A METHOD FOR DEFINING OPTIMUM SERVICE GRANULARITY

René Wiersma, Master of Informatics, HU University of Applied Sciences, Utrecht, The Netherlands, rene.wiersma@student.hu.nl and Eridium Consultancy, Rotterdam, The Netherlands, rene.wiersma@eridium.nl

Pascal Ravesteyn, Center for People, Process & Technology, HU University of Applied Sciences, Utrecht, The Netherlands, pascal.ravesteijn@hu.nl

Abstract

In the past 10 years a new paradigm called Service Oriented Architecture (SOA) has emerged that is based on the development, deployment and reuse of (web) services which can easily be assembled in different ways allowing organizations to quickly adapt to changing business needs (Cox and Kreger, 2005). However while SOA has a large potential for business one of the most complex issues in any SOA project is to define the right granularity of the services. While Steghuis (2006) and others state the importance of optimum service granularity our research tries to answer the HOW question by using the Design & Engineering Methodology for Organizations (DEMO) (Dietz, 2006). Our research consists of a literature study and the execution of two essential case studies at organizations in the Netherlands. Validated by interviews that were held at the organizations the case studies learned that the applied methodology is directive in obtaining a functional optimum in service granularity for business services and information services.

Keywords: Web services, Granularity, SOA, DEMO

1 INTRODUCTION

In the past 10 years an important change has occurred in how to develop, integrate and reuse information systems. A new paradigm called Service Oriented Architecture (SOA) has emerged that is based on the development, deployment and reuse of (web) services which can easily be assembled in different ways allowing organizations to quickly adapt to changing business needs (Cox and Kreger, 2005). The SOA concept benefits business because IT can be developed and implemented much faster and at lower development costs. Also it makes organizations processes more adaptable to change. Even the definition of Service-Oriented Architecture points at the common benefits of flexibility and reusability. Regarding the aspect of flexibility Weske (2007) refers to the definition of Burbeck (2000) that services gain flexibility by runtime coupling with the service registry. This dynamic coupling of services is not reached most of the times. Contrary to the definition of Burbeck (2000) Weske (2007) states better to speak about enterprise services. Reuse is reached if a service contains functionality with a clear business value and can be used directly (Weske, 2007).

Besides this SOA also requires that organizations evaluate their business models to fit service-oriented analysis and design techniques, deployment and support plans, and carefully evaluate partner, customer, and supplier relationships (Papazoglou and van den Heuvel, 2006). However while SOA has a large potential for business one of the most complex issues in any SOA project is to define the right granularity of the services. The quest for the right granularity is complex if the promises of flexibility and reusability must be obtained. Most of the times a senior technical specialist answers this question on gut feeling. A method for service definition is important in environments where a lot of services are available and the set of available services changes over time. Burbeck (2000) states services to be grounded on shared organizational principles. These principles makes sure services operate without errors, support flexibility and can be joined together to fit in business processes.

The question how services should share organizational principles or how services should be modeled to obtain flexibility and be able to adopt organizational changes is not answered yet. Steghuis (2006) states that the quest for service granularity has been addressed in many articles (De Jong et al, 2010, Foody, 2005; Papazoglou et al., 2006; Sims, 2005; Feuerlicht et al., 2007; Rosen, 2007) but none of these sources answers the question of how to define service granularity properly, neither do they provide some kind of concrete guidelines. Most sources just mention the importance of finding a right service granularity.

While Steghuis (2006) and others state the importance of optimum service granularity our research tries to answer the HOW. In this paper we show that with the use of the Design & Engineering Methodology for Organizations (DEMO) (Dietz, 2006) a guideline can be found for finding an optimum service granularity on the business, informational and data levels of organizations.

In the remainder of the paper we will first describe our research method, and then in section 3 we explain the concept of service granularity. Section 4 briefly describes the DEMO method while in section 5 the use of this method to define service granularity is shown. The used approach is validated via case studies in section 6 and conclusions are given in section 7.

2 RESEARCH METHOD

As a starting point in developing a guideline for finding an optimum service granularity we chose the Information System Research Framework of Hevner, March, Park and Ram (2004). This is based on the fact that Hevner et al. (2004) propagate that studies in the IS research domain contain both descriptive and prescriptive research.

The descriptive part of the research (knowledge-producing activity) aims to understand and explain how service granularity is defined, while the prescriptive approach (knowledge-using activity) aims at improving service oriented architecture (March and Smith, 1995; Hevner et al., 2004).

The research consists of two major activities based on the framework. First a literature study of existing research was conducted (the knowledge base). Based upon this study it was decided to use the DEMO method as a foundation to develop our guideline on defining service granularity.

Secondly by using and extending the DEMO method in two case studies the guidelines on how to define optimum service granularity were developed. In the framework of Hevner et al. (2004) these activities are related to the 'environment' and 'develop/build' aspects.

Guided by the case study structure of Bryman and Bell (2007) two essential case studies have been executed at organizations. One case study has been executed at Pretium Telecom in Haarlem, the Netherlands. Pretium Telecom provides telecom services for about one hundred and fifty thousand customers in the Netherlands. The second case study is about a fictitious value chain as part of an assignment at the HU University of Applied Sciences in Utrecht, the Netherlands.

At Pretium Telecom interviews were conducted with the Chief Information Officer and the lead technical architect. The interview questions were based on the Enterprise Engineering Framework of Op 't Land (2008). The framework structured the interview to cover both the business context and the technical aspect of software development. Based on the interview results a DEMO organizational construction diagram is created. To validate this model it is discussed with participants of the interviews.

The second case study is grounded in the architectural description as part of the assignment. The architectural description followed the Enterprise Engineering Framework structure

3 WHAT IS SERVICE GRANULARITY

Granularity is a term that reflects the degree of modularity of a system. Papazoglou et al. (2006) define service granularity as the unit of modularity of services. It is the amount of functionality that is exposed by a service. There exist two types of granularity of services. Fine grained services typically implement a single atomic operation and exchange limited amounts of data. Coarse grained services implement high-level business functions (Feuerlicht and Wijayaweera, 2007).

Steghuis' (2006) research concluded that several types of granularity exist for different kind of services. Several authors are referring to several types of services. Papazoglou et al. (2006) uses the service types Business Services, Infrastructure Services, and Component Based Service Realisations. Erl (2005) uses the service types Orchestration Services, Business Services, and Application Services. Schekkerman (2004) describes the Business of Organization Services, Information (System) Services, and Technology Infrastructure Services, which is part of the Capgemini Integrated Architecture Framework. This paper uses the service types Business Services, Information Services, and Data Services. The definition of these service types are aligned with the Integrated Architecture Framework of Capgemini and also with the Enterprise Engineering Framework (Op 't Land, 2008), and it is aligned with the three homogenous layers of the organization theory (Dietz, 2006).

Based on literature study Steghuis (2006) made a classification of the aspects of granularity and drew it on the several service types. This paper adjusted the terminology of service types used by Steghuis. This paper uses the service types Information Service and Data Service respective to the service types Information System Service and Software Service as used by Steghuis. Table 1 shows the classification of aspects of service granularity grouped by service type.

Business Service	**Information Service**	**Data Service**
Functionality	Functionality	Functionality

Flexibility in Business processes	Flexibility in Business processes	
Problem Complexity	Cognitive and Structural Complexity	Cognitive and Structural Complexity
Reusability	Reusability	Reusability
Composability	Composability	Composability
	Reusability of Legacy	
Sourcing	Sourcing	
Genericity	Genericity	Genericity
Context-independence	Context-independence	Context-independence
	Performance	Performance

Table 1 – Classification of service granularity aspects grouped by service type (Steghuis, 2006)

The relation with DEMO is substantiated by the foundation of the organization theory. The organization theory exists of three homogenous systems: the B-organization (for Business); the I-organization (for Intellect); and the D-organization (for Document) (Bunge, 1979), applied by Dietz (2006). This concludes that the B-organization provides business services, the I-organization provides informational services and the D-organization provides document or data services.

The results of the two case studies are being tested against the aspects of service granularity. In preparation the ontological models of the organizations are modeled. An ontological model on business level gives overview of the essential business activities. These activities are named as transactions between actors. A transaction is defined by the execution of production activities and coordination activities within a generic social pattern (Dietz, 2006). The ontological model is the organizational construction diagram and draws the organization abstracted from its implementation.

The DEMO methodology is briefly explained in the next paragraph.

4 WHAT IS DEMO?

DEMO is a method for ontology based enterprise modelling. An ontological system is defined as the cohesion of the composition, the structure, and the environment of the system (Dietz, 2006). DEMO has proven to be an effective method for decomposing organizations. The method is based on the Ψ-theory. The Greek letter is pronounced like PSI and stands for Performance in Social Interaction. This is the basic paradigm of the theory and is about the performance of the organization related with the social interaction of the organization or other systems. DEMO is grounded on three organizational layers from the organization theorem, called the B-organization that performs ontological transactions, the I-organization that performs infological transactions, and the D-organization that performs datalogical transactions. Figure 1 expresses the organizational theorem.

The business organization is the essential organizational layer that communicates and produces facts to realize business results. This organizational layer exists out of actors who are producing unique and definitive facts. On this organizational layer the actors interact with other social entities (actors) in the system. For example in our case study at Pretium Telecom an actor is the contract manager who requests the bank (which also is an actor) to proceed with the automatic money transfers for the current period. The activities between these two actors within the system generate new facts. These new facts are the transfer of money from one account to another account.

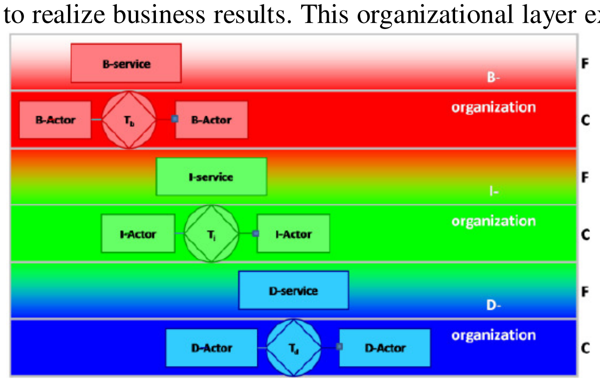

Figure 1: The organizational layers of the organization theorem

With the informational organization layer we mean the information processing layer. Information processing is the calculation of data via algorithms into other data and presents the information to the business layer. Within the informational organization of Pretium Telecom an information transaction exists that calculates an invoice amount based on the usage figures, the rates, contract data and elapsed time.

The data organization layer has the responsibility of storing, copying, searching, changing and removing of data. The transactions on the data layer are not familiar with information, but only with the raw data itself. The data organization layer is only allowed to communicate with the informational layer. This means that the data layer provides the data that is required for an informational process and stores the data afterwards. One of the data transactions at Pretium Telecom is the storage of contracts in a database table.

5 DEFINING SERVICE GRANULARITY USING DEMO

The Cambrigde University Press dictionary (2009) translates optimum in "most likely to bring success or advantage". The process for defining the optimum service granularity starts on the business layer of the organization by creating an organizational construction diagram. Every transaction on this layer is part of the essential business where only unique facts are produced. For each case study we conducted the organizational construction diagram. This paper elaborates only on the case study at Pretium Telecom.

5.1 Business Services

A part of the organizational construction diagram of Pretium Telecom in figure 2 consists of three transactions. These are T01, T02 and T03. Every transaction serves a unique and essential goal of the organization. As mentioned before by Dietz (2006) a transaction is the total set of coordination acts and production acts within a generic social pattern. This means concrete for transaction T01: Admit new contract that all data required for execution of the production act to process new contracts including all communicative acts to get the desired result are embedded. The transaction is being executed by actor A01: Contract admitter following the request of actor CA01: Prospect. Grounded in the DEMO theory every transaction can only produce one fact. The services definition of Burbeck (2000) says that every transaction adds specific business value. Therefore a transaction is synonym with a business service.

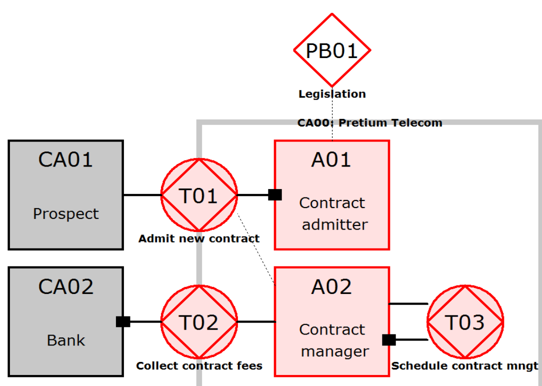

Figure 2: Pretium Telecom

Figure 2: Pretium Telecom

Every transaction in the organizational construction diagram is tested against the determinants of service granularity of Steghuis (2006) as shown in table 1.

For the determinant of functionality Fenton et al. (1997) states that business functionality adds value to an organization. Every transaction in the organization construction diagram is part of the essence of the organization and therefore is every transaction of added value for the business.

The Cambridge University Press dictionary (2009) translates flexibility in "able to change or be changed easily according to the situation" while Steghuis (2006) concluded that the meaning of flexibility in business and IT is the level of changeability of business processes or IT modules. Every transaction in the organizational construction diagram embeds the generic social pattern. This pattern determines the process of communication including the production of the fact. Due to the generic character of the pattern every transaction follows the same process. The inter process dependencies can be connected randomly. This supports the flexibility of the business process orchestration on the level of transactions.

Steghuis (2006) puts one of the four areas of the description of complexity by Fenton et al. (1997) within the domain of business services. This is the complexity of the problem. Steghuis (2006) advises to split the problem in smaller parts to reduce the amount of complexity. This suggestion makes the service granularity lower grained in case the amount of complexity it too big to oversee at once. The DEMO theory does not support this kind of splitting transactions, unless every transaction is producing a unique fact that could be composed into a solution for the total problem. At business level this is a decomposition of production activities for executing a transaction. The decomposition of causally related transaction types is important for the notion of a business process (Dietz, 2006). This is called the composition axiom in DEMO.

Steghuis (2006) refers to the definition of Basset (1997) for the reusability determinant. Basset (1997) defines reusability like the following: “Reuse is the process of adapting a generalized component to various contexts of use”. Steghuis (2006) concluded that while services are defined the granularity of services will be small grained because of the reusability determinant. In this case services contain only a small unit of functionality. Following the theory of DEMO makes reusability of services very clear on all layers of the organization. Every transaction has only one actor who is responsible for producing the fact, but the request for producing the fact can originate from different actors. This shows that reusability is completely embedded into the DEMO theory and is expressed visually by the design.

The compose ability determinant is addressed in the composition axiom of the DEMO theory and already explained in the discussion about complexity.

The sourcing determinant of service granularity is about ordering services to be aligned with sourcing strategy. DEMO is a successful language in expressing implementation decisions within the field of business and IT (Op ‘t Land, 2008). For example, in figure 2 one can discuss about outsourcing transaction T01. In case a decision has been made to outsource T01, the responsible actor A01 becomes redundant within the business domain. Within the DEMO model the impact on the organization becomes clear. The dashed arrow between transaction T01 and actor A02 shows an information link for tacit knowledge. This means that actor A02 needs the knowledge of the production facts, stored within transaction T01, while performing the production acts that are associated with its responsibility. The information links between actors and (transaction) banks restrict the nature of the interaction to the information exchanged (Hoogervorst, 2009). This is called interstriction.

Steghuis (2006) cites Foody (2005) on the aspect of generality. Foody (2005) says that services should be designed from the business point of view. The usability of services by the business is of greater importance compared to simple or generic services. In essence the transactions within an organization construction diagram are designed based on the production facts an organization produces. This makes a direct link to the business value. The aspect of generality is implicitly embedded in every transaction, because DEMO focuses on the ontology of the organization only. For example transaction T01 in figure 2 is capable of processing all kinds of telecom contracts for Pretium Telecom. At the ontological level the processing parameters are of subordinate importance for executing a transaction.

Papazoglou (2003) uses the concept of “loosely coupled” when discussing context independency. This means that a service should be able to execute unaware of the context it runs in. The transactions of the organization construction diagram do not support to be fully context independent. The embedded generic social patterns within a transaction enable to create a chain of transactions fulfilling a process. This chain of transactions causes several waiting moments in the process of communication. In this way a parent transaction can only be finished when the child processes are finished. These dependencies are made visible in a transaction pattern diagram of DEMO, which is not part of this paper. For example the relationship between the transactions T02 and T03 is explained in figure 2. The process starts with transaction T03. At a certain time actor A02 starts transaction T03 to manage contracts. This communication process leads to the request for execution of transaction T02 at the bank. Only when transaction T02 finishes, transaction T03 can finish. This is a periodical cycle. Another context dependent factor is shown in figure 2. Actor A02 can only execute transaction T02 in case actor A02 owns explicit tacit knowledge about transaction T01. The DEMO theory calls this dependency interstriction and is made visible by a dashed line.

Based on the case study DEMO models we can conclude that the transactions defined in the organizational construction diagram addresses the determinants of service granularity at the business layer.

5.2 Informational services

The informational layer, as part of the organization theorem, is about collecting information and providing that information to the business layer. The organization theorem defines that the function of the informational layer supports the construction of the business layer (Dietz, 2006). Every actor on the business layer needs information to perform its responsibilities. That information can be versatile, but at least it is associated with a transaction on the ontological level or it is required by interstriction on the ontological level. Thorough information analysis should demonstrate which information exactly is required. De Jong, et al. (2010) uses a methodology that focuses on the situational information and operational information required for executing the responsibility reasoning from the business layer. This method of information analysis results into the exactly necessary information that is required per business transaction on the ontological layer. This paper will not elaborate further on De Jong, et al.'s method of analysis.

The infological model of Pretium Telecom associates the ontological layer by actors CAI01 and CAI02 (see figure 3). These actors are equal to the internal actors on the business layer of Pretium Telecom. Two infological transactions are defined to execute the responsibilities of the actors. These transactions provide access to the informational layer of the organization theory. Because De Jong, et al. (2010) reuses the concepts of the DEMO theory while researching the ontology of the informational layer and the document layer, the characteristics of business transactions also count for transactions on the informational layer. This also means that the determinants of service granularity can be measured with informational transactions.

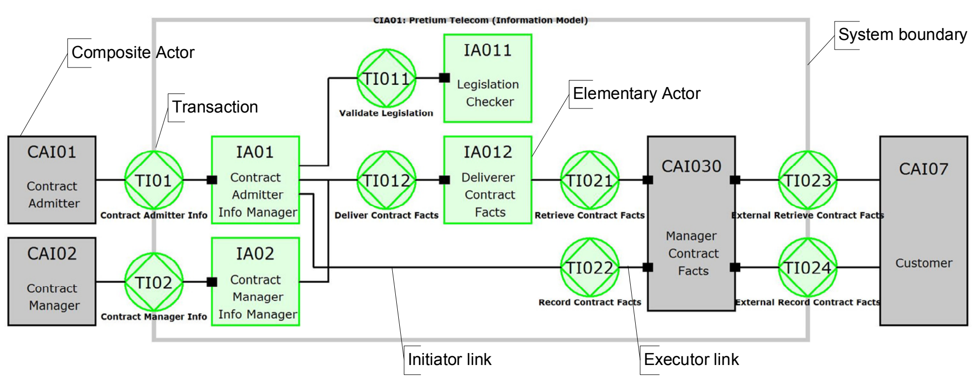

Figure 3: Infological model Pretium Telecom

Despite that DEMO models are abstracted from implementation and order; the case studies showed, based on experience that it is recommended to create the informational models in a left to right order.

This way of modelling supports reasoning from a front-end towards a back-end. In this case the front-end functions as a portal towards information, and the back-end as the gateway to the data.

The front-end and back-end approach was applicable for analyzing the informational models in both case studies. The pattern allows to visually splitting the infological model in three areas. Figure 4 shows the front-end and back-end approach in the infological model of Pretium Telecom. The front-end of the informational model is substituted by I.A.P, which stands for Information Access Point. This is the entrance for an actor to get access to the informational layer. The information access path can be implemented in several ways, like a portal or other kinds of user interface. Associated with I.A.P. is INF.S, which stands for Information Services. Per I.A.P. at least one information service is connected. This directly shows which information services are required for an actor. An actor within the domain of Information Services shall access certain banks of facts in processing information. In case the bank of facts is part of the organization it is modelled like actor CAI030. See figure 4. Otherwise, in case the bank of facts is not part of the organization it is modelled like actor IA011. This type of actor points at retrieving information from an external Legislation bank of facts.

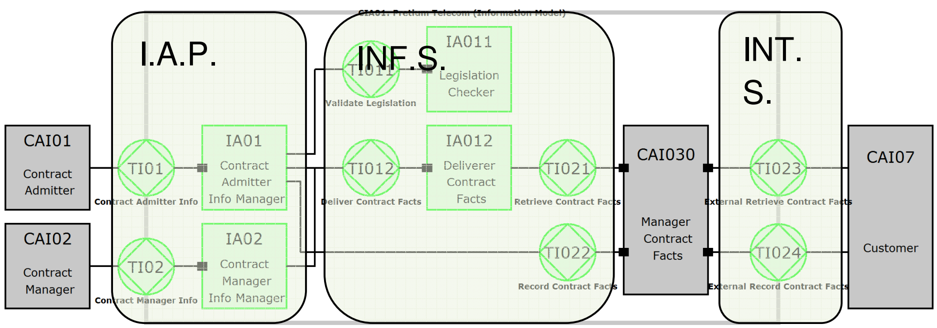

Figure 4: Infological model Pretium Telecom (Front-end and Back-end)

The third area that is recognized in the informational model are the Integration Services, abbreviated as INT.S.. The integration services support the information exchange in the value chain. This has been modelled in figure 3, where a customer can verify and update its own contract information. The integration services are specifically mentioned for external information exchange outside the borders of the organization. Web services can be an applicable implementation pattern for this kind of services.

When the aspects of service granularity for information services from table 1 are being measured on the informational model, then the determinant of functionality is supported by the fact that every information service is primary supporting the construction of the business layer.

The flexibility of the business process is determined at the ontological level. The informational layer exists by supporting the business process and provides information towards that business process. How the information is being interpreted is not relevant within the informational layer.

Cognitive or structural complexity is described by Fenton et al. (1997) as respectively the unit of understanding and interpretation by human beings of complexity in software and the unit of complexity in the structure of software programs. The informational layer itself does not say anything about complexity of the implementation of services. The informational model does provide guidelines for implementation, like transactions TI021 and TI022 in figure 3, where retrieval of information is separated from saving information. An actor could execute several calculations or derivations before the information is provided. The information analysis method by De Jong, et al. (2010) gives a better overview of the required information.

Reusability of information services becomes directly clear in figure 3. Both information managers, actor IA01 and actor IA02, use the informational transaction TI012. The reusability of legacy can be made clear in the same way.

The composability determinant is almost not applicable for the informational layer. The informational model provides a clear overview in the essential information transactions that are required in supporting the business layer. Information analysis will show when an information object must be composed from other information objects.

The infological model can make proposals supporting decision making on sourcing of informational actors within or outside the organizational borders. The infological model expresses this by creating new integration services that are associated with the external information provider in case the information provider is placed outside of the organization. Sourcing of information services are supported by the principles of the service-centric approach (Arnold et al., 2002). By integrating external information sources into the information process shows the reusability of third party services. E.g. in the case of Pretium Telecom can a credit check be incorporated on the infological level to check a new customer before the contract is accepted. The credit check service is not part of the responsibilities of Pretium Telecom, but it lies somewhere else.

The transactions in the infological layer do not provide a clear view on the business rules that are applicable for the infological transaction at runtime. A business rule provides a specific actionable guidance to implement business policies, which are non-actionable directives whose purpose is to govern or guide the enterprise (Hoogervorst, 2009). The absence of a clear view on business rules makes informational transaction generic, but when the informational transaction will be implemented those rules become valid. Because the information service does not have any knowledge of the business context, the information service must support to process all kinds of information objects. Due to modelling essential information services only, a generic and concise information service can be designed.

De Jong, et al. (2010) says that informational actors can only possess explicit knowledge. Every information service is responsible for collecting its own data. The information service is not dependent on information from other information services, but maximum limited by the executing order within the business process. The principle of "loosely coupled", as used by Papazoglou (2003) in explaining context independency is therefore partly applicable. E.g. in figure 3 the transaction TI012 can only be executed when actor IA012 has collected the required data from transaction TI021.

The infological model is abstracted from its implementation. Therefore the infological model cannot provide a guideline for performance. Foody (2005) wrote in his weblog a guideline for information services. An information service or an integration service must be executed between five milliseconds and 5 seconds and the size of the message should not be greater than one megabyte. During implementation of the information services and integration services the performance aspect must play an important role defining the size of the service. The performance aspect itself shall not change the informational model in its essence.

The directive of Information Access Points, Information Services and Integration Services, in relation with a thorough information analysis shows that the infological model meets most of the aspects for service granularity. The essential informational transactions are modelled within the infological model and those transactions are primary supportive for the business transactions on the business layer of the organization.

5.3 Data Services

The data logical layer of the organization is about saving, retrieval, removal, and transportation of data. The organization theorem defines that the function of the data layer supports the construction of the informational layer (Dietz, 2006). The functions on the data logical level are of great importance for the actors within the informational layer (De Jong, et al., 2010).

The data logical model of Pretium Telecom associates the infological layer by the actors CAI030 and IA011 (see figure 3). These actors are both associated with the data logical layer. The actor CAI030 is associated because of its responsibility to retrieve and store data. The actor IA011 is associated because of its responsibility to retrieve Legislation data, which is available in an internal or external database.

When the functionality aspect for service granularity from table 1 is being measured with the data logical model, then the aspect of functionality is substantiated by the fact that every data service is primary supportive to the infological layer of the organization.

The data logical model does not provide an answer for the service granularity aspect for cognitive and structural complexity. The data logical model shows that data needs to be stored, retrieved, and transported only, but the model does not show for which data objects these activities count per data logical transaction.

Reusability of data services is expressed in the data logical model by the function of an actor. One actor is responsible for storing data, one actor for transporting data, etc.. Due to the generic responsibility an actor has, the data service can be reused for all data storing request respectively data transporting requests.

The service granularity aspect of composability is not visible within the data logical model. The model shows the essential data logical transactions that are required by the informational layer.

The transactions in the data logical layer do not provide any details about the underlying data structures that are required to process a data service. This makes the data logical layer generic in usage. The business rules that counts for data, when implementing a data structure are not visible from the data logical model. Therefore this kind of business logic must be implemented separately.

Context independency, as aspect of service granularity, is supported by the data logical layer by associating explicit responsibilities per actor for data collecting, data storing, and data transporting. Every actor does not have to know anything about other actors when executing its responsibility. This supports the statement of “loosely coupled” services, which makes the data services in the data logical model completely context independent.

The performance aspects of service granularity on the data logical layer are equal with the informational layer. The data logical layer is implementation independent and therefore cannot provide a guideline on the topic of performance.

Regarding the data logical model associated with the aspects for service granularity concludes that the data logical model’s abstraction level is too high. It does not contribute much in obtaining optimum service granularity.

6 FINDINGS FROM THE CASE STUDIES

Designing an information architecture grounded in the requirements of the business and/or the market, and the opportunities of IT are one of the driving forces for the IT function (Poels, 2007). The information architecture is translated into construction principles. These principles are directive in selecting standard components or developing custom made software (Poels, 2007).

Both case studies started from the construction principles. The ontological model of business services shows the essence of the organization without forcing any form of implementation. Measuring the aspects of service granularity showed the connection between business transactions on the business layer and business services. Every business service can be associated with the business value it obtains and how it supports the goals of the organization. The business service is not only about executing its responsibility, but also about the coordination activities within the social domain.

Every business service represents a part of the ontology of the organization and therefore every business service provides a unique unit of functionality. This precise amount of functionality within a business services results into an optimum in service granularity. The aspect of context independency is limited by the interstriction possibilities between business services. With help of a transaction pattern diagram the relative context independence of business services is shown.

To switch between the ontological business level and the ontological information level a business actor can shape itself into an information actor (De Jong, et al., 2010). A thorough information analysis must be executed that results in information services. The case studies showed that the list with information objects is directive in defining the service messages that actors interchange. Case studies showed that the information analysis must not be limited to the requirements of the business level only, but also must take into account the already existing physical data structures, like existing database models. The information analysis associates the business requirements with the data.

The information services that are designed grounded in thorough information analysis can be split up in three areas. These are: Information Access Points, Information Services, and Integration Services. The Information Access Points provides a clear organization of the information model. They are services that provide access to the information. The Information Services are supporting the business requests for information. These information services are supporting the business process and represent a large part of the essential information supply. The integration services are specifically mentioned to support the information integration with external parties. These information services are not primary supportive to business actors, but are supportive for the business goals in the area of the extended enterprise. A practical form of implementation is web services.

Every information service is a piece of functionality that must be supported by the system. This provides structure to the functionality that has to be designed. The information service is an expression of the complete social pattern of coordination and production. This means that the information service itself must support the social pattern as well. Based on the case studies results this is hard to combine together in one physical service. By creating a composite service at the implementation level, the DEMO informational model provides a functional directive in obtaining optimum service granularity. When the functional information service model of DEMO is being implemented, the service granularity determinants complexity and performance require extra attention, because these aspects are not directly supported by the methodology.

Both case studies showed that the data logical layer of DEMO has a too high level of abstraction. This makes it impossible to conclude that the data logical layer contributes to obtain optimum service granularity. The physical implementation of the data structures are a prescribing factor, which is not taken into account in the data logical model. The infological model, a thorough information analysis, and the existing data structures are together qualifying optimum service granularity for data services.

7 CONCLUSION

The hypothesis of this paper says that DEMO in combination with current scientific research of De Jong, et al. is directive for an optimum service granularity. Based on the results of the case studies the hypothesis is considered to be valid. The business model and the informational model of DEMO are both offering enough leads to make it plausible that optimum service granularity can be reached on a functional level.

At ontological layer the DEMO transactions are synonym with business services. The business services of both case studies are measured against the aspects of service granularity of Steghuis (2006). For almost all aspects the business services own the characteristics to achieve optimum service granularity. The context independency aspect of service granularity is not measured in the organization construction model, but with the transaction pattern diagram.

One of the case studies conclusions is the categorization of the information services on the infological layer. These categories are: Information Access Points, Information Services, and Integration Services. The information access points associate the business actor with the informational layer and offer the business actor the information supply that is required for executing the business service. The information services are basically the essence of the information supply and supportive for the business layer. The integration services are mentioned for information integration with external parties, like in an extended enterprise.

Every type of information service is measured with the aspects of service granularity. For almost all aspects the information services supports an optimum service granularity. Because that the information services could be prescriptive for physical implementation, the remark must be made that the informational model gives a functional directive in obtaining optimum service granularity. During implementation extra attention must be given on the aspects of complexity and performance.

The final conclusion based on the results of the case studies is the high abstraction level of the data logical model. Therefore this model hardly contributes in obtaining optimum service granularity.

The two case studies in this paper showed that the applied methodology is directive in obtaining a functional optimum in service granularity for business services and information services. Further research should demonstrate if the methodology used in this paper could be raised as best practice for obtaining optimum service granularity.

References

Arnold, B., and Op 't Land, M. (2002). An Architectural Approach to the Implementation of Shared Service Centers. In: Dutch National Architecture Congress 2002 (LAC2002). Zeist, The Netherlands

Basset, P., (1997). Framing Software Reuse – Lessons from the Real World. (1st ed.). New York: Prentice Hall

Burbeck, S. (2000). The TAO of E-Business Services – The Evolution of Web Applications into Service-Oriented Components with Web Services. Tech. rep., IBM Software Group

Bunge, M.A., (1979). Treatise on Basic Philosophy. A World of Systems, Vol. 4. Dordrecht: D. Reidel Publishing Company

Bryman, A., and Bell, E. (2007). Business Research Methods – second edition. Oxford University Press

Cambridge University Press. Cambridge Advanced Learner's Dictionary. Retrieved April 2009 from http://dictionary.cambridge.org/

Cox, D. and Kreger, H., 2005. Management of the service oriented-architecture life cycle, IBM systems Journal, Vol 44, No 4.

Dietz, J.L.G., (2006). Enterprise Ontology – Theory and Methodology. Springer Verlag

De Jong, et al., J., and Dietz, J.L.G., (2010). Understanding Organization Realization. 6th International Workshop on Cooperation & Interoperability – Architecture & Ontology. DESRIST 2010

Erl, T., (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall

Fenton, N., and Pfleeger, S., (1997). Software metrics – a Rigorous & Practical Approach. (Second Edition ed.). Boston: PWS Publishing Company

Feuerlicht, G., Wijayaweera, A., (2007). Determinants of Service Reusability, New Trends in Software Methodologies, Tools and Techniques, H. Fujita and D. Pisanelli (Eds.) vol. 161, pp 467-474, IOS Press

Foody, D. (2005). Getting web service granularity right. Retrieved April 2009 from http://www.soa-zone.com/index.php?/archives/11-Getting-web-service-granularity-right.html#extended

Hevner, A.R., and March, S.T., and Park, J. and Ram, S. (2004). "Design Science in Information Systems Research", *MIS Quarterly*, 28

Hoogervorst, J.A.P., (2009). "Enterprise Governance and Enterprise Engineering." Diemen, the Netherlands. Springer

March, S. T. and G. F. Smith (1995). "Design and natural science research on information technology." Decision Support Systems 15(4): 251-266.

Op ‘t Land, M., (2008). Applying Architecture and Ontology to the Splitting and Allying of Enterprises. Ph.D Thesis, Delft University of Technology

Op ‘t Land, M., and De Jong, et al., J., and Goedvolk, H., (2008). Enterprise Engineering Framework. Not yet published

Papazoglou, M., (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. In: Fourth International Conference on Web Information Systems Engineering. IEEE. Roma, Italy

Papazoglou, M., and Van den Heuvel, W. (2006). Service-Oriented Design and Development Methodology. International Journal of Web Engineering and Technology

Poels, R., (2007). Haal meer uit uw ICT: Interventies die ertoe doen. Scriptum

Rosen, M., (2007). A Methodology for Designing Service-Oriented Applications. Cutter Information

Schekkerman, J., (2004). How to survive in the jungle of Enterprise Architecture Frameworks: Creating or choosing an Enterprise Architecture Framework. Trafford

Sims, O. (2005). Developing the Architectural Framework for SOA – Part 2 – Service Granularity and Dependence Management. CBDi Journal, June 2005

Steghuis, C. (2006). Service Granularity in SOA Projects: A Trade-off Analysis. MSc Business Information Technology, University of Twente

Weske, M. (2007). Business Process Management: Concepts, Languages, Architectures. Springer-Verlag Berlin Heidelberg