8-2010

Information Modeling: The Need of Semi-Automatic Model Analysis and Transformation

Jens Weller
Technische Universität Dresden, IWM, jens.weller@tu-dresden.de

Martin Juhrisch
Department of Business Management and Economics, Technische Universität Dresden, Dresden, martin.juhrisch@tu-dresden.de

Knut Großmann
Technische Universität Dresden, IWM, groszman@iwm.mw.tu-dresden.de

Information Modeling: The Need of Semi-Automatic Model Analysis and Transformation

Jens Weller
Technische Universität Dresden, IWM
jens.weller@tu-dresden.de

Martin Juhrisch
Technische Universität Dresden, WiSe
martin.juhrisch@tu-dresden.de

Knut Großmann
Technische Universität Dresden, IWM
groszman@iwm.mw.tu-dresden.de

ABSTRACT

Information models are seen as an important tool within the information systems discipline as well as in non-IS domains. However, modeling is a highly manual task and causes extensive effort. Modeling methods focus only on the creation of models, without giving concrete instructions of an appropriate use of them. An increased efficiency could be achieved if the necessary tasks are solved completely model based. Then algorithms, based on a set of rules, perform model operations. Once a model of the business problem has been created, the problem solving is carried out in the model space. However, several shortcomings prevent semi-formal models from being a suitable resource of an automated solution process. Problems result from a multitude of modelling aims, objects and procedures. In this paper an approach for a model-driven management is presented that aims at the specific problem of identifying service candidates in a service-oriented architecture.

Keywords

Information model, model analysis, method engineering

INTRODUCTION

In today's companies relevance of ecological issues is raising constantly. The ongoing market crisis calls for innovative ideas and energy-efficient products. Additionally, upcoming public regulations and laws increase the need for a change. Using light-weight materials is seen as one strategy to increase energy-efficiency of products; using renewable raw materials as another one (Milwich et al., 2006). In the recent years different research projects have been started aimimg the development and implementation of these technologies. Thereby, new production processes were designed according to the introduced technologies (Großmann and Wiemer, 2007). Hence, in different research projects different production processes using different hard- and software are developed.

These processes, however, have certain similarities. Especially storage and logistic processes might be similar. As these processes, however, are developed within different research projects with different researchers involved, different hard- and software might be used for addressing the same process. Thus, parts of the research results cannot be reused within later projects what leads to higher development costs. In 2004, a team of German researchers have joined finding solutions to overcome that problem (Hufenbach, 2006). Therefore, the production processes of several research projects shall be analyzed to find similar process components. These components will be harmonized and afterwards stored in a technology catalog for reasons of reuse.

For the identification of such components, the SOA paradigm offers interesting methods of resolution. SOA offer a modern approach to create services (a process component in our meaning) for reasons of reuse. For the identification of the services, there are many model-based approaches. However, these approaches fail in the case of a high number of distributed developed process models with complex interdependencies. Thus, existing methods guiding such an identification (Ivanov and Stähler, 2005) could not been used. The detection of similarities and differences between process models is the prerequisite for an automatic model comparison.

In this article we propose an approach that offers a way of identifying service candidates (process components) based on numeric ratios, which can be derived from process models. Thereby, we focus on the context of the model, not on the layout.

Thus, our approach can be used even in distributed modeling projects, where models are created with different levels of detail, different languages, different points of view, etc. We present ratios that guide the identification of service candidates out of process models and show the feasablity of the approach by implementing in the context of the above introduced research program.

Our research is a matter of design science (Hevner, 2004). Following the research methode presented in (Verschuren and Hartog, 2005), the paper is structured as follows. In the next section we start with a requirement analysis. Based on the identified requirements, ratios for the automatic analysis of process models are created afterwards. Finally, the implementation of the approach within the research program is demonstrated. The paper ends with a discussion, summarizing the research results and exposing open questions regarding the identification of service candidates.

REQUIREMENT ANALYSIS

Model ratios as basis

Ratios (or performance indicators) are used as prospective instrument and come up to an important information and control function (Staudt, 1985). Reichmann and Lachnit emphazise the impact of ratios as information in descision making processes in managerial sourroundings (Reichmann and Lachnit, 1976; Staudt, 1985).

A general ratio requirement is to map quantifiable data onto a concentrated form (Reichmann, 1995). Hence, we can derive two central characteristics. On the one hand it is presumed that every ratio implies informational value since its purpose is the condensation of data volumes to a single measurand. That measurand is the foundation to evaluate a certain situation subsequently. In addition, it is presumed that the data is quantitative measurable on a metric scale (Jäger-Goy, 2001). The construction of ratios is bound to the following requirements:

- Fitness for a particular purpose. The ratio should correspond with the information needed in a specific situation.
- Exactness. The exactness is conditioned by its reliability and validity.
- Actuality. The space of time between measurement and evaluation ought to be minimal.
- Cost-benefit ratio. The effort to survey the ratio should not cause costs above the value of the findings (Haufs, 1989).
- Simplicity and traceability. A result of a measurement must be simply interpretable.

The informational value of a ratio grows if it is connected into a comprehensive ratio system (Ester, 1997). This conclusion rests upon the assumption that single rather a few ratios are not able to exhaustively reflect the complexity of a system. Furthermore a multitude of single ratios impede the view to the essential issues (Wissenbach, 1967). Thus, with an ordered set of ratios we mean a ratio system. The ratios are correlated and provide as a whole information about a certain issue (Frank, 2001). Hence, the existence and catenation of at least two or more single ratios is condition precedent to create a ratio system.

A ratio can either be derived from superior ratios or developed concurrently and related with each other in a quantitative model. In the absence of such quantitative correlation, relations can be derived from empirical coherences as well (Ester, 1997). The creation of ratios and their coherency should always base on a comprehensive theoretically founding. The purpose should not be to respond to a specific question exempt from theory. For futher information concerning the development of ratio systems we refer to (Ester, 1997). As a general rule, the measurement of ratios in process models is related to single subprocesses or parts of the entire model. Seeing that, ratios are comparatively easy to implement in process models due to the fact of the straightforwardness to decompose processes into subprocesses. Considering a ratio system we have to assure to put measurement results of individual subprocesses into perspective. Thus, this is mandatory since we have to avoid the formation of suboptima (Engelke and Rausch, 2002). Besides, the examination of various subsystems associated with a multidude of ratios necessitates a concentration to a few significant ratios that combine collaboratively a maximized informational value.

The coverage of exclusive quantitative resp. quantifiable issues appears to be problematic (Ester 1997). In the case of additional required qualitative statements that are not reproduceable as a ratio, this information stay to a large extent outside of the examination. The adoption of ratios in process models is a broadly discussed issue in literature. Though, on most cases existing approaches focus on the measurement of complexity in process models. This means to survey in the first place the comprehensibility and (Gruhn and Laue, 2006; Cardoso, Mendling, Neumann and Reijers, 2006) and secondly the maintainability and correctness of models (Cardoso et al., 2006). We use ratios to identify eligible service candidates, a survey that bases not only on the measurement of complexity of process models. In fact criteria for the design of appropriate ratios have to be outlined in the following section.

Identification of service candidates

Grounded on the concepts of contract, service, and interface, the SOA paradigm aims the service relation to a semi- or fully automated activity in processes. This happens following the contract terms in which the characteristics of the activity's implementation are defined (Dietzsch and Goetz, 2005). The service functions – meaning differentiated and autonomously working functions of a service also usable by other services – are utilised by the interface of an application.

Schwemm et al. deduce five design principles from literature: business orientation, self-containedness, modularity, interface orientation and interoperability (Schwemm, Heutschi, Vogel, Wende and Legner, 2006). Services are business oriented if their functional scope is geared to the required objects. Services are modular and selfcontained if resources with high dependency to each other are combined in one service.

The design principles interface orientation and interoperability base upon the assumption that services represent stable interfaces that are entirely specified using technical and business metadata (Schwemm et al., 2006). As a complete and formal specification of processes in business models can not be presumed, we constrain the deduction of ratios to the principles of business orientation, self-containedness and modularity.

Business orientation

This design principle refers to the granularity of a service function. The granularity equates to the scope of functionality that is provided with the service function (Griffel, 1998). A service is business oriented if it contains these business objects that are essential to perform a certain business activity (Schwemm et al., 2006). The objects could be modeled and interconnected as information objects using a conceptual data modeling language and be asigned to processes in business process models. If so, service candidates are a process rather than a bulk of processes that perform a common business task and access similar information objects. Hence, the information objects of the processes, which constitute a service, must show a high coherence. A measure of the coherence of a system is the cohesion (McCabe, 1997). A high grade of cohesion describes a high coherence of the elements of a service. The contrary implies a low coherence.

Self-containedness

To what extent a service can be evaluated as selfcontained determines its maintainability. Following Simon, selfcontained systems are better to maintain compared to dependent systems since modifications just imply marginal modifications at neighboring systems (Simon, 1962; Wand and Weber, 1990). Simon operationalizes selfcontainedness by dint of coupling (Simon, 1962). Coupling is a measure for the pairwise coherence between several subsystems (Wand and Weber, 1990). A single or an amount of processes could be identified as a service candidate if this process resp. this amount is independent of other processes. A process is independent of other processes if its business objects are firstly not used by other processes and secondly the objects transferred to other processes are of little complexity (Yourdon, 1979). Accordingly this process could be automated as service without to hazard the other processes.

Modularity

By compling with the modularity principle during the design of a service, the complexity of the service could be reduced, parallel execution of services realized and uncertainty eliminated (Baldwin and Clark, 2000).

The fundamental idea of modular design can be traced back to the work of Parnas (Parnas, 1971). Selfcontained functional units are merged and provided with a defined interface (Balzert, 1998). Balzert defines a module as representation of a functional unit or a semantically related functional group that is self-contained; posseses defined interfaces for external access and is in matters of its scope qualitatively and quantitatively manageable and understandable (Balzert, 1998). Analogical to business orientation, modularity is operationalized using the criteria of cohesion. Efforts are being made to quantify the distance dimension for miscellaneous decompositions and correlations.

DESIGN

Single processes as well as an amount of processes using the same information objects can be identified as a service candidate according to previously observed design priciples autonomy, self-containedness and business orientation. Single processes can be identified as a service candidate if these processes are characterized by low coupling and high cohesion. An amount of processes can be identified as service candidates if they are characterized by high cohesion among the processes under consideration and low coupling towards outside processes. In this section coupling and cohesion will be operationalized to identify single processes and an amount of processes as service candidates. Furthermore requirements are derived for modeling languages to create process models that can be used to identify service candidates out of these models.

To enhance understanding of the ratios developed within this paper, we introduce an example, which will be used for demonstration for each of the ratios presented in this paper (see Figure 1). For the example model, we use the modeling language developed in the research projects mentioned in the introduction. The process view illustrates a process flow with objects manufactured (material opbjects) and the machines and information used to control the process (resource objects). The data-view represents the relations between the used resource objects. This view is similar to the modeling notation "SERM" which is also used within the Architecture of Integrated Information Systems (ARIS).

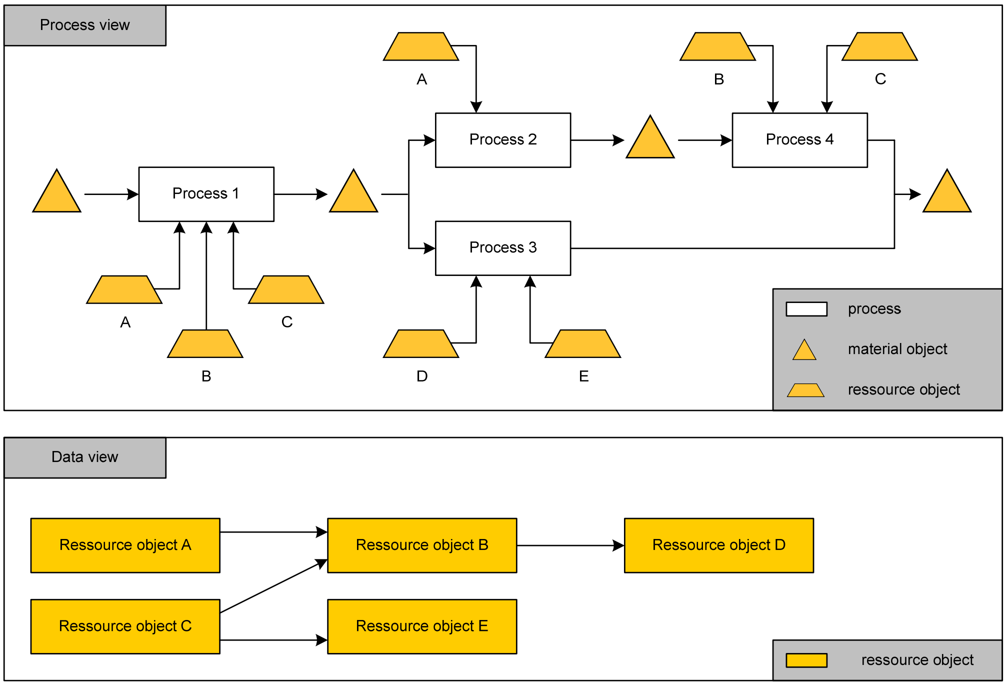

Figure 1: Service candidate identification example

Identification of single processes as a service candidate

Single processes can be identified as a service candidate if its information objects are coupled as little as possible with information objects used by other processes. In case there are any information objects shared with other processes, these objects have to be as little complex as possible (Yourdon, 1979). To identify the number of information the process under examination (Process i; P_i) shares with other processes, we identify the intersection (I_{it}) of the information objects of P_i and the other processes. Process 1 (P_1) for example shares the information object A with P_2 and the information object B and C with P4. The total intersection of P_i and all other processes (I_{1t}) is composed of A, B, and C. Its modulus ($|I_{1t}|$) results three.

To identify the complexity of information objects shared with other processes, we derive the amount of relationships that is necessary to combine all information objects shared with another process j within the data view of the model. The total amount of all relationships of information objects within the intersection of the process under examination with the other processes composes the total complexity of the process $\sum C(I_{ij})$. For example, the total complexity of P_1 is composed of the number of relationships necessary to combine the information objects within I_{12}, I_{13}, and I_{14}. I_{12} contains the information object A, which is a single object and therefore can not be combined with other objects. I_{13} does not contain any information objects while I_{14} contains B and C as information objects. To combine B and C there is only one relationship required. Therefore the total complexity $\sum C(I_{1t})$ is one. Coupling can be derived with the following ratio:

$$\text{Coupling} = |I_{it}| + \sum C(I_{ij})$$

The less a process is coupled with other processes the more independent is the process and the less an automatisation of the process will influence other processes. Thus, processes with low coupling are possible service candidates. According to our example P_3 will constitute an adequate service candidate as can be seen in Figure 2.

	P_1 (A,B,C)	P_2 (A)	P_3 (D,E)	P_4 (B,C)
P_1 (A,B,C)	-	I_{21}={A} $C(I_{21})$=0	I_{31}= ø $C(I_{31})$ 0	I_{41}={B,C} $C(I_{41})$=1
P_2 (A)	I_{12}={A} $C(I_{12})$=0	-	$S_{3,2}$= ø $C(I_{32})$=0	I_{42}= ø $C(I_{42})$=0
P_3 (D,E)	I_{13}=ø $C(I_{13})$=0	I_{23}= ø $C(I_{23})$=0	-	$I_{4,3}$={E} $C(I_{31})$=0
P_4 (B,C)	I_{14}={B,C} $C(I_{14})$=1	I_{24}= ø $C(I_{24})$=0	$I_{3,4}$=ø $C(I_{34})$=0	-
I_{it}	I_{1t}={A,B,C}	I_{2t}={A}	I_{3t}=ø	I_{4t}={B,C,E}
$\|I_{it}\|$	3	1	0	3
$\sum C(I_{ij})$	1	0	0	1
Coupling	4	1	0	4

Legend:	
I_{it}	Intersection of shared information objects between Process i (Pi) and the other processes.
$\|I_{it}\|$	Modulus of intersection of shared information objects between Process i (Pi) and the other processes.
$\sum C(I_{ij})$	Accumulated number of relationships of the intersections of information objects used by process i and process j, whereas j symbolizes each of the other processes (Total complexity).

Figure 2 Example of coupling (single processes)

A second ratio to identify single processes as service candidate can be derived from cohesion. We derive cohesion of a process from the complexity of its information objects. Thus, a process is characterized by high cohesion if its information objects can be combined with as litte relationsships as possible. The lower the number of relationships is required to combine the information objects the higher results cohesion of the information objects and thus, the more suitable the process serves as a service candidate. To norm processes we introduce N as the number of information objects a process uses and obtain the following ratio:

$$\text{Cohesion}_i = N - C_i$$

We apply this ratio to our example within Figure 3. That way, P_1 uses the information objects A, B and C. These information objects can be combined within the data view using two relationships. Therefore, the complexity of the information objects process 1 uses results two. After the norming process the cohesion of P_i results one, as P_1 uses three information objects that have a complexity of two. The higher cohesion results the more suitable the process results as a service candidate. Within our example P_1 and P_2 form suitable service candidates.

Process	C_i	Cohesion
P_1 (A,B,C)	2	1
P_2 (A)	0	1
P_3 (D,E)	3	-1
P_4 (B,C)	1	-1

Legend:	
C_i	Number of relationships that are necessary to combine the information objects process i has in common with any other process.

Figure 3: example of cohesion (single processes)

Identification of an amount of processes as a service candidate

An amount of processes forms a service candidate if its processes are characterized by high cohesion among the processes and low coupling to outside processes. As introduced in the requirements analysis section services should provide adequat granularity. While a high granularity of services reduces reuse, low granularity constricts comprehension of services as context is lacking (Aier and Schönherr, 2004; Schwinn and Winter, 2005). Therefore, we limit our study to the granularity of two processes that form one total process. This limitation is choosen as the procedure to assess suitability as a service candidate for two processes can be easily adopted for more than two processes by successively adding one process after another. To identify two processes as a service candidate, the number of information objects between two processes P_i and P_j are identified. The combination of processes that have the most information objects in common is selected. The corresponding ratio is:

$$\text{Coupling} = \max |I_{ij}|$$

As demonstrated in Figure 4 P_1 and P_2 as well as P_1 and P_4 share information objects. P_1 and P_4 exhibit the biggest intersection of shared information objects.

	P_1 (A,B,C)	P_2 (A)	P_3 (D,E)	P_4 (B,C)
P_1 (A,B,C)	-	I_{21}={A}	I_{31}= ø	I_{41}={B,C}
P_2 (A)	I_{12}={A}	-	I_{32}= ø	I_{42}= ø
P_3 (D,E)	I_{13}= ø	I_{23}= ø	-	I_{43}= ø
P_4 (B,C)	I_{14}={B,C}	I_{24}= ø	I_{34}= ø	-
$\|I_{ij}\|$	$\|I_{12}\|=1$ $\|I_{13}\|=0$ $\|I_{14}\|=2$	$\|I_{21}\|=1$ $\|I_{23}\|=0$ $\|I_{24}\|=0$	$\|I_{31}\|=0$ $\|I_{32}\|=0$ $\|I_{34}\|=0$	$\|I_{41}\|=2$ $\|I_{42}\|=0$ $\|I_{43}\|=0$
Coupling	2	1	0	2

Legend:	
$\|I_{ij}\|$	Modulus of intersection of shared information objects between process P_i and another process P_j.

Figure 4: example of coupling (an amount of processes)

Processes that can be combined to a single service candidate are furthermore characterized by low coupling to outside processes. For our example combined processes are P_{12} (a combination of P_1 and P_2) and P_{14} (a combination of P_1 and P_4). By combining information objects of P_1 and P_2 to P_{12} and P_1 and P_4 to P_{14} the ratio for identifying single processes as a service candidate can now be applied (Coupling = $|I_{it}|+ \sum C(I_{ij})$). Figure 5 demonstrates that P_{12} exibits a coupling of three and P_{14} a coupling of 1. Therefore P_{14} is prefered to P_{12}.

	P_1 (A,B,C)	P_2 (A)	P_3 (D,E)	P_4 (B,C)	$\|I_{ij}\|$	$C(I_{ij})$	Coup-ling
P_{12} (A,B,C)	-	-	$I_{12,3}$=ø	$I_{12,4}$={B,C}	$\|I_{12,3}\|=0$ $\|S_{12,4}\|=2$	$C(I_{12,3})=0$ $C(I_{12,4})=1$	3
P_{14} (A,B,C)	-	$I_{12,4}$= {A}	$I_{14,3}$=ø	-	$\|I_{14,2}\|=1$ $\|I_{14,3}\|=0$	$C(I_{14,2})=0$ $C(I_{14,3})=0$	1

Legend:	
I_{ij}	Intersection of shared information objects between Process i (Pi) and the other another process j.

Figure 5: example of coupling (an amount of processes)

Processes that can be combined to a single service candidate are characterized by high cohesion. Again we apply a ratio for identifying single processes as a service candidate ($\text{Cohesion}_i = N – C_i$) to on our previously identified pair of processes P_{12} and P_{14}. Figure 6 shows the cohesion for these two processes. P_{12} and P_{14} use the same information objects. Therefore, both process combinations are characterized with the same cohesion of one and can thus not be differentiated by cohesion.

Process	C_i	Cohesion
P_{12} (A,B,C)	2	1
P_{14} (A,B,C)	2	1

Legend:	
C_i	Number of relationships that are necessary to combine the information objects process P_i has in common with any other process.

Figure 6: example of dependent and independent coupling

IMPLEMENTATION

Project description

In 2004 a project was initiated in Germany to consolidate research activities for the development of products made of new compound materials (Huf). Therefore, 11 institutes have joined. E. g. textile engineers focus on the creation of two- or three-dimensional textile out of different kinds of yarn, including glass fiber. Additional several lightweight construction engineers examine technologies to transform the textile into an end-product.

One sub-project focus the support the other groups in structuring, collecting and analyzing data from technical experiments. The data shall be used to ensure constant quality of the production process when the new technologies are transferred from university into practice. The activities of this sub-project include the modeling of the production process itself. Therefore, the modeling language introdced in the design section is uses. To support future research projects, different process models shall be compared to identify reusable process components (Hufenbach).

Process documentation and Generation of Ratios

For the documentation of the processes within the university project, we used the modeling tool “Cubetto Toolset” (Cubetto, 2010). The software was given to us free of charge. It offers three major advantages that were relevant within the project: Firstly, the tool allows the adjustment of the underlaying modeling language. Thus, it was possible to add concepts necessary for the creation of rations. Secondly, distributed modeling of processes is supported by an integrated configuration management (CM) system. Thus, we were able to merge process models of past projects. And thirdly, it is possible to increase the functionality of the tool by using so called plugins. Thus, the creation of ratios could be implemented easily.

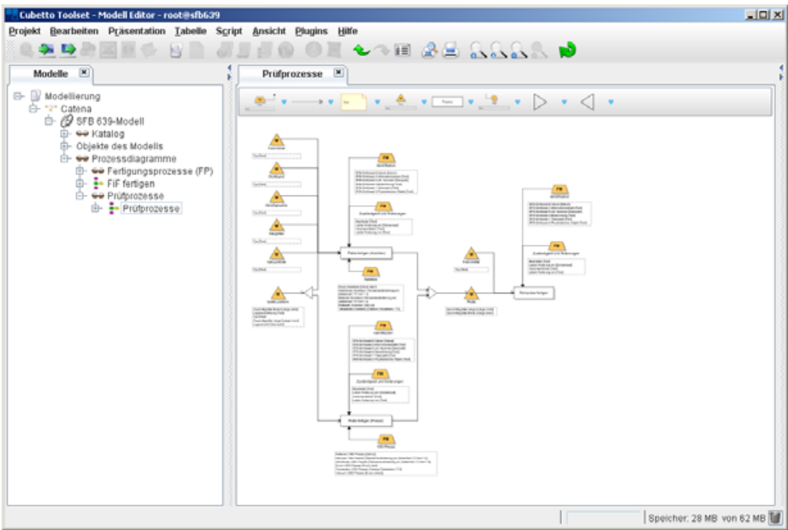

Figure 7: Modeling tool used for the prototypical implementation of the presented approach

The processes were documented directly by the several departments (textile engineers and lightweight construction engineers, e. g.) which were supported by the CM system (see Figure 7). Additionally, processes from previous research projects were documented and integrated using the Cubetto Toolset. Due to the integration it was possible to identify similar process components (services). For implementing the approach, a plugin for the Cubetto Toolset was created. It analysis the integrated model and provides functionality to generate the introduced ratios.

CONCLUSION

In the previous sections we introduced an approach that supports the identification of service candidates out of process models. The feasablity of the approach has been shown by using it within the research project. Thereby candidates for reusable process components (services) have been identified in spite of the large amount of models. Our research has shown, the generated ratios, however, sometimes pointed to processes that were too small for a meaningful reuse or that were to general and needs further process documentation. Reasons for this are the different levels of detail of the documented process models. These problems, however, cannot really be avoided in a distributed modeling environment. Rules can be given when starting the modeling process, but it is not possible to find formal rules concerning the level of detail.

As we stated, our approach currently focuses only three of the presented principles of service identification. Thus, our future research focuses the design and implementation of ratios regarding the principles not being considered for the current approach. Furthermore, we will investigate how the indentified process components can be used when developing new products and appropriate production processes.

REFERENCES

1. Aier, S.; Schönherr, M.: Modularisierung komplexer Unternehmensstrukturen. In Industrie Management 2 (4), 2004, S.39-42
2. Balzert, H.: Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum, Heidelberg, Berlin, 1998; S. 571-574
3. Baldwin, C.Y.; Clark, K.B.: Design Rules, The power of modularity, The MIT Press, Cambridge, 2000
4. Becker, J., Algermissen, L., Falk, T., Pfeiffer, D., and Fuchs, P. "Model Based Identification and Measurement of Reorganization Potential in Public Administrations - the PICTURE-Approach," The Tenth Pacific Asia Conference on Information Systems (PACIS'06), 2006
5. Cardoso, J.; Mendling, J.; Neumann, G.; Reijers, H.: A discourse on complexity of process models. In (Eder, J.; Dustar, S. Hrsg.) Proceedings of the BPM Workshop, 2006
6. Cubetto: Specification, Semture GmbH, Dresden, http://www.semture.de/cubetto. 2010
7. Erl, T.: Service-oriented architecture: concepts, technology, and design, Prentice Hall PTR, 2005
8. Engelke, M.; Rausch, A.: Supply Chain Management mit Hilfe von Key Performance Indikatoren. In (Stölzle, W.; Gareis, K. Hrsg.): Integrative Management- und Logistikkonzepte. Festschrift für Prof. Dr. Dr. h.c. H.-Chr. Pfohl, Wiesbaden, 2002; S. 205-235
9. Ester, B.: Benchmarks für die Ersatzteillogistik: Benchmarkingformen, Vorgehensweise, Prozesse und Kennzahlen. In (Pfohl, C. Hrsg.): Unternehmensführung und Logistik, Erich-Schmidt Verlag, 1997
10. Frank, U.; Jung, J.: Prototypische Vorgehensweise für den Entwurf anwendungsnaher Komponenten. In Proceedings of the Verbundstagung VertIS, 2001.
11. Frank, U.: Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze. In Wirtschaftsinformatik 43 (3), 2001; S. 283-293
12. Fremantle, P., Weerawarna, S., and Khalaf, R. "Enterprise Service - Examining the emerging field of Web Services and how it is integrated into existing enterprise infrastructures," Communication of the ACM (45:10) 2002, pp 77-82.
13. Griffel, F.: Componentware. Konzepte und Techniken eines Softwareparadigmas. dPunkt, Heidelberg, 1998
14. Großmann, K.; Wiemer, H.: Modellierung der Prozesskette für textilverstärkte Verbund-Bauteile: Prozesscharakter, Modellanforderungen und Beschreibungsmittel, In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, (3) 2007.
15. Gruhn, V.; Laue, R.: Complexity metrics for business process modes. In (Abramowics, W.; Mayr, H. Hrsg.) Proceedings of the 9th international conference on business information systems, 2006

16. Haufs, P.: DV-Controlling: Konzeption eines operativen Instrumentariums aus Budgets, Verrechnungspreisen, Kennzahlen. Heidelberg, 1989
17. Hevner, A.R.; March, S.T.; Park, J.; Sam, S.: Design Science in Information Systems Research. In MIS Quarterly 28 (1), 2004; S.75-105
18. Hufenbach, W.: Textilverstärkte Verbundkomponenten für funktionsintegrierende Mischbauweisen bei komplexen Leichtbauanwendungen, Sonderforschungsbereich 639 der DFG an der TU Dresden, 10. Dresdner Leichtbausymposium, Dresden, 2006.
19. Ivanov, K.; Stähler, D.: Prozessmanagement als Voraussetzung für SOA. In OBJEKTSpektrum 12, 2005; S. 1-5
20. Jäger-Goy, H. Führungsinstrumente für das IV-Management, Dissertation, Johannes-Gutenberg-Universität Mainz, Frankfurt am Main, 2001; S. 127
21. Juhrisch, M.; Weller, J.: Connecting Business and IT: A model-driven Webservice based Approach. In: Proceedings of the 12th Pacific Asia Conference on Information Systems (PACIS), 2008, S. 1469-1479
22. McCabe, T.J.: A complexity measure. In IEEE Trans. Software Eng. 2 (4), 1976; S.308-320
23. Parnas, D.L.: Information distribution aspects of design methodology. In (Freiman, C.V.; Griffith, J.E.; Rosenfeld, J.L. Hrsg.): Information Processing 71 (1), North-Holland, 1971; S. 339-344
24. Reichmann, R.; Lachnit, L.: Planung, Steuerung und Kontrolle mit Hilfe von Kennzahlen. In Schmalenbachs Zeitschrift für betriebliche Forschung 28, 1976; S. 705 – 723.
25. Reichmann, T.: Controlling mit Kennzahlen und Managementberichten: Grundlage einer systemgestützten Controlling-Konzeption, 4. Aufl., Vahlen, München, 1995
26. Richter, J.P.; Haller, H.; Schrey, P.: Serviceorientierte Architektur. In Informatik-Spektrum 28, 2005; S.413-416
27. Scheer, A.: Aris - Business Process Modeling. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000
28. Schwemm, J.W.; Heutschi, R., Vogel, T.; Wende, K.; Legner, C.: Serviceorientierte Architekturen: Einordnung im Business Engineering. Working Paper of the HSG, University of St. Gallen, 2006.
29. Schwinn, A.; Winter, Ro: Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration. In (Ferstl, O.K.; Sinz, E.J.; Eckert, S.; Isselhorst, T. Hrsg.): Wirtschaftsinformatik. Physica-Verlag, Heidelberg 2005
30. Simon, H.A.: The architecture of complexity. In Proceedings of the American Philosophical Society 106, 1962.
31. Staudt, E.: Kennzahlen und Kennzahlensysteme. Grundlagen zur Entwicklung und Anwendung. Berlin, 1985; S. 20-24
32. Verschuren, P.; Hartog, R.: Evaluation in Design-Oriented Research. In Quality & Quantity 39, 2005; S. 733-762
33. Wissenbach, H.: Betriebliche Kennzahlen und ihre Bedeutung im Rahmen der Unternehmerentscheidung – Bildung, Auswertung und Verwendungsmöglichkeiten von Betriebskennzahlen in der unternehmerischen Praxis. Berlin; S. 15, 1967
34. Wand, Y., Weber, R.: An ontological model of an information system. In IEEE Trans. Software Eng. 16 (11), 1990; S.1282-1292
35. Yourdon, E.: Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Prentice Hall, Upper Saddle River, 1979.