Webbles: Programmable and Customizable Meme Media Objects in a Knowledge Federation Framework Environment on the Web

Micke N. Kuwahara, Yuzuru Tanaka

Meme Media Laboratory, Department of Information Science and Technology
Kita 13, Nishi 8 Kita-ku, Sapporo 060-8628 Japan
mkuwahara@meme.hokudai.ac.jp

Abstract. Webbles are interactive digital objects that resides inside Webble World which is a new web-based knowledge media architecture, available through an Internet browser, that allows us, not only to publish compound documents with or without embedded services and tools into a world-wide shared repository like the Web, but also to reedit any compound documents available from such a shared repository, through direct manipulations, and through composition of new compound documents for local reuse and for publication to the shared repository. The reediting of compound documents denotes the extraction and recombination of their visual portions to compose a new compound document with a new layout and new interoperability of the extracted components. A huge variety of objects ranging from traditional web documents with text and images, to interactive tools, virtual labs, games and community corners can be created and recreated with these building blocks of 'meme Lego'.

Keywords: Meme Media Objects, Customizable and Configurable Web, Shared knowledge, Distributional Resources, Interaction and Participation.

1 Introduction

This document attempts to explain the theoretical, philosophical and technical foundation of a newly developed type of digital objects named Webbles.

1.1 The Philosophy and goal

Webbles are based on the philosophy of memes, which is coined by Dawkins [1], that every thought and all knowledge which humans share 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 make human knowledge fit the meme description so that knowledge may easier spread, evolve and enhance in a shared environment. One of these attempts was done by Tanaka and his group at Meme Media Laboratory in Hokkaido University, Japan, which gave us the theoretical as well as the practical creation of the IntelligentPad [2]. The purpose of meme media was to work as smart containers 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.

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, they are subject of 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 which the user can configure so that they may solve some tasks, or present some content the user wants to share. Such compound Webbles can be combined as a set of tools and interactive content presentation devices in order to form a Webble application. When satisfied the user can republish the creation to the web and 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 production 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.

As our understanding of culture and knowledge sharing gains, along with the evolvement of today's Web to Web 2.0 and 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.

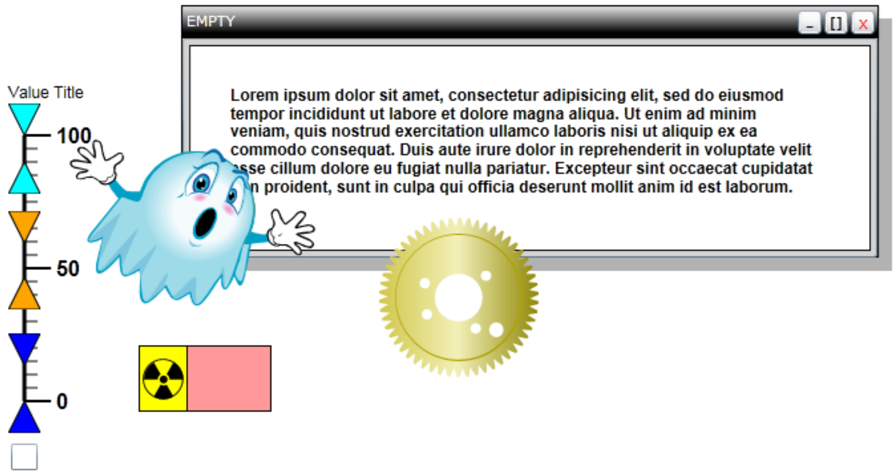
1.2 The Birth of Webble World

Webble World has submerged over a period of 3 years, and it is now entering its final iteration of refactoring in order to take full advantage of its core technology and stretch its meme media capabilities to its outer realms. The current running version is fully functional though 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.

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, 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 existed, even the most recent one was already several years old, and though that maybe was not a major problem in itself, they also lacked many of the new requirements and suggestions formulated by the EU project, like open source, complex graphics, full web compatibility, web browser integration and independence of OS and browser applications, just to mention a few, as well as suffering from some flaws in community accessibility, mainly due to being commercial applications with a limited or no solution for meme object distribution in an open environment.

In 2007, it was therefore 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 can be designed and used, reusability 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 therefore believe that the limitations of the primitive Webbles, the Webble gadgets and the Webble application lies more or less completely within the limitations of the imagination of the human resources. 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 that concept even further, these Webbles, may be the next stepping stone toward the idea of a shared mind.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit sse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat n proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Fig. 1. Six different Webbles, both primitive and compounded and configured.

2 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), for those who have attempted to develop any major application with previous implementations of IntelligentPad. The difficulties range from shape restrictions, problems of pad distributions across users, copyrights issues, and (although sound) development restrictions to preserve generic usage, as well as the latest addition to obstacles sandbox and cross-domain restrictions.

Some probable major reasons why IntelligentPad is not a worldwide, well known, fact today among computer users, even though the concept has been around within the academic community for almost two decades, are for example that many developers seldom develop systems with real re-usage in mind, and it often takes half the time to make a fast simple non re-usable solution than it takes to make a generic IntelligentPad one, at least in the beginning when primitive pads must be developed, therefore many IntelligentPad products are almost impossible to separate and re-use in other environments. Also the fact that previous Implementations have been commercial products owned, developed and controlled by major corporations, used internally and sometimes with no major intention to release to the public has been in the way of expanding the user community. The incitement to buy a sandboxed development tool only useful for people owning the same product has been slightly limited.

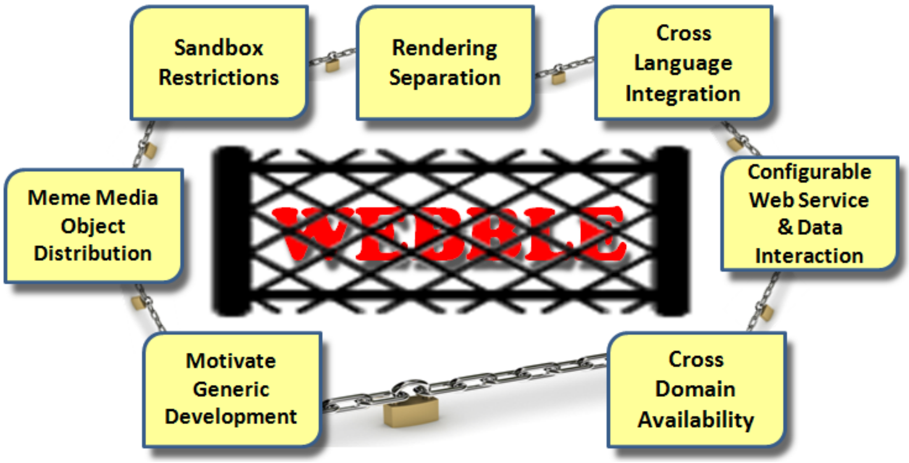

Sandbox Restrictions
Rendering Separation
Cross Language Integration
Meme Media Object Distribution
WEBBLE
Configurable Web Service & Data Interaction
Motivate Generic Development
Cross Domain Availability

Fig. 2. The current obstacles and restrictions range from a few technical, some cultural, to a majority of legal issues, but breaking most of these are not too far from reach.

All these problems have mainly been in the implementation of the concept, and few lies in the theories, which are of course also constantly evolving with each attempt to implement the idea of Meme Media and knowledge federation.

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 clickable 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.

As the demand for culture and knowledge sharing increases, the need for new, more open and configurable tools emerges, where users transform from being only users and spectators into becoming participants and contributors as well. Such a tool needs to support all types of users, from any background and with any set of skills. It must be open and free. It should be simple to access from anywhere in the world and it should easily adapt to new needs with mainly one internal goal; to share the collected minds of us all.

3 Results

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.

3.1 Theory

Webble is the shortening 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 the final 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, true 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).

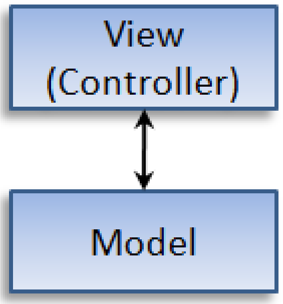


Fig. 3. Scheme of the two internal parts of a view and a model that makes up a Webble.

These are two separate entities with similar structures but with different roles. 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.

The next final iteration of Webble World refactoring will take this separation even further and implement the MVVM (Model-View-View-Model) concept which in reality means that the current View will be separated even further and remove all the drawing and rendering into its own group, which will be referred to as the view while the logic for user interaction and the logical concept of displays and other former Webble view core code will be within a group called View-Model or Webble Base.

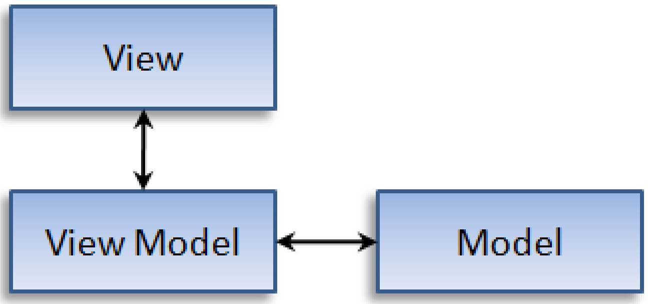


Fig. 4. Scheme of the future three internal parts of a view , view-model and a model that will make up a Webble.

This new separation will allow a more proper approach when dealing with bindings of visual objects and its underlying methods and event handlers, as well as opening up for the possibility to replace the rendering and drawing used today (XAML) with other types of drawing (HTML5, Flash, SVG etc)

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 bilateral one.

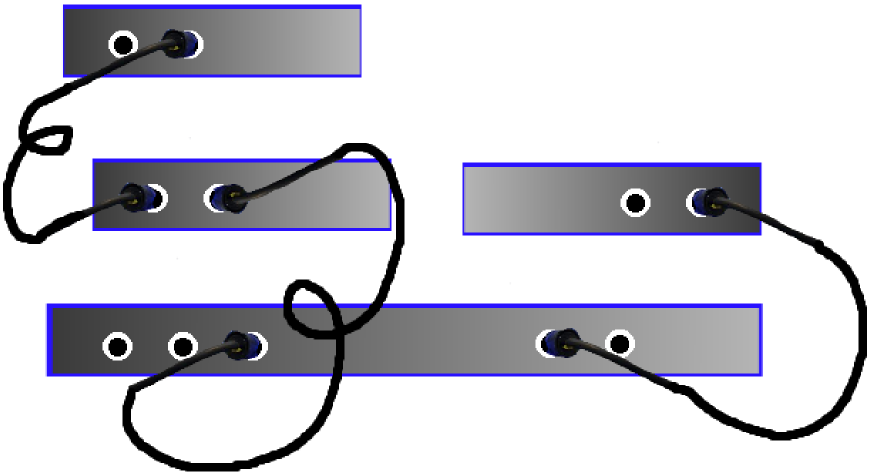
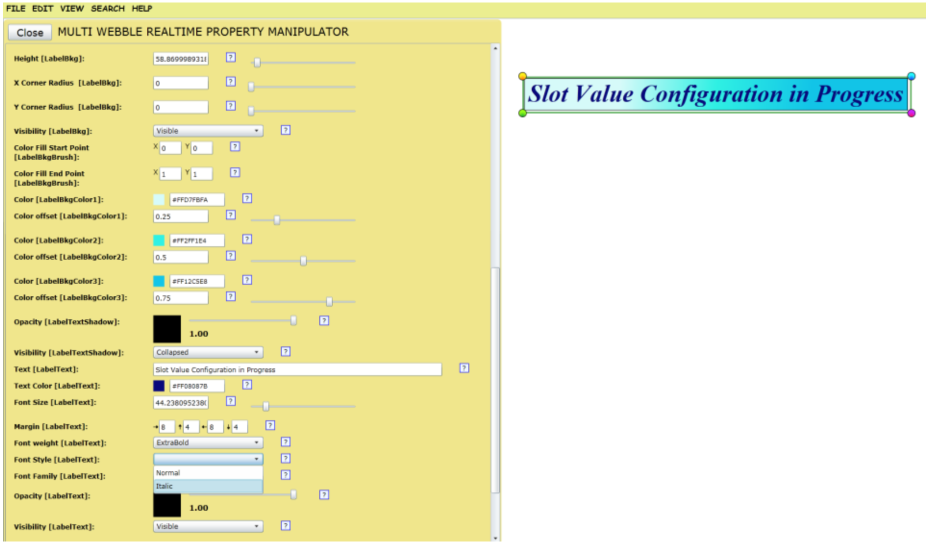


Fig. 5. Webbles connected with different slots used for communication between the Webbles.

Slot values can be of numerous types, from the classical numerals and strings, to more 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 of which type it values can be.


FILE EDIT VIEW SEARCH HELP
Close
MULTI WEBBLE REALTIME PROPERTY MANIPULATOR
Slot Value Configuration in Progress

Height [LabelBkg]: 58.869998931
X Corner Radius [LabelBkg]: 0
Y Corner Radius [LabelBkg]: 0
Visibility [LabelBkg]: Visible
Color Fill Start Point [LabelBkgBrush]: X 0 Y 0
Color Fill End Point [LabelBkgBrush]: X 1 Y 1
Color [LabelBkgColor1]: #FFD7FBFA
Color offset [LabelBkgColor1]: 0.25
Color [LabelBkgColor2]: #FF2FF1E4
Color offset [LabelBkgColor2]: 0.5
Color [LabelBkgColor3]: #FF12C5E8
Color offset [LabelBkgColor3]: 0.75
Opacity [LabelTextShadow]: 1.00
Visibility [LabelTextShadow]: Collapsed
Text [LabelText]: Slot Value Configuration in Progress
Text Color [LabelText]: #FF08087B
Font Size [LabelText]: 44.238095238
Margin [LabelText]: 8 4 8 4
Font weight [LabelText]: ExtraBold
Font Style [LabelText]: Normal, Italic
Font Family [LabelText]:
Opacity [LabelText]: 1.00
Visibility [LabelText]: Visible

Fig. 6. The Webble Platform provides a slot value configuration tool in order to change the values of a Webble, both visual and non visual.

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.

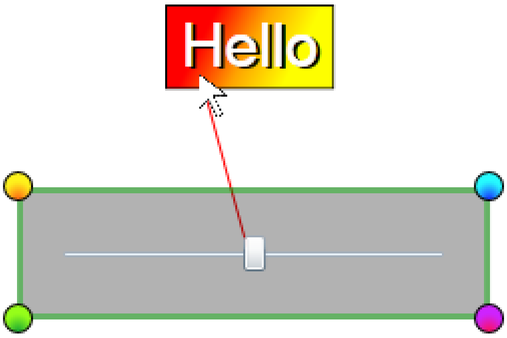


Fig. 7. Creating a Parent/Child relationship with mouse operation.

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 connection 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 concerned 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 primitive Webble that can handle that. A similar structure of communication is going on between the Model and the View within the Webble itself.

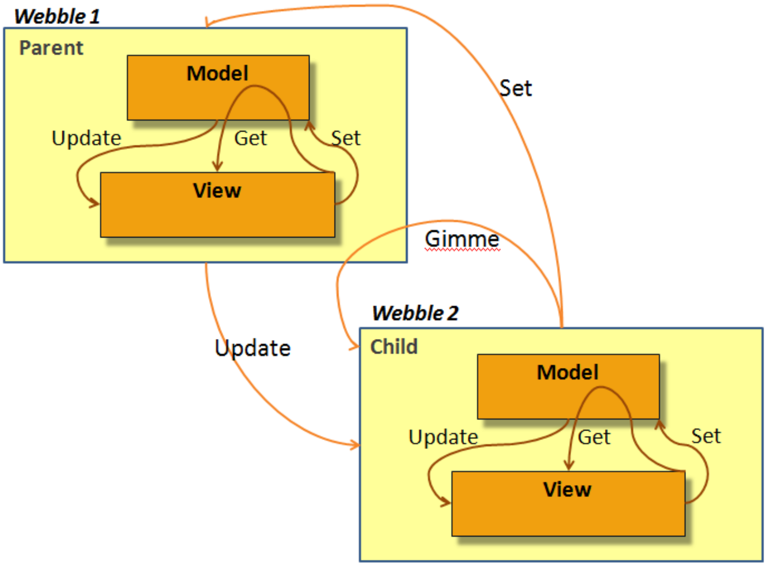


Fig. 8. Scheme over slot communication between Model and View as well as between related Webbles.

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.

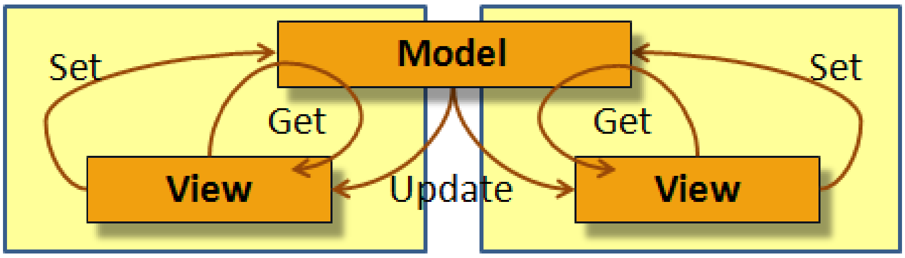
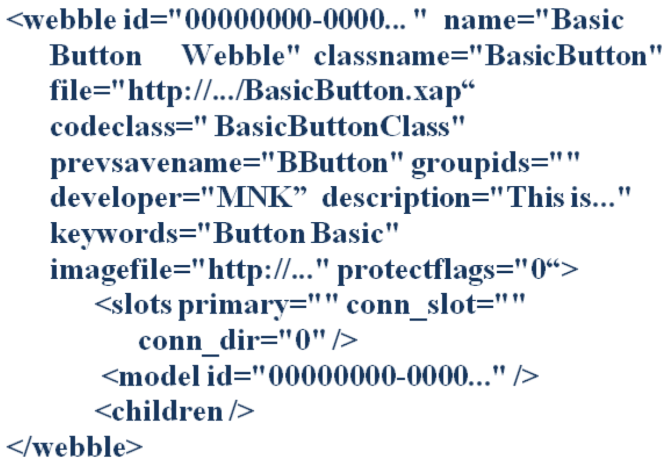


Fig. 9. Two Webbles Sharing the same Model.

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.

In the final iteration of code refactoring where the View will be further split up into the actual visual part and the part that contains the visual handlers and handlers for user interaction, the Webble will not be aware of how the Webble is drawn and rendered, just be assure that it will be, by the part that is told to handle that.



```
<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. 10. Parts of the XML definition for a primitive (non-configured) Webble.

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 and much more depending on Webble class. It is this XML file that separates primitive Webbles (code generated only) from compound ones.

There are two levels of Webble construction, one is the creation of primitive Webbles which means to create some useful features in code to wrap them into a Webble, and then to publish its executable online. The other level is the configuration of primitive Webbles in order to create an XML definition file and then to save that online. In both cases making the creation available for the public.

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 vast available number of primitive Webbles will allow most users to create gadgets and application without writing a single line of code.

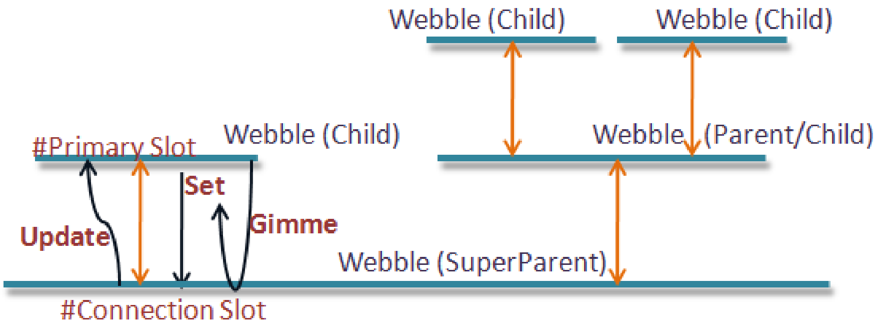


Fig. 11. Webble interoperability through slot communication.

What have just been described is the theoretical description of a default Webble in its basic state and behavior. But a Webble developer coding a primitive Webble can of course extend, and 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, 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.

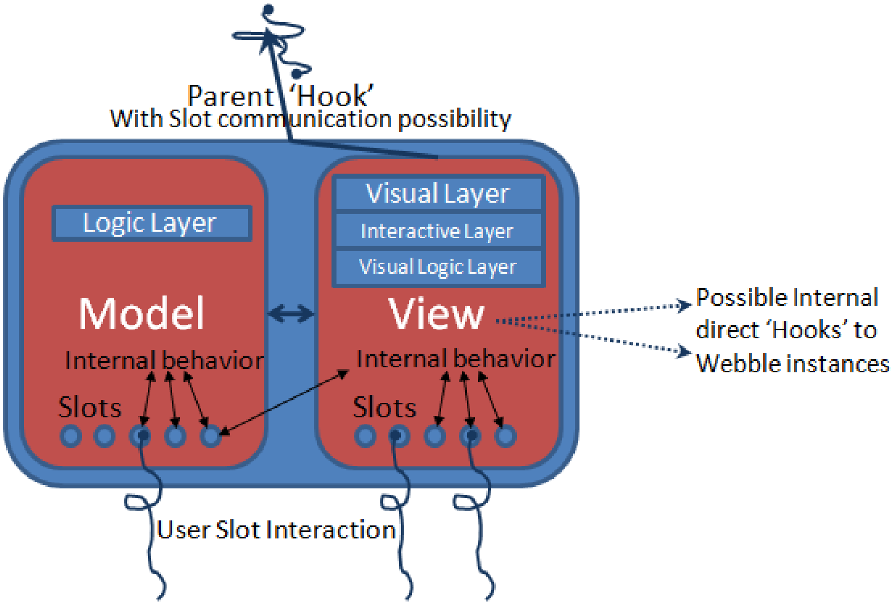


Fig. 12. A scheme figure of the internals and the interfaces of a Webble.

3.2 Enabling Technology

The technology behind the Webble is first and foremost Silverlight [3][9], a Microsoft web application framework, distributed as a browser plug-in, that provides functionalities similar to those in Adobe Flash, integrating multimedia, graphics (both pixel and vector graphics), animations and interactivity into a single runtime environment with support for a list of CLI languages and development tools. The current version, 4.0, was released in April 2010. It is compatible with multiple web browser products used on Microsoft Windows, Linux, and Mac OS X operating systems. Mobile devices, starting with Windows Phone 7 Series, will likely become supported in 2010. Silverlight is still under development, but it has definitely left its infant years since its birth in 2007 and today it includes most of what are needed for powerful Rich Internet application building and more are yet to come.

Silverlight applications are mainly programmed in C# (or VB) with the backup of XAML scripts, which is a xml based language for vector graphics first introduced with Window Presentation Foundation. And the relationship with WPF does not entirely stop there, actually Silverlight is in many ways WPF's younger brother, or its cousin, which, though smaller and less equipped, have compensated that with its availability and closeness to the World Wide Web. Silverlight is naturally and easily integrated with well known script and mark up languages as HTML, JavaScript and IronPython etc.

The base of the requirements, that finally lead to choosing Silverlight as the technical platform were that these meme media object had 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. Silverlight more or less fulfilled all these requirements and even some more as for example that Microsoft also provides good developing interfaces for Silverlight development, both from a programmer's point of view as well as that of graphic designers, with tools like Visual Studio .Net and the MS Expression Suite.

Fig. 13. Though Silverlight was not the only option for Webble development it has definitely proven to be a sound choice.

The Silverlight plug-in is small in size and easily installed, like for example Flash, in less than a minute. It is basically browser- and OS- independent. It is developed by a fairly trustworthy company that intends to support the Silverlight framework for a long time to come. And it is supporting full-fledged, well-known and established programming languages like C# or VB, basically giving all the programming power in the world to the developer if she is up to the task. Silverlight also provide something called OOB (Out of Browser) mode which allows any Silverlight application, or Webble in this case, to be stored locally on a computer as a desktop application and by doing so confirming a higher trust level of the application and therefore increase the power of available solutions and implementations.

But it should be mentioned that many other technologies were considered, even before Silverlight was even a known option. And for some of these other technologies, even prototypes were developed. For example SVG, JavaScript, XUL, Java and Flash were some of those technologies examined closer, but which at the end fell short compared with Silverlight. Having said that, in the name of fairness, against other technologies, one must make it clear that even though Silverlight was, and still is, considered the best choice, it does not deliver 100% in all areas. Silverlight and

Microsoft legal departments even in some parts makes the development of Webbles an initially painful and depressive experience since Silverlight is blocked and sandboxed in several ways from interacting freely with other Web resources like plain HTML sites. It is something one hope will change in the future, but at the moment has to be solved by quite complex workarounds and so called hacks.

The internals of the Webble elements, the Model and the View, are as stated before in the hands of a Webble programmer (even those with limited skills) using managed code, mainly C# (or VB), maybe with optional combinations of ASP.NET, HTML, JavaScript, Web service support and Ajax technologies, to create a specific behavior or feature within the Webble. In other words, if it can be done with code at all, it can be done by a Webble.

A finished and compiled Webble with its containing resources and additional dll files is then ready to be published online. The process of publishing a Webble contains the tasks of putting the Silverlight code package (a so called XAP file) on some online server and also generate the XML Webble definition file of how the Webble should look and behave when it comes alive. The Webble World platform provides a tool for doing this easily and automatically, but it can also be done by hand if required.

It is by this XML file the Webble is identified and found, as well as telling any Webble Platform where to find the Webble code XAP package. 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 (and Silverlight) exploit to the limit.

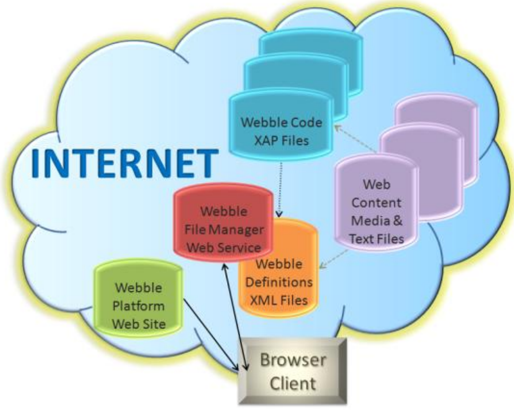

INTERNET
Webble Code XAP Files
Web Content Media & Text Files
Webble File Manager Web Service
Webble Definitions XML Files
Webble Platform Web Site
Browser Client

Fig. 14. How the Webble World client interact with the Internet to locate Webble definition files, Webble code packages and Webble media content.

The figure above shows that the Webble code, the XML definitions and other media content, which the Webble needs, can all be stored anywhere, separately online, and can still be gathered together on the client browser to perform its task.

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 platform Webbles may use completely other techniques in order to store and find available Webbles.

3.3 Webble World Version 1.0

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.

A hardcore Webble programmer would download the development pack and probably use Visual Studio .NET, maybe in combination with Expression blend for XAML editing, in order to build and compile the Webble. The developers pack includes templates for both Model and View, necessary Webble core dll:s which assure that the Webble conform to the meme media architecture and documentation that explains how it all connect. When the primitive Webble is finished and it is running fine in the localhost test platform also provided in the developers pack the developer would then return to the online Website, where he would use the provided publishing tool in order to auto generate a Webble definition XML file and also upload the code to an online server, giving a name to the new Webble.

Before one may publish a Webble, one must first register as Webble developer on the same web site. This is required in order to properly store the Webbles but also so that the system automatically can bind the creation to the registered developer for credit reasons. This is not a demand from a strict Webble point of view, but only a solution adopted by the current Webble World platform in order to implement some kind of simple maintenance of Webbles uploaded to the server. Therefore all current Webbles have data positions in their XML definition file for keeping track of developers and usage, but since they care themselves little or not at all about this data, and it all up to the developer of the platform Webble how it is used or collected, the next final iteration and refactoring of the Webble core system will probably remove this data as default data and instead add it as slots when the platform Webble so requires. The same principle will be applied to other platform dependant data that is today stored by default outside the slot list. This is because other Webble Platforms may solve the problem of user registrations and Webble ownership differently the current Webble World does.

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. 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 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, earlier mentioned as XAP files, which the platform will download and unpack 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.

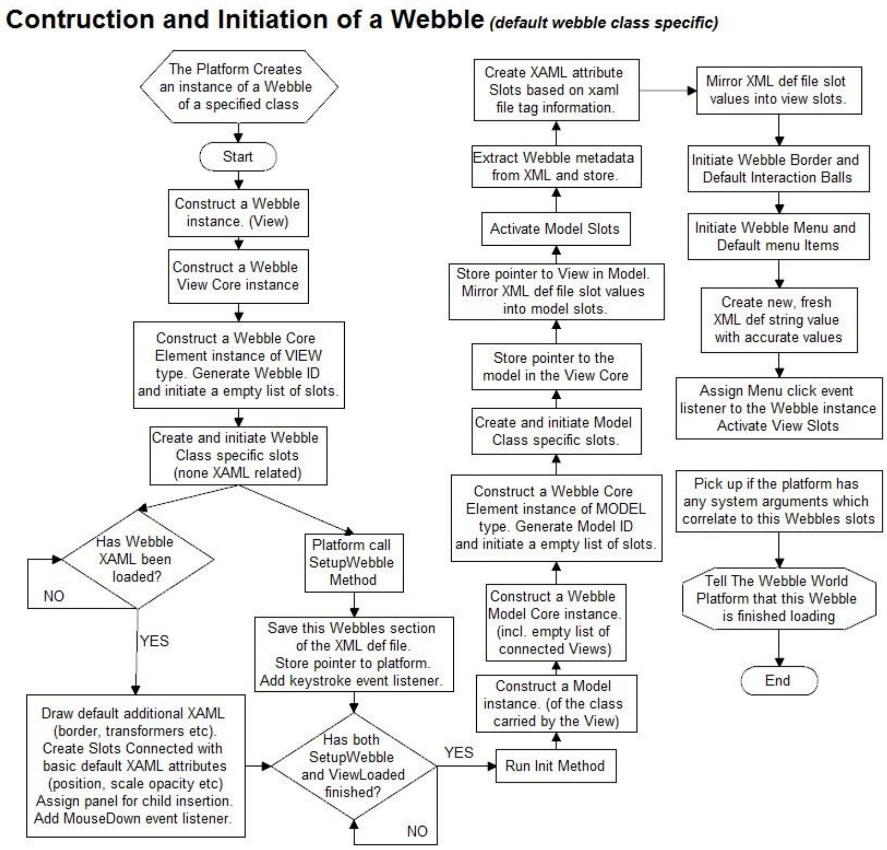
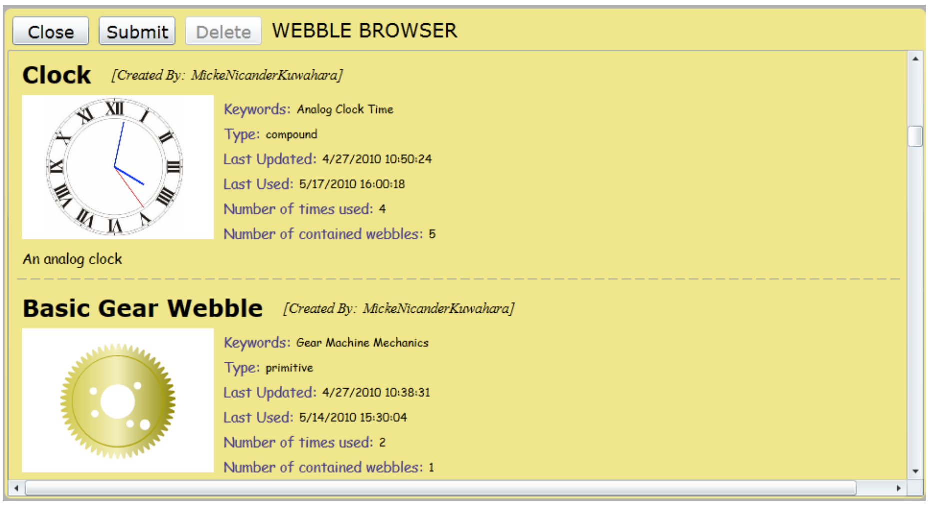

Contruction and Initiation of a Webble (default webble class specific)
The Platform Creates an instance of a Webble of a specified class
Start
Construct a Webble instance. (View)
Construct a Webble View Core instance
Construct a Webble Core Element instance of VIEW type. Generate Webble ID and initiate a empty list of slots.
Create and initiate Webble Class specific slots (none XAML related)
Has Webble XAML been loaded?
NO
YES
Draw default additional XAML (border, transformers etc). Create Slots Connected with basic default XAML attributes (position, scale opacity etc) Assign panel for child insertion. Add MouseDown event listener.
Platform call SetupWebble Method
Save this Webbles section of the XML def file. Store pointer to platform. Add keystroke event listener.
Has both SetupWebble and ViewLoaded finished?
YES
NO
Run Init Method
Construct a Model instance. (of the class carried by the View)
Construct a Webble Model Core instance. (incl. empty list of connected Views)
Construct a Webble Core Element instance of MODEL type. Generate Model ID and initiate a empty list of slots.
Create and initiate Model Class specific slots.
Store pointer to the model in the View Core
Store pointer to View in Model. Mirror XML def file slot values into model slots.
Activate Model Slots
Extract Webble metadata from XML and store.
Create XAML attribute Slots based on xaml file tag information.
Mirror XML def file slot values into view slots.
Initiate Webble Border and Default Interaction Balls
Initiate Webble Menu and Default menu Items
Create new, fresh XML def string value with accurate values
Assign Menu click event listener to the Webble instance Activate View Slots
Pick up if the platform has any system arguments which correlate to this Webbles slots
Tell The Webble World Platform that this Webble is finished loading
End

Fig. 15. The process schema of constructing a Webble and initiating it.

The work of the platform in the current version is merely to provide an environment, a pond, where all the Webbles can swim freely, and to keep track of these Webbles existence. In the final iteration of Webble World, the main part of the platform will itself become a Webble and the framework will be slimmed down to a minimum of necessities, just enough to allow a Webble to load itself, which from a clean Silverlight plug-in approach only takes a few lines of code. 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.

The second level of Webble developer is the one who will locate Webbles online, download them into the Webble World Framework, inside the browser and then starts manipulation and configure them together.



WEBBLE BROWSER

Clock *[Created By: MickeNicanderKuwahara]*

Keywords: Analog Clock Time
Type: compound
Last Updated: 4/27/2010 10:50:24
Last Used: 5/17/2010 16:00:18
Number of times used: 4
Number of contained webbles: 5

An analog clock

Basic Gear Webble *[Created By: MickeNicanderKuwahara]*

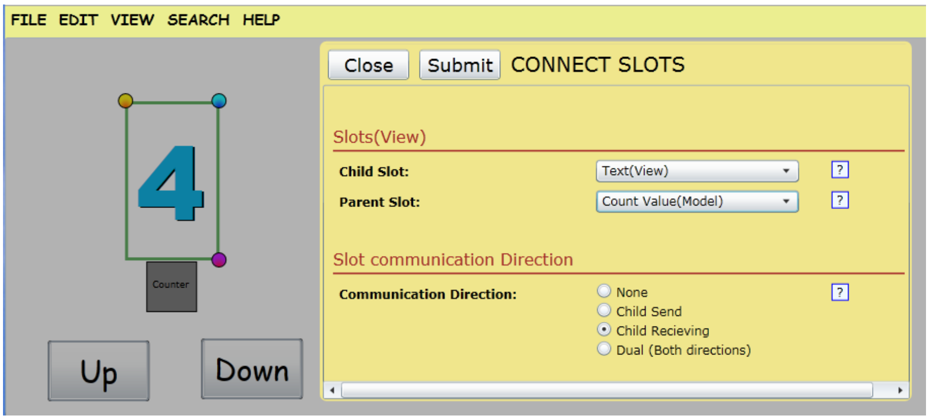
Keywords: Gear Machine Mechanics
Type: primitive
Last Updated: 4/27/2010 10:38:31
Last Used: 5/14/2010 15:30:04
Number of times used: 2
Number of contained webbles: 1

Fig. 16. Browsing for available Webbles in the Current Webble World.

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, 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, where 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.



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. 17. When a slot connection is submitted the values will immediately synchronize with each other and any effect that may have to 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.

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 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.

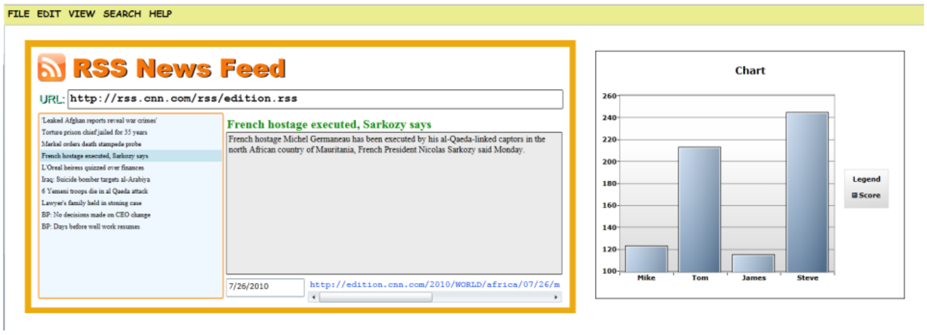
As soon as it is saved, it is now available for use by any Internet visitor who may use the Webble World platform's Search tool to find Webbles online, by name, description, developer or popularity.

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 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 Current Webble World Projects

Besides the forever ongoing task to provide more and more primitive Webbles to the Webble community where some of the latest has been a Chart Webble, Parallel Coordinate Webble, Custom XAML Webble and an RSS Webble just to mention a few, there are some other ongoing large projects where Webble application is being developed and actively used. By explaining the basics of these project we wish to give a further understanding of some of the things Webbles are capable of.



http://rss.cnn.com/rss/edition.rss

'Leaked Afghan reports reveal war crimes'
Torture prison chief jailed for 35 years
Merkel orders death stampede probe
French hostage executed, Sarkozy says
L'Oreal heiress quizzed over finances
Iraq: Suicide bomber targets al-Arabiya
6 Yemeni troops die in al Qaeda attack
Lawyer's family held in stoning case
BP: No decisions made on CEO change
BP: Days before well work resumes

French hostage executed, Sarkozy says

French hostage Michel Germaneau has been executed by his al-Qaeda-linked captors in the north African country of Mauritania, French President Nicolas Sarkozy said Monday.

7/26/2010

http://edition.cnn.com/2010/WORLD/africa/07/26/m

Fig. 18. A chart Webble and a compound Webble which is designed and build around the RSS primitive Webble, are just some of the small additions to the Webble World.

The first which was shortly mentioned in the Introduction of this paper is the ACGT project (Advancing Clinico Genomic Trials) funded by EU, which was started in 2007 and ended in September of 2010 and which goal was to develop extensive tools and technologies in order to improve the environment and the work flow for preparing, performing and analyzing Clinical Trial data for cancer research. One major part in that project was played by Meme media laboratory who developed a Webble application which task was to help the Trial chairman to build and design the medical trial treatment plan, and later through the same tool support the treating physicians to collect data on the patients and monitoring the ongoing trial and finally, still in the same Webble tool, as the trial is finished, help researchers and clinicians to analyze all the collected data and present it in a useful and highly visual manner with parallel coordinates, charts and life tables.

This Webble application was built alongside the research and development of the actual core system of Webbles and was given the name 'Trial Outline Builder', TOB for short. Another note worth mentioning is that TOB was implemented inside another Web application developed in Java and communicates with it via JavaScript, further showing the versatility of Webbles and the Webble platform.

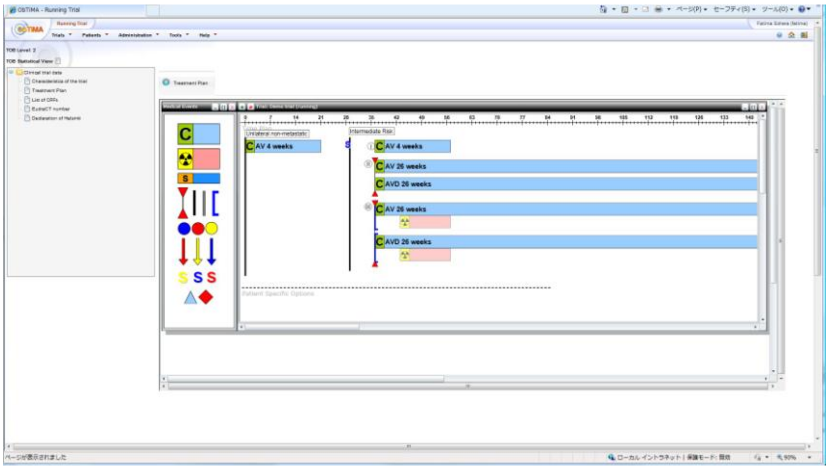


Fig. 19. The TOB application in trial treatment plan design mode, running inside the ObTiMA(Java) website online.

In January of 2011 the sequel of the ACGT project will begin, named P-Medicine (personal medicine) where the TOB application will be further refined and improved and benefitting from further improvements of the Webble core system. TOB has already received highly international acclaim, and as the project continues it will surely be a true beacon for future developers of Webble application.

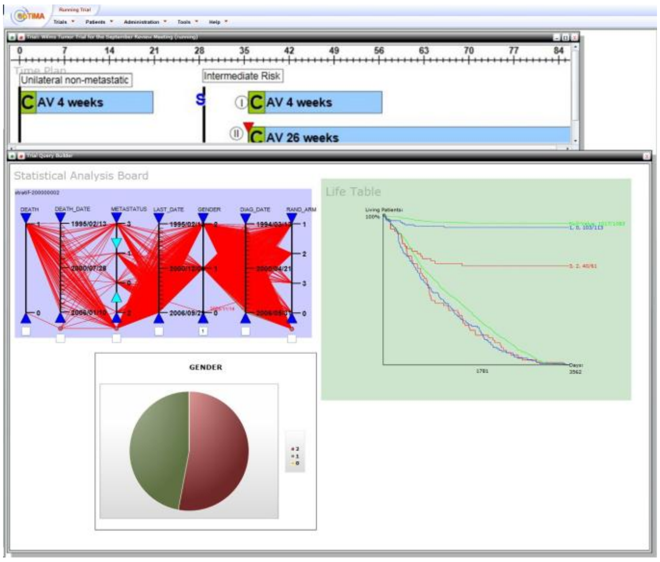
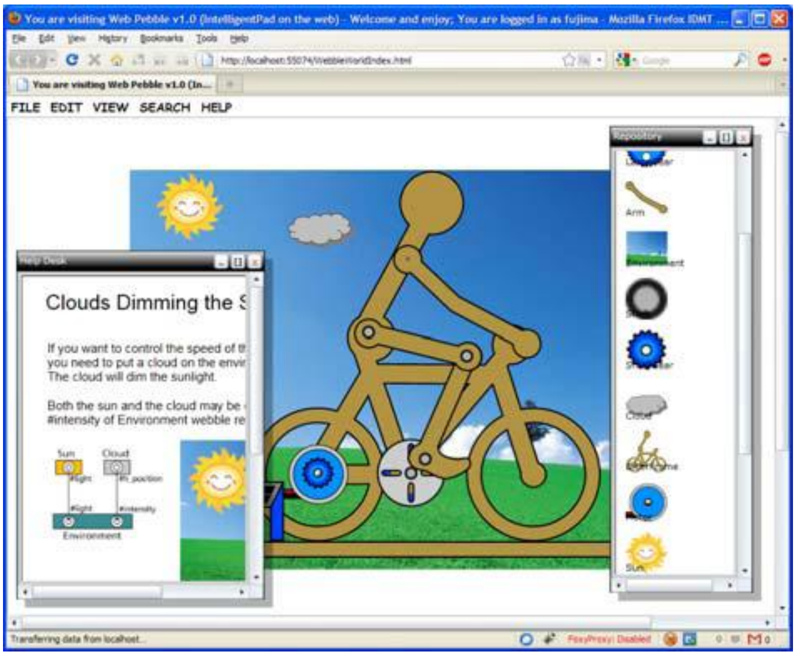


Fig. 20. The clinical data analyzing mode in TOB; all done with Webbles.

The second project which goes under the name 'Solar Bike' is conducted at the Fraunhofer institute in Erfurt in Germany as a part of a major research project on children's media and e-learning. The solar bike is a Webble application that works as a virtual laboratory and test environment where the user can connect gears and engines and batteries and other components in order to build a working solar cell driven cyclist. All done by Webbles.



You are visiting Web Pebble v1.0 (IntelligentPad on the web) - Welcome and enjoy; You are logged in as fujima - Mozilla Firefox IDMT ...

Clouds Dimming the S

If you want to control the speed of th
you need to put a cloud on the envir
The cloud will dim the sunlight.

Both the sun and the cloud may be
#intensity of Environment webble re

Fig. 21. The solar bike Webble application with its build platform, object menu and Tutorial window. All are Webbles.

Especially the help and support part of the application has received positive response. The reason for that is the fact that all are Webbles, even the manual and tutorial, so manual objects can easily be transferred from the information container into the actual building environment and immediately starts working.

The Solar Bike project has also generated further projects, where for example one is planning to create a Webble application for use in classroom environment and e-learning situations in one University Course in Erfurt, Germany.

The third ongoing project, though very recently started, is the Assets project in Europe which is a part of the Europeana project that collects and gathers all digital knowledge and documents and media from all over Europe. In this project a Webble application that can search and display this digital content has been requested, and where the user further can combine and reorganize the media objects found and republish it as a new book or document (still related and connected with the original data sources) which also may be read or printed.

This is the major use of Webbles and Webble world that is currently going on and probably more will come.

4 Discussions

The use of web browsers as the base for Webble technology was an early and crucial choice since earlier IntelligentPad-based systems had suffered from problems regarding accessibility and distribution. A pure, free and open web platform should solve these problems and make it available and accessible to the common public.

The main intention with this open, web based development tool, with several levels of development, based on skill and purposes, that may be used by anyone, anywhere, to share any form of human thinking, both static and interactive, allowing full interactivity and participation and re-editing of all content has been to promote human collaboration and communication. Any Website existing today could be made in Webbles, and if they were, building new ones would take much less time. And it would also be easier by all skill ranges of Internet users to be creative and to participate in the process.

All it takes is a few dedicated primitive-Webble developers (using C# and Silverlight) in order to feed a much larger group of compound Webble developers (slot connecting, property- & behavior control) which in turn can feed even a greater group of users and super users (bloggers, community visitors & ordinary web surfers).

5 Conclusions

In order to simplify participation, to enhance the options of collaboration and to allow both re-usage and re-combinations of web resources for both skilled and ordinary web users, we have developed a framework called Webble World where users can publish, configure, combine and save a multitude of independent modules or meme objects which wrap both contents and functionalities. These objects are called Webbles.

In today's megatrend of mash-ups and programmable web, as well as Internet users increasing will to participate by creating, reshaping and contributing, Webble World offers a wide range of possibilities to be a part of the evolution of human knowledge.

Our future studies include the evaluation of Webble World in real-time use situations in order to confirm that all intentions with the framework are reached, and if they are not in some cases, figure out how to make them so.

We also are within the next month beginning the final iteration of refactoring of the core system as a continuance of previous evaluations of the system. This refactoring mainly include the further separation of the Webble View in order to have less dependencies between rendering and the View Logic, in current Webble case separation the XAML from the Webble itself. Further we intend to empower the individual Webble with most of the current platform's skills, in order to be able to view everything as Webbles, even platforms. We also aim to prepare Webbles to be more adaptable to unknown environments like outside the browser, on the desktop or places where the Webble platform and the Webble Web services does not exist.

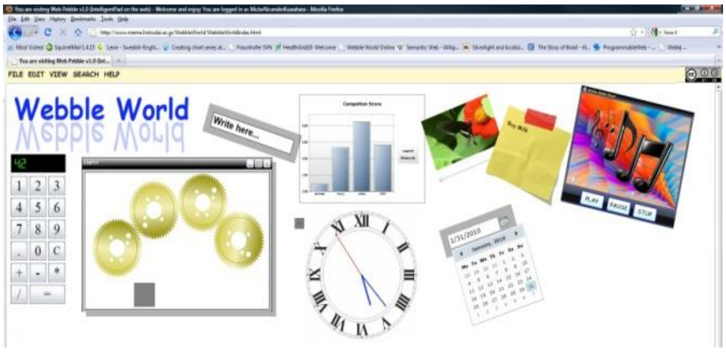


Fig. 22. Some Compound Webbles designed and configured inside the Web browser.

Webble World may be visited, studied & enjoyed at http://www.meme.hokudai.ac.jp/WebbleWorldPortal, [4] where the developer's kit is also available.

6 Reference

1. Dawkins, R.: The Selfish Gene. Oxford University Press, New York (1976)
2. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience, New Jersey (2003)
3. Beres, J., Evjen, B., Rader, D.: Professional Silverlight 4. Wiley Publishing (Wrox), Indianapolis (2010)
4. Webble World, http://www.meme.hokudai.ac.jp/WebbleWorldPortal/, (2010)
5. Blackmore, S.: The Meme Machine. Oxford University Press, New York (1999)
6. Czernicki, B.: Next-generation Business Intelligence Software with Silverlight 3. Apress, Berkley (2009)
7. Fujima, J.: A Unified Framework for Organizing, Accessing, and Federating Web Resources. Sapporo. Hokkaido University (2006)
8. Lutteroth, C., Weber, G.: End-user GUI customization. ACM, Proceedings of the 9th ACM SIGCHI New Zealand Chapter's International Conference on Human-Computer Interaction: Design Centered HCI. pp. 1-8. New York (2008)
9. MacDonald, M.: Pro Silverlight 2 in C# 2008. Apress, Berkley (2009)
10. Tanaka, Y. Imataki, T.: IntelligentPad: A hypermedia system allowing functional compositions of active media objects through direct manipulations. IFIP 11th World Computer Congress. pp. 541-546. (1989)
11. Tanaka, Y.: Knowledge Federation over the Web, Based on Meme Media Technologies. Lecture Notes in Computer Science: Federation over the Web. (2006)
12. Tanaka, Y.: Meme media and a world-wide meme pool. ACM, Proceedings of the fourth ACM International Conference on Multimedia. pp. 175-186. New York (1996)
13. Wikipedia - Interactive Web and Web 2.0 [Information Portal], http://en.wikipedia.org/wiki/Web_2.0, (2010)
14. Wikipedia - Semantic Web and Web 3.0 [Information Portal], http://en.wikipedia.org/wiki/Semantic_Web, (2010)
15. Programmable Web - Mashups, API's and the web as platform. Programmable Web. http://www.programmableweb.com/, (2010)