Methods of Creating Graphical Interfaces of Web Applications based on the Example of FLEX Framework

Adam Dębiński, Bartosz Sakowicz, Marek Kamiński

Abstract - The article, by describing features of sample application for project management, presents methods of creating graphical interfaces of modern applications that run in web browser environment. Sample program is written with the use of FLEX Framework and some FLEX supporting tools.

Keywords – FLEX Framework, RIA, GUI.

I. INTRODUCTION

World Wide Web since its beginning in 1990 has come a long way marked with constant evolution. Since 1993, when the first widely known, graphical web browser – Mosaic – was created [3], web graphics and web GUIs have been the subject of the same evolution [6]. From the simplest, low-resolution 2D images, through Flash based vector graphics, to advanced 3D components and transformations – resources that web designers have at their disposal in conjunction with high-speed Internet connections and high-power computers of the final users gives that slowly the boundaries of imagination becomes the only limitation.

II. RICH INTERNET APPLICATIONS (RIAS)

RIAs join best functionalities of desktop applications and the Internet. (Fig. 1)

The first ones offer rich and intuitive graphical interfaces with great capabilities, which make using them easier and prevent unwanted frustrations. Internet, from the other hand, provides perfect platform for created applications. Thanks to its global nature it allows everyone with active Internet connection to access available RIAs.

By joining all of before mentioned advantages, RIAs appear to be fascinating and impressive, new medium, which by providing exciting and interactive, graphical interfaces are perfectly suited for creating B2B1 and B2C2 applications and services [7].

III. FLEX FRAMEWORK

Flex was published by Macromedia in March 2004 as an expensive server product designed for creation and deployment of applications. Since version 2.0 however, its nature changed drastically, making Flex a client-side technology which, additionally, was much more affordable. Introducing version 3.0 Adobe3 made Flex SDK4 open-source and free, charging only for the charts library and development environment – Flex Builder.

Flex Framework is the combination of at least a few technologies. It is not a standalone application but rather it consists of following elements [2]:

- **Languages**: MXML and ActionScript 3 (for creating Flex applications a combination of two languages is used. MXML is a markup language similar to HTML, which main purpose is to describe layout of the application. ActionScript 3 is an object, scripting language used to implement business logic and interaction events of the components of the application).

- **Components library**: Flex SDK (it’s the set of graphical user interface components like buttons, lists or containers used for building Flex applications. Beside charts library, all SDK is free and open-source).

- **Integrated Development Environment**: Flex Builder (it’s the environment created on the base of Eclipse IDE used for code editing, application

1 B2B – business to business
2 B2C – business to consumer
3 In the meantime Macromedia was bought by creators of Acrobat Reader
4 Software Development Kit
compiling, debugging and efficiency testing. Unfortunately it’s not a cost free product. Adobe sells it in two versions: costing $249 Flex Builder Standard and costing $699 Flex Builder Pro, which offers additionally tools for memory and efficiency management and charts library [5]).

- **Runtime environment for multiple web browsers:** Flash Player (created applications are run in the web browser environment with the help of Flash Player plug-in. There’s also a possibility of running Flex applications in the standalone mode using AIR5 Environment, provided free of charge by Adobe).

IV. "PROJECT DESIGNER” APPLICATION DESCRIPTION

Project Designer is an application created with Flex Framework, which main purpose is to improve projects planning, creating and maintaining processes [8]. Program tries to achieve this goal by moving most of the functionality that person responsible for project management could possibly need – to graphical environment. Almost every single action can be done with a few mouse clicks and moves. At the same time Project Designer is the tool advanced enough that it could successfully find its place in companies, considering project management an ordeal, because of the complexity of solutions included in competitive applications.

Project Designer consists of three key views responsible respectively for project management, project’s stage management and resource management. Switching between them takes place in strictly defined points of application, however they’re intuitive and don’t present any orientation difficulties for the user. What’s more, most of the functionalities of each view is visible directly after loading the view and those that are hidden, can be found in easily predictable places, what further improve user experience.

V. PROJECT MANAGEMENT VIEW

Fig. 2 presents application right after first launch. It can be easily seen that it consists of three main parts: starting from the top there is a time ribbon, workspace and a simple toolbar. Time ribbon consists of three sets of buttons positioned in three rows. Every row represents successive date field starting from years (top row), through months (middle row), to days (bottom row). After application launches, only years row is visible, which consists of three buttons representing current year and two following ones. When user presses any year, then button representing it takes all available width of the years row and months row shows up. Pressing any of the month buttons causes resizing of pressed button and showing up of the days row. Pressing any button above the lowest row causes all the rows bellow pressed button to hide and the row with button to return to initial state.

![Fig. 2: Project management view](image)

Algorithm of creation of successive rows works as follows. Every row and every button placed in it is described by time range attribute (start and end date). Algorithm launches every time this attribute is changed for a row. First step is to create an array of labels that will be displayed on buttons. The mechanism of ActionScript language is used here that allows access to field of any class by providing the name of the field in the form of String literal (Fig. 3).

```javascript
var rok = dateRange.startDate.fullYear;
var rok = dateRange.startDate."fullYear";
```

Fig. 3 Example of accessing field of a class by providing name of the field as a String literal [4]

Thanks to this mechanism it was possible to write one all-purpose method, getting field name from the nest level of the time ribbon. Labels are obtained by iterating through all values from start date to end date for appropriate date field and changing numeric values to text in the case of months. Next, buttons are created in the amount equal to the amount of labels. Width of each of them is measured as row width divided by number of buttons. In the next step, for every created button, based on the date field that it’s representing, date range field is set.

Workspace, located in the middle of the application is the main area of user’s work. It’s here where project management is mainly happening. Workspace mirrors time ranges defined by buttons in time ribbon, what is stressed by vertical lines, positioned to mirror positions of spaces between buttons. Every change of the time ribbon structure is immediately reflected in generating of the workspace and every element that it contains. Blue vertical line represents current date.

Last element of the described view is toolbar. It consist of components that help decreasing chaos that can appear when to much projects are managed at the same time. First of them is slider for changing projects’ height. Typically they are displayed with both name and time range visible. However, when user decides that this view is unreadable for him, he can switch size to one of the two smaller ones.

---

5 Adobe Integrated Runtime
Next functionality allows filtering of visible projects by their current status, if user doesn’t want to see projects that are ex. finished or still planned.

„Left“ and „Right“ buttons are used to scroll time ribbon one unit of actually active row to the left or right allowing more smooth work with application and improving user experience.

Fig. 4. Time ribbon scrolled 8 days right from December 2009

Every graphical component that represents project consists of three main elements: icon, reflecting current status of the project, button, showing every change in the time range of the project and a label with project name. (Fig. 5).

Fig. 5. Sample project component

To create a new project, all that is necessary is to press LMB\(^6\) at some place in the workspace and move the mouse with LMB pressed until desired length of a project is reached. Time range changes are visible on the button in real time so acquiring desired time boundaries of the project is quite easy.

Editing of the project is equally trivial. User can change both start and end dates at the same time by dragging project component around the workspace or he can change only one date at a time by resizing component with handles located at the both sides of it. Every change in the „x“ coordinate of the left or the right side of the component results in launching an algorithm estimating current dates.

The algorithm starts with searching for reference button. If the result of dividing ribbon row width by number of buttons in it was an integer the reference button is last button in the row. If it wasn’t, the reference button is last button in the row that has width bigger than the rest of buttons. It guarantee that all buttons to the left and to the right of the reference one are of the same width. Next, „x“ coordinate of the right side of the button is set as a reference point. After that, algorithm measures the distance between the reference point and the point, for which a date is being searched. The distance is divided by appropriate width of the button and the result represents the offset against the value of the date field visible on the reference button. Rest from this division is additional offset (in days) that allows to precisely estimate quested date.

Dates can be also edited more traditionally by pressing the button on the project component. This way a popup window with two calendars will be opened allowing to freely choose desired dates. The only restriction is that the start date must come before the end date.

Beside dates, user is allowed to modify other attributes as well. After double-clicking on the project component a pop-up with attributes editing tools can be accessed. It allows to modify project component color and transparency as well as project name and status. (Fig. 6)

If user wants to start editing stages of the project, he can do it anytime by double-clicking on a project and choosing “Edit stages” from a menu.

VI. STAGE MANAGEMENT VIEW

Fig. 7 presents stage management view right after switching to it. The only difference to the previous view is additional functionality implemented by the ribbon – time boundaries. With projects, user wasn’t in any way limited in how long his project was or when it started. Stages from the other hand are not so liberal. Time ranges of the ribbon are set to those of currently edited project. Fig. 7 presents this kind of situation. Time ribbon is shorter because edited project time range is set to 15.05.2009 – 28.10.2009.

Stage component differs quite much from the project one. The biggest difference is introducing of component division. (Fig. 8)

---

\(^6\) Left Mouse Button
That way much greater flexibility was achieved. Each part has its resources defined separately what allows better time, workspace and resource management.

Most of the functionality related to component editing is implemented in the same way as in the project management view (moving the mouse while holding LMB for creating, dragging and resizing the component for editing), however in the stage view, user additionally gains the ability to change the date separating both parts of the component. Attributes are set with the help of appropriate pop-up as well. Stage component is also “equipped” with two triangles. Their function is to hold handles of the connections used for mirroring real-life dependencies between the stages and representing workflow of the project.

Creation of the connection is accomplished by placing mouse pointer above one of the triangles, moving the pointer above another triangle with LMB pressed and releasing LMB.

Stage management view also allows user to create additional split/join components. Using them gives the ability to control simultaneous starting and ending of multiple stages.

All before mentioned functionalities can be seen at Fig. 9. By double clicking the stage and choosing „Edit resources” option access to the resource management view can be gained.

VII. RESOURCE MANAGEMENT VIEW

Resource management view is really a pop-up that allows few basic resource management actions. It consists of two graphical lists. Right one contains all resources available for currently logged on user, the left one – resources applied to edited stage. Fig. 10 shows the exact moment of moving the resource from one list to another. Icons that can be seen to the left of the resource item represent type of the resource (human, device). Little globe is the indication of global resource that is available for both parts of the stage.

VIII. CONCLUSIONS

Modern IT projects head for moving most of the “behind the scenes” calculations to final user’s devices. By doing that costs of maintenance and support of such applications drop drastically. However to achieve this kind of functionality web applications have to mirror desktop programs as well as it is possible, including presentation layer. Adobe with its Flex Framework, as well as the rest of RIA frameworks making competition, by providing advanced, mature tools for creating such applications, show that they’re the ones that’ll be shaping the future of the IT.

REFERENCES