

The Benefit of Enterprise Ontology in Identifying Business Components

Antonia Albani, Jan L.G. Dietz
Delft University of Technology
Chair of Information Systems
PO Box 5031, 2600 GA Delft, The Netherlands
{a.albanij.l.g.dietz}@tudelft.nl

Abstract. Companies are more than ever participating in so-called value networks while being confronted with an increasing need for *collaboration* with their business partners. In order to better perform in such value networks information systems supporting not only the intra- but also the inter-enterprise business processes are necessary in order to enable and ease collaboration between business partners. Therefore, they need to be *interoperable*. As the basis for building these information systems the concepts of *enterprise ontology* and *business components* are very promising. The notion of enterprise ontology, as presented in this paper, is a powerful revelation of the essence of an enterprise or an enterprise network. Reusable and self-contained business components with well-defined interaction points facilitate the accessing and execution of coherent packages of business functionality. The *identification of business components*, however, is still a crucial factor. The reported research seeks to improve the identification of business components based on the *ontological model* of an enterprise, satisfying well-defined quality criteria.

1 Introduction

Due to drastic changes in the competitive landscape, enterprises are more and more focusing on their core competencies, outsourcing supporting tasks to their business partners. Companies are therefore becoming part of so-called *value networks* [1-3] with the increasing need to identify, improve, and automate as much as possible their core business processes. In order to enhance the competitive advantage of value networks an effective *collaboration* between enterprises is of great relevance. Technological innovations such as global, web-based infrastructures, communication

Please use the following format when citing this chapter:

Albani, A., Dietz, J.L.G., 2006, in IFIP International Federation for Information Processing, Volume 214, The Past and Future of Information Systems: 1976–2006 and Beyond, eds. Avison, D., Elliot, S., Krogstie, J., Pries-Heje, J., (Boston: Springer), pp. 243–254.

standards and distributed systems, enable the implementation of business processes in and the integration of business processes between companies, thus increasing the flexibility of the business system and enabling the *interoperability* of their information systems. However, the deployment of the information and communication technologies does not always meet expectations. While developing inter- and intra-enterprise information systems, it is necessary to use a suitable *methodology* for modeling the business domain. Additionally, information systems need to be modeled on a high-level of abstraction that is understood also by business people, who are defining the requirements and using the respective systems. The use of *business components* for the development of a high-level information system is valuable since they 'directly model and implement the business logic, rules and constraints that are typical, recurrent and comprehensive notions characterizing a domain or business area' [4] (all other components are considered either to deliver services to these business components or to offer some general functionality). The identification of business components thus is the first step in the development of an information system according to current standards. It is a very crucial one and, therefore, it should be performed at the highest possible level of quality. In the field of identifying reusable and marketable business components there is still little research initiative to date (e.g., [5-7]). As recognized by [8], 'more formal methodologies are needed to make the component based software development paradigm into an effective development tool'.

The starting point is the set of requirements that have been elicited from the business domain, preferably on the basis of an abstract model of the organizational activities. In [9] some quality criteria are proposed regarding such a model, which we adopted for our current research: it should be *consistent* (i.e., there are no contradictions or irregularities), *comprehensive* (i.e., all relevant issues are dealt with), *concise* (i.e. the model does not contain superfluous matters), and *essential* (i.e., it shows only the deep structure, independent of the realization and the implementation of the enterprise). We call a model of the organizational activities of an enterprise that satisfies these requirements enterprise *ontology*. Most of the current process modeling techniques, like the Petri Net [10, 11], Event Driven Process Chains (EPC) [12], and Activity Diagrams [13], next to the traditional flow charts, do not satisfy all of the quality criteria mentioned. The notion of business process is not well defined and there exists no distinction between business and informational actions. Consequently, the difference between business processes and some other types of process remains unclear. This leads to the conclusion that they do not specifically address business processes but can be used for any discrete event process. Other approaches, as e.g. from the Language/Action Perspective (LAP), claim to offer a solution for the mismatch between social perspectives and technical perspectives by explicitly focusing on business specific communication patterns, where social beings achieve changes in the (object) world by means of communication acts [14]. The *enterprise ontology* [15] methodology is an approach that incorporates LAP and that additionally distinguishes between essential (business), informational and documental actions. Because of these advantages, we chose the methodology referred to, also known as DEMO (Design and Engineering Methodology for Organizations), for producing the ontological model of an enterprise, providing the basis for identifying business components.

Based on the enterprise ontology, this article introduces a new method for the identification of business components. It is structured as follows: To exemplify the usability of the approach, the domain of *strategic supply network development* (SSND) and its ontological model is introduced in section 2. SSND is used throughout the paper as an example domain for inter-enterprise collaboration. In section 3, the method for identifying business components is applied to the SSND case. Discussions of the results as well as the conclusions that can be drawn are provided in section 4.

2 Enterprise Ontology and its Application to the SSND Case

The example domain of *strategic supply network development* comes from the domain of strategic purchasing [16-19]. The most evident differences regard the functions with cross-enterprise focus. Purchasing has become a core function in enterprises in the 90s. Current empiric research shows a significant correlation between the establishment of a strategic purchasing function and the financial success of an enterprise, independent from the industry surveyed [17]. One of the most important factors in this connection is the buyer-supplier-relationship. At many of the surveyed companies, a close cooperation between buyer and supplier in areas such as long-term planning, product development, and coordination of production processes led to process improvements and resulting cost reductions that were shared between buyer and suppliers [17]. In practice, supplier development is widely limited to direct suppliers (suppliers in tier-1), without taking into consideration the suppliers in subsequent tiers. Because of the increasing importance of supplier development we postulated the extension of the traditional frame of reference in strategic sourcing from a supplier-centric to a supply-network-centric scope [20]. This refocuses the object of reference in the field of strategic sourcing by analyzing and selecting supplier networks instead of single suppliers. The details of the domain are described while introducing the enterprise ontology of the SSND case.

As motivated in the introduction, we use the enterprise ontology for modeling the business domain according to DEMO [14, 15, 21, 22]. As is explained in [15, 21, 22] a distinction is made between production acts and facts and coordination acts and facts. Consequently, two worlds are distinguished: the production world (P-world) and the coordination world (C-world). The transaction axiom aggregates these acts/facts into the standard pattern of the (business) transaction. The complete ontological model of an organization consists of four aspect models. The Construction Model (CM) specifies the composition, the environment and the structure of the organization. It contains identified *transaction types*, which are executed by associated *actor roles* and describes the *links* to relevant information stored in *production* or *coordination banks*. The Process Model (PM) details each single transaction type of the CM by means of *transaction patterns*. Next to these patterns, it contains the *causal* and *conditional relationships* between transactions. The PM is based on business process patterns [22] and shows how the distinct transaction types are related. The Action Model (AM) specifies the *action rules* that serve as guidelines for the actors in dealing with their agenda. The State Model (SM)

specifies the *object classes*, *fact types* and ontological *coexistence rules* in the production world.

Based on this method, the ontology for the SSND case has been constructed. Space limitations prohibit us to provide a more extensive account of how the models in the figures hereafter are developed. Also, we will not present and discuss the Action Model. The basic idea of the SSND example is the identification of suppliers, located not only in tier-1 but also in the subsequent tiers, which are able to deliver specific components to the original equipment manufacturer (OEM) for constructing a specific product. This is established in sending out an offering request for a specific product to the tier-1 suppliers, which execute a bill-of-material explosion in order to decide which products need to be requested from their suppliers. This repeats until the request has reached the last tier. The information is then aggregated and split-lot transferred to the initial tier. Fig. 1 exhibits the Construction Model of the SSND case. The corresponding Transaction Result Table is shown in Table 1.

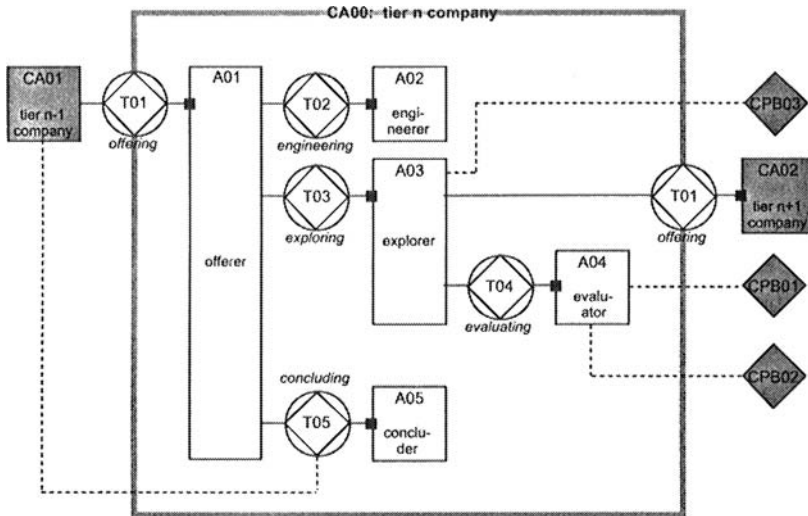


Fig. 1 Construction Model of the SSND case

Table 1. Transaction Result Table of the SSND case

transaction type	resulting P-event type
T01 offering	PE01 supply contract C is offered
T02 engineering	PE02 the BoM of assembly A is determined
T03 exploring	PE03 supply contract C is a potential contract
T04 evaluating	PE04 supply contract C is evaluated
T05 concluding	PE05 supply contract C is concluded

The top or starting transaction type is the offering transaction T01. Instances of T01 are initiated by the environmental actor role CA01, which is a company in tier

n-1 and executed by actor role A01. This company asks the direct supplier (company in tier n) for an offer regarding the supply of a particular product P. In order to make such an offer, A01 first initiates an engineering transaction T02, in order to get the bill of material of the requested product P. This is a list of (first-level) components of P, produced by A02. Next, A01 asks A03 for every such component to get offers from companies that are able to supply the component. So, a number of exploring transactions T03 may be carried out within one T01, namely as many as there are components of P which are not produced by the tier n company. In order to execute each of these transactions, A03 has to ask companies for an offer regarding the supply of a component of P. Since this is identical to a starting transaction T01, we model this also as initiating a T01. Now however, the executor of the T01 is a company in tier n+1. Consequently, the model that is shown in Fig. 1 must be understood as to be repeated recursively for every tier until the products to be supplied are elementary, i.e. non-decomposable. Note that, because of the being recursive, an offer (the result of a T01) comprises the complete bill of material of the concerned component of P.

Every offer from the companies in tier n+1 is evaluated in a T04 transaction. So, there is a T04 for every 'output' T01, whereby each company can have its own evaluation rules. The result of a T04 is a graded offer for some component of P. So, what A03 delivers back to A01 is a set of graded offers for every component of P. Next, A01 asks A05, for every component of P, to select the best offer. The result is a set of concluded offers, one for every component of P. This set is delivered to A01. Lastly, A01 delivers a contract offer to CA01 for supplying P, together with the set of concluded offers for delivering the components of P. Because of the recursive character of the whole model, this offer includes the complete bill of material of P, regardless its depth.

The CM in Fig. 1 contains three external production banks. Bank CPB01 contains the data about a company that are relevant for the evaluation of offers. Bank CPB02 contains the different evaluation methods that can be applied. In every instance of T04, one of these methods is applied. CPB03 contains identifiers of all companies that may be addressed for an offer. The dashed lines represent access links to these banks. Lastly, in the transaction result table (see Table 1), the supply of a product by a (supplying) company to a (customer) company is called a contract.

Fig. 2 exhibits the Process Model of the SSDN case. A coordination step is represented by a (white) disk in a (white) box; it is identified by the transaction number (see Table 1) and a two-letter extension: rq (request), pm (promise), st (state), or ac (accept). A production step is represented by a (gray) diamond in a (gray) box; it is identified by the transaction number. For modeling the SSND example case the so-called basic pattern (*request, promise, execute, state, accept*) has been used.

From the state T01/pm (promised) a number of transactions T03 (possibly none) and a number of transactions T05 (possibly none) are initiated, namely for every first-level component of a product. This is expressed by the cardinality range 0..k. Likewise, from the state T03/pm, a number of transactions T01 and a number of transactions T04 are initiated, namely for every offer or contract regarding a first-level component of a product. The dashed arrows, from an accept state (e.g. T02/ac) to some other transaction state, represent waiting conditions. So, for example, the

performance of a T03/rq has to wait for the being performed of the corresponding T02/ac.

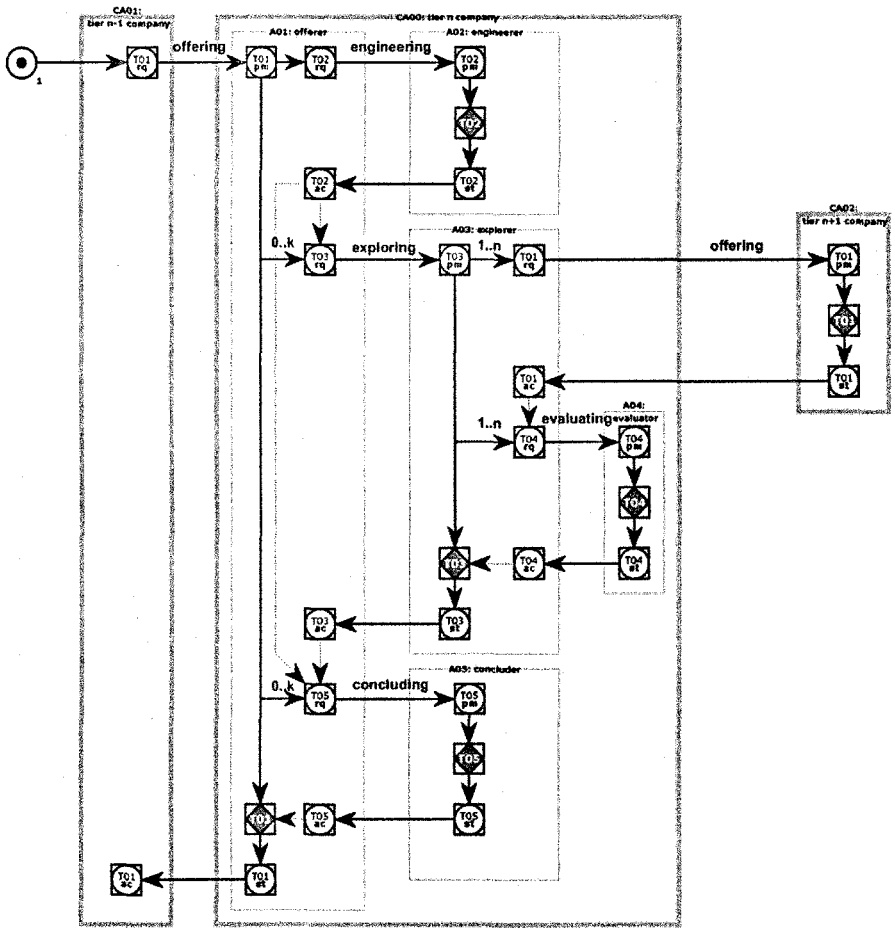


Fig. 2 Process Step Diagram of the SSND case

Fig. 3 exhibits the object fact diagram (OFD) and Table 2 the object property table (OPT). Together they constitute the State Model of the example case. The OFD is a variant of the ORM model [23]. Diamonds represent the (unary) fact types that are the result of transactions, also called production fact types. They correspond with the transaction results in Table 1. A roundangle around a fact type or a role defines a concept in an extensional way, i.e. by specifying the object class that is its extension. For example, the roundangle around the production fact type “C is evaluated” defines the concept of evaluated contract. Properties are binary fact types that happen to be pure mathematical functions, of which the range is set of, usually ordered, values, called a scale. Instead of including them in an OFD they can be more conveniently represented in an Object Property Table (Table 2). The information

items as defined in the SM, including the derived fact types, constitute all information that is needed to develop a supply network for a particular product.

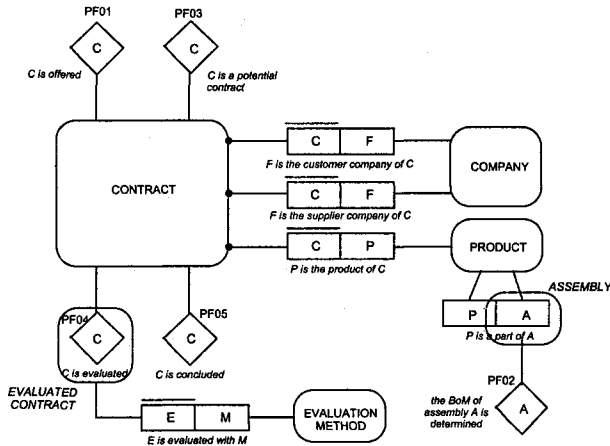


Fig. 3 Object Fact Diagram of the SSND case

Table 2. Object Property Table of the SSND case

property type	object class	Scale
< company information >	COMPANY	< aggregated data >
< contract terms >	CONTRACT	< aggregated data >
evaluation_mark	CONTRACT	NUMBER

3 Identification of Business Components in the SSND Case

Having introduced the main models of the ontology of an enterprise, the information gained in the models is used for the identification of business components. The principle of modular design, on which business components are based, demands reusable, marketable, self-contained, reliable and manageable business components. They need to provide services at the right level of granularity and to have a formal and complete specification of its external view. The enterprise ontology, as introduced in section 1, provides the necessary basis for the realization of business components. With the enterprise ontology for the SSND case, the complete information related to that business domain is available. The *three dimensional method for business components identification (BCI-3D)*, applied in this section, aims at grouping business tasks and their corresponding information objects into business components satisfying defined metrics. The metrics used – being minimal communication between and maximum compactness of business components – are the basic metrics for the component-based development of inter-enterprise applications.

Since the identification of business components is strongly dependent on the underlying business model, the BCI-3D method uses the object classes and fact types from the SM and the process steps from the PM, including their relationships. One can distinguish between three types of relationships necessary for the identification of business components. The relationship between single process steps, the relationship between information objects and the relationship between process steps and information objects. A relationship type distinguishes between subtypes expressing the significance of a relationship. E.g., the relationship between single process steps expresses – based on their cardinality constraints – how often a process step is executed within a transaction and therefore how close two process steps are related to each other in that business domain. The relationship between information objects defines how loosely or tightly the information objects are coupled, and the relationship between process steps and information objects defines whether a corresponding information object is used or created while executing the respective process step. All types of relationship are of great relevance in order to define which information object and process steps belong to which component.

The relationships are modeled in the BCI-3D method using a weighted graph. The nodes represent either information objects or process steps and the edges characterize the relationships between the nodes. Weights are used to define the different types and subtypes of relationships and build the basis for assigning nodes and information objects to components. Due to display reasons the graph is visualized in a three-dimensional representation having the process steps and information objects arranged in circles, and without showing the corresponding weights (see Fig. 4).

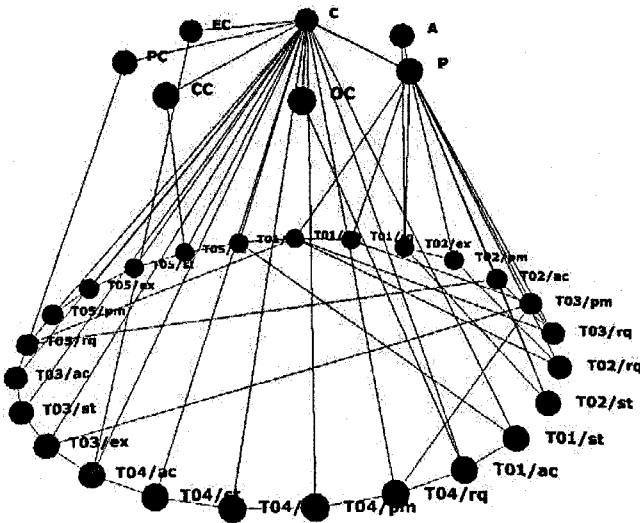


Fig. 4 Relevant relationships for the business component identification method (BCI-3D)

The graph shows all process steps and information objects with the relevant relationships of the SSND case. The shortcut names for the information objects are:

P (Product), A (Assembly), C (Contract), CC (Concluded Contract), EC (Evaluated Contract), OC (Offered Contract), and PC (Potential Contract). All information needed is gained from the enterprise ontology models introduced in section 0.

The relationship between the process steps is provided through the sequence in which the single process steps are executed (visualized by a *link* between two process steps). For the identification of business components we distinguish between *standard*, *optional* and *main* relationships, defining the significance of the relations between two process steps. This is expressed in Fig. 2 by the cardinality constraints assigned to the links. If a link has the cardinality range 0..k we call it an *optional* link, since the following process step does not need to be executed. If the cardinality range is set to 1..n then we call it a *main* link, indicating that the following process step is executed at least once. If no cardinality range is assigned, we call it a *standard* link, having the following process step executed exactly once. For the different types of links different weights are assigned in the weighted graph.

Let us have a closer look at the information objects. Fig. 3 and Table 2 introduce different types of potential information objects, namely, the object classes, fact types, and property types. Since property types define specific data belonging to an object class we do not consider that information in the BCI-3D method. Object classes, which are provided by external information systems – that concern the data, which is provided by the external production banks (see Fig. 1) – are traded in a special way in the BCI-3D method.

In addition to the relationships already introduced, the relationships between information objects and process steps play an important role in the business component identification method. The information to define those relationships can be gained from a Create/Use Table, showing which objects and facts are used or created in which process steps. For the SSND case, the relationships between information objects and process steps and their weights are visualized in Fig. 5.

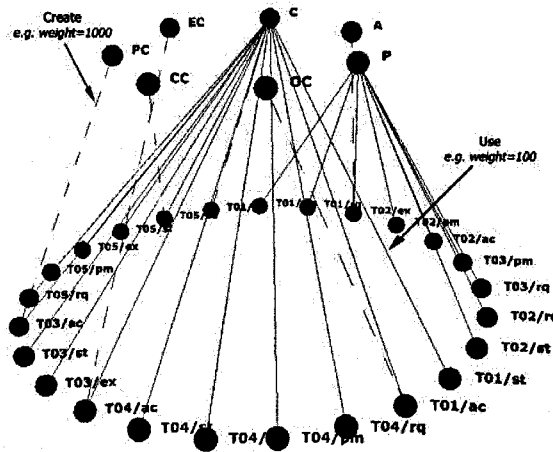


Fig. 5 Relationship between business services and information objects

In order to provide optimal grouping while minimizing communication and to ensure compactness of components, an optimization problem needs to be solved for

which a genetic algorithm has been developed. A detailed description of the algorithm would go beyond the scope of this paper. Applying the BCI-3D method to the graph introduced in Fig. 4 (including the different types of relationships with the corresponding weights as shown e.g., in Fig. 5) results in the following graph partition (see Fig. 6). Two business components can be identified immediately: one containing the business tasks related to *product management* and one containing the business tasks related to *contract management*.

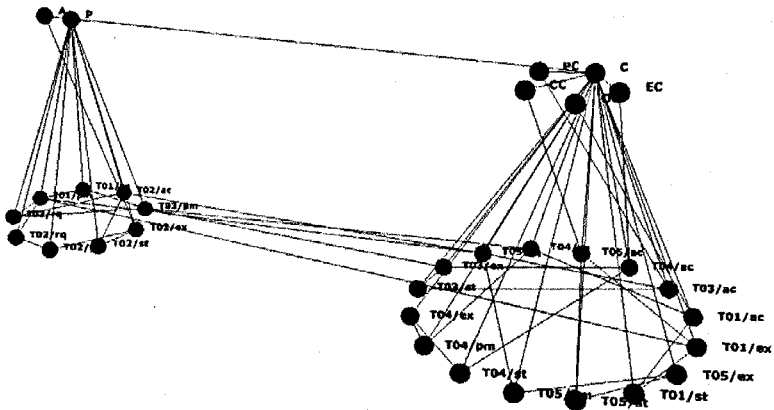


Fig. 6 Identified business component

Although in this small example only two business components were found, we have demonstrated a systematic identification of business components, considering all relevant information of the domain, which would not have been possible without a detailed analysis and modeling of the corresponding domain. The domain information is essential; it provides the basis for the next relevant steps, namely, the implementation of the business components.

4 Conclusions and Future Work

In this paper, we have addressed the problem of identifying business components, defined as the highest-level software components, i.e. the software components that directly support business activities. Although component-based software development is commonly considered superior to traditional approaches, it leaves the basic problems in requirements engineering unsolved: there are no criteria for determining that a set of identified components is complete (no component is missing) and minimal (there are no redundant components). The component-based approach also adds a new problem: the identification of components. We have addressed both problems by starting from the ontological model of the enterprise at hand. The methodology, as presented and demonstrated in this paper, does solve these problems in a satisfactory way.

First, the enterprise ontology constructed by means of DEMO is an appropriate model of the business domain. It satisfies the quality criteria as proposed in the introduction. As a consequence, the identified business components do really and directly support the business activities in which original new facts are created. Moreover, since the process steps (cf. Fig. 2) are atomic from the business point of view, one can be sure to have found the finest level of granularity that needs to be taken into account. Also, one can be sure that this set of process steps is complete and minimal.

Second, the BCI-3D method, based on the resulting models and tables of the enterprise ontology, provides an automated approach for the identification of business components. In using the BCI-3D method, different business component models could be generated for several business domains, e.g., for the domain of strategic supply network development (as presented in this paper), of educational administration, of automotive industry and of network operators. Its predecessor, the BCI method (Business Components Identification) [24], focused only on relationships between information objects and process steps, without taking into account the other types of relationships (between different information objects, and between different process steps). With this 'old' BCI method, however, we also generated component models for several domains. Worth mentioning is the one in the area of Customer Relationship Management (CRM) and Supply Chain Management (SCM) with more than 500 information objects and 1000 business functions [25]. But since the BCI method did not take into account all information available from the business domain, and was not implemented as a weighted graph, the optimization algorithms did not generate optimal solutions. We used all experiences with the old BCI method in order to develop a superior method for the identification of business components, the BCI-3D method. We demonstrated the suitability of the presented methodology by means of the SSND case. Because of its recursive nature, this case is not at all trivial. Yet, as we have shown, the process of identifying the business components is transparent and systematic.

Further investigations for the BCI-3D method are needed in evaluating reliability and stability of the resulting component models. Also, the types of relationships used and the corresponding weights assigned to the relationships need to be verified in additional example cases.

References

1. D. Tapscott, Ticoll, D., and Lowy, A., *Digital Capital: Harnessing the Power of Business Webs*. 2000, Boston.
2. T.W. Malone and Lautbacher, R.J., *The Dawn of the E-Lance Economy*. Harvard Business Review, 1998(September-October): p. 145 - 152.
3. B.J. Pine, Victor, B., and Boynton, A.C., *Making Mass Customization Work*. Harvard Business Review, 1993. 36(5): p. 108-119.
4. F. Barbier and C. Atkinson. *Business Components*, in *Business Component-Based Software Engineering*, F. Barbier, Editor. 2003, Kluwer Academic Publishers Group. p. 1-26.
5. A. Réquillé-Romanczuk et al. *Towards a Knowledge-Based Framework for COTS Component Identification*. In *International Conference on Software Engineering (ICSE'05)*. 2005. St. Louis, Missouri, USA: ACM Press.

6. Y.-J. Jang, E.-Y. Kim and K.-W. Lee. *Object-Oriented Component Identification Method Using the Affinity Analysis Technique*. In *9th International Conference on Object-Oriented Information Systems (OOIS)*. 2003. Geneva, Switzerland: Springer-Verlag.
7. K. Levi and A. Arsanjani. *A Goal-driven Approach to Enterprise Component Identification and Specification*. Communications of the ACM, 2002. 45(10).
8. P. Vitharana, F. Zahedi, and H. Jain. *Design, Retrieval, and Assembly in Component-based Software Development*. Communications of the ACM, 2003. 46(11).
9. J.L.G. Dietz. *Deriving Use Cases from Business Process Models*. In *Conceptual Modeling - ER 2003, LNCS 2813*. 2003: Springer Verlag.
10. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods and Tools*. MIT Press, MA, 2001.
11. K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Monographs in Theoretical Computer Science. Vol. 1, Basic Concepts. 1997: Springer Verlag.
12. A.-W. Scheer. *ARIS - Business Process Modeling*. 2 ed. 1999, Berlin: Springer.
13. OMG, *OMG Unified Modelling Language, Version 2.0*, in *Secondary OMG Unified Modelling Language, Version 2.0*, Secondary OMG, Editor. 2003: Place Published. p. Pages.http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML.
14. V.E. van Reijswoud, J.B.V. Mulder, and J.L.G. Dietz. *Speech Act Based Business Process and Information Modeling with DEMO*. Information Systems Journal, 1999.
15. J.L.G. Dietz. *Enterprise Ontology - Theory and Methodology*. 2006: Springer Verlag.
16. L. Kaufmann. *Purchasing and Supply Management - A Conceptual Framework*, in *Handbuch Industrielles Beschaffungsmanagement*. 2002, Hahn, D, Kaufmann, L. (Hrsg.): Wiesbaden. p. 3 - 33.
17. A.S. Carr and J.N. Pearson. *Strategically managed buyer - supplier relationships and performance outcomes*. Journal of Operations Management, 1999. 17: p. 497 - 519.
18. R. McIvor, P. Humphreys, and E. McAleer. *The Evolution of the Purchasing Function*. Journal of Strategic Change, 1997. Vol. 6(3): p. 165 - 179.
19. L.M. Ellram and A.S. Carr. *Strategic purchasing: a history and review of the literature*. International Journal of Physical Distribution and Materials Management, 1994. 30(2): p. 10 - 18.
20. A. Albani et al. *Dynamic Modelling of Strategic Supply Chains*. In *E-Commerce and Web Technologies: 4th International Conference, EC-Web 2003 Prague, Czech Republic, September 2003, LNCS 2738*. 2003. Prague, Czech Republic: Springer-Verlag.
21. J.L.G. Dietz. *The Atoms, Molecules and Fibers of Organizations*. Data and Knowledge Engineering, 2003. 47: p. 301-325.
22. J.L.G. Dietz. *Generic recurrent patterns in business processes*. In *Business Process Management, LNCS 2687*. 2003: Springer Verlag.
23. T.A. Halpin. *Information Modeling and Relational Databases*. 2001, San Francisco: Morgan Kaufmann.
24. A. Albani, J.L.G. Dietz, and J.M. Zaha. *Identifying Business Components on the basis of an Enterprise Ontology*. In *Interop-Esa 2005 - First International Conference on Interoperability of Enterprise Software and Applications*. 2005. Geneva, Switzerland.
25. B. Selk et al. *Experience Report: Appropriateness of the BCI-Method for Identifying Business Components in large-scale Information Systems*. In *Conference on Component-Oriented Enterprise Applications (COEA 2005) in conjunction with the Net.Objectdays*. 2005. Erfurt, Germany.