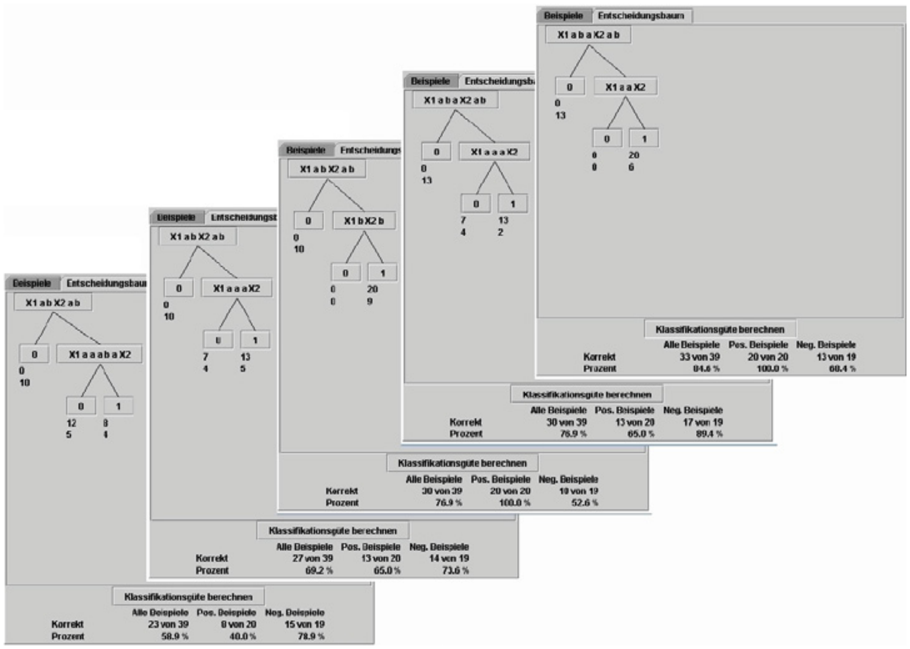
If the target of learning is some algorithm, challenging the target means to systematically probe the target by means of developing challenging input data. Just for illustration, think of the following application cases.

- There is some sorting algorithm such as *bottom-up heap sort.* Usually, students are learning about the more conventional heapsort algorithm before they are facing bottom-up heapsort.
  - Prepare input data such that bottom-up heapsort is not better than classical heapsort.
  - Prepare input data such that the advantage of bottom-up heapsort over heapsort is enormous.
- The decision tree algorithm *ID3* deserves to be called one of the oldies, but goodies of knowledge discovery and data mining.
  - Challenge the algorithm by some sample as small as possible which forces ID3 to generate a comparably complex decision tree.
- For NP-hard problems, there are usually some appealing heuristic solutions. Especially when the problem is intuitively understandable like, for instance, the *traveling salesman problem*, heuristics frequently seem to be ingenious. Choose any implementation of one of the heuristics like, for instance, the *nearest neighbor heuristics* or the *insertion heuristics.*
  - Construct some graph on which the heuristics does not work well.
  - In contrast, construct a certain example graph to which the heuristics applies perfectly.

For a playful learning experience when dealing with those problems, direct execution is essential. Learners should be released from executing operations and, instead, be able to put much emphasis on pondering problem details.

Readers who are familiar with the one or the other sample case mentioned above may admit that being able to challenge the target in the way described can be seen as a clear indicator of mastery.

Challenging the target is just one of the many educational patterns supported by means of direct execution. Figure 8 illustrates another example of changing decision trees by hand which immediately return results of varying quality.



| | Alle Beispiele | Pos. Beispiele | Neg. Beispiele |
|---|---|---|---|
| Korrekt | 23 von 39 | 8 von 20 | 15 von 19 |
| Prozent | 58.9 % | 40.0 % | 78.9 % |

| | Alle Beispiele | Pos. Beispiele | Neg. Beispiele |
|---|---|---|---|
| Korrekt | 27 von 39 | 13 von 20 | 14 von 19 |
| Prozent | 69.2 % | 65.0 % | 73.6 % |

| | Alle Beispiele | Pos. Beispiele | Neg. Beispiele |
|---|---|---|---|
| Korrekt | 30 von 39 | 20 von 20 | 10 von 19 |
| Prozent | 76.9 % | 100.0 % | 52.6 % |

| | Alle Beispiele | Pos. Beispiele | Neg. Beispiele |
|---|---|---|---|
| Korrekt | 30 von 39 | 13 von 20 | 17 von 19 |
| Prozent | 76.9 % | 65.0 % | 89.4 % |

| | Alle Beispiele | Pos. Beispiele | Neg. Beispiele |
|---|---|---|---|
| Korrekt | 33 von 39 | 20 von 20 | 13 von 19 |
| Prozent | 84.6 % | 100.0 % | 68.4 % |

**Fig. 8.** Challenging decision trees with regular patterns on a fixed set of input data

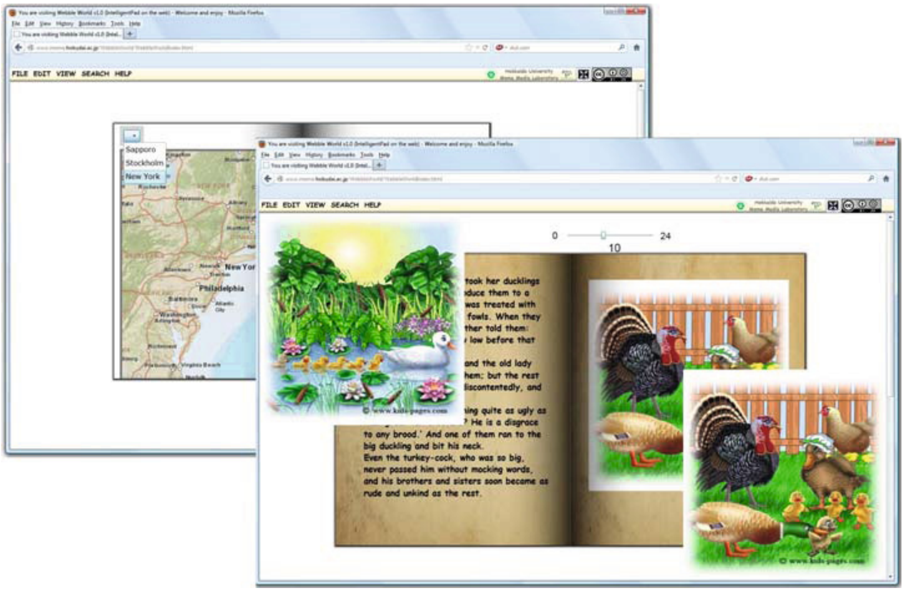
Webble technology–to be more precise: meme media technology, in general–allows for many more sophisticated educational patterns. One of the key features is that composite meme media objects may be decomposed in different ways. For instance, users may peel just one object off a larger construct to replace it by another one. Intermediately, the composite object is loosing its ability to execute the function it has been built for. After plug in of another component, the larger composite object becomes immediately operational, again.

This enables teachers to set up a large variety of playful exploratory settings. Think of composite objects in which replacing one component by another does drastically change the behavior ... or not. Replace, for instance, one correct procedure for solving a certain sub-problem by another procedure implementing some heuristics which in many cases returns them same result, but not always. This calls for a competition in finding the simplest cases showing the difference.

## 6 Summary and Conclusions

In accordance with the foreseeable establisment of HTML 5 as a W3C standard, one may expect a new generation of meme media to enter the stage quite soon.

Webbles based on HTML 5 may sit on pages of e-books where they allow for an enormous variety of functionalities. In harsh contrast, the vast majority of contemporary e-books are astonishingly boring. They allow for turning pages in a more or less fancy way and may contain slightly varying media such as videos. All this is still far behind the potentials of contemporary technologies sketches of which may be found on the Webble World Portal [22].



took her ducklings
duce them to a
was treated with
fowls. When they
ther told them:
low before that

and the old lady
hem; but the rest
liscontentedly, and

ing quite as ugly as
? He is a disgrace
to any brood.' And one of them ran to the
big duckling and bit his neck.
Even the turkey-cock, who was so big,
never passed him without mocking words,
and his brothers and sisters soon became as
rude and unkind as the rest.

**Fig. 9.** Digital books by Micke Kuwahara based on Webble technology as found on [22]

The screenshot on the left side of figure 9 in the background illustrates some case where an e-book accesses the Web whereas the screenshot in the foreground exemplifies the possibility to duplicate and extract some contents from an e-book.

There are currently only a very few publications such as [24], e.g., discussing "the textbook of the future". The authors of [24] advocate, for instance, the personalized externalization of knowledge, the textbook as an experiment box, and the textbook as an interface for communication and exchange (ibid., abstract). All this sounds like the re-invention of memetics from [8] to [30].

Issues of direct manipulation and direct execution as well as game-based learning are beyond the limits of those considerations.

The present contribution has been focused narrowly on direct execution with some emphasis on playful interaction and learning without explicitly stressing the book metaphor. For the human interaction with meme media, it does not matter whether or not the form of making contents and functionalities available resembles conventional books or not.

The author admits, nevertheless, that from a commercial perspective it might be useful to stick with the traditional book metaphor to get publishers engaged.

For purposes of learning, however, the traditional book format is irrelevant. What matters are interest attracting and thought provoking functionalities as well as features which allow for exploration beyond the limits of human beings left to their own devices.

As already said above, some special emphasis has been put on the paradigm of direct execution and on game-based learning. Naturally, there is direct execution which is not playful at all. And there is some game-based learning which does not need any direct execution. But the intersection of both areas–to use such a mathematical metaphor–is of particular interest.

It applies to a large variety of learning problems that effects of changing parameters are not easy to foresee and that finding out actual effects may be time-consuming and tedious. The field of computability theory as investigated by Arnold and Reichelt [5] is just an extreme theoretical, but illustrative example.

In those applications, direct execution turns out to be particularly benefical. Learners may literally play with parameters and observe the effects of settings. Some learners may be challenged by the task of finding parameters that cause particular target effects. For the purpose of mastery, they are enforced to exhibit the deeper reasons underlying the mechanisms under consideration, i.e. to learn.

Further features of Webble technology allow for making approaches to playful exploratory learning based on direct execution even more affective and effective.

Consider, for instance, the dashboard approach by Sjöbergh and Tanaka [29] by means of which a variety of external data may be sucked into a Webble system for visualization and direct execution. Approaches like that allow for a situated e-learning which integrates realistic and up to date information into learning environments. This seems particularly valuable for bringing the technology into corporate learning applications. It is definitely motivating for human beings in learning environments relating to their workplace, if the data are not made up out of thin air. Furthermore, realistic data are not only motivating, but allow for more easily understanding relationships.

For other applications such as natural language learning and training, real texts and phrases are known to be more effective than artificial constructs and, thus, the dynamic access of Webbles to data from world-wide resources provides an added value to language learning and language skill development.

To sum up, the present contribution's message is that Webble technologies provide characteristics valuable for technology enhanced teaching and learning. Beyond the limits of just saying so, it is the author's intention to provide a few examples and illustrations. Because studying involved contents usually requires a certain in-depth engagement, one of the sections above is a bit more elaborate.

**Acknowledgements.** First of all, it is necessary to direct the readers' attention to Yuzuru Tanaka who has been inspired by Richard Dawkins' work [8] and took the initiative to create the idea of *meme media* [30] and encouraged a larger number of young scientists to contribute to the design and the implementation of *meme media technology*. Some of this work is reflected within other contributions to this conference. Yuzuru Tanaka truly deserves to be named the father in spirit of meme media. The present author is grateful to him for numerous inspirations and for much support allowing for several years of exciting and fruitful cooperation.

The author has been drawing much benefit from cooperation with several of Yuzuru Tanaka's former students. Currently, Jun Fujima is working in the author's department, where he is continuously feeding the pool of meme media with his digitalized ideas. In many cases, he has been providing implementations on demand–essential contributions to the present author's work.

## References

1. Alexander, C.: A The Timeless Way of Building. Oxford University Press, Oxford (1979)
2. Angluin, D.: Finding patterns common to a set of strings. J. Computer and Systems Science 21, 46–62 (1980)
3. Arnold, O.: Die Therapiesteuerungskomponente einer wissensbasierten Systemarchitektur für Aufgaben der Prozeßführung. St. Augustin: infix (1996)
4. Arnold, O., Jantke, K.P.: Therapy plans as hierarchically structured graphs. In: Fifth International Workshop on Graph Grammars and their Application to Computer Science, Williamsburg, Virginia, USA (November 1994)
5. Arnold, O., Reichelt, V.: Meme media technology in higher education: The recursive function theory case study. In: Arnold, O., Spickermann, W., Spyratos, N., Tanaka, Y. (eds.) WWS 2013. CCIS, vol. 372, pp. 110–118. Springer, Heidelberg (2013)
6. Arnold, S., Fujima, J., Jantke, K.P.: Storyboarding serious games for large-scale training applications. In: Proc. 5th Intl. Conf. Computer Supported Education, CSEDU, Aachen, Germany (2013)
7. Blackmore, S.: The Meme Machine. Oxford University Press (1999)
8. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)
9. Fujima, J., Jantke, K.P.: The potential of the direct execution paradigm: Toward the exploitation of media technologies for exploratory learning of abstract content. In: Urban, B., Müsebeck, P. (eds.) Elearning Baltics 2012: Proceedings of the 5th International eLBa Science Conference, pp. 33–42. Fraunhofer Verlag (2012)
10. Franz, H., Thomas, L., Egon, W.: Processing of hierarchically defined graphs and graph families. In: Monien, B., Ottmann, T. (eds.) Data Structures and Efficient Algorithms. LNCS, vol. 594, pp. 44–69. Springer, Heidelberg (1992)
11. Hutchins, E.L., Hollan, J.D., Norman, D.A.: Direct manipulation interfaces. Human-Computer Interaction 1, 311–338 (1985)
12. Jantke, K.P.: Pedagogical patterns and didactic memes for memetic design by educational storyboarding. In: Arnold, O., Spickermann, W., Spyratos, N., Tanaka, Y. (eds.) WWS 2013. Communications in Computer and Information Science, vol. 372, pp. 143–154. Springer, Heidelberg (2013)
13. Jantke, K.P., Arnold, O.: Graphgrammatik-Konzepte in der Therapieplanung für komplexe dynamische Prozesse. In: Holzer, M. (ed.) Proc., Herrsching 4. GI Theorietag "Automaten und Formale Sprachen", pp. 42–47. Universität Tübingen, Wilhelm-Schickard-Institut (1995)

14. Jantke, K.P., Grieser, G.: Aspects of knowledge evolution in algorithmic learning. In: CSIT 2006, Amman, Jordan, April 5-7 (2006)
15. Jantke, K.P., Knauf, R.: Didactic design through storyboarding: Standard concepts for standard tools. In: Proceedings of the 4th International Symposium on Information and Communication Technologies, Cape Town, South Africa, January 3–6, pp. 20–25. Dublin, Ireland: Computer Science Press, Dublin (2005)
16. Jantke, K.P., Knauf, R.: Taxonomic concepts for storyboarding digital games for learning in context. In: Helfert, M., Martins, M.J., Cordeiro, J. (eds.) 4th International Conference on Computer Supported Education, CSEDU 2012, April 16-18, vol. 2, pp. 401–409. SciTePress, Poro (2012)
17. Jantke, K.P., Lengyel, D.: Die Realität in virtuellen Welten. Zeitschrift für e-Learning (1), 7–22 (2012)
18. Jantke, K.P., Spundflasch, S.: Understanding pervasive games for purposes of learning. In: Proc. 5th International Conference on Computer Supported Education, May 6-8. CSEDU, INSTICC, Aachen (2013)
19. Kirsten, D.: Properties of formal languages of therapy plans created by graph grammars (overview). In: Kurib, M., Worsch, T. (eds.) 5. Theorietag "Automaten und Formale Sprachen", pp. 132–142. Giessen: Justus-Liebig-Universität (1995)
20. Kleene, S.C.: The theory of recursive funxctions, approaching its centennial. Bulletin (New Series) of the American Mathematical Society 5(1), 43–61 (1981)
21. Knauf, R., Sakurai, Y., Tsuruta, S., Jantke, K.P.: Modeling didactic knowledge by storyboarding. Journal of Educational Computing Research 42(4), 355–383 (2010)
22. Kuwahara, M., Tanaka, Y.: Webble portal (2009), http://www.meme.hokudai.ac.jp/WebbleWorldPortal/
23. Lange, S., Wiehagen, R.: Polynomial time inference of arbitrary pattern languages. New Generation Computing 8, 361–370 (1991)
24. Neuhaus, W., Kirstein, J., Nordmeier, V.: Didaktische Funktionen des Lehrbuchs der Zukunft. Phydid B – Didaktik der Physik (2012), http://phydid.physik.fu-berlin.de/ (published online on January 19, 2013)
25. Board, P.P.A.: Pedagogical Patterns: Advice for Educators. Joseph Bergin Software Tools (2012)
26. Rogers Jr., H.: Theory of Recursive Functions and Effective Computability. McGraw-Hill (1967)
27. Shneiderman, B.: The future of interactive systems and the emergence of direct manipulation. Behavior and Information Technology 1, 237–256 (1982)
28. Shneiderman, B.: Direct manipulation: A step beyond programming languages. IEEE Computer 16, 57–69 (1983)
29. Sjöbergh, J., Tanaka, Y.: Visual data exploration using webbles. In: Arnold, O., Spickermann, W., Spyratos, N., Tanaka, Y. (eds.) WWS 2013. CCIS, vol. 372, pp. 119–128. Springer, Heidelberg (2013)
30. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)

# Meme Media Technology in Higher Education: The Recursive Function Theory Case Study

Oksana Arnold and Vincent Reichelt

Erfurt University of Applied Sciences, 99085 Erfurt, Germany
{oksana.arnold,vincent.reichelt}@fh-erfurt.de

**Abstract.** Computability theory is highly abstract and very demanding particularly to those students who have defective prior knowledge in mathematics or feel attracted only by directly applied topics of studies. However, understanding the foundations of computability is essential to computer science. Partial recursive functions form some key approach. Webble implementations of the basic functions of primitive recursion together with webbles for the basic operations of substitution and primitive recursion finally completed by the minimum operator provide a memetic framework for highly interactive, playful studies of computability theory. Didactically driven refinements of the approach are discussed in detail.

**Keywords:** meme media, webble technology, computability theory, recursive function theory, playful learning.

## 1 Memetics and Meme Media Technology

Dawkins [4] has coined the term *meme* to name units of non-biological evolution. Think, just for illustration, of fashion and architecture (see [3] for a large number of illustrative cases). There is something inherited over decades or even centuries. Apparently, changes such as mutation and cross-over take place. And all you know in fashion and architecture is subject to natural selection à la Darwin [5]. Ideas come and go; only those memes that are fit enough stay for a longer time.

Tanaka [18] has undertaken the endeavor of carrying over Dawkins' seminal ideas to the digital world where memes may be encapsulated in *meme media*. The computational power of contemporary computers allows for boosting the speed of memetic evolution of meme media by implementing the functionalities of reproduction, mutation, and cross-over taking place when using meme media.

Seen from the perspective of *memetics*, computer and computer networks as well as human beings working with and manipulating meme media form a biotope in which the digitalized memes–meme media objects–live and evolve. Humans unfamiliar with those ideas need some time to adopt the perspective of memetics.

The present paper is much less philosophical in character. It takes memetics and meme media technologies according to [18] and [10] as a basis for innovative applications in teaching certain involved topics of theoretical computer science.

## 2 Teaching the Theory of Effective Computability

For studying computer science, it is of some relevance to know what can be done by means of digital processing technology and where the limits are. When facing particular application problems, it frequently helps to relate them to already known problems and to insights about their principle solvability and complexity. In particular, when facing some essentially unsolvable problem, it might be nice to learn about the problem's hardness as early as possible.

*Computability theory* (known under many different names including terms such as *recursion theory*, *recursive function theory*, and the like) is providing the basic concepts, methodologies, and insights [9,12,14,15].

Due to its generality, the theory of effective computability relies on involved mathematical theories and methods. Consequently, many students who misunderstood informatics as a discipline of hacking code according to ones own taste are having particular difficulties with a firm foundation.

There is a widespread practice of teaching computability theory like a course in mathematics. The results are frequently disappointing.

Computation, by its very nature, usually proceeds in small steps, one by one. Particular models such as so-called TURING machines reflect this explicitly. Some other approaches such as the hierarchical definition of partial recursive functions are more algebraic in spirit. There are initial constituents and certain operators. In an algebraic sense, such a system uniquely defines the smallest class of objects containing all the initial constituents and being closed under the operations.

Such a, so to speak, algebraic approach to computability assumes as initial objects a couple of functions and operators as follows. For simplicity, one usually considers functions defined over the set of natural numbers.

- The initial functions are
  - the unary successor function $s$,
  - the $n$-ary constant functions $c_k^n$ with $c_k^n(x_1, \ldots, x_n) = k$,
  - the $n$-ary projection functions $p_k^n$ with $p_k^n(x_1, \ldots, x_n) = x_k$.
- The primitive operations are
  - substitution,
  - primitive recursion.
- The operation completing the approach is called
  - the minimum operator.

Note that there are enormously many simplifications possible. Due to the availability of the successor function $s$, there is no need for any other constant functions than the zero constants $c_0^n$, just for instance.

When you stick with explanations of how to define any computable function on the basis of these initial functions using the possibly repeated application of the three operators, it remains a course in mathematics.

Therefore, there have been developed and implemented meme media variants of all the constituents above [1]; see the following section. Having those meme media objects at your fingertips, you may playfully experiment by plugging one into the other. The feature of stepwise computation shows again.

Teaching the theory of effective computability means more than introducing some particular computability concept like partial recursive functions as above. Fundamental educational goals are

- an understanding of the generality of the varying classical approaches to effective computability,
- an understanding of several key results such as RICE's theorem and their fundamental, so to speak, philosophical interpretation,
- an understanding and an operational interpretation of syntactic results such as KLEENE's normal form theorem,
- the mastery of some application cases by means of an interpretation and an application of fundamental results.

The question for the deployment of new media and innovative approaches to teaching and learning to rather abstract contents is throwing some new light at the key results of recursive function theory. Some results are more amenable to exploratory studies than others.

Let us have a closer look at KLEENE's normal form theorem [13] (see also [15], §1.10, Theorem X) and RICE's theorem [16] (see [15], §2.1, not numbered). Here are the results in comparison valid for any so-called GÖDEL numbering $\varphi$ $I\!N$ is the set of natural numbers, $Pr$ is the class of primitive recursive functions, $\mathcal{R}$ is the class of total recursive functions, $\mathcal{P}$ is the class of partial recursive functions and $\mu$ denotes the minimum operator:

KLEENE's Theorem

$$\exists \alpha, \beta \in Pr \; \forall f \in \mathcal{P} \; \exists n \in I\!N \; \forall x \in I\!N \quad f(x) = \alpha(\mu y[\beta(x,y) = 0])$$

RICE's Theorem

$$\forall F \subseteq \mathcal{P} \; \forall d \in \mathcal{R} \;\; (\; (\forall n \in I\!N \; d(n) = 0 \leftrightarrow \varphi_n \in F) \;\longleftrightarrow\; F = \emptyset \vee F = \mathcal{P} \;)$$

It is difficult to imagine that human learners get a better understanding of RICE's theorem by any way of playing around with certain recursive function definitions.

In contrast, one may set up numerous exercise which essentially deal with KLEENE's normal form theorem. Take any definition of some partial recursive function which contains two nested calls of the minimum operator. Ask the students to find another definition which is equivalent to the given one, but contains only a single occurrence of the minimum operator. The appropriateness of the task is guaranteed by the KLEENE normal form theorem.

By the way, pondering the equivalence of the definition given to the students to the students' solution is an interesting scientific problem in its own right.

Let us summarize this section briefly. The theory of effective computability is quite involved. Some results have an enormous reach, but are difficult to master. A mastery of the theory of effective computability does definitely require some intensive studies and may be hardly achieved just through some playful activities. But pondering the essentials of such an involved theory may be substantially supported by providing technologies and tools for some exploratory endeavor–*exploring and understanding the abstract by direct manipulation of the concrete.*

## 3 Webbles of Recursive Function Theory

As described in section 2, the recursive function theory is accompanied by writing terms to describe certain functions. Instead of writing formulas, functions can appear as media objects as described in [1,7,8]. This section provides an overview of this previous work about Webbles of Recursive Function Theory.

First, the implementation of the six basic functions (see section 2) as media objects using the webble environment was performed. The three initial functions 1. constant; 2. projection; 3. successor, the two primitive operators 4. substitution; 5.primitive recursion with free slots for media objects (corresponds to the input functions) and 6. the minimum operator (see figure 1).

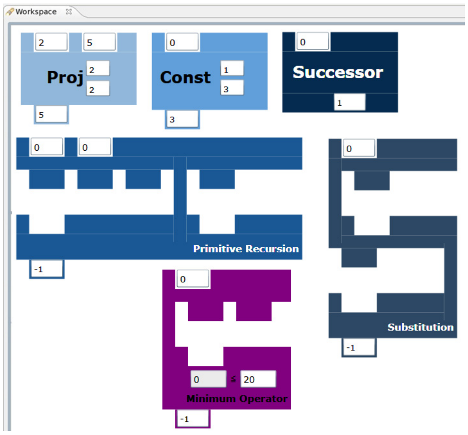


**Fig. 1.** The six basic primitve recursion webbles (PRF-webbles)

Each media object can be directly manipulated (see section 4), so its easily possible to change the values and properties of each operator. For example a substitution operator, which substitutes one unary function into another one, can be directly manipulated into a substitution operator, which substitutes one binary function into a unary one, within seconds by simply editing one property of the media object.

A major advantage of these media objects to terms and term equations is the possibility to directly manipulate the formula by simply plugging some function objects into the operator object. The result is a new media object which describes a new function. To understand the new formula you can enter some values into the media object. In addition defect formulas are avoided because its only possible to construct correct formulas with the media objects. Another advantage is the possibility of hiding parts of the formula by grouping some media objects.

## 4 Direct Execution and Playful Learning

Webbles representing any computable function or any operator may be directly manipulated on the screen–in accordance with Shneiderman's paradigm [17] of direct manipulation–and plugged one into the other for exploratory learning. Because every meme media object representing some computable function is not only representing, but *implementing* the function, more features are available. The meme media implementation provided (see section before) meets the requirements of so-called *direct execution*, a term coined by Fujima and Jantke in [6].

Direct execution has some varying aspects. First of all, it means that a meme media object fed with data is able to run immediately. Second, it refers to the completion of meme media objects. Think of some incomplete definition of a computable function which has input data sitting in their respective positions, but some part of the definition being not yet completed. As soon as the missing meme media objects are plugged in, the completed object is ready to run and executes the computation immediately.

Direct execution is an interesting refinement of direct manipulation which, after its introduction by [6], surely needs some more in-depth investigation and some m ore comprehensive experimentation.

In the present application case, direct execution allows for some type of playful exploratory learning as follows. Learners are given some composite media object not yet fully defined (see left object in figure 2). Input data (in this case 3 and 5) are sitting in place. The task is to complete the definition such that the resulting composite media object implements a certain target behavior (in this case adding any 2 input numbers). When a particular syntactically correct media object is inserted, the completed objects executes the computation as shown above. Now, it is ready for experimental validation.

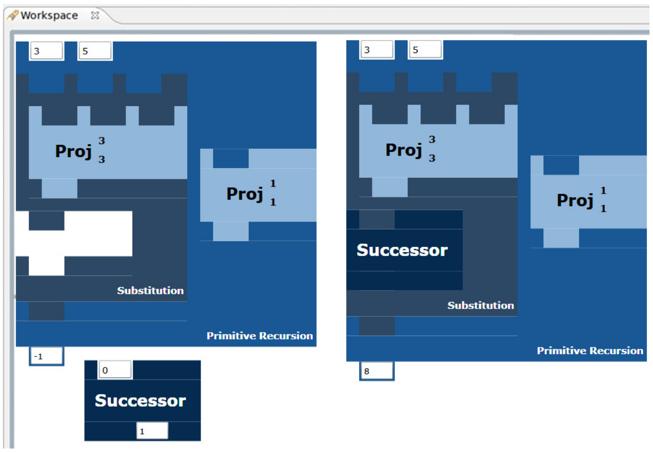
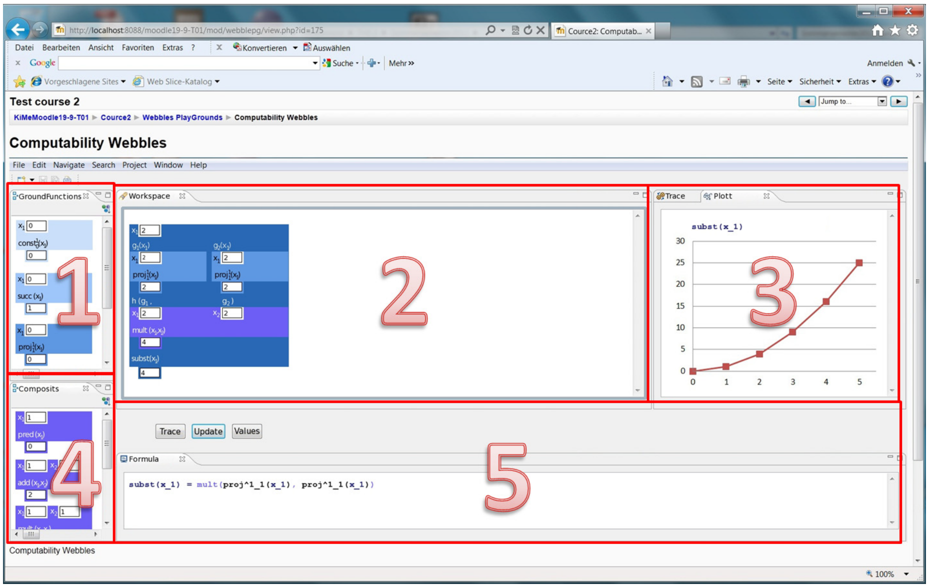


**Fig. 2.** Direct manipulation at work–some toy example before and after plugging in

## 5 Didactic Needs Reflected in Webble Technology

In section 2 we described the challenges which arise by teaching the recursive function theory to students. Out of these challenges, some didactic needs result to the learning program. This chapter gives an overview about these needs based on the planned program features. Figure 3 shows a fictional screenshot from the finished program. The red rectangles illustrate the five main segments each marked with a red number for the following description.



```
subst(x_1) = mult(proj^1_1(x_1), proj^1_1(x_1))
```

**Fig. 3.** Fictional screenshot from the finished learning program

1. If you want to create more complex primitive recursive functions you need a way to add the six ground functions: projection, successor, constant, substitution, primitive recursion and minimum operator (see section 3). The first segment *GroundFuntions* shows a menu which provides a simple way to achieve this task. The user can add the ground functions as PRF-Webbles to the *Workspace* (see figure 3 number 2) by simply clicking on the corresponding item.
2. The second segment *Workspace* is the main operating window. The users can change the attributes and values of the six ground functions and create new functions by dragging them together. When assembling new functions it is checked if they fit together. For example, when a projection has three input values, it needs a substitution or primitive recursion functions which has a placeholder for a input function with three input values. After the users have created a new function, they can "group" it to hide the inner functions

and label it (Figure 3 shows a purple grouped "mult" function). If the users want to create more complex functions they might want a possibility to add some self created functions. This is implemented through a save function which adds new (complex) functions to the users *Composits* (see figure 3 number 4).

3. Segment three shows a *plot* from the created function. This *plot* can help understanding the computation of the function, because you can see more than one input value. Nevertheless representing functions with more than three inputs is not well possible. For functions with more inputs there should be a option to select three -to be displayed- inputs. Another feature in this segment is the *trace*. In this window its possible to see each computation step. For instance it would display for the multiplication of 4 and 2 the following: 0: (MULT 4 2);1: (MULT 4 1);2: (MULT 4 0);2: MULT returned 0;1: MULT returned 4;0: MULT returned 8. When "ungrouping" the MULT function the traces shows also the sub-steps, up to the ground function level.
4. Segment four *Composits* displays the already mentioned self created functions for each user. It is possible to add these functions to the *Workspace* (see figure 3 number 2) by clicking on them and to export them into a xml-file. Exporting functions could be useful in many scenarios. For example students could exchange their functions to see other solutions, to work together on a more complex function or to send them to the teacher for correction.
5. Users may want to present different formulas from the lecture or books as PRF-Webbles. The last segment contains a formula editor which provides this functionality. In this editor the users can input PRF-formulas to create PRF-Webbles in the *Workspace* or they can click on PRF-Webbles in the workspace and display the associated formulas. Although the formula has to be in a special syntax similar to the latex formula syntax the user has as much freedom as possible. For instance he can choose his own function and variable names. This facilitates the work with different sources. When you want to display the formula you can choose if you want to display it in latex format or if you want to display it with superscript and subscript which makes it easier to read.

## 6 Summary and Insights into Future Work

In summary it can be stated the following. The theory of effective computability is for many students difficult to understand. Providing technologies and tools for exploratory endeavor – *exploring and understanding the abstract by direct manipulation of the concrete* – might promote the learning process (see section 2 and 4). Previous studies dealt with displaying functions as media objects instead of writing formulas [1,7,8] (see section 3). This paper deals with a more thorough discussion of the didactic needs for a learning program with these media objects and shows possibilities of implementation of which some have already been implemented (see section 5).

In the future this research field will open more interesting questions. As functions can be learned by their function curve [11,2] it might be possible to create PRF-Webbles out of the function *plot* (see section 5, segment three) or to create the corresponding formula in the formula editor. However, the implementation of this functionality is not easily possible and requires an in-depth discussion. Other interesting question are: How will such a learning program be accepted by users? How will it enhance the knowledge of the users? What other functions want the users? For these purposes, it is possible to carry out a study with students who will use this program in the now commenced semester at the University of Applied Sciences Erfurt.

## References

1. Arnold, O., Fujima, J., Jantke, K.P., Tanaka, Y.: Exploring and understanding the abstract by direct manipulation of the concrete. In: Helfert, M., Martins, M.J., Cordeiro, J. (eds.) 4th International Conference on Computer Supported Education (CSEDU 2012), April 16-18, vol. 2, pp. 100–107. SciTePress, Porto (2012)
2. Arnold, O., Schulz, A.: Technologies, implementations and applications for the Kleistian development of thoughts in speech. In: The 1st IEEE Global Conference on Consumer Electronics, GCCE 2012, October 2-5, IEEE, Makuhari Messe (2012)
3. Blackmore, S.: The Meme Machine. Oxford University Press (1999)
4. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)
5. Flechsig, K.-H.: Kleines Handbuch didaktischer Modelle. Neuland, Eichenzell (1996)
6. Fujima, J., Jantke, K.P.: The potential of the direct execution paradigm: Toward the exploitation of media technologies for exploratory learning of abstract content. In: Urban, B., Müsebeck, P. (eds.) Elearning Baltics 2012: Proceedings of the 5th International eLBa Science Conference, pp. 33–42. Fraunhofer Verlag (2012)
7. Fujima, J., Jantke, K.P., Arnold, O.: Media multiplicity at your fingertips: Direct manipulation based on webbles. In: Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, pp. 217–222. IEEE (2012)
8. Jantke, K.P., Fujima, J., Arnold, O., Schulz, A.: Memetic communication media – concepts, technologies, applications. In: Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, pp. 260–265. IEEE (2012)
9. Kleene, S.C.: The theory of recursive funxctions, approaching its centennial. Bulletin (New Series) of the American Mathematical Society 5(1), 43–61 (1981)
10. Kuwahara, M., Tanaka, Y.: Webble portal (2009), `http://www.meme.hokudai.ac.jp/WebbleWorldPortal/`
11. Board, P.P.A.: Pedagogical Patterns: Advice for Educators. Joseph Bergin Software Tools (2012)
12. Machtey, M., Young, P.: An Introduction to the General Theory of Algorithms. Elesevier, New York (1978)
13. P'eter, R.: Recursive Functions in Computer Science. Akad'emia Kiad'o, Budapest (1981)
14. Shneiderman, B.: The future of interactive systems and the emergence of direct manipulation. Behavior and Information Technology 1, 237–256 (1982)

15. Rogers Jr., H.: Theory of Recursive Functions and Effective Computability. McGraw-Hill (1967)
16. Shneiderman, B.: Direct manipulation: A step beyond programming languages. IEEE Computer 16, 57–69 (1983)
17. Tanaka, Y.: IntelligentPad as meme media and its application to multimedia databases. Information & Software Technology 38(3), 201–211 (1996)
18. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)

# Visual Data Exploration Using Webbles

Jonas Sjöbergh and Yuzuru Tanaka

Meme Media Lab., Hokkaido University, Japan
{js,tanaka}@meme.hokudai.ac.jp

**Abstract.** We describe a system for visual exploration of data built using pluggable software components called Webbles. The system specifies a small common interface, a set of slots the plugins are expected to have, and any Webble following this interface can be plugged in at runtime. The system contains several types of visualization components, some built from scratch, some built by writing Webble wrappers for existing software, and some built by writing small interface wrappers for existing Webbles. The visualization components allow for interactive exploration of data, and selections or grouping of data in one visualization component are propagated to other components automatically. Interaction is done through direct manipulation of the visualization results.

## 1 Introduction

Nowadays very large collections of data are common in many areas. Sometimes the real world system generating the data is difficult to model, what parameters influence the model in what way may not be known in detail. Thus, even though we can collect lots of data that probably contains information we are interested in, using this data to predict what we want to know may still be difficult.

One example from our research is snow plowing and snow removal in Sapporo, a city with almost two million citizens that gets six meters of snow per year. The cost of removing snow are around 15,000,000,000 yen per year (150 million dollars). Using modern IT technology, it may be possible to decrease cost, improve quality, or remove snow in a more environmentally friendly way.

We have data from many sources: weather stations (temperatures, snow fall, wind, etc.); taxis and private cars regularly recording speed and position; snow plowing and snow removal records; logs from call centers for snow related problems; traffic jam sensors; traffic accident reports; bus traffic logs; social media like Twitter (are people talking about problems with snow?), and more. A lot of these data are likely to be related to snow removal, e.g. less complaints when snow removal works well or lower average speeds of cars and buses when snow is not removed. The system is of course very complex, though, and it is difficult to model what factors will have what impact on e.g. traffic conditions.

The data is very high dimensional, very sparse (e.g. a few thousand cars with sensors driving over 100,000 road segments means low coverage), and often of low quality. There are sensor failures leaving "holes" in the data, manually entered data (e.g. call center complaints) with input errors, and unreliable data sources, e.g. people on social networks may not be serious or even telling the truth. These factors make naive application of many automatic analysis methods difficult.

We believe that for problems that are difficult to model, interactive visualization and exploration of data can be useful. Once the problem is understood well enough to be modeled, standard statistical approaches can be used to find optimal solutions but before we can model the system such methods are difficult to use. Interactively exploring data helps in understanding and give insights, and if it can be done by domain experts (not only data mining experts) they can use their expert knowledge and intuitions about the domain too.

We present a data exploration system, the *Digital Dashboard*. It was built using pluggable software components called Webbles [1], the latest generation of the Meme Media IntelligentPad system [2]. Meme Media aims to make services and functionality as easy to reuse as copy-pasting texts or images already is.

## 2 System Description

The *Digital Dashboard* is being developed in a project focusing on snow removal and it was designed to be easy to use by both experts and non-experts. The main user interface is direct manipulation of the visualization results. Aside from the *Dashboard* Webble itself, there are two types of components: *data sources* and *visualization components*. New components of both types can be built and plugged in at any time, even when already running. The only thing required is that components adhere to a simple interface.

User interaction is handled by each component separately and thus any type of interaction is possible. Existing components allow for instance clicking on bars in a histogram to select data items that correspond to those bars, or selecting areas on a map to select data items from those areas. Selecting separate groups of data, e.g. two different areas on a map, can be used to select non-continuous subsets of data or to create groups of data to compare to each other. When selections in one component change, all other components are immediately updated.

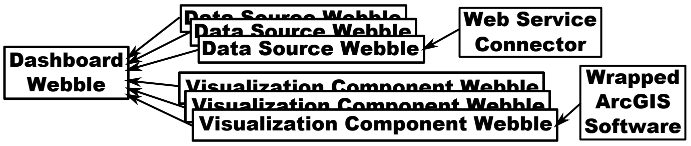


**Fig. 1.** An example Webble hierarchy. All components are children of the Dashboard Webble, and some children wrap other Webbles that are in turn connected as children.

An example diagram of the connections between the Webbles is shown in Figure 1. The *Dashboard* Webble is the top parent of all Webbles. The visualization components can have children of their own, if they for instance wrap the functionality of software not developed for the *Dashboard*. The data source Webbles can also have children, e.g. Webbles for accessing Web services. No slots are

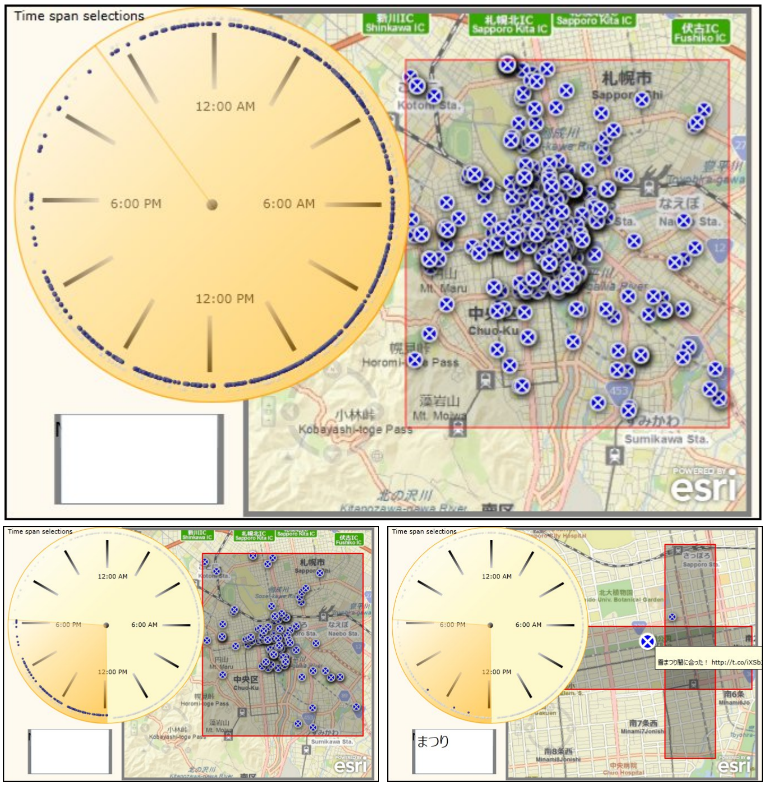


**Fig. 2.** Interacting with Twitter data. **Top:** Data contains timestamps (24h clock, upper left), text (search box, lower left), and GPS tags (map). **Bottom Left:** Selecting only afternoon data. **Bottom Right:** Restrict to city center areas, zoom in, and show only tweets with the word “matsuri” (“festival”). Mouse-over shows an individual tweet.

directly connected but the *Dashboard* (parent) Webble listens to slot changes from its children and reads and sets slot values of the children when appropriate. In Figures 2 and 3, all visible Webbles are visualization components, except the map which is a Webble developed independently and later wrapped to work with the *Dashboard*. The *Dashboard* Webble and the data source Webbles are all invisible, but can be made visible if there is a reason to interact with them.

## 2.1 Usage Example

Figures 2 and 3 show a simple example of using the *Digital Dashboard* to explore data from our snow removal project. The user opens the *Webble World* web page[1] and loads the *Digital Dashboard* Webble. The user also loads a data source Webble for some data collected from Twitter. The data contains short

[1] `http://www.meme.hokudai.ac.jp/WebbleWorld/WebbleWorldIndex.html`

text messages that are timestamped and GPS tagged. To visualize these data, the user loads a 24 hour clock visualization component, a text query component, and a map, resulting in the top image in Figure 2. The map shows locations were tweets were sent and the clock displays dots at times tweets were sent.

Next (bottom left image) only tweets from the afternoon are shown, by selecting (clicking and dragging) a part of the clock. This automatically updates the dots displayed both on the clock and the map. The selection is further restricted to two smaller areas by clicking and dragging on the map. The user also zooms in, and selects only tweets containing the word "festival" (in Japanese) using the text query component (bottom right image). The contents of an individual tweet are also examined by hovering over the location of the tweet.

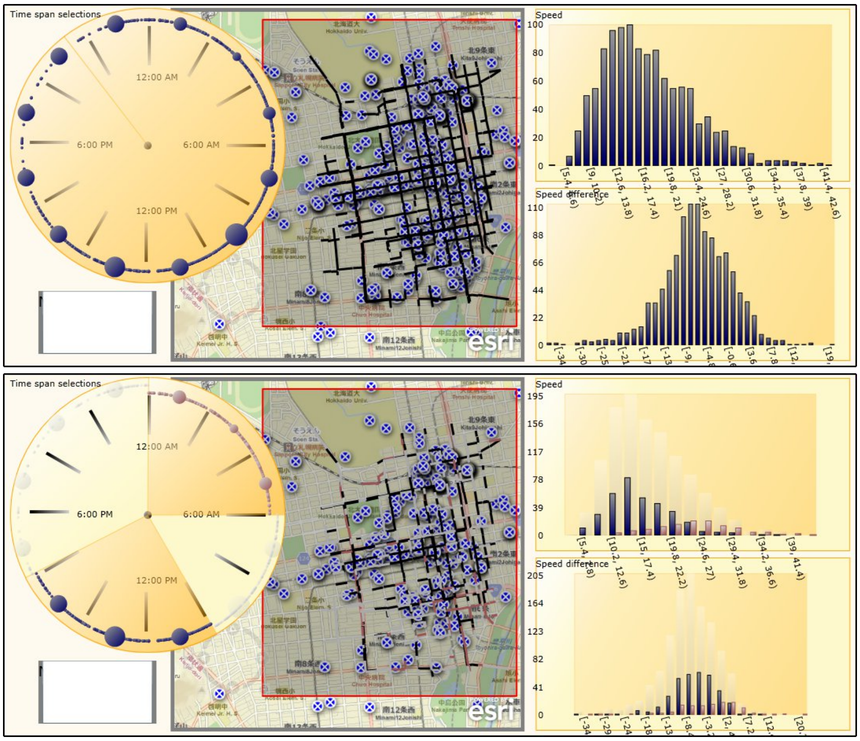


**Fig. 3.** Mash-up of Twitter data and taxi probe car (per road segment) data. **Top:** Data is joined on timestamps (24h clock, upper left) and geographic position (map, center). Twitter data also has a text filter. Taxi data also has speed and speed difference compared to summer (histograms). **Bottom:** Grouping data by time, night vs. noon.

In Figure 3 the user continues by adding a new data source with taxi probe car data. This contains road segments. For each road segment in the city there is the geographical location, a timestamp, the average speed of and average number of sensor equipped cars passing each 5 minute interval, and the difference in average speed compared to the speed in the summer. The data were further averaged over two hour intervals in this example.

The user also loads two copies of a histogram visualization component. The user joins the two data sources on the timestamp data, then shown in the 24 hour clock, and on the geographical information, shown on the map. The average speed is shown using the top histogram, and the drop in average speed compared to summer in the bottom histogram (top image in Figure 3). The user then (bottom image) groups the data into two groups, night time data and noon data, by clicking and dragging two times on the 24 hour clock component. The speed histogram shows that taxis have higher average speeds at night (light bars) than at noon (dark bars).

An example of trying to find something more useful is selecting only road segments with large speed downs compared to summer conditions, by selecting this part of the histogram showing speed downs and selecting only night time traffic using the clock component. Speed downs can be caused by many things but if the winter traffic is slow even at night (little traffic) the cause is often ice or snow related. It is then possible to zoom in on problem areas and see what people on Twitter are mentioning there ("slippery"? or "construction work"?). Adding other data sources like traffic accident data may also give useful information.

## 2.2 Data Source Components and Their Interface

The data source components hide the actual data format from the system, making the system independent of the underlying data storage. This makes it easy to mash-up data from different sources, e.g. visualizing probe car data stored in a relational database and complaints to a call center stored in a plain text file together with geographically tagged text messages received from a social networking service in real time. The data source plugins are responsible for transforming the data to a standard format used by the *Digital Dashboard*. We have generic plugins that access relational databases and format the result of an SQL query into the expected format, and plugins for accessing web services. This makes it easy to add new data sources if they are of any of the already supported types. The system currently expects data in XML format and it can look like this:

```
<data name="probe car data">
   <field name="timestamp" datatype="time-of-day">
      <row><id>19a</id><value>10:15</value></row>
      <row><id>35d</id><value>11:11</value></row>
      ...
   </field>
   <field name="speed" datatype="numerical">
   ...
   </field>
</data>
```

The XML contains the name of the data source to present to the user, names and data types of the data fields in the data set, and data values. All data values have an ID field and any data with the same ID coming from the same data source are assumed to be connected.

The data types are used to determine which visualization plugins can visualize what data. A clock plugin for timestamps will only be matched to data of that type, and plugins that visualize geographical locations will only be matched to data containing geographical information, etc. The system itself does not need to understand the data types, the only requirement is that the data source and the visualization plugins agree on the name and format of the data type. This means that it is possible to add plugins that handle new data types that the *Dashboard* system has never seen without changing anything in the system.

The interface of the data source components is very simple. They are expected to have two slots: *PluginName* and *DataValues*. The *PluginName* slot should contain a string with the name to display to the user when necessary. The *DataValues* slot should contain the data in XML.

When doing a mash-up of several data sources, the *Dashboard* is used to choose how to join the different data sets. Data fields in separate data sets that have the same data types can be joined. An example: we might have data with latitude, longitude, time-of-day and text in our *Twitter* data source; data with latitude, longitude, time-of-day, and speed in our *Probe car* data set; and data with time-of-day and temperature in our *Weather sensor* data set. Then the *Twitter* and *Probe car* data sets could be joined on the time-of-day and on the location, and all three data sets could be joined on the time-of-day. Selecting a geographic area using some visualization component would then remove subsets of both the *Twitter* data and the *Probe car* data (data points outside the selected area), but would not affect the *Weather sensor* data. Selecting only data from the night time would affect all data sources, and selecting only data containing the word “snow” would only affect the *Twitter* data.

## 2.3 Visualization Components and Their Interface

The *Digital Dashboard* hides the properties of the data and the existence of and interaction with any other visualization components from each visualization component. The parts of the visualization component that interact with the rest of the *Dashboard* are thus simple, and more effort can be spent on the actual visualization. Visualization components are expected to have the following slots:

*PluginName*, should have a string indicating what the plugin should be called in the user interface, e.g. when selecting which plugin to use for certain data.

*RequiredDataFields*, a slot with XML describing what data fields this plugin wants. A plugin showing dots on a map for locations of sent text messages and displaying the message text when the mouse pointer is over the message location could specify that it wants a latitude, a longitude, and a text field. Fields can be optional, e.g. it could be OK that some items have no text but all items must have latitude and longitude. A plugin can accept several different sets of data fields, for instance a plugin showing road segments could accept fields with start and end points or fields with a start point, a direction, and a segment length.

*DataValues*, a slot where the *Dashboard* inputs the data to be visualized, in one of the formats specified by the previous slot.

*GroupColors*, XML indicating which group of data should be visualized in what color. This slot is used to coordinate the use of color between all plugins so that e.g. green dots on the map correspond to green bars in the histogram.

*LocalSelections*, a slot with a dictionary mapping data item IDs to group IDs. The mapping shows which data items are selected, and in which group it has been placed. The selections are the selections in this component only. If the user has selected one group of data with speeds above 60 km/h and another group of data with speeds below 10 km/h by selecting bars in a histogram, the dictionary would map each ID belonging to data items with speeds above 60 km/h to group 1, IDs of data items with speeds below 10 km/h to group 2, and all other IDs to group 0 (the unselected items).

*GlobalSelections*, also a dictionary mapping data item IDs to group IDs. Here, the mapping shows the global selection status. A data item may be put into group 1 in one component but be unselected in another (globally unselected), or items belonging to the same group in one component may be in different groups in another component and thus be in different groups globally.

The *Digital Dashboard* coordinates the visualization components and aggregates the information from them. When the local selections in one component changes, the system updates the global selection status of any data items affected and then updates all the visualization components affected by any changes.

The system also hides the data sources from the visualization components, and transforms the data source XMLs to new XMLs with the data needed by each visualization component, extracting wanted fields and removing unwanted ones, and renaming the fields to the expected names. The system will try to automatically connect data fields and visualization components. If there is only one field of a certain type (e.g. only one time-of-day field) in the data, this field is automatically assigned to all components that expect such a field. When there are several fields to choose from, the system can make guesses or it can be left to the user to select from a list of the fields of the correct type.

### 2.4 Wrapping Existing Functionality or Software

There are templates for making both visualization and data source plugins for the *Digital Dashboard*. A template is a Webble that has the slots needed and some skeleton code for common operations on slot changes, e.g. iterating over the global selection dictionary. Building a new Webble that follows the *Dashboard* interface is fairly easy but requires some programming knowledge.

Meme Media aims to reuse existing functionality and existing software can be used in the *Dashboard*. An unrelated Webble that can visualize data, handle user interaction, or interact with a data source, can be used by making an adapter Webble. The adapter Webble just needs to transform the data etc. between the format of the existing Webble and the format of the *Dashboard*. Depending on how different these are, this could be very easy, e.g. an almost empty adapter

Webble that just forwards data from slots with the proper names to differently named slots, or it can be a fairly complex task.

We built an adapter Webble that made use of several different existing Webbles to handle visualization and interaction with geographical data. The adapter Webble uses one existing Webble to display data on a map by simply transforming the XML on the *DataInput* slot to a comma separated list and sending it to the data slot of the existing Webble. It also takes user area selections on the map from a different existing Webble, and matches the selection coordinates to coordinates in the *DataInput* slot to create a mapping for the *LocalSelections* slot. Adding an adapter like this is easy and it took about an hour for this adapter.

The Webble technology also allows wrapping of software that was not built using Webbles. The ArcGIS[2] software, not built using Webbles, has been wrapped to run as a Webble and to work with the *Dashboard*. Depending on the software, wrapping can be fairly easy or it can require a lot of work. Once wrapped, software can be used in any Webble application.

Wrapping a new data source to work with the *Dashboard* is generally easy. There are generic Webbles that access relational databases and web services and output data in the expected format. For many data sources, adding them is as easy as adding the URI, then specifying a table name and a login.

## 2.5 Strengths and Weaknesses of the Digital Dashboard System

Direct manipulation is intuitive and easy to understand for non-experts, so the *Dashboard* user interface is easy to use. Clicking and dragging the mouse to select e.g. an area on a map is a familiar and easy to understand operation. Selections and groupings in one component are reflected immediately in other components. All visualization components allow direct interaction with the visualization results. Selecting the data you are interested in by interacting with the thing that tells you it is interesting is powerful and convenient.

Setting up a new data exploration using already existing components is easy. No programming is required, just select plugins and connect them to the system. Changing what each component visualizes is easily done using drop-down lists.

The system is very generic, and data sources containing data types the system has never seen before can be dropped in at any time. As long as there are visualization components that know these data types, the system will start handling everything automatically. The system also does not need to know what type of visualization or interaction any visualization component does, and the individual components do not need to know or care about other components. This means any type of data storage, and any type of interaction, etc. is allowed.

Since the system is built using Webbles, it runs in a Web browser and no special installation is necessary (though a Silverlight browser plugin is needed). The Webble technology also means that components developed for the *Digital Dashboard* can easily be reused in other systems and some components have

[2] ESRI (2012) ArcGIS API for Silverlight (ver. 2.4) `http://help.arcgis.com/en/webapi/silverlight/`

already been reused in systems built by others. The pluggability of the system makes it easy to extend the functionality, to prototype new functionality, to add more data sources to mash-up with previous data, etc. When no available component does what the user wants, wrapping existing software is possible.

The biggest weakness of the system right now is that it is quite wasteful with regards to memory. A lot of data is duplicated in different ways. Using XML for the data also uses a lot of memory. We plan to extend the system to also allow e.g comma separated lists of values instead of XML, though this will not help with data being duplicated in both the system itself, in an adapter plugin, and in the final visualization Webble, etc.

## 3 Other Similar Systems

Here we give a short overview of other systems for visualization and exploration of data, and point out some differences between them and our system.

*RapidMiner* [3] is an open-source prototyping system for knowledge discovery and data mining. It is widely used and supports very many data mining and machine learning algorithms. It is used to graphically set up work flows for data mining. It supports multiple views of the same data and you can go back and change some step of the work flow and get a different visualization results. Like in our system, the underlying data format is hidden from the data mining operators or visualization components, and changes made to a work flow are reflected in all views of those data. The interaction when setting up work flows is a visual process, but, there is no interaction with the visualized results.

*Snap-Together Visualization* [4] (*Snap*) is very similar to our system. Different visualization components are connected and e.g. selecting data in one component is automatically reflected in connected components. As in our system the interaction can be bidirectional so interactions with the second component are reflected in the first component too. Components can be connected in different ways, so selecting an item may select related items in one connected view, and open details about the selected item in another view. *Snap*, like our system, has a small interface that components need to follow and it is possible to wrap existing software with an interface wrapper (in for instance Visual Basic). Unlike our system, *Snap* expects the data to come from a database (supporting ODBC) and uses Microsoft's COM for component (process) communication.

The *VERD* system [5] uses IntelligentBox, another version of Meme Media, for data visualization. It works with relational databases and has a wrapper to treat Web resources as relational schema too. Using direct manipulation, it allows interactive data exploration using various visualization methods. All visualization views set up are also treated as relations, making it possible to apply the same operations to visualization views as to raw data. *VERD* requires the IntelligentBox environment while our system runs in any Web browser, and in *VERD* the interaction only goes downstream in the visualization flow, i.e. interaction with one component does not affect components earlier in the flow.

*DEVise* [6] is a data exploration system for relational databases. It supports multiple views that can be connected so that e.g. zooming in one view is reflected in another view, and connections can be bidirectional. *DEVise* supports many types of operations on the data, but the types of user interaction with the visualization results are somewhat limited.

*Tioga-2* [7] (now called *Tioga DataSplash*) is a direct manipulation system for setting up visualizations of database contents. Multiple visualizations can be connected and changes in the range to visualize in one are then automatically reflected in others. *Tioga* has a fairly limited set of visualization primitives and only works with relational databases as the data source.

## 4 Conclusions

We described a system for interactive exploration and visualization of data. It is built using pluggable components and it is easy to plug in new data sources to mash up data, and easy to plug in new components for new types of visualization. All components are connected and interactive, and e.g. selections of data in one component is immediately reflected in all other components with related data.

We believe that systems such as this are useful especially for problems that are hard to model. Exploration of the data using interactive visualizations to gain ideas on how to model or analyze the data, or what subsets of data to model in what ways, is helpful. Directly interacting with the visualization results telling you that something is interesting is a powerful and convenient way to further explore the parts that look interesting.

## References

1. Kuwahara, M., Tanaka, Y.: Webble world – a Web-based knowledge federation framework for programmable and customizable Meme Media objects. In: The IET Intl. Conf. on Frontier Computing 2010, Taichung, Taiwan, pp. 372–377 (2010)
2. Tanaka, Y.: Meme Media and Meme Market Architecture. IEEE Press, Piscataway (2003)
3. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid prototyping for complex data mining tasks. In: KDD 2006: Proceedings of the 12th ACM SIGKDD, Philadelphia, PA, USA, pp. 935–940 (2006)
4. North, C., Shneiderman, B.: Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In: Proceedings of AVI 2000, Palermo, Italy, pp. 128–135 (2000)
5. Sugibuchi, T., Tanaka, Y.: Integrated visualization framework for relational databases and web resources. In: Proceedings of IHI 2004, Dagstuhl Castle, Germany, pp. 159–174 (2004)
6. Livny, M., Ramakrishnan, R., Beyer, K., Chen, G., Donjerkovic, D., Lawande, S., Myllymaki, J., Wenger, K.: DEVise: Integrated querying and visual exploration of large datasets. In: Proceedings of SIGMOD 1997, Tucson, AZ, USA, pp. 301–312 (1997)
7. Aiken, A., Chen, J., Stonebraker, M., Woodruff, A.: Tioga-2: a direct manipulation database visualization environment. In: Proceedings of ICDE 1996, New Orleans, LA, USA, pp. 208–217 (1996)

# The Geographic Information Science Body of Knowledge 2.0: Toward a New Federation of GIS Knowledge

Nigel M. Waters

Dept. of Geography & GeoInformation Science, George Mason University,
4400 University Dr., MS 6C3, Fairfax, Virginia, USA 22030
nwaters@gmu.edu

**Abstract.** Geographic Information Science is a relatively new discipline. Indeed the teaching of the technology of Geographic Information Systems did not really begin as a university subject until the US based National Center for Geographic Information Analysis developed a "Core Curriculum" in 1990. This Curriculum was revised in the late 1990s in an ongoing effort to provide educational resources for GIS university lecturers. Within the GIS community, a perceived need to update this Curriculum, on an ongoing basis, led to the development of a "Body of Knowledge" by representative organizations from academia (primarily the US University Consortium of Geographic Information Science), professional bodies (e.g., the Association of American Geographers) and the world-wide, industry leading GIS software manufacturing company, Esri. This Body of Knowledge was published in 2006 and it is now freely available online. However, it has become dated and originally there was no process implemented for keeping it current. As a consequence there is now a new project to develop the Body of Knowledge 2.0 (BoK 2.0). On April 13th, 2013, the first meeting of those involved in the development of BoK 2.0 will be held at the Association of American Geographers annual conference in order to plan a process for completing this project. The BoK 2.0 will be developed using collaborative technologies such as wikis. These will be facilitated by new immersive computer environments similar to the technology used in Second Life. This keynote talk will explore how these new technologies, including new developments arising from social media, will integrate a world-wide knowledge federation of Geographic Information Science. Links to these approaches and the Webble World environment will be explored and suggested.

**Keywords:** GIS, Body of Knowledge, Core Curriculum, federation, Webble World.

## 1 Introduction

This paper begins with a short review of the history of Geographic Information Science (GIS), both from a research and a pedagogic perspective. This narrative, which will have a strongly Anglo-North American focus, recounts the origins of the technology from the mid-1950s and the subsequent founding of the National Center for Geographic Information and Analysis (NCGIA) in the late 1980s. The role of the NCGIA in stimulating research and the growth of academic geography in North America will be described and special emphasis will be invested in the development of the Core Curriculum (both the original and the subsequent revised versions) in promoting the teaching of GIS in North American Universities and indeed around the world. The subsequent role of the original GIS Body of Knowledge (BoK 1.0) in furthering the development of the discipline after the turn of the century will be described. The limitations of this approach in that it lacked depth and represented a static, unchanging and unevolving view of the discipline will be discussed. Current attempts to produce a GIS Body of Knowledge that is dynamic, uses a wiki-like approach, involves new immersive media and a Webble World philosophy to capture the ongoing growth and evolution of GIScience will be considered along with speculations on the future of curricula within the discipline.

## 2 The Origins of GIS

The history of Geographic Information Systems and Geographic Information Science has been described by [15], [5], [40] and [41]. Geographic information has long been highly valued and, in describing the history of the discipline, many GIS texts cite the importance of military, geographic information, giving as an example the thematic overlay maps of the French cartographer, Berthier, who used this device to show the sequence of troop movements at the Battle of Yorktown in 1781 during the American War of Independence. Equally important in the history of GIS, during the subsequent century, was the work of British physician, John Snow, in plotting a map of cholera deaths in London in 1854. The concentration of the victims' residences around the nearby, water pump in Broad Street and the cessation of the outbreak, following the removal of the pump handle, allowed the water borne cause of the disease to be determined for the first time.

The modern era of Geographic Information Science is generally considered to go back to the 1950s [4]. Coppock and Rhind [5] split the first 40 years of GIS history into four periods: 1) the Pioneer Era (1950s-1975); 2) the Government Supported, Experimental Period (mid-1970s to early 1980s); 3) the Commercial Period (early 1980s to the start of the 1990s); 4) The User Dominance Era (from the 1990s onwards). For each of these periods the importance of conceptual developments, software and hardware innovations, the influence of academia, commercial enterprises and governments were examined. Here we will look primarily at the first three phases of this history and then return to the story later in the paper.

Phase 1 included such conceptual developments as the map overlay or "layer cake" approach which allowed the integration and the investigation of the influence of many spatial variables. This approach was popularized by Ian McHarg in his book Design with Nature [26]. Steinetz et al. [31] include a complete discussion of this methodology and demonstrate convincingly that the original conceptual framework was the work of Jacqueline Tyrwhitt whose essay in the Town and Country Textbook published by the Architectural Press, London, includes a detailed discussion of how map overlays can be used as a planning tool. A group of scholars in the Department of Geography at the University of Washington in Seattle also made conceptual innovations at this time and one of the most important was the introduction of the geographical data matrix by Berry [3]. His representation of geographical data in matrix form with the rows representing the spatial units and the columns their attributes rapidly became the de facto standard for organizing GIS data. It was during this period that the first fully functional GIS, the Canada Geographic Information System (CGIS), was put into production. This system, which was originally a collaboration between Roger Tomlinson's company, Spartan Air Services of Ottawa, and the Canadian Government's, Canada Land Inventory, pioneered a series of hardware and software innovations [41]. During this time academia's contributions were largely due to the work undertaken at the Harvard Lab for Computer Graphics and Spatial Analysis which had been founded in 1965 by Howard Fisher and produced computer mapping packages such as SYMAP and SYMVU among others that were widely used throughout North American Universities. William Warntz followed Fisher as the Director of the Lab in 1969 and made some truly innovative contributions concerning the critical features of three dimensional surfaces that led to the use of triangulated irregular networks (TIN) models to provide compact storage of surface features [39], [37].

Phase 2 was characterized by the improvement of many of the fundamental algorithms of coordinate geometry (COGO) on which GIS software development depended. These included point-in-polygon and polygon overlay routines as well as shortest path and location-allocation algorithms, among others. By the end of this period GIS software was beginning to move away from its mainframe roots and on to minicomputers and, eventually, the newly introduced IBM microcomputers.

Phase 3, the commercial period, saw the increasing dominance of the major software companies such as Intergraph, MapInfo, Caliper Corporation and Esri. Despite the dominance of the major GIS companies, software packages were also developed in University Labs. These packages included Yale University's Map GIS, Ohio State's Map for the PC and, perhaps most significantly, for it is the lone survivor, the Idrisi GIS software from Clark University. It was in this phase that a number of major educational initiatives were launched. In the UK, in 1985, the government set up Lord Chorley's Committee of Enquiry into the Handling of Geographic Information by computer [29]. This subsequently led to the founding of a number of regional research laboratories (RRLs) at universities throughout the UK. Soon after, the US National Science Foundation held a competition for a National Center for Geographic Information and Analysis (NCGIA). The winning bid came from a consortium of three universities: the lead was the Department of Geography at the University of California at Santa Barbara, the two supporting institutions were the Department of Geography at the State University of New York at Buffalo and the Department of Surveying at the University of Maine at Orono.

Funding from the National Science Foundation is primarily for research and the newly formed NCGIA at Santa Barbara immediately launched a number of research Initiatives. Nevertheless, researchers at the NCGIA felt that most geography departments throughout North America were severely understaffed in terms of GIS expertise often only having one or two or perhaps no faculty members who were immediately prepared to teach semester long courses in the subject. To help correct this deficiency the NCGIA set out to develop a year long, three semester course in GIS instruction. The Core Curriculum that was developed included 75 lectures, each semester having 25 courses developed by approximately 50 different professors. This original Core Curriculum was marketed in 1990 for $200 as a set of three manuals containing the 75 lectures and was accompanied by a series of laboratory exercises. The lectures were essentially notes with questions and references. The NCGIA Core Curriculum in GISystems was a great success. Eventually more than 2000 copies were purchased in over 70 countries, after being translated into at least eight languages (including Chinese, French, Hungarian, Japanese, Korean, Polish, Portuguese and Russian). The original version may still be found at: http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/toc.html . The enduring popularity of this, much dated, series of lectures is attested to by the fact that if you enter the words "spatial interpolation" into a Google search engine the first hit that you are likely to get is to Lecture 40 from the original Core Curriculum.

## 3 The NCGIA Core Curriculum in GIScience: A New Beginning

GIS is a rapidly changing field because of new software, hardware, and algorithmic and conceptual developments and as a result within less than ten years it was felt that the Core Curriculum had to be updated. It was determined that now it should go online and it should reflect the fact that the "discipline" of Geographic Information Systems had changed its name. Specialists in the field no longer wanted to teach about software systems they wanted, instead, to teach Geographic Information Science. The new name had been introduced by Michael Goodchild in an article in one of the leading journals in the field [17] and now many journals reflected this new thinking by changing the word "systems" to "science" in their titles (for example, in 1996, just four years after Goodchild published his seminal article the name of the International Journal of Geographical Information Systems was changed to the International Journal of Geographical Information Science).

The new set of Core Curriculum lectures may be found here: http://www.ncgia.ucsb.edu/education/curricula/giscc/ . If one goes to this website a number of features will be noticed. First, the new Core Curriculum is much larger with 194 lecture topics listed. Second, many topics have no lecture associated with them, only the title is listed. Third, some of the topics for which there is a lecture refer back to the original, 1990, Core Curriculum. Finally, it will be seen that the new Core Curriculum is also now a static document having not been updated since August 13th, 2000. Indeed, a note was posted on the website at that point explaining that the project had been placed in a state of "suspended animation" since December 1998. This note [6] states that the Core Curriculum would not be continued because other resources such as the Digital Library of Earth System Education [13], the Esri Virtual Campus [14] and the Center for Spatially Integrated Social Sciences [8] were all becoming available as supplementary resources that would support aspects of GIS education. NCGIA's revised Core Curriculum has not been updated since 2000. However, the CSISS website presents a cornucopia of resources and links to resources for those seeking help regarding the teaching of GIScience. Curiously, I could not find a link to the CSISS website in the UCGIS Body of Knowledge [11] that we will consider next.

Obviously the field has moved on with several generations of new software revisions to the leading GIS packages, new hardware developments, new algorithms and new ways of interacting with computer technology including web-based GIS, mobile GIS, web services, data mining technologies and the all-important rise of social media platforms such as Facebook and Twitter among others [38].

## 4 The UCGIS and the GIS Body of Knowledge

The Geographic Information Science and Technology Body of Knowledge publication [11] was an initiative of the US-based University Consortium for Geographic Information Science (UCGIS). In some sense the UCGIS was an outgrowth of the NCGIA for "in 1990 the NCGIA Board of Directors recommended that a more broadly-based organization be established to promote and support the field" [34]. Although the current UCGIS website provides almost no details of the history of the development of the organization, it was founded soon after a national conference on the issue was held in Boulder, Colorado. At the time of writing it had 75 US universities among its member organizations. Although the UCGIS exists primarily to support research and education at US universities a number of academic organizations (including the Association of American Geographers: AAG) and GIS software manufacturers (e.g. Esri, manufacturer of the industry leading ArcGIS software) have affiliate status. Non-American universities can belong to the organization as international affiliate members and presently two Austrian institutions (University of Salzburg and the Technical University of Vienna) and one Mexican umbrella academic organization (CentroGeo) have affiliate member status. The UCGIS website lists two academic and educational goals as part of its mission statement: first to "Advance research in the field of Geographic Information Science" and second to "Build scholarly communities and networks to foster multi-disciplinary GIS research and education".

To fulfill these goals and to broaden the mandate that the NCGIA Directors had advocated, the UCGIS supported a Model Curricula Task Force under the Chairmanship of Duane Marble from Ohio State University (this may explain why the NCGIA's second Core Curriculum in Geographic Information Science was essentially abandoned in 2000). The Task Force included twelve other leading GIS academics from US universities and two industry representatives from Esri. The Task Force continued in place from 1998 until 2003 when, having laid the groundwork for the Body of Knowledge, it was superseded by an Editorial Board, under the Managing Editorship of David DiBiase (at that time on the faculty of Penn State University and the Chair of the UCGIS Education Committee). This Editorial Board existed from 2004 to 2006 when it published the first and only edition of the Body of Knowledge (BoK). The BoK Advisory Board that also functioned from 2004 to 2006 included 54 of the leading US GIS academics (plus one industry expert).

The BoK (First Edition: BoK 1.0) was originally sold by the AAG (one of its publishers) but today can be downloaded from this website: http://www.aag.org/galleries/publications-files/GIST_Body_of_Knowledge.pdf. It was copyrighted by both the AAG and the UCGIS. The BoK 1.0 was essentially a list of 330 topics that were organized into 73 units and ten knowledge areas [11]. DiBiase noted that each of these topics was specified in terms of their educational outcomes so as to permit the derivation of instructional outcomes and assessment tools. It was intended to serve as a resource for curriculum design. According to DiBiase the final document incorporated the suggestions of 31 additional reviewers from the GIScience community (this was in addition to the input provided by the editorial board, the task force and the advisory board, described above).

In their introduction to the BoK 1.0, DiBiase et al. [11] recognized the role played by the original Core Curriculum which was widely tested within the GIS community before it was published [7]. However, the editors of BoK 1.0 also cited criticism of the original Core Curriculum referring to the work of Unwin and Dale [35] and Jenkins [18] who objected to the Core Curriculum's "inconsistencies and overlap, focus on content, rigid structure, and lack of a mechanism for updating" [11]. There is some irony here for the BoK 1.0 also had a rigid structure and a complete lack of any explicit mechanism for renewal. The BoK 1.0 derives much of its approach, philosophy and resource material from the UCGIS special issue on GIS Education published by the URISA Journal in 2003 [20] and edited by two of the members of the Model Curricula Task Force mentioned above, one of whom (Kemp) continued as an editor of the BoK 1.0 . The paper by Gaudet et al. [16] published in this special issue lists 39 competencies "required for success in the geopspatial technology profession". In the original article this is part of a reasonably sophisticated Geospatial Technology Competency Model although the BoK 1.0 attempts to go further than these competencies. These competencies were divided into four groups: technical, analytical, business and interpersonal.

The list of 39 competencies within these four groups is far from useful for a number of reasons. First, it lacks detail, sometimes citing whole disciplines e.g. computing programming and conflict management (the latter is the domain of a whole school or faculty within my own university). Second, it privileges two application areas, environmental and geological, over all others. One of the standard texts in the field [24] has chapters on 14 GIS application areas, and this is but a partial list. Third, even the analytical competencies have a touch of irony for Systems Thinking is included and yet there is nothing about the Philosophy or Practice of Science, which is odd considering that the entire field had the decade before replaced Systems with Science as the S in GIS. Fourth, some of the competencies are banal. Thus Leadership Skills are listed as a Core Competency followed by Questioning, whatever that is intended to mean. Fifth, with respect to the computer programming competency, the level of expertise and skill, the objectives and educational needs of GIScientists, as compared to computer scientists, are quite different. The GIScientist needs to acquire knowledge quickly so that it can be applied. Programming is best taught to a GIScientist by showing the student examples of code that configures algorithms such as those required in COGO. For example, the solution to a point-in-polygon operation requires the use of a number of coordinate geometry algorithms or “tricks”. The formulae and corresponding code can simply be shown and explained to the GIScientist and then it can be packaged “lego-like” [33] and combined into a GIS program. Essentially a somewhat similar approach is taken in Esri’s ArcGIS ModelBuilder function.

Eventually, of course, the editors of BoK 1.0 expanded Gaudet et al.’s 39 competencies (DiBiase [9] provides a brief discussion of the inadequacies of the former model) into the ten knowledge areas mentioned above (Analytical Methods; Conceptual Foundations; Cartography and Visualization; Design Aspects; Data Modeling; Data Manipulation; Geocomputation; Geospatial Data; GIS&T and Society; Organizational and Institutional Aspects), and then subdivided these into 73 units and 330 topics. The topics were up-to-date and comprehensive and were supplemented by extensive Reading Lists for each of the ten knowledge areas. Unfortunately, these reading lists were not incorporated and linked in to the topics and thus tended to be quite general and became immediately dated. Most importantly the BoK 1.0, unlike the Core Curricula of the NCGIA, did not explain how to do anything. It was not a text. GIS instructors would have to write their own lectures and, more significantly, their own lab exercises. In the age of the Internet providing a list of 330 topics that various GIS course is a useful contribution for once the terminology is known the GIS instructor can use tools such as Wikipedia, supplemented with Google Scholar searches, to build their own lecture material. Moreover, software companies such as Esri, which has its own publishing company, have been quick to supply tutorial manuals and courses (both hard copy and online) which provide the lab exercises needed to gain familiarity in solving GIScience problems using commercial, GIS software. This is true for both gaining a generic familiarity with the software (see, for example, the text Mastering ArcGIS by Maribeth Price [28]) and for gaining fluency in specific application domains such as Health GIS [21].

Although DiBiase et al. [12], Johnson [19], and Prager and Plewe [27] (all papers authored or co-authored by contributors to the BoK 1.0) have written positively about subsequent developments resulting from the publication of BoK 1.0, we have a less sanguine view which will be discussed below. Prager and Plewe’s evaluation, using two courses to evaluate the curriculum, was particularly positive: “...the Body of Knowledge enables the comparison of objectives, curriculum and actual outcomes across multiple programmes in a way that has not been possible in the past. That the domain knowledge is systematically encoded has the potential to enable the development of a disciplinary ontology…”.

The editors of the BoK 1.0 were aware that they would need to develop a path to the future, a method for continuously renewing the program that had been advocated. Consequently, the final section of the BoK 1.0 is titled: “Where is the UCGIS Model Curricula project headed?” There they suggested that “A frequently revised GIS&T Body of Knowledge…. will be needed to help educational institutions respond to needs of the dynamic GIS&T enterprise” [11]. To date this has not happened. More specifically the editors list a set of products and services that the BoK 1.0 should have helped to facilitate and these included: 1) self-assessment procedures for GIS&T degree and certificate programs; 2) exemplar pathways that GIS students could follow to pursue specific career objectives; 3) a second edition of the BoK 1.0. None of these recommendations has taken place on a discipline wide basis. Of most concern was the lack of action on a new edition of the BoK 1.0. It is this issue that we now address.

## 5 The Body of Knowledge 2.0

The editors of BoK 1.0 suggested that "a new editorial team or Task Force be empanelled within two years to consider methodologies for critique and revision that would lead to a second edition no later than 2012." A six year time line for a revised BoK was suggested because this is the average time over which renewals of the Computer Science Curriculum has taken place in the US. However, this did not occur. More recently the UCGIS organization has begun to investigate the development of a BoK 2.0 and there has been some activity to which we will now turn, although no new BoK has yet emerged.

Part of this new activity has focused on a new Geospatial Technology Competency Model [10]. This new interest has been due, in part, to the US Department of Labor's Employment and Training Administration (DOLETA) recommendations in the field of GIScience. Ten geospatial occupations are now defined, the first six of which are new: Geospatial Information Scientists and Technologists; Geographic Information Systems Technicians; Remote sensing Scientists and Technologists; Remote sensing Technicians; Precision Agriculture Technicians; Geodetic Surveyors; Surveyors; Surveying Technicians; Mapping Technicians and Cartographers and Photogrammetrists. Presumably there is not only a need to define competencies within each of these positions but also a need to develop academic pathways towards those competencies, pathways that provide an integrated and explicit curriculum. In 2012 the Urban and Regional Information Systems Association (URISA) in cooperation with DOLETA issued a brief describing a new Geospatial Management Competency Model (GMCM) [36]. This model describes the 74 competencies and 18 competency areas for managers in the geospatial industry.

These developments in the US have stimulated an interest in Europe in developing similar definitions of how to define a Core Curriculum/Body of Knowledge that characterizes GIS expertise. Perhaps the leading proponent of the use of the BoK 1.0 to inform, systematize and integrate the GIScience educational community in Europe is Frans Rip. At the 2011 LeGIO-WORKSHOP, GIS-education in a changing academic environment, Rip gave the Keynote Paper: GI-Education: The Impact of EduMapping [30] where he advocated using the categorization used in the BoK 1.0. He cited the work of Masik [25] who for her master's thesis surveyed 113 respondents from 99 universities in 27 countries (25 were European). Her results showed that 40% of the respondents were aware of the BoK 1.0, while 22% were actually using it and a further 25% intended to use it. Given European familiarity with the US research the Association of Geographic Information Laboratories Europe (AGILE) launched the EduMapping Initiative in 2009 "to characterize [the] nature [of GI Education and to make]….. courses and curricula comparable on their content" [30]. Rip's only concern was that he knew that US researchers were actively working on a BoK 2.0 and realized that the BoK 1.0 might soon be replaced. We now consider current, ongoing research on the BoK 2.0.

## 6 The Body of Knowledge 2.0: Current, Ongoing Research Initiatives

In 2010 the US National Science Foundation (NSF) funded a foundational proposal from Sean Ahearn and his collaborators to begin developing the Body of Knowledge 2.0 [1]. The proposal began by stressing the importance of GIS&T to the US economy and the role of BoK 1.0 in providing the first comprehensive ontology of the field. It then continued by explaining that the BoK 2.0 would create a "transformational, dynamic environment for pedagogy, knowledge building, discourse, collaboration, and research in GIS&T by leveraging persistent immersive synthetic environments (i.e. Second Life, OpenSim etc.) ontological analysis, knowledge mining and visualization approaches".

These researchers proposed to resolve a number of issues when designing the BoK 2.0, including: the limitations of BoK 1.0; whether to use a top-down or bottom-up design or a hybrid; how best to visualize and navigate the knowledge domain; how to reconcile collaborative contributions from a virtual community with authoritative expert knowledge; what technical and institutional mechanisms could nurture the development; and whether a perpetual virtual environment be created that would cater to different levels of expertise and to educational, research and professional needs. Three workshops were envisaged: 1) for knowledge discovery and design; 2) for visualization of the knowledge domain; and 3) to create the dynamic wikis and virtual collaborative environments. The final goal was to enable "functioning virtual communities of students, teachers, practitioners, industry, and researchers".

Although research on BoK 2.0 is in its infancy more details may be found in a presentation that Ahearn and his colleagues made at Oak Ridge National Laboratory [2]. In this presentation the approach used in developing what they refer to as GIS&T BoK 2 is beginning to come into clearer focus. The various foundational components of the system will include a core ontology (BoKOnto). This will be managed, accessed and comprehended, in part through visualization technologies (BoKVis). Contributions to the store of knowledge will come through wiki like software environments (BoKWiki). The knowledge contained within the system will be referred to as a knowledge corpus which might include knowledge artifacts such as papers, presentations, curricula, syllabi, texts, methodologies, algorithms. This level of detail will thus address one of the primary criticisms of BoK 1.0, namely its lack of detail and depth. The final component will be a reference system that allows the knowledge to be addressed easily with agents similar to Apple iPhone's Siri technology or, more attractively, with a visualized, 3D immersive environment akin to Second Life. This will be developed as a persistent virtual environment, BoKPVE. These proposed technologies link us directly to the work undertaken by Webble World researchers, which we now address.

## 7 Knowledge Federation, Webble Worlds, Pervasive Computing Environments, and Other New Technologies for BoK 2.0

Those involved in the NSF Funded research on GIS&T BoK 2.0 are to be congratulated on their visionary proposal. The use of immersive environments, such as Second Life, has great potential to enhance the GIScience learning environment. The BoK 2.0 is unlikely to be completed any sooner than 2014 and if it is to have at least a six-year shelf life then it will have to serve the GIScience community to 2020 and beyond at which time we can expect that technology will have made significant new advances. It is to these newly emerging technologies that we now turn.

We begin with a discussion of Yuzuru Tanaka's meme media technology [33]. It is quite remarkable that while Professor Tanaka has demonstrated the usefulness of his meme media technology and its subsequent technological developments in integrating GIS, the GIS community involved in the BoK 1.0 and 2.0 research appears unaware of the usefulness of his recent research in promoting the integration of a Body of GIScience knowledge. A particularly lucid discussion of the main aspects of these developments is to be found in Tanaka et al. [32]. In this paper he described how their meme media computing technologies could be used to make the Web work as a pervasive computing environment (PCE). This approach, which is now being advocated for use in the development of GIS&T BoK 2.0, is described by Tanaka and his colleagues as an open system of computing resources where developers can interactively select these resources to complete tasks. The PCE is designed to leverage a knowledge federation. A federation describes a set of digital resources that are not assumed, a priori, to be interoperable. A knowledge federation is defined as a set of document based computing resources. In Tanaka et al. [32] case studies are often web based but the authors cite previous knowledge federations of scientific simulations, digital libraries and research activities. All three, but especially the latter two, will be of considerable interest to the developers of BoK 2.0. Although there are two types of federation research, first, those that involve service providing and service consuming programs usually over the web and, second, federations that are user defined, Tanaka et al. [32] facilitate the less commonly studied user defined federations. Indeed the meme media's IntelligentPad use of slots to provide interoperability resonates with Ahearn et al.'s [2] advocacy of structuring the BoK's knowledge referencing system with identity and similarity sockets.

Examples of linking the meme media knowledge federation to legacy applications including 2D GIS and 3D geographical simulations are described by Tanaka et al. [32]. More recently Tanaka and his colleagues have extended their work on meme media into what they now refer to as Webble World technology. Webbles are meme media objects that incorporate the generic behavior common to all such objects as well as specialized behaviors specific to a given Webble but, more importantly, they are now customizable [22], [23]. It is to be hoped that extensions of meme media as described in the existing Webble World literature and new developments that will be discussed at the Webble World Summit to be held in Erfurt, Germany, June 3rd to 5th, 2013, will provide new ways in which those developing the BoK 2.0 knowledge environment can exploit this technology.

## 8 Conclusion

Our final thoughts are informed by both the needs of the North American GIScience community, and by those in Europe [30] and elsewhere and by developments in the Webble World literature.

To be successful the BoK 2.0 initiative will have to move away from the paper based hierarchical list of topics that characterized the BoK 1.0. Instead it will need to leverage the advantages of being both an online, easily accessible system using both traditional internet based technologies as well as agent-based audio advisors and immersive, visual environments. In order to be accepted the knowledge base will have to be developed by teams of researchers that are specialists in subject domains within GIScience. These subject specialists will need to develop an ontology for the discipline that can be constructed lego-like, using technologies similar to Webble World, into syllabi that can be assembled into academic and trade curricula that resolve into career pathways. Procedures for determining compliance with this ontology will also require formulation. These teams will have to be inclusive and involve academics, industry experts (e.g. Esri), and professional accreditation specialists from organizations such as URISA as well as major industry employers and national, government agencies that define job expertise and expectations (cf. DOLETTA).

Further concerns revolve around whether existing research such as Prager and Plewe's [27] investigation of the effectiveness of the BoK 1.0's ontology can be scaled up for the development of BoK 2.0 since this study only evaluated two courses. Even more significant are the organizational issues that are likely to emerge: is the BoK 2.0 best managed as a top-down or bottom-up process and how are academic contributors likely to have their efforts acknowledged and rewarded? Should the BoK 2.0 exploit the successful Research Initiative approach pioneered by the NCGIA? And how will both the interdisciplinary and multinational nature of the task be managed? Solving these issues will be of great benefit to the GIScience, Computer Science and other academic communities developing curricula that wish to find broad acceptance.

**Acknowledgement.** I would like to thank Chris Oxendine for his help and support in preparing this manuscript.

## References

1. Ahearn, S., DeMers, M., Plewe, B., Skupin, A.: Geographic Information Science and Technology BoK2: Foundational Research, NSF Proposal, `http://www.fgdc.gov/ngac/meetings/june-2010/geographic-information-science-body-of-knowledge.pdf` (accessed March 23, 2013)

2. Ahearn, S., DeMers, M., Plewe, B., Skupin, A.: Geographic Information Science and Technology BoK2: Foundational Research, NSF Cyber-GIS, Conference, September 28-30, 2011. Oak Ridge National Laboratory, TN (2011)
3. Berry, B.J.L.: Approaches to Regional Analysis: A Synthesis. Annals of the Association of American Geographers 54(1), 2–11 (1964)
4. Clarke, K.C.: Getting Started with Geographic Information Systems, 5th edn. Pearson Education, Upper Saddle River (2010)
5. Coppock, J.T., Rhind, D.W.: The History of GIS. In: Maguire, D., Goodchild, M.F., Rhind, D.W. (eds.) Geographical Information Systems, pp. 21–43. Longman, London (1991)
6. Core Curriculum, Update News (August 2013), `http://www.ncgia.ucsb.edu/education/curricula/giscc/aboutgiscc.html#news` (accessed March 23, 2013)
7. Coulson, M.R.C., Waters, N.M.: Teaching the NCGIA curriculum in practice: Assessment and evaluation. Cartographica 28(3), 94–102 (1991)
8. CSISS, Center for Spatially Integrated Social Science, `http://www.csiss.org/` (accessed March 23, (2013)
9. DiBiase, D.: Next Steps for the GIS&T Body of Knowledge, `http://proceedings.esri.com/library/userconf/educ08/educ/papers/pap2003.pdf` (accessed March 23, 2013)
10. DiBiase, D., Corbin, T., Fox, T., Francica, J., Green, K., Jackson, J., Jeffress, G., Jones, B., Jones, B., Mennis, J., Schuckman, K., Smith, C., Van Jan Sickle, J.: The New Geospatial Technology Competency Model: Bringing Workforce Needs into Focus. URISA Journal 22(2), 55–72 (2010)
11. DiBiase, D., DeMers, M., Johnson, A., Kemp, K., Luck, A., Plewe, B., Wentz, E. (eds.): Geographic Information Science and Technology: Body of Knowledge. Association of American Geographers and University Consortium on Geographic Information Science, Washington, D.C (2006)
12. DiBiase, D., DeMers, M., Johnson, A., Kemp, K., Luck, A., Plewe, B., Wentz, E.: Introducing the First Edition of Geographic Information Science and Technology Body of Knowledge. Cartography and Geographic Information Science 31(2), 113–120 (2007)
13. DLESE, Digital Library for Earth System Education, `http://www.dlese.org/library/index.jsp` (accessed March 23, 2013)
14. Esri, Esri Virtual Campus, `http://training.esri.com/gateway/index.cfm` (accessed March 23, 2013)
15. Foresman, T.W.: The History of Geographic Information Systems: Perspectives from the Pioneers. Prentice Hall, Upper Saddle River (1998)
16. Gaudet, C., Annulis, H., Jon Carr, J.: Building the geospatial workforce. URISA Journal 15(1), 21–30 (2003)
17. Goodchild, M.F.: Geographical Information Science. International Journal of Geographical Information Systems 6(1), 31–45 (1992)
18. Jenkins, A.: Through a model darkly: An educational postscript. Cartographica 28(3), 103–108 (1991)
19. Johnson, A.: UCGIS Body of Knowledge - proposed and unanticipated benefits and possible future initiatives. In: Proceedings of the EUGISES 2008 conference, Cirencester, UK, September 11-14, `http://www.eugises.eu/proceedings2008/johnson.pdf` (accessed March 23, 2013)
20. Kemp, K., Wiggins, L.: Introduction to the Special Issue on GIS Education. URISA Journal 15(1), 4–6 (2003)

21. Kurlan, K.S., Gorr, W.L.: GIS Tutorial for Health, 4th edn. Esri Press, Redlands (2012)
22. Kuwahara, M., Tanaka, Y.: Webble world – a web-based knowledge federation framework for programmable and customizable meme media objects. In: IET International Conference on Frontier Computing, Theory, Technologies and Applications, Taichung, Taiwan, August 4-6, pp. 372–377 (2010)
23. Kuwahara, M., Tanaka, Y.: Advanced "Webble" Application Development Directly in the Browser by Utilizing the Full Power of Meme Media Customization and Event Management Capabilities. In: 2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Victoria, Australia, 9-13 July, pp. 211–216 (2012)
24. Longley, P., Goodchild, M., Maguire, D., Rhind, D. (eds.): Geographical Information Systems, 2nd edn. Wiley, New York (1999)
25. Maisik, K.: The usage of UCGIS Body of Knowledge. In: AGILE Pre-conference workshop presentation, European universities, Guimarães (2013), `http://itcnt05.itc.nl/agile_old/Conference/2010-guimaraes/Workshops/Workshops%20Rip%20Agile%202010/Usage_of_BoK_Kreet.pdf`
26. McHarg, I.: Design with Nature. Doubleday, New York (1969)
27. Prager, S., Plewe, B.: Assessment and Evaluation of GIScience Curriculum using the Geographic Information Science and Technology Body of Knowledge. Journal of Geography in Higher Education 33(1), S46–S69 (2009)
28. Maribeth Price, M.H.: Mastering ArcGIS, 5th edn. McGraw-Hill, New York (2011)
29. Rhind, D., Mounsey, H.: Research policy and review 29: The Chorley Committee and "Handling Geographic Information. Environment and Planning A 21(5), 571–585 (1989)
30. Rip, F.: GI-education: the impact of EduMapping. In: Hubeau, M., Steenberghen, T., Van Balen, K., Van Orshoven, J., Vileikis, O. (eds.) Keynote Paper, LeGIO Workshop Proceedings, GIS-education in a changing academic environment, Leuven, Belgium, November 18, (2011) `https://ees.kuleuven.be/legio/Workshop_Proceedings_LeGIO.pdf#page=7` (accessed March 23, 2013)
31. Steinitz, C., Parker, P., Jordan, L.: Hand-Drawn Overlays: Their History and Prospective Uses. Landscape Architecture 66(5), 444–455 (1976)
32. Tanaka, Y., Fujima, J., Ohigashi, M.: Meme Media for the Knowledge Federation Over the Web and Pervasive Computing Environments. In: Maher, M.J. (ed.) ASIAN 2004. LNCS, vol. 3321, pp. 33–47. Springer, Heidelberg (2004)
33. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. Wiley, New York (2003)
34. UCGIS. University Consortium on Geographic Information Science website, `http://ucgis2.org/content/about-us` (accessed March 23, 2013)
35. Unwin, D., Dale, P.: An educationalist's view of GIS: Some educational and sociological concerns. Journal of Geography in Higher Education 14(2), 166–169 (1990)
36. URISA. Geospatial Management Competency Model (June 8, 2012), `http://www.urisa.org/files/GMCM%20matrix%20consensus%206-8-12-1.pdf` (accessed March 23, 2013)
37. Warntz, W., Waters, N.: Network Representations of Critical Elements of Pressure Surfaces. Geographical Review 65(4), 476–492 (1975)
38. Waters, N.: Social Networks and Regional Science. In: Manfred, M., Fischer, P. (eds.) Handbook of Regional Science, Springer, Berlin (in press, 2013)

39. Waters, N.: Representing Surfaces in the Natural Environment: Implications for Research and Geographical Education. In: Mount, N., Harvey, G., Aplin, P., Priestnall, G. (eds.) Representing, it Modeling and Visualizing the Natural Environment: Innovations in GIS 13. Ch. 3, CRC Press, Boca Raton (2009)
40. Waters, N.: Modeling the Environment with GIS: A Historical Perspective from Geography. In: Clarke, K., Parks, B., Crane, M. (eds.) Geographic Information Systems and Environmental Modeling, pp. 1–35. Prentice-Hall, Upper Saddle River (2002)
41. Waters, N.: Geographic Information Systems. Encyclopedia of Library and Information Science 63, 98–125 (1998)

# Pedagogical Patterns and Didactic Memes for Memetic Design by Educational Storyboarding

Klaus P. Jantke

Fraunhofer IDMT, Children's Media Department, 99094 Erfurt, Germany
klaus.jantke@idmt.fraunhofer.de

**Abstract.** Memetics and meme media technologies deployed for some purpose of technology enhanced learning need a certain systematization. Didactic principles, patterns of didactically driven activities, and the like may be seen as memes. Those memes are encapsulated as meme media occurring in digital representations of anticipated learning experiences named storyboards. Digital storyboarding is the preferred technology of designing anticipated learning experiences based on didactic knowledge. Encapsulated didactic memes–knowledge, principles, artifices, use cases–occur in digital storyboards and, through using, changing and re-using, may be subject to inheritance, mutation, cross-over and natural selection.

**Keywords:** memetics, meme media, didactic memes, memetic design, didactic design, pedagogical patterns, storyboarding.

## 1 Introduction

The present paper may be seen from different perspectives such as (i) didactics resp. pedagogy, (ii) patterns and pattern inference, (iii) media storyboarding, and (iv) memetics and meme media technology. The latter is crucial within the context of the Webble World Summit 2013 and, hence, will be emphasized throughout the main part of this paper. The other three aspects will be briefly considered within the present introduction and may reoccur on demand throughout the subsequent sections of the paper.

The philosophy, so to speak, underlying meme media technology widespread by publications such as [47] is derived from Richard Dawkins' seminal book [11]. Susan Blackmore is providing an introduction intentionally aiming at a much wider audience and, in some sense, explaining Dawkins' philosophy folklike [10]. The description in this paper must be necessarily much shorter and, perhaps, unsatisfactory such that interested readers are urgently directed to [11] and [10]. Richard Dawkins' fundamental message is that, apparently, there does exist non-biological evolution. Illustrative examples may be found in architecture and fashion, for instance.

Interestingly–as far as the author remembers his reading of the two books–neither Dawkins nor Blackmore address issues of didactics. This papers is aiming at filling the gap directed towards innovative solutions in educational technology.

### 1.1 Pedagogical Patterns and Didactic Memes

Since the early work by Wolfgang Ratke and Johan Amos Comenius in the first half of the 17th century, humans are pondering principles of effective teaching. The majority of principles is of some quite rough granularity and vagueness [12] which, consequently, does not allow for an appropriate digital representation. This does even apply to recent approaches which, though being called patterns, are far from the precision of patterns in other domains [42]. Dana Angluin [2] has provided an unprecedented study of introducing and exploiting formal patterns.

When some so-called pedagogical patterns are sufficiently focused and precise, they may be encapsulated and, possibly, turned into meme media objects.

### 1.2 Patterns and Pattern Inference

When the pattern concept entered scientific discourse, it was quite vague [1]. Although there has been much progress in the past [15], even recent approaches frequently remain quite uncertain as can be seen in [9] and [42], for instance.

As said before, Dana Angluin [2] has provided an unprecedented study of introducing and exploiting formal patterns. In her approach, clarity and precision are inevitable, because she is aiming at particular precise algorithmic results: algorithms able to learn patterns from examples of their instances.

It depends on the goals of scientific investigations whether handwaving is sufficient or not. When patterns are intended to play a crucial role in processes (see [16], e.g.) or when patterns are intended to reveal essential properties of processes, of human experience, and the like (see [22], e.g.), there arises a natural desire for clarity and precision.

Within the present author's work, patterns are always seen as formally well defined properties which may or may not have instances in given concrete data. Whether or not there is an instance of some pattern in some data shall be decidable (see [20], section VIII, and [23] for some more elaborate application).

### 1.3 Educational and Hypermedia Storyboarding

For several years already, the author's work on storyboarding is following the standards set in [26]. Storyboards are hierarchically structured graphs, an idea adopted and adapted from earlier work on dynamic plan generation in complex dynamic environments [3] (see also [4], [5] and, more recently, [6]).

The composite nodes are named episodes, whereas the atomic nodes are named scenes. Composite nodes may be subject to substitution by other graphs. In contrast, atomic nodes have some semantics in the underlying domain. They may represent documents such as videos, pictures, or text files in formats like pdf, e.g., but they may also represent some activities of human learners, co-learners, teachers, tutors, or those actions performed by a digital systems.

The usage of composite nodes in some storyboard graph allows for a certain remarkably declarative representation of the anticipated experiences on different levels of granularity. This relates to the idea of layered languages of ludology [36].

## 2 Towards Educational Memes

Seen from Dawkins' perspective, didactics is the science of educational memes.

Let us begin with an almost trivial, but practical case of game-based learning. Pervasive games are a very peculiar type of games [41] bearing unique potentials for game-based learning [31,32]. Before becoming involved in a case study, there is the necessity to be correct and to point to the other coin of the medal. It is apparently difficult to design pervasive games successfully, as discussed in [31]. Some games such as REXplorer [7] and Epidemic Menace [37], for instance, have been supported with large amounts of money, but failed badly. The one had been developed to be a permanent touristic game, but vanished after only three months in operation, and the other one had been played just two times. The pervasive game selected for a slightly closer introductory inspection is named Invisible Buildings being designed as a game for teaching in schools [48].

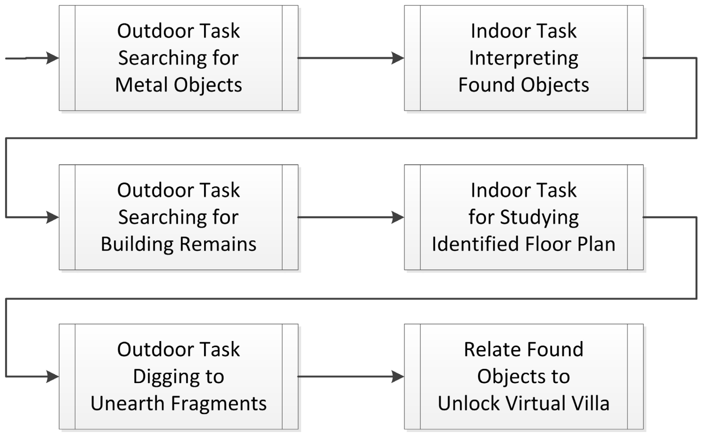

Outdoor Task Searching for Metal Objects
Indoor Task Interpreting Found Objects
Outdoor Task Searching for Building Remains
Indoor Task for Studying Identified Floor Plan
Outdoor Task Digging to Unearth Fragments
Relate Found Objects to Unlock Virtual Villa

**Fig. 1.** Top Level Storyboard for the Game Invisible Buildings

Figure 1 is showing some top level storyboard of Invisible Buildings as investigated in [31]. The representation above reveals some educational pattern, so to speak. There are outdoor activities followed by related indoor activities. In some more detail,

– there are playful outdoor activities aiming at motivation and fun providing data of a virtual world
– which are carried over from the virtual world to the subsequent session
– where, in some sense, the true learning takes place indoor by processing the data from the virtual world on a computer discussing and interpreting the virtual findings.

This way of teaching and learning may be reused in varying conditions, thus, possibly representing some meme of didactics.

For the purpose of further investigations, one may go into some more detail of playing the educational pervasive game Invisible Buildings. Finding the right level of abstraction is known to be essential, but involved [36].

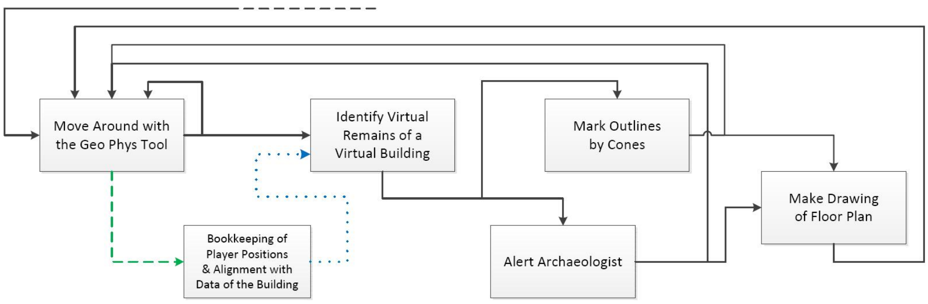

Move Around with the Geo Phys Tool
Identify Virtual Remains of a Virtual Building
Mark Outlines by Cones
Make Drawing of Floor Plan
Bookkeeping of Player Positions & Alignment with Data of the Building
Alert Archaeologist

**Fig. 2.** Storyboard of the Episode "Outdoor Task Searching for Building Remains"

The dotted and the dashed lines in the storyboard of figure 2 above indicate communications with the game server. When dealing with issues of technology enhanced learning, one may go far beyond the limits of conventional thoughts about didactic principles, rules, patterns, and the like as in [12] and [42], e.g. In e-learning, there may also occur patterns of human-machine communication.

The question for memes is the question for essential ideas that may be reused directly or that may be subject to modification or even to cross-over.

In the area of pervasive educational games, there do exist some standard ideas of when and how to deploy the communication between learners/players and the system. Clearly, the bookkeeping of player positions (see fig. 2) is crucial [31].

One may expect a larger amount of memes resulting from the efforts to manage cognitive load in technology enhanced learning [33,39,40,43,45,46]. Questions for motivation [38] and many further issues lead to memes as well. The complexity is not surprising, if didactics is the science of educational memes.

Work on pedagogy explicitly stressing the idea of having reusable templates of a somehow uniform appearance such as the so-called pedagogical patterns project [42] seems particularly relevant for pondering memes of didactics.

For the purpose of the present paper, it is particularly important to find those meme candidates which are sufficiently precise to potentially allow for a digital encapsulation as meme media using established technologies [47,35].

Patterns occur in extremely different variants ranging from [1] through [8] and [9] to [42], and all of them might be relevant to the present author's endeavor. However, patterns that are particularly vague are quite difficult to be (re-)used. "Nobody is Perfect" ([42], p. 81) and "New Pedagogy for New Paradigms" (ibid., p. 151) are extreme examples of uselessly waving around. Other concepts such as "Built-In Failure" (ibid., p. 25) are thought-provoking. Attempts to clarify what might be behind the phrase lead to some rather different implementations.

## 3 Storyboarding with Educational Memes

If didactics may be seen, in the light of [11], as the science of educational memes, didactics for technology enhanced learning deals with educational meme media.

### 3.1 Storyboarding Technology Enhanced Learning

Applied to technology enhanced learning, storyboards are the tools representing the learners' anticipated experiences at planning time. This explains the basic relationship of planning and storyboarding (see [17] and, for the background [26] on storyboarding and [3,4,5] on planning). Recent accounts of both sides that are closely related can be found in [27,34] and [6], respectively.

Very roughly speaking, storyboards are tools for anticipating forthcoming experiences. As such, they contain descriptions of what shall happen and of how those future events are related: nodes and edges between them. Naturally, one may think of forthcoming experiences on quite different levels of abstraction. More abstract thoughts and their representations are inevitable to keep long and complex processes under control. In contrast, detailed thoughts and their representations are inevitable as well for proper preparations of affective and effective experiences.

Nodes that are composite are named episodes and those that are atomic are named scenes (see [26], for a comprehensive treatment). In terms of discrete mathematics and computer science, storyboards are hierarchically structured graphs (very much like plans for dynamic applications, see sources from [3] to [6]). Scenes have a certain semantics in the domain. The semantics may be digital (documents, videos, links, and the like) and even executable. Episodes are to be substituted by (sub-)graphs. Note that repeated substitutions including limited recursion may be possible. The semantics of an episode is determined by the semantics of its constituents.

Edges between nodes determine relationships such as timely order and conditional dependence. They may also specify data flow (very much like in planning). Loops and branches are possible. Branching may have different semantics such as parallelism, choice, and the like.

In general, there is a need for annotations to clarify the semantics of nodes and edges.

Only those storyboards are taken into account (within the present paper and in related work such as [17], [20], [26], [27], [31], and [34]) which are digital objects. As such, they allow for collaborative design supported by information and communication technologies, for automatic checking of crucial properties, and for some tool-based implementation.

Practically, designing a storyboard means to take predefined units–episodes, scenes, and templates of them–from some repository and plug them together. The processing will frequently require modifications for the purpose of adaptation to current needs. Annotations for specifying semantics will be necessary as well.

When smaller objects are executable, their combination may result more complex objects implementing useful functionalities (see [13] and [14]).

## 3.2 Educational Memes in Digital Storyboards

Let us begin with an introductory example adopted and adapted from [27], fig. 3.

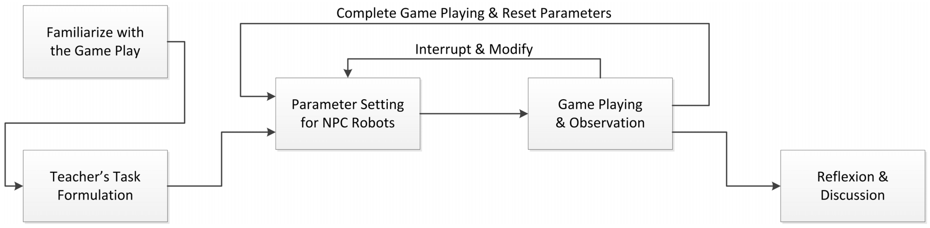

Complete Game Playing & Reset Parameters
Interrupt & Modify
Familiarize with the Game Play
Parameter Setting for NPC Robots
Game Playing & Observation
Teacher's Task Formulation
Reflexion & Discussion

**Fig. 3.** Storyboarding Extra Game Play and Meta Game Play in GORGE

GORGE is some simple browser game (see [19] for the essentials) freely available on the Internet (see http://www.kindermedien.idmt.fraunhofer.de/gorge/ for the game and [29] for a qualitative study). Both [19] and [29] are provided on this page for free download. The game runs in English, Japanese and German. GORGE bears some potential of playfully learning about Artificial Intelligence.

For playful learning, some taxonomic principles may be invoked [18,21,24]. The two principles explored in [27] are *extra game play* and *meta game play*. Extra game play means the feature of leaving the game play on purpose before completion, whereas meta game play means the feature of repeatedly playing with some purpose in mind. There are several games developed with focus on one of these or both taxonomic concepts having certain effects of game playing in mind [18].

The two loops backward in figure 3 illustrate the way in which extra game play and meta game play, resp., is anticipated in contextual game play of GORGE.

The two nested loops on display in figure 3 form (an instance of) some meme. The related storyboard sub-structure might be seen as some meme media object. For reuse in another storyboard, it may be copied and modified to meet the peculiar needs of the new application under consideration.

For the practical deployment of meme media technology in storyboarding educational processes, it is fundamental to have media objects which may be literally picked up for (re-)use, thus, becoming subject to direct manipulation [44] or, even further, to direct execution [13]. When pondering what other authors call didactic models [12] or educational patterns [42], it is decisive to find ways of digitally encapsulating ideas which might be considered memes.

Following the introductory example above, let us now undertake an attempt to use one of the so-called pedagogical patterns of [42]. "Built-In Failure" is said to be a pedagogical pattern (ibid., pp. 25-26). There is the obvious, though rather general motivation that "learning comes from experience, and much useful experience comes from failure. but a learner who lacks confidence will fear failure, and this fear impedes or even prevents learning."

This leads to the pedagogical intention to build in failure into anticipated learning processes making it a normal constituent which does not cause any learner's fear.

But what does failure mean? How does the concept of failure depend on the domain and, perhaps even more involved, on the way of teaching and learning?

In particular cases, students get a task and are asked to find some solution meeting certain criteria of quality. Note that the pattern description in [42] is not sufficiently clear to decide whether or not the following formalization meets the authors' intention. In the wordy description of the pattern, there does occur some sample about students making pottery (ibid., p. 25) in which no criteria

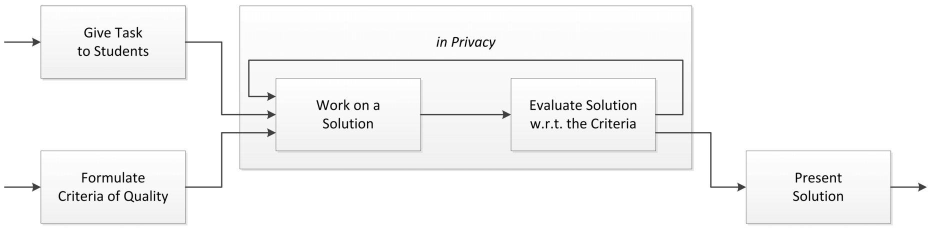

Give Task to Students
Formulate Criteria of Quality
in Privacy
Work on a Solution
Evaluate Solution w.r.t. the Criteria
Present Solution

**Fig. 4.** Storyboarding Formalizing the Pedagogical Pattern "Built-In Failure"

of quality or success are mentioned. Without criteria of success, the reasoning process changes. The corresponding cutout of a storyboard results from the one in figure 4 by simply eliminating one node. An illustration may be dropped here.

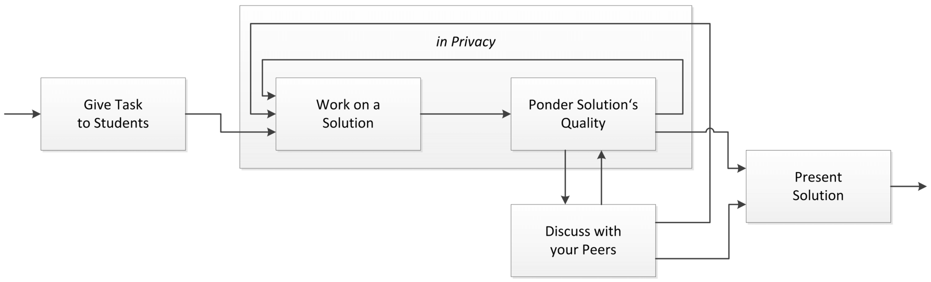


**Fig. 5.** Storyboarding some Variant of "Built-In Failure" Without Criteria of Success

Because searching for some problem's solution without any criteria of success is problematic, one is easily lead to another meme encapsulated as shown in fig. 5.

The cases discussed so far illustrate the appropriateness of building blocks allowing for an easy replication, mutation, and cross-over as supported by the current Webble Technology, for instance [35].

But meme media allow for much more than just plugging building blocks into each other. They support the propagation of data via slot connections [47] and enable direct execution [13]. Both features do not seem so particularly relevant to the few cases discussed so far within the present paper.

Among the numerous ideas and principles of collaborative learning, there is the "divide and conquer" meme: set up some task, divide it into two subtasks, let two (groups of) learners solve these tasks independently, and synthesize a solution to the original problem by combination of the solutions to the subtasks.

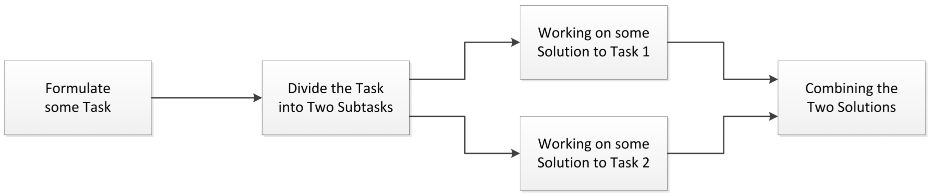

Formulate some Task
Divide the Task into Two Subtasks
Working on some Solution to Task 1
Working on some Solution to Task 2
Combining the Two Solutions

**Fig. 6.** Encapsulation of the "Divide and Conquer" Collaborative Learning Meme

For a particular learning domain, the different nodes of the storyboard cutout shown in figure 6 may have different implementations. Within some first variant,

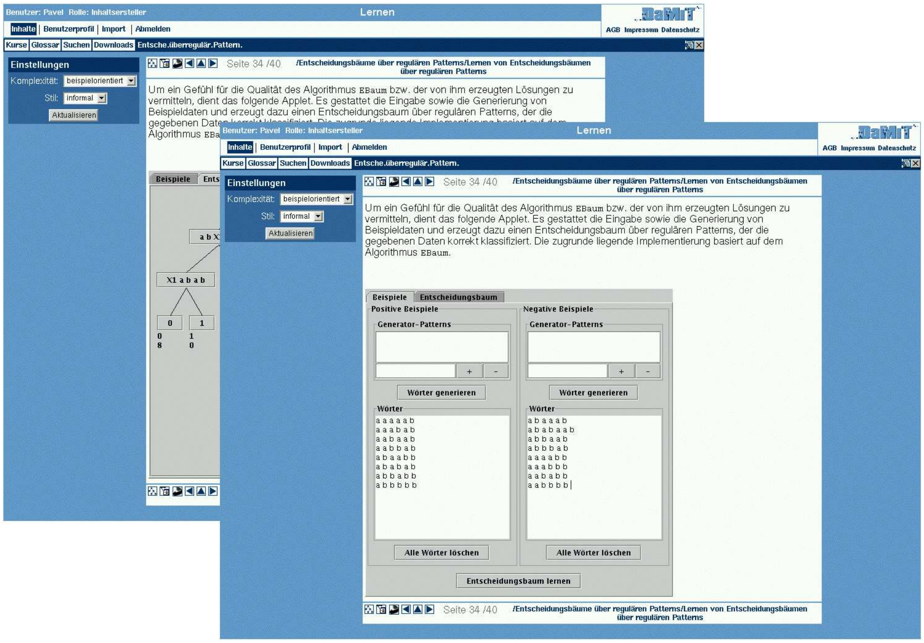
– 'Formulate a Task' is a teacher's duty,
– 'Divide the Task ...' is an activity of the whole group,
– 'Working on some Solution ...' is a task of separate individuals or groups,
– 'Combining the Two Solutions' is an activity of the whole group again.

Within a second variant of implementing the visualization on display in figure 6, only the two 'Working on some Solution ...' nodes represent human activities. All the others have a certain computational meaning. Just for illustration, there may be some task generator (maybe, reading tasks from a database), a certain procedure of dividing tasks into subtasks, and some program merging solutions of subtasks into solutions of the original tasks.

The second variant sketched does illuminate that the edges of a storyboard may have different semantics (see [26] for more in-depth discussions including the semantics of nodes). Edges do not only represent the flow of control. They specify data flow as well. The flow of data between nodes with some operational semantics may be fully automated. Assume you plug one node into another one. In this way, some slots are connected. Data flow from the slot of one meme media object to the corresponding slot of the other one according to some predefined direction. Those settings of data flow may be adjusted by hand, if necessary. In fig. 6, e.g., the rightmost node runs a merging algorithm when data are put in.

When discussing didactic concepts in [25] (see also section 4 on exploratory learning in [30]), there has been investigated some particular idea usefull for learning algorithms, their reach, strengths, weaknesses, and details of behavior.

For this idea, we introduce the term *Challenging the Target* (of learning). Figure 7 shows an application within the data mining tutor DaMiT [25,28].


Benutzer: Pavel Rolle: Inhaltsersteller — Lernen

Inhalte | Benutzerprofil | Import | Abmelden — AGB Impressum Datenschutz

Kurse | Glossar | Suchen | Downloads — Entsche.überregulär.Pattern.

Einstellungen
Komplexität: beispielorientiert
Stil: informal
Aktualisieren

Seite 34 /40 — /Entscheidungsbäume über regulären Patterns/Lernen von Entscheidungsbäumen über regulären Patterns

Um ein Gefühl für die Qualität des Algorithmus `EBaum` bzw. der von ihm erzeugten Lösungen zu vermitteln, dient das folgende Applet. Es gestattet die Eingabe sowie die Generierung von Beispieldaten und erzeugt dazu einen Entscheidungsbaum über regulären Patterns, der die gegebenen Daten korrekt klassifiziert. Die zugrunde liegende Implementierung basiert auf dem Algorithmus `EBaum`.

| Positive Beispiele – Wörter | Negative Beispiele – Wörter |
|---|---|
| a a a a a b | a b a a a b |
| a a a b a b | a b a b a a b |
| a a b a a b | a b b a a b |
| a a b b a b | a b b b a b |
| a b a a b b | a a a a b b |
| a b a b a b | a a a b b b |
| a b b a b b | a a b a b b |
| a b b b b b | a a b b b b |

**Fig. 7.** The "Challenging the Target" Meme Used in the Data Mining Tutor DaMiT

There is a target algorithm for learning decision trees with nodes consisting of regular patterns. Challenging this algorithm means to construct lists of examples and counterexamples (front screenshot in figure 7) such that the algorithm is forced to generate comparably complex trees (screenshot in the back of figure 7).

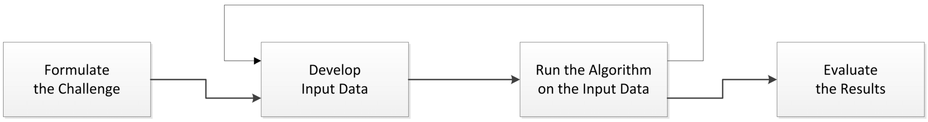


**Fig. 8.** Encapsulation of the "Challenging the Target" Exploratory Learning Meme

To implement the pedagogical concepts discussed and storyboarded throughout this paper, one needs meme media objects representing scenes and episodes which may be plugged together as illustrated by the storyboard figures above.

## References

1. Alexander, C.: A The Timeless Way of Building. Oxford University Press, New York (1979)
2. Angluin, D.: Finding patterns common to a set of strings. J. Computer and Systems Science 21, 46–62 (1980)
3. Arnold, O.: Die Therapiesteuerungskomponente einer wissensbasierten Systemarchitektur für Aufgaben der Prozeßführung. St. Augustin: infix (1996)
4. Arnold, O., Jantke, K.P.: Therapy plan generation as program synthesis. In: Arikawa, S., Jantke, K.P. (eds.) AII 1994 and ALT 1994. LNCS, vol. 872, pp. 40–55. Springer, Heidelberg (1994)
5. Arnold, O., Jantke, K.P.: Planning is learning. In: Dilger, W., Schlosser, M., Zeidler, J., Ittner, A. (eds.) GI Fachgruppe Maschinelles Lernen, Jahrestagung 1996, number CSR-96-06 in Chemnitzer Informatik-Berichte, pp. 12–17. TU Chemnitz (1996)
6. Arnold, O., Jantke, K.P., Vogler, C., Beick, H.-R., Opfermann, J.: The reach of dynamic plan generation for enterprise planning Web services. In: Proc. IET Conference on Frontier Computing, Taichung, Taiwan, August 4-6 (2010)
7. Ballagas, R., Walz, S.P.: REXplorer: Using player-centered iterative design techniques for pervasive game development. In: Magerkurth, C., Röcker, C. (eds.) Pervasive Game Applications – A Reader for Pervasive Gaming Research, vol. 2, pp. 255–284. Shaker Verlag (2007)
8. Betts, T.: Pattern recognition: Gameplay as negotiating procedural form. In: Copier, M., Kennedy, H., Waern, A. (eds.) Think Design Play: The fifth international conference of the Digital Research Association, DiGRA/Utrecht School of the Arts (2009)
9. Björk, S., Holopainen, J.: Patterns in Game Design. Charles River Media, Hingham (2004)
10. Blackmore, S.: The Meme Machine. Oxford University Press (1999)
11. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)
12. Flechsig, K.-H.: Kleines Handbuch didaktischer Modelle. Neuland, Eichenzell (1996)
13. Fujima, J., Jantke, K.P.: The potential of the direct execution paradigm: Toward the exploitation of media technologies for exploratory learning of abstract content. In: Urban, B., Müsebeck, P. (eds.) Elearning Baltics 2012: Proceedings of the 5th International eLBa Science Conference, pp. 33–42. Fraunhofer Verlag (2012)
14. Fujima, J., Jantke, K.P., Arnold, O.: Media multiplicity at your fingertips: Direct manipulation based on webbles. In: Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, pp. 217–222. IEEE (2012)
15. Jantke, K.P.: Patterns in Digital Game Playing Experience Revisited: Beitrage zum tieferen Verstandnis des Begriffs Pattern. In: Diskussionsbeiträge 22, TU Ilmenau, IfMK (2008)
16. Jantke, K.P.: The pattern experience evaluation program. In: Ligęza, A., Nalepa, G.J. (eds.) Proc. DERIS 2009: Intl. Workshop on Design, Evaluation and Refinement of Intelligent Systems, November 28, pp. 70–75. AGH University of Science and Technology, Kraków (2009)
17. Jantke, K.P.: Why storyboarding? Why not planning? In: Tadeusiewicz, R., Ligęza, A., Mitkowski, W., Szymkat, M. (eds.) Proc. 7th Conference Computer Methods and Systems, 26-27 November 2009, pp. 123–128. Oprogramowanie Naukowo-Techniczne, Kraków (2009)

18. Jantke, K.P.: Extra Game Play & Meta Game Play. Report KiMeRe-2010-03, Fraunhofer IDMT, Abtlg. Kindermedien (May 2010)
19. Jantke, K.P.: The Gorge approach. Digital game control and play for playfully developing technology competence. In: Cordeiro, J., Shishkov, B., Verbraeck, A., Helfert, M. (eds.) Proc. CSEDU 2010. 2nd International Conference on Computer Supported Education, April 7-10, pp. 411–414. INSTICC, Valencia (2010)
20. Jantke, K.P.: Logical formalization and reasoning for computerized interactive storytelling. In: Wang, Y., Luo, Y. (eds.) Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing (PIC 2010), Shanghai, China, December 10-12, pp. 851–857 (2010)
21. Jantke, K.P.: Toward a taxonomy of game based learning. In: Proc. Intl. Conference on Progress in Informatics and Computing, Shanghai, China, 2010, December 10-12, pp. 858–862 (2010)
22. Jantke, K.P.: Patterns of game playing behavior as indicators of mastery. In: Ifenthaler, D., Eseryel, D., Ge, X. (eds.) Assessment in Game-Based Learning: Foundations, Innovations, and Perspectives, pp. 85–103. Springer, Heidelberg (2012)
23. Jantke, K.P.: PCP-Spiele. Report KiMeRe-2012-04, Fraunhofer IDMT. Abtlg. Kindermedien (July 2012)
24. Jantke, K.P., Gaudl, S.: Taxonomic contributions to digital games science. In: Bradbeer, R., Ahmadi, S. (eds.) 2nd International IEEE Consumer Electronic Society Games Innovation Conference, pp. 27–34. IEEE Consumer Electronic Society (2010)
25. Jantke, K.P., Grieser, G., Lange, S., Memmel, M.: DaMiT: Data Mining lernen und lehren. In: Lernen, Wissensentdeckung und Adaptivität (LWA-2004), Fachgruppentreffen Maschinelles Lernen (FGML), Berlin, 4-6 October (2004)
26. Jantke, K.P., Knauf, R.: Didactic design through storyboarding: Standard concepts for standard tools. In: Proceedings of the 4th International Symposium on Information and Communication Technologies, January 3– 2005, pp. 20–25. Computer Science Press, Dublin (2005)
27. Jantke, K.P., Knauf, R.: Taxonomic concepts for storyboarding digital games for learning in context. In: Helfert, M., Martins, M.J., Cordeiro, J. (eds.) 4th International Conference on Computer Supported Education, CSEDU 2012, April 16-18., vol. 2, pp. 401–409. SciTePress, Poro (2012)
28. Jantke, K.P., Lange, S., Grieser, G., Grigoriev, P., Thalheim, B., Tschiedel, B.: Work-integrated e-learning – the DaMiT approach. In: Sawodny, O., Scharff, P. (eds.) 49. Conference Proceedings Internationales Wissenschaftliches Kolloquium, TU Ilmenau, 27.-30. September, vol. 2, pp. 333–339. Shaker Verlag (2004)
29. Jantke, K.P., Lengyel, D., Neumann, A., Hoppe, I.: Time to play Gorge – Time to learn AI: A qualitative study. In: Hambach, S., Martens, A., Tavangarian, D., Urban, U. (eds.) Proc. 3rd Intl. Elba Science Conference Elearning Baltics 2010, pp. 99–110. Fraunhofer Verlag (2010)
30. Jantke, K.P., Lunzer, A.: Search, comparison and evaluation in exploratory e-learning with subjunctive interfaces. In: Third Conference on Professional Knowledge Management, 1st Workshop on Learner-Oriented Knowledge Management & KM-Oriented E-Learning (LOKMOL 2005), Kaiserslautern, Germany, 2005, April 11-13 (2005)
31. Jantke, K.P., Spundflasch, S.: Storyboarding pervasive learning games. In: Proc. International Conference on Advanced ICT for Education, Hainan, China, September 20-22 (2013) (submitted)

32. Jantke, K.P., Spundflasch, S.: Understanding pervasive games for purposes of learning. In: Proc. 5th International Conference on Computer Supported Education, CSEDU, May 6-8, INSTICC, Aachen (2013)
33. Kalyuga, S.: Managing Cognitive Load in Adaptive Multimedia Learning. Information Science Reference, Hershey (2009)
34. Knauf, R., Sakurai, Y., Tsuruta, S., Jantke, K.P.: Modeling didactic knowledge by storyboarding. Journal of Educational Computing Research 42(4), 355–383 (2010)
35. Kuwahara, M., Tanaka, Y.: Webble portal (2009), `http://www.meme.hokudai.ac.jp/WebbleWorldPortal/`
36. Lenerz, C.: Layered languages of ludology – Eine Fallstudie. In: Beyer, A., Kreuzberger, G. (eds.) Digitale Spiele – Herausforderung und Chance, pp. 35–64. Verlag Werner Hülsbusch (2009)
37. Lindt, I., Ohlenburg, J., Pankoke-Babatz, U., Ghellal, S.: A report on the crossmedia game Epidemic Menace. ACM Computers in Entertainment 5(1), 1–8 (2007)
38. Malone, T.W.: Toward a theory of intrinsically motivating instruction. Cognitive Science 4, 333–369 (1981)
39. Mayer, R.E.: Cognitive theory and the design of multimedia instruction: An example of the two-way street between cognition and instruction. New Direc- tions for Teaching and Learning 2002(89), 55–71 (2002)
40. Mayer, R.E., Moreno, R.: Nine ways to reduce cognitive load in multimedia learning. Educational Psychologist 38(1), 43–52 (2003)
41. Montola, M., Stenros, J., Wærn, A.: Pervasive Games: Theory and Design. Elsevier/Morgan Kaufman (2009)
42. Pedagogical Patterns Advisory Board (ed.) Pedagogical Patterns: Advice for Educators. Joseph Bergin Software Tools (2012)
43. Salomon, G.: The differential investment of mental effort in learning from different sources. Educational Psychologist 18, 42–50 (1983)
44. Shneiderman, B.: The future of interactive systems and the emergence of direct manipulation. Behavior and Information Technology 1, 237–256 (1982)
45. Snow, R.E., Salomon, G.: Aptitudes and instructional media. Education Technology Research and Development 16(4), 341–357 (1968)
46. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science 12, 257–285 (1988)
47. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)
48. Winter, M., Pemberton, L.: Unearthing invisible buildings: Device focus and device sharing in a collaborative mobile learning avtivity. In: Parsons, D. (ed.) Innovations in Mobile Educational Technologies and Applications, pp. 77–95. Information Science reference, Hershey (2012)

# The Potentials of Meme Media Technology for Web-Based Training at the Emergency Situation Map

Sebastian Arnold and Jun Fujima

Fraunhofer IDMT, Children's Media Department, 99094 Erfurt, Germany
{sebastian.arnold,jun.fujima}@idmt.fraunhofer.de

**Abstract.** Disaster management is a very important field of civil protection. For effective protection of the population, some governmental institutions or agencies provide training in crisis management teams. An important aspect of working in a crisis management team is managing and presenting information on the current situation. It is difficult to learn the practical implementation of the information representation only from such short term training opportunities. A preliminary Web-based practice of the workflow within a realistic situation will help to solve the problem and increase the quality of the training. This paper discusses the possibility of building such a Web-based training environment for teaching and learning the situation map which is powered by the Webble technology.

**Keywords:** disaster management, emergency situation map, Web-based training, e-learning.

## 1 Introduction

Disaster management is a very important field of civil protection in Germany not least because of the background of changing climate. For effective protection of the population, it is important that the coordination of the rescue units are arranged smoothly and efficiently. In May 2004 the German federal government established the Bundesamt für Bevölkerungsschutz und Katastrophenhilfe (BBK)[1]. It is located in a division of the Bundesministerium des Inneren (BMI)[2] and takes over tasks of civil protection and disaster relief support with the consent of the most competent and highest federal authorities. As a central training institution of the BBK, the Akademie für Krisenmanagement, Notfallplanung und Zivilschutz (AKNZ)[3] takes an important part for training of pension planning and disaster management at all levels of decision-making. Through the provision of a wide variety of seminars and events, in addition to theoretical training

[1] Federal Office of Civil Protection and Disaster Assistance.
[2] Federal Ministry of the Interior.
[3] Academy of Crisis Management, Emergency Planning and Civil Protection.

and practical exercises in the area of crisis management. It offers training for the work of crisis management staffs.

An important aspect of working in a crisis management team is managing and presenting information on the current situation. It turned out in the practical exercises that the understanding of the practical work flow of the information recording and processing to the representation in a map was theoretically available, but the practical implementation was difficult. One of the reasons of the difficulty is that trainees have different knowledge about the procedure of the map management. As a consequence of these problems, the learning outcome that can be gained from the practical exercises is reduced. A preliminary web based practice simulation of the work flow within a realistically situation would help to solve the problem and increase the quality of the training.

Of course there exist different map and crisis management systems. For example, sceptros [1] developped by the "Beerware-Lizenz" allows users to manage the list of forces in use and the map very fexible. It is fully usable for map management. The second example is the eden project [2] from the sahana software foundation. It is an flexible online platform for humanitarian projects. With its wide flexible structure it can be adapted very fast to the local conditions. Fraunhofer IOSB institute is also developing systems for crisis management for example a smart control room [3] which also could be used to manage situation maps and forces via real time information collection and presentation, in response centre. These systems are designed for managing crisis situations and especially for map management, but not developed for teaching how to do this management.

This paper describes the possibility of building such Web-based training environment for learning the situation map powered by the Webble technology [4]. The Webble technology is one of the latest Meme Media [5] platform. It allows users to combine some visual objects in the Web browser window and exchange the compound objects with other users. We believe that this feature supports teachers and learners in the process of the situation map representation training well. We discuss possible scenario of Webble-based station map training and necessary functionalities to achieve the provided scenarios.

## 2 Emergency Situation Management Training in AKNZ

### 2.1 Practical Exercise in the AKNZ and Benefits

The aim of the practical exercises is to apply the knowledge about administrative and logistical structures in crisis management out of the lessons in realistic scenarios.

The diversity of the participants is on the one side good for the exchange of information about differences in other cultures and competencies. This promotes new ideas and proposals of the civil protection on both sides. But unfortunately from these circumstances results the issue that there are big differences in the state of knowledge about the civil protection and related organisation focused on Germany. However, in order to maximize the learning outcomes from the practical exercise, it is desirable that the participants of the exercise have a solid knowledge of each part of staff work.

A staff within the crisis management team in Germany consists of up to 6 disciplines and if necessary the associated consultants: Personal management(S1), situation management (S2), action (S3), materials and food (S4), public relations (S5), communication structures (S6), Psychosocial support (S7) and sifter. The allocation of staff units is related to the extent of the crisis but the minimum cast consists of S1, S2, S3 and S4. This proposal described in this paper is related to the special training unit of S2.

The responsibilities of S2 extends over, reading, interpreting and processing the incoming messages up to the final presentation of the information on the situation map "Lagekarte". So that each staff member can easily acquire information about the situation. Unfortunately, the mediation of the work flow in its current form is only in theory, therefore not sufficiently possible to visualize it. In other words, the learner must have not only declarative knowledge about facts but also procedural knowledge to use the learners declarative knowledge [6]. To improve this situation, we would like to propose an interactive training program for the situation map, called MiLK (Medientechnologie fr die Interactive Lage-Karte).

One of the targets is to improve the teachers opportunities of assembling new tasks for the students by providing interactive software. After the students solved a task, the teacher gets a message that the task is solved successfully and the collected statistic data about the action sequences during the solution process of the student. The generated tasks can be saved and reused or adapted if the teacher later wants to expand an existing scenario. After an initial phase the workload of the teacher can be reduced to a minimum. The second target is to improve the knowledge of the learner. This is to be achieved in that way that the student applies the knowledge about the duties of S2 and the tools available to him, which they have gained in the lessons and also on the online learning platform [7] in an first practical way. Through the repetition of the different tasks, learners will deepen their knowledge about the meaning of the symbols and the work flow. Similarly, this allows to generate procedural knowledge about the connections of the individual tools with the work process. This ensures that the student later in the practical exercise is familiar with the work process and not thus fail at the basic questions. As a result the students gain an increased growth of knowledge through practical exercise. In addition it is an easy way to practice work flows in realistic scenarios with the better preparation for seminars. Moreover a better cost benefit ratio for both sides will be reached.

### 2.2 What Is the Emergency Situation Map?

For a successful management of incidents, a solid information base of the crisis team is essential. The collected information is converted into a standardized form on the "Lagekarte", so as to be available for every individual staff members in the

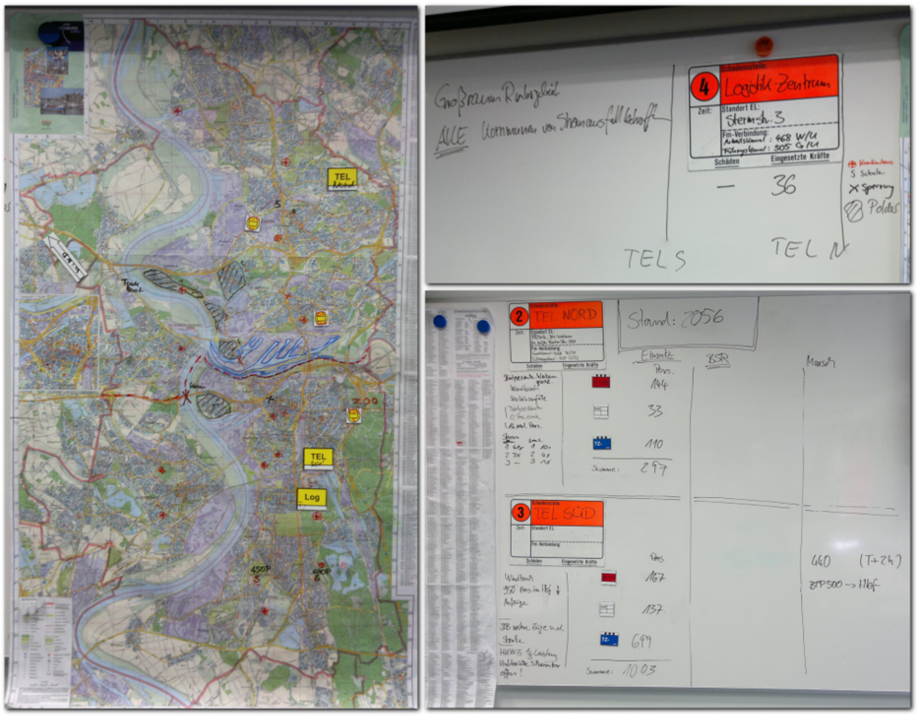


**Fig. 1.** Sample situation map of the practical exercise

crisis management team. In the picture you can see an example for "Lagekarte" (Fig. 1). The table with the forces overview is related in this special scenario to the role of S2 but normally it is the task of S1. It shows several damaged areas and the type and positioning of the individual units in the disaster areas as well as a part of the current weather situation graphically. In relation to the situation it is also possible that the weather is outsourced to a special weather map. For example in an scenario with multiple collisions the weather situation is not so important whereas a scenario with a big storm front the weather plays a major role and then the amount of information is too big for one map. The standard in catastrophic management today is the real map due to the volatile security but here are also to detect several tendencies for the usage computer-based of virtual maps or complete rooms with wide screen displays controlled via cameras and gestures [3]. For efficient work of a crisis management team, a rapid distribution of those information to the individual departments is essential for the success of the operation. It is therefore important that the map manager who holds in the staff for crisis management the position of S2 [8] transfers incoming information to the situation map as timely and properly as possible. To ensure this, it is essential that the map manager does the work as perfectly as possible.

## 3 Training Scenario for Situation Map Management

To fulfill the responsibilities, the S2 must get familiar with the following tools. The first one, the 4-way form 4-fach Vordruck which is used as a communication tool for getting various information. This form consists mainly of the recipient, the message field, the sender field and some administratively relevant information fields. Information transferred by the four-colored forms must be interpreted and represented in the situation map. To do so, the S2 have to have an extensive knowledge about tactical symbols. The following will illustrate an example of how a task is generated by the teacher and solved interactively by the learner. Following through the work flow from the beginning to the end, learners can obtain additional knowledge about the individual tools and symbols.

As an overall view of the system, it is divided in 2 main parts: one for the teacher and another for the student. For teachers the system provides modular elements available to build a fictional training task. Out of this elements, teachers can adjust the system to fit the desired teaching scenario. After creating the task the teacher safe the task so the learner can open it, or he/she send it to the learner directly. This depends on the implementation method.

The learner opens the new webble task, solves it with suitable feedback and saves or sends it back to the teacher.

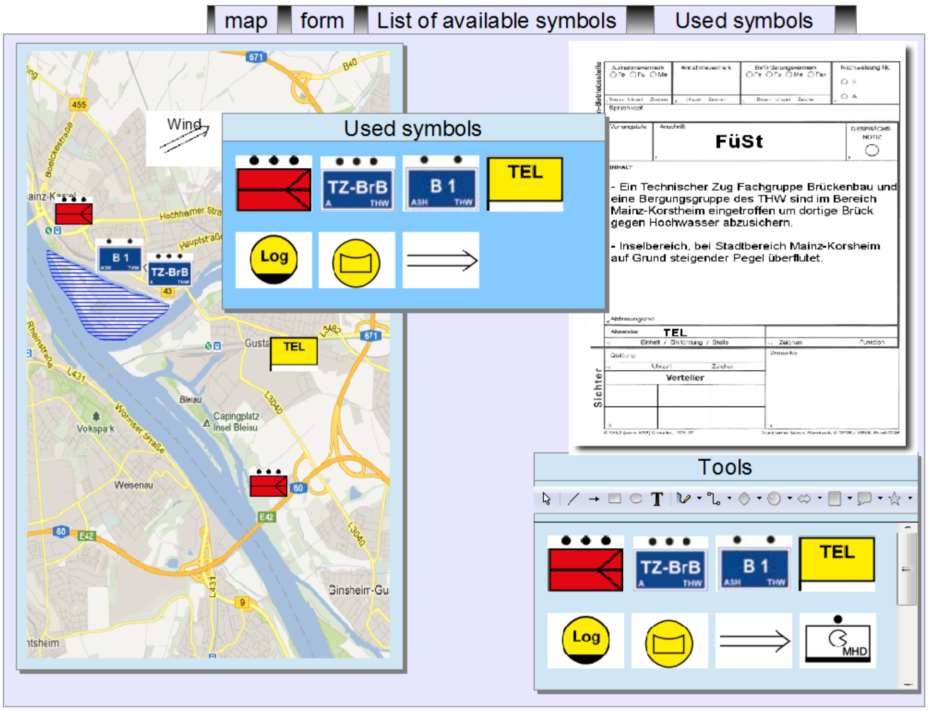
To reduce the workload for teachers, the correction of the solution is automatically done by using some programmed logics in the interactive solving process by learners. So, teachers can just monitor successfully completed tasks.

### 3.1 Example Scenario Description in General

An example scenario could be represented as follows: A teacher has a map element, a selection for tactical symbols for building a situation map, and a "4-fach Vordruck" form which he/she describes a situation. Also another element can be used for a brief summary of the current situation. This could be used as part of a fictional situation briefing to summarize the situation. The individual elements can be scaled individually and their contents are adapted automatically. If a user drags one element and drops it over another, the dropped element is embedded onto another element. In this way, learners can put the elements together to complete a task. Each workspace has a task bar allowing to view hidden elements. It will be extended automatically when a new element appears on the surface. The following part will explain the parts of the interface in detail.

### 3.2 Teachers Interface

In Fig. 5 you can see a sample presentation of the working surface for the teacher. At the beginning of creating a new task, the teacher creates a new incoming message out of a template on a 4-fach Vordruck. As similar to a real situation, it would serve as the starting point for the work of S2. In this template he/she can change all fields. In this case he/she fills in the sender, the content of the message, and the receiver.



FüSt

- Ein Technischer Zug Fachgruppe Brückenbau und eine Bergungsgruppe des THW sind im Bereich Mainz-Korstheim eingetroffen um dortige Brück gegen Hochwasser abzusichern.

- Inselbereich, bei Stadtbereich Mainz-Korsheim auf Grund steigender Pegel überflutet.

Absender TEL

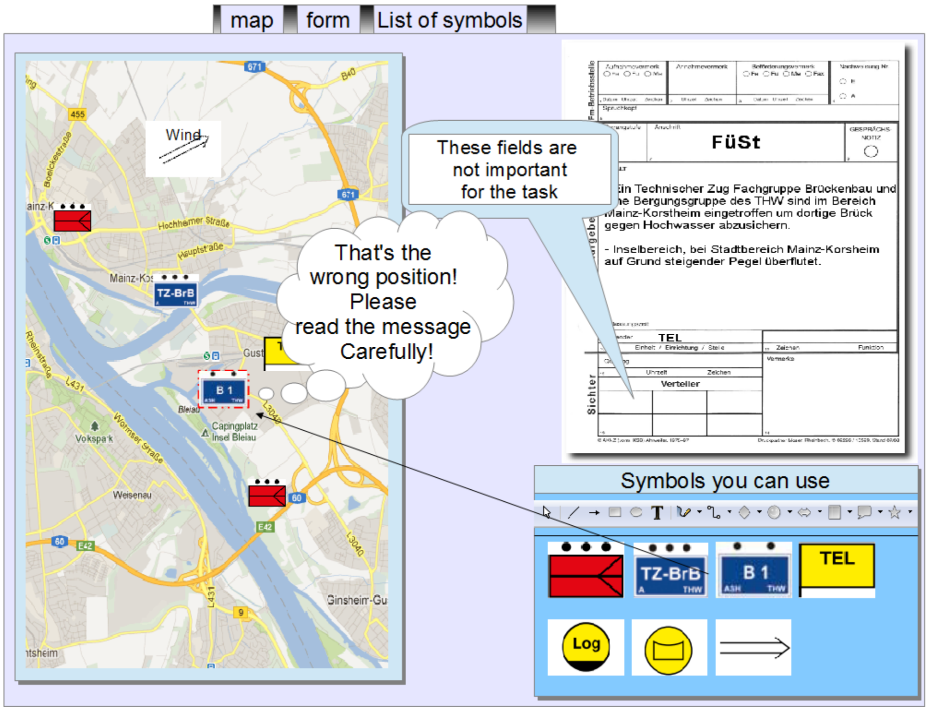
**Fig. 2.** Teacher's view of the workspace

In conjunction with the tactical symbol element, named as "Tools", he/she creates a graphical presentation layer using drag-and-drop operations. For this purpose, he simply pulls the required symbols from the pool of available symbols to the desired location on the map. The symbols which are placed on the map are automatically added to a new element which includes the available signs for the learner. He also can expand the list of symbols in "Used symbols" for the learner with additional symbols out of "Tools" which are not placed on the map via drag and drop. This is one example to increase the difficulty level for the learner. Additionally he/she can mark damaged areas like flooding on the map with hand drawings which are also located in "Tools".

After finishing the compilation, he/she stores the task. So it is published for the learner.

### 3.3 Learners Interface

In this example the student simply open the generated task like you can see in the picture and the workspace will be opened (Fig. 3). Now he gets displayed the "4-fach Vordruck", "Lagekarte", and a collection of symbols he can use to solve the task in his working area. An additional message may comes up for


map
form
List of symbols
Wind
TZ-BrB
B 1
TEL
Symbols you can use
Log
FüSt
These fields are not important for the task
That's the wrong position! Please read the message Carefully!

Ein Technischer Zug Fachgruppe Brückenbau und die Bergungsgruppe des THW sind im Bereich Mainz-Korstheim eingetroffen um dortige Brück gegen Hochwasser abzusichern.

- Inselbereich, bei Stadtbereich Mainz-Korsheim auf Grund steigender Pegel überflutet.

**Fig. 3.** The learner's workspace with some feedback examples

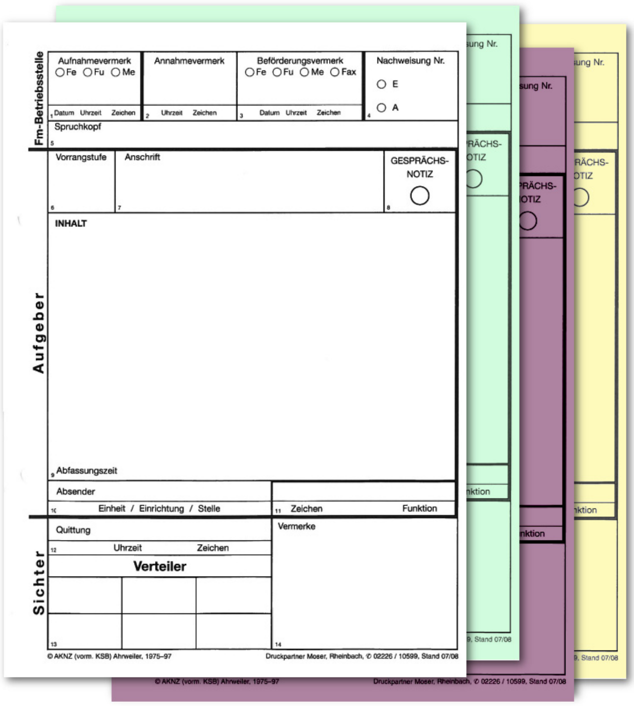
fields which are not filled or not necessary for the task. The learner starts to represent the information on the "Lagekarte" which are contained in the "4-fach Vordruck". For this, he only has to place the needed symbol with drag and drop from the symbol box to the places on the map, which are described in the message. The flooded area can be marked with the drawing toll. If the student made some mistakes in this point, he get provided some feedback depending on the chosen symbol, the number of attempts, the location on the map, and the relative position to other symbols. Finally the goal of the learner is to solve the problem correctly. This can be done by trial and error or conscientious. When this is achieved, he gets a feedback and it will be saved automatically and marked as resolved, then sent back to the teacher. So, if it is desired it can provide a overview over the solution method of the learner.

## 4 Technical Requirements

The sample scenario presented in the previous section consists of the 3 most important elements for the work of S2: "4-fach Vordruck", "Lagekarte", and tactical symbols. In this section these are described in detail. It is divided in an explanation of the presented possibilities and the required data collection as well as transmission and further additions which are not included in the example. General all relevant information to the learning progress are collected in a statistic which is provided to the teacher after finishing the task.

## 4.1 4-fach Vordruck

All elements in the above-mentioned “4-fach Vordruck” are customizable by the teacher to allow the students to get the greatest variety of scenarios. If the learner as in this example, has to concentrate only on the main task as a map leader, it is allowed to the teacher to only fill in the sender, content of message and the receiver. All further empty fields will be marked as not necessary from the teacher or from the system after a request to the teacher. If the teacher has forgotten to insert something, the system notifies to him and he cannot proceed until he decides to fill in now or later. If the teacher thinks it is necessary that the learner has to identify mistakes in the form, he can fill these items and mark as not visible. In this case, the learner must explicitly identify the exact content. In the case of failed attempts partial solution will be displayed to him. This flows also back into the overall statistics of the user actions. Another method may be to leave the field blank so that the learner needs to be aware that something is missing. This could e.g. be used for the first exercise units.


Fm-Betriebsstelle

1 Aufnahmevermerk ○ Fe ○ Fu ○ Me — Datum Uhrzeit Zeichen
2 Annahmevermerk — Uhrzeit Zeichen
3 Beförderungsvermerk ○ Fe ○ Fu ○ Me ○ Fax — Datum Uhrzeit Zeichen
4 Nachweisung Nr. ○ E ○ A
5 Spruchkopf

Aufgeber

6 Vorrangstufe
7 Anschrift
8 GESPRÄCHS-NOTIZ ○

INHALT

9 Abfassungszeit

Absender
10 Einheit / Einrichtung / Stelle
11 Zeichen Funktion

Sichter

Quittung
12 Uhrzeit Zeichen
Verteiler
13
Vermerke
14

© AKNZ (vorm. KSB) Ahrweiler, 1975–97 Druckpartner Moser, Rheinbach, ✆ 02226 / 10599, Stand 07/08

**Fig. 4.** 4-fach Vordruck

**Additions of "Vordruck".** It is conceivable to extend the logic of the form so that the content of the text is checked in real-time and searched on the basis of specific keywords. That enables us to automatically create a list of symbols which are used and place it automatically on the map. In order to further extend this, it would be possible to implement a temporary position of the symbols on the location map. They can be taken from from inferences from the context of the message. Another possibility is that the teachers can link via drag and drop new words for a symbol with the elements placed on the map if he uses those words that are not yet known to the system. Thus, in the next use of the word, the appropriate symbols are positioned in the quick selection.

## 4.2 Lagekarte

The teacher can choose a map out of the pretended scenario maps. If it is desired to adapt the environment to a new location, the teacher can incorporate additional maps. For example, if they are presented in digital form he can drag and drop the image file out of the explorer in to the map element. All positioning data are saved in relation to scaling and the size of the image so that it is zoomed in and out. The individual symbols, drawn on the map, are automatically associated with the position on the map and stored with a unique ID. The icons can be moved by the teacher at any time and the data is updated. On the part of the learner, the symbols may vary depending on the difficulty and according to initial position. The task will be achieved by moving the symbols to the right position, which is interpreted from the incoming message. In this case(Fig. 3), the wrong or right position is indicated by a constantly updated green or red border around the icon. The time for solving, relative position of symbols, movement of symbols and scaling of the map are recorded to analyze the behaviors of the student in detail.

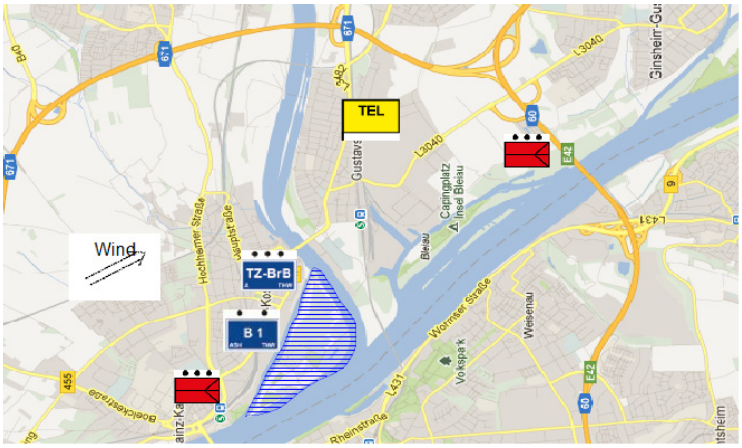


**Fig. 5.** Example situation map

**Extensions of Lagekarte.** Existing map services like Google Maps can be used to dynamically adapt the content. It could also be extended to take the places out of the map and the analyzed text in the message to generate a preliminary positioning of the elements to the map. To realize this, a complex implementation of a map service may be needed.

### 4.3 Symbol Elements

The symbols to be placed on the map include a unique ID and the icon which is displayed on the element. In addition, they contain location data that is transferred when users place it on the map. This should have a relation to the scale factor of the map, because the position may be different depending on the scale on the map. The positioned tactical symbols can have additional information, such as name of the unit, strength and equipment, and aisle information which some electronic map management systems provide. From this data, all unit-lists will be generated depending on the circumstance by the scope of S2. The generation of the list could also be done manually. This content will be displayed in a new table element next to the map, and the learner enters the values of the units which are appearing in the text. This is only possible if the icon was already placed on the map. Thereby the sequence of actions is trained. Here the learner gets also a feedback, for example, to prevent him from filling in swapped numbers. At the same time the descriptions on the map will be also updated. This possibility with unit list is related to which environment the learner should be trained. These are only some possibilities for the information recording processing and representation of such application.

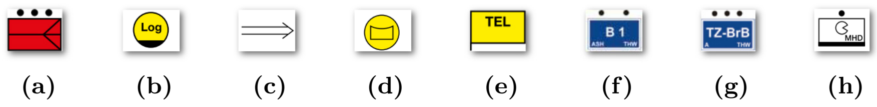


**Fig. 6.** Example of symbol from recommendations for tactical signs in civil protection: (a) platoon of firefighter, (b) logistic, (c) event direction, (d) collection site, (e) technical operations management, (f) rescue team THW[4], (g) technical division platoon with bridge building THW, (h) supply squad of Malteser

## 5 Conclusion and Outlook

We all should be aware that civil protection in Germany and last but not least around the world is becoming more and more important. To ensure the best possible protection of the public we need a variety of volunteers and material but also the corresponding know how. The AKNZ represents in this context, an important institution for the mediation of this know-how.

[4] Technisches Hilfswerk: `http://www.thw.de/EN/Homepage/homepage_node.html`

To further enhance the quality of education of the helpers in civil protection, this paper proposes a training system for constructing an interactive emergency situation map to provide a small contribution to the huge challenge of disaster management and civil protection.

On one hand, this training system provides an easy to customize environment for teachers to present an easy to use tool to teach the work of the map leader.

On the other hand, it is an easy way for the learner to interactively become familiar with the procedures of the work processes of S2 and to deepen and further develop the existing knowledge. Thus the student is better prepared for the challenges in practical exercises. This should lead to a more efficient flow in practical exercises. The resulting gain in time can be used to develop the scenarios, which in turn lead to a more optimized preparation for the reality. All described ideas are implementable on Webble World platform.

Each of these elements can exchange data with other elements. It is possible to evaluate the the learners solution without human control. Thus the work for teachers is reduced, which is an advantage for managing a large number of students. In the end, this type of training system is not only a useful interactive interface, but represents a communication tool between teachers and students, allowing to check and exchange the flow of information on the state of the learner automatically.

**Acknowledgments.** We would like to acknowledge the AKNZ. We participated in the 4th International BBK Summer Academy in 2012. This was a big inspiration for this work.

## References

1. Adler, P., Adler, F., Dipl, M.P.: dipl., dr. phil., dipl. Release of Sceptros (2013)
2. Sahana Software Foundation: The Sahana Eden Essential Guide (2012)
3. Wirth, S.: Presseinformation zur cebit 2011: Leben in gefahr it fr das katastrophenmanagement (2011)
4. Kuwahara, M., Tanaka, Y.: Webble world portal (2009), `http://www.meme.hokudai.ac.jp/WebbleWorldPortal/`
5. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press (2003)
6. Anderson, J.R.: Language, memory, and thought. Lawrence Erlbaum (1976)
7. BBK: Aknz online learning platform (2013), `http://www.bbk-virtuelle-aknz.de/index.php`
8. Ansschuss für Feuerwehrangelegenheiten, Katastrophenschutz und zivile Verteidgung (AFKzV): Führung und Leitung im Einsatz, FwDV 100 (1999)

# Meme Media and Knowledge Federation: Past, Present, and Future

Yuzuru Tanaka

Meme Media Laboratory, Hokkaido University, Sapporo, Japan
tanaka@meme.hokudai.ac.jp

**Abstract.** This paper gives a historical survey of our meme media research since 1987. Its origin was the idea of the synthetic media architecture. Then in 1993 came the reorientation of the architecture toward the meme media architecture, focusing on the international sharing, reediting, and redistribution of various knowledge resources. This was followed by our efforts to merge the meme media architecture with Web technologies, and made us come across a new aspect of meme media technologies, i.e., knowledge federation which focuses on how to enable people to extract arbitrary functions from open Web applications and/or Web services and to make them interoperate with each other for improvisational composition of a complex application from available open Web resources. Finally, this paper gives future perspectives on meme media in the context of cloud computing and exploratory big data analytics.

**Keywords.** Meme Media, Knowledge Federation, Synthetic Media, Meme Pool, Web Resources.

## 1 Introduction

The history of meme media research started with an idea of the synthetic media architecture IntelligentPad in 1987 and its implementation in Smalltalk 80 in 1989 [1]. Then the IntelligentPad Consortium was organized by major IT companies including Fujitsu, Hitachi Software Engineering, Fuji Xerox, NTT, and NEC in 1993. In this year we had TED (Technology Entertainment and Design) Conference in Kobe, where we reoriented our media architecture to meme media and meme pool architectures [2][3], and showed its live-demonstration. The dissemination of Web browsers after 1995 made us consider how to use the Web as a world-wide meme pool, i.e., a repository of meme media objects. Around the same time, consortium member companies released commercial versions of IntelligentPad developed in C++. Then we came across the problem of how to wrap a Web resource like Web applications or Web services into a meme media object. Once being wrapped, such resources can be easily combined together to interoperate with each other. This idea opened a new vista of knowledge federation, especially improvisational knowledge federation, i.e., dynamic federation of knowledge resources over the Web [4]. This idea, however, required two different system environments, i.e., the Web environment and a meme media system environment. An ideal solution to avoid the use of more than one environment was to unify these two environments, which resulted in the proposal of the WebbleWorld system in 2010 [5]. This paper will review the history of meme media and give new perspectives of meme media and knowledge federation technologies. Chapter 2 describes the origin of the idea about synthetic media architecture. Chapter 3 will describe the reaction of industries, immediately after our proposal of synthetic media, for applying this idea to the rapid development of client systems. Chapter 4 will describe the counterbalancing reflection from the academic side against such reactions by industries, and the resultant proposal of meme media architectures. Chapter 5 will describe the new aspect of meme media called improvisational knowledge federation to extract some functions from more than one Web applications and/or Web services and to federate them to work together with other meme media components for composing complex applications. Improvisational knowledge federation based on meme media technologies has enabled us to extensively utilize open Web resources including data sets, application tools and services by making them interoperate with each other together with other meme media tools. Chapter 6 will describe the Web-based meme media system WebbleWorld which has successfully merged meme media architecture with Web technologies. Chapter 7 and 8 will explain two new applications of WebbleWorld both of which deal with big data and requires exploratory visual analytics of them. One is the integrated IT support of the clinico-genomic trials on cancer, and deals with clinical trial data, patient genomic data, and patient DICOM image files. The other is the social cyber-physical system for the optimization of snow plowing and removing in Sapporo, and deals with both retrospective and real time probe car data and probe person data, meteorological sensor data, traffic accident records, complaints from citizens, and snow plowing and removing operation records. Chapter 9 will give our future perspectives on meme media technologies in the context of cloud computing and big data analytics.

## 2 Synthetic Media Architecture as the Origin

The history of meme media research started with the idea of synthetic media architecture in 1987 and its implementation in Smalltalk 80 as IntelligentPad in 1989 [1]. The basic idea came from the observation that both multimedia documents and visual tools like a hand calculator share the same type of composition structures on a screen. Each of them has a base pad with a set of embedded pads on it. Each embedded pad may again be a composite pad with a base pad and its embedded pads. The first observation may conclude that there are also some difference between the two, i.e., component pads of a tool may have some functions, while those of a multimedia document have no functions. However, each component of a multimedia document on a screen also has a function, like a text-editing function of an embedded text pad, or an image-editing function of an embedded image. Therefore, we can conclude that, from an architectural point of view, there are no difference between multimedia documents and tools. Both consist of functional pads pasted together to define nested structures which are purely hierarchical. The hierarchical structure among component pads defines a parent-child relationship between a pad and another pad embedded in it. These pads may interoperate with each other. For example, a number button pad embedded in a calculator base pad sends its assigned one digit number to the base pad, while a display pad of a calculator changes its display value when a new value is calculated by the base pad. An image pad embedded in a text pad may change its position in the text whenever the text is updated. The change of its value or its position may be both interpreted as the state change of the child pad. Each parent-child connection may be associated with such kind of interoperability between pads. Based on these ideas, we tried to develop a virtual sheet of paper called a pad with some dedicated functionality.

Based on the standard design pattern for GUI objects, we made a decision to define each pad as an MVC triple, where objects M, V, C respectively represent its model, view and controller. The model defines its internal mechanism with its state, while the pair of its view and controller defines the display object of this pad, namely how it is visually represented and how it behaves against user events on it. Furthermore, we simplified the standard MVC architecture by removing a direct message sending link from C to M. Figure 1 shows a simplified MVC used to implement each pad.

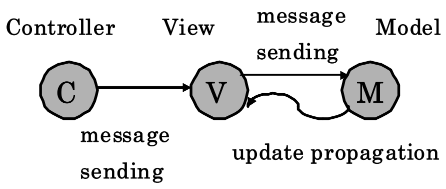


**Fig. 1.** A simplified MVC used to implement each pad

The next decision we made was the MVC-based representation of each composite pad. Figure 2 shows the standard MVC representation of a window with embedded sub windows. In this representation there is a clear difference between the base window and its sub window. The former has its model, while the latter has no model.

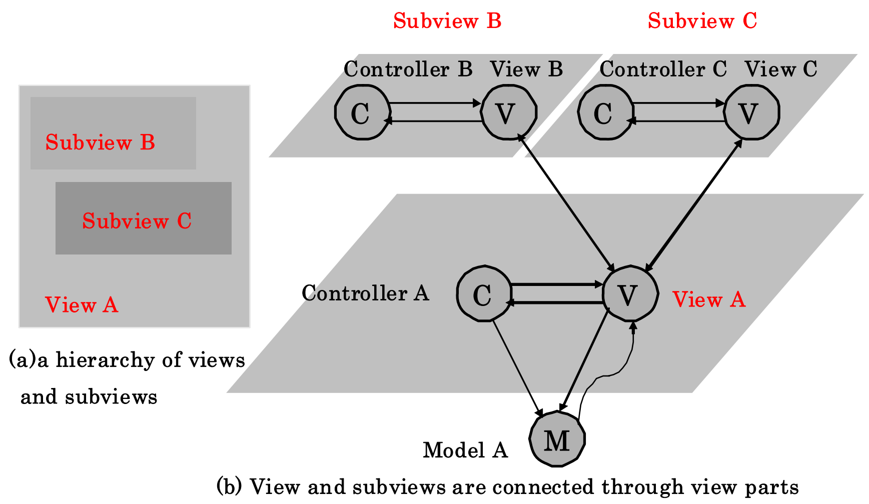


**Fig. 2.** The standard MVC representation of a window with embedded sub windows

Our goal is to establish a uniform representation of pads without distinguishing sub pads from base pads. A sub pad should be able to be used as a base pad of another composition, and vice versa. Therefore, we chose the architecture shown in Figure 3 for composite pads.

There were still some alternative choices on how to establish each parent-child connection. Figure 3 uses the view connection. The controller connection may not work since there is no way to propagate the state change of the parent model to its controller, and hence to the child pad. The model connection seems to work. We did not choose this because of a different reason related to the implementation of shared copies.

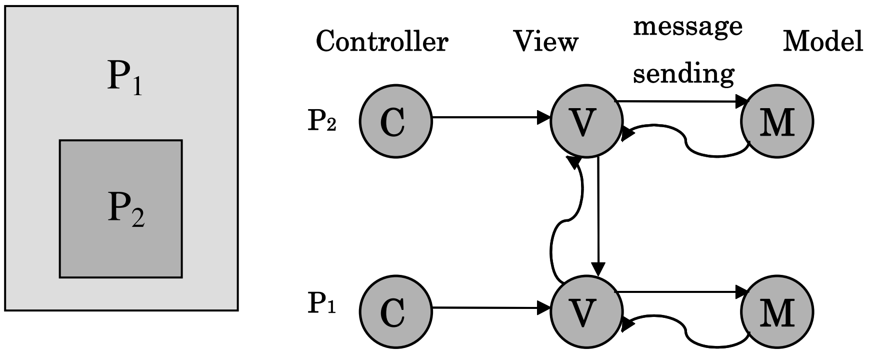


**Fig. 3.** MVC representation of a parent-child relationship between two pads

A shared copy of a pad share the same internal state with its original but may have an independent position and size from its original copy. This implies that two shared copies of the same pad share the same model but have independent display objects as shown in Figure 4. Each of these shared copies should be able to accept child pads. This implies that parent-child connection cannot be model connection but should be

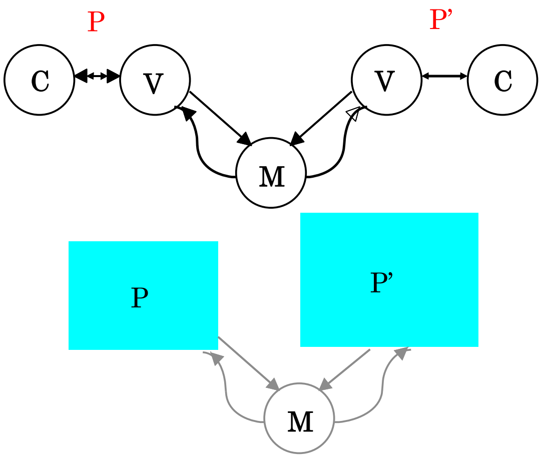


**Fig. 4.** Shared copies of a single pad

view connection. A shared copy of a composite pad can be defined as shown in Figure 5. Each pad or composite pad embedded in a shared copy of a base pad can be interpreted to work as a modification of the view of this copy.

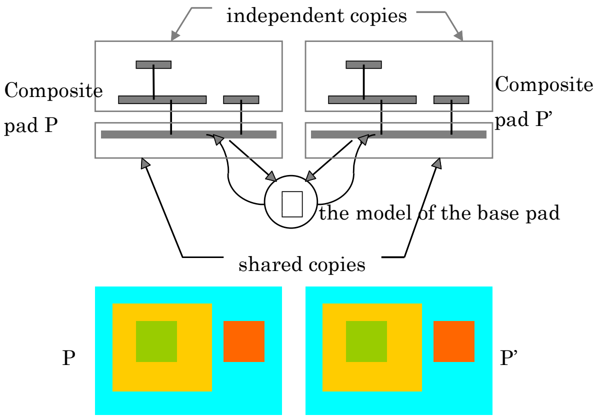


**Fig. 5.** Shared copies of a composite pad

In order to make each pad pluggable, i.e., be coded independently from the pad to which it is connected, each reference to a specific function of the pad to be connected should be specified in an indirect way. Such a reference in the coding should internally use a parameter, which is bound to the specific function of the pad to be connected when the connection is defined at composition time. The specific function of the pad to be connected is identified by a pair of the pad identifier and its function name. Our synthetic media architecture represents each function of a pad as its input and/or output port, which we call its slot. Each slot is identified by its name. In order to access a slot, our architecture provides two standard messages, i.e., "set" and "gimme" messages. The "set" message may take two parameters, i.e., a slot name and a value to be sent to the slot. The "gimme" message takes one parameter to specify a slot. These two messages when used in coding of a pad need not specify their slot parameters. When a pad is pasted on another pad, the system asks the user to which slot of the parent pad it is connected. The user can just specify one of the slots of the parent. This slot of the parent pad is called the connection slot. The slot-name parameters of a "set" message and a "gimme" message are automatically added as the name of the connection slot when these messages are issued to the parent pad. The parent-child connection also uses the "update" message which notifies the update of the parent pad to all of its child pads. This message takes no parameter. The default usage of the set message is to send a value from a pad to its parent pad, while the default usage of the other two messages is to get the new value of the connection slot whenever its parent pad is updated. Figure 6 shows a pad connection with standard messages between two pads.

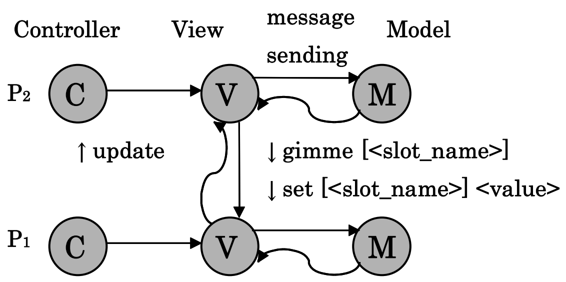


**Fig. 6.** Standard messages used for the interoperation between a child pad and a parent pad

Our synthetic media architecture also provides other types of interoperability among pads. Drag-and-drop of a pad on another pad may invoke the target pad to process the object pad in some way like making its copy or analyzing its construction structure to visualize.

## 3 Synthetic Media System as a Toolkit for Developing Client Systems

Around 1993, major IT companies like Fujitsu, Hitachi Software Engineering, Fuji Xerox, NTT, and NEC became interested in using the synthetic media system IntelligentPad for developing client systems, and established a consortium. For this purpose, Fujitsu and Hitachi Software Engineering jointly developed a C++ version of IntelligentPad system in collaboration with our group in 1995. Several years later, US-based venture company K-Plex released another commercial version Plex Ware. The accumulation of reusable components as pads in each company gradually formed a sufficiently large library, and allowed its application developers to use IntelligentPad as a toolkit for constructing each application of some typical type within a significantly reduced time, say one thirtieth of the conventional way of development. They also reused a common construction structure of composite pads developed for similar applications. Such a common construction structure for a specific type of applications, together with a set of reusable pads for the same type of applications, forms its application framework. Figure 7 shows a typical application framework for form interface systems of databases [6]. It consists of a database pad which communicates with a database system, several buttons, a query input pad, and a record pad which holds one of the records retrieved from the database. A record pad has several child pads, each displays a text, a number, or an image depending on which slot of the record pad it is connected to. This application framework is nothing but what was provided by each major database management system as a form construction kit in mid 90s.

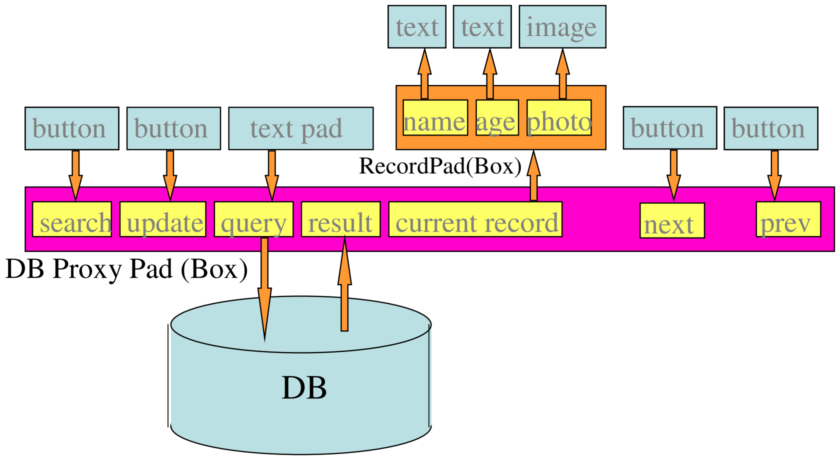


**Fig. 7.** A typical application framework for form interface systems of databases

Around the same time, we in the academy made another challenge to extend synthetic media architecture from two dimensional representation media to three dimensional representation media [7]. This extension was named IntelligentBox. While, in IntelligentPad, each child pad is clipped by its parent pad, each child box in IntelligentBox is bound by the coordinate system spanned by its parent box. We were especially interested in its application to interactive 3D information visualization [8][9] and interactive virtual studio for easily making computer animation movies using interactively controllable virtual movie cameras [10].

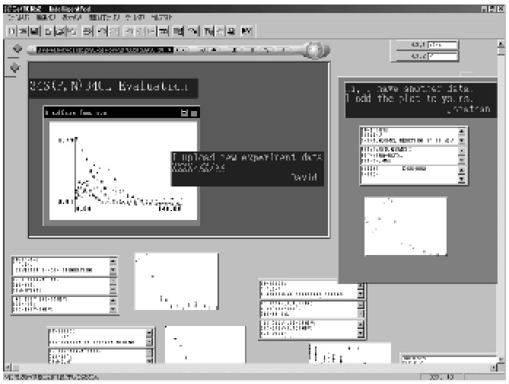
## 4 Meme Media Architecture as Reorientation of the Architecture

The present author's motivation of developing IntelligentPad was different from the goals of these companies. It was to accelerate the repetition of the "plan-do-see" cycle for supporting creative thinking. Once we come up with a new idea, we will make a plan, then try to execute this plan, and observe and analyze the result to conceive a new idea. What makes the repetition of this cycle take more time than our thought process is the difficulty of executing a plan. The execution of a plan with a computer system requires the development of a new application to check the result or to analyze something. His original motivation of developing the synthetic media architecture was to speed up this phase of developing a new application on demand by enabling people themselves to easily combine reusable pads to compose complex applications as composite pads without writing a single line of code.

While the same technology had also revolutionized the software development of client-side applications especially when it was used with appropriate application frameworks, our group in Hokkaido University became interested in the world-wide sharing and reusing of knowledge resources represented as composite pads around early 90s, and named this aspect of synthetic media "meme media" [2][3] after the concept of memes introduced by Richard Dawkins in his “Selfish Gene” [11]. He pointed out the similarity between biological evolution and cultural evolution. Ideas, for example, are replicated for reuse by other people, recombined to generate a new idea, transferred with some errors replacing some portions of the originals, and finally evaluated by some community to survive or disappear. These correspond to the replication, recombination, mutation, and natural selection of genes in biological evolution. Therefore he coined a new word “memes” to denote a cultural counterpart of genes, i.e., imaginary units for carrying information or knowledge, by combining “gene” and “mimesis”. While memes are imaginary vehicles of information and knowledge, meme media are real vehicles carrying externalized information and knowledge. The meme media concept was first proposed with its live demonstration in May 1993 at TED (Technology Entertainment and Design) Conference held in Kobe. This was the first TED held outside US. Meme media requires not only component-based media architecture but also a world-wide repository of composite pads to and from which people can publish and retrieve composite pads. We named such a repository “meme pool” [12].

Since our proposal of meme media was earlier than the release of Mosaic ver. 1.0 Web browser in November 1993, we developed a meme pool called “piazza” independently from the later Web technologies (Figure 8). After the rapid increase of Web users, however, this independence from the Web technologies brought in a serious problem of how to migrate Web resources as well as users’ legacy local resources into a meme media system environment. We also realized that it is much easier to develop a meme pool system based on Web technologies [12].

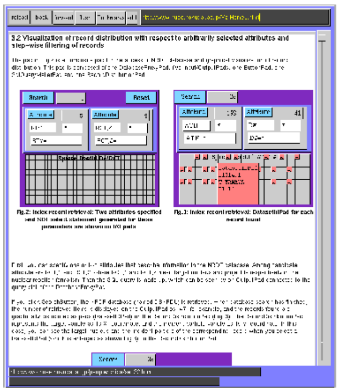
Our first attempt was to use Mosaic browser to embed any composite pad in an HTML Web page (Figure 9). The embedding was done by using a special file type to recognize a file of a composite pad for automatically downloading its save format representation and converting it to a composite pad. This composite pad is put on the HTML Web page at a specified location as a floating object which is not actually embedded in this HTML document but moves on this page in synchronization with the page scroll as if it is embedded in the page. The HTML document reserves an empty space on which this pad can sit.



[illegible] have another data.
I add the plot to yours.
[illegible]

I upload new experiment data
[illegible]
David

**Fig. 8.** Piazza working as a meme pool of pads


3.2 Visualization of record distribution with respect to arbitrarily selected attributes and step-wise filtering of records

[illegible]

Fig.2: Index record retrieval: Two attributes specified and SQL select statement generated for these parameters are shown in IO pads

Fig.3: Index record retrieval: DatasetListPad for each record found

[illegible]

**Fig. 9.** Use of Mosaic browser to embed any composite pad in an HTML Web page as a floating object over this page

For the migration of open Web resources including Web applications and Web services into a meme media environment, we developed wrapper technologies to wrap them into pads. The wrapping of a Web service is to develop a blank-sheet pad which works as a proxy objet of the Web service using SOAP or restful Web service protocol. It was not a difficult task to develop a wizardry system which asks users only to specify some parameters of a target Web service for automatically creating its

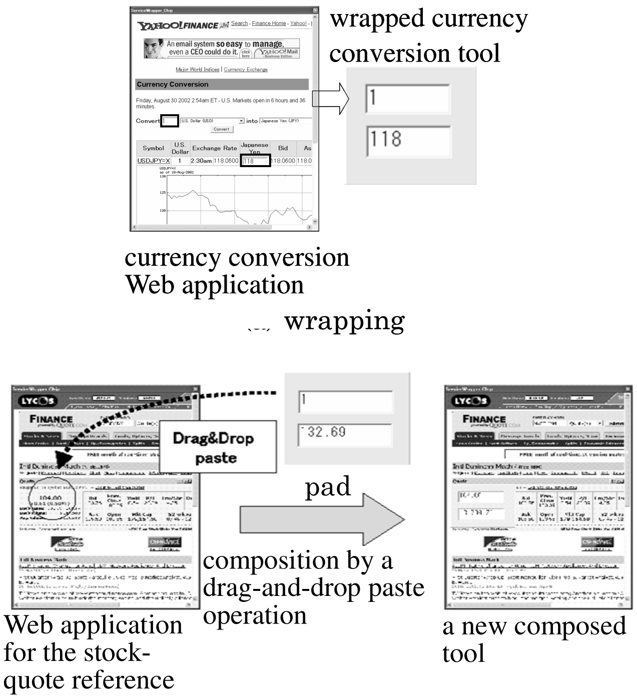

wrapped currency conversion tool
1
118
currency conversion Web application
wrapping
Drag&Drop paste
1
32.69
pad
composition by a drag-and-drop paste operation
Web application for the stock-quote reference
a new composed tool

**Fig. 10.** Chip framework for the knowledge federation of more than one Web application

proxy pad. The wrapping of Web applications, however, required a special technology. Our group developed two wrapping technologies, Chip in 2003 [13][14] (Figure 10) and C3W in 2004 [15][16] (Figure 11). Both of them enable us to specify some input forms of an arbitrary Web application and some portions of its output page to work as respectively input ports and output ports, and to create a pad embodying the function with these IO ports extracted from this Web application. These technologies are applicable to any Web applications whose output pages are not generated by Java script.

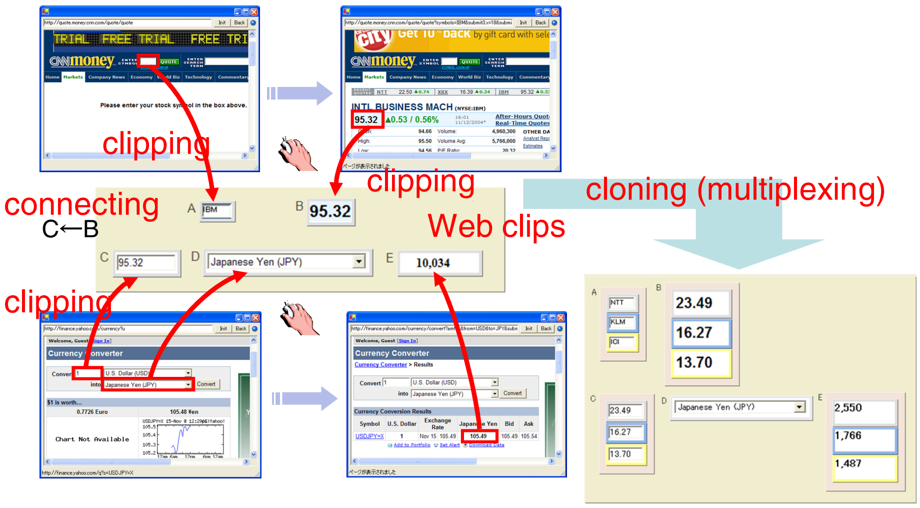


**Fig. 11.** C3W framework for the knowledge federation of more than one Web application

## 5 Knowledge Federation

Around the same time we observed a new demand of federating existing data sets, tools and services both in complex enterprise systems and open Web environments. Big enterprises were suffering from how to integrate applications, that were developed independently for different purposes, to keep the integrity. Data integrity was actually the main purpose of having introduced database management systems for the integrated management of data across different departments. While DBMS guarantees the data integrity at the data storage level, use of PCs at the client side made it easy to develop a huge number of applications for different purposes without considering the integrity across different data processing applications. Furthermore, some applications should refer to the output of other applications, which requires the coordination of more than one application interoperating with each other. The open Web environment, on the other hand, significantly increased reusable data sets, applications, and services that are open for public use. Even a conventionally closed application system may consider the use of some related open data sets, applications, or services to enhance its evidence-based analysis and/or decision making.

The required interoperability in these situations cannot be achieved by the system integration approach in which each component is initially designed to be integrated with other components in the way initially designed. The above situations require interoperability among data sets, applications, and services that are not initially designed to be integrated into a single system. Such interoperability requires federation of objects instead of their integration. The present author coined the word "knowledge federation" in 2004 to denote the achievement of interoperability among knowledge resources including data sets, applications, and services, that are not initially defined to interoperate with each other, without rewriting the code of any of these resources [4][17]. Later, the word "knowledge federation" turned out to denote not only such achievement of interoperability among knowledge resources but also semantic reorganization of knowledge resources and even the social federation of people's knowledge [18]. We distinguish the first two as functional knowledge federation and semantic knowledge federation [19]. The third one is referred to as social knowledge federation.

Our wrapping technologies to extract some portion of a Web application or a Web service and to wrap it into a pad enabled us to federate Web applications and/or Web services with other pads. Since both Fujitsu version and K-Plex version of IntelligentPad could also wrap any Microsoft Office objects into pads, these two IntelligentPad systems had achieved to provide a knowledge federation framework for resources represented as pads, Microsoft Office objects, Web applications, and Web services. Since the extraction and wrapping of various knowledge resources can easily be performed on demand without any coding, we called such functional knowledge federation as improvisational knowledge federation.

## 6 WebbleWorld to Merge Meme Media Architecture with Web Technologies

With our new framework for improvisational knowledge federation of resources based on meme media technology, we began to look for its complex applications. Around that time, we were invited to participate in a new EU's FP6 Integrated Project ACGT (Advancing Clinico-Genomic Trials on Cancer) as one of the 26 teams. The participation in EU projects, however, has some regulation. We cannot bring in any commercial product for system development. Therefore, we made a decision to develop a fully open meme media system. We also wanted to unify the meme media architecture with de fact standard Web technologies. As mentioned above, our knowledge federation of Web resources was performed in the meme media environment which is different from the Web environment. In order to publish a composite application obtained as a federation of knowledge resources into the Web, we need to convert the composite pad to a Web application defined in HTML. We called this translation "the flattening". If we can develop a meme media system only using de fact standard Web technologies, such a system can be considered as an extension of the Web to the memetic Web. WebbleWorld was thus developed [5]. The current WebbleWorld exploited the Microsoft Silverlight framework to save the development cost, which however introduced two major problems, i.e., dependence on the Miceosoft's future strategy on Silverlight and poor cross platform compatibility. In 2013, we have started to develop a new HTML5 based WebbleWorld system which will become available soon.

Figure 12 shows the architectural features of WebbleWorld. It is a Web-top meme media system. Composite pads or webbles run in a Web page. Therefore, the extracting and the wrapping of a Web application function can be done inside the Web without even temporarily going outside the Web. Therefore, we call this extension the memetic Web.

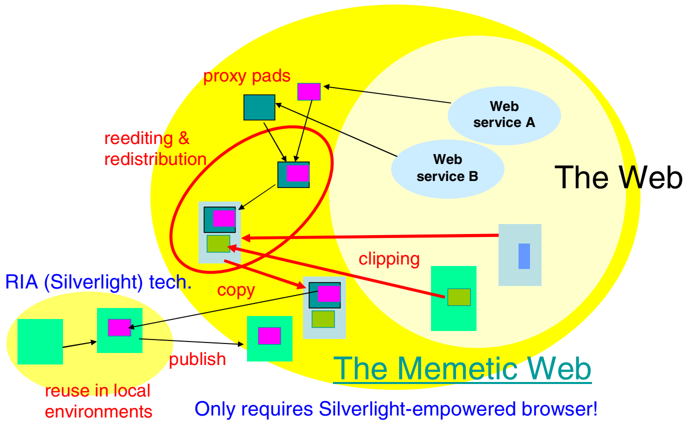


**Fig. 12.** The architectural features of WebbleWorld

## 7 Application to the Integrated Support of Clinical Trial on Cancer

In the ACGT project from February 2006 to July 2010, we developed TOB (Trial Outline Builder) jointly with Norbert Graf, the Director of the Clinic for Pediatric Oncology and Hematology at Saarland University [20]. TOB is an integrated visual environment for (1) designing the CRF (Case Report Form) of each therapeutic event, (2) designing a trial master plan as a flow diagram by copy-and-pasting each event to draw a flow diagram, (3) using a single path in the flow diagram to guide both the treatment of a specific patient and the data input to the CRF of each event, and (4) supporting visual analytics of accumulated patient data after the completion of the trial. Figure 13 shows these second, third, and fourth stages of TOB system. The trial master plan starts with the stratification which corresponds to the registration of each patient to the trial and the dispatch of the patient to an appropriate treatment path depending on his or her case. Each treatment path is a sequence of different types of therapeutic events like radio therapy events, chemotherapy events and surgery events. Each path may have diagnostic events at some points of the flow. Some treatment path may have a randomization followed by more than one treatment branch. Each patient dispatched to such a path is randomly dispatched to one of the branches called treatment arms after the randomization. A trial master plan also has some number of adverse events outside its whole flow. These correspond to the treatment events conducted by clinicians at some emergency by neglecting the planned flow. One of the main purposes of the visual analytics with TOB, for example, is to find out specific patient cases for which some treatment arm after the randomization may show better survival rate than other arms. In order for analysts to find out such cases, the visual analytics stage of TOB provides a parallel coordinate system with various visualizations of trial data including the life table. Each coordinate in the parallel coordinate system represents an entity of the CRF of some treatment event in the master trial plan. Each polygonal line in such a parallel coordinate system represents an individual patient. Each coordinate allows us to select only those patients whose value of the corresponding entity falls between two arbitrarily specified boundaries. Users can specify more than one pair of such boundaries in each coordinate. Suppose that an exploratory filtering of patients in the parallel coordinate system results in a life table visualization in which one treatment arm after the randomization shows better survival rate than others. Then we may conclude that this treatment arm should be chosen for those patients currently selected by the parallel coordinate system. This leads to the so-called personalized medicine. Our knowledge federation framework based on the meme media architecture enabled us to rapidly develop TOB which accesses the trial data stored in the ObTiMA database developed by Fraunhofer IBMT. We easily wrapped ObTiMA database into a pad. Since the WebbleWold is a Web top meme media system, TOB was easily embedded in the Web portal of ObTiMA.

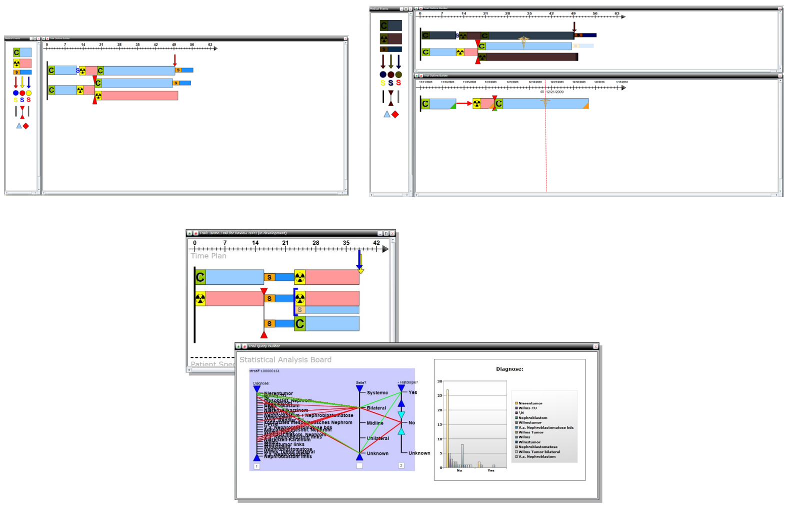


**Fig. 13.** TOB showing its second, third, and fourth stages

The improvisational knowledge federation capability of the meme media architecture also enabled us to easily extend TOB by bringing in the reference to patient genomic data and DICOM image files. Figure 14 shows the heatmap visualization of gene expressions of some set of genes for a specified set of patients. This visualization also interoperates with the patient selection by the parallel coordinate system. Such an ad hoc introduction of new components with their interoperation with existing components can be implemented just by defining a new parent-child relationship with an appropriate slot connection.

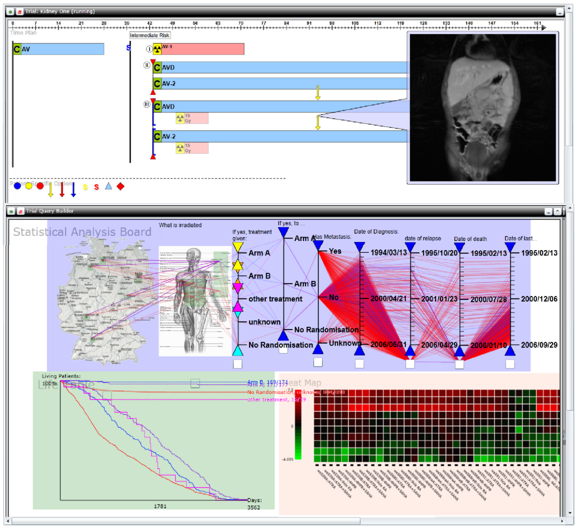


**Fig. 14.** TOB with the heatmap visualization of gene expressions and a patient's DICOM image

## 8 Application to Social Cyber-Physical Systems for Optimizing Social System Services

Around 2010, the present author was asked by MEXT (Ministry of Education, Culture, Sport, Science and Technology) of Japan to make a new R&D Plan for the next generation IT infrastructure technologies for optimizing social system services. This turned out to be a one year feasibility study project in 2011, and then in 2012 a five year government initiative project on social cyber-physical system technologies for optimizing social system services. This is a joint project with NII (National Institute of Informatics), Osaka University, and Kyushu University. Our team picked up the snow plowing and removing in Sapporo as a complex social system service. Sapporo has 1.9 million people and annual snow fall of 6 meters, and spends more than 15 billion yen every year just for snow plowing and removing.

Our preparatory study showed that macro analyses of probe car data for the whole city area would not give any meaningful knowledge about the influence of snow fall, snow plowing and removing on traffics [21]. Clustering analyses of road segments with respect to their traffic changes in a day told us that even those segments along the same route may fall in different clusters on a day with heavy snow fall. In summer time, they almost fall into the same cluster. This means that the influence of snow on each road segment may follow different models even on the same route. Therefore, we may think that influence of snow on traffic in the whole city cannot be modeled by a single monolithic model covering the whole city. It should be considered as a complex system of different models each of which may be a simple monolithic model. In order to obtain meaningful knowledge from such a complex system, we first need to find out a set of subsystems modeled by the same monolithic model or same type of models. This situation is quite similar to the analysis of clinical trial data, in which it is very important to find out a specific case of patients, for which one of the treatment arms after the randomization shows a significantly better performance than the others, to find out personalized medical treatments.

Here we also exploit an exploratory visual analytics approach. Figure 15 shows the geospatial digital dashboard system we developed for the exploratory visual analytics of the probe car and probe person data in comparison with time, area, snow fall, snow plowing and removing, traffic accidents, and weather conditions. Figure 15 has a 24 hour clock on which you can specify any time interval to retrieve those probe car data during this time interval. It also has a calendar with the weather data of each day. On this calendar, you can specify only those days with heavy snow. It also shows a map where probe car data and traffic accident records can be visualized, and you can specify an arbitrary area of your concern. Each chart on the right hand side show, for example, the distribution of average speed, car population distribution, and accident distribution, all in terms of the number of road segments. On each distribution chart, you can specify a certain range of values to choose only those road segments falling into this range. The geospatial digital dashboard system enables us to dynamically change the quantification conditions on each of the various parameters as well as the spatiotemporal conditions to find out a set of road segments that are similarly influenced by snow fall, and/or snow plowing and removing. The obtained set will be used for further analyses based on frequent pattern mining and/or further clustering.

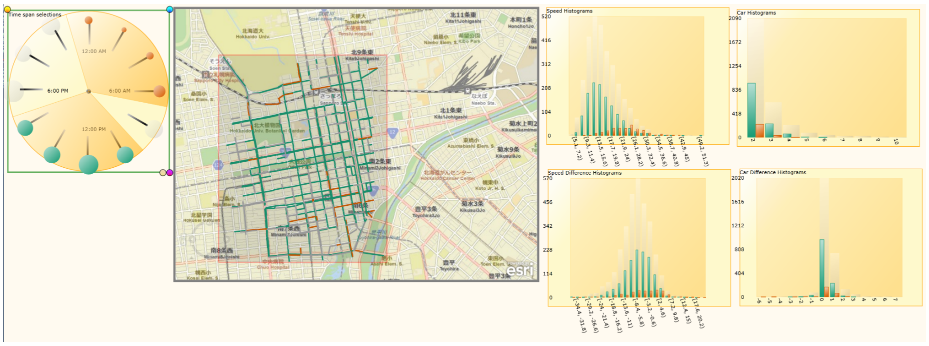


**Fig. 15.** Geospatial Digital Dashboard System with a 24 clock, a map visualization, and charts

## 9 Concluding Remarks with Future Perspectives

Our new HTNL5 based WebbleWorld which will become available soon will work as a potential platform for people to edit and publish Web documents with embedded complex interactive tools and services, to reuse and copy some of such embedded tools published in other people's Web documents, to federate them or their copies for composing new complex knowledge resources, and to publish them as Web applications. Knowledge resources thus represented as Web documents are cross platform compatible. The Web and its search engine will work as the meme pool of such knowledge resources and the search service for people to find out appropriate knowledge resources or webble components embedded in some Web pages. They can be semantically organized and managed using the Semantic Web technologies. From technological point of view, we can utilize all the available Web technologies to develop various utilities for editing, publishing, reorganizing, managing, and searching for knowledge resources. From users' point of view, we can enrich Web documents by embedding complex interactive visual applications in them. A Web page or an e-book page defined with this technology may have an embedded picture like Figure 14 or 15, which is not actually a picture but an interactive application environment for you to play with. You may make a copy of a chart in Figure 15 embedded in some Web page, and use it in the TOB environment like Figure 14 embedded in some e-book page with its connection to some of the entity of the CRF of some treatment event in TOB.

As explained in preceding chapters, a webble may work as a client-side graphical user interface of a Web service. A parent-child relationship between two of such webbles may be implemented in either of the following two alternative ways. The connection defined at the client side may be used for the actual data communication between the two Web services. The same connection may be also considered as a referential link used only for invoking the direct data communication between the two Web services without going through two webbles. The latter mode of parent-child connections enables us to use the HTML5 version of WebbleWorld as a GUI builder system for cloud computing systems. WebbleWorld will enable users to easily federate various services provided by a cloud computing system to compose a complex visual application environment as a Web portal without writing a single line of code.

Through our involvement in three major research projects, EU's FP6 ACGT project and the following FP7 project p-medicine, and the Japanese government initiative project on the social cyber-physical system technologies for optimizing social system services, we recognized that real-world big data cannot be simply analyzed just by today's advanced data mining and clustering algorithms based on some single monolithic modeling of the target world. These projects told us that real world data are much more complex and require more than one different type of monolithic modeling that form a network of mutual interaction. In the analysis of such a system, it is important to first find out each hidden subsystem that can be modeled by a single mathematical model. This implies the importance of taking the exploratory visual analytics approach instead of directly applying a single analysis algorithm to the whole system. Visual analytics requires a large library of both visualization tools and analysis tools that can interoperate with each other. Because of the evolutional nature of research-oriented data analyses, such a library should not be a closed one but should be open for future extension. Therefore, the import of open tools and services published over the Web into the visual analytics environment is fundamental to keep the system up to date. The wrapping technologies, especially the generic wrapper technologies, play the important role for importing open resources.

The last but not the least topic of knowledge federation is its security. Knowledge federation enables us to easily compose a phishing site by federating an original Web application with an interception service. In order to prevent such evil attempts, we need to develop a secure federation technology, the R&D of which is the most important challenge for meme media and knowledge federation to make significant contributions to the next generation ICT.

## References

1. Tanaka, Y., Imataki, T.: IntelligentPad: A Hypermedia System allowing Functional Composition of Active Media Objects through Direct Manipulations. In: Proceedings of the IFIP 11th World Computer Congress, San Francisco, USA, pp. 541–546 (1989)
2. Tanaka, Y.: From Augmentation to Meme Media. In: Proc. of ED-Media 1994, pp. 58–63. Vancouver (June 1994)
3. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)
4. Tanaka, Y., Fujima, J., Ohigashi, M.: Meme Media for the Knowledge Federation Over the Web and Pervasive Computing Environments. In: Maher, M.J. (ed.) ASIAN 2004. LNCS, vol. 3321, pp. 33–47. Springer, Heidelberg (2004)
5. Kuwahara, M., Tanaka, Y.: Webble World – A Web-based Knowledge Federation Framework for Programmable and Cutomizable Media Objects. In: Proc. of the IET International Conference on Frontier Computing Theory, Technologies and Applications, Taichung, Taiwan, IET inspec, August. 4-6, pp. 372–377 (2010)
6. Tanaka, Y.: Integration of Synthetic Media and Databases. In: Proc. of ADTI 1994, Nara, pp. 292–305 (1994)
7. Okada, Y., Tanaka, Y.: IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications. In: Proc. of the Computer Animation 1995 Conference, Geneva, pp. 114–125 (1995)
8. Ohigashi, M., Tanaka, Y.: From Visualization to Interactive Animation of Database Records. In: Arikawa, S., Nakata, I. (eds.) DS 1999. LNCS (LNAI), vol. 1721, pp. 349–350. Springer, Heidelberg (1999)
9. Sugibuchi, T., Tanaka, Y.: Database Visualization Framework Based on Relational data Model. In: Proc. of the 13th European-Japanese Conference on Information Modelling and Knowledge Bases, Kitakyushu, vol. 105, pp. 300–312 (2003), Information Modelling and Knowledge Bases XV, Frontiers in Artificial Intelligence and Applications, vol. 105, pp. 282-294 (January 2004)
10. Okada, Y., Tanaka, Y.: IntelligentBox: Its Aspect as an Interactive Animation System. In: Proc. of SCI99/ISAS99, Orland, July-August, pp. 198–201 (1999)
11. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)

12. Tanaka, Y.: Meme Media and a World-Wide Meme Pool. In: The Fourth ACM International Multimedia Conference, pp. 175–186. Boston (November 1996)
13. Itoh, K., Tanaka, Y.: A Visual Environment for Web Application Composition. In: Proc. of 14th ACM Conference on Hypertext and Hypermedia, Nottingham, pp. 184–193 (August 2003)
14. Tanaka, Y., Itoh, K., Kurosaki, D.: Meme Media Architectures for re-editing and redistributing intellectual assets over the Web. Internationall Journal of Human-Computer Studies 60(4), 489–526 (2004)
15. Tanaka, Y., Ito, K., Fujima, J.: Meme Media for Clipping and Combining Web Resources. World Wide Web: Internet and Web Information Systems 9(2), 117–142 (2006)
16. Fujima, J., Lunzer, A., Hornbæk, A., Tanaka, Y.: Clip, connect, clone: combining application elements to build custom interfaces for information access. In: Proc. of the 17th Annual ACM symposium on User Interface Software and Technology UIST 2004, Santa Fe, pp. 175–184 (October 2004)
17. Jantke, K.P., Lunzer, A., Spyratos, N., Tanaka, Y. (eds.): Federation over the Web. LNCS (LNAI), vol. 3847. Springer, Heidelberg (2006)
18. Karabeg, D.: Knowledge Federation, `http://knowledgefederation.ning.com/`
19. Tanaka, Y., Fujima, J., Kuwahara, M.: Meme Media and Knowledge Federation. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 2–21. Springer, Heidelberg (2008)
20. Sjöbergh, J., Kuwahara, M., Tanaka, Y.: Visualizing Clinical Trial Data Using Pluggable Components. In: Sjöbergh, J., Kuwahara, M., Tanaka, Y. (eds.) Proc. of the 15th International Conference on Information Visualization IV 2012, Montpelier, pp. 291–296 (July 2012)
21. Tanaka, Y., Sjöbergh, J., Moiseets, P., Kuwahara, M., Imura, H., Yoshida, T.: Geospatial Visual Analytics of Traffic and Weather Data for Better Winter Road Management. In: Cervone, G., Lin, J., Waters, N. (eds.) Data Mining for Geoinformatics: Methods and Applications, Springer, Heidelberg (to appear 2013)

# Author Index