

---

# Visual Web Application Composition Using WebPads

Marcel Karam\*, Sergiu Dascalu\*\*,  
Rami Santina\*, Zeina Koteich\*, Rana Awada\*

\*American University in Beirut, Lebanon

\*\*University of Nevada, Reno, USA

*APLE-2006*, August 21, 2006, Baltimore, MD

## Outline

---

- Introduction
- Related Work: Pads and Pads Architecture
- WebPads: Concept, Architecture and Tool
- VisualWebC: Additional Features
- On WebPads, Agility, and the SPLE Context
- Experimental Studies
- Future Work & Conclusions

## Introduction #1

---

- Many current web applications are *service-oriented*
- Typically, emerging web applications seeking to provide similar services make use of shared deep annotation architectures, which involve:
  - Database owner
  - Annotator
  - Querying party
- Other approaches rely on loosely coupled solutions such as Service-Oriented Architectures (SOA)
- The above models are convenient, but have drawbacks and are not very suitable for remote service-oriented interactions

3/22

## Introduction #2

---

- The approach we propose has the following main characteristics:
  - Relies on the ability of *wrapping* web services during design and implementation
  - Allows web engineers to simply integrate such services “*visually*” (using icons) in the web application being built
  - Incorporates wrapping techniques into a Model-View-Workflow (MVWf) framework
  - Extends Ito and Tanaka’s (2003a) concept of *Pads* and builds upon their *Intelligent Pads Architecture (IPA)*

4/22

## Introduction #3

---

- Main characteristics of the proposed approach (cont'd):
  - While Ito and Tanaka's Pads wrap services of existing web applications into *new desktop applications*, WebPads wrap them into *new web applications*
  - The newly proposed WebPads architecture extends Ito and Tanaka's IPA by focusing on *MVWf-based* web engineering solutions
  - A supporting environment, *VisualWebC*, has been built for design and development experimentations
- Our goal has been to improve the effectiveness of the web application development and maintenance (upgrading) process

5/22

## Related Work #1

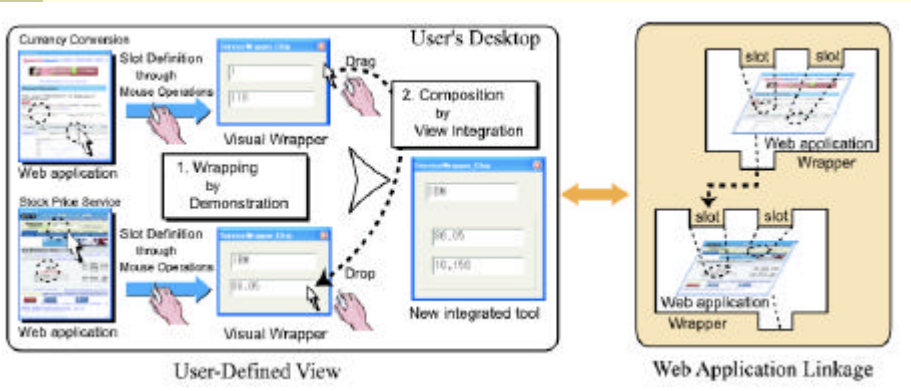
---

- Ito and Tanaka (2003a):
  - Pads, Intelligent Pads Architecture
- Ito and Tanaka (2003b):
  - Visual wrapping, functional linkage
- Fujima, Lunzer, Hornboek, and Tanaka (2004):
  - Clipping, connecting, cloning
- Distinguishing aspects of our approach:
  - WebPads for building web applications
  - MVWf-based architecture emphasis
  - Specific supporting tool, VisualWebC

6/22

## Related Work #2

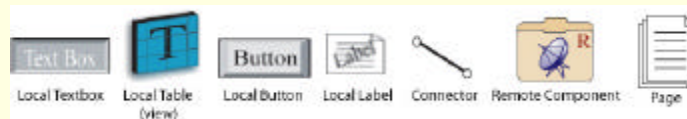
- Example of wrapping from Ito and Tanaka (2003b):



7/22

## Web Pads #1

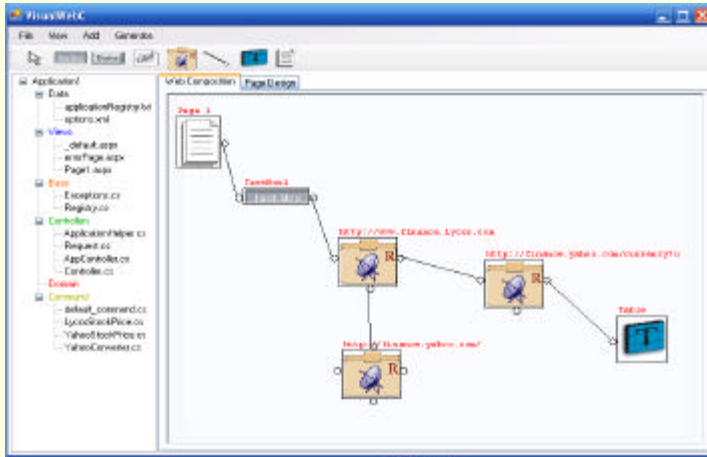
- Visual components (currently):
  - Several local components
  - One remote component



8/22

## Web Pads #2

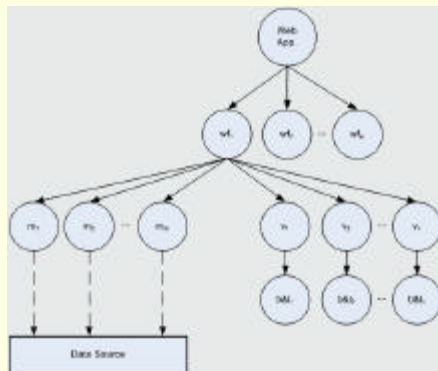
- Sample application (to illustrate concepts & tool)



9/22

## Web Pads #3

- MVWf framework of VisualWebC:
  - Workflows, views, models, and layers



10/22

## Web Pads #4

---

- Transition rules and commands captured as an XML document (part of web application generation). Sample file:

```
1- <command name="LycosStockPrice">
2-     <status value="CMD_OK">
3-         <forward>YahooConverter</forward> </status>
4-     <status value="CMD_ERROR">
5-         <forward>YahooStockPrice</forward> </status>
6- </command>
7- <command name="YahooStockPrice">
8-     <status value="CMD_OK">
9-         <forward>YahooConverter</forward> </status>
10-    <status value="CMD_ERROR">
11-        <view>errorPage</view> </status>
12- </command>
13- <command name="YahooConverter">
14-     <status value="CMD_OK">
15-         <view>Page1</view> </status>
16-     <status value="CMD_ERROR">
17-         <view>errorPage</view> </status>
18- </command>
```

11/22

## Additional Features

---

- DLL component built to handle triggering of remote web services and generating results
- Algorithm designed for extracting the outputs of a remote service as a list of results (this facilitates dealing with different types of inputs and outputs)
- Option included in VisualWebC to allow aliasing of remote components
- Facility for importing existing pages, loading them, and using them when generating the new web application

12/22

## Some current limitations

---

- The WebPads solution is “sensitive” to changes in the design of remote websites that provide services incorporated in the new application
- Changes in the I/O format of a wrapped component’s content may also cause malfunction of the new application
- “Provider” websites need be opened before the website that uses them is fully loaded; this means slower initial operation of the latter (the new application)

13/22

## Experimental studies #1

---

- Looking for answers to the following questions:
  1. How effective is our tool in the overall production of remote service-oriented web applications in development teams that were educated on the concepts of programming in *MVWf*-based frameworks but not in those of *VisualWebC*?
  2. How effective is our tool in the overall production of remote service-oriented web applications in development teams that were not educated on the concepts of programming in *MVWf*-based frameworks but were given a full tutorial on the use of *VisualWebC*?

14/22

## Experimental studies #2

- Setting:
  - 2 study groups, A and B, each of
    - 3 teams, each consisting of 3 students (1 graduate student, 1 senior undergrad, 1 sophomore undergrad)
  - Problem: Implement, using VisualWebC, a web application that, given the capital of a country, retrieves the following information for that capital:
    - 6-day weather forecast
    - List of hotels in the capital
    - Headlines of well-known international newspapers containing the capital's name
  - Group A was given a tutorial on MVWf (using Ruby on Rails framework) while Group B was given a tutorial on VisualWebC

15/22

## Experimental studies #3

- Results group A:

	<i>VisualWebC</i> Tutorial	<i>MVWf</i> Tutorial	Average correctness	Average time
<b>Group A Average</b>	No	Yes	97.7%	33.7 min
<b>Team A1</b>	No	Yes	97.5%	39.0 min
<b>Team A2</b>	No	Yes	100.0%	37.0 min
<b>Team A3</b>	No	Yes	95.5%	25.0 min

16/22

## Experimental studies #4

---

### ■ Results Group B:

	<i>VisualWebC</i> Tutorial	<i>MVWf</i> Tutorial	Average correctness	Average time
<b>Group B Average</b>	Yes	No	94.0%	14.7 min
<b>Team B1</b>	Yes	No	97.5%	15.0 min
<b>Team B2</b>	Yes	No	89.0%	17.0 min
<b>Team B3</b>	Yes	No	95.5%	12.0 min

17/22

## WebPads, Agile Development, SPLE #1

---

- WebPads-based development solutions are agile in nature:
  - Fast
  - Flexible
  - Adjustable
  - Reconfigurable
  - Component-based (largely, well tested and trusted)
  - Service-oriented (customer oriented)
  - Developer-oriented (user friendly)

18/22

## WebPads, Agile Development, SPLE #2

---

- WebPads can also serve well several major SPLE concerns:
  - Flexible reuse of software components
  - Dual software process (*domain + application* engineering)
  - Documentation, process and product organization
- One R&D direction to pursue could be on combining the soundness of the SPLE framework with the versatility of the agile approach; web applications might offer an excellent area of investigation for this.
- Example
  - Generic: online shopping *domain*
  - Specific: online shopping of automotive parts *application*
- The intersection of web development, webpads-based design solutions, agile methods, and software product line engineering, represents an open field of research

19/22

## Future Work & Conclusions

---

- In terms of web application engineering, the WebPads-based approach aims to increase the effectiveness of both development (design, implementation) and maintenance (evolution, upgrading)
- Direction of future work include:
  - Minimizing the impact of data format changes that happen at remote service websites
  - More rigorous model using client/server agents
  - Further development of the associated prototype tool, VisualWebC
  - Development of larger-scale applications
  - Exploring web engineering from a combined agile + product line perspective

20/22

# Main References

---

- Karam, M., Keirouz, W. and Hage, R. (2006) "An Abstract Model for Testing MVC and Workflow Based Applications," *Procs. of the 2006 IEEE Intl. Conference on Internet and Web Applications and Services (ICIW'06)*, pp. 206-211.
- Leff, A. and Rayfield, J.T. (2001) "Web-Application Development Using the Model-View-Controller Design Pattern," *Procs. of the 5<sup>th</sup> IEEE Enterprise Distributed Object Computing Conference*, pp. 118-124.
- Ito, K. and Tanaka, Y. (2003a) "A Visual Environment for Dynamic Web Application Composition," *Proceedings of the 14<sup>th</sup> ACM Conference on Hypertext and Hypermedia*, pp. 184-193.
- Ito, K. and Tanaka, Y. (2003b). "Visual Wrapping and Functional Linkage of Web Applications," *Proceedings of the Workshop on Emerging Applications for Wireless and Mobile Access*, Budapest, Hungary.
- Fujima, J., Lunzer, A., Hornboek, K. and Tanaka, Y. (2004). "Clip, Connect, Clone: Combining Application Elements to Build Custom Interfaces for Information Access," *Procs. of the 17<sup>th</sup> ACM Symposium on User Interface Software and Technology*, pp. 175-184.

21/22

# Thank you! ... Questions?

---



22/22