# Online CASE Tool for Development of Web Applications

**Igor Jugo**
Department of Informatics
University of Rijeka
Omladinska 14, 51000 Rijeka, Croatia
ijugo@ffri.hr

**Abstract**. *Web based software ranging from full fledged distributed information systems, web based interfaces to larger classic information systems, web news portals and communities to dynamic websites of small companies is slowly but surely outnumbering the „static" Web. Research shows that the majority of them is not built using any kind of structured, disciplined, quality assurance process although many web software development methodologies and some CASE tools have been developed, tested and proven effective in the last 5-10 years. Developers usually justify the absence of a process in their work by time limits, strict budgets, need for additional software, process learning period, etc. In this paper we propose a CASE tool for the development of web applications that is available and used online by developers and clients. In this way web developers would work in their natural (online) environment, using a tool that enforces a structured approach, clients would be included in the life cycle reviewing and giving feedback, thus yielding better quality web applications. Creating a web based CASE tool with a classic application usage paradigm requires the application of latest web 2.0. technologies integrated with the strict formal rules of BPM, ERA and other notations used in the software development process.*

**Keywords.** online CASE tool, online web application development, web 2.0 technologies

## 1 Introduction

One of the basic ideas behind the creation of CASE tools is the possibillity to shorten the implentation phase of the software engineering process that ususally consumes up to 60% of the project time. Considerable time would be saved by automatic code generation based on the process and data diagrams crated and inserted through the CASE tool editors and into the data dictionary. The generated code would be consistent with the created models, syntax error free, and test data could also be autometically generated to help the developers test the software. The overall quality of the software is always based on the quality and accuracy of the requirements, models and constraints put in the CASE tools by all the members of the software development team. Full code generation of general software is one of the botllenecks of CASE tool technology but it has been made functional in focused areas by providing a set of process patterns that can be customized and connected to the data model.

As web applications have become ubiquitous and web interfaces to the classic infomation systems a default feature, we feel that an online CASE tool for the development of web applications as well as web application interfaces to large information systems would improve and facilitate the development and quality of such applications. The time for development would be shortened by using the CASE tool generators and the quality would be enhanced by the metodical process used to describe the application being generated which is an important fact, given the research results of the web software development practices in Croatia [9] that proved that „ad-hoc" and basic prototyping approaches are being used most of the time.

## 2 CASE definitions

CASE tools embedd methods and techniques for software development, software engineering knowledge and methodologies as well as the apropriate user interfaces.

Fisher [6] claims that CASE tools reduce or eliminate most of the design and development problems of complex software products by automatic generation of the majority of the product based on the crated models and diagrams specified by the software engineers. Thus, CASE tools enable software engineers to focus on the system architecture instead of system implementation. The ultimate goal of CASE tools by Fisher [6] is to „separate the design from the implementation of the software“ because the more separated these two processes are the better. Fisher [6] also states his definition of CASE that assumes „using tools that provide leverage in the requirements analysis and design stages as well as using tools that automatically generate source code based on the specifications and models. Brumec [6] defines CASE as „expert systems for software engineering“.

Today, most of the CASE tools are focused on some parts of the SE process like source control management, configuration management, prototyping, data dictionarys, user interface generators.

Sommerville [10] proposes a classification of CASE tools on tools (focused), workbenches (multiple tools supporting a particular SE phase) and environments (support the software life cycle, consist of more workbenches). This classification can be seen in Figure 1. The Figure does not display the herarhical nature of the classification.

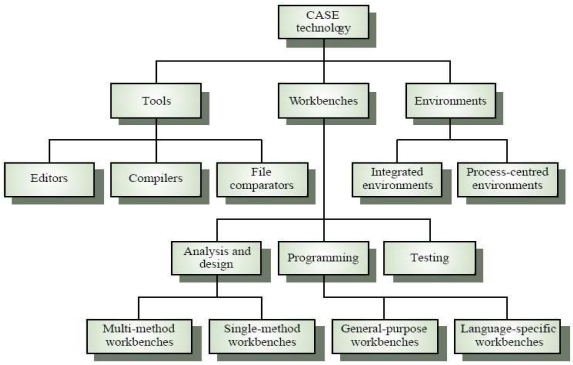


Figure 1. CASE tool classification [10].

CASE tools are usually categorized by the phases of the SE process they support. Usually they are divided in only two groups: UpperCASE (analysis and design) and LowerCASE (development, debugging, testing, QA. Brumec [4] proposes a more detailed categorization ranging from application generators to Integrated CASE environments. The categorization is displayed in Figure 2.

The CASE tool we propose in this paper will, in its first implementation an application generator tool, but in future implementations it can be extended to support all the phases of the life cycle thus becoming a „Full Life Cycle CASE“ tool.

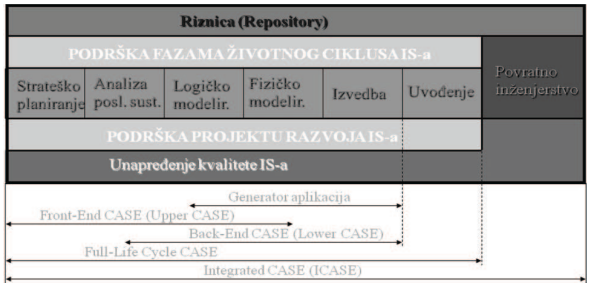


Figure 3. CASE tool categories [4].

# 3 Software methodologies and CASE tools

Full Life Cycle tools often support some defined set of activities, rules and tasks e.g. software engineering methodologies.

In 1990s Oracle developed the CASE*Method [1] based on the structural approach. It is a highly structured top-down process consisting of strategy, analysis, design, build and documentation, transition and production phases. Each phase has a defined set of activities and artifacts it generates that are used in the next phases. The whole process is supported by a set of CASE tools (Dictionary, Designer, Generator, Project, Bridge…).

In the following years object oriented analysis and design and object oriented development has been developed (powered by UML and OO programming languages) as well as iterative and component approaches to SE. New methodologies that integrate these advancements have been developed. One of such methodologies is the Rational Unified Process (RUP) [8] developed in IBM. The basic characteristics of this process are iterative development, requirements management, component architecture, visual modeling in UML, risk management, quality assurance, configuration and change management. Most of its disciplines are also supported by focused CASE tools.

All of these methodologies use various modeling techniques to describe the processes and data of the clients business. Today, new approaches like Business Process Modeling (BPM) are being used to eliminate the differences between the real business process and the model of the process crated by analysts. BPM Notation is being used to formally describe and model the current and future processes.

Following the fast development of the World Wide Web [3] and its economy, complex web applications have been created. Experts became interested in applying existing or creating new approaches to their development. These efforts created “web engineering” described in [7] as “establishment and use of sound scientific, engineering and management principles and disciplined and systematic approaches

to the successful development, deployment and maintenance of high quality Web-based systems and applications".
Further development of the WWW yielded even more interest which resulted in the creation of "The framework for Web Science" by a group of scientists at MIT [2].
Web engineering created many methodologies for web application development of which we would like to mention two: WAE2 and WebML.
WAE2 (Web application engineering) [5] is a method created in IBM, strongly influenced by RUP. It uses UML for the analysis and UML extension called "Web Application Extension" for the design phase. Web Application Extension is a set of stereotypes representing all the standard parts of web sites and applications like pages, server pages, hyperlinks, forms, fields, frames, etc. It introduces the User Experience (UX) model that contains modeled screens, storyboards and realizations, navigational paths and map, user input, screen compartments, etc., used to model the hypertext part of the web application.
WebML – Web Modeling Language [16] is a part of the methodology supported by the WebRatio CASE tool. This tool is a Java based software application, unlike the online tool proposed in this paper.
Using CASE tools for supporting the methodological approach should have the synergic effect on the final software product. The formal and disciplined approach should ensure quality (software meets specifications and purpose), and code generation by CASE tools should shorten the implementation phase, reduce number of errors, and testing.
The CASE tool proposed in this paper follows the iterative approach, starting from input of functional requirements, creation of the process, data and user interface models, writing detailed module and function specifications, code generation, and finally application generation.

# 4 WADOC(T) – Online CASE tool for development of Web

WADOCT is an acronym for Web Application Development Online Case Tool which clearly describes some of its most important characteristics. To further explain the ideas behind this project we can analyze parts of the name:
„WEB APPLICATION DEVELOPMENT" – states that the purpose of this tool is to support the development of web applications. Here we must clearly define and differ the terms web page, web site and web application. A "web page" is a:
a) text document containing some information structured using HTML (HyperText Markup Language) – this web page is called static.
b) text document containing code written in some server-side scripting language that is interpreted after a client request and generates some information structured using HTML. The code is executed on the server and can access multiple data sources and prepare the data (information) for presentation – this web page is called a dynamic.
A static or a dynamic page can be part of a web site which is a set of web pages interconnected using hyperlinks into a clear and informative structure, stored at a web server at a unique named location (URL) which makes it available to users worldwide.
A web application is defined in [7] as: "… software system based on technologies and standards provided by World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web browser". This definition should be extended because it mentions only the static technologies governed by W3C like HTML, and does not consider the programming languages, DMBS's, "third party" applications (that ensure required and needed additional functionality of the web browser) and other services and data sources every web application is based on. Therefore we could make a wider definition and define a web application as a software system based on a hyperlinked structure of dynamic pages containing code written in server-side programming languages, backed up by a database management system and other data sources and services, delivering the required content to the client in the form of information structured by technologies and standards of the W3C through a user interface, the Web browser application.
„ONLINE" – one of the special characteristics of this tool is the fact that it would be available on the World Wide Web making it unique to the best of our knowledge. Its online availability gives it the same advantages (and some disadvantages) all web applications have to classic desktop applications.
„CASE"– this is a CASE tool that supports the process of engineering a complete web application.
The basic plan of WADOC creation is to deploy a functional application generator in the first phase then add more functionality to support the full life cycle of the web application of web information system.

## 4.1. WADOC specifics

While having mentioned its most specific characteristics – the fact that this tool would be available online and the fact that it is CASE tool for web application development there are six aspects of this tool to be pointed out:
a) The (web) software that a user is developing is created and located on a test web server after the first iteration which makes it readily available for access and testing. After the first iteration that creates a minimal functional system we can define more functionality, define other modules and data structures and create another version. There is no additional software or installation procedure required. In this

way the development team and clients can be physically distant but still work together online. The clients can see the software, comment it, and answer questions e.g. be involved with the project.

b) WADOC has no need for custom interpreters and a compiler considering it is already located on a web server that has all the needed software for executing server side code and a DBMS installed. The proposed CASE tool would be created and would generate web applications based on free open source technologies known as the LAMP platform (LAMP = Linux OS, Apache web server, MySQL DBMS and PHP programming language). In this way the construction of the tool is simplified.

c) By applying Web 2.0 approach and technologies the usage paradigm of a classic web application will be changed to fit the classic application usage – users will be able to insert and upload data or documents, create and edit models and code seamlessly. The user interface would be created according to W3C standards and user experience best practices and follow the usual programming metaphors software professionals are used to.

d) WADOC enables an unlimited number of users per project. An initial user could register and add numerous other users (in various roles) to the new project. They will be able to work at different parts of the project at the same time, while the integrity of the work would be secured by the locking mechanisms and source control.

e) Another Web 2.0 characteristic is the active participation of users. As WADOC would create a full data dictionary for each project, new users could browse through the database of finished projects or use some parts of their functionality which would further facilitate the development of new projects.

f) One of the most complex part of the system would be full integration of all artifacts thru ought the project, from the list of requirements down to the functions and modules. This system would enable users get additional information about any part of the project (e.g. from what requirement was some module created or for a finished process we could open a code view for some part of the process model).

## 4.2. WADOC structure

In the previous chapters we stated that the first realization of WADOC will be an application generator. Brumec [4] defines the following characteristics needed to consider an application generator a CASE tool:

a) Keeps a system description for reuse

WADOC has and internal database that keeps all data about projects like: relation schema, function lists, code generated, user interface parts, generated web pages with full HTML, CSS, JavaScript and other code. After the project has been finished a full application can be stored on the web server or prepared to be transferred to the client web server.

b) Create a ERA model or a relation model of data

Considering that WADOC is a Web 2.0 mashup application the relational database editor can be imported from existing web applications like PhpMyAdmin (a specialized web application offering a visual interface to the MySQL DBMS for editing an executing relational schemas). When the basic WADOC architecture is extended to Full Life Cycle ERA method visual editor will be integrated. In this case WADOC would use custom built or COTS JavaScript components like MxGraph. Such components would have to be edited to add special notation symbols as well as the formal rules behind the ERA method. From the visual model a relational schema can be exported and executed on the test web server.

c) Use a data dictionary

As mentioned in previous paragraphs WADOC would create a data dictionary to keep all the necessary data about an application.

d) Use a knowledge base (or patterns) for creating procedures

As most web applications today are a part of larger information systems (for example e-banking web application is a part of the larger bank information system) we can state that the most of their functionality consist of reports (selects), forms (inserts), record changes (edit, delete). Because of this WADOC would in its application generator form offer a number of application procedure models which the user can then customize and connect to the data defined in the relational database or ERA model editor. The customizing process can be done using wizards or high level code editing or both. I the Full Life Cycle form WADOC would enable users to create a full process model using a visual editor (again based on custom build or COTS JavaScript components like MxGraph customized to enforce the formal rules of process modeling), from which a list of modules and functions can later be created and then customized by writing pseudo code. On such a detailed structure definition code would be generated by the WADOC code generator module that would combine the model created by the user and its own database of procedure models and the list of PHP functions.

e) Connect program procedures based on the process model

A web application is a set of connected dynamic pages. After a process model has been completed, a hierarchical list of modules would be derived. Based on that list the code generator would create a new virtual page per module and create functions bodies on the page. Each function would then be created

based on the descriptions in the model, the data model and the pseudo code (if a custom procedure).
WADOC user interface editor would enable the users to create a model of the interface based on that (virtual) page list of procedures and a set of HTML symbols. For example many web applications that display a list of records have a form (mask) below the list to add new records. In this way the user can drag/ drop some procedure like "DisplayAllRecords" to the virtual page, define the area that procedure will take on the page, and then create a form and its field, and then connect the form with the necessary data. In this way a full set of user interface models for the application can be made.
The user interface editor will also be used to define the global navigation structure of the application.
All the mentioned visual editors (process modeling editor, data modeling editor and user interface editor keep their data in two XML files. One file is used to preserve the layout of the model, and the other one to keep record of the model elements and their connections. The second XML file can easily be transferred into a database and later used for code generation. The WADOC structure is displayed in Figure 3.

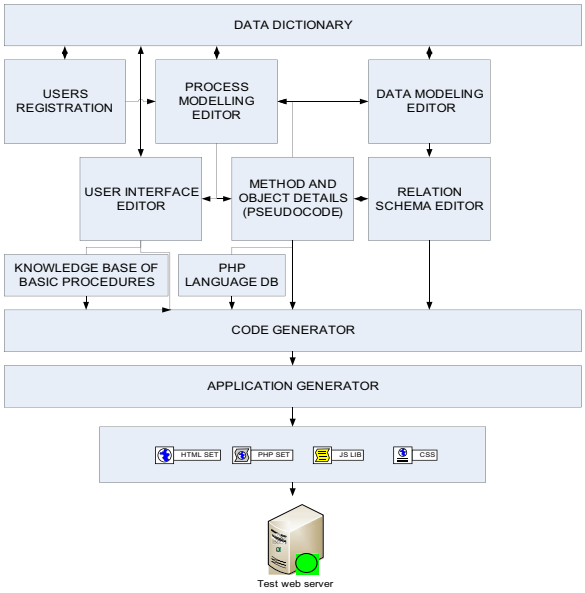


Figure 3. WADOC structure

The code generator would use the data dictionary, the W3C standards [17] for page structure, own procedure models and PHP function database to generate the code. The code would be connected to virtual dynamic pages. Each generated part of the code would be available for editing and review. As mentioned before full code generation is a bottleneck for many CASE systems but in this focused tool with a rather small list of used functions the code can be generated.

The application generator is the final step of the process that consists of the following algorithm:

```
create application directory @web server
create application database
create users
execute the relational schema @database
create CSS
create CONFIG
copy JS LIB'S
WHILE have_virtual_pages
  create new file
  open file
  create method tree with QS vars
    WHILE have_methods
      create method
      add code
      close method
   LOOP
add user interface elements to page
set default method
add connection to CSS
add connection to CONFIG
add connection  to JS lib's
LOOP
```

After the generation process the user will be able to use and test the web application. Ali files generated by user (or based on user input) can be edited while in production (test) phase. Also the models can be changed and the whole application generated again.

## 4.3. WEB 2.0 technologies used in WADOC

### 4.3.1. XHTML and CSS 2.0

XHTML (HyperText Markup Language) is the only language used to structure information on the web pages. It is defined by the XHTML specification document [18] issued by the World Wide Web Consortium. The basic structure of every web page is the same (every page consists of a heading (invisible – holds meta data and links the files that need to be included or referenced) and the body (visible to user, contains all the information structured by HTML)) which allows us to define the page creation process in the application generator. XHTML is a set of strict rules imposed on the general specification of the HTML. HTML is based on the box model meaning that elements that are opened/started first must be closed last. In this way we can create various structures to define the position and order of elements on a page.
CSS (Cascading Style Sheets) [13] is a style sheet language used exclusively to format the information and structure of HTML elements. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors and a declaration block. A declaration-block consists of a list of semicolon-separated declarations in braces. Each declaration

itself consists of a property, a colon (:), a value, then a semi-colon (;). In CSS, selectors are used to declare which elements a style applies to, a kind of match expression. Selectors may apply to all elements of a specific type, or only those elements which match a certain attribute; elements may be matched depending on how they are placed relative to each other in the markup code, or on how they are nested within the document object model. CSS is heavily used in Web 2.0 applications to completely separate the content from the presentation as well as to create a classic application like interface. Also, separate CSS files are created for each application so that it can be modified to share the look and feel of the client's web site.

#### 4.3.2. JavaScript platforms

JavaScript is a scripting language most often used for client-side web development. It was the originating dialect of the ECMAScript standard. It is a dynamic, weakly typed, prototype-based language with first-class functions. The primary use of JavaScript is to write functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page. JavaScript is heavily used in WADOC user interface in two ways: using JavaScript platforms (or libraries) to power online editors and using XmlHttpRequest object which is the main object behind AJAX that enables „behind the scenes" client->server communication.
JavaScript libraries are a set of objects, methods and special handlers built to add more functionality to a web application then initially made available by the browser's integrated JavaScript interpreter. On of the libraries used in this project is JQuery [14] which will enable „classic application"like manipulation of records and tables in the WADOC user interface (for example sorting tables, sliding elements to and of the workspace, etc.).
Another JavaScript library that can be used to power the visual editors of WADOC id MxGraph [15] which is a set of finished web based editors. This library would have to be modified with custom symbols and formal constraints that will support creating valid data models and process models.
AJAX (Asynchronous JavaScript and XML) [12] is the common name for a set of existing and evolving technologies: standards-based presentation using XHTML and CSS; dynamic display and interaction using the Document Object Model; data interchange and manipulation using XML and XSLT; asynchronous data retrieval using XmlHttpRequest; and JavaScript as the language that binds the static and the dynamic part of the page. In WADOC most data manipulation will be made using XmlHttpRequest enabling the user to use this web tool like a classic desktop application.

# 5 Conclusions

In this paper we presented a project of creating a complex web based CASE tool for development of web applications. Since many large classic information systems need a web interface we believe this tool should be focused on this group of web applications. CASE tools have been used for years as support for large software products development. In the first part of this paper we have defined what CASE tools are, their taxonomy and advantages. CASE tools are mainly used as support to a well defined software process or methodology like RUP. In this way software quality is assured by the process and the implementation phase is shortened by code generation. Many processes have also been proposed for web application development but are sometimes too complex and to abstract for developers. One of the most important characteristic of the Web Application Development Online Tool is the fact that it is an online tool available to all WWW users. It enables the users to create virtual collaboration between members of the development team and the clients. It uses the latest Web technologies to create a seamless desktop like environment. JavaScript, XML and DBMS powered editors enable the creation of detailed data model using ERA method, detailed process models and hierarchical module decomposition as well as user interface model for particular modules, pages and overall application. Editors would have detailed rules and constraints in order to ensure correct data and process modeling. Every module and function can be detailed by entering constraints, expected inputs and outputs. All the data about the project is kept in the data dictionary and connected thru ought the process. The most complex part of the tool would be the application generator. During the generation process the generator must follow his base algorithm, monitor errors, execute shell binaries, create database schema, and generate the code. Our future work will be based on the ERA modeling rules integration in JavaScript editors and optimizing the code generators algorithm. We believe that this tool would increase the quality of web application by adding a structured approach to its development process as well as reduce errors and shorten project realization time.

# References


[1] Barker, R., **CASE*Method Task and Deliverables**, Addison Wesley, USA, 1990

[2] Berners-Lee, T. i dr., **A Framework for Web Science**, MIT, 2006.

[3] Berners-Lee, T., Fischetti, M.: **Weaving the Web, the original design and ultimate destiny**

**of the World Wide Web**, Commercial Press, USA, 2000.

[4] Brumec, J., **Information and software engineering lectures**, PPT, FOI, 2004.

[5] Conallen, J. : **Building Web Applications with UML**, 2nd ed., Addison-Wesley, USA, 2003.

[6] Fisher, A.S., **CASE: using the newest tools in software development**, Wiley, SAD, 1988.

[7] Kappel, G.. : **Web engineering**, Wiley, Germany, 2006.

[8] Kruchten, P., **The Rational Unified Process An Intoduction**, Addison Wesley,SAD, 2004.

[9] Plantak-Vukovac, D., **Istraživanje paradigmi razvoja web mjesta**, magistry thesis, FOI, Varaždin, 2007.

[10] Sommerville, I. : **Software Engineering**, 7th edition, Addison Wesley, USA, 2004.

[11] Murugesan, S., Deshpande, Y., Hansen, S., Ginige, A. : **Web engineering: A new Discipline for Development of Web-based Systems**, Proceedings of the first ICSE Workshop on Web Engineering, LA, USA, 1999.

[12] **Ajax: A New Approach to Web Applications**, available at `http://www.adaptivepath.com/ideas/essays/archives/000385.php` , Accessed 24th June 2008.

[13] **Cascading Style Sheets Level 2**, `http://www.w3.org/TR/CSS21/`, Accessed 24th June 2008.

[14] **jQuery**: `http://jquery.com/`, Accessed 24th June 2008.

[15] **MxGraph workflow editor**, `http://www.mxgraph.com/demo/mxgraph/editors/workfloweditor`, Accessed 24th June 2008.

[16] **Web Modelling Language**: `http://www.webml.org/webml/page1.do`, Accessed 24th June 2008.

[17] **World Wide Web Consortium (W3C)** `http://www.w3.org/`, Accessed 24th June 2008.

[18] **XHTML™ 1.0 The Extensible HyperText Markup Language** http://www.w3.org/TR/xhtml1/, Accessed 24th June 2008.