Towards a Better Comprehensibility of Web Applications: Lessons Learned from Reverse Engineering Experiments

P. Tramontana

G. A. Di Lucca, A.R. Fasolino

Dipartimento di Informatica e Sistemistica

University of Naples Federico II, Italy

Web Applications (WA): problems and open issues

- The development of Web sites and applications is increasing dramatically to satisfy the market requests. The software industry is facing the new demand under the pressure of a very short time-to-market and an extremely high competition.

⇒ Web sites and applications are usually developed without a disciplined process: poor documentation is produced to support the subsequent maintenance and evolution activities, thus compromising the quality of the applications

Managing existing Web Applications

⇒ Due to the large number of employed technologies, understanding, maintaining and evolving a dynamic application is a complex task …

- *Reverse Engineering* methods and techniques have been proposed for…
  - Analyzing the functional behavior of an existing WA
  - Reconstructing the architecture of the WA
  - Capturing and reusing the design of the application
  - Modeling static and dynamic views by UML diagrams (use cases, sequence and class diagrams)
  - …

Problems in analysis

- Presence in a WA of several heterogeneous software components, developed with different technologies and coding languages

- Absence of effective mechanisms for implementing the well-known software engineering principles of modularity, encapsulation, and separation of concerns, may significantly make harder the comprehension of an existing WA

Reverse Engineering Web Applications

Five main steps:

1) Static Analysis of the WA

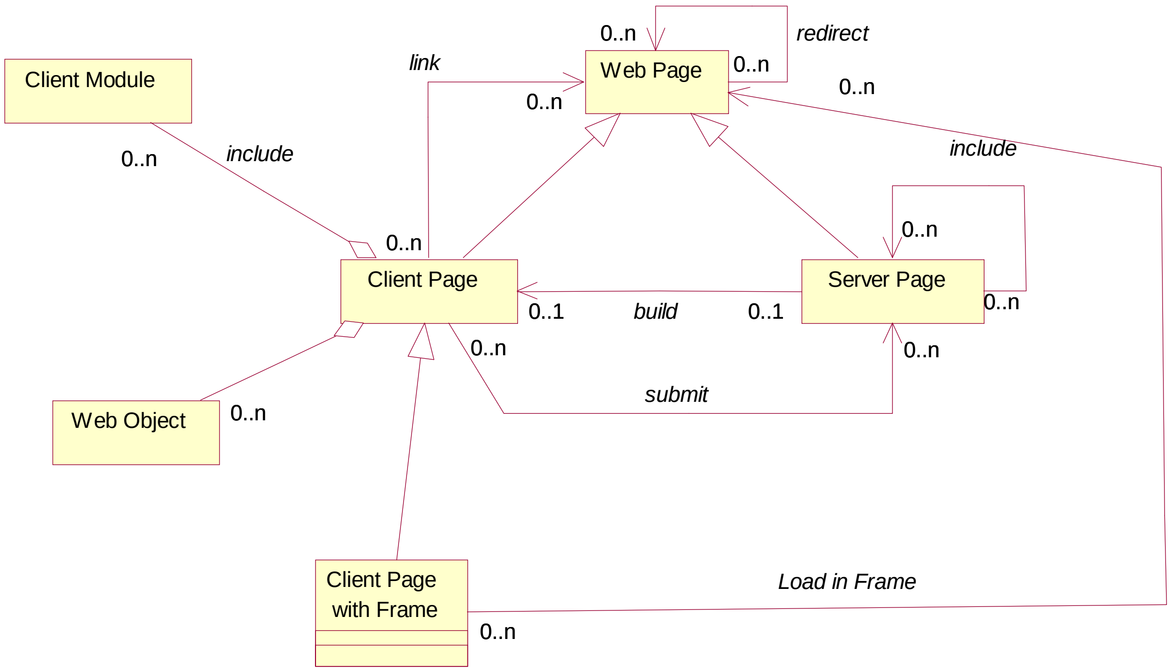
2) Dynamic Analysis of the WA

3) Automatic Clustering of the WA

4) Validation of the clustering

5) Abstraction of UML diagrams

The conceptual model of a WA



- *Components*: Client pages, server pages, client page with frames, client modules, web objects.
- *Relationships*: Link, submit, redirect, build, load_in_frame, include.

1) Static Analysis of the WA

- This kind of analysis can be carried out with the support of a multi-language code parser, such as WARE, a tool that statically analyzes HTML code, server script code (Vbscript, JScript, PHP), client script code (Vbscript, Jscript, Javascript).

2) Dynamic Analysis of the WA

- Recover of additional information about its components and inter-relationships.

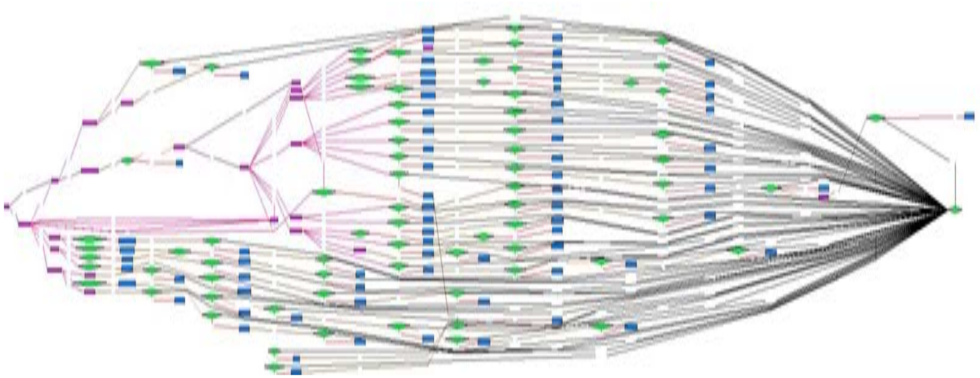
```
<%
 page1="a.html"
 page2="b.html"
param=Request.Form("parameter")
  x=Month(Now)
  If x>0 then
    response.write("<a href="+
page1+"?par="+param+">Go to page A </A>")
  else
   response.write("<a href="+
page2+"?par="+param+">Go to page B</A>")
  end if
%>
```

Example:

links between this page and a.html and b.html

2) Dynamic Analysis of the WA

- After static and dynamic analysis, we have the Web Application Graph (WAG), where every component is a node (the shape and the color of the node depend on the type) and every edge is a relationship (the color depend on the type)



3) Automatic Clustering of the WA

- The algorithm groups components of a WA into meaningful (highly cohesive) and independent (loosely coupled) clusters

- It evaluate the degree of coupling between interconnected components depending on both the typology and the typology of the connections, and propose a clustering configuration that includes clusters with high intra-connectivity and low inter-connectivity

4) Validation of the clustering

- Proposed clustering is submitted to a Concept Assignment Process (CAP) in order to validate them

- We distinguish between:

  - Valid clusters
  - Invalid clusters
    - Incomplete clusters
    - Divisible clusters
    - Spurious clusters

5) Abstraction of the UML diagrams

– Every functionality retrieved after Concept Assignment Process is used to reconstruct a use case diagram

Experimenting the reverse engineering approach

- The proposed RE approach has been experimented with 6 real WAs with different characteristics, and implemented using ASP, Javascript, PHP, and HTML technologies.

- According to Huang and Tilley's classification three of them were class 3 applications, two were class 2 applications and the last were a class 1 application.

1) Static Analysis of the WA

Summary data about the analyzed Was

Component type	WA1	WA2	WA3	WA4	WA5	WA6
Server page	75	105	21	0	0	0
Client Static page	23	38	19	80	45	257
Client Built page	74	98	20	0	0	0
External web page	0	0	5	2	34	8
Client script block	132	225	113	261	4	3
Function in Client script block	48	32	60	68	1	4
Form	49	100	5	0	25	5
Server script block	562	2358	40	0	0	0
Function in Server scripts	0	11	0	0	0	0
Redirect operation in server blocks	7	0	0	0	0	0
Redirect operation in client blocks	0	0	41	0	0	0
Anchor to Hypertextual link	45	266	121	162	448	1508

2) Dynamic Analysis of the WA

Problems :

- Connections with a dinamically istantiated value
- Connections dinamically istantiated
- Connections realized in extra-script object (e.g.: java applets, flash objects)

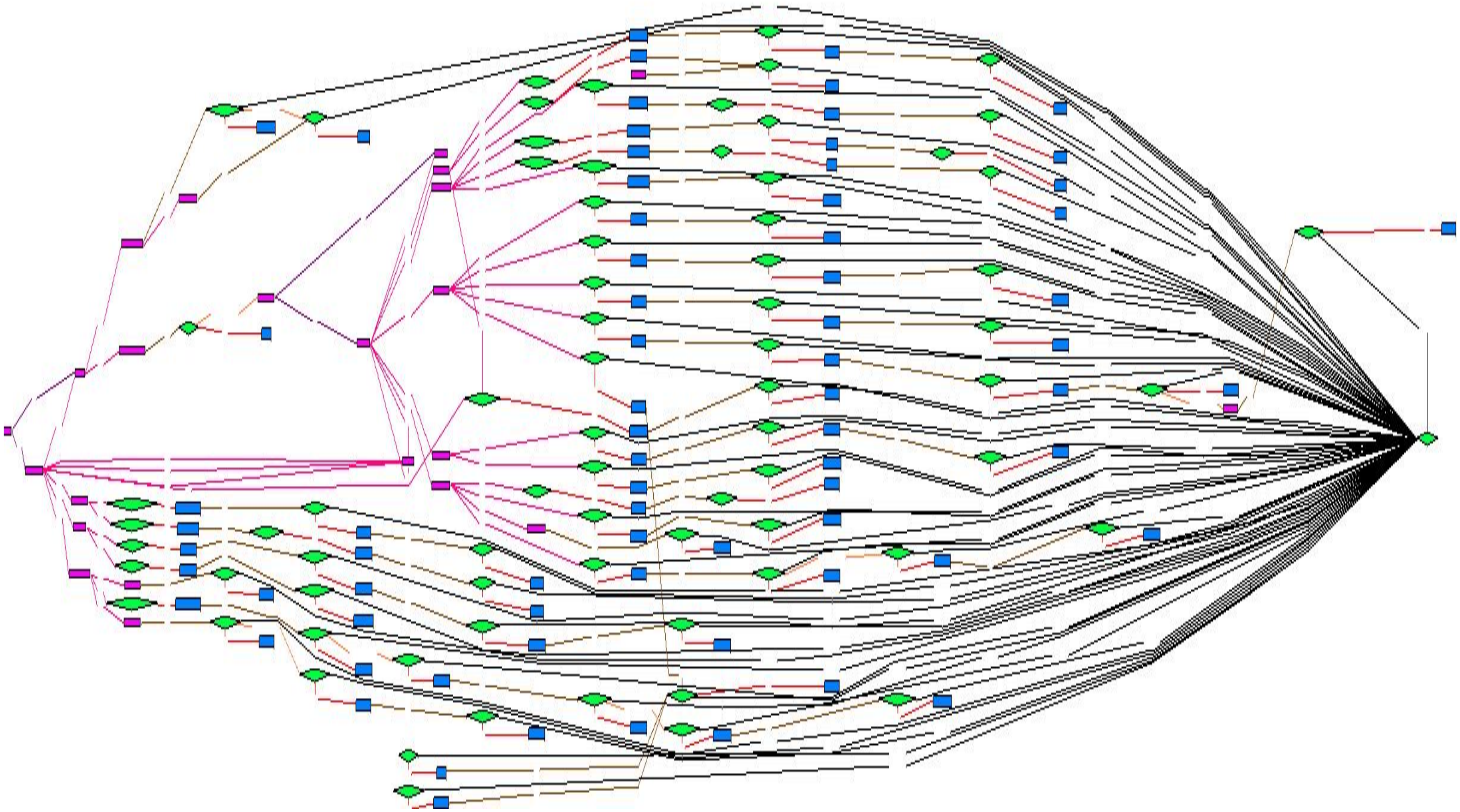
2) Dynamic Analysis of the WA

Dynamically retrieved information

Component type	WA1	WA2	WA3	WA4	WA5	WA6
Redirect operation in server blocks	7	0	0	0	0	0
Anchor to Hypertextual Link	0	1	9	0	27	0
Submit Form	0	32	0	0	0	0
Redirect operation in client blocks	0	0	27	0	0	0

2) Dynamic Analysis of the WA

WAG: Web Application Graph



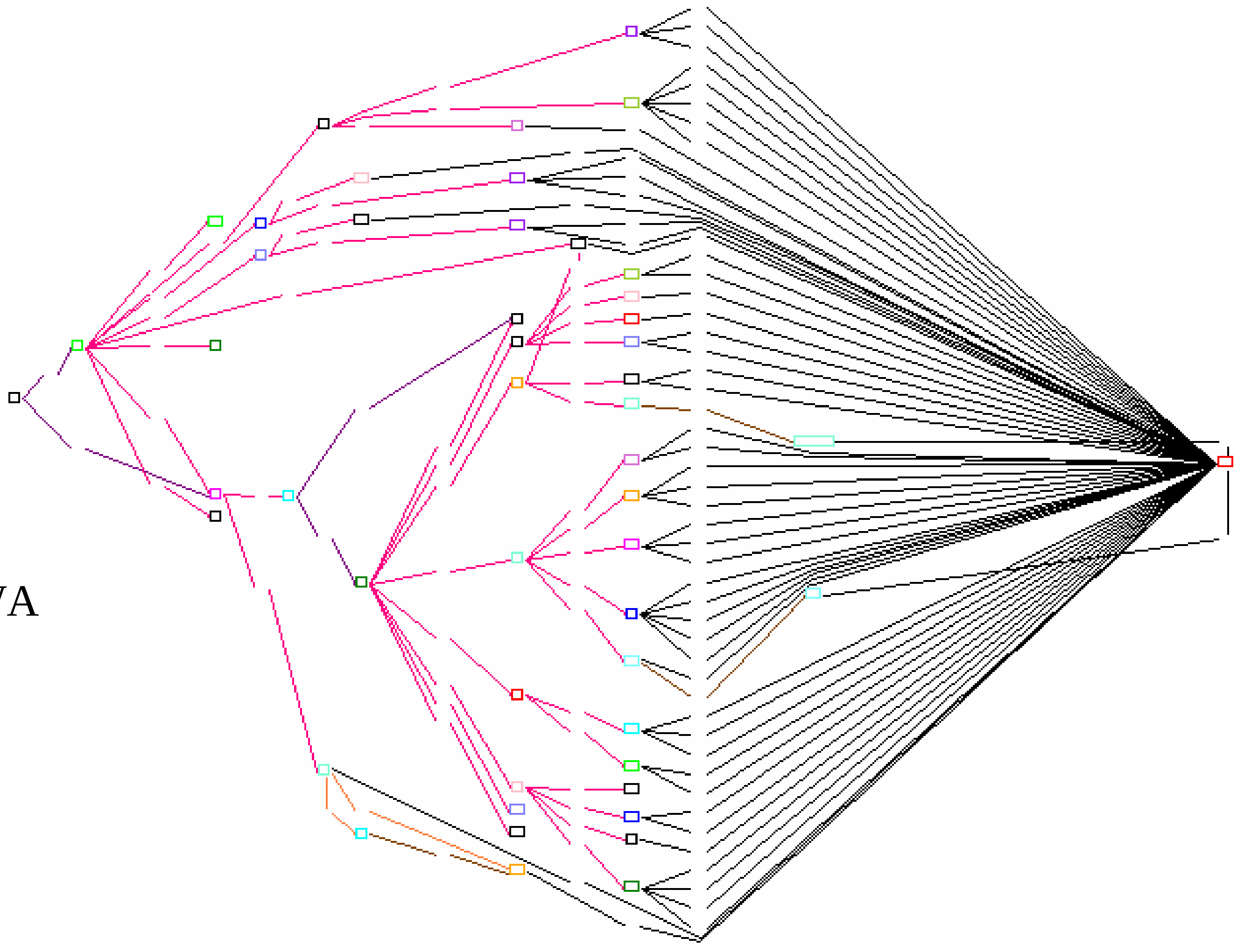
3) Automatic Clustering of the WA

- WARE tool automatically selects the optimal clustering of the WA, according to a quality metric based on the evaluation of the degree of the intra-connectivity of the clusters and of the degree of inter-connectivity.

Component type	**WA1**	**WA2**	**WA3**	**WA4**	**WA5**	**WA6**
Number of Clusters	48	101	31	58	31	122
Average # Pages per Clusters	3,58	2,41	2,03	1,55	1,45	2,17

3) Automatic Clustering of the WA

The clustered graph of the WA



3) Automatic Clustering of the WA

Problems:

- Presence of many navigation links, such as back links and cross links. These links are not representative of semantic relationships among the pages (e.g.: links associated with navigation bars)

4) Validation of the clustering

	WA1	WA2	WA3	WA4	WA5	WA6
Number of initial clusters	49	101	27	49	31	115
Number of spurious clusters	0	0	1	1	0	0
Number of split clusters	0	0	2	0	5	12
Number of incomplete clusters	8	15	3	0	0	0
Number of accepted clusters	41	86	21	48	26	103
Number of new clusters obtained from the spurious ones	0	0	1	2	0	0
Number of new clusters obtained from the subdivided ones	0	0	5	0	13	31
Number of new clusters obtained by merging incomplete clusters	3	7	1	0	0	0
Number of final clusters	44	93	28	50	39	134

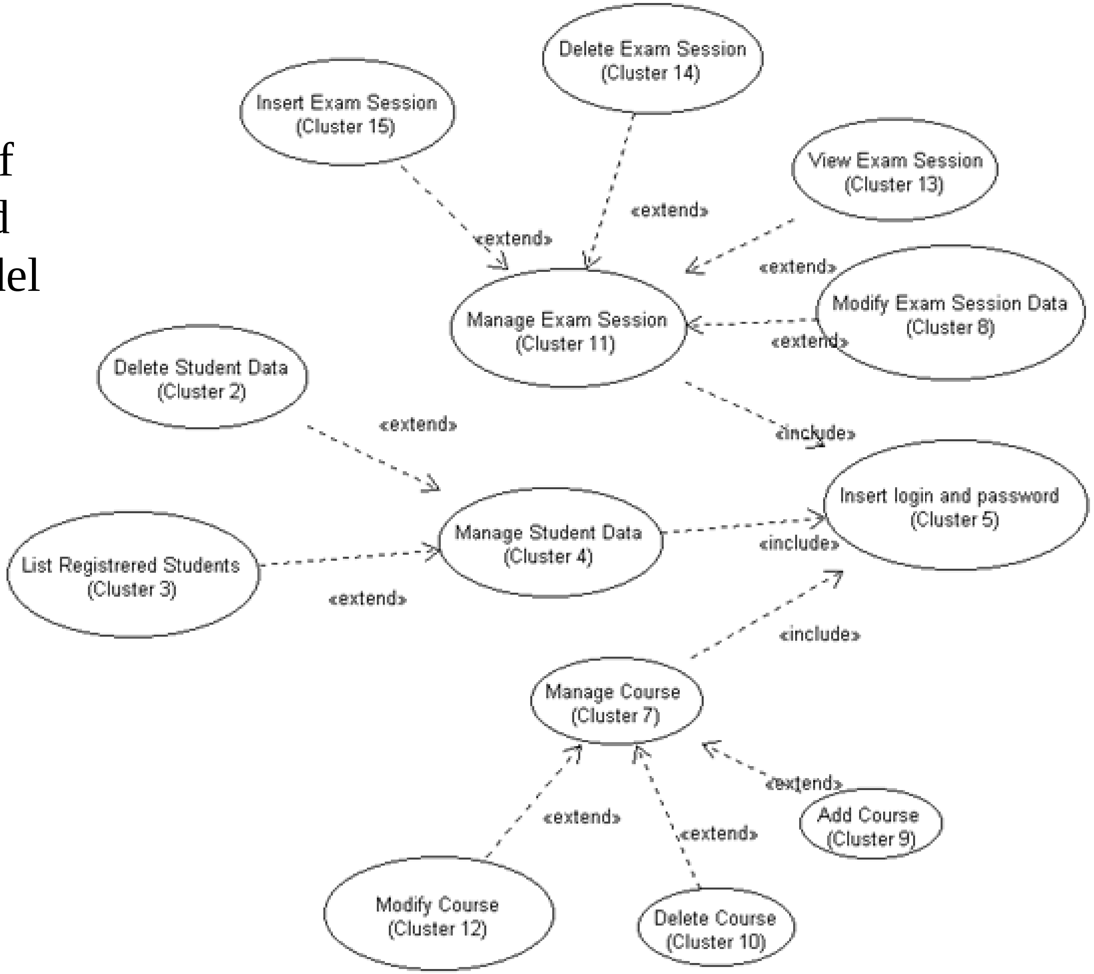
5) Abstraction of the UML diagrams

Result of the Concept Assignment Process

	WA1	WA2	WA3	WA4	WA5	WA6
Clusters that realise a use case	31	55	19	20	23	121
Clusters implementing coordinator/ home / menu page	12	13	2	11	12	16
Isolated Clusters	0	9	7	19	4	0
Clusters implementing Utility Modules	1	16	0	0	0	0

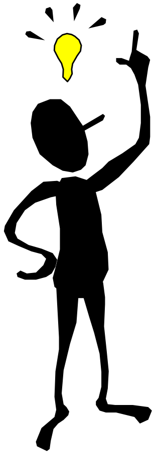
5) Abstraction of the UML diagrams

An excerpt of the recovered use case model



Discussion and lessons learned

- Dynamic analisys is time-consuming and expensive, since it cannot be completely automated.



In order to reduce the effort devoted to static/dynamic analysis, programmers should try to introduce explicit (static) links in the code each time it is possible

Discussion and lessons learned

```
<%
   param=Request.Form("parameter")
  x=Month(Now)
If x>0 then  %>
 <A href=a.html?par=<%=param%> >Go to page A
 </A>
<%  else %>
 <A href=b.html?par=<%=param%> > Go to page B
 </A>
<%  end if %>
```

√ Links between this page and a.html and b.html are retrieved with static analysis.

×The value of the parameter can be retrieved only with a multi-page dynamic analysis

Discussion and lessons learned

```
<%
 page1="a.html"
 page2="b.html"
param=Request.Form("parameter")
  x=Month(Now)
  If x>0 then
    response.write("<a href="+ page1+"?par="+param+">Go to page A </A>")
  else
   response.write("<a href="+ page2+"?par="+param+">Go to page B</A>")
  end if
%>
```

- Links aren’t retrieved with static analysis.
- Their value can be retrieved with a (simple) data flow analysis

Discussion and lessons learned

```
……….
<%
 'Read from a database the value to assign to page1 (i.e. a.html)'
 'Read from a database the value to assign to page2 (i.e. b.html)'
…………
 param=Request.Form("parameter")
 x=Month(Now)
 If x>6 then
   response.write("<a href="+page1+"?par="+param+">Go to page A </A>")
 else
  response.write("<a href="+page2+"?par="+param+"> Go to page B</A>")
 end if
%>
```

- Links aren't retrieved with static analysis.
- The value of the link can be retrieved only with a data flow analysis, comprehending to database

Discussion and lessons learned

•Some WA have a lot of navigation links, such as back links and cross links, and they have no semantic mean

•Use of *frames* reduces the number of navigation links

•Semantic of the link can be described with the ‘name’ attribute

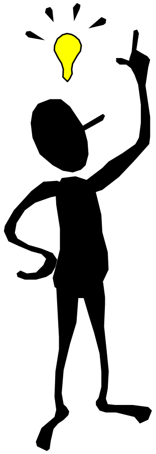
```
<title> Argument B </title>
…
<a name=crossA href=“argumentA.html> Argument A </a>
<a href=“argumentB1.html> Argument B.1 </a>
<a href=“argumentB2.html> Argument B.2 </a>
<a name=crossC href=“argumentC.html> Argument C </a>
<a name=backHome href=“index.html> Home Page </a>
```

Discussion and lessons learned

•The validation of the cluster represent a more expensive step of the RE process: we have to examine all the page of the cluster and their execution. So, this phase can be quite expensive if the clusterization has a low affidability

•Mirror the conceptual structure of the application into the directory structure of the file system, by locating groups of functionally related files into the same directory

Discussion and lessons learned



- Employ an internal documentation standard in order to annotate each main component of the WA

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<TITLE> check </TITLE>
<META NAME="Purpose" CONTENT="This page checks Login and Password of a Teacher,
then it redirects to Teacher Home Page">
<META NAME = "Incoming Links from Pages:" CONTENT = "/autenticazionedocente.html">
<META NAME = "Outgoing Links to Pages:" CONTENT = "/autenticazionedocente.html,
/areadocente.html">
<META NAME="Input Parameters" CONTENT="login,password">
<META NAME="Output Parameters" CONTENT="">
<META NAME = "Session Variables" CONTENT = "loginOK, matricola">
<META NAME="Included Modules" CONTENT="login,password">
<META NAME="Database" CONTENT="../basedatisito.mdb">
<META NAME="Images" CONTENT="bgmain.gif">
</HEAD>
...
```

Conclusions

- This paper presented an approach for Reverse Engineering Web Applications, and illustrated the results of an experiment carried out to assess which characteristics of a WA mostly affect its comprehensibility

- Lessons learned: the programmers did not abuse the mechanisms offered for obtaining dynamic behavior.

Future work

- More rules to improve quality of WA
- More powerful tools to automate dynamic analysis and concept assignment process
- Experimenting the reverse engineering approach to other case studies in order to validate and improve the methodology