Chapter 3. Web Applications and Mashups

Table of Contents

Web Applications

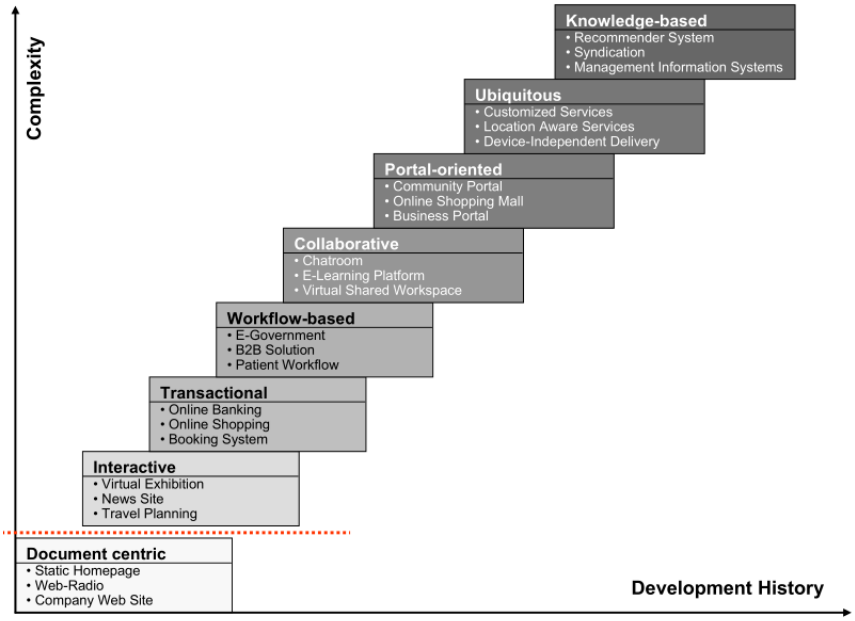
The definition of Web Application is a very similar way as the Web 2.0. We can find several articles and papers about it and its a challenging task to approve only one for it. In that way, we can use the outline from Wikipedia which collects it from several sources and describes it as the following: "A web application or web app is any application software that runs in a web browser or is created in a browser-supported programming language (such as the combination of JavaScript, HTML and CSS) and relies on a common web browser to render the application".

While this description contains most of the necessary keywords its useful to see the most frequently referenced one too: "A Web application is a software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web browser." - Source: Kappel, G., Pröll, B., Reich, S., Retschitzegger, W. (eds.), Web Engineering – Systematische Entwicklung von Web-Anwendungen, dpunkt.verlag, 2003 (in German).

It implies two things, one from the software aspect, saying that static HTML pages are not Web applications, and one from the interface aspect, outlining that Web services are not Web applications.

The evolution of the Web implies form its nature to include software solutions for serving content. As we seen in the earlier chapters, the third phase of Web 1.0 reached that level where we can say Web applications are present because they are not only simple "HTML file serving" solutions but complex multidisciplinary applications. Web 2.0 implies greater collaboration between websites, and greater interaction with website users. In fact, website might be the wrong term; as more and more functionality is provided via the browser, Web application is becoming a better description.

Web application categories



Web Application categories

On Figure 9 we can see the most widely used categories for Web applications. The development of a Web application can be started in any of these categories but complex Web applications can be assigned to several categories at once, e.g. Online Shopping Malls where transactional, workflow based and collaborative aspects could be founded at the same time. Naturally, new fields of applications are continuously created, e.g., location-dependent services are appeared just in the recent past. Newer categories are generally more complex but this does not mean they can fully replace the older generation and each has its own specific fields of application, e.g. Mashups which will be discussed int the next chapter as a new category near by the portals.

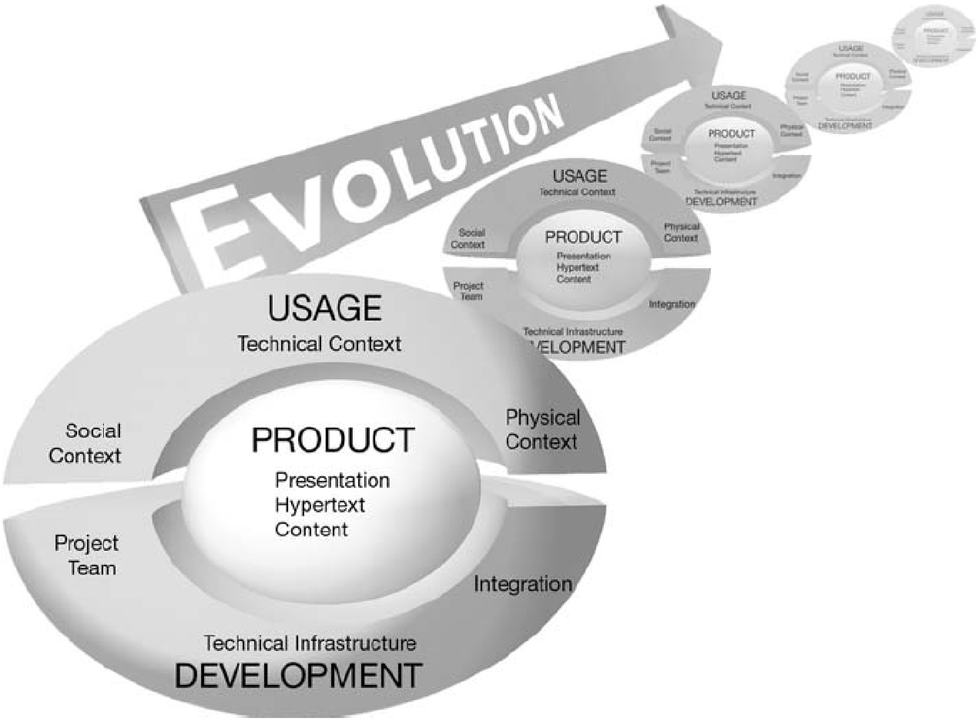
While the first category of them means the read-only web, the second category made the shift to the real web application development. This state could be treated as the starting point of the web applications. However, these category only means the usage of Common Gateway Interface (CGI) and HTML forms which offer a first, simple form of interactivity where pages and links to other pages could be generated dynamically according to user input. The real expansion arrived when more interactivity was achieved by the support of the database system forming the transaction-oriented part of the apps which made available the online transactions, e.g. online banking, shopping or booking.

All the remaining categories are really complex solutions are requires complex business solutions. Workflow-based Web applications allow the handling of workflows within or between different companies, public authorities, and private users where the appropriate Web services are guarantee the interoperability. This is the place where the e-... prefix is mostly applied: e-commerce, e-government, e-health, etc. Collaborative Web applications support cooperation in case of unstructured flow of activities and high degree of communication, like wikis, chat rooms or e-learning solutions. Portal-oriented Web applications provide a single point of access to separate, potentially heterogeneous sources of information and services.

The highest two categories are representing a more complicated ones. Ubiquitous Web applications provide customized services anytime, anywhere, for any device – this is the “ubiquitous access”. It has a very important precondition: knowledge of the context in which the Web application is currently being used in order to make dynamic adjustments to the Web application. Sadly, existing Web applications usually offer a limited form of ubiquity. Knowledge-based Web applications are forming the top and currently visionary layer where the goal is to present information on the Web not merely for humans, but in a machine processable form. Facilitate knowledge management on the Web based on Semantic Web technologies.

Characteristics of Web Applications

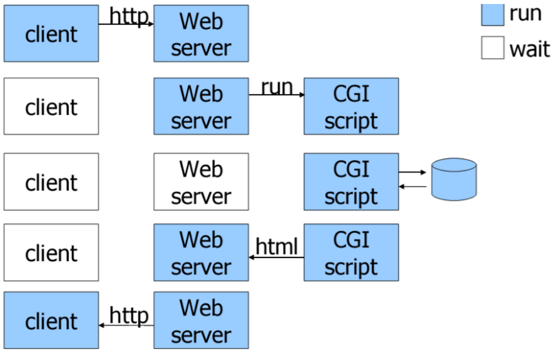
Web applications differ from traditional, not Web-based applications, in a variety of features. There are ones that traditional applications lack completely (e.g. non-linear navigation) or that are particularly pronounced in Web applications (e.g. frequency of updates). However, the presence and strength of a certain characteristic depend partly on the type of Web application, e.g., e-commerce systems vs. digital libraries. It means that proven methods from other related disciplines (e.g., software engineering or Hypermedia) are totally inadequate and thus new solutions have to be developed and have to be adapted to the needs of Web Engineering. All of these characteristics are well-formed on the following figure which is from the ISO/IEC 9126-1 standard for the evaluation of software quality.



ISO/IEC 9126-1

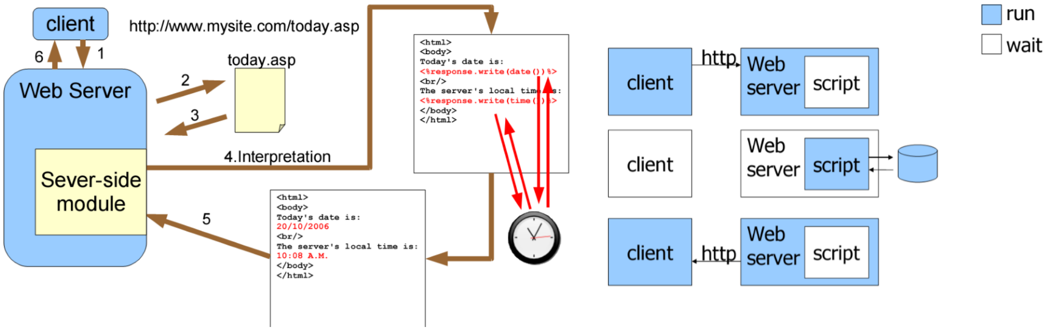
Traditionally, Web applications can be considered as a specific variant of client-server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures. In the early days of the Web each individual web page was delivered to the client as a static document, but from 1995, when Netscape is released its client-side scripting language called JavaScript, the World has been changed. Instead of sending data to the server in order to generate an entire web page, the embedded scripts of the downloaded page can perform various tasks. A little bit later an other technology is appeared to raise the user experience in a higher level, that was the Flash player plug-in from Macromedia. The real "Web application" term was introduced by Servlet specification of the Java language in 1999.

The evolution of the technologies could be seen in the following figures. The first one highlights the traditional Web 1.0 solution, the shift to the first dynamic webpages using CGI scripts:



Web 1.0 phase 1

The second phase was the introduction of the server side programming languages:

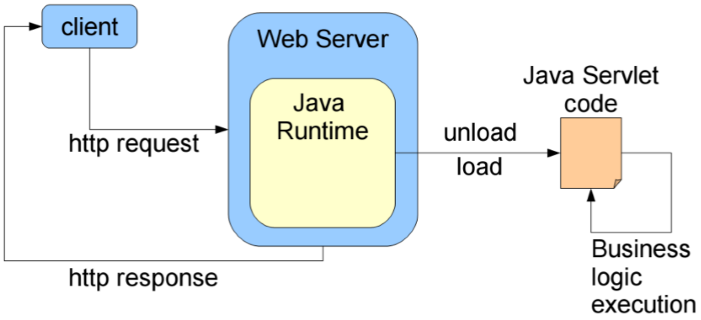


```
<html>
<body>
Today's date is:
<%response.write(date())%>
<br/>
The server's local time is:
<%response.write(time())%>
</body>
</html>
```

```
<html>
<body>
Today's date is:
20/10/2006
<br/>
The server's local time is:
10:08 A.M.
</body>
</html>
```

Web 1.0 phase 2

The third phase was the introduction of the application servers:



Web 1.0 phase 3

At the first times that was a good strategy for application software companies to provide web access to software previously distributed as local applications. Depending on the type of application, it may required the development of an entirely different browser-based interface, or merely adapting an existing application to use different presentation technology. However, legacy application is not always extendible for Web presence, and if so, than the solution could be very fragile. In practice, we could found that a scalable and responsive Web application need to use Web technologies from the beginning rather than extending existing solutions.

Writing of web applications is often simplified by web application frameworks. These frameworks facilitate rapid application development by allowing a development team to focus on the parts of their application which are unique to their goals without having to resolve common development issues such as session handling, workflows or simple cases as user management. While many of these frameworks are open source, this is by no means a requirement.

The use of web application frameworks can often reduce the number of errors in a program, both by making the code simpler, and by allowing one team to concentrate on the framework while another focuses on a specified use case. Frameworks can also promote the use of best practices. The general description of Web applications and its supporting architecture could be found in the following chapter.

Advantages

- Web applications do not require any complex "roll out" procedure to deploy. A compatible web browser is all that is needed;
- They require no upgrade procedure since all new features are implemented on the server and automatically delivered to the users;
- Web applications could work together with other server-side solutions.
- They also provide cross-platform compatibility in most cases (i.e., Windows, Mac, Linux, etc.).
- With the advent of HTML5, programmers can create richly interactive environments natively within browsers. Included in the list of new features are native audio, video and animations, as well as improved error handling.

- Modern web applications support greater interactivity and greatly improved usability through technologies such as AJAX that efficiently exchange data between the browser and the server.
- Web applications allow for easier introduction of new user devices (e.g. smartphones, tablets) because they have built-in browsers and responsive themes could adopt the new layouts.

An enterprise level Web application involve groups including software engineers, database experts and Web designers as well. Modern solutions are strictly separates the responsibility between the teams but in the first times it was not so clear and different parts could be mixed in one file - like the early versions of ASP or JSP.

Prev Up Next
Web 3.0 - beyond the Semantic Web, a way to global SOA? Home Mashups