

Is HTML5 Right Technology for the Development of Business Software Applications?

Zoran Lovrekovic¹, Tomislav Lovrekovic²

¹Higher Education Technical School of Professional Studies, Skolska 1, Novi Sad, Serbia

²Novine Novosadske, Dunavska 24, Novi Sad, Serbia

Abstract – This paper discusses whether today very popular HTML5 meets all the requirements which are placed for technology for business software applications development. Other technologies for development of the Rich Internet Applications, such as Apache Flex (formerly Adobe Flex) and Microsoft Silverlight are compared with HTML5 in terms of business software applications development.

Keywords – Business software applications, HTML5, Apache Flex, Microsoft Silverlight.

1. Introduction

Most well developed economies abandon traditional industrial way of doing business. New ways of doing business lead the world into a new, Knowledge age. In a time of rapid change, the past is less and less connected with the future. It is becoming increasingly difficult to guess or predict what life will look like tomorrow, which of the many new technologies will become indispensable, and which new products will emerge and remain competitive. Additionally there is more and more new and diverse knowledge that can be immediately put into practice. As a result, it is becoming increasingly difficult to determine which knowledge is “right”. How does one get by with the vastness of the new knowledge available? When the pace of life was slow, it was easy to manage a relatively small amount of knowledge. Today, we can all witness that the pace of life has become hectic and everything is changing rapidly [1]:

- Products are not single-purpose any more, but multi-purpose or even universal;
- More and more knowledge is invested in such products;
- More and more knowledge is required in order to be able to use these products.
- Business globalization rate is increasing more and more;

- The competition level is increasing;
- Technology is constantly changing;
- Workforce is more and more diverse;
- Suggested innovations are realized faster;
- The complexity of managerial surrounding is increasing faster;
- Decision making in business must be very fast
- Workplaces become mobile
- All are networked with everyone

How to have a quick glimpse of the possibilities and the current status of a supplier? How can a supplier see our production plans at a glance, in order to timely prepare for our demands for raw materials and thus gain business confidence from a known customer? How can a commercial office quickly check the inventory at remote facilities of the retail chain? How to determine which model is going where, who’s working better, and who is not working hard? How to coordinate activities with the cooperation and business partners in a better way [1]?

It is obvious that business software applications, under such circumstances must be developed as a web applications. However, Web applications, in addition to all the good qualities and the ability to consolidate operations into a single business process, to connect isolated “islands” in one unit, unify business processes of all participants in the venture, have some disadvantages. One of the most obvious problems is a “poor” graphical user interface.[2] If you wish that users like business software applications, they must have a user interface as a desktop applications, so they must be RIA, and must be suitable for using them on mobile platform (smart phone and tablets).

Also, business software application must be able to work with databases as integral part of application.

HTML5 is nowadays seen as the ideal technology for development such applications, but is it really so?

2. Criteria for evaluation of technologies for the development of business software applications

In accordance what we said in the introduction, it is obviously that main criteria for evaluation of technologies for the development of business software applications are:

- Ability to build complex Graphics User Interface,
- Ability for data visualisation, and find a proper way to cope with Context vs. Content,
- Stream instead Page manipulation,
- Client-Server communication based on Events instead on Requests,
- Ability to decide wheter application logic will be on server or on to client side,
- Ability to build highly interactive applications,
- Ability to build application that can be work on-line and off-line,
- Cross-platform interoperability,
- The existence of an integrated development environment (visual programing, testing, debugging, optimizing, ...), and
- Robustness and independence of the type and version of web browser

HTML5 is a markup language used for structuring and presenting content for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML 4 as of 1997) and, as of December 2012, is a candidate recommendation of the World Wide Web Consortium (W3C) [3].

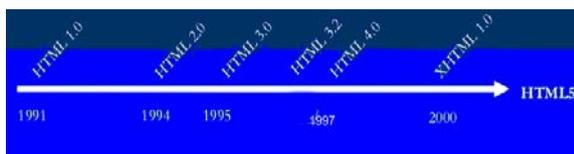


Figure 1. Evolution of HTML

HTML5 extends, improves and rationalises the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a potential candidate for cross-platform mobile applications. Many features of HTML5 have been built with the consideration of being able to run on low-powered devices such as smartphones and tablets.

HTML5 do not provide us directly with UI widgets, except adding datagrid and menu tags:

```
<datagrid></datagrid>
<menu></menu>
```

There are no implementation of these tags yet, so we have no idea what they might look like.

HTML5 adds many new input types, and offer new way of validate user input without much effort. For example, you can check the validity of an e-mail address (which would otherwise be a huge Regular Expression). You can also query a boolean to see if a form is valid, and you can use pseudo classes for styling invalid form fields. [3]



```
input type="list", ="range",
="date"
```

Figure2. Examples of HTML5 form input tags



```
input type="email", ="url"
checkValidity
```

Figure3. Example of validation of user input

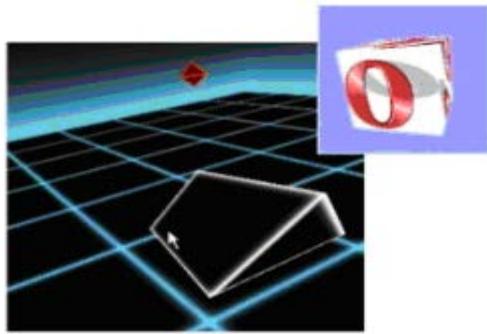
With new tag `<canvas></canvas>`, HTML5 allows for images and vector-graphics to be directly rendered with JavaScript, which allows simple animations.



```
<canvas></canvas>
```

Figure 4. Example of using `<canvas>` tag

The HTML5 specifications suggests that there may be a 3d context for the canvas object, and few browsers are experimenting with what that might look like. 3D graphics means creating scenes, setting up cameras/lighting, and doing 3d transformations [3].



`canvas.getContext('3d')`

Figure 5. 3D possibilities in HTML5

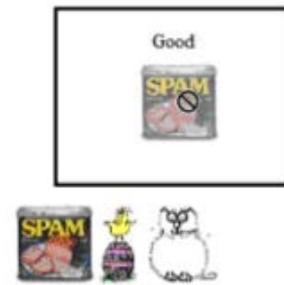
Once you have graphics, you want to manipulate them. Canvas only gives you raw pixel access, which means that bitmap manipulation is possible, but in a hard way.

HTML5 finally introduces native browser support for audios and videos:

```
<audiocontrols src="">
</audio>
and
<videosrc="">
</video>
```

Regarding communications between client and server side, in HTML5 you can still use XMLHttpRequest in asynchronous communication, or XMLHttpRequest, which is an API available to web browser scripting languages such as JavaScript introduced in earlier versions HTML. But HTML5 offer us a WebSocket, which is a protocol providing full-duplex communications channels over a single TCP connection [5]. The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011, and the WebSocket API in Web IDL is being standardized by the W3C. The WebSocket protocol makes possible more interaction between a browser and a web site, facilitating live content and the creation of real-time games. This is made possible by providing a standardized way for the server to send content to the browser without being solicited by the client, and allowing for messages to be passed back and forth while keeping the connection open. In this way a two-way (bi-directional) ongoing conversation can take place between a browser and the server.

HTML5 makes possible to dragging and dropping inside application, but also dragging, dropping, copying and pasting from the desktop and accrossdiferent applications.



`DataTransfer,`
`DragEvent`

Figure 6. Example of drag and dropp in HTML5

Offline access means that application must cache its resources and data, and know weather is online or offline. HTML5 applications have the ability to do all of this, but there is no implemetations of it yet.

The evolution of HTML has come as a result of the competing forces of innovation and standardisation. Whilst innovation and competition drive the technology forward, adding new capabilities to HTML and its associated technologies of JavaScript and CSS, standardisation tries to ensure that the web developer can expect the same features and APIs accros various browsers. These two forces need to be delicately balanced. HTML5 emerged at a time when this balance was tipped slightly too far towards standardisation.

HTML5 is a collection of various new APIs and features for web applications developers, tackling the gap that Flex and Silverlight fill. Despite the HTML5 specification being far from complete, modern browsers are already adopting various parts of the specification, such is the evolutionary nature of HTML. However, older browsers must be upgraded to a newer version to add feature support, in contrast to the plugin model, where only the plugin runtime needs to be upgraded.

In order to use the new features, developers need to be aware of the level of support for the particular feature they wish to use. In general terms modern browsers like Chrome and Firefox have support for many of these new features, whereas Internet Explorer does not have a good track record for feature support. Although with Microsoft starting to embrace HTML5, this looks to change with never versions of IE. In order to determine how well-supported a specific HTML5 festure is, you need to first look at browser support. The support for vaiuous HTML5 features accros browser versions is given at the Table 1 [4].

	IE	Firefox	Safari	Chrome	Opera	iPhone	Android	Reach
Canvas	6.0 ³	3.0	3.0	3.0	10.0	1.0	1.0	97%
local storage	8.0	3.5	4.0	4.0	10.5	2.0	2.0	81%
Video H.264			3.0	5.0		3.0	2.0	20%
Geolocation		3.5	5.0	5.0		3.0	2.0	49%
Form features		3.7	4.0	4.0	10.0			23%
WebWorkers		3.5	4.0	3.0	10.6			50%
WebSockets		4.0 (beta)	5.0.2	4.0	11.0			24%

Table 1. The support for various HTML5 features across browser versions

As you can see, HTML5 features support varies considerably across browser. There are some features such as canvas which are quite widely supported and are being actively used today. However, it should be noted that for IE8 or below, you need to install a plugin to support canvas. Other features, such as WebSockets, for example, are in a very early stages of adoption.

There is no IDE for HTML5 development. That means that there is no possibility for visual programming, and testing, debugging, and optimizing application in a formal, unique way. Productivity, for that reason, may be poor, and quality of application significantly depends of level of skills of developers. The lack of such a tools for support developing applications, means that HTML5 developers need to be more skilled to develop quality web applications. Ultimately this pushes up the cost of development for HTML application, compared to an equivalent created with Flex or Silverlight.

Apache Flex, formerly **Adobe Flex**, is a software development kit (SDK) for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform. Initially developed by Macromedia and then acquired by Adobe Systems, Flex was donated by Adobe to the Apache Software Foundation in 2011 and promoted by Apache to a top-level project in December 2012. Flex applications can be developed using standard IDEs, for example Eclipse, the free and open source IDE FlashDevelop, as well as the proprietary Adobe Flash Builder. It uses MXML language for defining user interface, and ActionScript language for implementing client side business logic, and communicating with server side language.

If we want to categorize Flex applications, we can say that Flex applications follow SOA architecture model, where Flex is used to create the client and then connect the client to data using services.

Let's go back to browsers and how web applications are delivered. When the browser makes a request, the server uses a combination of static content (HTML/CSS/JS code) and scripts (these scripts may query a database or call other scripts, but in the end they output HTML/CSS/JS) to prepare a page. This page gets loaded and rendered by the browser. A key element here is that, usually this page (or response) has the presentation markup and the data baked into the same message.

When a new state of the application is to be presented, the browser makes a new request and the server prepares the page. The client "just" renders the data.

Flex applications work differently. The server sends the compiled Flex application (the SWF file) that runs inside the browser using the Flash Player plug-in. Usually, this SWF file holds only the client-side business logic. If data are needed (from a database for example) the Flex application makes a request for those data. The server sends only the data (this can be in XML, JSON, AMF3 format), and the client knows how to represent this data visually. What we have here is a service oriented architecture: The Flex application is the client—a client that can consume data services from the server. The application can change state without refreshing the page or reloading the SWF file in the browser. The application is a client that can do more than "just" render data. Thus using Flex and Flash Player it is possible to create almost anything that makes sense to deploy on the web, from games to applications, to widgets that are integrated within "classic" web applications, and more.

The Flash Platform consists of multiple deployment run-times, development tools, and frameworks that are integrated across the full Adobe Creative Suite. Here is a list of some Flash Platform elements:

- The Flash Player: Flash Player is a browser plug-in which allows us to deploy web based applications to Windows, Mac, Linux, Android, and Blackberry.
- Adobe AIR: Adobe AIR is a run-time that allows us to deploy native applications to Windows, Mac, Android, iOS, and Blackberry.
- Flash Professional: Flash Professional is a tool for developing timeline based animations.
- Flash Builder: Flash Builder is an IDE to help programmer's write advanced code. **Flash Builder** (previously known as Adobe Flex Builder) is an integrated development environment (IDE) built on the Eclipse platform that speeds development of rich Internet applications (RIAs) and cross-platform desktop applications, particularly for the Adobe Flash platform. Flash Builder 4 is

available in three editions: Standard, Premium and Educational. The package is available free of charge for non-commercial use by students and unemployed developers. Flash Builder offers built-in code editors for MXML and ActionScript and a WYSIWYG editor for modifying MXML applications. Flash Builder includes an interactive debugger, allowing developers to step through code execution while inspecting variables and watching expressions. Flex Builder also offer support for performance analysis. The profiling view displays statistical information about memory use in addition to function call execution time.

- Apache Flex: Flex is the Software Development Kit that helps programmers build, debug, and deploy Enterprise applications with the Flash Platform. Flex includes a UI Component library, a SWF compiler, a command line debugger, an application profiler.

Flex prides itself on providing many diferent UI widgets (containers, lists, menus, navigators, etc.), as you can see at the Fig.7.

For a forms, Flex adds a couple nifty controls and offer validating user input without much effort. The data that a user enters in a user interface might or might not be appropriate to the application. In Flex, you use a *validator* to ensure the values in the fields of an object meet certain criteria. For example, you can use a validator to ensure that a user enters a valid phone number value, to ensure that a String value is longer than a set minimum length, or ensure that a ZIP code field contains the correct number of digits.

In typical client-server environments, data validation occurs on the server after data is submitted to it from the client. One advantage of using Flex validators is that they execute on the client, which lets you validate input data before transmitting it to the server. By using Flex validators, you eliminate the need to transmit data to and receive error messages back from the server, which improves the overall responsiveness of your application. Flex validators do not eliminate the need to perform data validation on the server, but provide a mechanism for improving performance by performing some data validation on the client.

Flex includes a set of validators for common types of user input data, including the following:

- Validating credit card numbers
- Validating currency
- Validating dates
- Validating e-mail addresses
- Validating numbers
- Validating phone numbers

- Validating using regular expressions
- Validating social security numbers
- Validating strings
- Validating ZIP codes

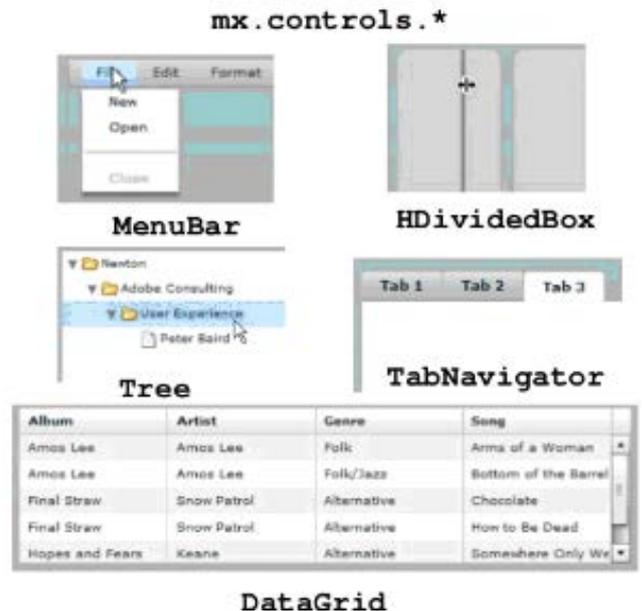


Figure 7. Examples of using Flex UI widgets

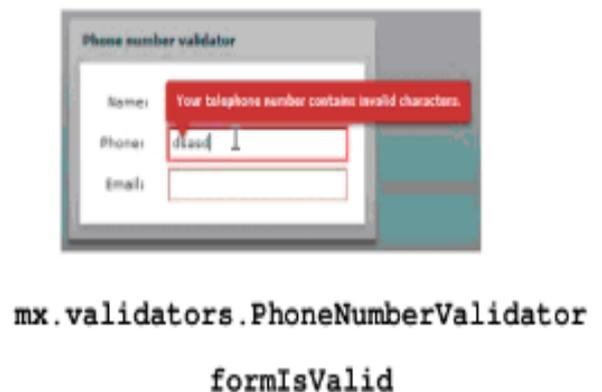


Figure 8. Example of using Flex Validators

Vector graphics means drawing lines and shapes of different styles. Flash (and, of course Flex also), has always had graphics as its focus and be amazing at this.

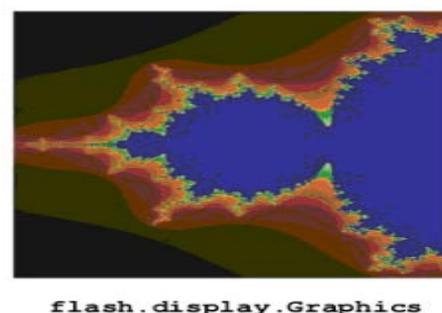


Figure 9. Example of using Flash Graphics class

Flash only recently offered actual native 3d support in Flash Player 10. Figure 10 Example of usage 3d Graphics in Flex

Regards Bitmap Manipulation, Flex lets you change the individual pixels, or easily apply different filters, effects, and transforms.

Flash always been the obvious choice for displaying videos on the web -You tube is the big example of that-and Flex makes it even easier with more controls.

It is so easy for Flex to allow dragging and dropping inside application, but also dragging, dropping, copying and pasting from the desktop and across different applications.



Figure 10. Example of drag and drop ability in Flex applications

What about client-server communications? In Flex applications you rely on server-side scripts to manage the databases. Flex offers you a way to call the server pages and get back the answer in Flex. Basically, you can connect a Flex client to a PHP back-end in two ways:

- over HTTP or,
- using sockets.

Over HTTP, there are four different ways to connect to a server data source: REST style services, web services (WSDL/SOAP), remoting (or RPC), and XML-RPC [6].

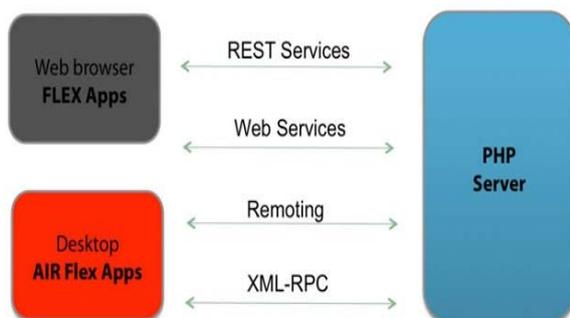


Figure 11. Flex – PHP HTTP based communication

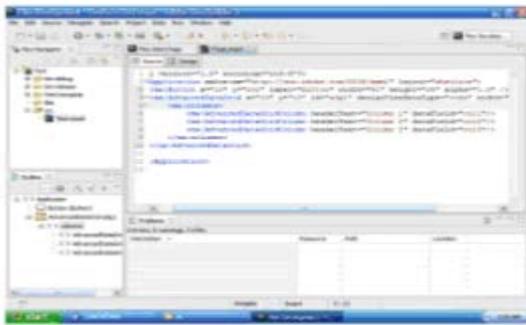
Using remoting you can leverage any PHP class you have on your server by calling any public method. Basically, from the Flex side you use an instance of RemoteObject as if it were the remote PHP class. Second, you can map the data model from the PHP side to an ActionScript data model, and have the conversion done automatically. This is extremely important, because when you use typed objects, you get the benefits of compile-time error checks and code completion. This means code that is easier to read and less prone to bugs. And third, the messaging format for this method, AMF3 (Action Message Format) is a binary format, which can be much faster and smaller compared to SOAP/XML/JSON, especially for big sets of data. AMF3 is also faster because it encodes data.

Socket communication lets developers create near real-time communication by pushing information directly to the client. Sockets are a practical and efficient way to send data from a server to a client. In a traditional HTTP model, the client sends a request to the server, the server processes it, the client receives a response, and finally disconnects from the server. To do this, HTTP requires the client to send a significant amount of extra information, including how long to maintain the connection, security and permission information, and information that identifies the request. When sending many bursts of data in a short period of time, as you might do with stock quotes or a chat application, your application is going to have to reconnect multiple times and perform several back and forth exchanges with the server, sending all that extra information each time to get the data. Each transaction can take twice as long since the client is sending a request and waiting for the response to reach it. That can be problematic for applications that need instant or near real-time feedback. Fortunately, there is an easy solution to this problem: sockets. Unlike with HTTP, when you use sockets, the application makes a single request to the server; the server opens a connection and maintains it so that it can push data out to the client whenever it wants. Both the client and server *bind* to a socket so they can listen for any changes in information, and both client and server can send information over that socket. That means the client will receive any data sent by the server over a socket without having to make a request. As a result you can cut the time it takes to receive data in half and your application can send and receive that data in near real-time [7].

Establishing Socket communication in Flex is easy and effectively.

It's always easier to build application with an IDE. For Flex, there is a Flash Builder built off the Eclipse platform.

FlashBuilder allow visual programming, testing, debugging, and optimizing application. When debugging Flex application, the key element to keep



FlexBuilder

Figure 12. FlashBuilder (formerly FlexBuilder)

in mind is: now, you have a client that is separated from the server, and problems may be on the client, on the server, or at the network level. Two new views are added to Flash Builder 4 that help you to debug the code: Network Monitor and Test Operation views.

Using Network Monitor view you can see all the communication that goes between the Flex client and the server. You can see both the raw data and the objects.

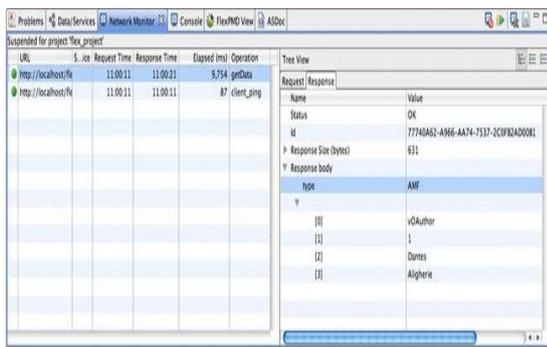


Figure 13. Using Network Monitor

Test Operation view, as the name implies, let's you to test the available operations.

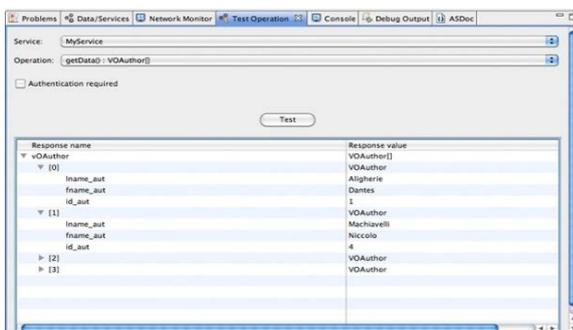


Figure 14. Using Test Operation view

Microsoft Silverlight is an application framework for writing and running rich Internet applications, with features and purposes similar to those of Apache Flash. The run-time environment for Silverlight is available as a plug-in for web browsers running under Microsoft Windows and Mac OS X. While early versions of Silverlight focused on streaming media, current versions support multimedia, graphics, and animation, and give developers support for CLI languages and development tools. Silverlight is also one of the two application development platforms for Windows Phone, but web pages which use Silverlight cannot run on the Windows Phone or Windows Mobile versions of Internet Explorer, as there is no Silverlight plugin for Internet Explorer on those platforms. Flex have much more components for build graphical user interface than Silverlight, and much more UI widgets. Also, much more browser have installed Flash Player plug-in (estimated 95%) than Silverlight plugin (estimated 65%). Therefore, it is obvious that Flex is much better choice than Silverlight.

3. Conclusion

In order to be effective and efficient, today's business software applications must be develop as a RIA web applications.

They also must be able to provide the real-time communication of client and server side based on evants instead based on request (over socket). Created software applications must be highly interactive and can be work on-line and off-line. There must be a possibility to be decided which part of the business logic to be on the server, and which part on the client-side. Cross-platform interoperability, existence of an integrated development enviroment (IDE), and robustness and independences of the type and version of web browser are also very importants.

Although it has been shown that both HTML5 and Flex can accomplish most of proposed characteristic, there are however some key differences between these two analysed technologies:

- Flex applications are running under Flash Player browser plugin. This is the approach that was used to bring Java applications to the web, with Java Virtual Machine (JVM) plugin providing the runtime for Java Applets. Therefore, Flash Player provide a runtime for Flex native content. The biggest advantage of the plugin model is that it gives Flex developers a sandbox free from the browser itself. Therefore it is also free from constraints of web standards and the associated issues of

cross-browser standards support, allowing plugins to provide that application feel and look is the same no matter in which browser it runs. When you use HTML5 to develop application, it (almost) never look same in a different browser, and even more, in a different version of the same browser. As we have seen, many HTML5 specifications are not browser supported yet.

- Flex has Integrated Development Environment. There is no IDE for HTML5 development. Of course, you can build the henhouse simply by hand mix mortar on the ground, and then just line up mortar and bricks. But if you build a house, you need a tools such as shovel, wheelbarrow, trowel, hood, cement-mixer, square, plumb line, and so on. If you build a building, you need more tools and construction machinery. Therefore, if you develop some small web site , HTML5 is quite OK. But if you develop some complex business applications, you'll need a tools (Integrated Development Environment).
- And again, many of HTML5 specifications are not browser supported yet. Many of HTML5 great specifications are more list of nice wishes today, than reality. Flex is fully matured technology which offers many advantages in developing software applications .

HTML5 has great potential and a promising future, but if you need technology that you will develop business software applications now, then Apache Flex is smart choice. And one more thing: despite Google's official announcement that the android version 4.0

and later will not support Flash Player, we all see that this is not true.

References

- [1]. Lovrekovic, Z.(2013), Why Knowledge Management, *Online Journal of Applied Knowledge Management*, ISSN 2325-4688.
- [2]. Lovrekovic, Z., Sukic, C., & Sabanovic, M. (2010). Desktop, web and web2.0 application comparing on the aspect of the realisation possibility the complex user interface. *Technics Technologies Education Management*, 5(4), 678-687.
- [3]. Fox, P, *Flex vs. HTML5 for RIAs*, Ignite Sydney, (2009). <http://www.slideshare.net/wuzziwug/flex-vs-html5-for-rias-presentation>
- [4]. Eberhardt, C, *Flex, Silverlight or HTML5?*, http://techylib.com/en/view/victorious/flex_silverlight_or_html5_time_to_decide
- [5]. <http://www.websocket.org/aboutwebsocket.html>
- [6]. http://www.adobe.com/devnet/flex/articles/flex_php_architecture.html
- [7]. Stewart, R, *Communicating with Flex and PHP overSockets*, <http://ria.dzone.com/users/ryanstewart>

Corresponding author: Lovrekovic Zoran

Institution: Higher Education Technical School of Professional Studies Novi Sad, Serbia

E-mail: lovrekovic@vtsns.edu.rs