

Identifying Reusable Components in Web Applications



P. Tramontana

U. De Carlini, A.R. Fasolino

Dipartimento di Informatica e Sistemistica
University of Naples Federico II, Italy



G.A. Di Lucca

RCOST – Research Centre on Software Technology
University of Sannio, Benevento, Italy

Outline

- Reusability in Web Applications
- Clones
- Clone Analysis Process
- An experiment

Web Applications (WA): problems and open issues

- Basic software engineering principles, such as modularity, encapsulation, or separation of concerns cannot be realized with some Web Application implementation technologies
- Maintaining and evolving these systems are difficult and expensive tasks.

Web Applications (WA): problems and open issues

- HTML language does not encourage the information content to be separated from its rendering
- Server and Client scripting languages don't encourage separation between Layout, Content and Business Rules

Web Applications (WA): problems and open issues

- Practice of developing new applications by cloning existing pieces of code is frequently adopted
- Web applications provide a valuable source of reusable software, which can be used with success to support their maintenance, reengineering and evolution.

Clone Definition

- *Clones*: Duplicated or similar portions of code in software artifacts
 - What kind of portions of code?
 - How measure similarity between these portions of code?

What kind of portions of code?

| Degree of granularity of a clone | Type of clone |
|----------------------------------|-----------------|
| Web Page | Client Page |
| | Server Page |
| Client Page inner components | Script Block |
| | Script Function |
| | HTML Form |
| | HTML table |
| Server Page inner components | Script Block |
| | Script Function |

Comparing source code

- Levenshtein distance:

Minimum number of insertions, deletions, and replacements of elements necessary to make two vectors equal

- Euclidean distance

Distance between portions of code

| Technique | Description |
|-----------|--|
| STH | Strings of HTML tags extracted from client pages are compared by the Levenshtein distance. |
| CTH | Counts of HTML tags extracted from client pages are compared by the Euclidean distance. |
| SOA | Strings of ASP built-in objects extracted from server pages are compared by the Levenshtein distance. |
| AMA | An array of software metrics extracted from ASP script blocks in server pages is used to compare server pages by the Euclidean distance. |

Metrics for AMA

| | Software Metrics |
|-------------|---|
| NTC | Total Number of characters per page |
| NTRC | Total Number of lines of code per page |
| NTBC | Total Number of ASP code blocks per page |
| NTSD | Total Number of declarative statements per page |
| NTSC | Total Number of conditional statements per page |
| NTSI | Total Number of cyclic statements per page |
| NTF | Total Number of Functions per page |
| NTS | Total Number of Subroutines per page |
| NTOP | Total Number of built-in ASP objects per page |
| LDBC | Levenshtein distance of ASP code blocks |

Phases of Clone Analysis Process

1. Separation of control and data component within Web pages
2. Clone Detection
3. Clone Validation
4. Template Generation
5. Repository population

Separation of control and data component

- Structure and data component are separated in HTML portions of code
- ASP block code is transformed removing blank characters and comments from the line, and substituting each data element (such as constants, or identifiers) by a dummy one.
- Consecutive blocks of ASP code will be merged in a single comprehensive block.

Clone Detection

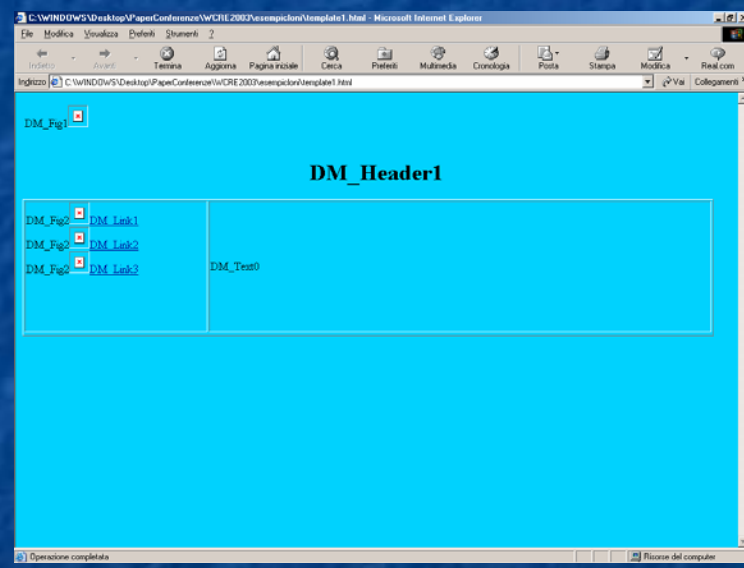
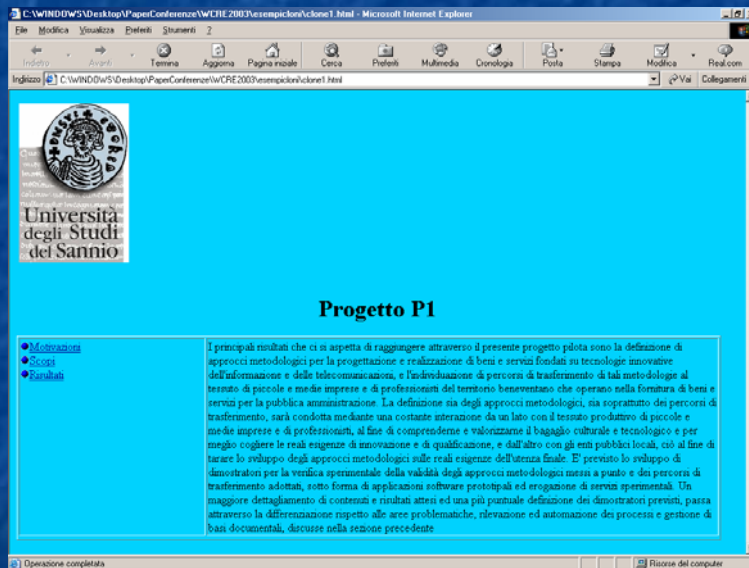
- Metrics are calculated for each component
- Clone analysis techniques are applied
- Clone and near-miss-clone are automatically detected with the support of CloneDetector tool

Clone Validation

- Clones (and near-miss-clones) are assessed by an expert, considering:
 - *meaningfulness* (does the considered clone implement a meaningful abstraction in the business domain, or in the solution domain?)
 - *completeness* (does the clone include all the lines of code necessary for implementing a valid abstraction?).
- Clones may be discarded, merged or accepted as reusable components.

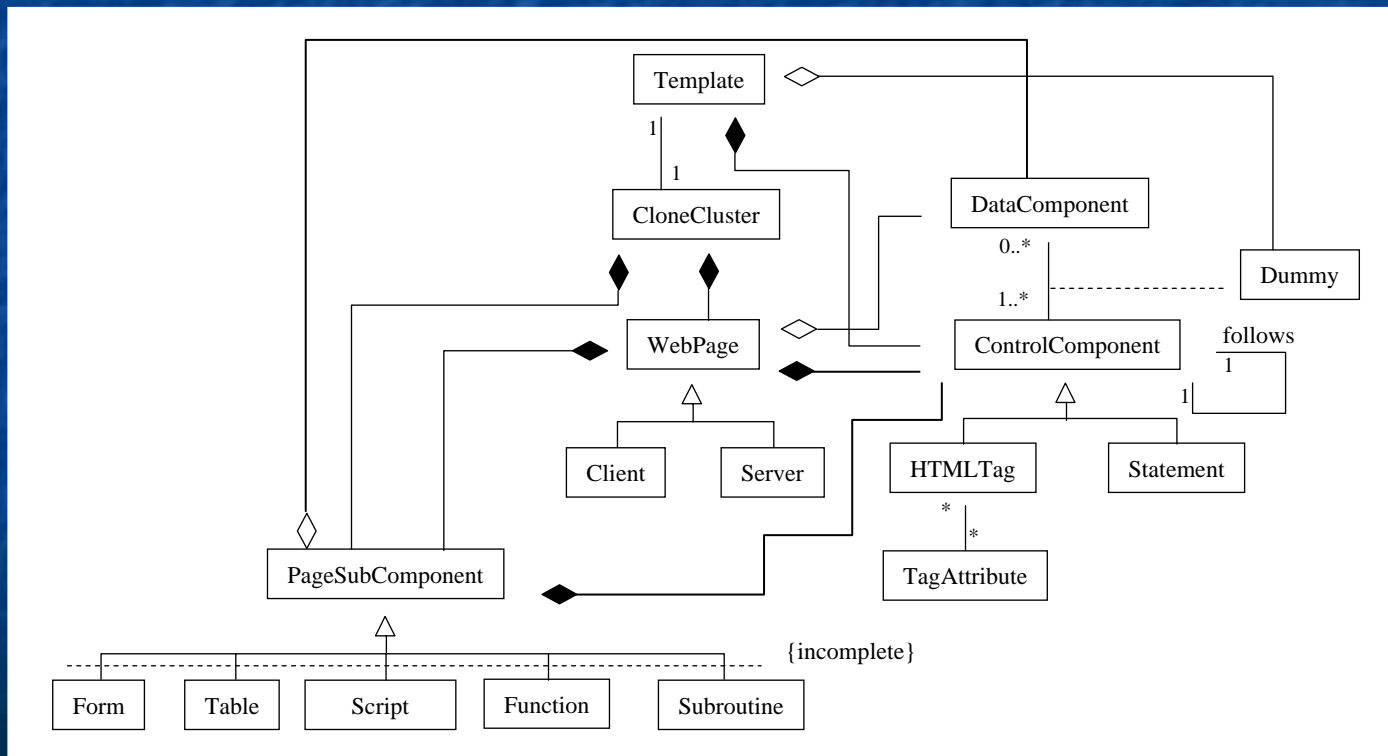
Template generation

- For each validated cluster of clones, a Template will be derived



Repository population

- Template information are stored in a repository, together with component information



Experiment

- An experiment has been carried out, with 4 real web applications
- We used CloneDetector tool to search exact clones of:
 - HTML pages (STH technique)
 - client scripts (STH technique)
 - functions in client scripts (STH technique)
 - forms (STH technique)
 - tables (STH technique)
 - ASP pages (AMA technique)
 - ASP script blocks (AMA technique)
 - functions/subroutines in ASP pages (AMA technique)

Results

| | WA1 | WA2 | WA3 | WA4 |
|--|-----|-----|-----|------|
| # HTML pages | 55 | 23 | 23 | - |
| # cloned HTML pages | 18 | 2 | 8 | - |
| # clusters of cloned HTML pages | 3 | 1 | 4 | - |
| # scripts in client pages | 3 | 12 | 10 | - |
| # cloned client scripts | 3 | 4 | 2 | - |
| # clusters of cloned client scripts | 1 | 2 | 1 | - |
| # functions in client pages | 12 | 10 | 10 | - |
| # cloned client functions | 12 | 4 | 2 | - |
| # clusters of cloned client functions | 4 | 2 | 1 | - |
| # forms in client pages | 1 | 8 | 10 | - |
| # cloned client forms | 0 | 2 | 9 | - |
| # clusters of cloned client forms | 0 | 1 | 3 | - |
| # tables in client pages | 7 | 4 | 3 | - |
| # cloned client tables | 0 | 0 | 0 | - |
| # clusters of cloned client tables | 0 | 0 | 0 | - |
| # ASP pages | 19 | 73 | 37 | 71 |
| # cloned ASP pages | 2 | 15 | 3 | 2 |
| # clusters of cloned ASP pages | 1 | 6 | 1 | 1 |
| # scripts in ASP pages | 75 | 576 | 150 | 5341 |
| # cloned ASP scripts | 4 | 0 | 36 | 0 |
| # clusters of cloned ASP scripts | 2 | 0 | 3 | 0 |
| # functions/subroutines in ASP pages | 5 | 0 | 3 | 165 |
| # cloned ASP functions/subroutines | 0 | 0 | 0 | 9 |
| # clusters of cloned ASP functions/subroutines | 0 | 0 | 0 | 3 |

Conclusions

- A clone detection method has been presented
- An experiment has been carried out
- Many cluster of cloned artifact has been detected and validated
- Some reusable component has been abstracted as templates

- Reengineering of static pages in a XML/XSL architecture may be performed
- Reuse of components or pages may be performed