

# A Modelling Environment for Business Process as a Service

Knut Hinkelmann<sup>1,2</sup>(✉), Kyriakos Kritikos<sup>3</sup>, Sabrina Kurjakovic<sup>1</sup>, Benjamin Lammel<sup>1</sup>, and Robert Woitsch<sup>4</sup>

<sup>1</sup> FHNW University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland

{knut.hinkelmann, sabrina.kurjakovic, benjamin.lammel}@fhnw.ch

<sup>2</sup> Department of Informatics, University of Pretoria, Pretoria, South Africa

<sup>3</sup> FORTH Institute of Computer Science, Heraklion, Crete, Greece

kritikos@ics.forth.gr

<sup>4</sup> BOC Asset Management, Vienna, Austria

robert.woitsch@boc-eu.com

**Abstract.** Business processes can benefit from cloud offerings, but bridging the gap between business requirements and technical solutions is still a big challenge. We propose Business Process as a Service (BPaaS) as a main concept for the alignment of business process with IT in the cloud. The mechanisms described in this paper provide modelling facilities for both business and IT levels: (a) a graphical modelling environment for processes, workflows and service requirements, (b) an extension of an enterprise ontology with cloud-specific concepts, (c) semantic lifting of graphical models and (d) SPARQL querying and inferencing for semantic alignment of business and cloud IT.

## 1 Introduction

Currently cloud components are offered in a way that is well understood by IT-specialists. Many small and medium enterprises (SMEs) are currently excluded from using the Cloud due to high entry barriers, related to missing technical expertise to evaluate cloud services and to prepare the enterprise for the cloud usage. There is a big gap between pragmatic, legally influenced business processes and a huge cloud market with numerous offerings that rarely consider business situations but focus on technical details.

The EU-funded project CloudSocket aims to support the wide usage of cloud computing to SMEs such that they can easily benefit from cost reduction, as well as, from the dynamic and adaptive IT infrastructure in order to reduce their administrative burden and enable agility as well as new business opportunities.

This is achieved by the concept of Business Process as a Service (BPaaS), which maps to the ability to autonomously run whole business processes in the cloud. It is the objective of our approach that business users do not have to care for technical details, but specify their requirements in a business language. Then, through the environment proposed, support for the alignment between the business and IT level can be achieved

mapping business-oriented models to technical ones that can drive the allocation and execution of business processes in the cloud.

The BPaaS Design Environment provides conceptual modelling tools for (a) designing domain specific business processes, (b) executable workflows, (c) additional description and rules for deployment as well as (d) key performance indicators. It not only allows the business user to model the requirements in a business language but also supports the smart alignment of business and IT in the cloud. This involves the identification of executable workflows for specific business process models through applying service discovery and composition techniques. This requires that the information about the business process, the service requirements and the workflows are represented in machine-interpretable format [16].

The BPaaS Design Environment (see Fig. 1) supports both *machine interpretation* and *human interpretation* of the enterprise model. The *human-interpretable, graphical* modelling is supported by the model stack as presented in Sect. 5. The machine-interpretation is supported by the BPaaS Ontology and used for alignment as described in Sects. 6 and 7. The integration of these two interpretation types is achieved by the semantical lifting of the graphical models (see Sect. 8).

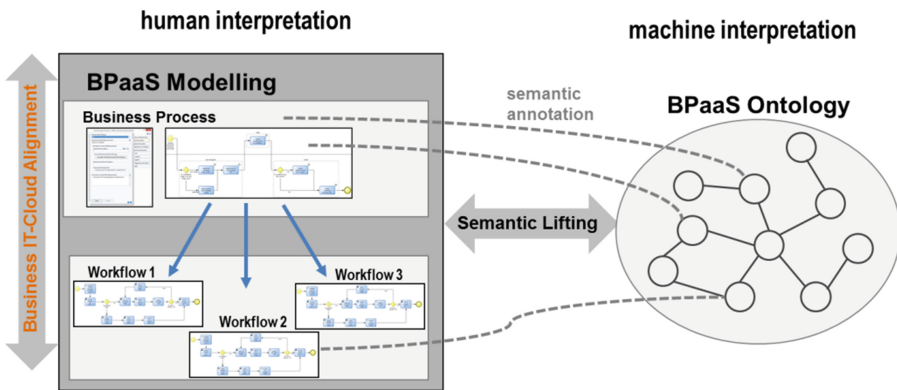


Fig. 1. General structure of the BPaaS Design Environment

## 2 Literature Review

Modelling the business processes, workflows and services in CloudSocket is part of enterprise modelling - the description and definition of the processes, structure, information and resources of an enterprise. According to Fox and Gruninger [11] an enterprise model must supply the information and knowledge necessary to support the operations of the enterprise. Enterprise modelling techniques are developed in several fields such as business process modelling, information modelling, systems modelling, and enterprise architecture.

Enterprise architecture (EA) models describe all relevant business structures, IT structures, and their relationships. The Zachman Framework is a two dimensional

matrix, in which the cells contain models [32]. Another well-known EA framework is TOGAF [26]. The overall enterprise architecture comprises a set of closely inter-related architectures: Business Architecture, Information Systems Architecture, and Technology Architecture. The ArchiMate Standard [28] introduces an integrated language for describing enterprise architectures.

OMG has developed several specialized modelling languages for enterprise modelling, for example Business Process Model and Notation (BPMN) [24], Case Management Model and Notation (CMMN) [25], the Decision Model and Notation (DMN) [26] and the Business Motivation Model (BMM) [21]. The primary purpose of these graphical modelling languages is to support communication between human stakeholders, although there do exist execution engines for BPMN and decision tables.

The purpose of ontologies in enterprise modelling is to formalize and establish the shareability, re-usability, assimilation and dissemination of information across all organizations and departments within an enterprise. Describing enterprise architecture as an ontology started in the 1990s with TOVE [9], The Edinburgh Enterprise Ontology [30] and the organizational memory [1]. More recent work is the Context Based Enterprise Ontology [19]. Den Haan [8] has used an enterprise ontology to realize a Model-Driven Enterprise Engineering.

In the context of enterprise ontologies, the semantic business process management approach aims to achieve a new dimension of business IT alignment. Adding semantics to business processes enables machine reasoning and allows exploiting the full potential of process automation [15].

Conventional cloud services offerings include software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). These offerings impose vendor lock-in and challenge the “developers to mix and match freely from diverse cloud services tiers” [22]. The concept of business process as a service (BPaaS) provides the flexibility of mixing different delivery models and focusing on the end-to-end business processes instead of single applications [22]. Flexibility is also achieved by the atomized and dynamic configuration possibilities supported through the monitoring of threshold values on business and technical metrics. New resources can be added or removed to/from the BPaaS according to the individual needs.

BPaaS represents an initial field of research. Most of the research work proposed focuses on how to define BPaaS and the respective candidate architectures to realise it [2]. Some work has concentrated on dealing with security aspects (e.g., anonymisation-based protocols for BPaaS fragments [3]). Finally, initial work has been conducted on how elasticity can be realised for BPaaS through a specific formal model and a respective elasticity framework [20].

### 3 Overview of the BPaaS Design Environment

The BPaaS Design Environment comprises two modelling components - the BPaaS Modelling Environment and the BPaaS Ontology, including the inference engine for the smart alignment (see Fig. 2). The BPaaS Modelling Environment encompasses the meta-model for the human-interpretable, graphical modelling languages, i.e. Business Process Model and Notation (BPMN) [24].

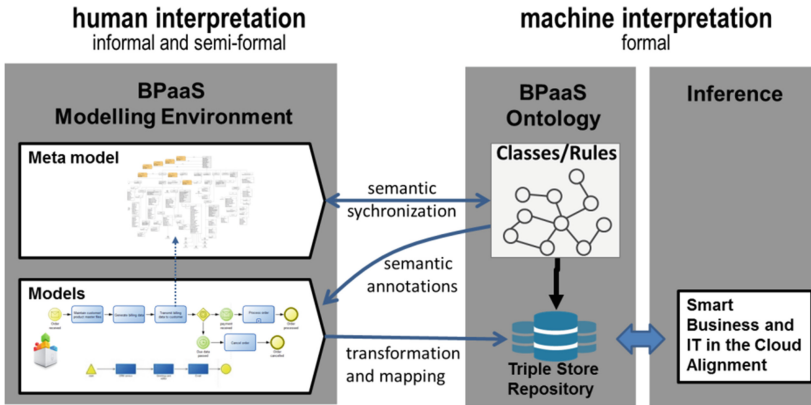


Fig. 2. Elements of the BPaaS Design Environment

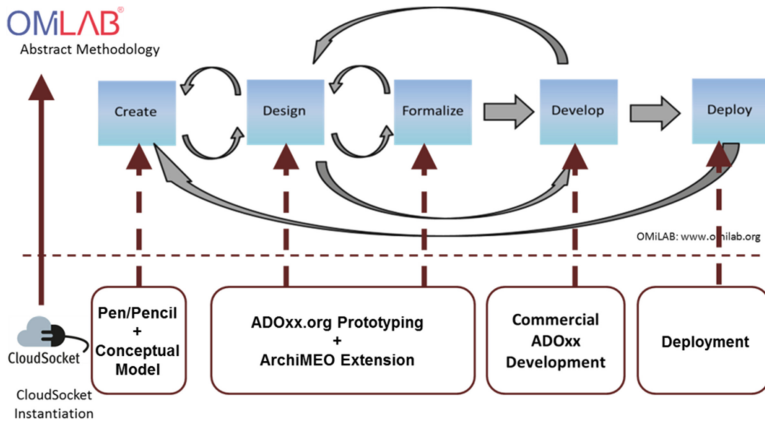
The graphical models can then be semantically annotated with the ontological concepts, which are defined in the BPaaS Ontology. This means, that both ontology and meta-model development have to be synchronized in the sense that the ontology contains class definitions describing the intended semantics of the elements of the graphical modelling language. For the design environment user, this approach provides the possibility of modelling the business process and annotating the elements modelled with corresponding functional and non-functional service specifications, such as business, technical and compliance requirements.

## 4 Methodology

The development of the BPaaS Design Environment was supported by the OMiLAB LifeCycle, which is the basis of Agile Model Method Engineering [17] and has been developed and successfully used in the Open Models Initiative (<http://www.openmodels.at>).

Figure 3 depicts in the upper part the abstract developing methodology proposed by OMiLAB and in the lower part the concrete instantiation in the context of the BPaaS Design Environment. The research presented in this paper focuses on the first three methodology phases.

- *Create phase:* In this phase the domain and scope of the modelling framework were determined and the class hierarchy was defined. This phase is comparable to steps 1 to 5 of the approach for ontology development [21]. We analysed real situations of small and medium enterprises and identified a list of so called competency questions, which served as a basis to determine the scope of the ontology [29]
- *Design and formalize phase:* Those two phases are combined using a rapid prototyping approach. In ADOxx.org rapid prototypes of the BPaaS Modeling Environment were implemented. In parallel, a first prototype of the BPaaS Ontology was realized as an extension of the already existing ArchiMEO ontology.



**Fig. 3.** Adapted OMiLAB methodology

For the development of the BPaaS Design Environment we analysed several real-world business scenarios. This was done in workshops with use case partners of the CloudSocket project. The business scenarios served as a starting point, since they represent real situations as they occur in enterprises. We implemented a cloud realization of a simple scenario - the sending of Christmas cards. This process is based on three main services (a) card designer, (b) customer relationship management, and (c) email service. Although this process seems to be simple, the underlying complexity increases by configuring the process such that particular requirements are met (e.g., industry compliance, data privacy, scheduling).

To determine the scope of the modelling framework we sketched a list of questions that the system should be able to answer. These questions are called competency questions. They have been introduced by Gruninger and Fox [13] as a method for enterprise engineering and ontology scope determination [29]. This approach is widely known and was amongst others adopted by De Leenheer and Mens [7], De Bruijn [6] and Cardoso [4].

In order to develop the competency questions we analysed the individual components of the CloudSocket ecosystem. Who are the involved actors? What kind of value objects are exchanged? What are the value activities? Who are the composite actors? Four areas of competency questions have been identified that can be raised:

- **General alignment:** Questions regarding the mapping of business processes to workflows, e.g., which workflows are available for a given business process.
- **Business perspective:** Questions with respect to payment, contract, monitoring and support of the BPaaS and questions prospective customers might ask in order to assess the trustworthiness of the cloud service provider
- **Security/legal perspective:** Questions that are important with respect to security/risks/functionality or for SMEs in general or for SMEs operating in highly regulated industries.
- **Technical perspective:** Questions with respect to data formats, platforms or implementation.

## 5 BPaaS Modelling Method

Models are representing part of reality or a vision in an agreed modelling language. The BPaaS meta model defines the (a) domain specific business layer and (b) the IT-Cloud relevant technical layer, as well as the interaction between them (see Fig. 4).

- The business process layer includes business process, organisation and design models.
  - A *process map* model gives an overview of the organisation’s processes. The modelling language of the BPaaS *business processes* is a subset of BPMN 2.0 that can be linked to: (a) service description (b) decision, and to (c) key performance indicator models. This subset is selected based on the authors’ practical experience and the analysis of the use cases.
  - An *organization model* can be built to illustrate a detailed structure of a working environment for the business process.
  - *Document models* represent documents (templates), which are utilized in the processes as input and output to activities.
- The IT layer consists of *workflow models*. Workflows are described with BPMN 2.0 [24] extended by execution-specific technical details and IT-related KPIs.
- The interaction between the domain-specific business layer and the IT layer is done by semantically lifting business process and workflow models with the *Service Description Model*. Within this model type process tasks can be semantically enriched by describing the requirements derived from the business process for Cloud Services. We categorized service requirements in (a) functional-, (b) input- (c) output- (d) non-functional- (e) business-, and (f) regulatory dimensions.

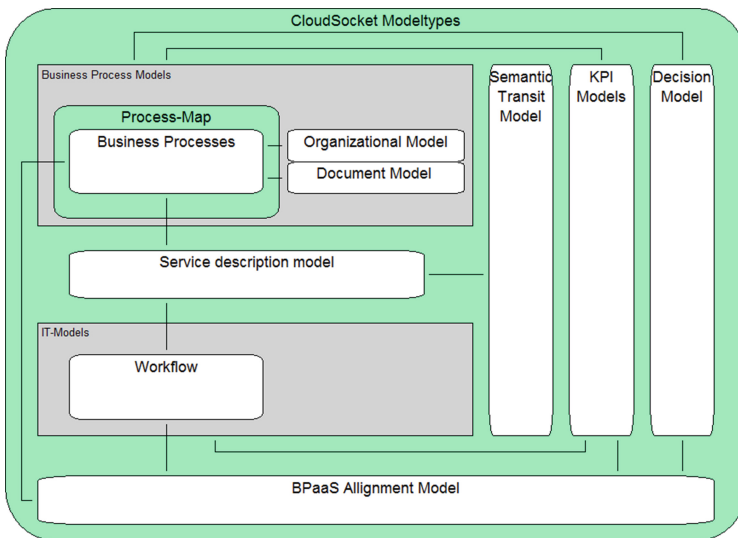


Fig. 4. The BPaaS model stack

- For both business and IT layers, there are links to the KPI and the decision models. The KPI cause effect model allows to model operational and strategic goals of cloud realizations. Such goals can be quantified by performance indicators. The aim of the decision model type, which corresponds to DMN, is to enable business users (e.g. analysts, technical developers) to comprehend the decisions and relate them to the data that might be held in the cloud.
- The Semantic Transit Model is used for semantic lifting of graphical models (see Sect. 8)

## 6 BPaaS Ontology

The BPaaS Ontology is implemented as an extension of the ArchiMEO enterprise ontology (<http://ikm-group.ch/archimeo>) with cloud-specific concepts, which are needed for smart alignment of the business and IT levels in the cloud. The cloud-specific extensions were determined from the analysis of the business scenarios and competency questions as described in Sect. 3. To enable a suitable and correct semantic lifting process, it is taken care that the BPaaS Ontology is consistent with the modelling method as described in Sect. 4.

ArchiMEO includes a top-level ontology, which contains general concepts, e.g. for location or time. Additionally it contains an Enterprise Upper Ontology with the concepts of the ArchiMate modelling language [28] as well as classes which represent the modelling elements of standard modelling languages like BPMN 2.0. The elements of these modelling languages are related to in the concepts coming from ArchiMate. For example, a BPMN activity is represented as a subclass of a Business Activity, which itself is a subclass of a Behaviour element in ArchiMate.

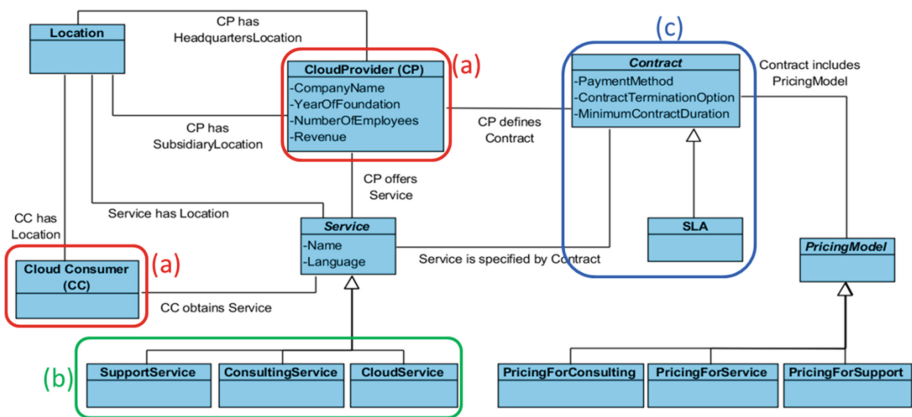


Fig. 5. Conceptual model of the business perspective

The BPaaS Ontology extends the ArchiMEO ontology according to the BPaaS requirements. Figure 5 depicts the class diagram of the overall conceptual model for the business perspective. The classes are integrated in the class hierarchy of ArchiMEO.

The class diagram highlights in red and labelled with (a) the classes of Cloud Provider (CP) and Cloud Consumer (CC). The ontology hierarchy in left part of Fig. 6 shows that these classes are modelled as sub-classes of the ArchiMate concept BusinessRole.

The right part of Fig. 6 shows the different kinds of services labelled with (b). Support and consulting services are classified as business services while cloud services represent a specialization of application services. A third part of the conceptual model (c) shows the (Cloud) Service Level Agreement (SLA), which is defined as sub-concept of Contract, which itself is a Business Object in ArchiMate.

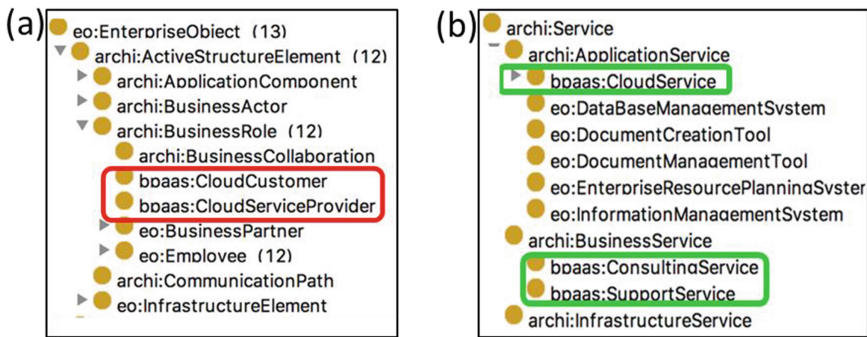


Fig. 6. Embedding cloud concepts into ArchiMEO

## 7 Alignment Support

The inference component for Smart Business and IT Alignment in the Cloud encompasses queries and rules to answer the already determined competency questions. These can be complex questions like: “Are there existing workflows for my business process?” or simpler like “Does the pricing model of the service allow payments per month?” or “Does the provider offer consulting services?”

The alignment is based on inference rules to propose workflows, services, and cloud providers that satisfy the requirements specified in the service description model referring to business process models and workflow models.

The inference engine applies the inference rules based on the SPARQL Inferencing Notation (SPIN), a W3C specification submission [18]. As a simple example, the following is a query that collects all Cloud providers offering consulting services:

```
SELECT DISTINCT ?cloudprovider ?service
WHERE {
    ?cloudprovider rdf:type bpaas:CloudProvider.
    ?cloudprovider bpaas:CPoffersService ?service.
    ?service rdf:type bpaas:Consultingservice .
}
```



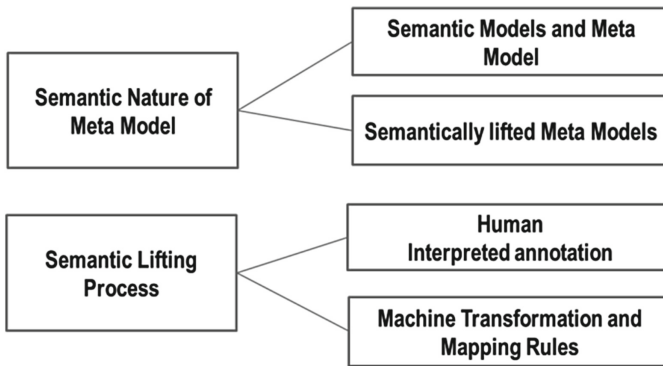
SPIN allows to link class definitions with SPARQL [31] queries to capture constraints and rules and to formalize the expected behaviour of those classes. This is used in the BPaaS Design Environment to specify mapping rules between elements. For example, it can be used to derive requirements for the location of data depending on the type of data:

*IF data contains personal data  
THEN location of data is EU.*

This rule uses the classes *personal data* and *location*, which are defined in the ArchiMEO and BPaaS Ontology. *Personal data* refers to instances of concrete process models.

## 8 Semantic Lifting

In order to apply the inference rules and queries for smart alignment, the content of the graphical models has to be translated into an ontology representation. This semantic enrichment of models and meta models corresponds to two approaches (Fig. 7).



**Fig. 7.** Realization types of semantic lifting

- *Semantic nature of the meta model*: The concepts of the graphical meta models have corresponding classes in the ontology. Some meta-models already include semantics, while for others their content has to be semantically-lifted or enriched. This approach is called semantic synchronization in Fig. 2.
- *Semantic Lifting Process*: Elements of the graphical models are annotated with knowledge from an ontology. This can be performed by humans via manual annotations, or by machines that follow pre-defined mapping rules. The semantic annotation, transformation and mapping of Fig. 2 belong to this approach.

In order to enable smart business and IT-Cloud alignment, a transformation is implemented, which creates a formal representation of the graphical models. The basic mechanism is based on the transformation approach of the LearnPAD project [9].

Semantic annotations allow for the human modeller to add semantics to the modelling element while creating the graphical models [16]. There are seven different ways of implementing semantic annotation.

- *Non-Supported direct linkage* provides the possibility to annotate by using freely chosen keywords that correspond to ontology classes.
- *Supported pre-defined direct linkage* is the mechanism of providing a list of possible semantic annotations. The content of the lists is taken from the ontology, thus the selection can be interpreted.
- *Supported Direct Linkage* is the scenario where tool support enables the possibility to select semantic concepts from the ontology model. The linkage is established via a so-called “semantic tunnel”, which can be implemented as Web-Services, which query the ontology tool to list all relevant semantic elements.
- *Indirect Linkage* describes the scenario where relevant concepts of the semantic target model are copied in a so-called “Semantic Transit Model”, in order to simplify the selection of a semantic concept within the source model environment.
- *Direct and Indirect Linkage*. This scenario combines the direct linkage as well as the indirect linkage. High level and preferable stable concepts are copied into the Semantic Transit Model, whereas the flexible direct linkage is provided for lower and probably more agile concepts.
- *Loose Coupling*. In this scenario an intermediate ontological layer is introduced that enables the loose linkage of concepts in contrast to the aforementioned direct linkage. Loose coupling does not introduce a new technical way of introducing semantics but introduces an intermediate ontology acting as reference.
- *Graphical Annotation*. This scenario uses the graphical position of objects for its annotation. It is hence the realization of a semantic whiteboard, where the background image is the model which is to be annotated. Semantic tags – similar to post-its – are placed close to a model object and hence annotate it.

## 9 Conclusion

The design environment of the smart Business and IT-Cloud Alignment uses informal (text), semi-formal (graphic) and formal (ontology, rules) knowledge representation languages to support the modelling and provisioning of BPaaS.

Business scenarios were analysed and competency questions were derived in order to determine the scope of the modelling framework. The BPaaS modelling method was implemented in the ADOxx meta-modelling platform. The model types were extended with algorithms and mechanisms for semantic lifting to connect the graphical models with the BPaaS ontology. The BPaaS Ontology contains the relevant classes for the smart Business and IT-Cloud alignment. A first version of the prototype is available free for download from the project website (<http://www.cloudsocket.eu>).

The specification of the queries and alignment rules requires competence in ontology engineering. In a future version we plan to use the Decision Model Notation also for specification of the rules for service discovery, composition and alignment. They shall be translated into executable rules for better supporting the smart alignment of business and IT in the cloud.

**Acknowledgement.** The research leading to these results has received funding from the European Community's Framework Programme for Research and Innovation HORIZON 2020 (ICT-07-2014) under grant agreement number 644690 (CloudSocket).

## Reference

1. Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., Sintek, M.: Toward a technology for organizational memories. *IEEE Intell. Syst. Appl.* **13**(3), 40–48 (1998)
2. Amziani, M., Melliti, T., Tata, S.: A generic framework for service-based business process elasticity in the cloud. In: Barros, A., Gal, A., Kindler, E. (eds.) *BPM 2012*. LNCS, vol. 7481, pp. 194–199. Springer, Heidelberg (2012)
3. Bentounsi, M., Benbernou, S., Atallah, M.J.: Security-aware business process as a service by hiding provenance. *Comput. Stand. Interfaces* **44**(C), 220–233 (2016)
4. Cardoso, Y.C.: *Creation and Extension of Ontologies for Describing Communications in the Context of Organizations*. Universidade Nova de Lisboa (2010)
5. *Cloud Service Level Agreement Standardization Guidelines*. EC Cloud Select Industry Group (C-SIG), European Commission (2014)
6. De Bruijn, J.: *Using Ontologies - Enabling Knowledge Sharing and Reuse on the Semantic Web*. Technical report, DERI-2003-10-29, Digital Enterprise Research Institute (DERI), Galway, Ireland (2003)
7. De Leenheer, P., Mens, T.: *Ontology evolution*. In: Hepp, M., et al. (eds.) *Ontology Management - Semantic Web, Semantic Web Services, and Business Applications*, pp. 131–176. Springer Science + Business Media Inc., Berlin (2008)
8. Den Haan, J.: *An Enterprise Ontology based approach to Model-Driven Engineering*. TU Delft, Delft University of Technology, 15 October 2009. <http://repository.tudelft.nl/view/ir/uuid:e2093132-9db7-4cba-bc68-9355f93cb9e3/>
9. Emmenegger, S., Hinkelmann, K., Laurenzi, E., Thönssen, B., Witschel, H.F., Zhang, C.: *Workplace learning - providing recommendations of experts and learning resources in a context-sensitive and personalized manner*. In: *MODELSWARD 2016, Special Session on Learning Modeling in Complex Organizations*, Rome (2016)
10. Fox, M.S., Barbuceanu, M., Grüniger, M.: *An organisation ontology for enterprise modelling: preliminary concepts for linking structure and behaviour*. *Comput. Ind.* **29**(1–2), 123–134 (1996). doi:[10.1016/0166-3615\(95\)00079-8](https://doi.org/10.1016/0166-3615(95)00079-8)
11. Fox, M.S., Grüniger, M.: *Enterprise modeling*. *AI Mag.* **19**(3), 109 (1998)
12. Gordijn, J., Akkermans, H.: *Designing and Evaluating E-business Models*. *IEEE*, 11–17 July/August (2001). <http://doi.org/10.1109/5254.941353>
13. Grüniger, M., Fox, M.S.: *Methodology for the design and evaluation of ontologies*. In: *Industrial Engineering* (1995)
14. Hepp, M.: *A methodology for deriving OWL ontologies from products and service categorization standards*. In: *ECIS* (2005)
15. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: *Semantic business process management: a vision towards using semantic web services for business process management*. In: *ICEBE 2005: IEEE International Conference on E-Business Engineering*, pp. 535–540 (2005)
16. Hinkelmann, K., Gerber, A., Karagiannis, D., Thoenssen, B., van der Merwe, A., Woitsch, R.: *A new paradigm for the continuous alignment of business and IT: combining enterprise architecture modelling and enterprise ontology*. In: *Computers in Industry*, vol. 80 (2015). <http://doi.org/10.1016/j.compind.2015.07.009>

17. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics, pp. 5–10 (2015)
18. Knublauch, H., Hendler, J.A., Idehen, K.: SPIN - Overview and Motivation. W3C Member Submission. <http://www.w3.org/Submission/spin-overview/> (2011). Accessed 23 December 2015
19. Leppänen, M.: A context-based enterprise ontology. In: Guizzardi, G., Wagner, G. (eds.) Proceedings of the EDOC International Workshop on Vocabularies, Ontologies and Rules for the Enterprise (VORTE 2005), pp. 17–24. Enschede, Netherlands (2005)
20. Lynn, T., O’Carroll, N., Mooney, J., Helfert, M., Corcoran, D., Hunt, G., Van Der Werff, L., Morrison, J., Healy, P.: Towards a framework for defining and categorising business process-as-a-service (BPaaS). In: 21st International Product Development Management Conference (2014)
21. Noy, N.F., McGuinness, D.L.: Ontology development 101: a guide to creating your first ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Palo Alto (2001)
22. Papazoglou, M.P., van den Heuvel, W.J.: Blueprinting the cloud. *IEEE Internet Comput.* **15** (6), 74 (2011)
23. OMG. Business Motivation Model, Version 1.1. Object Management Group OMG (2010). <http://www.omg.org/spec/BMM/1.1/PDF/>
24. OMG. Business Process Model and Notation (BPMN) Version 2.0. Object Management Group OMG, Needham, MA (2011). <http://www.omg.org/spec/BPMN/2.0/PDF/>
25. OMG. Case Management Model and Notation (CMMN), Version 1.0. Object Management Group OMG, Needham, MA (2013). <http://www.omg.org/spec/CMMN/1.0/PDF>
26. OMG. Decision Model and Notation Version 1.0. Object Management Group OMG, Needham, MA (2015). <http://www.omg.org/spec/DMN/1.0/PDF>
27. The Open Group. TOGAF® Version 9.1. Van Haren Publishing (2011)
28. The Open Group. ArchiMate 2.1 Specification. The Open Group. <http://pubs.opengroup.org/architecture/archimate2-doc/> (2012). Accessed 3 November 2015
29. Uschold, M., Gruninger, M.: Ontologies: Principles, Methods and Applications. Technical report, University of Edinburgh Artificial Intelligence Institute AIAI, p. 191 (1996)
30. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The enterprise ontology. *Knowl. Eng. Rev.* **13**(1), 31–89 (1997)
31. W3C. SPARQL 1.1 Overview. W3C Recommendation. <http://www.w3.org/TR/sparql11-overview/> (2013). Accessed 30 December 2015
32. Zachman, J.A.: John Zachman’s Concise Definition of The Zachman Framework. Zachman International (2008). <http://zachmaninternational.com/>