What's inside the Box? Prospects and Limitations of Semantic Verification in Process Modeling

Michael Fellmann[1], Frank Hogrebe[2], Oliver Thomas[1], Markus Nüttgens[2]

[1]Universität Osnabrück,
Institut für Informationsmanagement und Unternehmensführung,
Fachgebiet Informationsmanagement und Wirtschaftsinformatik,
Katharinenstraße 3, 49069 Osnabrück
{Michael.Fellmann|Oliver.Thomas}@uni-osnabrueck.de

[2]Universität Hamburg,
Fakultät Wirtschafts- und Sozialwissenschaften,
Lehrstuhl für Wirtschaftsinformatik,
Von-Melle-Park 5, 20146 Hamburg
{frank.hogrebe|markus.nuettgens}@wiso.uni-hamburg.de

Abstract: Business process models support the optimization, reengineering and implementation of IT applications in administration and economics. In this area, models are important to manage complexity. In order to realize their full potential, the correctness of business process models is of significant importance. The paper applies an ontology-driven approach that aims at supporting semantic verification of process models. The approach is based on the formalization of the semantics of individual model elements by annotating them with concepts of a formal ontology. In order to ensure semantic correctness, semantic verification rules are introduced and it is demonstrated how machine reasoning provides for the automation of verification tasks. The approach is demonstrated using real-life process models taken from a capital city.

1 Introduction

Process models serve for optimization, reengineering and implementation of supporting IT systems. Due to the importance of process models, model quality is important. According to ISO 8402, quality is "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs". Facets of quality are – amongst others – adequate coverage of the domain or system to be modeled, appropriateness in respect to the abstraction level of the representation (scale), detail of representation (granularity) and the correctness of a model. We concentrate on correctness as the most fundamental quality aspect. Among the aspects of correctness are essentially:

(a) syntactical correctness,
(b) correctness in regard to the formal semantics,

(c) correctness in regard to linguistic aspects focusing on the labels used in models,
(d) correctness in regard to the coherence of connected models and
(e) compliance to rules and regulations focusing on the correctness of the model's content and thus on semantic correctness.

While there are numerous verification approaches available to ensure (a-d), only a few approaches focus on (e) in the sense of the verification of the semantic correctness. With the term "verification", we denote criteria targeting the internal, syntactic and semantic constitution of a model. In contrast to that, validation means the eligibility of a model in respect to its intended use [De02, p. 24] – in other words: if the criterion is something outside the model [CB08; Me09, p. 2]. Following this distinction, we call the procedures to ensure semantic correctness "semantic verification". The proposed approach of semantic verification allows performing additional checks on process models. Such checks are possible by annotating process models with instances of a formal ontology containing terminological knowledge of the domain under consideration. The ontology in conjunction with an inference engine can then be used to automatically verify several aspects of models based on the semantics of the individual model elements. This decoupling from human labor makes semantic verification scalable even in incremental approaches to model construction where a model has to be re-verified repeatedly. An important additional benefit thereby is that the semantic verification rules can be formalized on a more abstract and generic level and the inference engine interprets them with the help of both explicitly encoded and inferred knowledge from the ontology. Therefore, it is possible to formulate semantic verification rules in a more natural and understandable way that accommodates to the nature of generic rules such as guidelines, conventions, recommendations or laws being rather abstract in order to ensure broad applicability.

The paper is organized as follows. In the related work section, we provide an overview of approaches and tools in the state-of-the-art of model verification. In the next section, we present a case study that motivates our approach. In the section "Ontology-driven Approach for Semantic Verification", we present our approach of semantic verification along with a rule classification and examples illustrating the application of such rules to the real-world problems of the case study. In the section "Limitations of Semantic Verification", we describe the limitations of semantic verification and in the last section, we summarize our work and look at future research.

2 Related Work

There are a lot of related works on procedures and methods for the verification of behavioral descriptions (see table 1). They partly originate from software engineering [Gr91] where they are discussed under the terms "model checking" and "theorem proving" [Cl08; CB08]. These approaches mainly concern dynamic aspects of model execution which are verified using finite state automata (FSM). A major problem to be tackled here is the explosion of the state space which is solved or alleviated by symbolic representations [XBS02] or reduction procedures [ADW08]. Approaches that aim at the verification of software-related systems and processes [Gr91], [CCO00], [DAC04], [Ba08] are sometimes also transferred to conceptual modeling [Va03]. Clearly, as we are engaged in verifying conceptual models, software processes are out of scope. Another research area is the verification of workflows [vA97], [SO00], [SOS05], [SCA07], [TBB08] where notions such as “soundness” [DvA04], “relaxed soundness” [vA98], “weak soundness” [Ma03] and “lazy soundness” [PW06] have been developed. Recent research extends workflow management to the verification of web services orchestrations [Na02], [XBS02], [FUM03], [GXS04], [vBK06], [KG08], [TBB08], [Ok09], [AM09].

Research Areas of Semantic Verification	Related Work in this Research Areas	Characteristics of the Research Approaches
Software Engineering	Awad et al. 2008 [ADW08], Chapurlat/Braesch 2008 [CB08], Clarke 2008 [Cl08], Gruhn 1991 [Gr91], Xiang et al. 2002 [XBS02]	These approaches mainly concern dynamic aspects of model execution. Semantic Verification is discussed under the terms “model checking” and “theorem proving”.
Software-related Systems and Processes	Barjis 2008 [Ba08], Cobleigh et al. 2000 [CCO00], Dong et al. 2004 [DAC04], Gruhn 1991 [Gr91], Varró 2003 [Va08]	The approaches are sometimes transferred to conceptual modeling, but most times concentrate on the verification of software processes and software process models.
Verification of Workflows	Dehnert/van der Aalst 2004 [DvA04], Sadiq/Orlowska 2000 [SO00], Sadiq et al. 2005 [SOS05], Salomie et al. 2007 [SCA07], Touré et al. 2008 [TBB08], van der Aalst 1997 [vA97], van der Aalst 1997 [vA98]	Workflow management emphasizes the execution semantics or runtime aspects of workflow models.
Verification of Web Services Orchestrations	Abouzaid/Mullins 2009 [AM09], Foster et al. 2003 [FUM03], Geguang et al. 2004 [GXS04], Kovacs and Gönczy 2008 [KG08], Nakajima 2002 [Na02], Okika 2009 [Ok09], Touré et al. 2008 [TBB08], Van Breugel/Koshkina 2006 [vBK06]	Similar to workflow management, the verification of web services orchestrations is mainly concerned with the execution semantics. Aspects of dynamic selection and invocation and their consequences in respect to verification are also discussed.
Verification of conceptual Process Models	Dijkman et al. 2007 [DDO07], Dijkman et al. 2008 [DDO08], Mendling 2009 [Me09], Mendling/v.d.Aalst 2008 [MvA08], van der Aalst 1997 [vA99]	In the area of the verification of conceptual process models, independent verification criteria have been developed such as „soundness” or „relaxed soundness“ which are used to detect shortcomings regarding the formal semantics.
Labels of conceptual process model elements	Becker et al. 2009 [BDH09], Friedrich 2009 [Fr09], Leopold et al. 2009 [LSM09], Mendling et al. 2010 [MRR10], Peters/Weidlich 2009 [PW09]	Goal of these approaches is improving the model quality and comprehensibility. This might be achieved by conforming to specific naming conventions or using glossaries.

Table 1: Research Areas of Semantic Verification

In the area of the verification of conceptual process models, formal criteria developed in the workflow management community have been adapted and extended resulting in independent criteria and definitions of e.g. „soundness“, „relaxed soundness“ or „well-structuredness“ which are used to detect shortcomings such as deadlocks, livelocks, missing synchronisations and other defects regarding the formal semantics [vA99], [DDO07], [MvA08], [DDO08], [Me09]. The verification in this sense abstracts from the individual semantics of model elements which is given by natural language and concentrates on formal procedures. Therefore, the formalization of the semantics of semi-formal process models is suggested [Ki06] and formal languages such as Petri Nets are heavily used.

Research regarding formal verification of conceptual models in general is still an active field. New approaches consider e.g. the verification of access constraints in semi-formal models [WMM09], the verification in the context of hierarchical models [vA03], [SCA07], [GL07], the consistency of related models which is discussed mainly in the realm of the Unified Modeling Language (UML) [EHK01], [VP02], [HKT02], [KTK02], [CKS10], lightweight approaches which do not rely on a formal language such as Petri Nets [PG08] as well as the verification of aspects related to the context of models. Such contextual aspects range from access rights [WMM09] to process artifacts such as business objects which participate in a process [XWW09], [KG07], [LDH08], [DHP09], [GS09] or goals and outputs produced by a process [SW04], [LBL06]. Also, metrics are studied mainly for error prediction and evaluation purposes [ARG06], [MNA07]. Further aspects include the quality of the model construction process [NM07] and interactive approaches for verification based on reduction rules [vDA05], [vDJ0J]. While we address the semantics of individual model elements which is expressed using natural language labels, these approaches do not consider this sort of semantics.

Verification of the labels of conceptual process model elements has gained substantial research interest during the last few years, mainly in respect of improving the model quality and comprehensibility [MRR10], [Fr09]. This might be achieved by conforming to specific naming conventions [LSM09] or using glossaries [BDH09], [PW09]. Although this stream of research is addressing the natural language labels of model elements, in contrast to our approach it does not try to formalize the semantics of model elements using formal knowledge representations such as ontologies. Instead, most of such approaches are concerned with purely linguistic analysis and tools such as word stemming, part-of-speech tagging and other approaches belonging to the field of computational linguistics.

Verification beyond formal semantics and linguistic aspects is discussed e.g. in the context of compliance. Compliance can be understood as the conformity of something such as a process model to the entirety of relevant legal liabilities, directives and rules as well as to the internal guidelines and best practices of an enterprise. This clearly goes beyond syntax and formal semantic and requires also checking the individual model elements and their semantics often expressed using natural language. We call approaches in this direction “semantic verification approaches” as they touch the content or the subject matter of individual model elements in a process model, i.e. “what happens”. While most approaches aim at detecting compliance violations caused by the model structure or execution semantics [SPH04], [GV06], [LMX07], [ADW08] by violating a prescribed modeling style [GL07i], [GL07ii] and hence enforce compliance prior to model implementation or execution, some approaches also aim at verifying running processes. Thus, they tackle the problem of changes of the conceptual model (i.e. the process schema) [LRD08] or process instances [LRD10] as well as analyzing already finished processes ex ante on the basis of process logs [vAB05]. Although these approaches address aspects of semantic correctness and partly make use of machine reasoning, they are different from our work as they do not make use of a knowledge structure such as an ontology expressed in a description logic or fragment of first order logic and the deductions which are made possible by using such representations in conjunction with an inference engine.

First approaches to ontology-based semantic verification can be found in the context of Semantic Web Services. Semantically annotated process models are verified with an emphasis on logical preconditions and effects which are specified relatively to an ontology [DFH07], [We09]. These approaches usually require both the annotation of preconditions and effects and hence enable to check if the model is consistent. They do not build upon a formal representation of the (intentional) semantics of individual model elements (i.e. what is “inside the box” of a model element). Following this argument, a function “receive guest” and “welcome guest” in a hotel service process may have the same preconditions and effects, but differ considerably. Our approach enables to capture such differences by using a single annotation of a model element in order to associate it with its intended meaning explicitly specified in a formal ontology. Semantic verification rules then allow to check if a model complies with a set of requirements using this explicitly specified meaning along with the deductions that are possible due to its formal representation. Our approach is therefore orthogonal to approaches considering preconditions and effects (i.e. what is “outside the box” of a model element). So far, there are only a few approaches using rules together with semantic process descriptions [Go08], [WHM10], [TF09] as well as frameworks for semantic verification related to compliance [ES08], [ESM08].

The dynamics of verification rules are a major problem of semantic verification approaches in general. On the one hand, rules are required to be detailed enough to be useful for verification of concrete models describing specific processes and on the other hand, rules are required to be generic enough to be applicable for a set of such processes. In contrast to rules and procedures for formal verification e.g. to detect deadlocks, rules for semantic verification more often change. This is the case since the subject matter is not (in contrast to formal semantics) the relatively stable modeling language and its use, but rather the content of the models expressed in a modeling language. As such, it is exposed to frequent changes due to the dynamics of the contemporary economic and legislative world (e.g. minimum age for customers of a product may change due to law). Some efforts address this problem area of changing rules and suggest graphical modeling languages such as BPSL (Business Property Specification Language) [LMX07] or to capture the required rules implicitly by providing negative examples [SM06]. Furthermore, patterns are sometimes discussed in this context [SPH04], [NS07]. However, despite such improvements in rule capturing a fundamental problem is still, that most approaches require a fine grained specification of rules conflicting with the rather abstract nature of compliance rules in the sense of guidelines, best practices and general principles. We extend the state-of-the-art by showing that ontology-based representations of process models provide for the formulation of generic verification rules which are applied to concrete process models using an inference engine in order to automate semantic verification. We apply our approach to real-world problems and therefore demonstrate that semantic verification is feasible and useful to solve real world problems.

3 Case study

The municipality we took as our case is one of the biggest cities in the country we accomplished our research in (region capital city). It has about 580,000 inhabitants and the public administrative authorities are employing about 9,100 employees, distributed over about 440 administration buildings. The structure is decentralized and subdivided into seven departments each with 48 assigned offices and institutes. Based on Fat Client Server architecture, the 6,000 IT-jobs are workplace-based and completely linked to each other via a communication system throughout the city. In view of the increasing international competition, the city is requested to rearrange its product and process organization, particularly as the support of enterprise-related activities becomes increasingly an essential position factor in the international competition. In the city, about 99% of the enterprises have less than 500 employees and can be considered as small or medium-sized enterprises, these are about 40,000 enterprises. The high objective of the city is to make the place even more attractive for enterprises in terms of their competitiveness with long-lasting effect. This shall be achieved by making the enterprise-related offers and services of the city even easier to access for enterprises, in terms of a One-Stop eGovernment. To reach this goal, the city has to model about 550 enterprise-related administrative processes. The process setting is highly relevant for the capital city, because several of the procedures are used about 15,000 to 25,000 times per year by the companies.

After having starting the project, we detected several inconsistencies in the collected data. Subsequently, we describe the modeling problems that we detected regarding resource problems. The administrative process models had several errors (E) regarding the correct usage of resources. Subsequently, we show two core modeling errors (E1, E2) in this area:

- (E1) Usually, process activities are executed by certain organization divisions. For example, the function “check the application of business registration” can only be executed by the civil servants of the business registration office. But in 44 of the process models there was a wrong department modeled or the organization unit missed completely.

So, there is a lack of resource usage rules like: *If a process uses an activity X, the process must (must not) use the resource Y.*

- (E2) Companies often combine several application cases. For example, in 24% of the cases the companies combine both, the application of business registration and an application of business building permission. In these cases, two different organizational units are responsible, the business registration office and the building authority. But in 13% of the cases one of the responsible organization units was missed.

So, there is a lack of resource occurrence rules like: *If a process demands a resource X, then it must also contain/involve that resource X.*

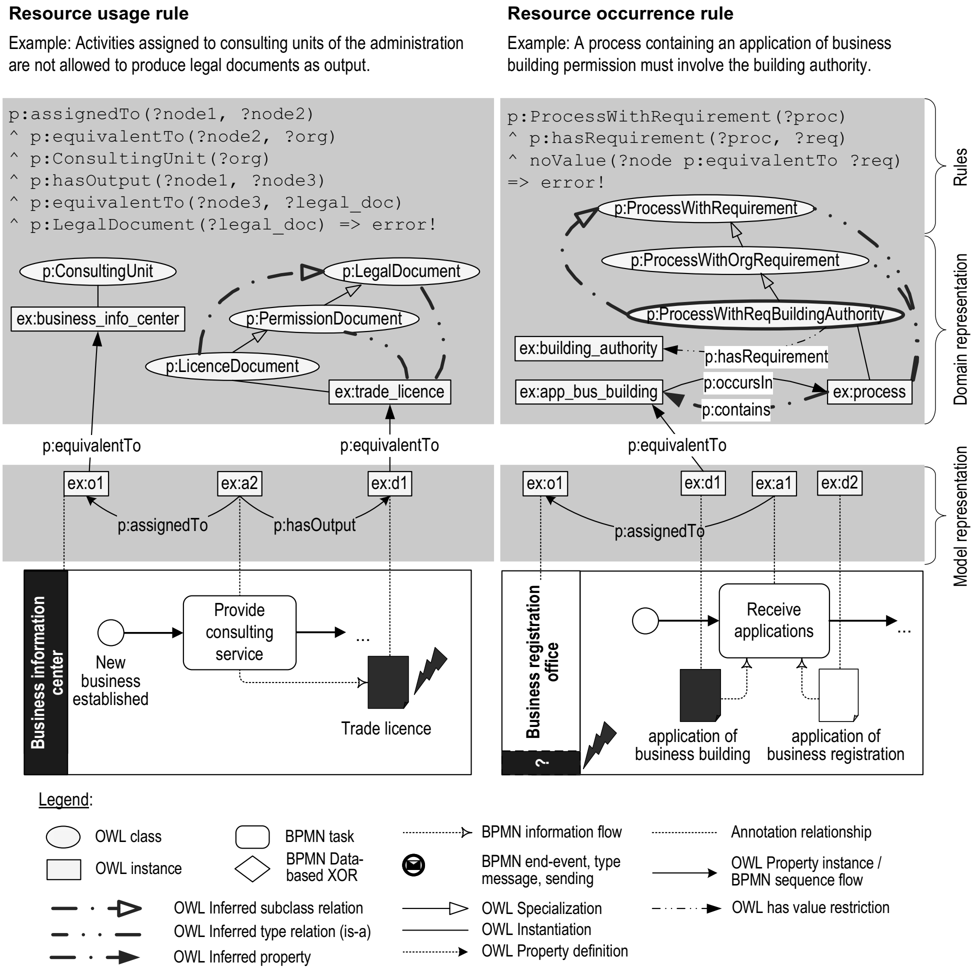
4 Ontology-driven approach for semantic verification

4.1 Classification of semantic verification rules

In our ontology-driven approach for semantic verification, we apply rules on top of ontologies to verify process models. In general, rules may be divided into deductive and normative rules based on Boley et al. [BKP07, p. 273]. Deductive rules are used to win new facts on the basis of existing facts through the use of logical implications. Normative rules are used to express conditions for the data used for an application or the logic used by it. This understanding implies that the ontology is either entirely true or contains incorrect facts. As we also want to express constraints which – when violated – result merely in warnings and thus leaving it up to human judgment to decide whether a model construct is correct or not, we do not call our rules “integrity rules”. Instead, we prefer the term “verification rules”, and as our rules are specified using concepts of a formal ontology, we call them “semantic verification rules”. The rule matter specifies the subject of a rule which is either the process, i.e. the set of nodes and arcs which constitute the core process graph, or the resources which are involved in the process. Due to space limitations, in the remainder of this paper, we focus exclusively on resource constraints, i.e. the latter aspect. Resource constraints can target at the structure of a process graph involving several resource-nodes connected by edges as well as at the occurrence of specific resource nodes anywhere in the process graph. According to this distinction, we differentiate between *resource usage* rules and *resource occurrence* rules. The result of the execution of a semantic verification rule may be a warning or an error.

4.2 Application to the case problems

In this section, we provide practical examples for the semantic verification rule types introduced in the previous section illustrating how our approach of semantic verification can be applied to the case problems given in section 3. As a prerequisite, the process model has to be represented in the ontology and annotated with ontology instances using the `p:equivalentTo`-property (see Fig. 1, due to space limitations, we only show some annotations). We use the prefix `p` for more general ontology contents and the prefix `ex` for contents related to concrete examples. On top of this ontology-based representation we apply our semantic verification rules. The ontology language OWL, used in our approach, only supports the formulation of rules via extensions. Such an extension is the Semantic Web Rule Language (SWRL) [HPB04] which extends OWL with IF-THEN-rules in the form of a logical implication. The rules presented in the examples are of this nature and can be formalized using SWRL. The rules have the general form of antecedent → consequent – i.e. if the antecedent (body) of the rule is true, then the consequent (head) must also be true. Since the consequent consists of error messages, it will not be true in a literal sense, it rather will be generated if the antecedent matches and the rule is fired. In the following, we elaborate on some of the abstractions and inferences possible by using terminological and domain knowledge. They are an important merit of our approach as they provide for the formulation of rather generic semantic verification rules applicable to concrete models by automated machine reasoning.



Resource usage rule

Example: Activities assigned to consulting units of the administration are not allowed to produce legal documents as output.

```
p:assignedTo(?node1, ?node2)
^ p:equivalentTo(?node2, ?org)
^ p:ConsultingUnit(?org)
^ p:hasOutput(?node1, ?node3)
^ p:equivalentTo(?node3, ?legal_doc)
^ p:LegalDocument(?legal_doc) => error!
```

Resource occurrence rule

Example: A process containing an application of business building permission must involve the building authority.

```
p:ProcessWithRequirement(?proc)
^ p:hasRequirement(?proc, ?req)
^ noValue(?node p:equivalentTo ?req)
=> error!
```

Fig. 1: Resource usage and resource occurrence rule

Resource usage rule: The rule in the given example (Fig. 1, left) fires, if any activity node assigned to an organizational node being an individual of `p:ConsultingUnit` produces a legal document as output. The example makes use of subsumption reasoning, so it can be inferred that `ex:trade_licence` of type `p:LicenceDocument` is-a `p:LegalDocument`. More formally, these inferences can be described using the DL-Syntax to show the fragments of a knowledge base which are relevant for the conclusions. The knowledge base is partitioned into a terminological part (TBox) and an assertional part (ABox) and inferred facts which may relate to both the TBox and ABox. For subsumption, the symbol $\sqsubseteq$ is used, type relations are indicated with $\in$ followed by the name of a TBox-element such as a class-name or property-name, properties in the ABox are written using brackets $\langle\ \rangle$, existential quantification is written as $\exists$, inverse properties are suffixed with an elevated dash $^-$ and individuals are enumerated using curly brackets {}.

TBox	p:PermissionDocument ⊑ p:LegalDocument p:LicenceDocument ⊑ p:PermissionDocument
ABox	ex:trade_licence ∈ p:LicenceDocument
Inferred T/A-Box	p:LicenceDocument ⊑ p:LegalDocument ex:trade_licence ∈ p:PermissionDocument ex:trade_licence ∈ p:LegalDocument

Resource occurrence rule: The rule in the given example (Fig. 1, right) makes use of a property `p:contains` being the inverse of `p:occursIn`, so that it can be concluded that `ex:process` contains the application of business building `ex:app_bus_building`. Based on this, it can be inferred that this process belongs to the class `p:ProcessWithReqBuildingAuthority` which is defined precisely as all processes containing an `ex:app_bus_building`. As `p:ProcessWithReqBuildingAuthority` is subsumed by `p:ProcessWithRequirement`, the semantic verification rule can operate on this abstract level using the latter class. Requirements are specified on the respective subclasses of `p:ProcessWithRequirement` using the hasValue-restriction of OWL which allows specifying a value of the requirement, i.e. an instance that must be present in the process that should be annotated to at least one of the process nodes. More formally, these inferences can be described using the DL-Syntax.

TBox	p:ProcessWithReqBuildingAuthority ⊑ p:ProcessWithOrgRequirement ⊓ ∃p:hasRequirement.{ex:building_authority} p:ProcessWithOrgRequirement ⊑ p:ProcessWithRequirement p:occursIn ≡ p:contains⁻
ABox	p:process ∈ p:ProcessWithReqBuildingAuthority ⟨ex:app_bus_building, ex:process⟩ ∈ p:occursIn
Inferred T/A-Box	p:ProcessWithReqBuildingAuthority ⊑ p:ProcessWithRequirement ex:process ∈ p:ProcessWithOrgRequirement ex:process ∈ p:ProcessWithRequirement ⟨ex:app_bus_building, ex:process⟩ ∈ p:contains

Based on these inferences, the rule then checks if there is not a single node in the process graph being annotated with that instance by using the noValue-extension of the Jena rule engine (jena.sourceforge.net).

5 Limitations of Semantic Verification

In general, SWRL and OWL work according to the so called "open world assumption" which is based on the assumption that facts not present in the knowledge base are unknown or undefined. Therefore, only rules conforming to the scheme x ^ y → error are possible. By using the Jena rule engine, we can extend the range of possible rules to include also rules of the form x ^ ¬y → error. That is, if some facts x are known and some other facts y are not present in the knowledge base, the failure to derive them is treated as a form of negation (Negation as Failure, NAF). With NAF it is possible to specify rules which fire, if something is missing in the process model. Clearly, semantic verification rules have some limitations. To begin with, they should not be regarded as a surrogate for verifications related to the meta-model or the grammar of the used modeling language. They are rather complementary to such verifications and correct models form the basis for additional semantic verification checks. Also, aspects regarding the execution semantics of models such as soundness etc. dealing mainly with the absence of deadlocks, livelocks and other anomalies are not covered by our approach.

Further limitations of semantic verification rules are that they depend on the availability of ontology and the annotation of process models. While in other areas such as the life sciences huge ontologies have been developed and standardized, the field of administration still lacks authorities who develop and standardize ontologies. However, this problem may partly disappear, if the terminology problem will be solved e.g. by defining structured vocabularies which can well serve as a skeletal structure for ontologies. Also, current tools for process model annotation are mostly in the state of research prototypes. In particular, functionalities for semi-automated annotations and annotation suggestions based e.g. on annotations previously made in the current model or the whole model repository have to be developed in the future in order to enable comfortable and cost-effective semantic verification. A major limitation of the current approach is that it is agnostic to the control flow of process models. At the moment, the only exception of that is the property `p:followedBy` connecting only nodes which form a sequence when the model will be executed and so it provides for rules such as "b should not be executed after a".

5 Conclusion and further Research

The approach presented in this paper showed how to use ontologies, rules and reasoning for the semantic verification of process models. The role of ontologies is to describe elements of semi-formal process models with machine processable semantics. In conjunction with an inference engine, the model elements can be classified thereby using facts automatically derived from the ontology. Based on these classifications of model elements, abstract semantic verification rules are used to decide whether the model conforms to rules and regulations.

Future versions of our approach will tackle some of the described limitations. As a next step, we plan to integrate a further pre-processing step which will mark the nodes in the graph according to their succession of logical connectors such as AND, XOR and OR. The capturing of information on such local contexts of parallelism or exclusivities to the ontology based representation of process models will allow advanced semantic verification rules such as “resource x must not be used in parallel branches” or “activity x should always be executed exclusively with activity z”.

References

[ADW08] Awad, A., Decker, G., Weske, M. (2008): Efficient Compliance Checking Using BPMN-Q and Temporal Logic. In: Proceedings of the 6th International Conference on Business Process Management, Dumas, M., Reichert, M., Shan, M.-C. (eds.), Berlin et al., pp. 326-341.

[AM09] Abouzaid, F., Mullins, J. (2009): Model-checking Web Services Orchestrations using BP-calculus, Electronic Notes in Theoretical Computer Science (255), pp. 3-21.

[ARG06] Aguilar, E.R., Ruiz, F., García, F., Piattini, M. (2006): Evaluation measures for business process models. In: Proceedings of the 21st annual ACM Symposium on Applied Computing, ACM New York, pp. 1567-1568.

[Ba08] Barjis, J. (2008): The importance of business process modeling in software systems design, Science of Computer Programming (71:1), pp. 73-87.

[BDH09] Becker, J., Delfmann, P., Herwig, S., Lis, L. Stein, A. (2009): Towards Increased Comparability of Conceptual Models – Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In: Arbeitsberichte der Universität Münster, Münster.

[BKP07] Boley, H., Kifer, M., Patranjan, P. L., Polleres, A. (2007): Rule Interchange on the Web. In: Antoniou, G., Aßmann, U., Baroglio, C., Decker, S., Henze, N., Patranjan, P.-L., Tolksdorf, R. (2007) (eds.): Reasoning Web: Third International Summer School 2007, Dresden, Germany, September 3-7, 2007. Berlin, Springer, pp. 269-309.

[CB08] Chapurlat, V., Braesch, C. (2008): Verification, validation, qualification and certification of enterprise models: Statements and opportunities, Computers in Industry (59:7), p. 711–721.

[CCO00] Cobleigh, J.M., Clark, L.A., Osterweil, L.J. (2000): Verifying properties of process definitions, ACM SIGSOFT Software Engineering Notes (25:5), pp. 96-101.

[CKS10] Chanda, J., Kanjilal, A., Sengupta, S. (2010): UML-Compiler: A Framework for Syntactic and Semantic Verification of UML Diagrams. In: Distributed Computing and Internet Technology, Berlin et al., pp. 194-205.

[Cl08] Clarke, E.M. (2008): The birth of model checking. In: Proceedings of the 15th International Conference Tools and Algorithms for the Construction and Analysis of Systems, York, pp. 1-26.

[DAC04] Dong, J., Alencar, P.S.C., Cowan, D.D. (2004): A behavioral analysis and verification approach to pattern-based design composition, Software and Systems Modeling (3:4), pp. 262-272.

[DDO07] Dijkman, R.M., Dumas, M., Ouyang, C. (2007): Formal semantics and automated analysis of BPMN process models, Preprint Technical Report 5969, Available from http://eprints.qut.edu.au/archive/00005969/.

[DDO08] Dijkman, R.M., Dumas, M., Ouyang, C. (2008): Semantics and analysis of business process models in BPMN. Information and Software Technology (50:12), pp. 1281-1294.

[De02] Desel, J. (2002): Model Validation - A Theoretical Issue? In: Esparza, J., Lakos, C. (eds.): Application and Theory of Petri Nets 2002 : 23rd International Conference, ICATPN 2002, Adelaide, Australia, June 24–30, Proceedings, Berlin, Springer, p. 23-43.

[DHP09] Deutsch, A., Hull, R., Patrizi, F., Vianu, V. (2009): Automatic verification of data-centric business processes. In: Proceedings of the 12th International Conference on Database Theory, St. Petersburg, Russia, pp. 252-267.

[DvA04] Dehnert, J., van der Aalst, W.M.P. (2004): Bridging the gap between business models and workflow specifications, International Journal of Cooperative Information Systems (13:3), pp. 289-332.

[DFH07] Drumm, C., Filipowska, A., Hoffmann, J., Kaczmarek, M., Kaczmarek, T., Kowalkiewicz, M., Markovic, I., Scicluna, J., Vanhatalo, J., Völzer, H., Weber, I., Wieloch, K., Zyskowski, D. (2007): Dynamic Composition Reasoning Framework and Prototype. In: Project IST 026850 SUPER, Deliverable 3.2, SAP.

[EHK01] Engels, G., Heckel, R., Küster, J. (2001): Rule-based specification of behavioral consistency based on the UML meta-model. In: UML 2001 – The Unified Modeling Language. Modeling Languages, Concepts, and Tools, Berlin et al., pp. 272-286.

[ES08] El Kharbili, M., Stein, S. (2008): Policy-Based Semantic Compliance Checking for Business Process Management. In: Modellierung betrieblicher Informationssysteme (MobIS 2008), Loos, P. et al. (eds.), CEUR Workshop Proceedings 420, Aachen, pp. 165-177.

[ESM08] El Kharbili, M., Stein, S., Markovic, I., Pulvermüller, E. (2008): Towards a Framework for Semantic Business Process Compliance Management. In: Proceedings of the 1st International Workshop on Governance, Risk and Compliance (GRCIS2008), Montpellier, France.

[Fr09] Friedrich, F. (2009): Measuring Semantic Label Quality Using WordNet. In: 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“, Nüttgens, M. et al. (eds.), Berlin, pp. 7-21.

[FUM03] Foster, H., Uchitel, S., Magee, J., Kramer, J. (2003): Model-based verification of web service compositions. In: Proceedings of the 18th IEEE International Conference on Automated Software Engineering (ASE 2003), October 6-10, Montreal, Canada, IEEE Computer Society, pp. 152-161.

[GL07] Gruhn, V., Laue, R. (2007): What business process modelers can learn from programmers, Science of Computer Programming (65:1), pp. 4-13.

[GL07i] Gruhn, V., Laue, R. (2007): Checking Properties of Business Process Models with Logic Programming. In: Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS 2007), Madeira, Portugal, pp. 84-93.

[GL07ii] Gruhn, V., Laue, R. (2007): Forderungen an hierarchische EPK-Schemata. In: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, 6. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“, Bonn, 2007.

[Gr91] Gruhn, V. (1991): Validation and verification of software process models. In: Proceedings of the European Symposium on Software Development Environments and CASE Technology, Berlin.

[GS09] Gerede, C., Su, J. (2009): Specification and verification of artifact behaviors in business process models. In: Service-Oriented Computing - ICSOC 2007, Berlin et al., pp. 181-192.

[Go08] Governatori, G., Hoffmann, J., Sadiq, S. W., Weber, I. (2008): Detecting Regulatory Compliance for Business Process Models through Semantic Annotations. In: Business Process Management Workshops. Berlin, Springer (LNBI 17), pp. 5-17

[GV06] Goedertier, S., Vanthienen, J. (2006): Designing compliant business processes with obligations and permissions. In: Business Process Management Workshops, Eder, J., Dustdar, S. (eds.), Berlin et al., pp. 5-14.

[GXS04] Geguang, P., Xiangpeng, Z., Shuling, W., Zongyan, Q. (2004): Towards the Semantics and Verification of BPEL4WS. In: Proceedings of the International Workshop on Web Languages and Formal Methods (WLFM2005), Newcastle, UK.

[HKT02] Heckel, R., Küster, J., and Taentzer, G. (2002): Towards automatic translation of UML models into semantic domains. In: Proceedings ICGT 2002 - First International Conference on Graph Transformation (ICGT2002), Barcelona, Spain, pp. 11-22.

[KG07] Koliadis, G., Ghose, A. (2007): Verifying semantic business process models in inter-operation. In: Proceedings of the International Conference on Services Computing (SCC 2007), Salt Lake City, USA, pp. 731-738.

[KG08] Kovacs, M., Gönczy, L. (2008): Simulation and formal analysis of workflow models. In: Electronic Notes in Theoretical Computer Science : Proceedings of the Fifth International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2006), 211, pp. 221-230.

[Ki06] Kindler, E. (2006): On the semantics of EPCs: Resolving the vicious circle, Data & Knowledge Engineering (56:1), pp. 23-40.

[KTK02] Koehler, J., Tirenni, G., Kumaran, S. (2002): From business process model to consistent implementation: A case for formal verification methods. In: 6th International Enterprise Distributed Object Computing Conference (EDOC 2002), Lausanne, Switzerland, pp. 96-107.

[LBL06] Lu, S., Bernstein, A., Lewis, P. (2006): Automatic workflow verification and generation, Theoretical Computer Science (353:1-3), pp. 71-92.

[LDH08] La Rosa, M., Dumas, M., ter Hofstede, A., Mendling, J., Gottschalk, F. (2008): Beyond control-flow: Extending business process configuration to roles and objects. In: Proceedings of the Conceptual Modeling 2008 (ER2008), pp. 199-215.

[LMX07] Liu, Y., Müller, S., Xu, K. (2007): A static compliance-checking framework for business process models. In: IBM Systems Journal (46:2), pp. 335–361.

[LRD08] Ly, L.T., Rinderle, S., Dadam, P. (2008): Integration and verification of semantic constraints in adaptive process management systems, Data & Knowledge Engineering (64:1), pp. 3-23.

[LRD10] Ly, L. T., Rinderle-Ma, S., Dadam, P. (2010): Design and Verification of Instantiable Compliance Rule Graphs in Process-Aware Information Systems. In: Proc. of CAiSE’10, June 9-11, Hammamet, Tunisia, Berlin, Springer (LNCS 6051)

[LSM09] Leopold, H., Smirnov, S., Mendling, J. (2009): On Labeling Quality in Business Process Models. In: 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“, Nüttgens, M. et al. (eds.), Berlin, pp. 42-57.

[Ma03] Martens, A. (2004): On Compatibility of Web Services. Petri Net Newsletter 65 (2003), pp. 12–20

[Me09] Mendling, J. (2009): Empirical Studies in Process Model Verification. In: Jensen, K., van der Aalst, W. M. P. (eds.): Transactions on Petri Nets and Other Models of Concurrency II, Berlin, Springer (LNCS 5460), pp. 208–224.

[MNA07] Mendling, J., Neumann, G., van der Aalst, W. (2007): On the correlation between process model metrics and errors. In: Proceedings of the 26th International Conference on Conceptual Modeling (ER2007), Auckland, New Zealand, Australia, pp. 173-178.

[MRR10] Mendling, J., Reijers, H.A., Recker, J. (2010): Activity labeling in process modeling: Empirical insights and recommendations. In: formation Systems, (35:4), pp. 467-482.

[MvA08] Mendling, J., van der Aalst, W. (2008): Advanced Reduction Rules for the Verification of EPC Business Process Models. In: Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems, Marburg, Germany, pp. 129-140.

[Na02] Nakajima, S. (2002): Model-checking verification for reliable web service. In: Proceedings of OOWS'02, satellite event of OOPSLA'02, Seattle, USA.

[NM07] Nelson, H.J., Monarchi, D.E. (2007): Ensuring the quality of conceptual representations, Software Quality Journal (15:2), pp. 213-233.

[NS07] Namiri, K., Stojanovic, N. (2007): Pattern-based design and validation of business process compliance. In: On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS – OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Proceedings, pp. 59-76.

[Ok09] Okika, J.C. (2009): Analyzing Orchestration of BPEL Specified Services with Model Checking. In: Proceedings of the PhD Symposium of the 7th International Joint Conference on Service Oriented Computing, Stockholm, Sweden.

[PG08] Pons, C., Garcia, D. (2008): A lightweight approach for the semantic validation of model refinements, Electronic Notes in Theoretical Computer Science (220:1), pp. 43-61.

[PW06] Puhlmann, F., Weske, M. (2006): Investigations on Soundness Regarding Lazy Activities. In: Proceedings of the 4th International Conference on Business Process Management (BPM 2006), Berlin, Springer (LNCS 4102), pp. 145-160

[PW09] Peters, N., Weidlich, M. (2009): Using Glossaries to Enhance the Label Quality in Business Process Models. In: 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“, Nüttgens, M. et al. (eds.), Berlin, 2009, pp. 75-90.

[SCA07] Salomie, I., Cioara, T., Anghel, I., Dinsoreanu, M., Salomie, T.I. (2007): A Layered Workflow Model Enhanced with Process Algebra Verification for Industrial Processes. In: Proceedings of the 2007 IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pp. 185-191.

[SM06] Simon, C., Mendling, J. (2006): Verification of Forbidden Behavior in EPCs. In: Mayr, H. C., Breu, R. (eds.): Proceedings of the GI Conference Modellierung (MOD2006), March, 22-24, Innsbruck, Austria, pp. 233–242.

[SO00] Sadiq, W., Orlowska, M.E. (2000): Analyzing process models using graph reduction techniques, Information Systems (25:2), pp. 117-134.

[SOS05] Sadiq, S.W., Orlowska, M.E., Sadiq, W. (2005): Specification and validation of process constraints for flexible workflows. Information Systems (30:5), pp. 349-378.

[SPH04] Speck, A., Pulvermueller, E., Heuzeroth, D. (2004): Validation of business process models. In: Proceedings of the 17th European Conference on Object-oriented Programming (ECOOP), July 21-25 2003, Darmstadt, Berlin, Springer (LNCS 3013).

[SW04] Soffer, P., Wand, Y. (2004): Goal-driven analysis of process model validity. In: Advanced Information Systems Engineering, Berlin et al., pp. 229-319.

[TBB08] Touré, F., Baïna, K., Benali, K. (2008): An Efficient Algorithm for Workflow Graph Structural Verification. In: Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Berlin, pp. 392-408.

[TF09] Thomas, O., Fellmann, M. (2009): Semantic Process Modeling – Design and Implementation of an Ontology-Based Representation of Business Processes. In: Business & Information Systems Engineering (1:6), pp. 438–451.

[vA97] van der Aalst, W.M.P. (1997): Verification of workflow nets. In: Proceedings of the 18th International Conference on Application and Theory of Petri Nets, Berlin, pp. 407-426.

[vA98] van der Aalst, W.M.P. (1998): The application of Petri nets to workflow management, Journal of Circuits Systems and Computers (8), pp. 21-66.

[vA99] van der Aalst, W.M.P. (1999): Formalization and verification of event-driven process chains, Information and Software Technology (41:10), pp. 639-650.

[vA03] van der Aalst, W.M.P. (2003): Inheritance of business processes: A journey visiting four notorious problems. In: Petri Net Technology for Communication-Based Systems, Berlin et al., pp. 383-408.

[vAB05] van der Aalst, W., de Beer, H.T., van Dongen, B.F. (2005): Process Mining and Verification of Properties: An Approach Based on Temporal Logic. In: On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Berlin, pp. 130-147.

[vBK06] Van Breugel, F., Koshkina, M. (2006): Models and Verification of BPEL. In: Technical Report, York University, Toronto, Canada. Available from http://www.cse.yorku.ca/franck/research/drafts/tutorial.pdf.

[Va03] Varró, D. (2003): Towards symbolic analysis of visual modeling languages, Electronic Notes in Theoretical Computer Science (72:3), pp. 51-64.

[vDA05] van Dongen, B.F., van der Aalst, W.M.P., Verbeek, H.M.W. (2005): Verification of EPCs: Using reduction rules and Petri nets. In: Advanced Information Systems Engineering, Berlin et al., pp. 372-386.

[vDJ07] van Dongen, B.F., Jansen-Vullers, M.H., Verbeek, H.M.W., van der Aalst, W.M.P. (2007): Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants, Computers in Industry (58:6), pp. 578-601.

[VP02] Varró, D., Pataricza, A. (2002): Metamodeling mathematics: A precise and visual framework for describing semantics domains of UML models. In: «UML» 2002 - The Unified Modeling Language, Berlin et al., pp. 449-456.

[We09] Weber, I.M. (2009): Verification of Annotated Process Models. In: Semantic Methods for Execution-level Business Process Modeling, Berlin et al., pp. 97-148.

[WHM10] Weber, I., Hoffmann, J., Mendling, J. (2010): Beyond soundness: on the verification of semantic business process models, Distributed and Parallel Databases (2010:27), pp. 271-343.

[WWM09] Wolter, C., Miseldine, P., Meinel, C. (2009): Verification of Business Process Entailment Constraints Using SPIN. In: Proceedings of the 1st International Symposium on Engineering Secure Software and Systems, Berlin et al., pp. 1-15.

[XBS02] Xiang, F., Bultan, T., and Su, J. (2002): Formal Verification of e-Services and Workflows. In: Proceedings of Web Services, E-Business, and the Semantic Web: CAiSE 2002 International Workshop, WES 2002, Toronto, Canada, pp. 180-202.

[XWW07] Xinhua, L., Weida, W., Wenjian, L. (2007): An Intelligent Methodology for Business Process Model Verification. In: Proceedings of the IEEE International Conference on Control and Automation (ICCA2007), Frankfurt, pp. 2381-2385.