

**Aspects of Building Web Application
Systems Using the MBASE Approach**

Anca-Juliana Stoica

**15th Int Forum on SCM/Focused Workshop,
USC-CSE, Oct.2000**

Contents

1. Introduction
2. Web Application Systems
3. Architecture
4. Web Technologies
5. Aspects of Building WAS
 - (a) The Process
 - (b) Requirements and Use Cases
6. Additional Modeling Issues Connected to WAS
 - (a) Web Application Extension of UML
7. Applying the MBASE Approach to WAS Development
8. Cost Modeling and Estimation
 - (a) Modeling and Estimation
 - (b) An Algorithm for Estimation based on Use Cases
9. Conclusion

I. Introduction

Web Application Systems

WAS is a Web System (WS) that allows its users to execute business logic with a browser. A WAS:

- is dynamic
- facilitates collaborative contents, and distributed communities on the web
- deals with many different stakeholders
- has to be architected for continuous change
- has to cope with periods of peak interaction
- uses specific Web technologies such as: HTML, XML, EJB, SSL, CGI, TCIP/IP, ASP, JSP, ...

Model Based (System) Architecting and Software Engineering

The MBASE approach:

- is the most complete system of interconnected models for software development to date
- has a sound theoretical foundation
- offers support for analysing model clashes
- has flexibility, adaptability, ease of extension.

We found that the MBASE approach is very useful for representation of the success, product, process, and property aspects of a WAS and that there is a synergy between developing WAS and developing MBASE, in the sense that:

- MBASE offers the framework for analysing and modelling WAS to meet the critical success factors for software projects
- WAS offers the information to enhance MBASE models (or create new ones) for this specific software systems.

II. Web Application Systems

Web Application Systems are built by webifying client/server systems and integrating legacy. Typically these systems are associated with:

- an infrastructure - the Web system
- a business logic - the Web application
- an architecture for the Web application
- enabling technologies, dynamic clients, and distributed objects.

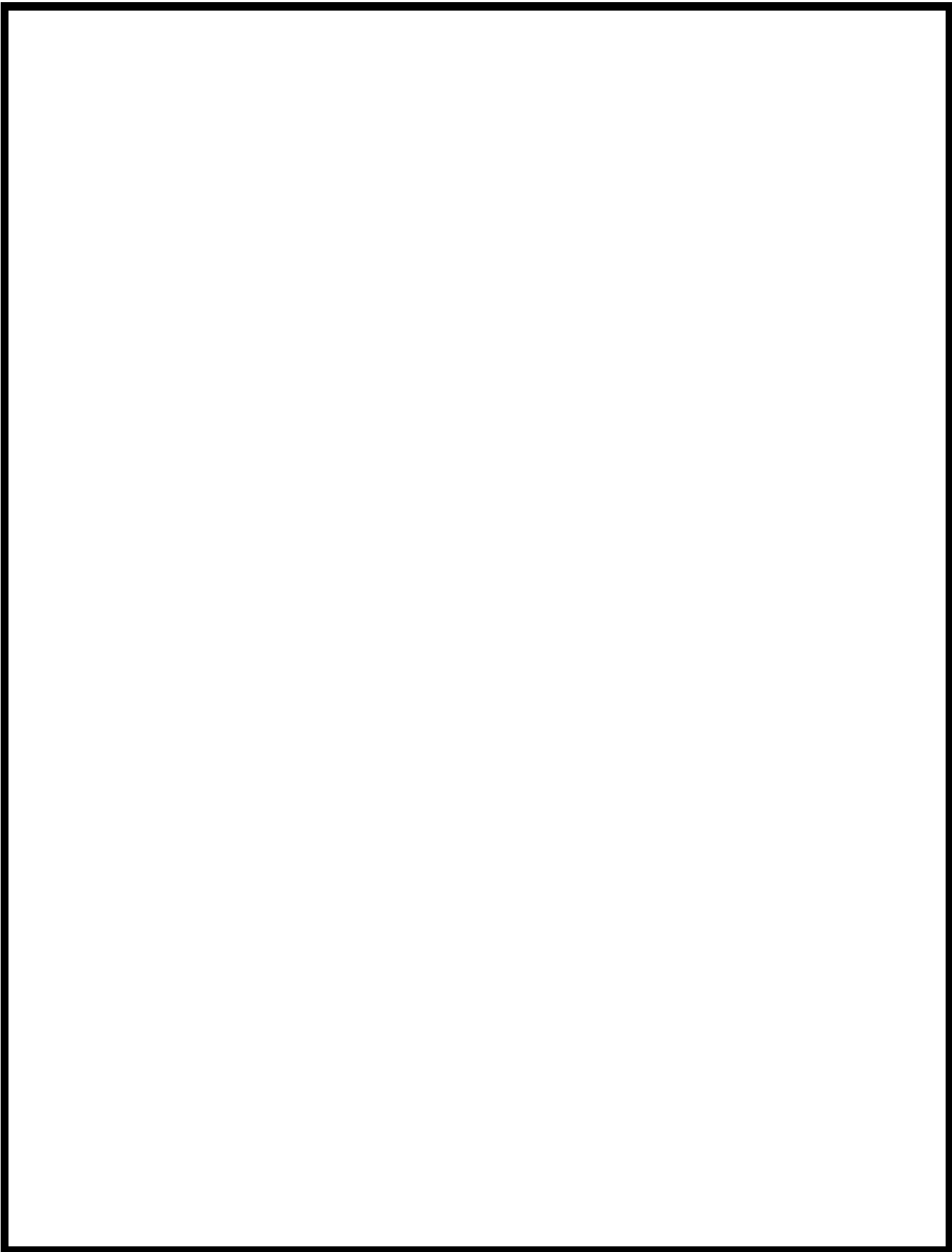
The most advanced such systems reengineer and automate complete businesses providing highly personalized, adaptive solutions. Some characteristics associated with developing WAS are:

- externally imposed and complex rules and regulations
- customer focus
- interface with legacy data structures
- increased concern for performance and reliability.

III. Architecture

The most basic architectural components are:

- client browser
- network connection
- Web server
- application server, which enables the system to manage business logic and state.



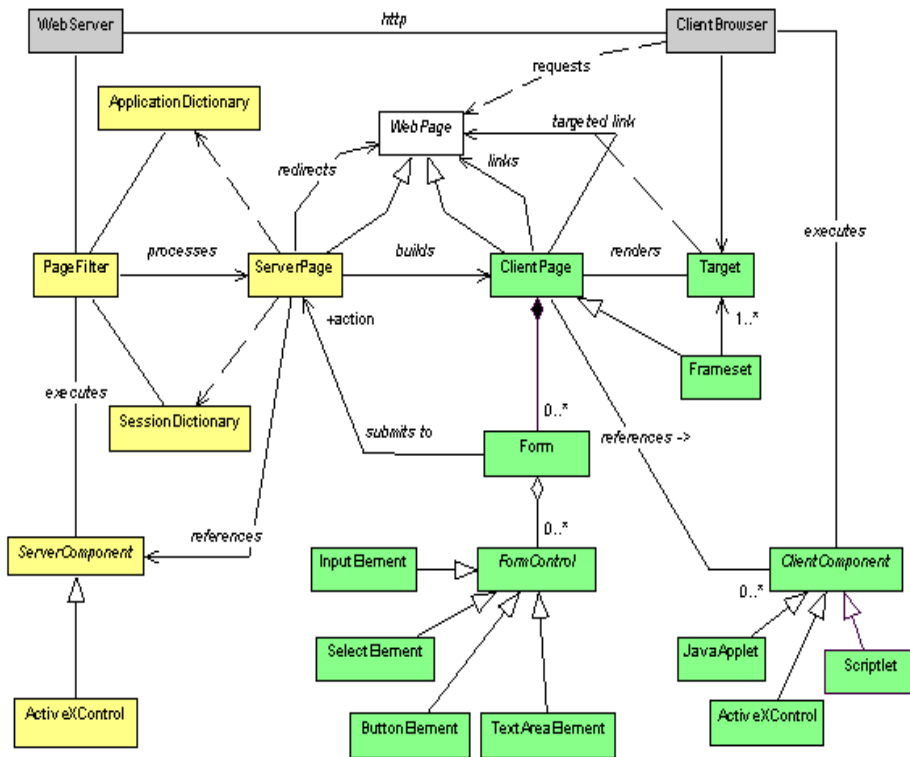


Figure 1: Generalized WAS architecture.

The basic architectural components are used in 3 architectural patterns, or fundamental structural organization schemas:

- Thin Web Client
- Thick Web Client
- Web Delivery System

The difference from traditional client/server systems is that WAS construct Web-specific components, namely Web pages. These are architecturally significant components because they exist both in the model and in the executable system.

IV. Web Technologies

Enabling Technologies

Provide the infrastructure for building Web applications, in particular the mechanism by which Web pages become dynamic and respond to user input. Examples include: CGI (Common Gateway Interface), Active Server Pages (ASP), Java Server Pages (JSP), servlets, and Web server APIs such as ISAPI (Internet Server API), NSAPI (Netscape Server API)

Enabling technologies are used as development environments that lead to new business opportunities. New business opportunities, in turn, lead to changed business practices (see Fig.2).



Figure 2: The role of enabling technologies.

Other Web technologies are used for:

- creating code from UML models: ASP, VBScript
- client-side scripting: HTML, XML, JavaScript, VBScript, Applets, Active X controls, DOM
- client-server protocols: HTTP, DCOM, CORBA/IIOP, Java RMI
- securing the WAS: schemes to protect confidential information over the net such as SET (Secure Electronic Transaction), SSL(Secure Socket Layer)

V. Aspects of Building WAS

A. The Process

- more customer orientation (“customer” here is defined as a user of a company)
- the role of use cases that: i) drive the process, ii) are a resource for nearly every activity in the process, iii) help manage and attack risks, and iv) every worker typically reviews the use cases to validate decisions made during work activities
- the relationships with the business process
- more stakeholders to consider for the Win-Win negotiations
- the influence of: enabling technologies, specific architectures, and new components (such as the Web pages)
- the “developing Web-time” approach
- the new approach to Web UI prototyping
- new artifacts

B. Requirements and Use Cases

- collection of documents and models to unambiguously describe the software to be built
- use cases capture and express the system behavior
- business models give input to the use case view and help to identify actors and use cases
- developing Web-time solutions imply to unify creative design and software engineering processes.

VI. Additional Modeling Issues Connected to WAS

A. Web Application Extension of UML

Icon

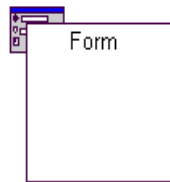
Server Page



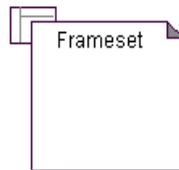
Client Page



Form



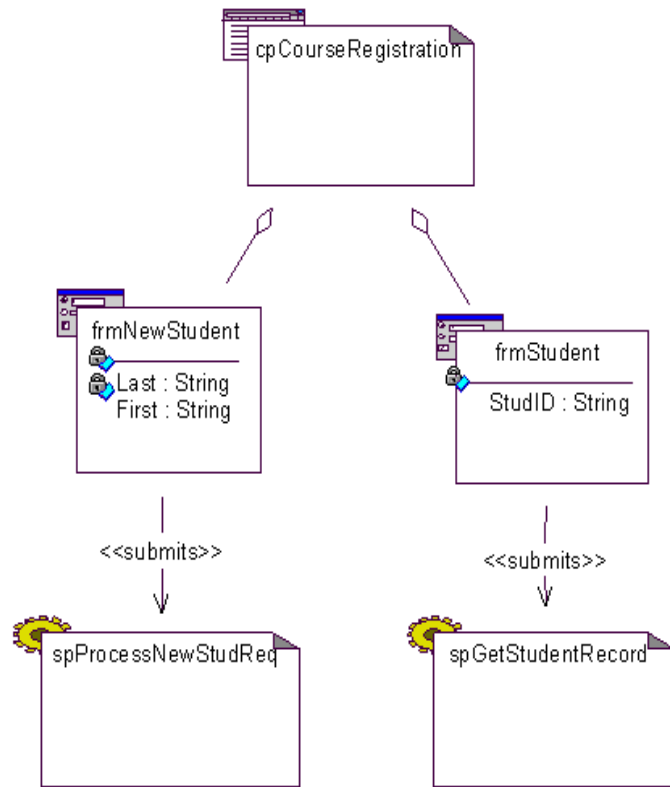
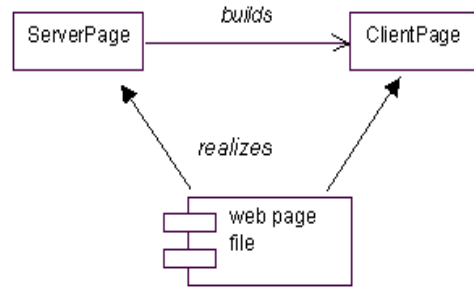
Frameset



Target



**Examples of using the Web Application Extension
of UML**



VII. Applying the MBASE Approach to WAS Development

A. Model Systems

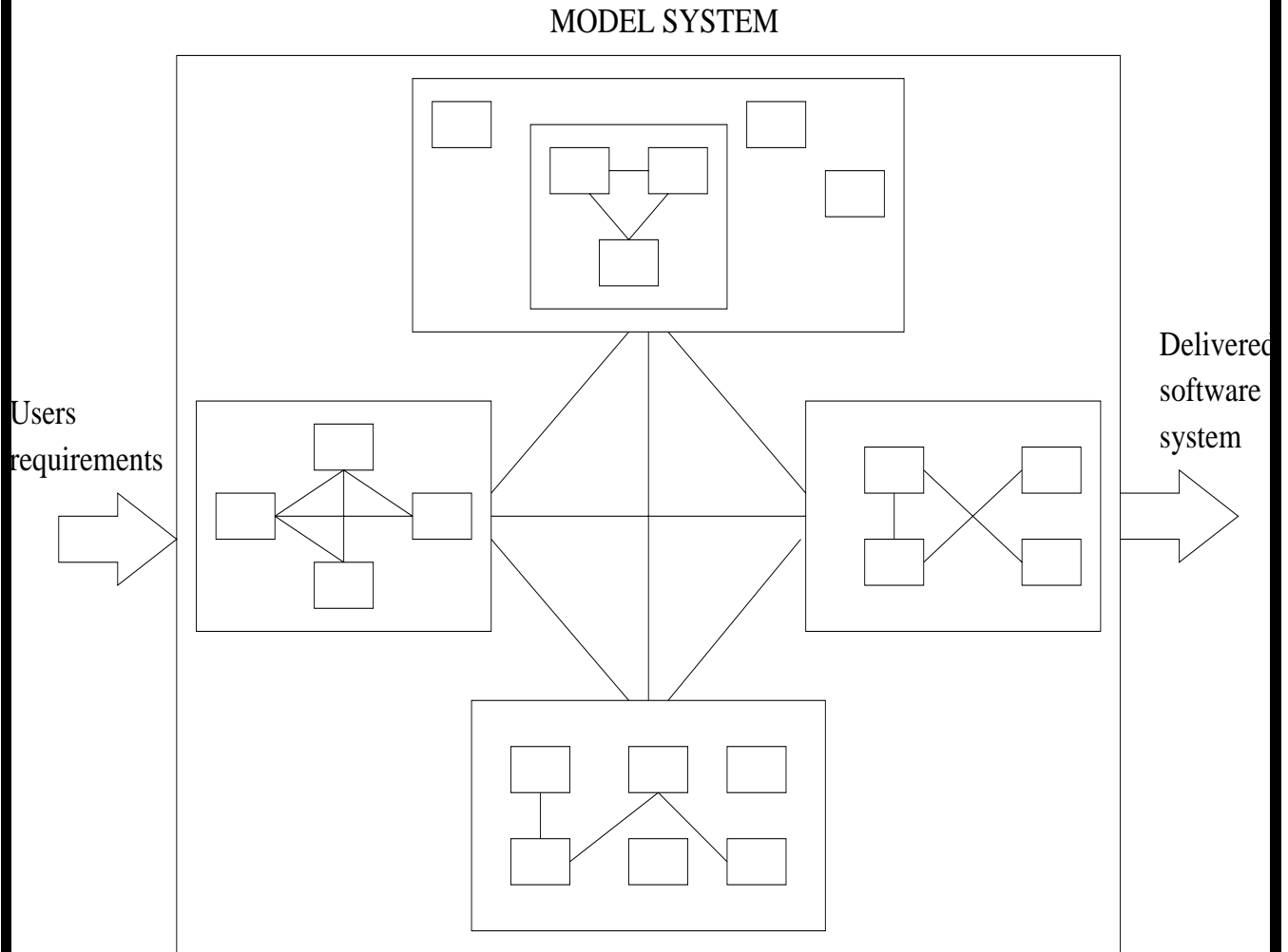
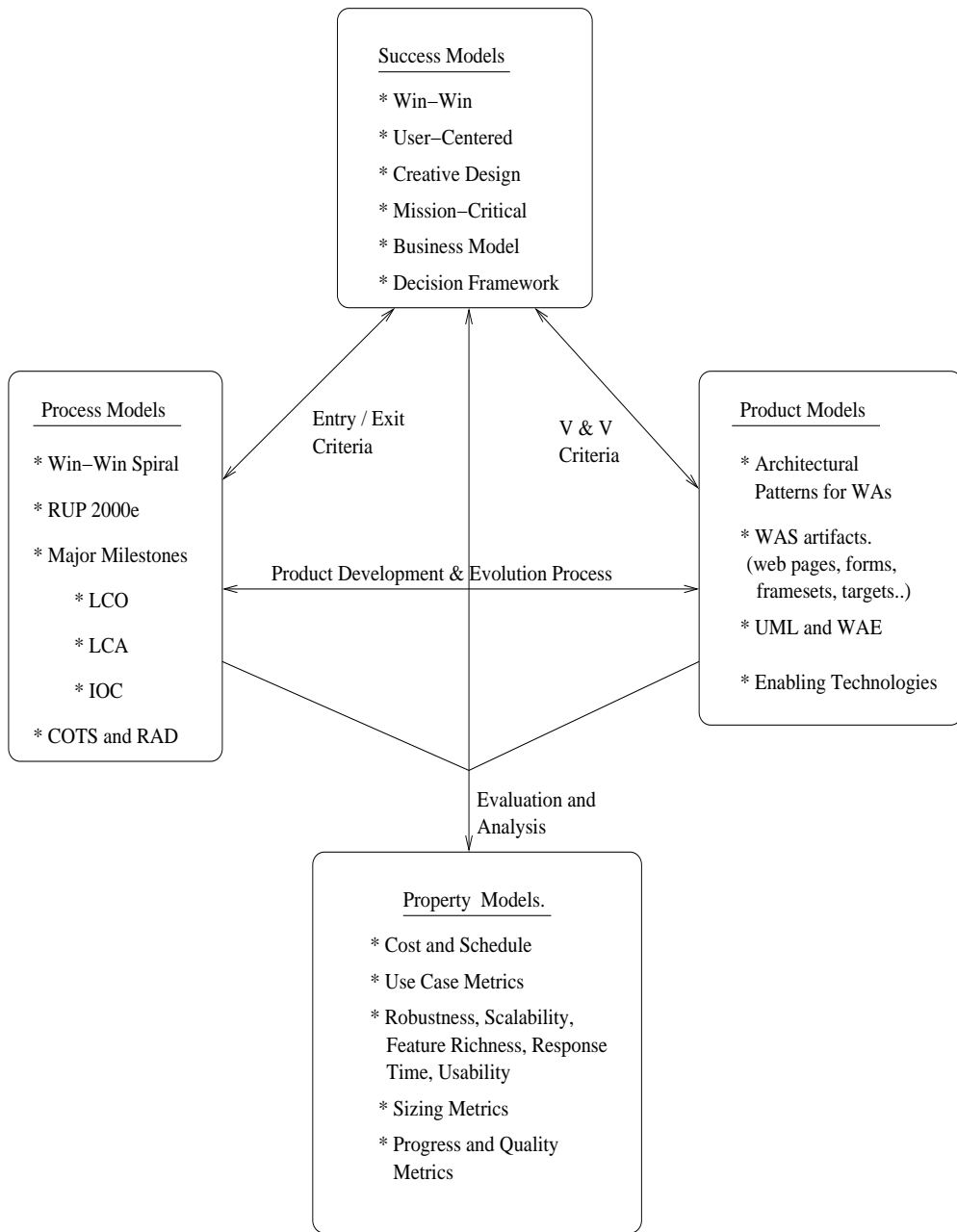


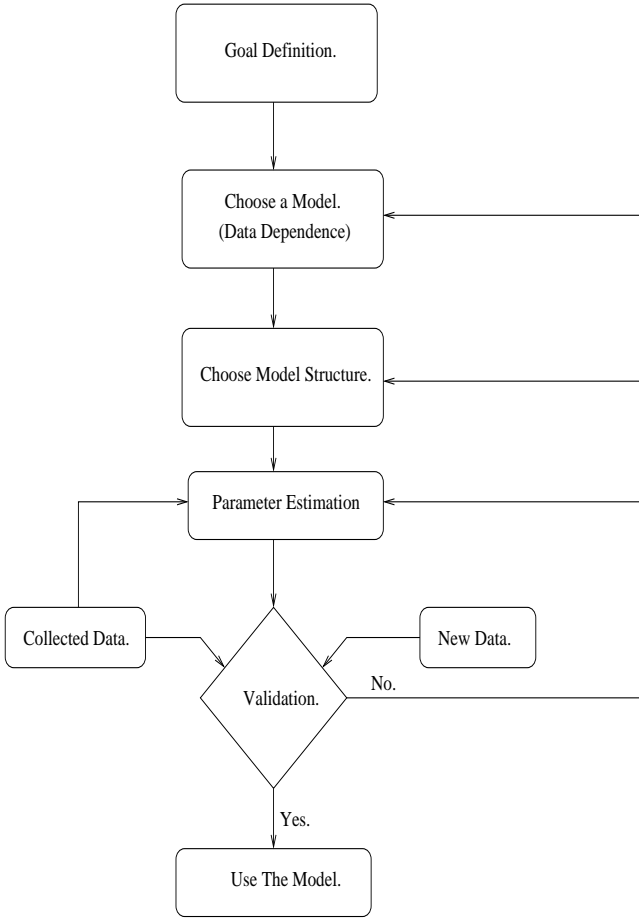
Figure 3: Model system.

B.The MBASE approach applied to WAS.



VIII. Cost Modeling and Estimation

A. Modeling and Estimation



An Algorithm for Estimating Work With Use Cases

1. Total actors

$$T_a = \sum_{i=1}^3 (na_i) * (Wa_i)$$

$i = S / A / C$. Complexity: Simple/Average/Complex.

na =number of average actors; Wa =Weight of an actor.

Actors complexity in terms of actor interaction mechanisms:
API / Communication Protocols / GUI.

2. Total Use Cases

$$T_{uc} = \sum_{j=1}^3 (nuc_j) * (Wuc_j)$$

$j = S / A / C$.

nu =number of use cases; Wuc =weight of a use case.

Use case's complexity in terms of transactions per U.C. or
number of analysis classes / U.C.

3. Unadjusted use case points

$$UUCP = T_a + T_{uc}$$

4. Technical Complexity of the Project

$$TCF = 0.6 + 0.01 * \sum_{k=1}^{13} (Vtf_k) * (Wtf_k)$$

13 =factors contributing to complexity (how difficult the system will be to construct).

TCF is similar to Function Points.

5. Experience level of the people on the project (The Environmental Factor).

$$EF = 1.4 - 0.03 * \sum_{l=1}^8 (Vef_l) * (Wef_l)$$

8 =factors contributing to efficiency.

6. Use Case Points

$$UCP = UUCP * TCF * EF$$

7. Project Estimate

$$E = UCP * ER \text{ [MH]} \text{ [MH]} = \text{man hour.}$$

$$ER - \text{Effort Rate [MH / UCP]}$$

8. Development Time

$$TDEV = E/N + \Delta T \text{ [Weeks]}$$

E - Effort [MW]. MW=man weeks.

N - Number of people in the development team.

ΔT - added time for woking out any team issues.

Reference: Geri Schneider, Jason P. Winters: “Applying Use Cases. A Practical Guide”. Addison-Wesley, 1998.

IX. Conclusion

- Web Application Systems (WAS) reengineer and change business practices
- Web specific elements of a modeling technique applicable for WAS are: Web pages, hyperlinks, their relationships to the back end elements of the system
- product models need to capture the execution of business logic in Web pages
- requirements specification and use cases use a series of translations to define a design model used to directly drive the code implementation
- designing decisions should be based on scientific methods instead of the ad-hoc ones
- new modeling and estimation techniques and algorithms based on use cases can be used by adding front-ends to the best existing cost models
- systems of interconnected models for software development are the best approaches for WAS.

References

- [1] *RUP99: Rational Unified Process*. Rational Software, 1999.
- [2] A.J.Stoica, *Facets of the Software Development Represented by Model Systems: Analysis and Enhancement*. 14th International Forum on COCOMO/SCM, 1999.
- [3] B.Boehm and D.Port, *Conceptual Modeling Challenges for Model-Based Architecting and Software Engineering*. USC Center for Software Engineering, 1998.
- [4] B.Boehm and D.Port, *Escaping the Software Tar Pit: Model Clashes and How to Avoid Them*. ACM Software Engineering Notes, Jan., 1999.
- [5] J. G.Schneider, *Applying Use Cases. A Practical Guide*. Addison-Wesley Object Technology Series, 1998.
- [6] H.Maruyama, K.Tamura, and N.Uramoto, *XML and Java. Developing Web Applications*. Addison-Wesley, 1999.
- [7] J.Conallen, *Building Web Applications with UML*. Addison-Wesley Object Technology Series, 2000.

- [8] R.Kalakota and M.Robinson, *e-Business*. Addison-Wesley Information Technology Series, 1999.
- [9] T.Quatrani, *Visual Modeling with Rational Rose 2000 and UML*. Addison-Wesley Object Technology Series, 2000.