Developing Dynamic Reports using Rich Internet Applications

Luís Matias

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto – Portugal

luis.matias@fe.up.pt

Abstract. In the business world, there are several software tools to generate reports automatically. We can do it with software like Crystal Reports and we can use different options to configure the reports. However, we observed that it is not flexible in several points like the structure of the data base or the changing of report's parameters after its generation. To solve this problem, we propose the using of RIA – Rich Internet Applications in order to build dynamic reports. These dynamic reports will allow us to change the report's contents and parameters after its first generation. With this, it will be possible to ask for different data in runtime, with no need for generating new reports. To implement this solution using RIAs, we tested and compared two different technologies: Microsoft Silverlight and Adobe Flash. We expected that the Silverlight should be more reliable than Flash in this context because it was integrated in the .NET framework. After our experiments, we concluded that the dynamic reports can be generated using RIAs. For that solution, we stated that Silverlight was better than Flash because it's easier to use and to develop with.

Keywords: Dynamic, Reports, Crystal, RIA, Flash, Silverlight.

1 Introduction

In the business world, over the last years, we witnessed the creation of several tools destined to support the automatic generation of reports from different sources and data types. These reports usually have rigid models that are independent from its data source if it has a previously defined structure.

One of the more capable and reliable software tool to do this type of interaction is Crystal Reports [1]. However, this type of applications has, due to its nature, high level restrictions to the information retrieval process based in the generated reports. It is possible to resume big data set in small lines, like annual invoices. In the opposite way, it is not allowed to do the inverse and see what lines fulfill that sum or change, in a runtime environment, the timeline that we want to report and analyze.

To fulfill this type of requests, it was proposed to develop prototypes of a new type of reports: dynamic reports. The dynamic reports are reports that are automatically generated and that allow analysis and changing with no need to do a new generation of it. We will be able to do it in runtime. To solve this problem, we used RIA's – Rich

Internet Applications. In this context, we tested and compared the two dominant technologies that implement RIA's nowadays: the Microsoft Silverlight and the Adobe Flash [10], in a way to understand which one of these technologies would be the best solution to our problem. We hope to conclude that it is possible to develop applications to support automatic generation of dynamic reports with the two chosen technologies, especially with Microsoft Silverlight.

This paper is structured as follows: Section no.2, titled "Automatic Report's Generation", states a general perspective about this problem. Section 3 has the title "Dynamic Reports with RIA's" and it states, in a general way, the RIA's basic characteristics and the main requirements to build a prototype created with this type of technology, based in Management Control. The Management Control basic functions and characteristics are also mentioned. In the section 4, the results obtained by the developing of the two prototypes are lightly described, considering the requirements previously defined. Section 5, titled "Comparative Study between RIA's technologies", exhaustively describes the results obtained through the analysis of the developing and the using of the prototypes. The sixth section of this paper presents a small discussion about the results obtained in the previous section. Finally, the section 7, "Conclusions", presents the main conclusions of the work.

2 Automatic Report's Generation

The automatic generated reports are commonly used since the early 90s, with the reports associated to the starters' software-based database management system for non-professional use: the Office and the Lotus Suite [2, 3]. However, it only became a standard feature when the Crystal Reports was introduced in the market. The Crystal Reports is a Business Intelligence application. This type of applications is commonly used in the collection, integration, analysis and presentation of information to support the decision. Originally developed for Business Objects, this application is used to develop and generate reports with several types of data sources. A large number of applications use OEM (Original Equipment Manufacturer) versions of Crystal Reports as its own tool of reporting generation.

The reports generated using Crystal Reports allow multi-applications and multi-database. They can also be used in the conception of those applications. However, the flexibility ends here: the Crystal Reports needs the data structure to be exactly the same as the first data base used to generate it. Despite the fact that we can configure the reports with different data sources, the report can never be changed after its generation. If you consider that the time needed to generate is as big as the data source, this must be considered as a serious problem to solve when we want to, for instance, compare reports related to different timelines.

3 Dynamic Reports with RIA

The dynamic reports are reports that can be changed after its generation, like a dynamic query with some parameters (i.e. the beginning and the end date of a period).

These reports are generated from a data set and they can be manipulated after its generation to demonstrate different perspectives. They must be portable and easily accessed, like an web portal, to be used in different situations. These characteristics present one clear requirement: we need to implement a web application to implement a dynamic report as it is described above.

However, it is not the best solution to use a traditional web application. As it is stated by Christodoulou Stygaras in [14], these applications present several problems with the processes, the configuration and the data. This last one is particularly important, because they do not support interactive explorations of the data, compelling the user to navigate the hypertext to see the desired data [15].

The traditional web applications were extended in several directions to improve interactivity and ease of use. The RIA followed one of these directions. It is the mix between the interactive and the multimedia user interface functionality of the Desktop applications and between the portability offered by the Web tools, allowing to build Rich applications with data and also with multimedia contents, like dynamic charts and media clips. Next, it is explained how the RIA can be a solution to implement Dynamic Reports and how we will use it.

3.1 RIA as a solution

The problem that we have is RIA related. There are several technologies available to implement a RIA. In this case, we compared two different technologies that were able to develop the same type of applications to use in this context.

The RIAs are web applications that are usually executed under browser's plug-ins. We described, in some bullet points, the main characteristics of RIAs:

- They support the graphical render and the inclusion of media clips – video and audio;
- They are easily installed because they are executed in a plug-in that adapts itself to the technological reality it finds (browser, operative system, etc.). This characteristic allows it to maintain the user experience with every platform used to execute it. The maintenance of these applications is simple because their plug-ins are automatically updated;
- They appear as more secure applications because they are usually pre-compiled and they sometimes use the sandbox's concept to the platform they are executed over. This will limit their access to the client.
- They have a better performance because they shrink the existing latency, comparing to the traditional web applications that need to constantly connect themselves to the server to allow it to process the data.

In order to develop an application to automatically generate dynamic reports, we chose to develop a prototype focused on Management Control.

3.2 Management Control Requirements

The Management Control is the discipline that studies the impact of the Strategic Management on the organizations based in metrics that evaluate their performance [5]. Using tangible values to measure intangible characteristics of the organization, like its branding or its CRM (client relationship management), it is possible to evaluate the current and future impact of a strategy on the operational results. We can easily verify, for instance, if the organization profit has grown like the expected and if the budget defined to certain activities is enough to achieve the defined goals.

We found some methodologies in the market which implement a more efficient Management Control over the organizations. One of the most important is the Balanced ScoreCard [4, 13]. With this, it is simple to justify our choice - because it is highly suitable to our software market: the top managers need - usually as support to their decision processes – of large amounts of data.
An example of a map needed by the Management Control, in its financial perspective, is the balance sheet's temporary map [17].

3.3 Prototype Requirements

After the analysis of the RIAs characteristics and of the Management Control requirements, we defined the following list of requirements to the prototypes [7]:

- To develop one or more Management Control maps in a application with dynamic contents and dynamic configurations, in runtime environment using RIAs;
- To develop mechanisms to obtain, in runtime, different perspectives of the maps;
- Strong visual components and highly intuitive and appellative aspect.
- To allow the automatic generation of bar charts or other chart types;
- To allow the generation of several maps simultaneously using different configurations using the automatic generation of charts;

These requirements were used to develop two prototypes of applications to automatically generate dynamic reports to support the Management Control. One of the prototypes was developed using Microsoft Silverlight and other with the Adobe Flash. The results are presented in the following section.

4 Results of the Developed Prototypes

Two prototypes of the desired application were created: one using Silverlight, other using Flash, as it is stated in the following sections.

4.1 Flash Prototype

This prototype was easily developed, using the Adobe Software Development Kit for Flash. It is, similarly to the other prototype, connected to a database running locally in a Microsoft SQL Server using a web service. It was not too hard to develop some

animations or a user friendly interface, but it was really hard to find a way to generate charts in runtime. It is really hard to develop your own Flash controls and you can't use anything from the OS. So, the last two defined requirements were not accomplished.

4.2 Silverlight Prototype

Like the previous one, the prototype was easily developed. It is possible to change some parameters of the map, as shown in the Figure 1.

Figure 1. Screenshot of the Silverlight prototype

Using the Silverlight Toolkit, it is possible to include chart components that are easily fed using LINQ – Language Integrated Query [16]. You can see the buttons on the left to navigate through the lines used in the represented sums, you can change the budget used to feed the map or the dates used to generate it. You are able to create a new map and you can save it in a file using Silverlight Isolated Storage, as you will be able to see in the further sections. Both prototypes can be accessed by a browser.

4.3 Prototype Development Results

The prototypes and the maps above prove that there are many things to improve from the old Crystal Reports maps. These prototypes are more flexible, more portable and more user-friendly than the older ones. Like that, they fill the gap found in Crystal Reports transforming the Dynamic Reports into a reality. Now, we need to know which of the technologies is best to continue this work on Dynamic Reports.

5 Comparative Study between RIA's Technologies

In the previous sections, the basic characteristics of a RIA and the prototypes developed using two different RIA technologies were explained. Considering these characteristics and the context we want to use them, we defined the following metrics to measure and compare the RIA's technologies:

User Experience – this metric pretends to measure the capacities of the technologies to develop a good user experience like the interactivity's levels, the graphic render, the quality of the media clips and the global performance;

Security, Communication and Client's Access - this metric pretends to measure the security level of the technology and how can we access and change the data in the Client and to communicate with other applications;

Installation and Execution – these metric measures defines the prerequisites needed to install and execute an application of this kind and which are the requirements of the client's application;

Maintenance and Reliability – we measure the effort to maintain this applications working as the reliability proportioned by the technology;

Developing Easiness – it is defined if it is – or if it is not – easy to develop applications with this technology;

These metrics were used to measure and compare the RIA's technologies: Silverlight and Flash, like you can verify in the following lines. A ranking from zero to five stars (✿) was defined for each one of the metrics to support a tangible evaluation of the technologies.

5.1 Adobe Flash

Using the previous section and the RIA's market analysis done in [7], the following analysis to Adobe Flash was concluded:

User Experience ✿✿✿✿ The Adobe Flash allowed a good user experience because it has several possibilities to include different rendered animations and audio/video streaming. It has a large amount of possibilities to support the interaction and a high level of styling. The Flash assures the quality of the reproduction independently from the client's monitor resolution (vector-based).

Security, Communication and Client's Access ✿✿✿ In order to communicate, Flash didn't demonstrate any functionality to allow the connection to Database Management Systems. However, it can use MSMQ – Microsoft Message Queuing [18] to communicate with other local applications. Due to its communication limitations, Flash is really secured to execute a presentation of this type. Flash is in the market from many years till now and, with that, it developed a real trustful relationship with their users: programmers and consumers.

Installation and Execution ✿✿✿✿✿ Flash supports the majority of existing operative systems and browsers like Windows 9x, Linux, Opera, IE, Mozilla Firefox, MacOS, Safari, etc... It can be executed in a browser's plug-in or in an isolated player as a standalone application. The applications developed using Flash are short-sized and the plug-in startup is really fast.

Maintenance and Reliability ✿✿✿✿ Flash's reliability is high due to its long years of existence. The Flash's plug-in is easily maintained because it is automatically updated. The generation of standalone applications requires a new compilation of all components every time the presentation is changed.

Developing Easiness ✿✿ It is not simple to develop in Flash. It only supports one programming language: the Action Script. This language requires a high learning curve because it is not a standard programming language and it is not used in other applications besides those who are developed with Flash. Therefore, the source code can't be used again in other context. It is not possible to use any OS (operative system) controls and the animations are built using the frames that are shown. Flash only uses the matrical transformations to animate its presentations - we must somehow assure that the client will maintain the application's frame rate. If this doesn't happen, our animation can last for 1 or for 5 seconds (!), for instance. Flash's developing environment was designed merely to develop Flash applications and it requires a long time to learn. This developing environment is graphical and, due to that fact, it is more suitable to use for designers than for programmers.

5.2 Microsoft Silverlight

Now, using the previous section and the RIA's market analysis done in [7], the following analysis to Microsoft Silverlight was concluded:

User Experience ✿✿✿✿✿ The Silverlight has a set of capabilities similar to Flash. Rendered animations, streaming of audio and video, etc...

However, it's global performance it greater than Flash's. The compilation of its applications is made using CLR - Common Language Runtime [11]. Every language that uses this type of environment is managed code's languages: languages that are compiled first in a virtual machine, then, executed by the CPU.

On the other hand, Silverlight presents a technology that distinguishes itself from other technologies: the Deep Zoom. The Deep Zoom allows us to randomly zoom in large images with an enormous performance [10]. The Silverlight possesses one more unique characteristic: graphic acceleration supported by hardware, with the technology Direct3D.

Security, Communication and Client's Access ✿✿✿ Like Flash, Silverlight hasn't got any functionality to allow the direct connection to Database management systems. However, it is not allowed to access the client file system. The simplest way to turn over this problem is to use a web service.. As far as security is concerned, we can define Silverlight like a pretty secure platform. This fact occurs because Silverlight is executed in a sandbox. A sandbox typically restricts the access to the platform's native API, controlling the resources that the application can and cannot use like the disk space and the memory space, the access to system's information, the read of input devices, etc... To minimize those effects, the Silverlight uses the Isolated Storage [9]. One of the Silverlight's biggest limitations is its need to communicate asynchronously [8].

Installation and Execution ✿✿✿ The Silverlight is also executed in a browser's plug-in but it does not support many standard platforms like the Linux operative

system or the Opera browser. It does not support any execution outside of a browser like a standalone application. Like Flash, the plug-in startup is fast but their files have a bigger size. As we observed, they are, in average, 10 times bigger as they don't compress any of its source files. Beside the fact that it needs several source code files to its execution, it's dependencies are not built-in in the application.

Maintenance and Reliability ✿✿✿✿ Like Flash, the plug-in has an easy maintenance because it is automatically updated. Despite its short time in the market, Silverlight uses technologies largely used to build business oriented applications (.NET,C#,VB .NET, CLR, etc.). Consequently, the programmers state a high reliability to Silverlight. Silverlight has different source files and because of that the files that are responsible for the communications protocols, data queries, design, etc. are easily identified. With that, it's easy to create new applications and functionalities using tested and existing source code (C# code, for instance).

Developing Easiness ✿✿✿✿ Silverlight mainly uses technologies from .NET framework in its applications like CSharp, VB, WPF, WCF, ASP.NET and LINQ. These languages are commercially used to develop applications with all type of business plans. Therefore, its source code can be used over and over again.

The used IDE (Integrated Development Environment) is also well known – Visual Studio – and the most programmers have a great familiarity with its use. Silverlight has the Expression Blend as IDE too. This tool allows us to edit our applications design and it is pointed to the designers themselves. The creation of Silverlight's animations is, again, very easy: it is possible to define time-based animations. We can merely define the first and the last state and the render software generates the remaining states throughout the time. The Silverlight use the Windows Presentation Foundation framework as platform to use several Windows's controls without any integration efforts.

However, this technology has negative points too. The projects debugging is simple because it is possible to edit just a small component of the project without compiling it all. But we need to consider the slow startup of the application because it needs an ASP.NET server to be running, located locally or abroad.

6 Discussion

Using the described work, the prototypes and the analysis previously done, we present in Table 1 a summary of the whole study.

	Adobe Flash	MS Silverlight
User Experience	✿✿✿✿	✿✿✿✿✿
Security, Communication and Client's Access	✿✿✿	✿✿✿
Installation and Execution	✿✿✿✿✿	✿✿✿
Maintenance and Reliability	✿✿✿✿	✿✿✿✿
Developing Easiness	✿✿	✿✿✿✿

Table 1. Comparative study between Flash and Silverlight

As we can see through the table analysis, in 25 possible points, the technologies could not achieve grades above 80%. The Silverlight had one point more than Flash. This is relevant but, in our opinion, is far from being decisive. In a technological perspective, the Flash's market share is high and this makes Flash the best option. However, observing the unique characteristics of both technologies, we can say that Silverlight has bigger potential to grow comparing with Flash and it will become a really serious competitor in a short time. The Flash is a tested and commercial format with great portability e adaptability to different platforms and its executables are short sized. These characteristics should maintain it as market leader in the near future.

However, in a market with a constant expansion as the RIA's one, there are characteristics that will define how will the software applications be in the future – the use of those applications in mobile devices. In this aspect, the Flash will have more constraints to move on with the natural market's evolution. In our opinion, those characteristics are 3D Hardware graphic acceleration; Time-based animations; Programming languages easy to understand and usable in different contexts.

Observing these characteristics, we can say that Silverlight presents itself in the front line to succeed Flash in the RIA's market. If Flash does not evolve itself in this way, it could become obsolete in a medium term period.

In the specific context of developing a tool to automatically generate dynamic reports, both technologies prove to be sufficient to accomplish it. However, the Silverlight presents a better performance due to its developing easiness and user experience, allowing building complex applications to automatically generate dynamic reports, in a richer and faster way.

7 Conclusions

Nowadays, the reports can no longer be represented by blank and white paper reports, strictly static in the way they represent information and hard to search and to extract relevant information.

The automatic generation of dynamic reports is, in our opinion, the future of this kind of information representation. The developing of this kind of tools using RIA is possible and highly profitable in the products' quality and programming perspective.

Comparing the technologies used to implement these solutions, the Microsoft Silverlight demonstrated to be the best solution for, essentially, two reasons: on one hand, its user experience that transforms the views and the editions of the reports in simple and pleasant operations. On the other hand, the developing easiness allowed by the using of the .Net framework is high because the Silverlight uses technologies belonging to that framework, well known and largely used in the whole world like CSharp.

As future work, we will develop a multi-database architecture that will allow us to develop, in a medium term future, a commercial product to automatically generate the dynamic reports supporting one, or more, platforms.

References

1. Peck, George. Crystal Reports 2008: The Complete Reference. McGraw-Hill Professional, (2008).
2. IBM. liveSite - Website Content Management Software. Obtained in 1st January of 2010. http://www.lotusmuseum.com/ (2009).
3. Microsoft. The Microsoft Office Fluent user interface overview. Obtained in 1st January of 2010,http://office.microsoft.com/en-us/products/HA101679411033.aspx (2009).
4. Kaplan, R. S., & Norton, D. P. The Balanced Scorecard. Harvard Business School Press. (1996).
5. Rodrigues, J. A.. Controlo de Gestão e Performance. Obtained in February 16th of 2009, http://www.indeg.org/cursos/mestradosexecutivos/contabilidade/controlo/ (2007).
6. Ezell, J. Silverlight vs. Flash: The Developer Story. Obtained in 19th February of 2009, http://weblogs.asp.net/jezell/archive/2007/05/03/silverlight-vs-flash-the-developer-story.aspx (2008).
7. Moritz, F. (January of 2009). Rich Internet Applications (RIA). Obtained in 5th January of 2009, http://www.flomedia.de/diploma/documents/DiplomaThesisFlorianMoritz.pdf
8. Microsoft. Data Points: Service Driven Apps With Silverlight 2 and WCF. Obtained 17th February of 2009, http://msdn.microsoft.com/en-us/magazine/cc794260.aspx (2009).
9. Microsoft. Isolated Storage. Obtained in 17th February of 2009, http://msdn.microsoft.com/en-us/library/3ak841sy(VS.80).aspx (2009).
10. Microsoft. Deep Zoom. Obtained in 17th February of 2009, http://msdn.microsoft.com/en-us/library/cc645050(VS.95).aspx (2009).
11. Microsoft. Common Language Runtime Overview. Obtained in 22th January of 2009, from http://msdn.microsoft.com/en-us/library/ddk909ch(vs.71).aspx (2008).
12. Microsoft. Microsoft Message Queuing. Obtained in 16th February of 2009, http://www.microsoft.com/windowsserver2003/technologies/msmq/default.mspx (2008).
13. Kaplan, R. S., & Norton, D. P. *The Execution Premium - Linking Strategy to Operations for Competitive Advantage.* Harvard Business School Press. (2008).
14. Christodoulou S. P., Styliaras G. D. and Papatheodorou T. S., "Evaluation of Hypermedia Application Development and Management Systems". 9th ACM conference on Hypertext and Hypermedia, pp. 1 – 10. ACM Press, Pittsburgh (1998)
15. J.C. Preciado, M.Linaje, F.Sanchez and S.Comai, "Necessity of methodologies to model Rich Internet Applications". WSE - Proceedings of the Seventh IEEE International Symposium on Web Site Evolution, pp. 7 – 13. IEEE Computer Society, Washinton DC (2005)
16. Microsoft. (March 2008). Swiss MSDN Team Blog : Silverlight 2 Beta 1 + WCF + LINQ to SQL. Obtained in 5th January of 2009, http://blogs.msdn.com/swiss_dpe_team/archive/2008/03/17/silverlight-2-beta1-wcf-linq-to-sql-a-powerfull-combination.aspx (2009)
17. Horwitz, Steven. "The Sensory Order and Organizational Learning." in *The Social Science of Hayek's the Sensory Order (Advances in Austrian Economics): 13*, from Roger Koppl and Steven Horwitz. Emerald Business (2008)