Resource-oriented Consistency Analysis of Engineering Processes*

Bernhard Bauer, Thomas Eisenbarth, Christoph Frenzel and Benjamin Honke
Programming Distributed Systems Lab, Institute of Computer Science, University of Augsburg, Augsburg, Germany

Keywords: Ontologies, Method Engineering, Process Management, Enterprise Knowledge Management.

Abstract: A number of popular engineering processes and methodologies emerged over the past years which attracted interest in research and industry. For process enactment, enterprises have to match requirements of engineering processes with existing resources. Therefore, it has to be analyzed if employees have sufficient skills or if infrastructure has sufficient capabilities to match the requirements of a situational process. Both, maintaining a skill and resource database and matching requirements against this database are an enormously challenging and time-consuming task. Additionally, this matching has to be carried out regularly as employees join and leave the company and skills change over time. Based upon a formal, logical base, we present an approach that combines ontological and metamodeling technical spaces and enables an automatic analysis of engineering processes regarding the existence of necessary resources.

1 INTRODUCTION

Today, a reasonable number of standardized software engineering methodologies, such as the V-Model XT (Friedrich et al., 2008) or the automotive-specific methodology of AUTOSAR (AUTOSAR Consortium, 2011), and reference processes, such as SPICE (Dorling, 2007) or CMMI (CMMI Product Team, 2011), exist for various domains. They usually define engineering methods by combining process knowledge about input/output products and required resources, such as technical infrastructure (e.g., tools and laboratories) and human skills (e.g., roles and responsibilities). Usually, enterprises can choose from a variety of skills and a large number of infrastructure components providing lots of capabilities, which can be used to fulfill project-specific requirements and to put processes into action.

Matching the requirements of engineering processes with existing enterprise resources is a people-intensive and error-prone task. It is not only necessary to have in-depth domain knowledge about skills, which are necessary to fulfill certain roles and responsibilities, but also innovative methodologies often create new resource demands. Additionally, resources and skills in a company change frequently due to shifting business and IT requirements, staff fluctuation, and employee training. Therefore, matching the necessary resources of a process with the available resources in a company has become a regular task in order to decide whether a development process can be executed.

In this paper, we present an approach for the automatic resource-oriented analysis of engineering processes w.r.t. the required resources of a process and available resources in a company. This is achieved through the utilization of semantic technologies, especially logical reasoning. Therefore, we combine two models, which are expressed in two separate Technical Spaces (TSs): process models are normally represented in the Metamodeling Technical Space (MMTS), whereas a model of the existing resources is expressed in the Ontological Technical Space (OTS). Both models are linked via a bidirectional mapping which translates between both TSs. In contrast to complex resource planning approaches, our resource analysis approach is supposed to offer a fast and lightweight evaluation, if a company has, in principle, the necessary resources to execute a given process. That is, instead of planning the execution of process actions, our approach validates, that at least one resource exists for each resource requirement of a process. This means, that we neglect time constraints induced by, e.g., concurrently executed process actions or processes. Additionally, this focus enables the presented approach to be used with less detailed processes, as well. The result of the resource analy-

*Grateful acknowledgement is due to the DFG (German Research Foundation) for their support of the project BU 809/7-2 SEMPRO[2] making this paper possible.

Bauer B., Eisenbarth T., Frenzel C. and Honke B..
Resource-oriented Consistency Analysis of Engineering Processes.
DOI: 10.5220/0003984302060211
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 206-211
ISBN: 978-989-8565-12-9

sis is a set of non-executable process actions and the according missing resources.

Using the presented approach, companies are capable of facing the following challenges:

1. *Assurance of Consistency of Particular Project Requirements with Internal (or External) Resources*. Even prior to project realization, project managers must know, whether or not a project-specific process can be realized based on existing resources. By applying our resource-based analysis, an automated and detailed consistency evaluation becomes possible. Therefore, project teams with fitting skills can be arranged more purposefully.
2. *Assurance of Standard Compliance According to Existing Resources*. Process improvement frameworks, such as SPICE or CMMI, define the availability of particular artifacts and the execution of dedicated activities. This often neglects that all activities are enabled and executed by resources, which must exist, too. By applying the automated resources analysis, the feasibility (often also referred to as consistency) of all activities and the artifacts can be ensured.
3. *Provision of a Company- or Customer-specific Resource Delta Analysis*. By applying a company-wide resource analysis, it is possible to determine missing resources in a company to reach a particular compliance level or some other resource-related goal.

2 FOUNDATIONS

As Process Models (PMods) often lack semantical knowledge, our approach combines benefits of the MMTS and the OTS (cf. (Kurtev et al., 2002)) as introduced in the following sections.

2.1 Ontological Technical Space

Originally designed for knowledge representation, semantic technologies are built upon logical structures, which are used by specialized reasoning software to check for consistency and to infer new facts. These knowledge representations are called ontologies, which in turn describe entities and relations in a specific domain (Uschold and Gruninger, 1996).

Accepted standards defined by the W3C include Resource Description Framework (RDF), which is used for conceptual description and modeling of information. Web Ontology Language (OWL) and the subsequent version OWL 2 are languages for more powerful knowledge description in ontologies as well (Grau et al., 2008).

A number of popular modeling tools for ontologies such as Protégé, NeOn Toolkit or KAON for intuitive definition of information and relations within a ontology exist. Subsequently such tools were heavily used to build large ontologies. Related technologies, standards, and tools for this knowledge representation quickly emerged, and established this branch of research area as a well-recognized area of computer science over the past years.

2.2 Method Engineering and the Metamodeling Technical Space

In software engineering, Method Engineering (ME) is a discipline, which concerns the design and management of product development from the inception of a product to its maintenance. ME intends to provide contingent ways of thinking, i.e., methods, which realize concrete products, artifacts, or other deliverables, in an appropriate way (Henderson-Sellers and Ralyté, 2010). Several methods and their outcomes are combined in the course of an project-specific situational process, afterwards.

For structuring relevant facets of a method, authors, such as Mirbel et al. (Mirbel and Ralyt, 2005), introduced the term Method Chunk (MC). According to the meta model, which was proposed by Henderson-Sellers et al. (Henderson-Sellers and Gonzalez-Perez, 2008) (cf. 1 (a)), a MC aggregates different types of method fragments. Originally, a MC was defined to consist of a product fragment to define relevant information objects, and a process fragment to describe the steps for manipulating information in a required manner. However, various authors (cf. (Harmsen, 1997; IEEE Computer Society Press, 2001)) have recognized the need for additional fragments to detail a method considering its required resources, such as roles (i.e., human-related factors) and tools (i.e., infrastructure-related factors). For example, in the well-known model definition language Business Process Modeling Notation (BPMN) 2 (Object Management Group, 2011), resources could be expressed by using swimlanes. Although, this way to model resource requirements is not capable of complex skill definitions and lacks further semantics.

3 APPROACH

In a nutshell, our approach for analyzing the feasibility of processes is the following: given a model of

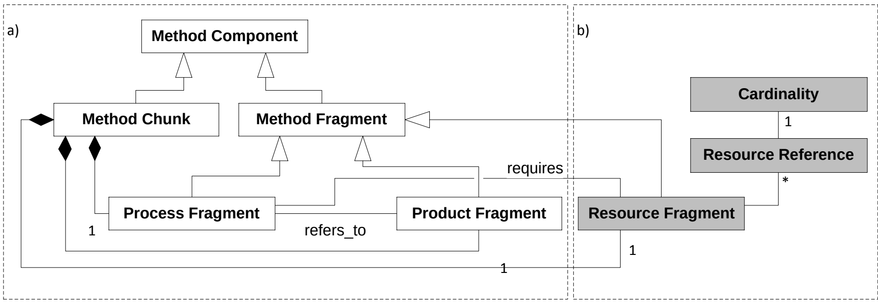

Figure 1: Meta model for method components according to (Henderson-Sellers and Gonzalez-Perez, 2008) extended by Resource Fragment.

the process and a semantic model of the available resources within a company or department, we utilize semantic technologies, especially logical reasoning, in order to come up with infeasible MCs and the respective missing resources. Therefore, both MMTS and OTS need to be integrated via a mapping. 2 depicts an overview of the concept with its three building blocks: *PMod*, *Resource Ontology (ROnt)*, and a *mapping* between these two worlds.

3.1 Process Model

The metamodel of Henderson-Sellers et al. (Henderson-Sellers and Gonzalez-Perez, 2008) does not consider the definition of resource requirements for MCs. Therefore, we extended the metamodel with a *Resource Fragment (RF)* as shown in 1 (b). The additional fragment indicates, that the selection of a MC is affected by resource-related factors, as well. Similar to product and process fragments, the RF is a container that holds detailed resource-related information, which in turn depend on the applied process definition metamodel. Put simply, an RF contains various resources, which are necessary to support the realization of the assigned MC. The information about resources is twofold: on the one hand, the RF holds an Resource Reference (RR) which represent specific roles or tools, and, on the other hand, an RR provides a cardinality attribute to indicate the required quantity of the respective resource. For example, a method such as *double-blind review* needs two roles of the same type to realize the four-eyes principle.

The left part of 2 shows the PMod, which represents a concrete process on a syntactical level in the MMTS, in the context of our approach. The process is realized using MCs, which are composed of three types of method fragments. Because of space limitations, we solely depict RRs, which are assigned to the respective MC via a not shown RF using a requires relation. The RR symbols show both, the cardinality as a hash symbol # on top and the name of the resource below. Note that, for clarity reasons, we are distinguishing between roles, i.e., employees with specific skills, and tools, i.e., infrastructural capabilities. However, since the approach presented here is generic regarding the concrete method definition language, this is not a requirement for a potential method definition approach.

3.2 Resource Ontology

The right part of 2 depicts the ROnt, i.e., a model of the resources within a company or department represented in the OTS. On the one hand, this model conceptualizes company-internal resources, and builds up a *resource taxonomy* of skills and infrastructural capabilities. On the other hand, it also comprises the *resource pool* of the company, i.e., concrete employees and infrastructure. By putting both in one ontology, it is possible to assign each employee to the skills he or she has and, respectively, assign each tool to the capabilities it provides.

The resource taxonomy is a hierarchical definition of the skills and capabilities that exist within a company. Thereby, skills refer to concrete knowledge, capabilities, or access rights an employee might have, e.g., tax law knowledge, project management skills, or access rights to personnel files. In order to come up with a reasonable taxonomy of the skills, they should be categorized and aggregated to umbrella terms. For instance, the skills Java programming and C programming are classified under the skill programming languages. The same ideas apply to the taxonomy of capabilities. A capability refers to a functionality of infrastructure components within the company. The information for building the resource taxonomy can be gathered from experiences or knowledge of experts, or from bodies of knowledge like (Abran et al., 2005).

The resource pool is a collection of concrete

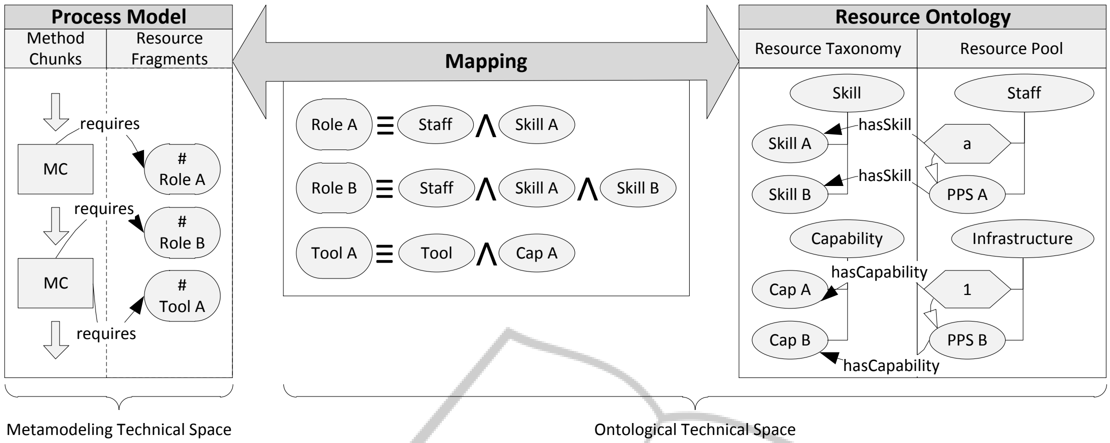

Figure 2: Concept of the resource-oriented analysis of processes.

members of staff and instances of tools (depicted as hexagons in 2) within the company or department. For instance, there can be a representation of the real employee Bob and the Laboratory X. Each member of staff and tool is linked to the skills and capabilities it provides via a hasSkill or hasCapability relationship, respectively. This allows the concrete specification of the abilities of the company's resources. Furthermore, there can be concepts, which define a specific set of skills or capabilities, respectively. They are called *Predefined Property Sets (PPSs)* and depicted as ellipses in 2. For instance, a concept software architect can define that an assigned employee has both a programming language skill and project management skill. This way, it is possible to represent company-specific job titles. Actually, PPSs can be seen as shortcuts for skill assignment by assigning a member of staff to them instead of single skills.

All in all, the ROnt acts as a skill database for human resources and, additionally, contains equipment and tools including their respective capabilities. This knowledge base is developed for a company once, and has to be kept up to date. Subsequently, it can be exploited in our approach to validate the feasibility of processes from a resource-oriented point of view.

3.3 Mapping

In order to exploit the ROnt for a feasibility analysis of a process, a translation between the RRs in the PMod and the skills and capabilities in the resource taxonomy is necessary. This is provided by the mapping depicted in the middle of 2. It is a separate ontology and, therefore, technically belongs to the OTS. The basic idea of the mapping is to define a RR as an aggregation of concepts from the resource taxonomy, i.e., skills or capabilities.

A mapping between concrete RRs and concrete skills or capabilities is represented as an ontological equivalence. Each RR is represented as a distinct concept in the mapping. Thereby, the matching between RRs in the OTS and in the MMTS can be based on, e.g., name equality. Now, the ontological RR is defined to be equivalent to either an employee with a set of skills or a tool with several capabilities. An example of a mapping is:

software architect is equivalent to Staff
and hasSkill software_architecture
and hasSkill project_management .

To enable the usage of the resource taxonomy in the mapping, it imports the whole ROnt. Hence, no specific matching between skills and tools in the mapping, and in the resource taxonomy is necessary because both reside in the OTS.

In general, the set of skills and capabilities for a RR is interpreted as a conjunction, i.e., the employee or tool has to have *all* stated skills or capabilities. However, depending on the expressiveness of the utilized ontology language, this simple semantics can be extended by more complex structures like disjunctions. This allows to express a logic like a requirements engineer has a skill in use cases or user stories. Note that the mapping defines a translation for RRs, hence, the cardinalities for the RFs are neglected.

The mapping can be seen as the definition of a matching between the RRs, i.e., RFs, of a specific method definition language and the resource taxonomy of a company. Hence, the approach of decoupling both PMod, and ROnt allows reusing the company's body of knowledge for analyzing the feasibility for processes independently from the used method definition language. It is solely necessary to define the mapping once for each combination of method definition language and resource taxonomy. Note that, however, it has to be ensured that the names of the RRs in the language are unique, i.e., no two different RRs have the same name.

3.4 Ontology-based Resource Analysis

Once the ROnt,i.e.existing resources within a company or department, and the mapping are defined as outlined above, this knowledge can be exploited to analyze the feasibility of a project-specific situational PMod w.r.t. the required resources of contained MCs. Thereby, the definition of the ROnt and the mapping in the OTS turns out to be an enormous advantage due to reasoning which enables the automation of this task.

The algorithm for computing feasibility of process models works as follows: Given some PMod, it returns the set of MC–RR pairs, which cannot be fulfilled by any employee or tool from the resource pool. In a nutshell, it checks whether each and every MC in the process definition can be executed, i.e., whether there are enough resources in the resource pool for the RF assigned to the MC. Specifically, the feasibility is analyzed by querying all resources from the resource pool, i.e., staff or tools, which match the required RRs of the RF and comparing their count with the cardinalities.

In order to gather the number of resources, which match a RR, the algorithm queries the ROnt. Behind the scenes, this query matches the RR in the MMTS to the RR concept in the mapping, i.e., the OTS. As outlined in 3.3, this can be done using name equality. Subsequently, this concept can be translated via the mapping into either an employee with a given set of skills or a tool with a specific set of capabilities. Using logical reasoning, the resource pool can now be queried for all resources, i.e., staff or tools, which fulfill this definition. The resulting resources are counted and their number returned to the algorithm.

The result of the algorithm is, that all infeasible MCs with the respective RRs, that cannot be fulfilled by the resources in the resource pool, are marked as infeasible in the given PMod. This information can be utilized by the user to either redesign the process, e.g., replacing the infeasible MCs, planning a training program for developing the missing skills, or purchase new or improve the available tools. Note, that the current algorithm does not provide the specific resources from the resource pool, which can perform a given MC. However, it is straightforward to exploit the ROnt and mapping for this task, as well.

4 DISCUSSION AND RELATED WORK

The need to conceptualize enterprise knowledge has been identified some time ago (Fox and Gruninger, 1998). However, most of this work focuses on either the MMTS or OTS only. On the one hand, methodological frameworks were developed to aggregate the particular interests of a company, such as applied processes, resources, guidelines, or products, and to benefit from available information in future projects. Some approaches are used to validate new PMods, or to evaluate the feasibility of projects by comparing company-specific capabilities with project-specific needs (Gruhn, 1991; Hug et al., 2008). They validate regarding deadlocks, availability of artifacts, or the reachability of dedicated goals based on the static structure of PMods. While those approaches do validate the availability of human or infrastructural resources through simple structural syntactical checks (e.g., comparing identifiers), most of them neglect the advantages of incorporating semantic knowledge and formal definition of such.

There has been work done presenting an interesting possibility to advance ways to model resources within graphical notation languages (BPMN 2 in this case). Although, it covers humans resources only (Cabanillas et al., 2011).

On the other hand, some approaches exclusively focus on the need for managing company-internal or -external resources using semantic technologies (cf. (Fadel et al., 1994; Dorn et al., 2007; Hepp and Roman, 2007)). Although, these approaches provide frameworks for managing resources, to the best of our knowledge, none of these relates engineering PMods with ontological resource management.

Through loose coupling between MMTS and OTS, existing PMods and ontologies can be used without expensive modification or information loss. Besides the definition of the mapping, efforts can be reduced to a minimum, and well-known design principles can be used in both TSs. The application of reasoning techniques reduces the efforts to search for particular resources manually. By determining the capabilities and skills of a specific resource concept in the ontology, all available resources of a company can be classified automatically, which provides project managers with most recent information about available resources and bottlenecks regarding the needs of some project. Even monitoring activities may benefit from our approach: as the capabilities of particular resources can automatically be inferred, resource fluctuations, i.e., changes in the resource pool, directly influence actual projects by propagating relevant changes to project managers, which are responsible for a respective mapped process fragment.

5 CONCLUSIONS

In this paper, we proposed an approach to analyze engineering processes regarding their feasibility based on a semantically-defined resource ontology. We successfully evaluated our approach in a separate case study, which we removed here due to space limitations. Our approach demonstrated, that process models from the MMTS can be combined with semantical knowledge from the OTS to assure, that process resource demands are met. Furthermore, reasoning within a formal resource ontology reduces efforts, which normally are necessary to manage highly fluctuating company resources. We presented two possibilities of defining skills for resources and defined a matching approach between the TSs. While most similar approaches allow definition and consideration of human resources only, our approach allows definition of all kinds of resources.

Our future research will focus on resource-oriented planning and scheduling of engineering process. Therefore, we plan to analyze control flow mechanisms of workflows to provide more detailed assertions about timing behavior or to optimize costs based on existing resources. That way, more sophisticated analyses of the processes' resource factors should help to efficiently tailor processes not only regarding their application scenarios, but also considering available resources. By considering the average processing time of MCs gathered from workflow engine logfile analysis as well as resource existence, we plan to expand our approach to provide sophisticated resource scheduling functionalities.

REFERENCES

Abran, A., Moore, J. W., Bourque, P., and Dupuis, R., editors (2005). *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society Press.

AUTOSAR Consortium (2011). AUTOSAR. Official website of the AUTOSAR development partnership.

Cabanillas, C., Resinas, M., and Ruiz-Cortés, A. (2011). Towards the definition and analysis of resource assignments in BPMN 2.0. *International Conference on BPM*.

CMMI Product Team (2011). CMMI. CMMI Main page managed by SEI.

Dorling, A. (2007). Spice: Software process improvement and capability determination. *Software Quality Journal*, 2(4):209–224.

Dorn, J., Naz, T., and Pichlmair, M. (2007). Ontology development for human resource management. In Stary, C., Barachini, F., and Hawamdeh, S., editors, *Knowledge Management Innovation Technology and Cultures*, pages 109–120. World Scientific Publishing Co. Pte. Ltd.

Fadel, F. G., Fox, M. S., and Gruninger, M. (1994). *A generic enterprise resource ontology*, pages 117–128. IEEE Comput. Soc. Press.

Fox, M. S. and Gruninger, M. (1998). Enterprise modeling. *AI Magazine*, 19(3):109–121.

Friedrich, J., Hammerschall, U., Kuhrmann, M., and Sihling, M. (2008). *Das V-Modell XT*. Springer.

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., and Sattler, U. (2008). OWL2: The Next Step for OWL. *Journal on Web Semantics*, 6(4):309 – 322.

Gruhn, V. (1991). Validation and verification of software process models. In Endres, A. and Weber, H., editors, *Software Development Environments and CASE Technology*, pages 271–286. Springer.

Harmsen, A. F. (1997). Situational method engineering. *Requirements Engineering*, 5(9):167–78.

Henderson-Sellers, B. and Gonzalez-Perez, C. (2008). *Comparison of Method Chunks and Method Fragments for Situational Method Engineering*, pages 479–488. IEEE Computer Society.

Henderson-Sellers, B. and Ralyté, J. (2010). Situational Method Engineering: State-of-the-Art Review. *Journal Of Universal Computer Science*, 16(3):424–478.

Hepp, M. and Roman, D. (2007). An ontology framework for semantic business process management. *Information Systems Journal*, pages 1–18.

Hug, C., Front, A., and Rieu, D. (2008). A process engineering method based on ontology and patterns. In *ICSOFT (ISDM/ABF)'08*, pages 29–36.

IEEE Computer Society Press, editor (2001). *Model driven process engineering*.

Kurtev, I., Bezivin, J., and Aksit, M. (2002). Technological spaces: An initial appraisal. In Editors, editor, *Language*, volume 2002, pages 1–6.

Mirbel, I. and Ralyt, J. (2005). Situational method engineering: combining assembly-based and roadmap-driven approaches. *Requir. Eng.*, 11:58–78.

Object Management Group (2011). Business process model and notation. *OMG document*.

Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136.