

# Creating sophisticated web sites using well-known interfaces

*Fabio Vitali*

Dept. of Computer Science  
University of Bologna, Italy  
fabio@cs.unibo.it

## Abstract

ISA, the Immediate Site Activator, is a novel approach to web page production developed at the University of Bologna. It does not require authors to learn the features and quirks of the interfaces of specialized tools, or even less to master the number of technologies that are needed nowadays to create professional looking web pages (HTML, CSS, Javascript, not to mention XML, XSLT, etc.); rather, ISA lets them keep on using their well-known authoring tool (such as Microsoft Word or Adobe Photoshop) and builds sophisticated, professional looking, clean HTML pages behind the scene.

## 1 Introduction

The distinction between authors and readers of hypertext artifacts has always been a source for heated discussions among researchers and professionals in the field: many early hypertext prototypes had a very blurred distinction of roles. In many cases, the hypertext system was meant as a idea-collector for individuals or small groups of people (Hypercard (Goodman, 1987) is the most famous example of such systems), in others (such as Xanadu (Nelson, 1993)) the main philosophical principle was to allow everyone the right to access and tailor all published material for one's own purposes: clearly a hypertext tool that had little in the way of editing hypertext nodes was not considered appropriate. Only some publishing tool (for instance Hyperties (Shneiderman, 1987), meant for the creation of on-site museum displays) clearly distinguished between authoring and browsing activities and role types.

Things changed with the World Wide Web. Although the first WWW client written for NeXT computers by Tim Berners-Lee (Connolly, 2000) was in fact an editor, the number of plain browsers very soon largely exceeded editors (Marc Andreessen, it is said, found too difficult to create an editor for the Solaris operating system, and simply dropped off the feature in his early Mosaic prototype), and for almost everybody in the world the activity of creating content for the web has always been separated from the activity of browsing and reading existing content.

This separation has continued ever since. Browsers have been given away for free, and have rapidly become extremely intuitive tools readily usable by the majority of technically unaware Internet users. On the contrary, web editors have concentrated in providing more and more support for complex content and presentation effects, much to the detriment of simplicity, intuitiveness and ease of learning. These tools cater mostly for a wide range of web professionals, highly sophisticated and technically proficient, that need tools to increasing productivity and impact in porting to the web all sorts of content that have been created by others.

Current professional tools for creating web sites range from page-oriented HTML authoring tools that let users create single finely-tuned HTML pages (e.g., Macromedia DreamWeaver, 2003), to very sophisticated multi-user content management systems that support complex authoring processes for large scale dynamic web sites (e.g. Vignette, 2003). At the other end of the spectrum, many editing tools also allow to output content in HTML, but the result is usually simple, ugly, and visibly naive: the HTML created by graphic applications (Adobe Photoshop or

Macromedia Flash) is just a wrapper for the inclusions of images or multimedia animations, while what is created by word processors such as Ms Word is ugly, esthetically plain, and even simply and clearly incorrect.

The choice when authoring web resources is therefore restricted to either a complex and expensive tool that needs to be learnt and mastered to be used proficiently, but creates sophisticated, state-of-the-art web pages, or a simple and well-known word processor or graphic application that can create HTML code (not natively), but creates highly unsatisfactory and not professional-looking pages. It is hardly surprising that the former is bought by web professionals that already know the technologies, are not afraid to use them, and strive for good-looking results, while the latter is preferred by non-professionals that need a quick and easy output on the web, regardless of the final esthetical effect.

A separate discussion is needed for weblogs and wikis. Weblogs (Blood, 2000) are tools for fast web publishing of personal diaries, mostly free-form outburst of individuals and small communities. Weblog software is usually open source, requires some sweat in the installation and configuration, but once properly working it allows easy addition of content through web forms. Wikis (Leuf & Cunningham, 2001) are collaborative tools for the shared writing and browsing, allowing every reader (or, in some case, every authorized user) to access and edit any page of the site, with simple to use web forms and a very intuitive and reasonable text-based syntax for special typographical effects. Both may allow HTML templates to be automatically added to the content to improve the esthetics of the final result. At the moment, both weblogs and wikis seem to address the needs of special communities and marginal interest groups.

Clearly, an additional category of tools is appropriate: tools that may lack the sophistication and control that are provided by professional applications, but provide better results than those obtained by exporting HTML from a word processor. The purpose of this paper is to provide some details about one specific experiment in this direction: ISA, the Immediate Site Activator.

ISA is a server-side application that can create automatically both static and dynamic web sites by receiving in input layouts from well-known graphical packages, and content from well-known word processors. The main purpose of ISA is **not** to provide easy-to-learn interfaces for a new tool, but to allow users to exploit their previous knowledge of specific tools for the new task of creating web sites. Complete adherence to the existing interface of the selected tools is in fact at the core of the ISA philosophy, and it is in our opinion an interesting and innovative approach in the creation of new computer application.

## 2 Providing content for Web sites

If we exclude these approaches and home-made amateurish efforts, "real" web sites are still mostly created with traditional tools. Depending on the size of the site and the economical investment, these tools may range between some hundred dollars to several hundred of thousands dollars. Web site production happens in three or four usually well-differentiated stages:

- **Architecture design:** the overall structure of the site is designed, the nature and organization of the content is decided, the overall look and feel of the site is thought of. This task can be done with a variety of tools, or even on paper, or it could simply reside within the designer's head.
- **Graphic design:** one or many layout templates are created that provide the overall look of the final pages. The layouts are composed of a variety of content boxes, graphic decorations and any other textual or graphic part that is repeated across a number of similarly looking web pages (navigation menus, copyright notices, logos, etc). This job often requires a specialized graphic application used by a professional graphic designer. No professional looking web site can do without this step.
- **Back-end application design:** whenever the final pages are created dynamically, out

of queries to a server-side information system, a database has to be designed, implemented and integrated with the web server. This task can be performed by hand or with a wide range of extremely sophisticated tools, but it requires a computer professional, often a programmer, that can deal with the complexity and the technicalities of such task.

- **Content generation:** the generation of the actual information content of the site, usually composed of text blocks (from small text paragraphs to multi-paged styled documents), images, multimedia objects and any content which is not repeated across the site but belongs to a single web page.

There are many cases (e.g., simple sites composed of some dozens or hundred pages) where the architecture design can be done without specialized tools, and no real back-end database is necessary. But always content and layout need to be blended together to build the final web page. This is done in a variety of ways, depending on the complexity of the project: a web designer can be asked to use an HTML editor to insert the content within the page layout, therefore creating by hand each single page of the site, or a complex DBMS-backed solution can be employed, that provides the authors with a web interface for the insertion of the content, which is then dynamically used for the creation of the final web pages.

In the first case, a web designer proficient with HTML and (say) Macromedia Dreamweaver is necessary every time any content of the site needs to be added or changed. In the second case, authors are required to learn the specifics of the web interface to the database, which often lack common word processing features such as styles, spelling checker, temporary saves, etc; it is common practice in such situations that the actual writing is performed locally on a desktop word processor and the final result is then copied and pasted in the appropriate boxes of the web interface. Packages of an intermediate complexity (such as Macromedia Contribute, 2003) provide support for both direct modification and copy and paste. In all cases, a proficient web professional is required to set up and maintain the overall structure of the site, establish and control the publication steps and (in simpler cases) actually create and place the final pages.

### 3 ISA for web page creation

ISA (the Immediate Site Activator) provides a completely different approach to web content creation. The main idea of ISA is to exploit standard desktop tools for the creation of content and layout, and employ a server-side application for the delivery of the final web pages. This allows both types of users to keep on using well-known interfaces for doing web-related tasks, and eliminates the need for a web professional in any part of the production and maintenance of the site.

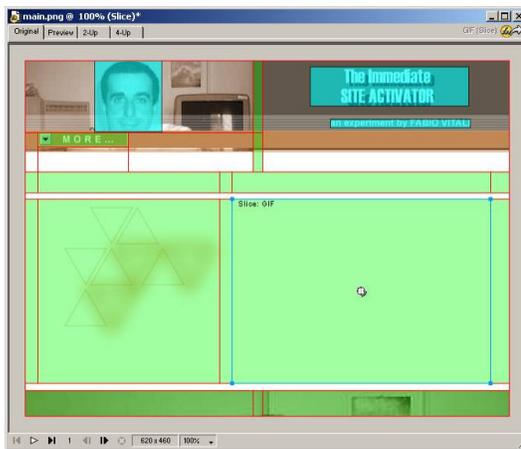
In the scenario of producing a web site with ISA, a graphic designer (having little or no awareness of the technicalities of the web technologies that will be employed) creates the overall graphical aspect of the page using a desktop tool such as Fireworks or Photoshop. She will then use the slicing tool of the application (many graphical packages implement it) to draw the active areas (the zones where the ISA package will act), and specify their properties.

ISA allows three kinds of areas to be created:

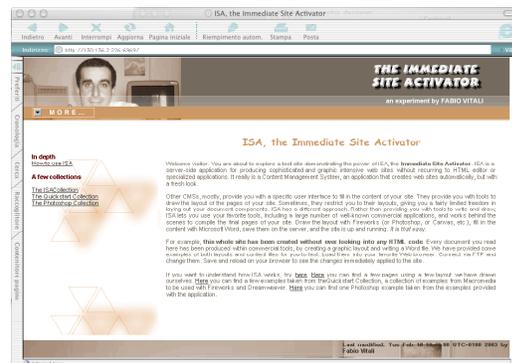
- **Text areas** In a text area (*fvblock*) one can name the parts of the content document that must be displayed or ignored. When the content document use styles (MS Word styles work fine, as well as CSS styles or HTML tag names), ISA bases the selection on them. The "body" style refers to the main section of the document. It is thus possible to separately specify what to display and to ignore as in "Include all of the body except for the "relatedlink" paragraphs".
- **Inclusion areas** In an inclusion area (*fvinclude*) sub-layouts or additional content documents can be placed. Navigation buttons that depend on the actual content, or

content fragments that are in external files (e.g. to create a *News* section, etc.) can be specified with the inclusion areas.

- Stretch and tile areas** A layout is liquid when it dilates to accommodate larger or longer content than expected or also to fill larger or longer browser windows than designed. A layout is usually composed of a layout table whose cells contain a slice of the overall graphics. Cells in the same column or row of the text content will therefore get taller or wider according to the dimension of the text. Stretch and tile areas (*fvstretch* and *fvtile*) allow these cell to contain not just the graphic as originally sliced by the table, but a stretched or tiled version of it. The overall result is of a homogeneous, liquid layout that correctly surrounds the text content of the page regardless of the dimension of the content. Of course, the decision between tiling or stretching the slice heavily depends on the type of graphics.

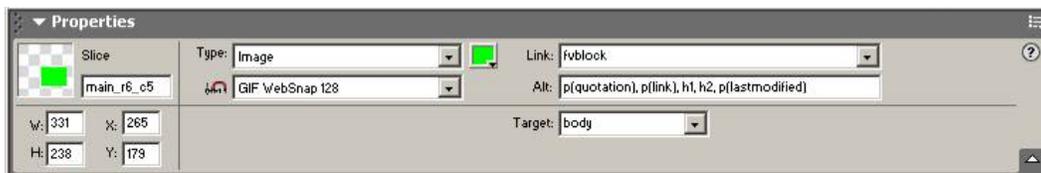


**Figure 1:** Drawing and slicing a layout



**Figure 2:** The final Web page created by ISA

The type of areas and the additional parameters are specified without any ad-hoc modification of the graphic package. Many graphic applications (including Macromedia Fireworks, Adobe Photoshop and ImageReady, Jasc Paint Shop Pro, Corel CorelDraw, Deneba Canvas, and countless others) allow slicing of the overall image in a table of cells; all of them include a mechanism to specify linking properties to the area. ISA uses the forms for specifying the link properties for the specification of its parameters. This requires no modification in the application package and a very low learning effort for using all of ISA features.



**Figure 3:** Specifying ISA parameters as link properties

After the layout has been created, the properties of the text are specified via CSS styles. ISA allows to specify a cascade of up to four stylesheets:

- A fallback sheet for basic default properties
- A browser-determined sheet for properties that are different in different browsers

- A layout-determined sheet for the properties that are specific of a layout
- A document-determined sheet for the properties specific to a single content document.

Each of them can be either created by hand (requiring the author to know CSS and deal with its syntax), or via MS Word: by saving a Word file with as many styles as foreseen in the real content documents, and the appropriate typographical properties, ISA will be able to deduce the CSS styles automatically and apply them to all the real content documents.

After the layout has been created, the content producer (having no knowledge of HTML or any other markup language) can proceed to write the content documents. He will use either an HTML editor, or, more frequently, a word processor. ISA knows and handles MS Word files with ease. The content producer will thus create any number of Word files, using styles as instructed by the layout designer, and saving them on the site as Web documents.

By providing the corresponding URLs in a browser window, ISA will merge the layout and the content document to form a complete web page. In usual web scenarios there would be the additional step of the web technologist joining the layout and content into a final artifact. ISA provides the services usually supplied by this intermediary.

## 4 Conclusions

While several tools already exist that remove the need for the intermediary web specialist, they still require layouts to be created in advance by sophisticated technologists, and indeed the task of creating such layouts can often be more complex and demanding than with less sophisticated applications. In our scenario, on the other hand, the designer can be technologically naive, just as the content author, and still be able to deliver the final product with no required awareness of the actual web technologies used.

ISA is a batch tool completely controlled by the data stored in the layout and content documents. No complex setup of the publication process is necessary. No specific web expertise is necessary in any part of the production process. The main goal of ISA is for users to keep on using their well-known tools, and to learn no new interfaces in order to create professional-looking web sites. ISA can be tested on line at <http://130.136.2.226:6969>. We consider it the first example of the leveraging of existing tools with no additional interface elements for a task that is somewhat outside of the main purpose of the tool itself.

## 5 References

- Blood, R. (2000). Weblogs: a history and perspective, September 2000, retrieved February 14, 2003 from [http://www.rebeccablood.net/essays/weblog\\_history.html](http://www.rebeccablood.net/essays/weblog_history.html)
- Connolly, D. (2000). A Little History of the World Wide Web, retrieved February 14, 2003 from <http://www.w3.org/History.html>.
- Goodman, D. (1987). The Complete HyperCard Handbook, New York: Bantam Books.
- Leuf, B. & Cunningham, W. (2001). The Wiki way. New York: Addison-Wesley, 2001.
- Macromedia corporation (2003). Macromedia Contribute Home Page, retrieved February 14, 2003 from <http://www.macromedia.com/software/contribute/>
- Macromedia corporation, Macromedia Dreamweaver Home Page, retrieved February 14, 2003 from <http://www.macromedia.com/software/dreamweaver/>
- Nelson, T.H. (1993). Literary machines, Ed. 93.1. Sausalito, CA: Mindful Press.
- Shneiderman, B. (1987). User interface design for the Hyperties electronic encyclopedia, 1st ACM Conference on Hypertext, November 1987, ACM Press, pp 189-194.
- Vignette corporation (2003). Vignette Content management Server home page, <http://www.vignette.com/>