A Conceptual View of Enterprise Resource Planning Systems as Services

Nicklas Holmberg and Björn Johansson(✉)

Department of Informatics, School of Economics and Management,
Lund University, Ole Römers Väg 6, Lund, Sweden
{nicklas.holmberg,bjorn.johansson}@ics.lu.se

Abstract. This paper brings forward a conceptual view, based on practical experiences from designing information systems as services. Viewing information systems (IS) as services is beneficial but still an unexplored approach in organizations. The aim of this exercise is to contribute to the knowledge base of IS designers and modelers. In the paper, we present an analysis of Enterprise Resource Planning (ERPs) systems through a conceptual lens of Service Oriented Architecture (SOA). This paper contributes to the debate on viewing ISs as services by presenting a view of SOA-architected ERPs as facilitating to fulfill business needs. This paper is influenced by systems and design thinking, and service oriented IS design. Based on shared promises between SOA and ERP we discuss the question whether SOA or ERP fulfills business needs? The analysis of ERPs from a SOA perspective provides us with the conclusion that the question is not about SOA or ERP but rather to provide SOA architected ERPs. It can be said that by viewing ERPs as services it is clear that the combination of ERPs and SOA could be seen as one way forward when designing ISs that aims at bridging gaps between IS and business e.g., processes and, allowing the business to fuse with IS forming servitized SOA based ERPs.

Keywords: Enterprise Resource Planning · Service Oriented Architecture · Business processes · Business rules

1 Introduction

At the end of 1990's there were a big hype among organizations to implement standardized software packages named Enterprise Resource Planning (ERPs) systems. Implementation of ERP systems was the prize organizations had to pay to compete in a constant emerging market. Despite the fact that a service dominant economy emerged and influenced organizations to be recognized as goods or service dominant, not much was done by dominant providers to design Information Systems (ISs) as services [1, 2].

ERP systems must reflect "reality" because they have profound influence on business processes, the inner workings of a business and thus on the way business runs. Manifesting the idea about; business and IS fusion forming a business oriented IS [3], captures much of the essence in the prerequisite for such reflection. Similar directions are discussed by Hirschheim and Klein [4] and Taylor and Raden [5]. Business owners have limited influence on ERPs design thus, vendor specific standardized software packages emerged as embedded business actors [6, 7].

V. Řepa and T. Bruckner (Eds.): BIR 2016, LNBIP 261, pp. 3–15, 2016.
DOI: 10.1007/978-3-319-45321-7_1

Implemented ERPs, to some extent, do not fulfill the promises that were indicated by vendors making organizations searching for other solutions. One solution presented is Service Oriented Architecture (SOA), and according to Forrester Research SOA penetration is stronger than ever [8].

Viewing IS as services is beneficial but still an unexplored approach for IS in organizations [1]. In addition, thinking of systems as services enables new systems design methods to emerge [1]. Indeed, new IS Development (ISD) methods aim to improve business communication and provide practical routes toward increased relevance of IS in business and society [1].

This paper is influenced from practical experiences of a national research project named VacSam. VacSam is a set of composed digital services shaping a servitized IS as a SOA architected Enterprise System (ES).

VacSam provides unique vaccination recommendations to any foreign child entering Sweden with a purpose to decrease child deaths due to preventable infectious diseases. VacSam exemplifies one of many applicable contexts for the suggested view of ISs e.g., decision support, diagnosis, predictive analytics.

From glancing at SOA it can be said that the conceptual architecture promises to service orient a business by bridging the gaps between IS and business processes permitting business to shape IS, automated through services [9, 10]. From a quick overview of the promises of ERPs it is indicated that ERPs promise to deliver a similar solution. However, if ERPs aim at bridging gaps through service-orientation is not clear. That brought us to explore ERPs from a service perspective, - a conceptual view of ERPs as services.

SOA is used as a lens for the conceptualization and as the architecture providing a service with properties and the suggested view with a concrete ground for explanation of what SOA services are. Because SOA shares promises expressed by ERPs we question whether SOA or ERP fulfills business needs? The view of ERPs and SOA as separate but related entities is more carefully discussed in future sections of this paper organized accordingly:

First, we present and define SOA and the concept of services in SOA. The section thereafter defines ERPs and discusses problematic issues with ERP implementation. The reason for doing so; is to be able to provide an exploration of designing ERPs as services, which is done in Sect. 4. In the final section concluding thoughts on what it means to design ERPs as services as well as giving some directions for future studies in this area is presented.

2 Service Oriented Architecture (SOA)

The presented approach to SOA departs from a none-technical point of view; (1) SOA as a conceptual architecture, (2) SOA manifesto and the basic principles of SOA and, (3) SOA realizing technologies. The purpose is to decrease the risk of putting SOA on a par with e.g., Web-services, one of many SOA realizing technologies [10].

SOA is a conceptual architecture functioning independent from choice of realizing technology [9]. During the last decade, SOA received criticism as an ambiguous buzzword only realizing obsolete application platforms e.g., standardized software packages. In 2007 Gartner [11] predicted less than 25 percent of large companies to manage their SOA projects by 2010.

This paper therefore argues that only realizing obsolete application platforms is not the intention of SOA [9, 10, 12]. Just as different designers have different understandings of different material and its respectively properties, SOA means different "things" depending on whom you ask [13].

Sincere efforts to operationalize SOA have been made. In 2005, Erl [10] established the basic principles for SOA. Eight basic principles could now intrinsically express Separation of Concerns (SoC) and properties for a SOA service. However, it was still unclear how SOA managed SoC in terms of which logic to encapsulate. A few years later in 2009, Arsanjani et al. [9] established the SOA Manifesto. Fourteen guiding principles stressed the importance of maintaining a business perspective in any SOA initiative [9]. To consider shared services therefore became more important than specific purpose implementations.

In 2009, the SOA manifesto, an extended abstract level of SOA, expressing high level business modeling guidelines was set e.g., 'to respect the social and power structure of the organization' [9]. To achieve architecture supporting the SOA manifesto the basic principles for SOA became of profound importance. The eight principles express properties that a SOA service must possess to be recognized as eligible and responsible. Supporting SoC, the basic principles express modularization and encapsulation realized through information hiding, also, commonly known in Object Oriented Programming (OOP).

"Conceptual", -a property of SOA, dates back to the origin of "service". At the time, the non-defined term "service" was and, sometime still is, the reason to the intrinsic confusion of what SOA is.

In the 1930's, the U.S. Department of Commerce's Standard Industrial Classification (SIC) provided a service a code of classification. In the late 1970's Hill [14] provided "service" a definition [2]: "[…] a service is a change in the condition of a person, or a good belonging to some economic entity, brought about as the result of the activity of some other economic entity, with the approval of the first person or economic entity." [14].

Thus, a SOA service changes a condition of a Service Provider (SP), because of an activity, corresponding to a request made by a Service Requestor (SR) to a SP through a transport medium e.g., Internet, with the approval of a SP.

Arsanjani et al. [9] suggests that service-orientation, a term encapsulating: service, frames what "one does". Service-orientation of SOA is then interaction between a SR, requesting a service from a SP, providing a service from a Service Directory (SD). That is similar to how Gustiené [15] stressed the importance of interaction as the base for service orientation which must support principles of SoC.

Then, "[…] Service-oriented architecture (SOA) is a type of architecture that results from applying service orientation." [9]. While, interaction is "[…] Related mutual actions occurring within a shared space of time or place." [16]. Interaction occurs through a transport medium and its direction is no simplistic association but guideposts indicating orientation of interaction in "reality". Then, an SD-listed-service permits peer-to-peer communication between SR and SD with approval of SP. It can therefore be said that service-orientation based on interaction permits a service to become a unit of communication enabling a SR, a SP and a SD, to interact within a shared space on a share time in a known real world direction i.e., SOA depicted in conceptual and data level in Fig. 1 accordingly:

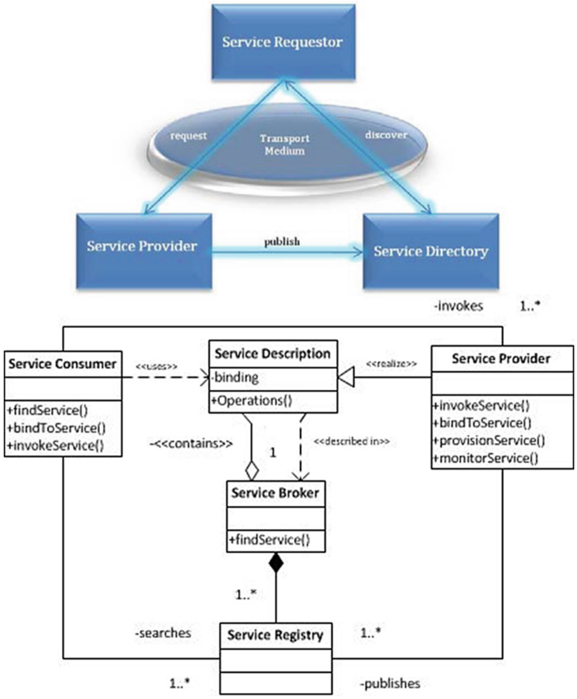

Fig. 1. Basic SOA model in conceptual and data level (use of Erl, 2005)

Industry bodies and e.g., OASIS Group and Open Group created formal definitions of SOA with intentions to facilitate SOA's implicit terminology and reduce its different meanings to: "A paradigm for organizing and utilizing distributed capabilities […]" [17]. According to the SOA manifesto SOA is realized with varying technologies and standards and functions independent from choice of realizing technology [9].

Based on this we define SOA accordingly: SOA is a paradigm that shapes a conceptual architecture, functioning independent from choice of realizing technology, providing abilities to describe a service, its properties and its orientation, for conscious change or design of a service-oriented business.

2.1 SOA Services

Addressing SOA services addresses SOA realizing technology. SOA realizing technology is used for designing a SOA service as a unit of communication realizing interaction. Services responsible for functional components, together shaping a SOA based

ES, could thus be viewed as components equipped with logical boundaries forming composable subject matters. Hence, a service is responsible for the logic it encapsulates independently existing, as an entity of its own right, from other services and ISs.

SOA realizing technologies are: e.g., Simple Object Access Protocol (SOAP), Universal Description, Discovery and Integration (UDDI), Web Service Description Language (WSDL) etc. Such technologies are architectural styles or patterns solving reoccurring known design problems quite contrary to conceptual SOA [10, 13] which rather benefits from being thought of in Alexandrian terms e.g.; design methodology applicable when suitable. Based on that, there is a plethora of SOA realizing technologies putting the basic principles of SOA into use and thereby supporting SoC.

Then the basic principles of SOA are: (1) Services are reusable, (2) Services share a formal contract, (3) Services are loosely coupled, (4) Services abstract underlying logic, (5) Services are composable, (6) Services are autonomous, (7) Services are stateless, (8) Services are discoverable [10] are what shape SOA services representing a part of the physical form of a SOA. Based on the same conditions we argue that functional areas shaping components of ERPs can be designed as services. That is better discussed and explained in Chap. 4.

3 Enterprise Resource Planning Systems

The ERP concept is broad and the market of ERP is dominated by a number of few companies including SAP, Oracle, and Microsoft. However, there are a number of key characteristics that more or less all ERP systems have making them a unique subtype of IS: (1) ERPs are standardized packaged software [18] designed with the aim of integrating an entire organization [19–21]. (2) The ERP ought to cover all information processing needs and to integrate the internal value chain with an organization’s external value chain through Business Process (BP) integration [19] and (3) Provide the entire organization with common master data [22]. From this it can be stated that ERPs have a high impact on organization’s business processes, but as argued by Millman [23] there exist problems, such as, it is either not used or is implemented in the wrong way.

The main problem presented is the misfit between ERP functionality and business requirements. Soh, Kien and Tay-Yap [24] describe this as a common problem when adopting software packages. The problem of “misfit” means that e.g., “Many people feel that the current ERP system has taken (or been given) a role that hinders or does not support the business processes to the extent desire” [7]. Then, ERPs are process-based or at least attempt to be process-based. According to Koch [25] the basic architecture building on a department/stab model as for instance SAP’R/3 makes ERPs not supporting the idea of BPs and thereby not the integration between different departments in an organization. It does not help that the ERP vendor attached some words about BPs onto their ERP if the basic architecture does not support BPs [25].

3.1 Functional Areas of ERP Systems Architecture

ERPs are often described from a functional perspective meaning that the systems architecture mimics a functional organizational description. That implies that each department has its own ERP component. However, the basic architecture of an ERP follows the master data thoughts [22]. Then, functional ERP areas use a unified database. Different ERP vendors describes this in different ways, however, the most common description is to discuss *modules*. Thus, the implementing organization implements a core module and then selects what modules to implement on top of the core module(s). The ERP architecture therefore builds on a vertical organizational description. The implication of that is that horizontal work tasks involving different departments are not clearly described in ERP architecture. Resulting in that users of ERPs could understand the ERPs as not supporting the business process they work with, resulting in a misfit between ERP and users interpretation of how the system fulfill their needs.

4 Designing ERP Systems as Services

ERPs as described above, builds to a high extent on functional areas e.g.; (1) Inventory, (2) Production, (3) Accounting, (4) HR, (5) Delivery, (6) BI, (7) Sales, (8) Engineering, (9) Production Planning, and (10) Purchase. However, the volatile nature of business makes it complex to implement the same ERP in all organizations. Based on the basic principles of SOA, functional areas of an ERP system could be designed as independent components, separated by logical boundaries, designed with the same accuracy as a single class or entity is [10, 26]. That view is based on modularization realized through information hiding and to learn ISD by "doing".

From the description of ERPs it can be stated that it is hard to see if its promises - bridging the gaps between IS and business processes - have been fulfilled. The same can be said about SOA promises. However, it seems that if combing the ideas of SOA when designing an ERP that may be a way forward to fulfill promises from both ERP and SOA.

From this it could be claimed that the desired result is to bridge the gap between BPs and IS so that business shapes IS into what could be described as a SOA architected ERP. The question is then how can SOA improve the design of ERPs? A tentative answer to that question could be that the focus moves from a functional view to a conceptual holistic view, meaning that functions in the ERP, if designed as services, could be seen and provided as applications that could be used in different BPs. In practice this could imply that an organization is permitted to deal with the problem of organizational support with a horizontal supportive IS.

On those conditions, functional areas of an ERP could form components shaped by services eligible to execute in SOA. Based on practical experiences from VacSam, it is shown that by composing digital services a SOA architected IS can be shaped.

Through the design science research initiative it can be said that this conceptual view of ERPs as services became even more evident.

Through the Enterprise Model (EM) (see, [27]) of the VacSam project it can be seen that Fig. 4 depicts that the five sub models of the EM express how business rules integrate in a business and how the business vision model casts the ground for the business strategy and common business goal; *fully vaccinated according to the Swedish vaccination schedule*.

Moreover, the EM depicts that (1) The business rules model (a) triggers the business process model, (b) defines the business concepts model, (c) uses the business resource and actor model and, (d) supports the business vision model. (2) The business process model in turn, requires the business rules model. (3) The business concepts model (a) defines the business rules models. (4) The business vision model motivates and requires the business rules model [27].

In addition, (5) the business resource and actors model, including General Practitioners (GPs), Subject Matter Experts (SMEs) and Vaccination Experts (VEs), is, responsible for the business rules model [27].

Based on that it can be said that Fig. 4 depicts the business models that were digitally transformed and automated through digital services forming applications that could be used in different BPs and shaping the servitized IS named VacSam:

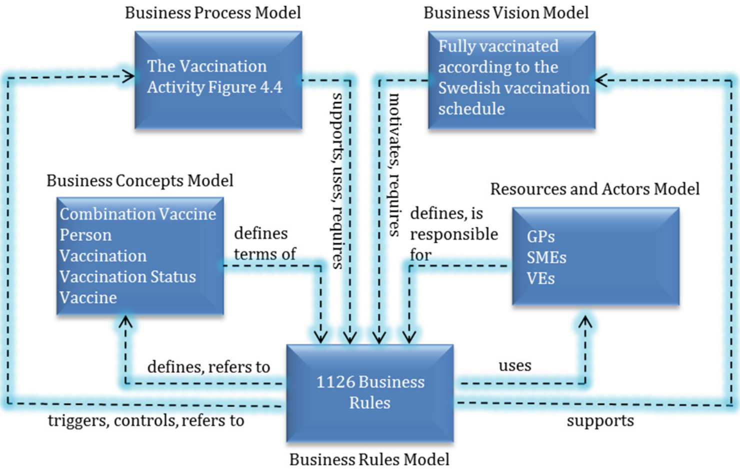

Fig. 2. The business model of VacSam (use of, [27])

If applying this view on the design of ERPs with the aim of integrating an entire organization, a noteworthy detail is that the five models of the business model in Fig. 2 fits e.g., the Zachman framework for Enterprise Architecture (EA) as integrals accordingly:

Hence, the business model of VacSam indicates the desired level of service-orientation and the desired result in the form of a SOA based ES. This is further exemplified through the BRs model and the BP model of VacSam. With the business concepts model in hand BRs it is possible to design well-formed business rules.The business rules model was thus

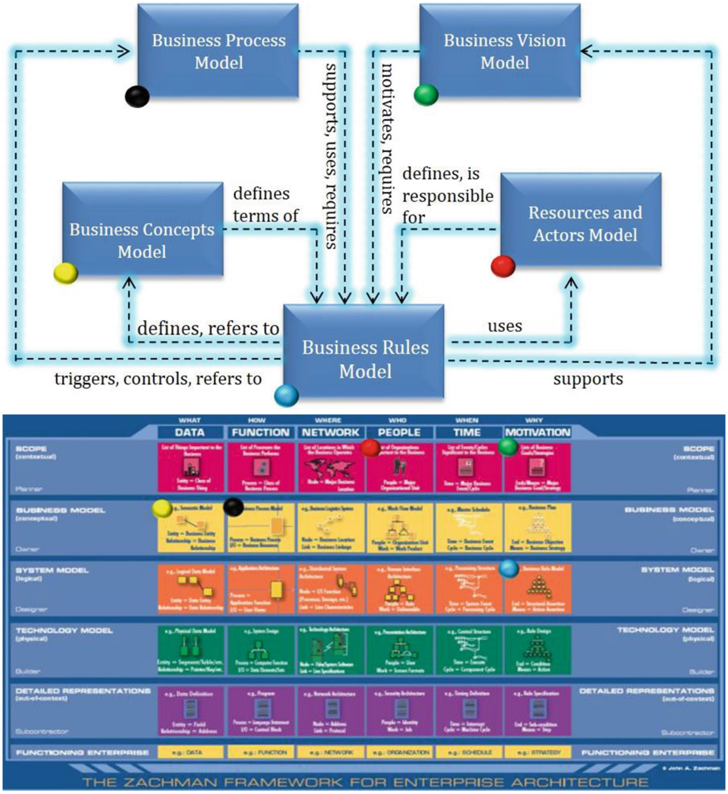

Fig. 3. The business model and its relation the EA framework [27]

constituted by 1126 BRs all designed according to the principles of Business rules approach (BRA) (Fig. 3).

Together the BRs forms business rules packages which in turn shapes decision logic centric SOA services expressing a businesses' "what", only exposing a WSDL according to the basic principles for SOA. Implementing the process logic centric SOA services in imperative JAVA results in an expression of a businesses' "how"

This means that all rule projects including a number of BRs is automated through digital services of their own right. Those decision logic centric services are meant for governing the business process presented in Fig. 4. The business process model in Fig. 4 depicts the process logic explored, extracted and implemented in VacSam:

Together the businesses' "what" and "how" implemented as SOA services support the inner functioning of the business process of Fig. 4. However, the business models per se, could be viewed as archetypes in terms of well-known "standard" ISD models. Thus, it is not the models that are of interest but their combination and service-orientation.

Through SOA, these models are service-oriented and automated hence modular and encapsulated realized through separate digital services kept by the service directory of

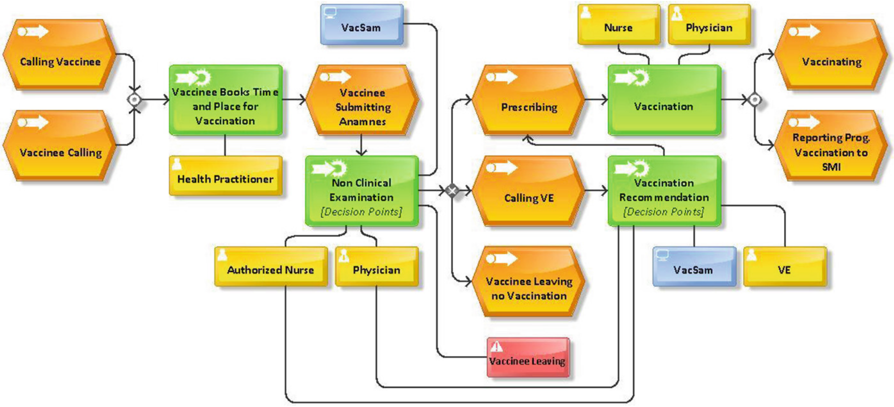
Calling Vaccinee
Vaccinee Calling
Vaccinee Books Time and Place for Vaccination
Health Practitioner
VacSam
Vaccinee Submitting Anamnes
Non Clinical Examination [Decision Points]
Authorized Nurse
Physician
Prescribing
Calling VE
Vaccinee Leaving no Vaccination
Vaccinee Leaving
Nurse
Physician
Vaccination
Vaccination Recommendation [Decision Points]
VacSam
VE
Vaccinating
Reporting Prog. Vaccination to SMI

Fig. 4. The Business Process for VacSam's Process Logic

Fig. 5 below. As a result, each service reflects part of a reality and together the services reflect a reality, a holism i.e., the Swedish vaccination recommendation activity. As a result business processes and IS merges to the servitized ES named VacSam through this SOA perspective:

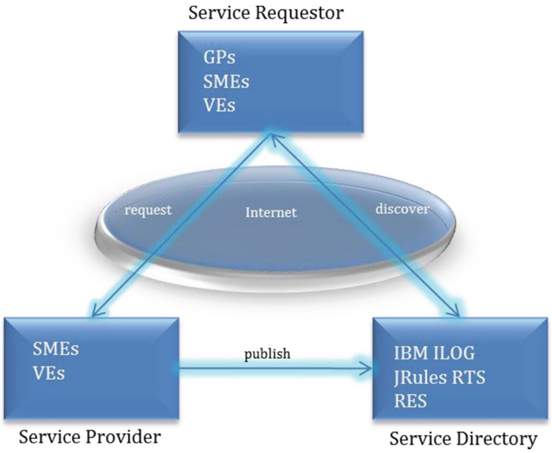

Fig. 5. The Intuitive SOA Orientation Model of VacSam (instance of Fig. 1) (use of, [10])

Figure 5 depicts that SOA has been realized both technically and conceptually. This means that in the VacSam-project SOA was implemented as:

The paradigm informing the design of models and frameworks for a conceptual architecture for interaction, functioning independent from realizing technology, providing abilities to describe a service, its properties and its orientation, for conscious change or design for business service orientation.

The intuitive orientation model, permitting inter-organizational communication, illustrates key actors in the SOA based on the actors-model of the business model. Each service listed in the service directory knows about the other services listed since they all share the same basic principles for SOA only exposing WSDL.

However, VacSam is not strictly an ERP system. On the other hand, from this perspective, VacSam corresponds to a component shaped by about 60 digital services used by GPs for diagnosis. Logically, diagnosis is similar to any Business Rules (BR) governed process reified into a functional IS-area and could most likely be compared to e.g., an accounting-module of a ERP. That is the foremost reason to why we consider SOA a conceptual architecture applicable in a plethora of contexts and not a pattern for routine design. This is made even clearer in Fig. 6.

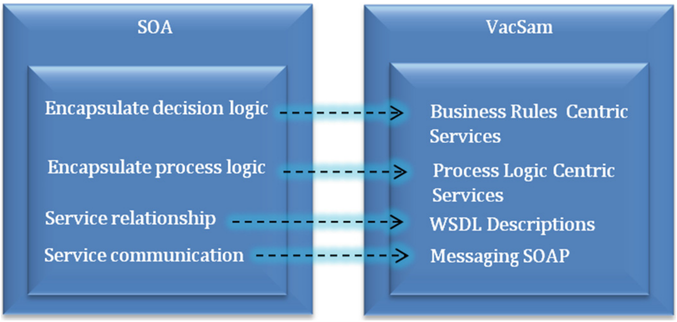
SOA
Encapsulate decision logic
Encapsulate process logic
Service relationship
Service communication
VacSam
Business Rules Centric Services
Process Logic Centric Services
WSDL Descriptions
Messaging SOAP

Fig. 6. How SOA Encapsulates Logic in VacSam (use of, [10])

Figure 6 depicts how SOA encapsulates logic of VacSam. It is clear from the figure that process logic and decision logic are encapsulated by separate species of digital services. However our idea draws on that each functional area of an ERP could be analyzed for extracting its process logic respectively decision logic for implementation into separate digital services. Through service composition the collection of different services could easily replace a component or a functional area of an ERP shaping a truly flexible IS.

Thus, SOA permits to design eligible services and thereby service orienting a business regardless of its character. With profound influence on “how” and “what” business runs, functional areas of any IS must reflect reality to be able to support business processes as a whole thus, bridging the gap between IS and BPs. Therefore it is crucial for the purpose, entitling the being, of an ERP to support decision points in a BP permitting or constraining its execution. Such decision points require business rules why the decision per se could be viewed as the connection between business processes and business rules. Moreover, any ERP must have such business rules as a ground for decisions.

This SOA approach also facilitates managerial IS capabilities in terms of e.g., a shared service repository and the fact that changing one SOA service will not affect the other services because a service is responsible for the logic it encapsulates. This ought to provide better chances for bridging the gap between business processes and IS and the chances for the IS to continue to function in the businesses' active equilibrium through managing change in an organized way.

From research on ERPs we recognize a lack of transparency regarding logic responsibility. What logic that shapes a functional area of an ERP component, is not clear. According to Morgan, [28], Graham, [29] and Von Halle [30] part of business logic shapes decision logic. The other part of business logic is shaped by BPs i.e., process logic [31]. Logic separation through SoC then has profound influence on foundational e.g., alethic logic, and is crucial for IS and ISD success [31]. Tentative results of such SoC is consistent automated business logic [30] -a promise made by Business Rules Approach (BRA) familiar as a support to SOA nowadays [28, 29], but, still an unrecognized approach for native ERP design.

BRs of BRA renounce from expressing "who", "when", "where" or "how" a business rule executes [28, 30, 32] or any temporal aspects managed by operational process logic of an IS [31] as can be seen above. Thus BRs express "what" [32]. BRs then either constrain BP activities from executing or permit them to execute attaining a state why BRs triggers BPs [27]. A BR could therefore be viewed as a definition or a delineation of an aspect of a business [28, 29, 33]. Then, BRs govern BPs [34]. And, BRs are recognized as the operational decision logic of an IS.

Quite contrary, BPs are recognized as the operational process logic of an IS [31]. With that distinction a business's "what" i.e., decision logic expressed by BRs and, "how" i.e., process logic expressed by BPs, becomes transparent and manageable as separate but interrelated components shaped by business objects advocating IS and business alignment [27, 31].

Without separation of logic, decision logic is scattered with process logic and application specific code in the same object, in plural forming components or modules, commonly known as obsolete legacy IS. That makes it hard to recognize what a functional area of an ERP is and which logic each component shaping a functional area of an ERP is encapsulating. Moreover, that would renounce SoC, SOA and BRA by being solely one track minded [26, 28, 29, 34, 35]. Even if there has been some progress, ERPs could be seen as quite far from supporting SoC, since it implies to "consume an elephant" rather than trying to break down problems into smaller manageable pieces, similar to objectification or break down of connections. That directs us to the conclusion that it would be beneficial viewing ERPs as services to a higher extent.

5 Conclusion

We have learnt that, ERP since it is a standardized software package demands adopting organizations to change BPs. However, if viewing ERPs as services, that would not be the case. The view would rather force BPs or their process logic to shape composable services forming one part of an ERP expressing "how", to achieve goals. The other part is shaped by BRs or their decision logic as services expressing "what", to achieve goals. When the two types of services are composed, they can be viewed as a component reflecting a functional area of an ERP providing a desired result similar to those provided by components for e.g., diagnosis or accounting. That would then correspond to a SOA-architected ERP.

Viewing ERPs as services explicitly renounce from any "silver bullet approach" but implies to break down problems into smaller pieces, supporting principles of SoC, and systematically design responsible services, supporting SOA, shaping components reflecting functional areas of an ERP in turn supporting business needs, one at a time. The analysis of ERPs from a SOA perspective provides us with the conclusion that the question is not about SOA or ERP but rather to provide SOA-architected ERPs. By viewing ERPs as services it is clear that the combination of ERPs and SOA could be seen as one way forward when developing software that aims at bridging the gaps between supporting IS and business processes. However, additional empirical research e.g., DSR on designing functional areas as components, shaped by SOA services, supporting important business problems, followed by evaluation, would cast a better ground for interesting future research on the suggested perspective of ERPs. To the best of our knowledge, this perspective of ERPs, BRs and BPs, is an important area for IS research providing more knowledge on how business and IS are independent but intrinsically related entities of today.

References

1. Alter, S.: Viewing systems as services: a fresh approach in the is field. Commun. Assoc. Inf. Syst. **26**(11), 195–224 (2010)
2. Chesbrough, H., Spohrer, J.: A research manifesto for services science. Commun. ACM **49** (7), 35–40 (2006)
3. El Sawy, O.A.: The IS core ix: the 3 faces of is identity: connection, immersion, and fusion. Commun. Assoc. Inf. Syst. **12**(1), 588–598 (2003)
4. Hirschheim, R., Klein, H.K.: Four paradigms of information systems development. Commun. ACM **32**(10), 1199–1216 (1989)
5. Taylor, J., Raden, N.: Smart (Enough) Systems, How to Deliver Competitive Advantage by Automating Hidden Decisions. Prentice Hall, Pearson Education, USA (2007)
6. Melin, U.: The ERP system as a part of an organization's administrative paradox. In: 11th European Conference on Information Systems, Naples, Italy (2003)
7. Askenäs, L., Westelius, A.: Five roles of an information system: a social constructionist approach to analyzing the use of ERP systems, In: Twenty First International Conference on Information Systems, pp. 426–434. Association for Information Systems, Brisbane (2000)
8. Robinson, J.: SOA still kicking, says Forrester (2009). http://www.information-age.com/channels/development-and-integration/perspectives-and-trends/1053022/soa-still-kicking-says-forrester.thtml. (Cited 22 Nov 2011)
9. Arsanjani, A., et al. The SOA Manifesto (2009). http://www.soa-manifesto.org/. (30 September 2010)
10. Erl, T.: Service-oriented architecture: concepts, technology, and design. Prentice Hall PTR, Pearson Education, Upper Saddle River (2005)
11. Gartner. Bad Technical Implementations and Lack of Governance Increase Risks of Failure in SOA Projects. Gartner Newsroom. Press Release (2007). http://www.gartner.com/it/page.jsp?id=508397. (21 October 2011)
12. Erl, T.: SOA Principles, The Service-Orientation Design Paradigm (2008). http://www.soaprinciples.com/p3.php. (04 March 2010)

13. Holley, K., Arsanjani, A.: 100 SOA questions asked and answered. Pearson Education, Inc., (2011)
14. Hill, P.T.: On goods and services. Rev. Income Wealth **23**(4), 314–339 (1977)
15. Gustienė, P.: Development of a New Service-Oriented Modelling Method for Information Systems Analysis and Design. Karlstad University, Karlstad (2010)
16. Alexanderson, P.: Adding Audibility, in Department of Informatics. Lund University, Lund (2007)
17. OASIS Group. Reference Model for Service Oriented Architecture 1.0. (2006). http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf. (06 October 2011)
18. Wieder, B., et al.: The impact of ERP systems on firm and business process performance. J. Enterp. Inf. Manag. **19**(1), 13–29 (2006)
19. Lengnick-Hall, C.A., Lengnick-Hall, M.L., Abdinnour-Helm, S.: The role of social and intellectual capital in achieving competitive advantage through enterprise resource planning (ERP) systems. J. Eng. Technol. Manag. **21**(4), 307–330 (2004)
20. Rolland, C., Prakash, N.: Bridging the gap between organisational needs and ERP functionality. Requirements Eng. **5**(3), 180–193 (2000)
21. Wier, B., Hunton, J., HassabElnaby, H.R.: Enterprise resource planning systems and non-financial performance incentives: the joint impact on corporate performance. Int. J. Acc. Inf. Syst. **8**(3), 165–190 (2007)
22. Hedman, J., Borell, A.: ERP systems impact on organizations. In: Grant, G. (ed.) ERP & Data Warehousing in Organizations: Issues and Challenges, pp. 1–21. Idea Group Publishing, Hershey (2003)
23. Millman, G.J.: What did you get from ERP, and what can you get? Financ. Executives Int. **5**, 15–24 (2004)
24. Soh, C., Kien, S.S., Tay-Yap, J.: Cultural fits and misfits: is ERP a universal solution? Commun. ACM **43**(4), 47–51 (2000)
25. Koch, C.: BPR and ERP: Realising a vision of process with IT. Bus. Process Manag. J. **7**(3), 258 (2001)
26. Dijkstra, E.W.: On the role of scientific thought. In: Selected Writings on Computing: A Personal Perspective, pp. 60–66. Springer-Verlag New York, Inc., New York (1982)
27. Bajec, M., Krisper, M.: A methodology and tool support for managing business rules in organisations. Inf. Syst. **30**(6), 423–443 (2005)
28. Morgan, T.: Business rules and information systems: aligning IT with business goals. Addison-Wesley, Boston (2002)
29. Graham, I.: Business rules management and service oriented architecture: a pattern language. Wiley, Chichester (2006)
30. Von Halle, B.: Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley & Sons, Inc., Computer Publishing, New York (2001)
31. Holmberg, N., Steen, O.: Business process and business rules modelling in concert for e-service design and business alignment. In: 1st International Conference on Cloud Computing and Services Science (CLOSER), Noordwijkerhout, The Netherlands (2011)
32. Date, C.J.: What Not How: The Business Rules Approach to Application Development. Addison-Wesley, Reading (2000)
33. Van Eijndhoven, T., Iacob, M.E., Ponisio, M.L.: Achieving business process flexibility with business rules. In: 12th International IEEE Enterprise Distributed Object Computing Conference, pp. 95–104 (2008)
34. BRG. The Business Rules Manifesto Version 2.0., November 1, 2003 (2003). http://www.businessrulesgroup.org/brmanifesto.htm. (04 January 2011)
35. Ross, R.G.: Principles of the Business Rule Approach. Addison-Wesley, Boston (2003)