Experiences from the Architectural Migration of a Joint Replacement Surgery Information System

Samuli Niiranen[1], Ari Välimäki[2] and Jari Yli-Hietanen[1]

[1]Department of Signal Processing, Tampere University of Technology, POB 553, 33101 Tampere, Finland. [2]AMV Oy, Lintulamminkatu 13, 33400 Tampere, Finland.

Abstract: The goal of this study is to present the experiences gathered from the migration of an existing and deployed joint replacement surgery information system from a classical 2-tier architecture to a 4-tier architecture. These include discussion on the motivation for the migration and on the technical benefits of the chosen technical migration path and an evaluation of user experiences. The results from the analysis of clinical end-user and administrator experiences show an increase in the perceived performance and maintainability of the system and a high level of acceptance for the new system version.

Introduction

Background

Tekoset is a joint replacement surgery information system developed to provide a centralized database for all the information related to the management of the surgery from pre-operative examination to operation and post-operative follow-up (Lehto et al. 2000). The system was originally developed in 1998–2000 for use in the Pirkanmaa region of Finland as a co-operation between the Pirkanmaa Health Care District, AMV Oy (www.tekoset.com) and the National Agency for Medicines. Its use has since expanded to other regions of Finland, including the Helsinki and Uusimaa, Päijät-Häme, and Central Finland health care districts as well as the Reumasäätiö hospital in Heinola. Altogether, the system serves 6 secondary care hospitals and 27 primary care health clinics. The total number of health care professionals using the system is ca. 120 and, as of March 2008, the data of approximately 70000 joint replacement surgeries have been recorded to the system.

The key functionalities of the system are

- structured data entry through elaborate sets of GUI controls and aggregate reporting of prosthetic components installed in hip, knee, ankle, shoulder, elbow and wrist joint replacement surgeries. The information includes the trade names, types, serial numbers and physical dimensions of the installed components. Also, additional details on the operation, such as immediate complications, are also recorded. The data entry is carried by the operating orthopedic hospitals. Aggregate information on the installed components is transferred annually to the National Agency for Medicines for quality assurance and national reporting purposes. This functionality replaces manual, paper-based reporting on installed prosthetic components by the operating hospitals.
- an automated call-up system for alerting post-operative follow-up sites on forthcoming follow-ups. This functionality removes the reliance on a manual call-up list management and enhances information transfer between operative and post-operative follow-up sites.

In addition to these key functionalities, the system provides for the registration of pre-operative examination data and follow-up notes (including pre- and post-operative scores measuring operation outcomes) as well as a diverse set of reporting functionality for institutions using the system.

Migration

The initial implementation of the Tekoset system was based on a classical, 2-tier client-server architecture as shown in Figure 1.

Correspondence: Samuli Niiranen, Tampere University of Technology, POB 553, 33101 Tampere, Finland. Tel: +358 40 8490719; Fax: +358 3 3115 3857; Email: samuli.niiranen@tut.fi.

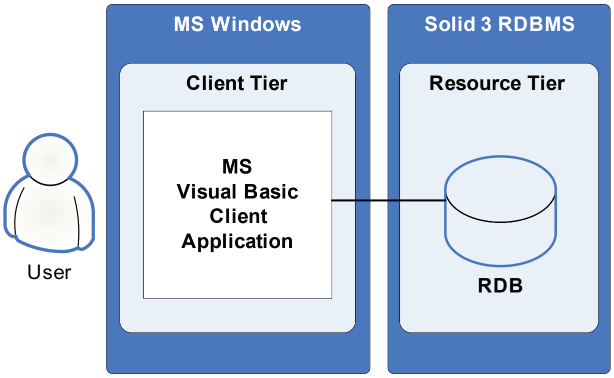

Figure 1. Initial software architecture of Tekoset.

The system consisted of a relational database deployed on a Solid EmbeddedEngine 3.3 relational database management system (Solid Information Technology, Cupertino, CA, U.S.A.) and a GUI client application built with the Visual Basic 9 development environment (Microsoft Corporation, Redmond, WA, U.S.A.). Open Database Connectivity (ODBC) was used to connect the client application to the relational database in a networked environment. All access to the database was implemented as (Atomicity, Consistency, Isolation, Durability) ACID-compliant transactions.

In early 2005, changing requirements from the existing customer base, competitive market pressure and the potential advantages of a more modern architecture resulted in the initiation of a project at AMV Oy for the migration of the Tekoset software architecture to a multi-tier, browser-based solution.

Why this choice of a migration path? The benefits of modern multi-tier architectures in health care information system design are well-established. These include reusability, flexibility, reductions in initial system development cost and effort, and, more controversially, provision of an easy, open migration pathway for future change of technology and system redevelopment (Chu et al. 2000). In similar fashion it is generally agreed that, the move towards thin-client, browser-based user interfaces from fat-client solutions simplifies system deployment and maintenance as the need for client installations is lessened and deployment administration is centralized. The result has been that new system development work in health care information systems has moved towards adopting these technologies (Haux, 2006). In the presented project, the most critical issue motivating the migration was the projected reduction in the customers' maintenance costs resulting from the adoption of a browser-based solution eliminating manual client installations. Another issue of importance was, if possible, to increase performance in terms of system responsiveness during regular use.

Considering architectural migrations, a number of successful migration strategies and patterns have been reported for use in similar projects related to mission critical information systems. See (Hasselbring, 2004) for one example where an incremental migration process is presented for converting legacy systems into multi-tier architectures. Another good reference on migration is (Bergey, 1997) which describes a framework for the disciplined evolution of legacy information systems. It provides guidelines, principally in the form of checklists, for transforming legacy systems into modern architectures.

In general, the key issues for consideration are the selection of best-fitting technologies for the multi-tier architecture as well as the mode and pace of migration. For this project, the JBoss 4 Java EE application server (Raleigh, NC, U.S.A.) and MySQL 5 relational database management system (Uppsala, Sweden) platforms were chosen as the deployment environment. Both represent widely used open source technologies with a number of attractive qualities approaching the capabilities of commercial solutions. The baseline for technology adoption was that the replacements should on all levels provide, at minimum, the same level of functionality, performance and robustness as the legacy technologies while providing the increased maintainability of a browser-based solution. As an example, MySQL 5 is the first version of the system with true support for ACID-compliant transactions, a capability already provided by the previous database platform. The graphical user interface (GUI) of the legacy system was finely tuned to meet customer requirements. Thus, the goal was to replicate in existing GUI as well as possible.

As shown in Figure 2, the current implementation of Tekoset is based on a 4-tier architecture. The client application is a web browser, served by presentation tier JSF (Java Server Faces) components generating the system GUI. Access to the database is provided through custom persistence components and transfer objects (TO) implemented as standard Java components. The standard JDBC (Java Database Connectivity) API was used to access the database from the integration tier. All access to the database was implemented as ACID-compliant transactions.

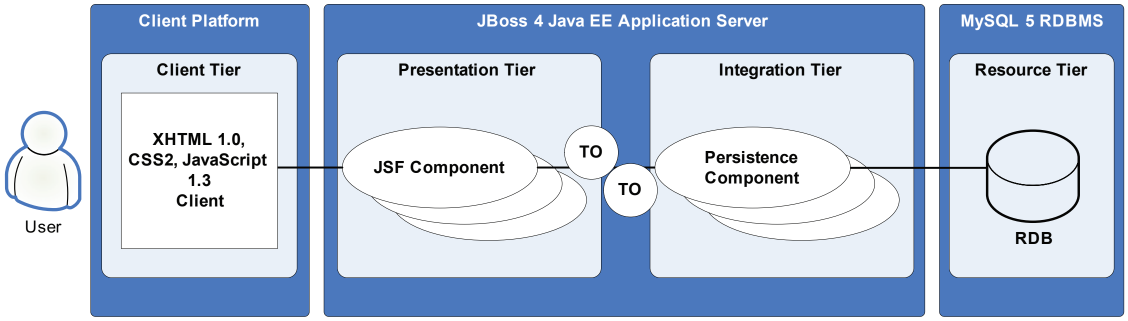

Figure 2. Current software architecture of Tekoset.

Concerning the mode and pace of migration, the project began with the replication of existing functionality and implementation of system enhancements requested by the existing customer base. In addition, a decision was made in the beginning of the process that the migration would be carried out in a single change-over operation. All legacy platforms and functionality would be simultaneously replaced with the new ones. The replication of legacy functionality and implementation of new functionality were naturally intertwined as the architectural migration proceeded. No major problems were encountered during this process. However, a key challenge, which required significant effort, was to provide the users with a graphical user interface experience similar or better to that of the legacy system. Visual Basic provides a much richer set of GUI controls as standard compared to standard HTML form elements. A related matter of contention was to maintain similar GUI speed in the browser environment. Much effort was spent to extend the standard JSF GUI components with client side Javascript functionality from component visibility selection and maintenance to validation in order to prevent needless page reloads, which would have made the user experience significantly slower than with the legacy application. No major obstacles were faced during the functionality migration process with system stress testing finalized and database migration software ready for deployment in March 2006.

The actual migration of customer deployments to the new platform was carried out one customer at a time beginning in April 2006 and ending in December 2006. No major technical problems occurred during the change-over process or during early operational use of the migrated system.

In the following experimental part of study we will provide and discuss the results of a survey registering the experiences of the end-users and the system administrators in relation to the change-over.

Methods

In March 2008, the authors contacted the administrators of the four Tekoset-installations which had migrated to the new system version in 2006 to notify their clinical end users that a web-based questionnaire had been set up to assess the performance and acceptance of the new system version in comparison to the old version. Perceived changes in performance and acceptance were queried on a Likert-scale in terms of overall speed-of-use (or responsiveness), ease-of-use in the context of performing different end-user tasks, quality of GUI layout, reliability (or error-freeness) and ease-of-maintenance (or maintenance workload) for administrators. The questionnaire also included a number of demographic and professional items to characterize the responders. In all, 25 users with experience on both the old and the new version were contacted by the administrators with a request to fill in the questionnaire. Up to two reminders were sent. Considering the analysis, questionnaire data pulled from a database were tabulated using Microsoft Excel.

Results

Table 1 lists the characteristics of the 15 responders; the response rate was 60% (15/25) overall. Considering the data, more than half of the responders were orthopedic surgeons who are the largest single user group of the system. Orthopedic surgeons primarily record operation data into the system including data on the installed prosthetic components and other operative details. Physical therapists mainly fill in the pre-operative

Table 1. Characteristics of the questionnaire responders (N = 15).

Characteristic	% (n)
Age	
<20	0% (0)
20–30	0% (0)
31–40	27% (4)
41–50	33% (5)
51–60	33% (5)
>60	7% (1)
Sex	
Male	73% (11)
Female	27% (4)
Profession	
IT administrator	13% (2)
Operating orthopedic surgeon	60% (9)
Physical therapist	20% (3)
Other nurse	7% (1)
Experience in profession (years)	
<10	20% (3)
10–20	13% (2)
>20	67% (10)
Organization	
Public primary care clinic	7% (1)
Public hospital	67% (10)
Private hospital	27% (4)

examination and post-operative follow-up data. Most of system end-users work currently in hospitals, organization of post-operative follow-ups at public primary care clinics is an ongoing reorganization.

Table 2 lists the distributions of the responses to the five items of the questionnaire. In summary, responses to items 1–4 indicate that for each item more than half of the responders (55% to 67%) perceived an increase in the performance of the system. The most positive response was given to item 3 while the negatives were the highest for item 1. Item 5 was applicable only to the two IT administrators who provided responses to the questionnaire. Their responses indicate agreement with an increase in system maintainability.

Discussion

The results from the analysis of clinical end-user and administrator experiences show an increase in the perceived performance and maintainability of the system and a high level of acceptance for the new system version. Considering the item with the largest share of negative responses, speed-of-use, there are many possible explanations for this. One is the fact that users do need to initially open the browser client which can take a longer time on older workstations than the launch of the legacy system client program. Also, occasional (slow) page reloads are practically unavoidable with a web client. On the other hand, in practical deployment testing the new RDBMS was observed to be significantly faster than the one it replaced. Supporting this, item 1 also had the largest share of "strongly agree" responses of all the items. Considering the study design, a longitudinal survey would perhaps have been a preferable methodology to the one used here. However, due to practical restrictions this was not possible in the presented project. Also, only the perceived, aggregate performance of the entire system was evaluated. A more detailed technical performance analysis would provide more generalizable results, for example, in terms of evaluating the difference in speed between the two database management systems with the same data.

On a general level, despite the success of the presented architectural migration project, often such endeavours represent one of the major challenges of practical informatics. The migration

Table 2. Responses to the questionnaire (N = 15 responders).

The new version ...	Strongly agree	Agree	Neither	Disagree	Strongly disagree	Not applicable
1. Is faster to use	27%	33%	27%	13%	0%	0%
2. Is easier to use	13%	47%	27%	7%	0%	7%
3. Has a better layout	27%	40%	27%	7%	0%	0%
4. Is more reliable	20%	33%	47%	0%	0%	0%
5. Is easier to maintain	7%	7%	0%	0%	0%	87%

process does need careful planning and execution for successful completion. As an example, a major technical pitfall is that without a careful design and implementation, a thin-client GUI can end up being less responsive than that of a conventional client application. Considering the overall need for architectural migrations of medical information systems, there is potential for the emergence of a vicious circle due to an ever increasing number of clinical systems providing increasingly complex functionality. As much of this functionality is typically tightly interconnected on a system level, the migration of a single system to a new platform often creates a cascade of changes in the entire health information system infrastructure of an institution. Still, specific problems need specific solutions and the presentation of monolithic, single-system electronic medical records as a solution to this problem can be argued to be an unrealistic approach.

In conclusion, the presented case serves as an example of a medical information system migration with a generally positive outcome in terms of customer satisfaction. Naturally, the results of this single study do not prove the appropriateness of the presented migration path in all medical information system migration scenarios.

References

[1] Lehto, M., Nevalainen, J. and Numminen, O. 2000. Informaatioteknologia tekonivelkirur-gian palveluksessa [Information technology in the service of joint replacement surgery]. *Suomen lääkärilehti [Journal of the Finnish Medical Association]*, 33:3250–3.

[2] Chu, S. and Cesnik, B. 2000. A three-tier clinical information systems design model. *International Journal of Medical Informatics*, 57:91–107.

[3] Haux, R. 2006. Health information systems—past, present, future. *International Journal of Medical Informatics*, 75:268–81.

[4] Hasselbring, W., Reussner, R. and Jaekel, H. 2004. The Dublo architecture pattern for smooth migration of business information systems: an experience report. *Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*:117–26.

[5] Bergey, J., Northrop, L. and Smith, D. 1997. Enterprise framework for the disciplined evolution of legacy systems. Technical Report CMU/SEI-97-TR-007, Carnegie Mellon University/Software Engineering Institute.