



LUND UNIVERSITY

The Service Oriented Business Process and Separation of Concerns - Modelling paradigms for Architectures and Business Processes

Holmberg, Nicklas; Steen, Odd

2012

[Link to publication](#)

Citation for published version (APA):

Holmberg, N., & Steen, O. (2012). *The Service Oriented Business Process and Separation of Concerns - Modelling paradigms for Architectures and Business Processes*. 1-12. Paper presented at The 35th Information Systems Research Seminar in Scandinavia.

Total number of authors:

2

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

The Service Oriented Business Process and Separation of Concerns - Modelling paradigms for Architectures and Business Processes

Nicklas Holmberg and Odd Steen

Department of Informatics, Lund School of Economics and Management, Lund University

nicklas.holmberg@ics.lu.se, odd.steen@ics.lu.se

Abstract. This paper deals with the contemporary concern of business service-orientation. The paper suggests that service-oriented Business Information Systems Development (BISD) permits quality aspects, -“ilities” e.g.; maintainability and modifiability, of BISs and BISD. In addition, this paper departs from viewing a Business Information System (BIS) as services and suggests Separation of Concerns (SoC) as a prerequisite for achieving a well formed digital ecosystem. Besides knowledge transfer the purpose of this paper is to indicate the importance of architecture incorporating SoC when service-orienting a business with the contemporary Service Oriented Architecture (SOA). By realizing business logic service-orientation in a nationwide research project our conclusion indicates that separate implementation of decision logic and process logic provides two different species of digital services implementing SoC. Consequently, responsible and eligible digital services become deliverables, service-orienting a part of a business representing vital quality aspects of ISD. The viability of separate digital services ensuring SoC facilitates the well-formed digital ecosystem i.e. the service oriented business process and business logic reusability and modifiability. The originality of this paper is based on a non-technical departure of SOA and on separation of process and decision logic as a mean to decreased child deaths due to preventable diseases.

Keywords: Business Process Modelling (BPM), Business Rules Approach (BRA), Service Oriented Architecture (SOA).

1 Introduction

High product quality is ultimately the main goal of Information Systems Development (ISD). From a software quality perspective, quality can be described as e.g. maintainability, modifiability, reliability, and efficiency. From a user perspective, quality is considered as e.g. learnability, predictability, and consistency. From a business perspective, quality can be thought of as e.g. business alignment, appropriateness, effectiveness, and agility. These different “ilities” are interdependent but also sometimes in conflict, which require methodological design support and approaches, compromises, and assessments (Meggerle and Steen 2002). All approaches and methods to design Business Information Systems, or any other IS, exist to “ensure” or frame different qualities of the IS to be.

In today’s world of global competition and fast changes, agility has become a needed quality of any modern organisation or business. Since IS already are and increasingly tend to be essential to do business, the agility of the business’ IS are increasingly important. Old silo

systems and slow and cumbersome systems maintenance and modifications hamper the business agility. Thus, businesses need agile and quickly modifiable IS, which in turn require design of and architecture for agility.

2 The Modifiable Information System

Historically, many of the software “ilities” were used to discuss and describe software quality on a low level, namely the actual source code. The software crisis of the late 1960’s and the problem of “spaghetti code” lead to efforts to engineer ISD as a quest for higher quality. Software source code should be constructed as modules with limited information interdependence (Parnas 1979). In addition, a module should have high cohesion (“do one thing only”) and low coupling (as separate as possible from other modules), and hide information from each other (Page-Jones 1988). Through this, in the best of worlds, IS would be more easily maintainable and modifiable since modules could be changed and replaced (and reused) without affecting other parts of the IS.

In 1976, Dijkstra viewed Information Systems Development (ISD) as composition of artefact entities. The composition should be correct from the beginning and not debugged into correctness (see, e.g., Dijkstra, 1976). In addition, Parnas (1979) could show that it is not advantageous to assume ISD from an engineering centric interpretation i.e. flowcharts and rigid persistent data models. Quite contrary, a list of design decisions would make out a great point of departure for ISD (Parnas, 1979). Each design decision should be encapsulated in an entity, hidden from other entities. Different types of entities with different responsibilities should be kept separate, i.e. Separation of Concerns, to enhance maintainability, reusability, and modifiability (for instance as the separate databases we have been using for the last 30 years or so).

Modularisation on a source code level is good and well, but the architecture of modern IS naturally differ much from the IS of Dijkstra’s and Parna’s days. Today’s IS are much more complex (and bigger) and modifiability requires much more than the possibility to change or replace small modules in the source code. Over the years, modularisation, information hiding, separation of concerns, low coupling etc. have gradually moved upwards in abstraction level (in e.g. Zachman’s framework) to encompass larger and larger “chunks” of an IS. The progression has gone from low-level source code, through independent objects and entities, and components, to services. The Separation of Concerns (SoC) is now not only between different modules in the source code or between different layers of the architecture, but rather between different responsibilities as the enactment of business data, business processes, and business rules.

Hence, engineering centricity in Information Systems Development excludes the importance of the “what” and “how” of a business and neglects the businesses nerve system and the way business runs. Soft factors i.e. humans representing business owners have direct impact onto business activities forming business events and deciding on business survival. Therefore business processes, business rules and an appropriate architecture managing SoC ought to provide service-oriented information. Based on that, our perspective on information systems departs from viewing systems as services (see, e.g., Alter, 2010). In addition business service-orientation is the quality aspect suggested in this paper. Thus, important decisions should be provided by loosely coupled, responsible digital services derived from business and designed by business owners.

The Service Oriented Business Process (SOBP) implies to abide SoC as a fundamental policy. The term takes us back to the basics of architectural importance corresponding to the *must* of an ability to describe the non-artificial term “things” in need of change or design, representing solutions to problems (see, e.g., Finkelstein, 2006).

In this paper Separation of Concerns (SoC) correspond to breaking down business logic by separating decision logic from process logic. In addition, process logic is viewed as entities kept between a business event and a business activity in a business process why Business Process Modelling (BPM) will be a central term in this paper. Decision logic on the other hand, spans above business processes (see, e.g., The Business Rules Manifesto) and governs business activities why Business Rules Approach (BRA) is another central term of this paper.

BRA advocates an agile approach to SOA. In addition, BRA and SOA shares service-oriented concepts (see, e.g., Graham, 2006) whereas SOA correspond to a conceptual architecture, functioning independent of choice of, and providing properties for, realizing technology. Thus, SOA becomes the core component -the hub, in achieving SoC forming a digital ecosystem increasing IS quality.

Based on that, ontological consistency between SOA and SoC, intrinsically expressed by the basic principles of SOA is. Hence, a species label could be put onto the artefact providing knowledge on what the “thing” is and be designed into correctness by putting SOA into use expressing “what” to design.

A digital ecosystem hence, SOA realization keeps responsible and eligible digital services which in turn is the solution to a real world problem i.e. reaching the desired level of service-orientation in a business. Hence, the continuance of this paper will elucidate on business process service-orientation and the species of digital services i.e. process logic centric digital services derived from a business process segment and decision logic centric digital services derived from e.g. regulatory business documents.

2.1 SOA and Business Service-Orientation

SOA is able to express what to do but not how to do it (see, e.g. Arsanjani et al., 2009), in line with the SOA Manifesto and the basic principles for SOA which in turn often is neglected why architectural mismatch occurs. Moreover, SOA provides the essential properties for a digital service to execute in a digital ecosystem. SoC includes soft factors e.g. humans and the business humans run why business modelling became a ruling skill whittling down on SOA as a buzzword.

Corresponding to the enterprise model by Bajec and Krisper (2005) (see Figure 2) one of the promises SOA expresses is to manage business consistency. A consistent business implies that all the five sub models are in parity and managed in concert. However, a phenomenon such as business rules governing a part of a business that does not exist occurs. That in turn, implies to complete the design of the current BP or on the other hand to remove obsolete business rules. However, without separating business rules from process logic and application specific code that seems like a dilemma to remedy with direct negative business impact corresponding to enterprise model inconsistency. Separation of Concerns, modularization and encapsulation realized through information hiding thus becomes accentuated concerns of interest.

SOA is often viewed as Web services, however that is a great misinterpretation of the term and has clouded several SOA initiatives to end up in an expensive obsolete application platform. SOA should be considered a conceptual architecture functioning independently from choice of realising technology e.g. Web services (see, e.g., Arsanjani et al., 2009; Erl, 2008).

The term conceptual corresponds to the origin of services and has an intrinsic confusion. Hence to service-orient a business is not about Web services alone (see, e.g., Erl, 2008). The term “service” established in the 1930’s (Chesbrough and Spohrer, 2006) and Hill (1977) suggested a definition of a service as:

“[...] 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.” (Hill, 1977, pp. 17)

Based on that, the notion of the term “service” rose from a period in time where deliverables were viewed as goods and services. That perspective developed a foundation towards service science by e.g. simultaneity of production and consumption which ended up in a broad range of different services (Chesbrough and Spohrer, 2006). The service based economy became a fact from which e.g. IBM found that their predominant revenue was from IBM Global Services Business. That site of IBM did not exist prior to the 1990s (Chesbrough and Spohrer, 2006). “Services” eventually became an emergent term in scholar fields and knowledge was produced with a point of departure in e.g., Informatics and Computer Science. In 2005 the term SOA was established in e.g. “Service-Oriented Architecture: Concepts, Technology & Design” by Thomas Erl. However, the soft aspect i.e. humans were still of course affected by BISs which at that time rarely were seen as services (see, e.g., Alter, 2010). In 2009 Arsanjani et al., (2009) suggested 14 guiding principles constituting the SOA manifesto stressing soft aspects by incorporating them into the conceptual SOA. Along with the basic principles by e.g. Erl (2008) “service” was given properties representing high level abstract SOA and business imperatives for SOA realization:

1. Respect the social power structure in the business.
2. Be aware that SOA is going to propose change on different levels of abstraction.
3. The extent of SOA may vary. Keep the efforts manageable and meaningful.
4. Products and standards can neither realize SOA nor the service oriented paradigm alone.
5. SOA can be realized with a multitude of technologies and standards.
6. Establish uniform standards for the business to base policies upon.
7. Strive for uniformity on the outside but allow diversity on the outside.
8. Identify services through cooperation with business owners and technical staff.
9. Maximize the use of services by considerate the current and future extent of use.
10. Verify that services are in parity with business goals.
11. Design services so that they correspond to real world use.
12. Separate the non-parallel transformable aspects of a system.
13. Reduce the implicit dependencies and publish external dependencies for in-creased robustness and simultaneous decrease the impacts of changes.
14. Arrange each service close to a holistic manageable functional unit on each level (Arsanjani et al., 2009, <http://www.soa-manifesto.org/>).

“Service orientation is a paradigm that frames what you do. Service-oriented architecture (SOA) is a type of architecture that results from applying service orientation.” (Arsanjani et al., 2009, pp. 1, <http://www.soa-manifesto.org/>)

Based on that, service science, corresponding to a mean and an effect of a service based economy, turned into new business needs. Businesses were in need to become service-oriented implying that artificial business modelling became crucial for business survival. A prerequisite for realizing that was of course to be clear of what a “service” is in a specific business context. That in turn was a prerequisite for being able to know which part of the business that was in need of service-orientation.

The 14 guiding principles corresponding to high level SOA ends in business reconstruction considerations. However, when designing a SOA the designer ought to consider the very basics of SOA e.g. the service requestor, the service provider and the service repository. Consequently, this paper suggests that the *interaction* between those three actors casts the ground of service-orientation. Hence, a digital service must obtain certain properties for being able to execute and live, corresponding to be stored, requested and provided, in a digital ecosystem. The basic principles of SOA intrinsically expressing Separation of Concerns, information hiding and encapsulation are suggested by Erl (2008) accordingly:

1. Service Contract: a digital service must be reusable.
2. Service Loose Coupling: digital services must be loosely coupled but still know about each other.
3. Service Abstraction: a digital service hides logic from other digital services.
4. Service Reusability: digital services encapsulate logic advocating reusability.
5. Service Autonomy: digital services have responsibility for the encapsulated logic.
6. Service Statelessness: a digital service does not store temporary data from previous requests server side.
7. Service Discoverability: digital services must be discoverable.
8. Service Composability: digital services executing in a SOA must be composable (Erl, 2008).

Thus, to deliver on the SOA manifesto takes; responsible, composable and discoverable digital services. The basic principles of SOA incorporated with the BR modelling principles of BRA leave us with separate digital services encapsulating decision logic, process logic and application specific code. The two different species encapsulate logic for being able to hide it from other digital services. Then, information hiding advocates reusability by making digital services loosely coupled. A digital service therefore becomes responsible for the logic it encapsulates why e.g. service requests do not have to be stored server side, advocating service reusability corresponding to one of the previous mentioned quality aspects and “ilities”.

However, a digital service must be discoverable in line with the very basics of SOA; a service requestor requests a discoverable service from the service directory in which the service is provided by a service provider -corresponding to service-orientation. The service requestor is of course interested in being provided with a useful desired result why digital services must be composable. Based on that, the right (corresponding to the requested) decision logic must be composed with the right (corresponding to Ibid) process logic. Thus, digital services must still know about each other (high cohesion) by sharing the same basic principles, for the desired level of service-orientation to be realized.

3 The Relationship between Business Processes and Business Rules and Separation of Concerns

The relationship as well as the difference between Business Rules (BR) and Business Processes (BP) can be described as the “what” and “how” of a BIS (see, e.g., Date, 2000). BR modelling focus on the “what” of a BIS in the form of “what must or must not be the desired state of affairs of this business.” The desired states are constrained and described by rules. BP modelling focus on the “how” of a BIS in the form of “how must the flow of work be carried out to attain the desired state of affairs of this business.” The desired states are thus reached through the sequence of activities and procedures of one or many BPs. Why the BP metaphor can be formulated as a business nerve system.

Since BRs of the Enterprise Model (EM) are connected to the business visions and goals (equivalent to the “why” column in the Zachman framework) BRs triggers and controls BPs, and BPs uses and requires BRs. This Separation of Concerns means that BRs are associated with the operational *decision logic* of a BIS, while BPs are associated with the operational *process logic* of a BIS.

As (almost) all BPs include points in the flow where decisions need to be taken (i.e., decision points), in order to end the process or branch to another flow, a BP depends on and requires rules. When these are inherent in a BP they are normally in the Event-Condition-Action (ECA) form, meaning: *when* “event” occurs, *if* “conditions” are fulfilled, *then* execute “action”. The problem of the ECA form of BR is that it mixes the *decision logic* and the *process logic* (see, e.g., Morgan, 2002) and therefore violates the principle of SoC. In addition, it contradicts the basic principles of service-oriented design, e.g. information hiding and logic encapsulation important for e.g. service design, maintainability, modifiability, and composition corresponding to the “ilities” thus, quality aspects of BIRD and BIS.

3.1 Decision Logic Governs Process Logic

BRs and BPs connect when a decision point is reached in a process flow and one or several BRs need to be fired to make a decision, which controls the continued flow. Based on this discussion, we will use the following definitions of the different logics associated with BR and BP:

- *Decision logic*: a rational and stateless set of interconnected rules leading to a decision based on known values of terms and facts, producing logic values of true or false; (corresponding to BR)
- *Process logic*: a rational and stateful flow of work with events, activities, actors, and decision points, transforming input to output. (corresponding to BP)

BR modelling according to the principles of BRA explicitly discourages from temporal considerations – when in time a certain rule should be executed is not part of the rule itself. A major difference between ECA and BR is thus the temporal aspect, and from a BRA perspective, the temporal aspect is the responsibility of the BP model and the process logic since business rules spans above business processes (see, e.g., BRG, 2003).

A business process event occurs at a certain time and ends in a business process activity at a certain time. “When”, “how” (corresponding to process logic) and “who” (corresponding to “Role”) is thus aspects addressed by process logic. In Figure 1, representing a business

process in EPC-notation, the process flows from event to activity which is displayed by the arrows between each box. However, that flow needs to be rational and stateful if e.g. a role should be able to know when to execute an activity.

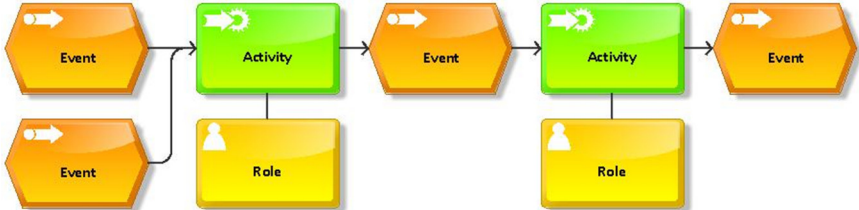


Figure 1: An EPC formatted Business Process

Figure 1 displays a business’s nerve system; business events and business activities are related by the black arrows that spans between them. Hence, that is what represents the rational and stateful flow of work with events, activities, actors, and decision points, transforming input to output. Process logic thus executes between an event and an activity why an event ends in an activity with actors, decision points and a state. An example of process logic is e.g. a calculation where the result is the output from the input. The result as such is enabled by e.g. the subtraction’s sign expressing “how” corresponding to process logic while “what” to subtract corresponds to decision logic. Hence, for being able to base a decision upon such a result the result must be treated as a known value for being able to produce logical values expressing “what”. Business rules and business processes should therefore be designed in concert (see, e.g., Holmberg and Steen, 2011). In addition, that reflects the importance of keeping parity between the sub models in the EM.

Based on that, BRs can be related to a BR-centric enterprise or business model, for instance the one suggested by Bajec and Krisper (2005) in Figure 2. An enterprise or business model is a high-level model, which tries to take an all-encompassing, albeit engineering like, view of the essential models for any BIS. It also relates the different models to each other to form a system of interconnected models. It shares many perspectives with the popular Enterprise Architecture model called the Zachman framework (see, e.g., Finkelstein 2006; Bajec and Krisper, 2005).

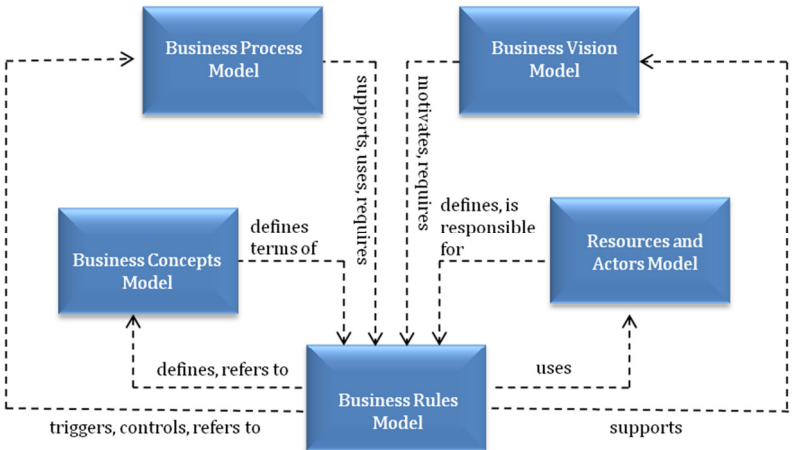


Figure 2: The Enterprise Model (Bajec and Krisper, 2005, pp. 428)

The model comprises five sub models.

1. The Business Vision Model represents the overall strategy of the organization, focusing on goal structure and on the problems standing in its way;
2. The Business Process Model describes the processes necessary to achieve the described structure of the goals and explains their input and output;
3. The Business Rule Model defines and maintains the explicit and implicit BR;
4. The Resource and Actors model focuses on the structure of the resources and their relation to actors;
5. The Concepts model establishes a common vocabulary for all the concepts and terms used in the organization. The model aims to minimize misconceptions and ambiguity of the ontology within the organization (Bajec and Krisper, 2005).

As can be seen in Figure 2, BRs play a major role and are connected to all the other models. From the perspective of this paper, especially the relationship between processes and business rules are important. Hence, BRA is a vital component in any service-oriented initiative and must be considered to achieve the desired level of business service-orientation.

That business rules (BR) are an important business asset is starting to become common knowledge. A relatively new approach that takes this perspective is the Business Rules Approach (BRA) (see, e.g., Ross, 2003). BRA focus BR analysis and modelling as an essential and distinctive area in BISD and treat BRs as “[...] first-class citizens in their own right” (Morgan, 2002, pp. 54). In von Halle (2002) Barbara describes a rather detailed and thorough approach to develop information systems which take into account the three major tracks in BISD of business data, business processes, and business rules. She calls her approach *STEP*, which is an acronym for: 1) separation of rules from other perspectives on the IS, 2) making rules traceable, 3) externalizing rules into an own layer or part of the IS and ultimately execute rules in a special component (such as a BR Engine or BR Management System) separate from the rest of the IS, and, 4) position rules for change such that they can be easily found, understood, changed, and then made effective without redeploying the whole IS corresponding to quality aspects of ISD and IS.

Hence, what is important with BRA is that BRs are acknowledged as a major aspect of IS and ISD and that BRs should be separate and external from other parts of the IS. By doing this, both migratability and modifiability of an IS are potentially easier to achieve to a higher degree than with the presently normal approach of burying the BRs deep within the source code. Separation of concerns and encapsulation, two classical principles in Software Engineering for achieving maintainable and modifiable software systems, is taken several steps higher in abstraction level to reach the business layer. This seems crucial for e.g. migration of legacy systems. Therefore, the benefits of the BRA in BISD are:

- BRs are written declaratively in natural language;
- BRs are managed and run separately from other parts of the BIS;
- BRs are maintained and modified by business people rather than engineers; and
- BR management systems (BRMS) accommodate running natural language BR as digital services and enables real-time changeability of the deployed BRs without reprogramming the BIS, (see, e.g., Morgan, 2002; von Halle, 2002; Graham, 2006).

As described in the introduction of this paper decision logic spans above business processes (see, e.g., The Business Rules Manifesto) and governs business activities. Thus, Business Rules Approach becomes crucial for the design, management and deployment of the descendant of the declarative paradigm of ISD, the business rules. Additionally, one of the

essential pillars of any Business Information System (BIS, see e.g., Carlsson et al., 2010) is decision handling, i.e. based on conditions in the form of input, stored data etc. a consequence (change a flag, update data, branch a flow etc.) is deduced. These kinds of decisions are necessary for automatic handling of for instance loan approval, eligibility checking, insurance claims processing, diagnosis and so on. Consequently, it is hard to imagine any automated information system without some kind of decision handling.

One of the problems when it comes to automatic decision handling built into software, e.g. traditional IT-supported information systems like order processing, is that the decision logic has been imperatively coded in a programming language (e.g. JAVA) and mixed in with the rest of the source code. This is a problem not the least when it comes to migrating legacy applications. In Earls (2002) the authors devise a method to extract business rules from legacy source code. They test their method on an IT system in their own organisation (British Telecom) and found for instance obsolete rules still in effect and active rules domain experts had no knowledge about. This was thus revealed only by systematic extraction of the business rules from the source code of the system.

4 Applied Business Logic Service-Oriented

The clarification of the two different species of digital services and business process service orientation realized the implementation of VacSam. If considering the soft aspects of this paper reveal why we are interested in business process service-orientation and Separation of Concerns, namely child prosperity.

VacSam is a composition of decision logic and process logic centric digital services responsible for part of the coordination of the Swedish vaccination activity. VacSam provides unique vaccination recommendations to any child despite that specific child's origin country. Thus, a child from any of the 193 sovereign states of the world could be phased into the Swedish vaccination program with the help of VacSam. That in turn is creating parity among children's vaccination statuses in Sweden. That was not trivial previous to VacSam because e.g. all 193 sovereign states of the world have their own vaccination schedule regulating inoculation (discussed in, Holmberg and Steen, 2011).

The business rules governing the Swedish vaccination activity were derived from regulatory documents and policy texts published by e.g. The National Board of Health and Welfare (Socialstyrelsen, 2008). The objects constituting the business rules represented the business concept model which provided us with the context specific ontology. The ontology as such was used to ensure that the right terms were related in the business object model. Clarifying how terms were related let us design well-formed business rules completing the business rule model. Those rules were implemented into about 60 decision logic digital services categorized by which vaccine each rule belong to. Figure 3 illustrates an excerpt of the decision logic digital services spanning from *Basic Vaccinated/SE* to *GeneralRules*.

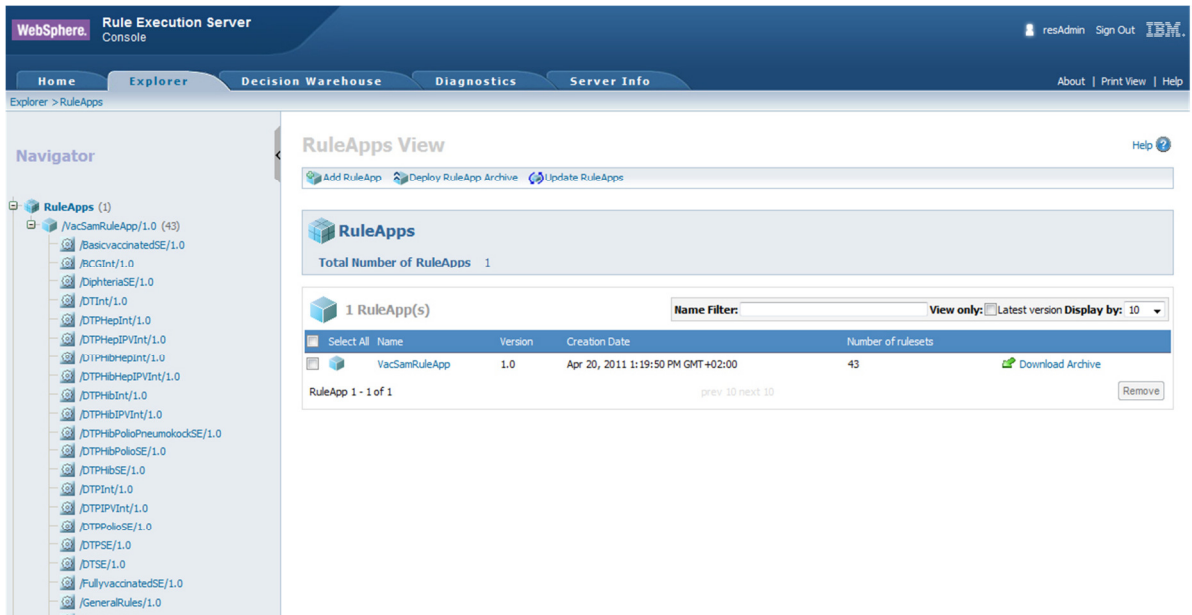


Figure 3: RES decision logic digital services view

Figure 4 represents a Web Service Definition Language (WSDL) view. WSDL contain all the necessary information for the digital service to be requested and used. Figure 4 is complementing Figure 3 and is the result of clicking one of the rule sets illustrated in Figure 3. However, Figure 4 illustrates how each rule set, corresponding to a package containing more than one business rule, is equipped with e.g. its own WSDL definition. Consequently, about 60 decision logic digital services (excerpt seen in Figure 3) provide the decision logic governing the Swedish vaccination recommendation activity.

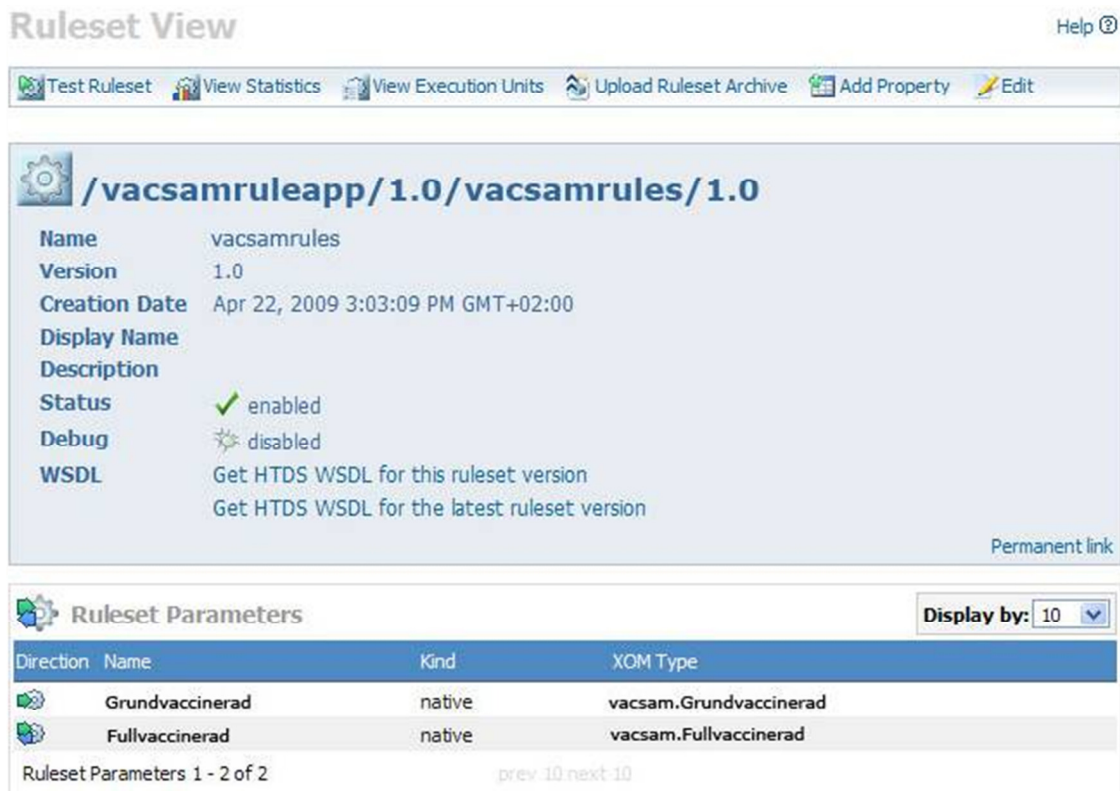


Figure 4: RES ruleset WSDL definition view

Business process modelling in line with the basic guidelines for BPM (see, e.g., Becker et al., 2010) allowed us to extract process logic corresponding to e.g., the calculation of a child's age from the Swedish vaccination recommendation activity. Modelling that part of the Swedish vaccination activity provided our initiative with a complete business process model answering to the well-formed business rules we have designed. That cumulative design process corresponds to what we name the service-orientation of the Swedish vaccination recommendation activity. The emergence of VacSam became evident (discussed in, Holmberg et al., 2011).

VacSam as such is composed digital services forming a digital ecosystem. It consists of about sixty decision logic digital services and two process logic and application specific code digital services. The recommendation per se corresponds to the purpose of VacSam; to decrease the WHO (2010) estimation of 1.4 billion yearly child deaths due to preventable diseases. In addition, to decrease the informal estimate of 70% overuse of vaccines, a waste of medical resources.

VacSam provides ease to those matters by composing the two species of digital services. However, the digital services constituting VacSam exist independently from each other but must be composed for being able to deliver the useful and desired result corresponding to the unique vaccination recommendation. In addition the digital ecosystem of VacSam answers to the desired level of service-orientation of the vaccination recommendation activity conducted in the Swedish health care sector. Consequently, an analogue business process was automated into a service-oriented digitalized component through SOA services which in turn is increasing vaccination efficiency in the sense of making governing business logic boundless available to the business owners of the Swedish vaccination activity and answers to the previously mentioned quality aspects of BIS and BIRD.

5 Conclusion

Separation of Concerns is a prerequisite for the suggested service-oriented BIS realization. Based on this paper business logic could be separated as process logic and decision logic corresponding to the importance of SoC in this paper. Decision logic governs and deals with the "what" of a business. Quite contrary, process logic represents the "how" of a business. That separation distinguish which logic a digital service encapsulates and thus, which information that is hidden from the outside world. Based on that, a digital service could be categorized by its content which facilitates digital service composition. -If we do not know what to compose missing knowledge impedes to design into correctness and the desired useful result would be hard to provide. Hence, process logic centric and decision logic centric digital services could be composed providing the right information to the right user on the right time at the right place.

Then, two separate species of digital services evolved. Interrelated by their origin business process and governing business rules designed by business owners the two species could be composed into a useful BIS representing the solution to a problem i.e. VacSam. Also, implementing decision logic and process logic in separate digital services better ensures SoC and facilitates the overall design of a well formed digital ecosystem i.e., a collection of service oriented business processes keeping quality-"ilities".

Possible benefits of business process service orientation are increased efficiency corresponding to boundless availability of intellectual capital and information. Hence, an

increased IS quality. In addition process and decision logic becomes modifiable and reusable. Because process logic and decision logic is no longer implemented in the same digital service one specific digital service becomes loosely coupled to one specific business process. The decision logic is only implemented once and has got one entrance similar to the process logic. Consequently, SoC, SOA, BRA and BPM could together form a digital ecosystem delivering on quality-“ilities”.

References

- Alter, S. (2010). "Viewing Systems as Services: A Fresh Approach in the IS Field." *Communications of AIS*, 26(11), 195-224.
- Arsanjani, A., Booch, G., Boubez, T., Brown, P. C., Chappell, D., deVadoss, J., Erl, T., Josuttis, N., Krafzig, D., Little, M., Loesgen, B., Thomas Manes, A., McKendrick, J., Ross-Talbot, S., Tilkov, S., Utschig-Utschig, C., and Wilhelmsen, H. (2009). "SOA Manifesto.". City.
- Bajec, M., and Krisper, M. (2005). "A Methodology and Tool Support for Managing Business Rules in Organisations." *Information Systems*, 30(6), 423-443.
- Becker, J., Rosemann, M., and von Uthmann, C. (2000). "Guidelines of Business Process Modeling Business Process Management", in W. van der Aalst, J. Desel, and A. Oberweis, (eds.). Springer Berlin / Heidelberg, pp. 241-262.
- BRG. (2003). "The Business Rules Manifesto.", R. G. Ross, (ed.). City: Business Rules Group.
- Carlsson, S. A., Hedman, J., and Steen, O. (2010). "Integrated Curriculum for a Bachelor of Science in Business Information Systems Design (BISD 2010)." *Communications of AIS*, 2010(26), 525-546.
- Chesbrough, H., and Spohrer, J. (2006). "A research manifesto for services science." *Communications of the ACM*, 49(7), 35-40.
- Date, C. J. (2000). *What not how : the business rules approach to application development*, Reading, Mass.: Addison-Wesley.
- Dijkstra, E. W. (1976). *A discipline of programming*, Englewood Cliffs, N.J.: Prentice-Hall.
- Earls, A. B., Embury, S. M., Turner, N. H. (2002). "A method for the manual extraction of business rules from legacy source code" *BT Technology Journal*. 20(4), 127-145
- Erl, T. (2007). *SOA Principles of Service Design*: Prentice Hall.
- Erl, T. (2008). "SOA Principles, The Service-Orientation Design Paradigm. SOA Systems Inc.". City.
- Finkelstein, C. (2006). *Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies*, Boston: Artech House Inc.
- Graham, I. (2006). *Business Rules Management and Service Oriented Architecture: A Pattern Language*, Chichester, England ; Hoboken, NJ: John Wiley.
- Hill, T. P. (1977). "On Goods and Services." *Review of Income and Wealth*, 23(4), 315-338.
- Holmberg, N., and Steen, O. (2011). "Business Process and Business Rules Modelling In Concert for e-Service Design and Business Alignment" *Ist International Conference on Cloud Computing and Services Science (CLOSER)*. City: Noordwijkerhout, The Netherlands.
- Holmberg, N., Steen, O., and Carlsson, S. (2011). "Service Orienting the Swedish Vaccination Recommendation Activity with the Business Rules Centric Digital Service VacSam. Service-Oriented Perspectives in Design Science Research", in H. Jain, A. Sinha, and P. Vitharana, (eds.). Springer Berlin / Heidelberg, pp. 376-386.
- Meggerle, T., & Steen, O. (2002). *IT-kvalitet i praxis systemutvecklares kunskap om och syn på kvalitet*. Lund School of Economics and Management, Lund University, Lund.
- Morgan, T. (2002). *Business rules and information systems: aligning IT with business goals*, Boston: Addison-Wesley.
- Page-Jones, M. (1988). *The practical guide to structured systems design* (2. ed.). Englewood Cliffs, N.J.: Prentice Hall.
- Parnas, D. L. (1979). On the criteria to be used in decomposing systems into modules. In E. Yourdon (Ed.), *Classics in software engineering*. New York: Yourdon Press.
- Ross, R. G. (2003). *Principles of the business rule approach*, Boston: Addison-Wesley.
- Socialstyrelsens föreskrifter om ändring i föreskrifterna (SOSFS 2006:22) om vaccination av barn (2008).
- World Health Organization. (2010). Immunization surveillance, assessment and monitoring Retrieved 2010-03-17, from http://www.who.int/immunization_monitoring/diseases/en/