The Analysis of Reporting Tools for a Cadastre Information System

Dariusz Król
Wroclaw University of Technology, Institute of Applied Informatics, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
dariusz.krol@pwr.wroc.pl

Jacek Oleksy
SIEMENS Software Development Center,
Strzegomska 52A, 53-611 Wroclaw, Poland

Małgorzata Podyma
Computer Association of Information BOGART Ltd.,
Rejtana 9-11, 50-015 Wrocław, Poland

Bogdan Trawiński
Wroclaw University of Technology, Institute of Applied Informatics, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
trawinski@pwr.wroc.pl

Abstract

The process of selecting and implementing of reporting mechanisms for a cadastre information system is described in the paper from the point of view of system requirements. Many factors as user expectations, report usage, ability of report designing and modifying, possibility of presenting complex data in a readable way, costs and licensing as well as the performance and scalability required have been taken into account. Three versions of reporting mechanisms based on Crystal Reports, Microsoft Reporting Services and XML technology have been implemented and tested.

1. Introduction

Numerous articles and books have been written on software engineering [4], [13], [19], [21], many deal with requirement engineering [10], [12], [14] and various with software quality assurance [5], [16], [20], [22]. Sommerville in his fundamental book on software engineering [21] presented a model of design process where architectural design constitutes the initial phase of designing systems and requirement specifying is one of the most important activities (see Figure 1). According to Pressman [19] during software development the quality of design concerns the specification of requirements and system design. There is

a lot one can say about high quality of realization if the implementation conforms to the design and the final product fulfils the requirement specification and is sufficiently effective. Glass [9] in turn proposed an intuitive formula for user satisfaction equivalent to good software where quality is defined implicitly:

User satisfaction = compliant product + good quality
+ delivery within budget and schedule

Another definition of software quality is given by Boehm [3] who enumerates the set of quality attributes:

Quality = portability + reliability + efficiency + human engineering
+ understandability + modifiability + testability

The process of selecting and testing of reporting mechanisms in a cadastre information system is reported in the paper as a part of developing software requirement specification. Four years of developing and maintaining cadastre systems in many information centres in Poland provided experience that users regard reporting functions as the most important in the system and require reporting tools of high quality. They expect reports presenting complex data clearly, allowing to select data using various criteria, enabling to change the scope of data shown in a flexible way and taking time as shortest as possible to generate.

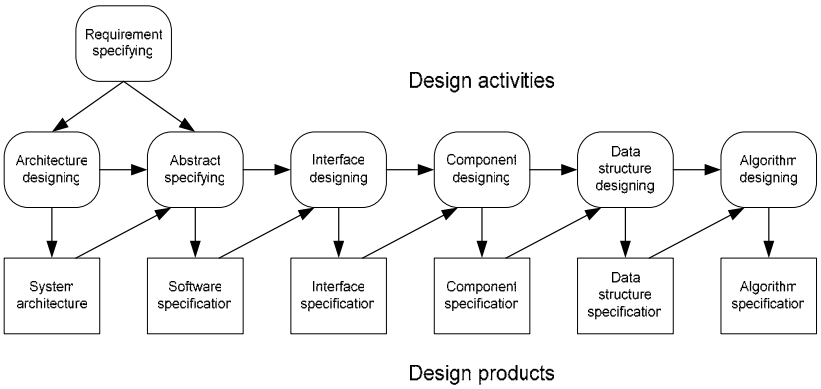

Figure 1. General model of design process [21].

Selection of a reporting tool for an information system is a difficult task. It should be accomplished carefully and many factors should be taken into account i.e. user expectations, ability of report designing and modifying, possibility of presenting complex data in a readable way, costs and licensing as well as the performance and scalability required. The study presented in the paper has been carried out by the Department of Information Systems at Wroclaw University of Technology in cooperation with a commercial software company. Different research methods have been used including the analysis of report usage on the basis of report logs maintained by the system, a questionnaire to reveal users'

requirements, review of the offers of reporting tool producers and finally the tests of performance and scalability.

2. Reports in the Cadastre System

The maintenance of real estate cadastre registers is dispersed in Poland. There are above 400 information centres located by district local self-governments as well as by the municipalities of bigger towns which exploit different cadastre systems. The majority of cadastre systems in Poland are developed in two-layer client-server architecture and deployed on Microsoft SQL Server or Oracle database management systems running on Intel servers. At present data in cadastre systems are not complete yet. However descriptive data of land premises are fully complete, but at present information centres are gathering the data of buildings and apartments and prices of premises. Numeric plans of real estate are being created or complemented too. The last legal regulations introduced a new object model of cadastre data and forced software companies to modify their cadastre applications.

The EGB2005 system for which we were looking for the best reporting tools is a new information system being under development which relational database complies with object structure of cadastral data. The EGB2000 system, which is a predecessor of the EGB2005 system and actually used, has been deployed in above 100 local governments throughout Poland while the EGB2000-INT system providing an internet access to cadastral databases is used in about 50 intranets and extranets.

The report users of the cadastre system can be divided into three groups. The first one consists of the workers of information and documentation centres, who everyday update the database of the system and use extracts from land, building and apartment registers as well as notifications of changes. They also inform and give different documents to the owners of real estate and to surveyors. The second group comprises the workers of local governments who utilize data to prepare administrative decisions and to prepare reports for management boards of local governments. The last group is called ‘branches’ that means institutions and companies which deliver water, gas, electrical energy, heat and others to the inhabitants of a given town or district.

There are 117 reports available in the system and they can be grouped into 9 types. In order to investigate how heavily reports are used in the system a special procedure maintaining a report usage log has been implemented. In Table 1 nine report types are presented together with the number of reports contained in each group and the mean daily usage of reports calculated on the basis of data taken from the report usage logs from three bigger centres for the period of June 2005. It can be easily seen that only three groups of records are frequently used by the users of the system. These are extracts from registers, notifications of changes and various lists of objects with their attributes.

Table 1. Report types and their mean daily usage.

Report types	No. of rep.	centre 1	centre 2	centre 3
Extracts from registers	19	124.7	91.4	291
Extracts from registers with a map	3	0	0	0
Notifications of changes	5	11.8	27.4	27.5
Lists of objects	34	36.1	2.6	84.3
Aggregate data reports	15	0	0.4	0
Governmental statistics	5	0	0	0
Reports of changes	7	0	0	0
Control of consistency	24	0	0	0
User monitoring	5	0	0	0

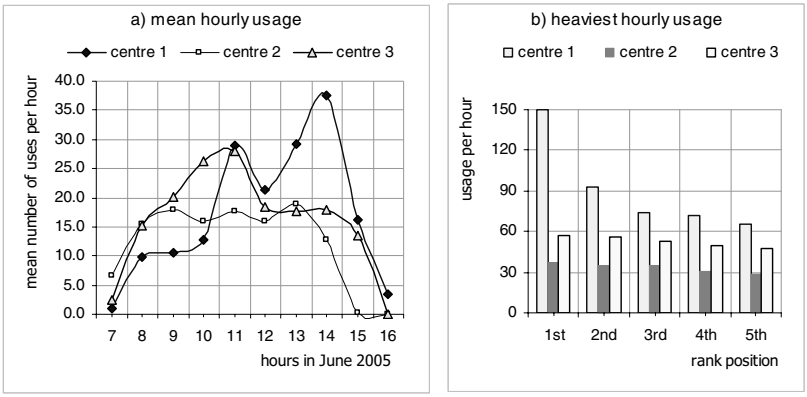

Figure 2. Hourly usage of reports in selected centres.

The report logs were also analyzed to reveal the heaviest hourly usage of reports in the system. Figure 2a presents mean hourly usage of all reports during June 2005. The top five results for each centre are shown in Figure 2b. Data indicate that the maximum hourly usage reached 150 what equals to only 2.5 per minute. Although it should be expected that the usage of reports will grow in the future, it is clear that the system is far from load limits yet.

As it has been already mentioned above high quality of reporting mechanisms in the cadastral system is required. Even simple reports are built of several elements. In Table 2 the number of elements of three most frequently used reports in the EGB2000 system is presented. These reports are (1) a list of parcels, (2) an extract from land register and (3) a notification of changes

introduced into land register. All the reports in the EGB2000 systems were designed using Crystal Reports 8.5.

Table 2. Number of elements implemented in selected reports.

Elements of a report	1.	2	3
Sections	25	51	158
Subreports	1	4	15
Groupings	3	20	19
Formulas	36	82	112
Totals	2	3	54
Conditional formatting expressions	13	40	128

3. Investigation of users' expectations

In order to investigate the expectations of report users a questionnaire was constructed and sent to 100 centres exploiting the EGB2000 system. The system users were asked to assess the importance of various reporting functions, to evaluate time of generating reports of different size, to estimate how often they use different types of reports and to judge to what extent the system functions fulfill their needs. Below selected results of 39 properly filled questionnaires are presented. In Table 3 you can see how respondents assessed the importance of different reporting functions. The same data are visualized in Figure 3.

Table 3. Assessment of the importance of reporting property (in percentage of responses).

No.	Property.	Not	Slightly	Important	Very
1	Time of generation	0	2	28	70
2	Readability	0	0	13	87
3	Aesthetic layout	0	6	51	43
4	Complex queries retrieving data	0	4	45	51
5	Saving search queries	0	27	58	15
6	Hiding or displaying chosen sections	0	9	58	32
7	Attaching additional data to report	0	13	58	28
8	Preview content before print	0	4	30	66
9	Modifying content before print	4	20	59	18
10	Designing new types of reports	2	17	55	26

No.	Property.	Not	Slightly	Important	Very
11	Export into Word format	2	8	38	52
12	Export into Excel format	2	0	40	58
13	Export into PDF format	6	27	21	46

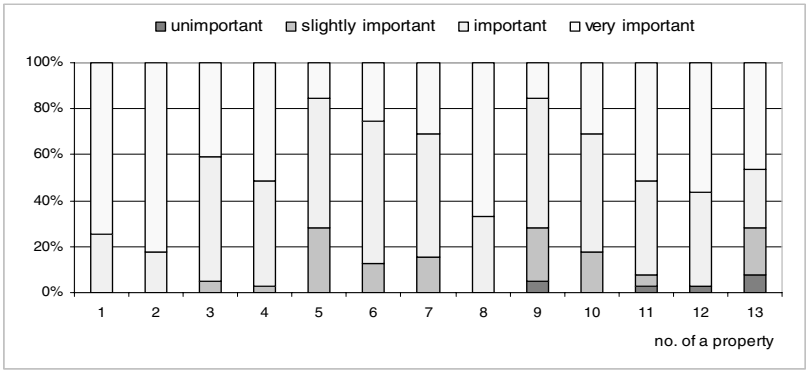

Figure 3. Assessment of the importance of reporting property (numbers as in Table 3).

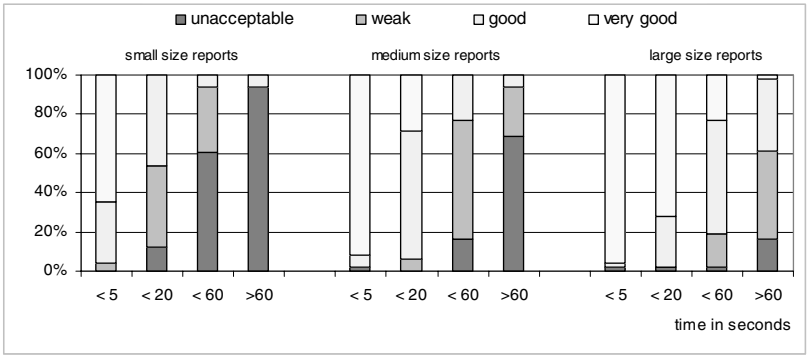

Figure 4. Acceptability of report generation time.

In turn in Figure 4 it is shown how users evaluate time of generating the extract form land register, which is one of the most frequently used reports. Percentages of responses are given for small size reports (up to 10 records), medium size reports (up to 100 records) and large size reports (above 100 records).

4. Overview of Reporting Tools used in Information Systems

On the market there are many reporting tools. Selected features of some of them are presented in Table 4. They can take data from various sources such as ODBC, JDBC, EJB, OLE DB, Oracle, DB2, Microsoft SQL Server, Access, BDE, XML, TXT and others. Almost all of them allow to export reports to such file formats as HTML, PDF, XML, RTF, XLS, CSV, TXT and some to DOC, LaTeX, TIFF, JPEG, GIF. The selection process should also take into account results of feature comparative analyses [8], [18] and benchmark tests carried out on reporting tools by their producers or by independent organizations [1], [2], [6], [7]. All benchmark tests have proved that such commercial tools as Actuate iServer, Crystal Enterprise and Cognos ReportNet significantly exceeded the requirements of the cadastre system considered in the paper. However those benchmark tests were accomplished using highly efficient and expensive hardware and were designed for the developers who want to build large-scale applications. Therefore the results were not usable for the decision process of selecting the most convenient reporting tool for the cadastre system. This was the main reason for the performance and scalability tests we conducted.

Prices shown in Table 4 are only illustrative and were selected from massive lists of prices. However just the prices usually have decisive influence on the selection of reporting tools for the cadastre system. Commercial tools are perceived by local governments to be too expensive and it was the main reason for that we started to design and program reporting mechanisms based on open source tools. We analyzed and tested three relatively new reporting tools: Crystal Reports 10, Microsoft Reporting Services and mechanisms based on XML. The analysis of costs and licensing conditions was also necessary because our main clients which are local governments used to force us to provide the least expensive solutions.

Table 4. Selected features of reporting tools.

Tool name	Producer.	Operating system	Programming environment	Price
Active Reports .NET	Data Dynamics	Windows	MS Visual Studio.NET	$599 - $1499
Crystal Reports 10	Seagate Software	Windows	any	$495 - $1995
DataVision	Jim Menard	Windows, Linux	Java	open source
FastReport	Fast Report	Windows	Borland Delphi	$89 - $359
Formula One e.Report	Actuate	Windows	Java	$495 - $995
IntelliView Suite	IntelliView	Windows, Linux	MS Visual Studio.NET,	$975

Tool name	Producer.	Operating system	Programming environment	Price
JasperReports	Teodor Danciu	Windows, Linux	Java, COM Java, MS Visual Studio.NET	open source
JReport	Jinfonet	Windows, Linux	Java	$2000
Rave	Nevrona	Windows	Borland Delphi, C++ Builder	$149 - $349
Report Builder Pro	Digital Metaphors	Windows	Borland Delphi	$249 - $749
Report Sharp-Shooter	9Rays.NET	Windows	MS Visual Studio.NET	$375 - $1750
ReportNet	Cognos	Windows, Linux, UNIX	any	$250-$2000.
SQL Server 2000 Reporting Services	Microsoft	Windows	MS Visual Studio.NET	included in SQL Server 2000 price
StyleReport	InetSoft	Windows, Linux	Java	$1495

5. Testing of Reporting Mechanisms

Three versions of reporting mechanisms were implemented in the EGB2005 system, i.e. mechanisms based on Crystal Reports 10 (CR10), Microsoft Reporting Services (MRS) and XML technology (XML). Each solution was tested with the most frequently used report i.e. an extract from land register unit. The following hardware was used in the experiment: Intel Pentium PC with 2.8 GHz clock and 2 GB RAM as the server. Windows 2000 Professional and Microsoft SQL Server 2000 were installed in the computer. The configuration is common for smaller information centres where the EGB2000 system is exploited.

5.1. Testing of reporting mechanism using Crystal Reports 10

The architecture of reporting mechanism using Crystal Reports 10 implemented in the EGB2005 system is shown in Figure 5. The example.rpt report containing SQL queries selecting data from database is called with a set of parameters. Report Engine sends formatted data in EPS to Report Viewer which enables to print or export the report.

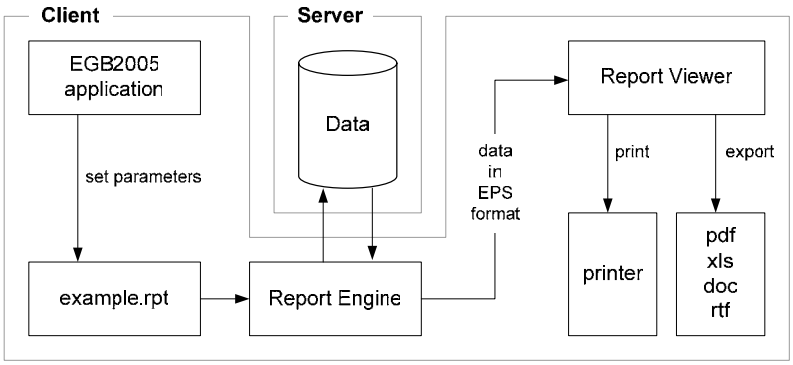

Figure 5. Architecture of reporting using Crystal Reports (CR10).

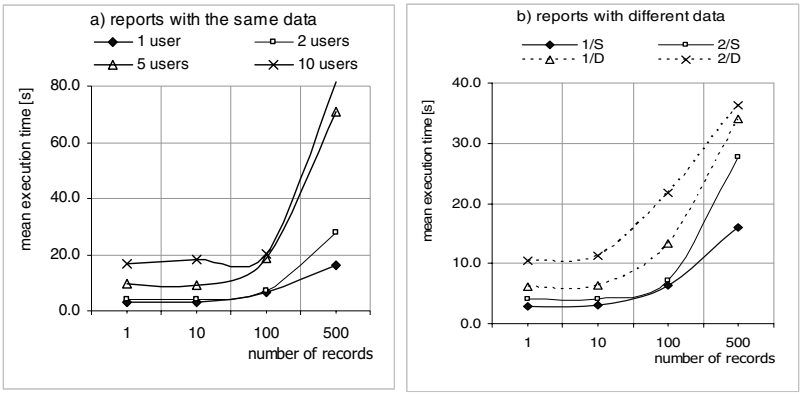

Figure 6. Test results of Crystal Reports mechanism (CR10).

The tests simulated continuous work of 1, 2, 5 or 10 users which generated about 20 the same reports containing 1, 10, 100 or 500 parcels. Mean time of generating individual reports was calculated. The results presented in Figure 6a allowed us to state that the performance of the reporting solution was acceptable, taking into account the results of users' expectations research and the analysis of report usage. In order to investigate what impact report caching could have on the performance further tests were carried out using the reports with different data this time. In Figure 6b 1/S and 2/S mean one and two users generating reports with the same data and 1/D and 2/D denote one and two users executing reports with different data. There were only slight differences between the results of both tests.

5.2. Testing of reporting mechanism using XML technology

In Figure 7 the architecture of reporting mechanism using XML technology implemented in the EGB2005 system is presented. The application formulates a SQL query and sends it to the DBMS and saves retrieved data in a XML file. Processor XSLT sends formatted report to a browser or enables to create the report in the PDF or doc format using XSLT FO.

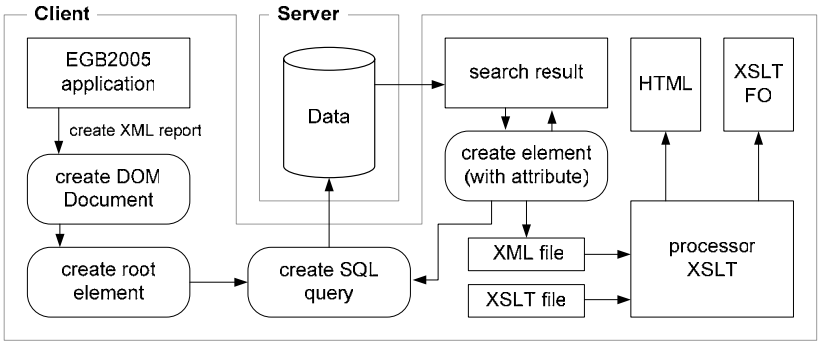

Figure 7. Architecture of reporting using XML technology (XML).

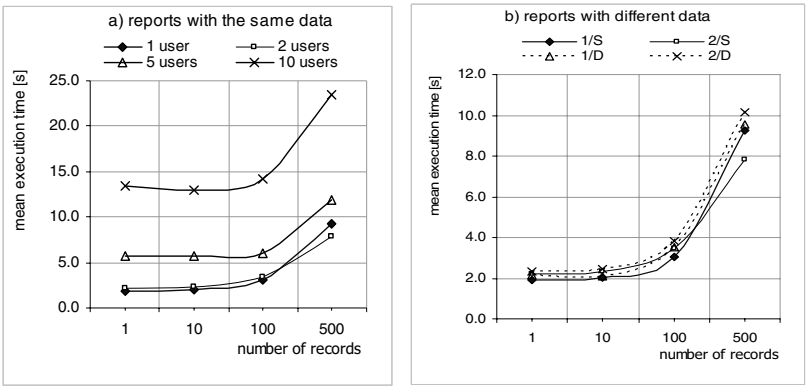

Figure 8. Test results of XML mechanism (XML).

The tests simulated continuous work of 1, 2, 5 or 10 users which generated about 20 the same reports containing 1, 10, 100 or 500 parcels. Mean time of generating individual reports was calculated. These tests gave the better results than previous tests especially as far as bigger number of records included in reports is concerned (see Figure 8a). This mechanism also seems to be

satisfactory taking into account the results of the questionnaire and the analysis of report usage.

Similarly, further tests were carried out using the reports with different data in order to investigate what impact report caching could have on the performance. In Figure 8b 1/S and 2/S also mean one and two users generating reports with the same data and 1/D and 2/D denote one and two users executing reports with different data. Almost no difference between the results of both tests could be observed.

5.3. Testing of reporting mechanism using Microsoft Reporting Services

In Figure 9 the architecture of reporting mechanism using Microsoft Reporting Services implemented in the EGB2005 system is presented. The application calls a report using a browser. Report Server uses a report definition retrieved from the Report Server Database and data taken from the data source to create a report. Then the report is transformed into the HTML format and sends to the browser.

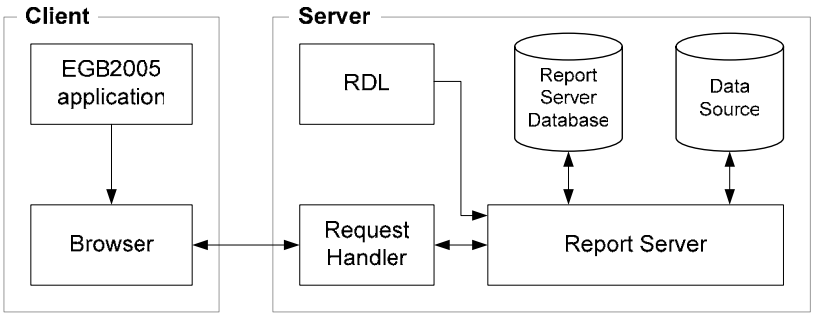

Figure 9. Architecture of reporting using Microsoft Reporting Services (MRS).

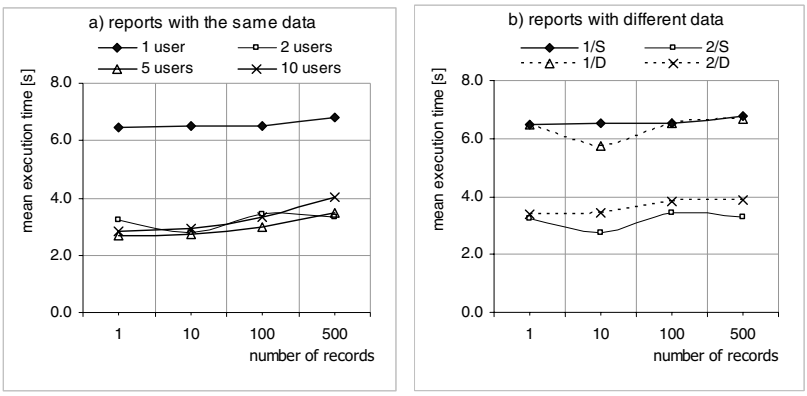

Figure 10. Test results of Microsoft Reporting Services mechanism (MRS).

The tests simulated continuous work of 1, 2, 5 or 10 users which generated above 20 the same reports containing 1, 10, 100 or 500 parcels. Mean time of generating individual reports was calculated. The Microsoft Reporting Services mechanism gave the best results (see Figure 10a). The analogous tests to investigate the impact of report caching did not reveal substantial differences (see Figure 10b).

6. Conclusions

The performance tests have proven that all three reporting mechanisms could fulfil the system requirements. Although the mechanism based on Microsoft Reporting Services revealed the best performance, it should be treated cautiously. It is relatively new and not all features and behaviours are known yet. Moreover we do not suppose that our clients having Oracle databases would maintain Microsoft SQL Server as their second database environment. In turn Crystal Reports is limited to only one level of subreports what considerably makes it more difficult to design reports selecting complex data.

After completing all tests we scored all reporting mechanisms. Eleven of the most important requirements were chosen: 4 of them were functional (F) and the others were non-functional (NF). Each requirement was assigned a weight of significance from the point of view of the cadastre system. Then reporting mechanisms were rated to what extent they satisfy individual requirements. The 5-point scale was used where 1 denoted "not at all" and 5 indicated "completely". The results are shown in Table 5. The final score calculated as the sum of rates multiplied by corresponding weights allowed us to make decision to base reporting mechanisms in the cadastre system on the XML technology.

Table 5. Scoring the reporting mechanisms implemented.

Type	Requirement	Weight	CR10	MRS	XML
F	Layout, readability, aesthetics	0.4	5	5	5
F	Ability to retrieve data	0.7	5	5	5
F	User intervention after generation	0.2	2	2	3
F	Export into popular document formats	0.5	4	4	4
NF	Programming environment	0.6	5	5	5
NF	Ability to represent complex data	0.8	3	4	5
NF	Time of implementation	0.5	4	4	3
NF	Client environment	0.8	4	5	4
NF	Server environment	1.0	5	2	5
NF	Licenses, prices	1.0	2	3	5
NF	Performance	0.9	3	5	3
	Total score		28.2	29.6	32.5

The XML technology seems to not have the drawbacks of other mechanisms tested taking into account all present aspects of the cadastre system usage at local governments. We appreciated wide scope of XML language functions and especially the ability to separate content processing form presentation layer and the dynamic access to the content of XML documents.

7. References

1. *Actuate 7 iServer vs. Crystal Enterprise 9*. An Interactive Reporting Performance Benchmark Project Final Report. National Technical Systems, 2003.
2. *Actuate 7SP1 Performance and Scalability on Windows 2003*. Actuate Corporation, 2004.
3. Boehm B., *Practical Strategies for Developing Large Software Systems*. Addison Wesley Longman, 1975.
4. Cadle J., Yeates D, *Project Management for Information Systems*. Pearson Education Limited, 2001.
5. Carriere S., Kazman R., Woods S., "Assessing and Maintaining Architectural Quality", *Proceedings of the Third Conference on Software Maintenance and Reengineering*, 1999.
6. *Cognos ReportnNet Scalability Benchmarks – Microsoft Windows*. Cognos, 2004
7. *Crystal Enterprise 9.0 Baseline Benchmark: Windows Server 2003 on IBM@server xSeries 16-way*, Crystal Decisions, 2003.
8. *Feature comparison. Microsoft SQL Server 2000 Reporting Services vs. Business Objects Crystal Reports / Crystal Enterprise*, Certia Business Intelligence Team, 2004.
9. Glass R.L., "Defending Quality Intuitively", *IEEE Software*, May/June 1998.
10. Górski J. (red.), *Inżynieria oprogramowania w projekcie informatycznym* (in Polish), MIKOM, 2000.
11. Król D., Podyma M., Trawiński B., "Selection and testing of reporting tools for an Internet Cadastre Information System", *Frontiers in Artificial Intelligence and Applications,* IOS Press, Vol. 130, 2005, pp.305-316.
12. Leffingwell D., Widrig D., *Managing Software Requirements. A Unified Approach*, Addison-Wesley, 2000.
13. Leishman T., "Extreme Methodologies for an Extreme World", *The Journal of Defence Software Engineering* Vol. 14 No. 6, 2001.
14. Linscomb D., "Requirements Engineering Maturity in the CMMI", *The Journal of Defence Software Engineering,* Vol. 16 No. 12, 2003.
15. *Microsoft SQL Server 2000 Reporting Services*. Product Evaluation Guide. Microsoft, 2004.
16. Paulk M. C., A *Comparison of ISO 9001 and the Capability Maturity Model for Software*, Carnegie Mellon University, 1996.
17. Peck K. G., *Crystal Reports 10: The Complete Reference*, McGraw-Hill, 2004.
18. Podyma M., *Review and comparative analysis of reporting tools for internet information systems*, M.Sc. Thesis (in Polish), Wroclaw University of Technology, 2005.
19. Pressman R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 2001.

20. Satpathy M., Harrison R., Snook C., Butler M.B., "A Generic Model for Assessing Process Quality", *In Proceedings of IWSM2000 - New Approaches in Software Measurement*, LNCS 2006, Springer Verlag, 2001.
21. Sommerville I., *Software Engineering*, Addison-Wesley, 2004.
22. Trendowicz A., Punter T., "Quality Modelling for Software Product Lines", *Proceedings of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2003.