

Application Integration by Meme Media Concepts

Hans-Rainer Beick

r.z.w. cimdata AG
Zum Hospitalgraben 2
D 99425-Weimar
hans-rainer.beick@rzw.de

Abstract: This paper makes a bridging between two rather new fields of research in IT, the Meme Media (MM) and the Enterprise Application Integration (EAI). It describes the main principles of a unique application extension which allows connections between different applications with mutual interactions. The interaction bases on selected Meme Media technologies and on Web techniques.

Meme Media ideas are used for a very simple and easily usable kind of connections. Every user should be able to connect applications and to organize some interactions between them. Like in the Meme Media systems, a slot stands for a special import and export parameter of the application. Connections are always connections between two single slots of two applications. A larger composite bases on such pairs of directly connected applications. There is a propagation concept for interactions through the whole composite.

It is said that Meme Media Objects always live in a biotope. The biotope may be a runtime system or some background applications managing the composition and the interaction.

The described extension is implemented in a special system, called WIP (Web Based Integration Pads). WIP is rather more a standard for a connection interface for applications than a system. For the integration of applications, the applications should not be embedded in such an additional biotope. All preparations for connections and all interactions should be managed locally, inside the applications themselves. A connection only means the mutual authorization for interacting.

The research was sponsored by the German Ministry of Economics and Labour in the Project IPWIAS, (FKZ: KF 641701KLF3, 2003-2005).

1 Aspects of the Enterprise Application Integration

At first, software programs were designed to run independently, without any interactions between them. In enterprises, more and more tasks have been transferred to software applications. The consequence was a conglomerate of different and not interacting applications depending on the special type of the enterprise. Such a system of applications cannot work effectively.

On the other hand, there is no chance to design a totally fitting complete system for every enterprise. You always have to compose single applications. But, these application should be able to work together. The aim of the Enterprise Application Integration (EAI) consists in putting together all the single application to an unique system.

Different categories of EAI are considered, like database linking, application linking, data warehouse, and common virtual systems. Primary, we want to make an easy and unique application linking. This linking should be the base for an simple common virtual system [Ke02, We04].

The basic step is the interaction between two single applications. Before, the application linking was done by a special designed and implemented interface between the both applications. Here, the applications should be extended by a special and nearly unique component organizing the interactions between them. The interactions base on sending and receiving data records.

2 The Meme Media Approach

The research on Meme Media was started in 1987 by Yuzuru Tanaka. The goal was to develop a toolkit architecture for the open integration of different software modules. Instead of using objects inside the programs like in object-oriented programming, the Meme Media approach wraps each object with a standard wrapper and treats it as a Meme Media object. Such a object can contain a document, an application program, or a service. It has an user interface, with a view on the screen and a standard set of operations, and a connection interface based on input output ports (*slots*) and a standard set of messages. The Meme Media objects are often denoted by *pads*, like pads on the screen.

The user can connect different pads by some easy user interface operations to a composite pad. In such a composite pad, there is a well defined and clearly arranged regime of the interaction of the different parts. All parts are removable. This fact offers the possibility for a reedition [Ta03].

It is said that Meme Media Objects always live in a biotope. The biotope may be a runtime system or some background applications managing the composition and the interaction [Ja04].

In the CHIP system by Kimihito Ito, the messages between the pads are Web documents. Every pad has the property of a Web server and of a Web client. Here, composite Pads often are dynamic Web application compositions.

The starting point of this work was to use the basic ideas of Meme Media for the integration of applications. The first steps in this direction have been gone by Jantke and Ito with the idea to wrap whole applications to pads and connect them as pads [It03].

In this approach here, the applications should not be wrapped. They should be extended by a nearly unique connection component. This extension is implemented in a special system, called WIP (Web Based Integration Pads) System. WIP is rather more a standard

for a connection interface for applications. The user interface for the connection issues is basically embedded into the context menu of the application.

What are the used concepts of Meme Media?

(1) A connection between two pads is a connection between one slot of each pad. Interaction by a connection means to send the value of the slot of the sending pad to the slot of the receiving pad. Only two methods are used for writing and reading such interaction messages:

GetSlotValue(slotName) and *SetSlotValue(slotName,newValue)*.

(In Meme Media, the approach is a little bit more complicate. All connections have a direction. It is said that one pad is the father and the other pad is the child. A child can only have one father. Therefore, there are some further conventions. Theses father-child relations guarantee a tree structure in a composite pad and prevent cycles in the propagation of actions, described in the point (2). This is very helpful for the user, for really large composites. For the application integration, such large composites are not expected and directed connections seem to be not necessary.)

In the standard case, the *SetSlotValue* method only stores the new value into the slot. But sometimes, the operation may be a special one.

(2) For composites of pads, there is a concept of propagation of action. In the standard case, if a pad gets an action (interaction message) from a connected pad it sends actions to all other connected pads.

(3) For sending message form one pad to another, Web techniques are used like in CHIP [It03].

What are the main differences between the Meme Media and the WIP approach?

(1) Connecting two pads in Meme Media really means to embed both pads into a common object. Connections in WIP are not physical connections, only an mutual authorisation for interaction. One application sends a document to the other one, and the other one will accept the order in the document, if there is some proof of authority inside the document to do it.

There is no common management for connected applications, no biotope. The whole management of connections and interactions is locally done inside the applications.

3 The WIP Extension

For the interaction of two applications, the structure shown in figure 1 is used. There is a unique object *WIPComponent* which is added to the application via a special interface object. In the interface, the list of used slots is defined. Methods *WIPGetting(slotName)* and *WIPSetting(slotName,newValue)* are added to the application which are used for the bilateral refreshment. There is also a method for starting some propagation.

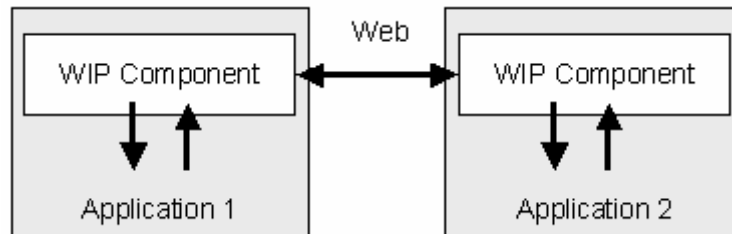


Figure 1: The structure of a connection between two WIP Applications

There are two kind of refreshment methods: The refreshment of the WIP component by the actual values of the application and the refreshment of the application by the actual values of the WIP component.

4 Dynamic Slots

The *WIPComponent* works slot based. There are different kinds of slots. **Application Slots** and **System Slots** are predefined. Application slots are added in the stage of application extension. They are related to some values inside the application. System slots are defined in the unique *WIPComponent*. A system slot is related to some internal value or some internal operation.

Display Slots, **Memory Slots**, and **Action Slots** have a different behaviour defined by the *SetSlotValue* method. A display slot only shows some internal parameter, for instance like the own IP address. (The *SetSlotValue* method makes nothing on this slot.) The memory slots have the standard behaviour. (The new value is stored in the slot, and nothing more is done.) By the help an action slot you can start some internal operation, for instance the adding of a new connection. Here, the new value is a parameter for the related operation. The slot value after the calling is often something like the result of the operation.

This slot concept allows a remote control of the *WIPComponent* and in some parts of the application by another application. That's why it is necessary to define which slots are **Open Slots** for an integration and which are not. It is done by the notation *Pool* in the next chapter.

In larger business applications like Enterprise Resource Planning systems, hundreds of application slots are needed. And for a special connection to another application, a special subset of the slot set has to be combined to one slot which will be used in the connection. It is not possible to predefined all useful slot combinations while programming. **Dynamic Slots** are necessary. Dynamic means that it is added while preparing a special connection. This adding is started by a menu item and done inside the *WIPComponent* by an action slot.

The WIP system uses some kinds of dynamic slots. **Transit Slots** are dynamic memory slots. They can be used for the transit of a parameter through some applications if there is a sequence of some applications in the integration. **Combination Slots** (more correctly variation slots) combines a sequence of slots to one slot. The value of a combination slot is the coded sequence of the values of the combined slots. (The *GetSlotValue* methode makes this coding. The *SetSlotValue* method firstly decodes the value and calls the *SetSlotValue* method for all contained slots, in the given order. The name of a combination contains all combined slot names, for instance **+a+b** for the slots *a* and *b*.

5 The Authorisation Concept for Connections

There are three levels of preparing integrations of some WIP extended applications: the pools, the integrations and the connections.

(1) All applications of an integration must belong to the same pool of applications. A pool is defined by a name and a password. It is like a user account for a computer. Before integrating, a common pool has to be created in all applications. This creation is always done locally, inside of each application.

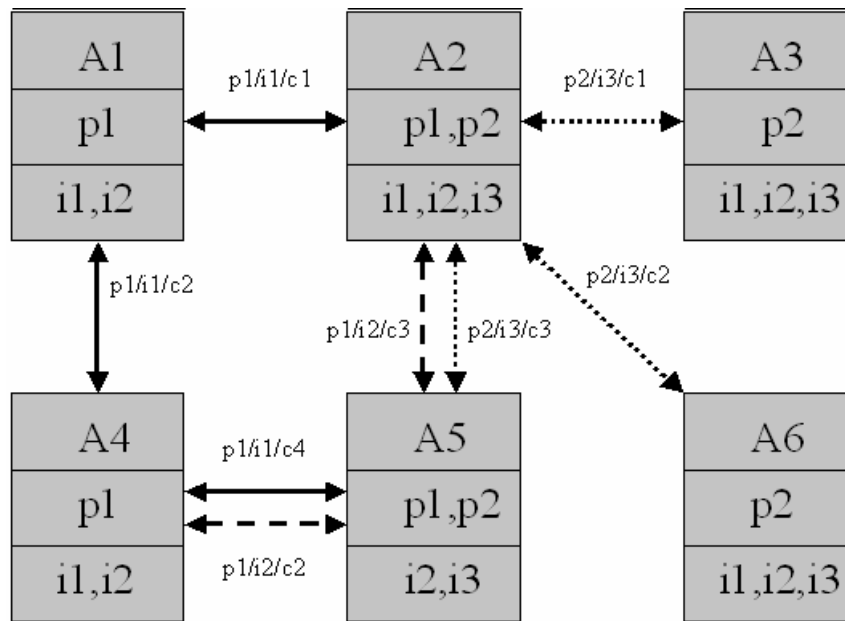


Figure 2: An example for a complex integration situation

(2) Next, an integration is created, belonging to one pool. If the pool data and the Web addresses of the other applications are known, you can create the integration in all other applications from your application. The integration is defined by a name, and describes one composite with a propagation of actions inside.

(3) If both applications have the same integration in the same pool, a connection can be created in each of the both applications.

Every application has a list of usable Slots. It depends on the application and is defined in the WIP extension of the application. For the most of business application, you need a finer concept for these usable slots. For some tasks, a special slot should be usable, and for other tasks it should not. For instance, if there is a database for staff data, for the some user the salary of the persons should be visible, but in general not. In WIP, it is implemented by the help of the notion *pool*. For every pool, a list of open slots can be defined inside the application. And every connection inside this pool can only the open slots of the application.

The second figure illustrates an integration situations with six applications A1, ..., A6, two pools p1 and p2, three integrations p1, i2, and i3 and eight connections. Two applications can have more than one connection. But, these connections must belong to different integrations. So, you can implement different types of connections between two applications.

6 A Simple Example

Two simple WIP extended applications are considered. They are frames with three fields for display and input. The fields are related to the slots *text1*, *text2*, and *text3*. The task consists in connecting all fields (in the same order). It should be done in a pool *pool1* and an integration named by *persons*. The first steps are done locally in both applications.

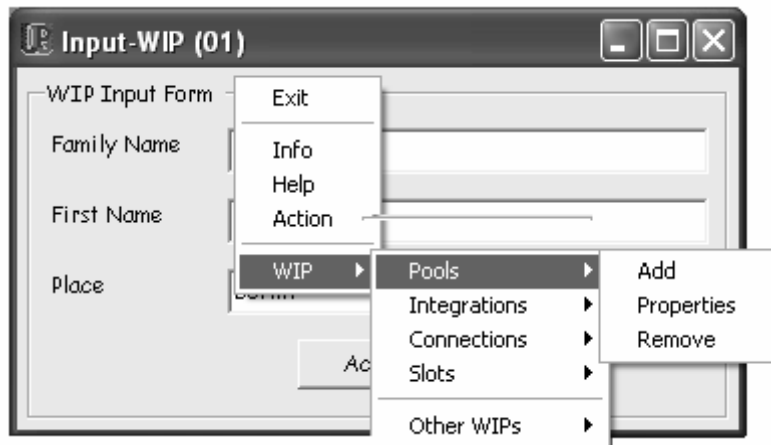


Figure 3: The creation of a new pool for preparing an integration
(The main WIP access is implemented inside of the context menu.)

- (1) The pool *pool1* is created, with the same password in both applications. All default open slots of the application including *text1*, *text2*, and *text3* are set to be open in the pool.
- (2) Then, an integration *persons* is created in *pool*.
- (3) The combination slot *+text1+text2+text3* is created. As a combination of open slots in *pool1*, it is also a open slot in *pool1*.

It is possible to do the steps (2) and (3) only in the first application, firstly for the application itself and then by remote control for the other application. The permission for the remote control is automatically given, because of the sent Web documents contains the pool name and password in a coded form.

Now, a connection between both connection is opened, done by the first application. On both side, the new combination slot and the standard propagation behaviour are used. Now, if we write some data into one input frame and finish this input, the data are sent to other application and displayed in its frame.

The figure shows two screenshots of web input forms. The top screenshot is titled 'Input-WIP (01)' and contains a 'WIP Input Form' with three text input fields: 'Family Name' (containing 'Meyer'), 'First Name' (containing 'Kurt'), and 'Place' (containing 'Berlin'). Below these fields is an 'Action' button. The bottom screenshot is titled 'Input-WIP (02)' and also contains a 'WIP Input Form' with three text input fields: 'Ort' (containing 'Meyer'), 'Vorname' (containing 'Kurt'), and 'Name' (containing 'Berlin'). Below these fields is an 'Action' button. Both forms have standard window controls (minimize, maximize, close) in the top right corner.

Figure 4: After an interaction, the three values are transmitted.

7 Future Works

For a real use of the WIP components in Business Application systems, a much higher level of security of the Web communication is necessary. This concerns two main aspects. The information sent from one application to another should be protected against eavesdropping. And in the WIP concept, the interaction between two applications means the access of one application to the other via the Web. How can you prevent a not authorised access to your application? For such a security management, only the first small steps are done.

References

- [Be05] Beick, Hans-Rainer: Aspects of WIP Technology Development. Workshop “Federation over the Web”, Dagstuhl 2005, Proceeding 2005, submitted.
- [It3] Ito, Kimihito: A Visual Environment for Dynamic Web Application Composition. 3rd International Workshop “Access Architectures for Organizing And Accessing Intellectual Assets”, Meme Media Laboratory, Hokkaido University Sapporo, Japan, Proceedings, 2003.
- [Ja04] Jantke, Klaus P.: The Biotope Issue in Meme Media Implementations. International Workshop “Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets”, Dagstuhl Castle, March 1-5, 2004, Revised Selected Papers, Lecture Notes in Artificial Intelligence, volume 3359, pages 99-107, Springer-Verlag Berlin Heidelberg New York, 2004.
- [Ke02] Keller, Wolfgang: Enterprise Application Integration: Erfahrungen aus der Praxis. dpunkt.verlag GmbH Heidelberg, 2002.
- [Li00] Linthicum, David S.: Enterprise Application Integration. Addison-Wesley, 2000.
- [Ta03] Tanaka, Yuzuru: Meme Media and Meme Media Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press and Wiley-Interscience, 2003.
- [We04] Webopedia Online Computer Dictionary for Computer and Internet Terms and Definitions, www.webopedia.com, 2004.