

THE RICH INTERNET APPLICATION

The Rich Internet Application are researched in the article, the main attention is paid to the characteristic and benefits of this technology.

The Web is already the platform for doing business efficiently and quickly. As the penetration of high-speed and broadband Internet access increases, Web technologies continue to evolve to deliver new user experiences and increased application utility. The Rich Internet Application (RIA) is another step in that evolutionary process. The Rich Internet Application reflects the gradual but inevitable transition of Web applications from the simple thin-client model of a traditional Web browser to a richer distributed-function model that behaves more like the desktop in a client/server model. Today these richer user experiences are being implemented with technologies such as Flash, Ajax, and Java, using standard Internet and Web protocols [1, p.3].

According to Preciado et al., current web, Multimedia and Hypermedia methodologies of web engineering community are incomplete and unable to respond to the new functionalities that users need [2, pp. 7-13]. Jeremy Allaire from Macromedia introduced the term "Rich Internet Application" (RIA) to overcome the drawbacks of both desktop and web applications. RIAs are proposed as a solution to answer the needs of the users. Normally, RIAs are loaded with some initial data by the client. Then, the RIA can manage data rendering, event processing and communication with the server only when it is required by the application. There are several directions and technologies to extend the interactivity in web applications by running part of the application in the client.

The technologies are classified into four categories:

- scripting-based in which the client side logic is implemented via scripting languages, such as JavaScript, and interfaces are based on a combination of HTML and Cascading Style Sheets (CSS). CSS is used to describe the style of a document written in a markup language, such as HTML;

- plugin-based, where advanced rendering and event processing are granted by browser's plugins interpreting XML, general-purpose programs or media files (Flash, Flex, Laszlo, Xamlon);

- browser-based, where rich interaction is natively supported by some browsers that interpret declarative interface definition languages. For example XML User Interface Language (XUL).

Web-based desktop technologies in which applications are downloaded from the web but executed outside the browser (Java Web Start, Window Smart Client) [3, pp. 353–360].

First and foremost, RIA is attractive because it utilizes a Web architecture. The most often cited reason for choosing a RIA architecture for an application is to avoid installation and application management issues. In most cases, a RIA architecture was chosen over a plain JavaScript/HTML approach, which might have moved the bulk of the application to the Web server. Using a RIA architecture allowed a high level of richness in content, analytics, or interaction. The following is a list of benefits cited by companies selecting a RIA approach to application development:

- increased application adoption. Engaging applications simply get more use. Companies rolling out new RIAs report much faster uptake of the new applications than their predecessors. It is not uncommon to see increases in customers and in linger rates with these applications, which often translates to higher revenues or transactions where applicable;

- reduced errors and workarounds. For process-oriented applications, a richer, more visually supportive application can pull users towards the preferred processes and reduce the level of workarounds and errors;

- better and faster decisions. Using highly visual analytics that incorporate real-time data with supporting content and/or animation enables facts, events, and trends to be understood in context, with greater depth and more quickly, thereby enhancing the quality of decisions made based on these facts or trends;

– reduced support costs. Many complex processes can become attainable through self-service via a well-designed RIA, thereby significantly reducing support costs or potentially eliminating parts of cost centers that supported these services. Where self-service is not utilized, simply the higher usability and visual interactivity of RIA can reduce support calls significantly;

– reduced training time. Good RIA design today means that an application rarely needs to feature a user manual or even help screens beyond hover bubbles and information or rich labeling and responsive validation/confirmation processes [4, p. 3].

With the increasing adoption and improvement in broadband technologies, fewer users experience poor performance caused by remote latency. Furthermore one of the critical reasons for using an RIA is that many developers are looking for a language to serve up desktop applications that is not only desktop OS neutral but also installation and system issue free.

RIA running in the ubiquitous web browser is a potential candidate even when used standalone or over a LAN, with the required webserver functionalities hosted locally [5].

List of References:

1. White Paper. Rich Internet Applications: Design, Measurement, and Management Challenges, Keynote Systems. - 2006. – 19 p.
2. Preciado J.C. Necessity of methodologies to model Rich Internet Applications / J.C. Preciado, M. Linaje, F. Sanchez, S.Comai. – Washington, 2005.
3. Bozzon A. Conceptual modeling and code generation for rich internet applications / A. Bozzon, S. Comai, P. Fraternali, G.T. Carughi – New York.
4. Perry R. Adobe Flash in the Enterprise: The Case for More Usable Software / R. Perry. – 2010. – 12 p.
5. Internet applications unit1 — Document Transcript [Electronic Data Interchange [Электронный ресурс] / slideshare. – Режим доступа: <http://www.slideshare.net/> – Загл. с экрана.