Integrating a web application with Siebel CRM system

Mika Salminen, Antti Seppälä

Helsinki University of Technology, course Business Process Integration: Special Course in Information Systems Integration, fall 2007

Abstract. This paper describes a case of integrating Siebel CRM (Customer Relationship Management) with Finnish Red Cross (FRC) training-site application. The main focus is on evaluating a suggested batch-transfer based integration architecture using use case scenarios and Software Architecture Analysis Method (SAAM) method. The batch approach is compared against synchronous web service -based architecture and good and bad sides of the both approaches are listed. For the FRC's case the batch-transfer based system is found to be more appropriate, because of the different availability requirements of the integrated systems.

1 Introduction

Background of the Study

The study is based on a real-life integration case in Finnish Red Cross (FRC). The charity organization has Siebel Customer Relationship Management (CRM) system already in place and the goal is to integrate parts of the CRM to their new training-site application. They have already created a suggestion for the integration architecture, but not yet implemented or fully evaluated it.

The training-site is a web application for managing the first-aid training courses arranged by the FRC. Course instructors can create classes and make them available for attendees through the application. Companies can assign their employees to the courses and the system keeps track of the course attendees, the courses they have attended and the competencies received from the courses. All the information is ultimately stored to their CRM system. In the future, the training-site will be the main tool for arranging the training for everyone, either individual or a company.

The main motivation to integrate the training-site with the CRM arises from the fact that FRC wants to be able to better benefit from the contact information gathered on the training-site and to have a central register for keeping track of course attendees and their received competencies. As Injazz J. Chen and Karen Popovich believe, a successfully implemented CRM can be customer driven and technology-integrated business process management strategy [1]. The contact information can be used for example to marketing purposes: The competencies are only valid for a limited period of time, so some kind of notification might be sent to people whose competency is about to expire.

FRC project team has already created one suggestion for the integration architecture, but it still needs to be evaluated to find out if it fits their needs. Evaluating the architecture and finding out how to improve it forms the base of this study. Our study ended up to do the evaluation by comparing the suggested architecture with an architecture using different kind of approach, presented in this report. Another architecture was only created to be used as a reference for the comparison.

Objectives of the study

The first objective of this study is to give an elaborate description of the integration architecture suggested by the FRC project team. To give background for the integration architecture description, the processes and data on the training-site are also described. Describing the suggested architecture is necessary to support the other objectives of the study.

The second objective is to find out the requirements and quality criteria for the integration architecture. What is really needed and what are the most important or crucial parts of the architecture. The FRC team mostly gives these, because they have the best knowledge about their specific needs.

The third objective is to verify that the suggested implementation meets the requirements set for the integration. When the required criteria for the evaluation is clear, the suggested architecture is evaluated against it to see how well it really fits the needs. The study tries to find the possible problems, bottlenecks and places for improvement in the architecture. This is required for the FRC's project to advance to implementation stage.

Research Questions

The research questions are derived straightforward from the objectives. Does the suggested implementation meet the requirements set for the integration? If not, what needs to be corrected?

Scope of the Study

The scope this study is limited to the process of data transfer between the Siebel CRM and training-site. All the specific technical details are ruled out as far as it is possible. Only the limitations set by the technology used will be considered. The architecture is evaluated and developed on high level.

Handling the architecture on abstract level allows concentrating solely on the real problem and makes the architecture and individual parts of it more applicable for different systems. The goal is not to develop a universal solution for CRM integration to company specific information systems. But on the other hand, our research should be generic enough to have also some value and ideas for other integration projects.

Methodology of the Study

The material for the report was gathered from literature, presentation and e-mail communication with FRC. Communication with the FRC contact Vesa Palmu was the main method for getting information about the architecture and its requirements.

The architecture evaluation is done based on use case scenarios made for integration architectures. The Software Architecture Analysis Method (SAAM) [2] is applied to evaluate and compare the architecture against a reference architecture explained in this study. The SAAM method was selected because it is described to be suitable for requirement based architecture review [2]. It was not obvious that SAAM is the best method to do the evaluation, but it seemed to be simple enough to use in the scope of the study.

SAAM has 5 different work phases, which are show in Fig 1. In our study, FRC had one architecture description already and another one is defined in this document for reference. The reference architecture is created to help with the evaluation and its purpose is to solve the integration problem in a different way. Without any comparison point for the evaluation it is hard to tell which features of the architecture are "good" or "bad" in some respect. Created reference architecture is a plausible option for similar integration issues.

When architectures and requirements have been described, the scenarios can be developed. While creating the scenarios, the architectures and requirements need to be kept in mind in order to end up with scenarios, which really test the desired aspects of the architecture. To get good results from the study the scenarios play a big role, because the overall evaluation is made based on the results of scenario evaluation.

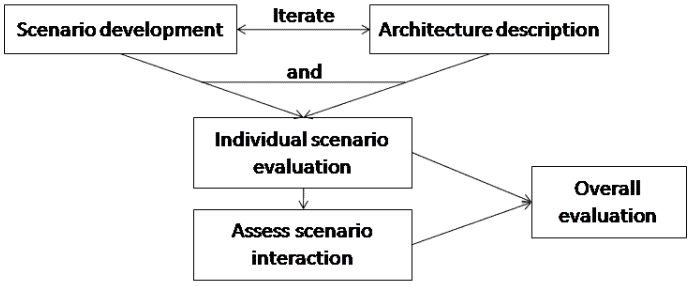

Fig 1: SAAM activities

For each scenario it is defined if it has direct or indirect relationship with the architecture. SAAM defines direct scenarios such that they can be implemented in architecture without modifications; indirect scenarios need modifications to architecture to be doable. Different scenarios may also have interaction with each other: indirect scenarios may require changes to the same components or connections [2]. During the project it was noticed that when architectures are described on a very high level, it is not always clear whether scenario can be applied to architecture directly or indirectly.

In SAAM it is assumed that scenarios are different in nature, so finding out the scenario interactions also tells if the architecture supports an appropriate separation of concerns [2]. The scenarios, created for the architecture evaluation, are based on the requirements for the architecture, to see how well the architectures satisfy these requirements. For indirect scenarios the needed changes for the architecture are listed.

In our study the overall evaluation includes comparing the two architectures and is based on the observations made in the scenario evaluations and scenario interaction assessment.

Structure of the Report

The suggested architecture is first described to give enough understanding about the problem domain. The database model of training-site is described shortly on detailed level to show the reader what kind of data the training-site needs to handle. The conceptual model of the integration architecture shows whether the data is created on the training-site or in CRM.

After the suggested architecture has been described, the requirements for the system are explained. These requirements define quite much what kind of use case scenarios, should be selected for the architecture evaluation.

When the requirements have been described, the architecture description of the reference architecture, against which the suggested architecture is compared, is explained. This reference architecture is developed to get some ground for comparison in the use case scenarios.

After the architectures' descriptions the use case scenarios are presented and, architectures' suitability for each case is explained. At this point the good and bad sides of the architectures are dug out based on the scenarios.

After use case scenarios the results gathered in research are analyzed. Conclusions about the results are drawn and improvement suggestions for the candidate architecture are given.

2 Training-site data

FRC provided an Entity-Relationship diagram of the training site (Fig. 2). In this section the diagram of the database is explained, just enough to get idea of what kind of information is stored in the training-site database. The main motivation for describing the data is not to show the presentation or structure of the data in the database, but to concentrate on the semantics of the data.

Courses are the definitions of classes. Course information consists of information such as course description, prerequisites and the competency the course provides. Classes are created based on a specific course.

Classes are instances of a single arrangement of course. About the classes, in the database is such information as the intended audiences, class languages and translations and the course this class represents. Accounts and contacts have references to classes they are involved or attended to.

Accounts represent companies or organizations or just individual people. Accounts can have multiple addresses, name and a VAT id to identify the account. Contacts that are involved with the account have reference to the account.

Contacts are instances of individual people with their contact information, membership number, id in Siebel, language etc. Contacts are the entities that have enrollments to classes and competencies.

Enrollment entity has the information about the contact enrollment, which defines, when the contact attended the course, if and when she completed it etc.

Competencies represent the skills gained on classes and they define only the competency name and id in Siebel.

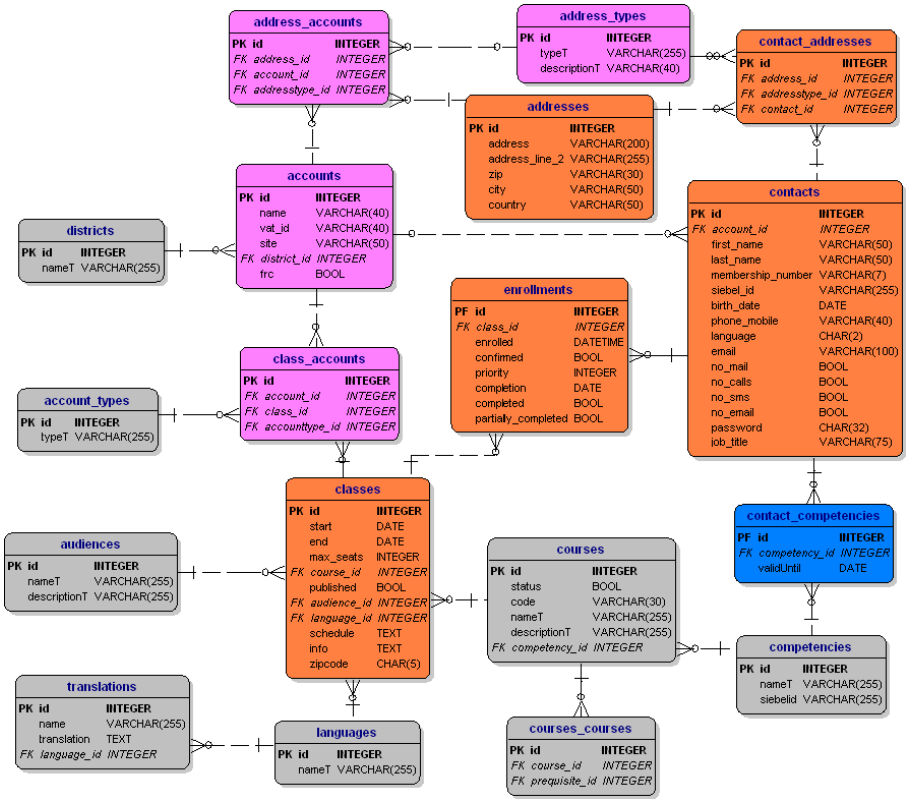

Fig 2: ER-diagram of the training site database

3 Suggested integration architecture

FRC representative described the architecture that was suggested for the integration [3]. The conceptual model of the architecture is presented in figure 1. The architecture is based mostly on batch transfers of data between the applications. The transfers are deployed using Secure File Transfer Protocol (SFTP) and are done periodically, usually once a month. Updates to the data on the training-site side are copied back to the CRM database.

This architecture is based on a classical Extract-Transform-Load [4] procedure, where data is extracted from the source, transformed into a form that receiver can read and finally loaded into the receivers database.

The actual data transferred from the CRM to training-site consists of the course instructor's information such as contacts and competencies. Also the course information with basic information and required competencies is transferred to the training-site.

Client information, basic information about the classes held and student information gets transferred from the training-site to CRM. Contact information is checked against the population register to check the correctness of the people's contact information. This is not done every time data gets transferred from the training site, but the checks are processed in different batches.

Most of the integrations are done automatically by loading the data into CRM system and out of it by using load tables. Only the course information transfer is not automated. Course information is not changed very often, so it is supposed to be left out of the automatic transfer and updated only when needed.

The suggested integration architecture is not very tightly coupled, although it is based on data integration [5]. In this case integration by data means that both systems use the same information but it might be presented to both systems differently and mapped from presentation to another in a transformation. If the data presentation changes in one end, the transformation can usually be changed to adapt to the change.

The architecture has also some features of "people as integrators" -architecture because of some of the course information transfer requires people to do the data transfer [5]. The course information is imported separately into both systems.

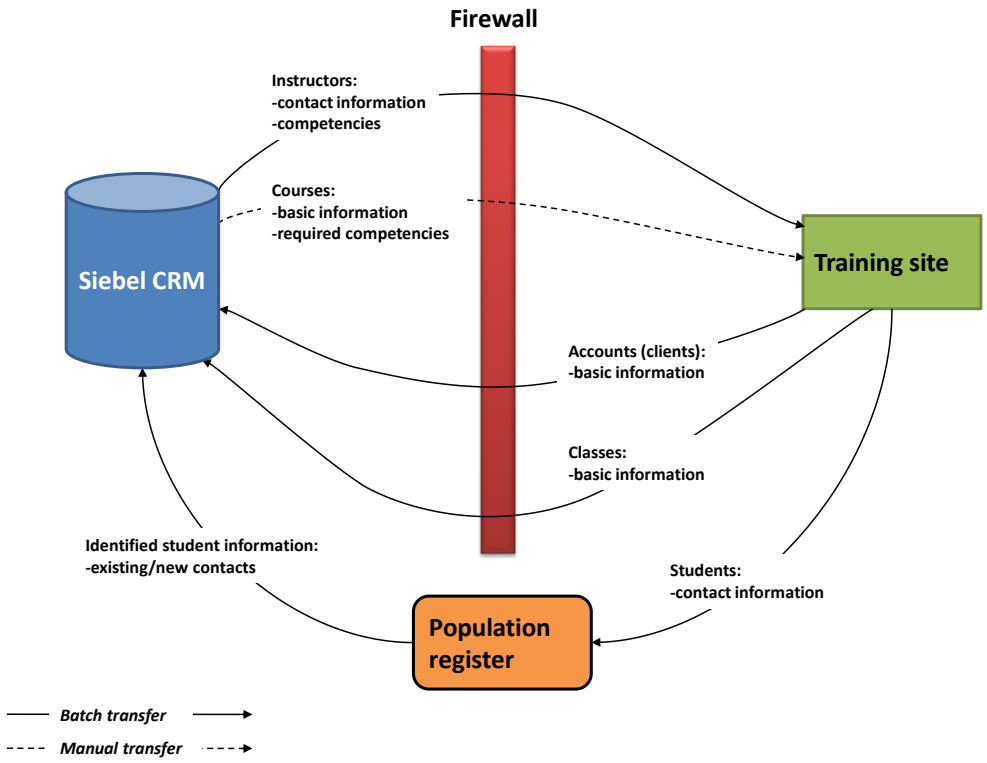

Fig 3: Conceptual model of suggested architecture

4 Requirements

Many requirements for the integration architecture were gathered in the meeting arranged for the topic introduction [3] and through email communication with the FRC contact Vesa Palmu. Many of the requirements are special requirements for this specific case, but it is assumed that the system needs to have some properties that were not separately mentioned by Palmu. For example good architecture needs to perform well enough and be maintainable to be usable. The project team elaborated additional requirements for a usable, good system.

Requirements explicitly set by FRC

Security

The training site is located in a public network and stores students' personal information. Therefore, the security of the personal data must be guaranteed. The system does not hold any sensitive information like social security numbers, but it does store names, email addresses and other contact information.

All the sensitive information is stored in Siebel CRM and it shouldn't be accessed from the training site. The data transfer between the sites has to be encrypted. It is not sufficient to use clear text since the server is located in a third party server hotel. This is to prevent an eavesdropper from gaining information.

Availability

Siebel CRM is not guaranteed to be online outside the working hours. But the training site should be available to people around the clock. This means that the training-site cannot be fully dependable on Siebel and it has to keep its own data store.

Data consistency

The data doesn't have to be synchronized in real-time between the systems. It's sufficient to update the data by scheduled batch jobs. The system must recognize if a student who enlisted to a course exists in the CRM database. Duplicate entries in students and companies should be avoided.

Data freshness

Data is checked against population register quite rarely due to the expenses. This limits the freshness of data and it is enough do data transfers approximately once a month.

Additional requirements

Performance

Mr. Palmu said nothing specific about the performance needs for the systems, but it is assumed that normal users of the system should not suffer from performance related usability problems. Both systems, the training-site and Siebel CRM, must not be slowed down too much in normal circumstances.

Scalability

The training-site is not something that people use every day, so the number of concurrent users and number of changes to data will not probably grow up very fast. Still the amount of data transferred between the Siebel and training site can be big if the transfer is not done very often and the data is buffered somewhere.

Maintainability

System should be sufficiently easy to maintain with resources available at FRC. Human resources are quite limited so the integration implementation should not require frequent maintenance. Requirements for the training-site can also change after it has been published, which requires the integration to be flexible enough to support additions and changes.

5 Reference architecture

Enterprise systems are getting more and more complex and they need to be more scalable, secure, and available than ever before, because the amount of data and users are increasing. Everyone is aiming to automate processes and reduce actual work done by individuals. The reference architecture that is used in comparison was developed to keep the future in mind and it uses an alternative approach to FRC's problem. It is designed to use commonly suggested service based integration [6]. It is also specified that this architecture is synchronous to get some overview of good and bad sides of the asynchronous architecture versus synchronous in this specific integration case. Selecting service based architecture for the comparison reference was done because many enterprise systems including Siebel seem to support services as the main integration interface for their system.

The reference architecture is designed not to use batch transfers and store data locally, but instead to allow real-time data operations through web services. For real-time access directly to Siebel data, the architecture needs to offer a seamless channel for communication. One solution that offers this kind of functionality is the Oracle's Siebel CRM On Demand [7].

It uses standard techniques like Web Services Description Language (WSDL) and Simple Object Access Protocol (SOAP) to provide Siebel integration as web services. The web services are described using WSDL, which is an XML based language for describing web service interfaces. The actual communication is made using SOAP, which uses XML messages to share business objects and service calls between the communicating parties.

If we reflect this architecture to SOA Reference Model [8], the use of WSDL implements the service description part of the Reference Model and SOAP is the service interface.

By using web services it doesn't automatically mean Service Oriented Architecture, but at least it allows adopting a loosely coupled integration. Siebel CRM On Demand is also scalable and it's relatively easy to describe new services [7] when new feature needs arise.

When the training site and Siebel CRM basically share the same database, there are no batch jobs or any human interaction needed for data transfer. The training-site relies on Siebel's data and its availability. All the processes done on training site communicates with the Siebel database directly through web services.

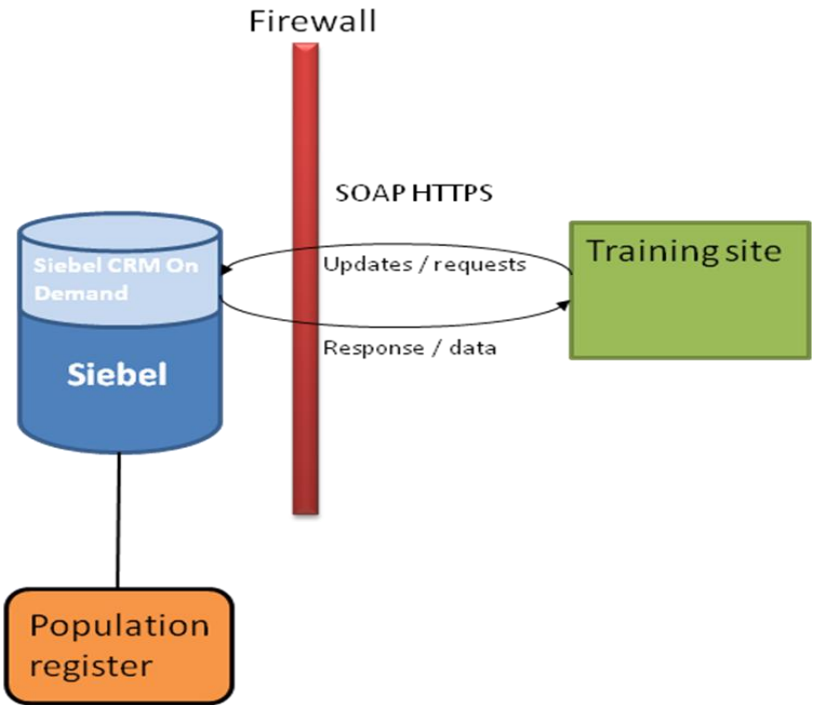

Fig 4: Reference architecture

6 Use case scenarios

The suggested, candidate architecture is analyzed against the reference architecture using use case scenarios for the system. These specific scenarios were selected because they test the different requirements given for the system and are well suitable to show the differences between the integration architectures. The cases have either direct or indirect relation to architecture. Relationship types and needed improvements are mentioned in the following tables.

Scenario 1: System needs to be improved to provide some real-time data from Siebel to students and instructors visiting the training-site.

	Batch-architecture		**Direct web service integration**
indirect	With batch architecture the data is old by default. Shortening batch-cycle for some data might enable it to be very up-to-date.	direct	All data is accessed and updated real-time using Siebel's database.

Scenario 2: Hacker should not be able to start a denial of service (DOS) attack against the training-site, which could make the CRM unusable during the attack.

	Batch-architecture		**Direct web service integration**
direct	Training-site does not query against the CRM database, so flooding queries through the training site does not affect the CRM database performance.	indirect	DOS attack could possibly slow-down the CRM, because the system is servicing also the training-site's queries. • One way to prevent this is to recognize the attack and close down

			the access to the web service during the attack • Query count per time period could also be always limited to some level that does not stress the CRM too much, but enables normal usage of the training-site

Scenario 3: If hacker compromises the site and gains direct, administrative access to the server, he or she should not be able to access personal data of contacts.

	Batch-architecture		**Direct web service integration**
direct/ indirect	All data in training-site database is usually accessible to administrative user in a way or another. However, personal data that is never needed on the training-site but is stored only in CRM cannot be accessed through training-site. • Securing the training-site well is necessary	direct/ indirect	Compromising the training-site does not give hacker direct access to database, but will probably enable her to query data through Siebel web services. She can also make immediate updates to the CRM's data. • Selecting the data that is accessible through web service should be done very restrictively • The training-site and all nodes that can access the CRM web services should be secured well

Scenario 4: Siebel is run down for maintenance. Instructor navigates to training-site to start a new class.

	Batch-architecture		**Direct web service integration**
direct	This scenario is supported in the batch architecture as such because the training-site has always a local copy of all the needed data.	indirect	In web service architecture the training-site is dependable of the CRM's web services. To make the architecture better support data access when the CRM is offline, following approaches could be used. • Aggressive caching of data read from the web service and buffering the updates to data while the CRM is down. These are Possibly complicated to implement efficiently

Scenario 5: Customer calls to FRC to change his or her contact information and email address. Shortly after the change, instructor starts a new class and invites the same user to attend the class.

	Batch-architecture		**Direct web service integration**
direct/ indirect	User might not get the invitation to the course or be able to log in to the training-site. • To overcome this problem, same changes could also be applied to training-site's database	direct	Customer's contact information always matches the data in CRM's database, because it is ultimately received from there.

	• To fully automatically support this kind of urgent changes, some extra functionality is needed. • For example urgent changes could be transferred in batches more often.		

Scenario 6: To check the correctness of the data received from the training site, the contact information is checked against population register.

	Batch-architecture		**Direct web service integration**
direct	All the incoming, changed data from training-site can be checked against population register before it is imported to CRM database.	indirect	All the changes and additions to contact data are done straight to the CRM database. • System needs to be able to distinguish between changed or added and unchanged data. • Layer that logs the changes to contact data needs to be implemented to allow distinguishing the changed data • When distinguishing the changed data from unchanged is possible, it can be checked periodically against population register

Scenario 7: A new field is added to contact information to support storing the gender of the contact. Training-site application is updated to support this change.

	Batch-architecture		**Direct web service integration**
indirect	• Training-site's database needs to be updated to support the new field. • The batch job has to be updated to transfer the new information.	indirect	• The web service definition needs to be updated to support gender field.

Scenario 8: A new law requires more people to have first-aid competencies. Number of training-site users rapidly grows and system has to scale to support the new number of users.

	Batch-architecture		**Direct web service integration**
indirect	• Tuning the training-site hardware and software to support the change might be enough to keep the performance good • Batch transfers might grow up very big and they have to be done quite often to keep the size of data	indirect	Conflicts in data between CRM and training-site should not be a big problem in this architecture because the queries through the web services are done in transactions. • To keep performance good, both, the

	transferred in one batch reasonable • Number of conflicts and errors that need to be resolved can be hard to handle		training-site's and CRM system's performance should probably be improved

Scenario 9: A new batch is received from population register to Siebel and there are few persons who have changed their names. The data needs to be updated to Training site.

	Batch-architecture		**Direct web service integration**
direct	Identifying a person is made according to identity number. Updated data will be transferred to training site on next batch run.	direct	Identifying a person is made according to identity number. The data is updated to Siebel and ready for viewing also in Training site.

Scenario 10: New personal information has been entered on Training site, which is a duplicate already existing in Siebel, but not all info matches.

	Batch-architecture		**Direct web service integration**
indirect	• Some rules must be created to decide do we want to trust the end user or the latest data from population registry.	indirect	• A new entry is created to Siebel and duplicates are handled separately when the database is checked against population register.

7 Analysis

In this chapter the results from use case scenarios are gathered and analyzed. The objective is to evaluate how well the suggested architecture answers to the challenges set by the scenarios and what should be improved. According to the SAAM method the interactions between different scenario solutions should be recognized and explained. The analysis is started by explaining these interactions and after that further architecture evaluation is done.

Scenario interaction analysis

The limiting factors of the architecture types are faced in the scenarios quite well. This can be recognized from the interactions between scenario solutions. Making suggested batch-based architecture to support real-time data is not possible without losing its benefits such as independence from the CRM system. Making batch updates very frequently makes it possible to have almost real-time data on training-site database but requires high availability of the CRM system. It might also cause performance issues, since large loads of data need to be read and written frequently.

On the other hand, making web service based architecture to buffer data to allow it to provide service even when the CRM system is offline brings the web service -architecture closer to the batch architecture. Web service based system would probably need a local database for keeping the buffered data and it would bring along many problems of the batch architecture. Data in the buffer could be inconsistent with the data in the CRM database; buffered data could be accessible for attackers gaining access to the training-site etc. To be able to support all possible needs of the training-site users during the CRM offline periods, the training-site should buffer all data from the CRM, but full data buffering would actually turn the web service –system into a kind of batch-based system.

It seems like the biggest tradeoff between the architectures is made with the availability of the CRM system. Web service -based architecture does not suit well into environment where the required web

services are not available during all the time the training-site is available. This seems obvious in a situation where all the data for the training site is provided by the backend CRM system.

Architecture based on the requirements

Use case scenario interaction analysis shows that the suggested architecture is not perfect solution for every purpose but the real question that the FRC is interested in is whether the architecture fits their needs. How does the architecture answer to their requirements?

The security of the architecture depends much of the security of the database the information is kept in. Attacker can possibly get access to all the data in the database on a server, which is available from public Internet. Training site's database should not contain information that is not absolutely necessary for the training site. Attacker has probably not very good chances to access other CRM data, than what is stored on training-site database, by compromising the training-site which could be the case with web service based approach. Batch transfer based architecture is probably better in security sense, because it does not give attacker access to all data in CRM even in worst case.

The availability of the training-site even when the CRM is offline is a very good side of the suggested architecture. Because CRM and training-site are implemented completely independently, only synchronizing the data periodically, the training-site's availability depends only on the availability of itself. Proper function of web service architecture depends always at some level on the CRM's availability. In this case where the CRM's availability cannot be guaranteed, but the training-site should always be available, batch transfer seems to be better solution.

The information consistency and between the databases might be an issue in suggested architecture. Because CRM and training-site keep both their local copies of data, the changes in one end are not immediately visible on the other side. Users might make conflicting changes to information and determining the wanted result is not always straightforward. Proper conflict resolving strategy must be developed to prevent batch transfers becoming painful operations, with lots of need for human interaction. Also making the batch transfer more often could reduce the stacked up conflicts and make the resolving easier. Web service based approach that would make the changes straight to the CRM database would not suffer from the conflict problem.

As stated before the suggested architecture does not enable data on either side to be always up-to-date. The freshness of data depends on how frequently the batch transfers are run. Although real-time data is not required or possible to achieve with batch transfers, it allows the adjustment of the data freshness by adjusting the update frequency. Web service based architecture would allow access to real-time data on both sides as they use the same data.

In performance-sense the batch-architecture does not place any load on the CRM system during its normal functioning. The performance of the system depends only on the performance of the Training site system. Batch transfers cause peaks in load periodically, so they cannot be done too often or should be done at nighttime. With web service based system the performance would be dependable of both systems. In that sense the batch based system is better in overall performance and better by the requirements.

Regarding the system scalability, the increase in number of users affects the number of changes made to the data. In batch transfer system this means that to scale the system up, batch transfers must be made less frequently, affecting the freshness of data, or bigger amount of data must be transferred in one transfer, which could increase the amount of conflicts. Also the training-site platform might need to be upgraded to support more users. These should not be very big issues or affect the fulfillment of the other requirements so the architecture can satisfy this requirement. Web service based architecture would require also scaling up the CRM platform performance, but any compromises regarding the data consistency or freshness would not be required.

Biggest maintainability problem with suggested architecture would probably be resolving the conflicts during the batch transfers. If the data is not updated much on the CRM side this problem can be ruled out. If the data presentation change on one side or additional data is brought from CRM to training-site, only the batch transfer definitions need to be updated. In web service based architecture the conflicts are not problems, and maintaining the service definitions is not probably bigger job than maintaining the batch transfer definitions. As a result the web service based architecture might be easier to maintain, but batch

transfer system should not be very difficult to maintain either if the conflict resolving can be done automatically.

Improvement suggestions

If the batch updates are not done very often there might be lots of conflicts and duplicate entries in data when it is imported from training-site database to CRM database. As stated before resolving conflicts manually might cause quite much extra effort so doing it automatically is preferable. Therefore, automatic resolving of conflicts should be investigated more and rule-set for the task created.

Another issue that should be resolved is to find out how to keep the necessary contact information on training-site fresh enough to support communication to users. For example if the e-mail address of the user changes in CRM, he or she might not be reachable anymore using the address stored on the training-site. Should some changes propagate to the training-site database faster than others and how this could be implemented?

The amount of data transferred in one batch transfer will be very big if all the training-site data is transferred in each batch. It should be considered if the batch transfers could do incremental updates to the data and only transfer changed data records. This requires all changes to the data to be logged, but would probably allow more frequent batch runs than once per month and that way more fresh data. It should be investigated if this is technically possible and does not complicate the architecture too much.

8 Conclusions

In this study we have show that the suggested implementation good enough to meet the requirements that are currently set, but it could benefit from certain improvements. Therefore we believe that the goals set for this study have been achieved. FRC have now got an outsiders review of their architecture and it have been verified to meet the minimum requirements set.

The batch architecture really seems to be a good solution in FRC's case, because the training site has to be available even when the Siebel isn't. The alternative web service based architecture fails in this environment or it will really difficult to implement properly.

Using of SAAM to accomplish the overall evaluation turned out to be a good tool in this kind of architecture review. The scenarios used could all be faced also in real life and they provided a good way to see possible problems in such situations.

Most problems are faced in keeping the data coherent while minimizing the amount of human interaction, which was also the main concern of FRC. This will be the most challenging part of the integration when it will actually be implemented.

9 References

1. Injazz Chen and Karen Popovich, Understanding customer relationship management (CRM), People process and technology, Business Process Management Journal Vol. 9 No. 5, 2003 pp. 672-688

2. Kazman, R.; Abowd, G.; Bass, L.; Clements, P., "Scenario-based analysis of software architecture," *Software, IEEE* , vol.13, no.6, pp.47-55, Nov 1996

3. Case introduction session with Vesa Palmu at Finnish Red Cross's Helsinki office 25.9.2007.

4. Wikipedia, Extract-Transform-Load, http://en.wikipedia.org/wiki/Extract%2C_transform%2C_load (accessed on 20.11.2007)

5. Helsinki University of Technology, Information and Computer Systems in Automation, AS-116.3110, slides of lecture held on 22.3.2007 by Ilkka Seilonen, http://www.automationit.hut.fi/file.php?id=730 (accessed on 1.11.2007)

6. Harikumar, A.K.; Lee, R.; Hae Sool Yang; Haeng-Kon Kim; Byeongdo Kang, "A model for application integration using Web services," *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*, vol., no., pp. 468-475, 2005

7. Oracle, Siebel CRM On Demand Web Services whitepaper, http://www.oracle.com/crmondemand/downloads/white_papers/crmondemand_webservices_whitepaper.pdf (accessed on 20.11.2007)

8. OASIS Open, C. Matthew MacKenzie et.al, Reference Model for Service Oriented Architecture, http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf, (accessed on 10.1.2008)