EXTENSIBLE ARCHITECTURES: THE STRATEGIC VALUE OF SERVICE-ORIENTED ARCHITECTURE IN BANKING

Baskerville, Richard, Department of Computer Information Systems, Robinson College of Business, Georgia State University, PO Box 4015 Atlanta, Georgia 30302-4015, email: baskerville@acm.org

Cavallari, Marco, Teamlab s.a., Via Curti, 2, 6901 Lugano, Switzerland, email: marco.cavallari@teamlab.ch

Hjort-Madsen, Kristian, The IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark, email: khm@itu.dk

Pries-Heje, Jan, The IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark, email: jph@itu.dk

Sorrentino, Maddalena, Dipartimento di Studi Sociali e Politici, Università degli Studi di Milano, Via Passione 13, 20122 Milano, Italy, email: maddalena.sorrentino@unimi.it

Virili, Francesco, Dipartimento Impresa e Lavoro, Università degli Studi di Cassino, Loc. Folcara, 03043 Cassino (FR), Italy, email: francesco.virili@eco.unicas.it

Abstract

Information and communication technology (ICT) has helped to drive increasingly intense global competition. In turn, this intensity increases the need for flexibility and rapid changeability in ICT to support strategies that depend on organizational agility. We report a comparative, cross-cultural case study of the implementation of Service Oriented Architectures (SOA) at a Scandinavian bank and a Swiss bank. The strategic rewards in the adoption of SOA appear to go beyond marketplace issues of ICT capability acquisition, and unexpectedly arise in the creation of an extensible organizational ICT architecture. The extensibility of the ICT architecture that results from the adoption of SOA provides potential for greater organizational agility (and thereby competitiveness).

Keywords: Architecture, Service-Oriented, Case Study, Extensibility.

1 INTRODUCTION

Information and communication technology (ICT) has helped to drive increasingly intense global competition. The upswing created by electronic commerce in the latter 1990s has continued to expand into the first part of the new century as global digital bandwidth increased. Companies increase globalisation through sourcing strategies that make geography increasingly transparent for digitally based communications and operations.

Agility is often considered a key success factor in order for organizations to compete in this global marketplace. Agility is a special form of flexibility that is both quick and easy. For organizational change and adaptability, agile organizations are nimble and light in motion. This increasing agility is a relative trait in various industry segments. For example, while heavy industries like automobile manufacturers may not be considered agile in comparison with a web-based service industry provider, many automobile manufacturers are *more* agile than they were a decade ago.

ICT is both driving and being driven by this global competition. Because organizations are becoming network and IT enabled, they intensify competition in their markets. Because the competition is intensifying, there is growing need for more advanced ICT. Service Oriented Computing (SOC) (Papazoglou and Georgakopoulos, 2003) and Service Oriented Architecture (SOA) are among the major technical advances that have attracted significant and growing adoption since 2000.

SOA has been the subject of some hype. The simple addition of Web services and XML to the application portfolio is not magically transforming organizational technical environments (Erl, 2004). We need a better understanding of the practical implications of principles behind SOA technologies. In this paper, we explore the strategic value of applying SOA for banking institutions, and examine the ultimate impact of the promising technology behind SOA.

Given the relatively recent availability of workable SOA, there are few empirical studies of the organizational impact of SOA adoption projects, especially for industries, such as banks, that are often encumbered with intractable legacy systems. In this paper we comparatively explore two organizations from the same industry but from different countries. We study SOA adoption in a Scandinavian Bank and in a Swiss bank. For purposes of anonymity we have named them “Northern Europe Bank” and “Central Europe Bank”. The purpose of this article is to report the insights into the organizational impact of SOA adoption in these two cases.

The remainder of this paper is organized as follows: The next two sections discuss the architectural challenges today in banking and the strategic concepts behind SOA and Web services in greater detail. We then outline our research method and present the two case studies. We then discuss the two banks’ different approaches to SOA and conclude with some of the key attributes of a SOA.

2 ARCHITECTURAL CHALLENGES IN BANKING

Banking provides a particularly interesting setting in which to study SOA adoption. Banks are struggling with heterogeneous legacy systems that are difficult to change and integrate. Deregulation as well as shrinking profit margins and rapid growth in online banking are impacting the global financial market and requiring rising agility. The traditional restrictive banking laws have resulted in very fragmented industry IT infrastructures, which must now be adapted to meet the new demands for customer-driven, specialized and highly automated virtual banking. Banks besotted with antiquated ICT architectures now confront rising demands for flexible and cost-effective information systems (IS) architectures. The ability to acquire a new ICT architecture is becoming a key competitive factor to support the new financial business drivers in banking.

These older architectures complicate integration of enterprise applications because the underlying elements have created "closed" architectures. Closed architectures restrict access to vital software and hardware configuration and force organizations to rely on single-vendor solutions for parts of its ICT (Mazursky, 1989). Banks are struggling with a range of applications running on different Cobol, .Net or Java platforms. The more-or-less closed architectures impede the banks' ability to offer new and integrated financial products, merge with competitors or consolidate replicative operations.

For example, in our Scandinavian case study, it is no trivial task to offer a range of different financial services as a consolidated customer package. The Northern Europe Bank had developed a product for private customers that collected a wide range of financial products into one package. The challenge with this kind of integrated products is to have integrated business and IT processes. Without integration it would take two hours for a sales representative to set up the new package for one customer. It now takes ten minutes using their integrated enterprise applications.

2.1 Four Challenges

Strategically, banks are facing four major challenges in relation to architecture.

First they need to enable *application integration*. For a general discussion about the meaning and significance of integration refer to (Markus 2000). Architectures for integration in banking are discussed in (Pan and Viña, 2004). Modern banks increasingly have to deal with a plethora of complex products and services managed by a high number of applications and systems, often operating on different platforms. In a typical banking setting, several layers of historically separate, but relevant IT artefacts accumulated over time, making it difficult to get such different systems to interact together (Pan and Viña, 2004). Therefore, IS development in modern banking requires a more and more significant effort of application integration to bridge, connect and incorporate new functionalities in the existing IT environments. For the above reasons, application integration is assuming a growing strategic relevance in banking.

Second, banks face a need for enabling *value reconfiguration processes* (Seifert and Wimmer, 2001; Homann et al., 2004). "To decrease costs and simultaneously enhance customer utility, banks are increasingly focusing on their individual core capabilities while exploring different sourcing options for non-core capabilities. Consequently, they are disaggregating their value chain into independently operable functional units. As communication capabilities reach higher levels of performance and reliability, these functional units are combined across corporate borders" (Homann et al., 2004, p. 34). This way, value chains are disaggregated and recombined into *value networks*. Such processes are connected to environmental complexity and instability. Environmental complexity demands value networks because these better serve complex market demands than value chains. Environmental instability demands value networks because value chains are cumbersome against unexpected market demands or other emergent opportunities.

Third, there is a need for ensuring *value preservation after M&A*. Partnerships, mergers and acquisitions reconfigure value, sometimes with significant value destruction (Bruner, 1999; Smith and O'Neal 2003). The dominant organization imposes its own management, culture, and practices seriously compromising the target organization's culture, knowledge, and distinctive capabilities. When the target organization's old systems are brutally replaced, knowledge and practices embedded in these systems are lost.

Fourth, there is a need for *more agile forms of IS development*. Banking is weighted with regulation and security, and consequently often conservative about IS development (ISD). Environmental instability leads to more organizational emergence and subsequently to more frequent system changes. Continuously redeveloping systems means moving from traditional development towards more agile ISD approaches (Truex et al., 1999; Baskerville et al. 2001; Baskerville and Pries-Heje, 2004).

The former and the latter aspects (respectively, application integration and agility in IS development) emerge with particular evidence in the two case studies observed, as discussed in section 7.

3 SERVICE ORIENTED ARCHITECTURE

From a strategic business perspective, SOA is built around the notion that services map to business functions (Datz, 2004). Within computing, the term architecture can refer to lower levels of abstraction, such as a particular computer's or family of computers' internal architecture. But SOA generally refers to organizational ICT architecture, meaning the unifying or coherent form being used to organize and design the construction, selection and interconnection of an organization's hardware, software and communications assets. In this way, SOA is an architectural style for building loosely coupled distributed systems that deliver application functionality as services to be used for end-user applications (Ho, 2003).

One of the primary sources of strategic value for SOA is its role of enabling technology for application integration. Enterprise Application Integration (EAI) is a business computing term for plans, methods, and tools aimed at modernizing, consolidating and coordinating the overall computer functionality in an enterprise (Lee et al, 2003). The SOA concepts represent a shift in architectural styles from a company-centric development approach that requires specific technologies to a development paradigm focusing on interoperability and open standards.

Technically the ideas behind SOA build on SOC, a computing paradigm in which services are the fundamental elements for developing applications (Papazoglou and Georgakopoulos 2003). In this paradigm ICT products are comprised of references to external components for the performance of various kinds of service. SOC can be used to wrap a service oriented facade around closed-architecture legacy systems, thereby converting these to be compatible with more open architectures. A SOA framework can take advantage of these wrappers to enable more flexible and cost-effective integration of financial services that continue to use legacy systems, especially in distributed settings (Datz, 2004). SOA promotes well defined, published, and discoverable interfaces that deliver reusable application functionality for distributed systems, i.e., where services are invoked over a network. In this way, SOA is more than a traditional ISD method because it embraces business process modelling and enterprise architecture as well as object-oriented design and distributed systems (Ho, 2003). In other words, SOA describes the overall physical design framework of IS layers, functionality and roles of such services while SOC delivers the underlying computing foundation

SOA is characterized by the following fundamental features: (1) It is based on services that can be readily integrated, (2) it is based on standards, (3) it is available on multiple platforms, (4) it provides self-contained (hence, loosely coupled) services, and (5) it incorporates and presupposes a contract that specifies the functionalities offered and at the same time, guarantees that they are replicable (Meredith and Bjorg, 2003). In particular, the fundamental features that guarantee reusability are the existence of a contract and the fact that the services are self-contained.

3.1 Web services

The Web services standard is an important proposal for the implementation of SOA. Web services is a component-based software standard similar to DCOM or CORBA, except Web services is XML-based and therefore web-enabled. Web services is a paradigm specifically conceived to allow systems running in different environments to interoperate via XML and other web standards (Arsanjani et al., 2003). The Web services standard is particularly suitable for application integration because it inherits reusability from component-based technologies like CORBA and DCOM, and it inherits extensibility and web infrastructure from XML.

4 RESEARCH METHOD

Given the exploratory research question, we selected an in-depth case study approach relying on data triangulation (Yin, 1994). Case studies facilitate multi-perspective analyses that consider not just the voice and perspective of the actors, but also those of the relevant groups of actors and the interaction between them (Tellis, 1997). This analytical depth leads to a holistic understanding of cultural systems of action (Feagin et al. 1990; Tellis 1997), providing the insight that satisfies exploratory questions.

We collected data by using both semi-structured interviews and document reviews, two important forms of evidence collection that satisfy Yin's (1994) principles of data collection: (1) multiple sources of evidence, (2) a case study database, and (3) a chain of evidence. Our mode of analysis used Yin's "pattern matching," a comparative analysis involving the establishment of non-equivalent dependent variables as a pattern. Each case has a variety of outcomes that we analyse to develop common explanatory patterns across the comparative case data. This qualitative research design supports the study of complex, dynamic social phenomena that are 'both context and time dependent' (Orlikowski and Baroudi, 1991). In our results, we engage in a class of generalizability that Lee and Baskerville (2003) call type "EE", the generalization of data to a measurement, observation, or other description.

We developed access to two banks, each in a different area of Europe. Our selection criteria limited us to banks with active SOA development projects. We sought different areas in order to distinguish comparable aspects of banking as an industry from contrasting aspects a regional or national culture. We studied projects in a Swiss bank an in a Danish bank. Two of the authors were participated in architecture related projects in the Northern Europe Bank and three of the authors had the same opportunity in the Central Europe Bank. Participant observation is common in case studies, but raises limitations because of a trade-off between its "unusual opportunities for data collection" (Yin, 1994) and the "potential biases" produced by close involvement. We minimized this bias through a full-day workshop in which all of the authors presented findings and non-participants scrutinized patterns proposed by participants.

5 CASE: CENTRAL EUROPE BANK

Business Model: Central Europe Bank, located in Switzerland, is part of a large financial group that operates both at the national and international level. The core business of Central Europe Bank is private banking (i.e. banking services, including lending and investment management, for wealthy individuals), where it is one of the market leaders. The typical customer of Central Europe Bank has a substantial wealth. The bank's distinctive values include discretion, privacy, professional service, personal trust, and individual relationships. Specialized professionals provide the core business services.

IT Strategy: One of the managers described well their approach to IT in an interview: "In IT adoption we are a fast-follower." Their attitude towards IT is quite conservative (e.g. the company's information system is managed internally by the IT Division), but, as a market leader, they pursue new opportunities offered by innovation, on the condition that it involves a low level of risk. Central Europe Bank displayed a general preference for well-defined and orderly rules (that often characterise the Swiss national culture). They set up an Enterprise Architecture Integration organizational unit, and started heavily investing in new IT resources (like middleware and infrastructure software) aiming at building a corporate-wide EAI architecture. This IT strategy was somewhat incongruous with other bank values: those of prudent, risk-averse Swiss bankers.

Project Description: A new CIO was attracted by the potential benefits of EAI and SOA, but he was also aware of the related high level of uncertainty, given the still immature technology, and the

competence gap due to lack of experience. The need for a new cashier management system (CMS) provided a low risk, “test bed” project opportunity for developing organizational expertise. This bank’s CMS is low risk because teller operations are typically minimal. The teller is not involved in portfolio management transactions, which is the bank’s core business. Moreover, customers rarely approach tellers for transactions like bill payment, cheque encashment, or cash withdrawals. Mostly tellers deliver a limited number of complex transactions of high average amount, along with other services like safe deposits management and non-cash items collection. Cashier management does impact the bank’s image, but it is not a mission-critical activity. Development of a new CMS was less risky than other operational process areas that were closer to the bank’s core business. Still, as it will become evident here below, the CMS project had a relevant strategic value, because it contributed to enhance organizational “fitness for future” (Thompson 1967, p.84). The organizational learning stemming from the CMS project contributed to achieve a more agile IS development process and a higher degree of application integration, based on Web services and Service Oriented Architecture. The consequent potential and actual business benefits are discussed here below.

Two years earlier, an internal project team had already conducted a thorough system analysis using a waterfall-style approach. The low priority of CMS had delayed its development until the new CIO envisioned it as a low risk learning opportunity to acquire the necessary SOA competences.

CMS development was partly outsourced to a vendor highly specialized in software development for financial and retail banking systems. Actually, the relationship was closer to a partnership than to traditional outsourcing (Ye and Agarwal 2003). Trust had a central role in determining project success, according to several interviewees both in the bank and the vendor. The project team included specialists from both the bank and the vendor. After a short revision of the existing analysis, extensive prototyping was then started. System prototypes were used as “boundary objects” (Carlile 2002) in order to incrementally review and test parts of the system with users, and also to discuss and negotiate new requirements.

In September 2002 a first prototype was delivered to six tellers and was used to clarify and refine the requirements that were later incorporated in subsequent releases. The full testing phase involved the chief cashier (head teller) in one branch for three months. In June 2003 the system began operation in a second branch. The old system was discontinued in December 2003.

Project Objectives: The CMS project had three main objectives: First, to introduce a new philosophy of service that innovated new organizational practices for cash management. Second, to leverage the recently introduced EAI through a pilot Web-services-based SOA (WS-based SOA thereafter) application, at an acceptable level of risk. Third, to build new competences and knowledge in the SOA field that will enable the bank to continually redevelop its systems as markets demand.

Project Challenges: The CMS project expected three main challenges: First, the new application philosophy would be a hurdle for developers experienced in older software development paradigms. The introduction of the new development paradigm would be a major technical challenge because it had to be tested within the Central Europe Bank. Few comparable solutions had been developed elsewhere in the banking market limiting previous experience and availability of competences. Second, the innovative software development tools such as Web services on a Microsoft .Net platform had, at the time, just entered the market. Third, the integration between the CMS and the legacy systems through EAI and SOA was complex. The bank’s pre-existing CMS was developed in the ‘80s in a hybrid environment with multi-terminal and LAN, OS/2, CICS and Cobol.

Project results: The integration of the system with the bank’s legacy system was achieved via EAI and SOA. The EAI unit worked in strict coordination with almost every functional unit belonging to the IT Division. The new CMS replaced the pre-existing client/server application with a multi-tier, web based system. The end-user (i.e. the bank teller) interacts with the application via a web-browser. The user interface was implemented in Microsoft ASP .Net technology and the business logic was implemented as Web services.

The new CMS enables a new way of working and interacting with the customer. It was decided to overcome the traditional "transaction based" approach in favour of a more sophisticated "operational approach". Within the CMS, a "service operation" is a set of single transactions. The desired service operation is assembled at the counter on the basis of customer's requirements. The whole operation can be traced on a step-by-step basis; alternative solutions (e.g. potential revenues across exchange rates and service charging) can be simulated and the effects of certain actions can be interactively discussed with the customer before actually launching the transactions. Transactions that meet customer's objectives are finally executed through a single command.

SOA Business benefits: Some of the key business benefits of SOA in the Central Europe Bank are connected to the shift towards an enhanced degree of *application integration*. The integration approach based on SOA, compared with a more traditional EAI approach, has significant added value. It enables decomposition and recombination of new and pre-existing functionalities into an overall service oriented reference model, The reference model, once implemented, hides much of the complexity of technical implementation and it masks the peculiarities of legacy software modules. For instance, in the CMS system, whenever a cash amount is involved, the number of bills exchanged should be defined. The legacy system, on the other hand, is able to deal only with a total amount for each currency. The new service, based on a reference model that is closer to the teller view, translates, whenever necessary, the number of banknotes into a total amount, masking the underlying legacy transaction and hiding the complexity of functionality extension.

Web services business benefits: According to several interviewees, the extensible, WS-based SOA was a key technological enabler for *more agile forms of IS development*: reusability was leveraged in order to discover and implement emerging functionalities in a radically new and partially undefined application concept. Web services, as an unexpected added value, were able to deliver a platform on which the features implemented by a single service could easily be adapted and modified without compromising the functionality and the stability of already developed/deployed "consumers" components. Fast development cycles were observed, even under Central Europe Bank's stringent quality standards.

The WS-based SOA was able to deliver a good level of flexibility without the need to freeze service specifications at any stage of the development; a degree of adaptability and rethinking could be accepted even at late stages during the implementation phase without compromising previous developments. Reusability of business components (i.e. of Web services) has already been demonstrated by subsequent projects in the Central European Bank, where new functionality was added to the original business components, e.g. by adding new methods or by adapting the existing ones. Very few Web services could be reused exactly as originally implemented: most of them required minor changes. Still, changes in the original Web services did not affect their consumers. Software modules, developed on the basis of the previous version of a business component, were usually not affected by extended functionality.

Experience gained in the CMS project showed that, in order to be reused by different consumers, a service component often needs to be extended. To this aim, Web services extensibility is a key enabler, unleashing the significant business benefits of reusability.

Moreover, at the Central Europe Bank, a new software development team was formed, that is rapidly acquiring skills and capabilities on .Net and Web services technologies. More than half of new internal IS development efforts are now based on WS-based SOA. The CMS project, and especially the partnership with the external vendor, had a key role in launching and nourishing an organisational learning process with high potential strategic impact on IS development practices.

6 CASE: NORTHERN EUROPE BANK

Business Model: With around 18.000 employees and more than three million private customers in Denmark, Norway and Sweden, Northern Europe Bank is today one of the largest banks in Northern

Europe. The bank offers a wide range of financial services ranging from insurance over traditional banking to real estate sales, with around one million home banking customers using a range of advanced financial services online. There are relatively few large providers of financial services in Scandinavia and customer loyalty has traditionally been very high. But the growing international competition that has emerged as a consequence of the open market in European Union and the technological innovations that facilitates the ”time-space distanciation” has challenged Northern Europe Bank’s dominant position.

IT Strategy: Northern Europe Bank’s IT strategy is today focused at aligning the business and technology environments to match the request for change ability and reusable business functionality. In terms of its value configuration, Northern Europe Bank distinguishes itself from many other companies that have been through a long series of acquisitions and mergers by consolidating its entire infrastructure on the same IT platform. Northern Europe Bank calls this strategy “one bank, one system” meaning that all divisions within the company work on the same IT-infrastructure. They developed a strategy called “first prover”. It means that they want to be among the first companies to reap the benefits from new ideas and technologies in the IT world to gain flexibility and cut IT spending.

Northern Europe Bank’s IT architecture was developed as a network centric architecture with a range of separate self contained "silo" application systems that were oblivious to other systems within the bank. The architecture was not only a roadblock to one-bank-one-system application integration strategy, it was a roadblock to the continuous redevelopment implied by their first prover strategy. To overcome these enterprise integration challenges, they championed a company wide SOA strategy beginning in 2002.

Project Description: Northern Europe Bank started running pilot-projects using component based development methods in the middle of 2001 and early 2002. At the end of 2002 the SOA concepts were introduced based on the limited theory available at that time and the experience gathered by the bank’s chief architect from participating in different international forums with leading software vendors. The management board has adopted the SOA strategy and a new company wide development model – based on the new SOA concepts – replaced Northern Europe Bank’s existing development model in the middle of 2002.

Project Objectives: Become a first prover of SOA as a means of integrating silo banking applications into a cohesive, integrated, enterprise information architecture.

Project Challenges: A good illustration of the complexity and redundant functionality that existed in the old IT architecture is the customer “packages” mentioned in the introduction. These packages were a combination of different financial products and they achieved great commercial success with over 300,000 customer packages being sold. From the sales rep’s perspective it was easy to just take the existing offerings and combine them into new packages. The problem was that the IT systems did not support the combination of the different offerings. Each time a customer bought a customer package, a sales representative had to access each individual bank product’s interface included in the package, cut data out from a Word document and write them into the customer’s individual package. Afterwards the customer signed the agreement and the sales representative finally created the package as a combination of all the systems. The ‘one bank, one system’ philosophy to some extent saved the bank because it was fairly easy to automate the business processes in the back office. This example illustrates the integration challenges that made the IT-department at Northern Europe Bank reconsider its current IT architecture.

Results: For Northern Europe Bank the move towards SOA is seen as a paradigm shift in the way they develop application systems. The implementation of new systems based on the SOA development principles has been accomplished as incremental steps while a service integration layer facilitates the integration of the many legacy systems in the bank. The in-house build broker-like service integration layer is based on XML and MQSeries technologies facilitating the publication of services for service providers and the lookup of service interfaces for service consumers. The service integration layer has

a many-to-many infrastructure, which makes it possible for applications on the same platform (e.g. Cobol) to communicate directly with each other via a local service registry. At the same time services across different platforms (e.g. Java and Cobol) can find and invoke each other's services across the company using their own service registry because the registries are replicated across the service layer. Because the services have well defined interfaces as their fundamental attribute, the integration with other services is conducted with no regard to the physical implementation of the services. The bank emphasises that one of key advantages of the broker-like service integration layer is that it intelligently (based on build-in knowledge) optimises which call mechanism that is the best for calling and receiving a service request.

The bank's external integration with customers and suppliers in the financial value chain has also been improved as a result of the SOA strategy. Even though the bank's SOA strategy has been internally focused at first, Web services can be used both in internal systems and by external partners. Over time the bank expects to sell and distribute hundreds of thousands of service packages. The Web services are exposing their service interfaces and operations in the service integration layer which offers great extensibility advantages. In other words, the wall-to-wall SOA strategy has build a flexible and cost-effective IS architecture for Northern Europe Bank that makes it less important whether services are internal or external.

7 DISCUSSION

Organizational Needs: Our analysis of these cases indicates several common organizational needs that appear to be growing for the two banks we studied. These needs are connected to rising demands for application integration and continual redevelopment in the banking sector. The answers analysed here were based on SOA and in one case on Web services; indeed, the Northern Europe Bank case illustrates that it is not necessary to use Web services internally to achieve an extensible architecture.

Table 1 and 2 below visualize how the SOA approaches in the two banks could respond to such demands, facing the four strategic challenges discussed in section 2. SOA potential and achieved results for the two projects are contrasted.

Northern Europe Bank **(SOA based on EAI integration platform+pure XML)**		
SOA Strategic Challenge	**SOA Potential result** **(***Null, Limited, Significant, High***)**	**SOA Achieved result**
Application integration	***High***: potentially no limits to extending Web services based AI to other applications, thanks to SOA	***Significant***: More than 1000 operations in production. E.g. the complete customer portal works with a SOA interface to the systems behind.
Value reconfiguration	***Limited***: pure XML has no standard architecture providing a standardized approach to value reconfiguration.	***Null:*** No actual business based on value reconfiguration at the moment.
Value preservation in M&A	***High:*** Northern Europe Bank active in M&A.	***Significant:*** The way SOA is implemented allows for better value preservation in M&A. This is actually being demonstrated at the time of writing – after the Northern Europe Bank bought two other banks
Agile IS development	***High:*** Developing new systems with existing services has the potential of being very agile. However, pure XML approaches may have a lower potential for standardized component based reuse than WS approaches.	***Significant:*** The gradual reconfiguration of the system portfolio towards SOA is providing progressively higher levels of flexibility. The current development model embraces and utilises the potential for agility that SOA provides.

Table 1: SOA strategic challenges and results in Northern Europe Bank.

Central Europe Bank **(SOA based on EAI integration platform+Web services)**		
SOA Strategic Challenge	**SOA Potential result** **(***Null, Limited, Significant, High***)**	**SOA Achieved result**
Application integration	***High***: Service Oriented architectures are a step towards modular design of business functionality, providing a method to reduce complexity and decrease the flexibility of coupling internal and external business units (Homann et al. 2004, p.34).	***Limited*** (compared to the whole application portfolio): SOA application integration model is adopted, at the moment, by the Cash Management System (CMS) and by another recent project (Portfolio Management System). Still, IS development strategy is now heavily based on SOA (see below).
Value reconfiguration	***Significant:*** Web services standard architecture provides a more standardized approach to value reconfiguration compared with pure XML. Drawbacks due to immature technology and overall complexity (WS security, performance, management…)	***Null:*** No actual business based on value reconfiguration at the moment. Indeed this opportunity had been taken into some consideration in an early stage of the CMS project, then abandoned. This is in line with the CE Bank conservative, risk adverse attitude and culture.
Value preservation in M&A	***Limited:*** WS-based SOA would facilitate value preservation, but Central Europe Bank was not active in M&A recently.	***Null:*** In the last few years no M&A operations.
Agile IS development	***High:*** Due to extensibility and reusability, enabled by WS-based SOA, as discussed above.	***Significant :*** Web services IS development team actually operating now. More than half of new internal IS development efforts are now based on agile IS development and WS-based SOA.

Table 2: SOA strategic challenges and results for Central Europe Bank.

SOA Business Drawbacks: The Northern Europe Bank case showed that the service concept was difficult to define in practice. They defined services in a way that is not directly in sync with the general definition. In the concrete they left out business processes. However, the way services were defined is aligned with the way WS uses the term. The problem suggests that companies adopting a SOA strategy may struggle when defining the difference between classes, components and services. Nevertheless good *application integration* and *value preservation* was achieved.

Central Europe Bank faced problems because its overall enterprise architecture strategy was still at an early stage. The bank had to train a new person to manage the CMS application under the complex new service oriented architecture.

Both bank cases illustrate how it is difficult to leverage the business benefits from a SOA strategy because aligning business and technology is still difficult with SOA. The alignment difficulty remains because the business community is not automatically included in the new development process. Indeed, the complexity makes it more difficult to include business managers in the development of SOA landscapes by adding new layers to the incomprehensibility of IT.

While SOA was applied differently in the two banks, architectural extensibility was a strategic feature that was both sought and achieved in the organisations' ITC.

8 IMPLICATIONS

The successful development of an extensible SOA by the two banks, along with the rather dissimilar nature of their approaches to this development, suggests a number of deeper insights that we can draw from the cases.

Agile systems development is not essential to agile organizational development. The stringent demands for security and reliability in banking systems are not conducive to the use of many of the agile development approaches. These approaches focus on development speed rather than product quality (Baskerville et al., 2001; Baskerville and Pries-Heje, 2004), and as such are typically unacceptable for banking systems. Each of the banks developed extensible, service oriented architectures, and did so for the most part with their usual development strategies while adopting new frameworks and tools. A still unclear evolution of traditional development practices may be under way with the adoption of SOA and Web services, enacting a more agile organizational development while attaining high quality, security and reliability standards.

Service Oriented Architecture is a means rather than an end. The context of both case settings was driven by organizational strategy. The move to the development of SOA at Northern Europe Bank was part of a strategy of aligning business and technology for agility. While something of an experiment, the development of SOA at Central Europe Bank responded to a strategy of innovation at calculated risk with a goal of retaining market leadership. Both organizations approached SOA as a means to achieve their strategies. In this sense, SOA is only one ingredient in a recipe for a much larger meal.

SOA builds affinities for recognized principles of software development. The architecture inclines those developing software within this architecture toward recognized principles of sound and professional software development. SOA is a naturally evolutionary architectural style through implementation independence and an extensible, open framework. In the Northern Europe Bank case, they achieved logical independence from the physical implementation of the code, and they realized an unexpected potential to offer services externally. In a different way, Central Europe Bank opened a closed architecture by wrapping Web services around a legacy system, converting physical dependence to a logical independence. These natural outcomes satisfy at least two central principles in software development: (1) the aim of building with and for reuse, and (2) controlling complexity with multiple levels of abstraction (Bourque et al., 2002).

SOA invokes an unfamiliar concept that raises barriers to adoption. The service concept is unfamiliar to many experienced system developers. Like the transition to object-oriented frameworks a decade ago, both Northern Europe Bank and Central Europe Bank encountered steep learning curves. Developers grappled with the essential differences between the fundamental concepts of SOC that lies at the heart of SOA. For example, in Central Europe Bank, a new IS development team was formed as a result of an organizational learning process enacted by first experiences with SOA and Web services. Furthermore, new skills were required also in tuning and managing applications based on SOA.

References

Arsanjani A., Hailpern, B., Martin, J. and Tarr, P. (2003), Web Services: Promises and Compromises, *Queue*, 1 1), March., pp. 45-48.

Baskerville, R. and Pries-Heje, J. (2004), Short cycle time systems development, *Information Systems Journal*, 14, pp. 237-264.

Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B., and Slaughter, S. (2001), How Internet Software Companies Negotiate Quality. *IEEE Computer*, 34(5), pp. 51-57.

Bruner, R. F. (1999), An analysis of value destruction and recovery in the alliance and proposed merger of Volvo and Renault. *Journal of Financial Economics, 51*(1), pp. 125-166.

Bourque, P., Dupuis, R., Abran, A., Moore, J. W., Tripp, L., and Wolff, S. (2002), Fundamental principles of software engineering - a journey. *The Journal of Systems and Software*, 62, pp. 59-70.

Carlile, P. R. (2002), A pragmatic view of knowledge and boundaries: Boundary objects in new product development, *Organization Science*, 13, 4, pp. 442-455.

Datz, T. (2004), Architecture – What You Need to Know About Service-Oriented architecture", In: CIO Magazine - Framingham., 17 (7) , pp. 78-88.

Erl, Thomas (2004) *Service-Oriented Architecture – A Field Guide to Integrating XML and Web Services*, Prentice Hall PTR, New Jersey.

Feagin, J., Orum, A., and Sjoberg, G. (eds.), (1991), *A case for case study*, University of North Carolina Press, Chapel Hill, NC.

Ho, H. (2003), What is Service-Oriented Architecture? O'Reilly XML.com, http://www.xml.com/pub/a/ws/2003/09/30/soa.html.

Hagel III, J. (2002), *Strategies for Achieving Profits Today and Growth Tomorrow through Web Services.* Harvard Business School Press, Boston.

Homann, U., Rill, M., Wimmer, A. (2004), Flexible value structures in banking, *Communications of the ACM*, 47(5), pp. 34-36.

Lee, A. S., and Baskerville, R. L. (2003). Generalizing Generalizability In Information Systems Research. *Information Systems Research, 14*(3), pp. 221-243.

Lee, J., Siau K., Hong, S. (2003), Enterprise integration with ERP and EAI. *Communication of the ACM* ,46(2), pp. 54-60.

Markus, M. L. (2000), Paradigm Shifts—e-Business and Business/Systems Integration, *Communications of AIS*, 4(19), pp. 1-44.

Mazursky, A. D. (1989), Closed-Architecture Systems. *The Journal of Information Systems Management*, 6(3), pp. 73-76.

Meredith, L. G., and Bjorg, S. (2003), Contracts and types, *Communications of the ACM*, 46(10), pp. 41-47.

Orlikowski ,W. and Baroudi, J., (1991), Studying information technology in organizations: research approaches and assumptions. *Information Systems Research*, 2(1), pp. 1-28.

Pan, A. and Viña, Á., (2004) An alternative architecture for financial data integration, *Communications of the ACM*, 47(5), pp. 37-40.

Papazoglou, M.P., and Georgakopoulos, D. (2003), Service Oriented Computing: Introduction, *Communications of the ACM*, 46(10), pp. 25-28.

Seifert, F., and Wimmer, A. (2001), Towards networked banking: the impact of IT on the financial industry value chain, *Proceedings of the European Conference on Information Systems (ECIS)*, Bled, Slovenia, pp. 474-84.

Smith, K., and O'Neal, E. (2003), Bank-to-Bank Deals Seldom Add Value, *ABA Banking Journal*, December, pp. 7-10.

Tellis, W., (1997), Introduction to case study, *The Qualitative Report [On-line serial]*, 3(2). http://www.nova.edu/ssss/QR/QR3-2/tellis1.html.

Thompson, J. D. (1967), *Organizations in Action*, McGraw-Hill, New York. New edition: 2003, Transaction Publishers, New Brunswick.

Truex, D. P., Baskerville, R., and Klein H. K. (1999), Growing Systems in Emergent Organizations, *Communications of the ACM*, 42(8), pp. 117-123.

Ye F. and Agarwal R. (2003), Strategic information technology partnerships in outsourcing as a distinctive source of information technology value: a social capital perspective, *Proceedings of the International Conference on Information Systems (ICIS)*, Seattle, pp. 304-15.

Yin, R. (1994), C*ase study research*. *Design and Methods*, Sage Publications, Thousand Oaks.