Agnes Koschmider, Judith Michael (eds.): EMISA Workshop 2021
18 CEUR-WS.org Proceedings

Future Research Directions for Improved Service Modeling

Benedikt Reitemeyer[1], Hans-Georg Fill[2]

Abstract: The continuous emergence of new business requirements and technological innovations requires the constant adaptation of an enterprise's service architecture. Due to the involved complexity, various modeling approaches have been developed for facilitating this task. In this paper we review existing service modeling approaches and derive novel research directions for this area that consider the semantics of the content exchanged via services. In particular, we propose to integrate techniques from machine learning and semantic technologies to automatically analyze, propose and integrate suitable services.

Keywords: Enterprise Modeling; Services; Machine Learning

1 Introduction

Since the first occurrences of distributed web-based systems in enterprise architectures and the up-coming of service technologies, the field has constantly evolved [TBB03]. Especially the availability of light-weight approaches such as REST-based services led to a dramatic increase in service endpoints [Ou17]. Today, almost any major IT platform offers endpoints, typically in the form of web-based APIs. Thereby, one key challenge for enterprises is to maintain an overview of the services, according responsibilities, technical interfaces, and versions. Beneath those challenges, organizations need to consider, how new services are integrated in existing architectures, while keeping track of their evolution. For targeting these challenges, machine-supported enterprise modeling approaches help to reduce complexity and align business and technical aspects [BW07, La04, RF19, Re20].

In line with discussions for integrating enterprise modeling into everyday practices and easing the use of modeling techniques through automation [Sa18], we investigate in the following three research questions: 1) What is the current status in service modeling? 2) What needs to be added in the future to improve service modeling? and 3) How can techniques from machine learning and semantic technologies aid in this context? The remainder of the paper is structured as follows. In Section 2, fundamental terms for a common understanding are explained. Next, the current state of modeling services is derived

[1] University of Fribourg, Research Group Digitalization and Information Systems, Boulevard de Pérolles 90, 1700 Fribourg, Switzerland benedikt.reitemeyer@unifr.ch
[2] University of Fribourg, Research Group Digitalization and Information Systems, Boulevard de Pérolles 90, 1700 Fribourg, Switzerland hans-georg.fill@unifr.ch

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2867/paper3.pdf

in Section 3. Future directions for service modeling are outlined in Section 4 before we present a first conclusion in Section 5.

2 Foundations

Service orientation is an established paradigm in software development, as well as in orchestrating distributed systems. While primarily aiming at separating software possession and ownership from its use, distribution further includes inter-organizationally, locally, technically and functionally distributed endpoints. Discovering suitable services for human and technical users has become challenging, demanding internal and external service providers to offer information on functional and nonfunctional capabilities, technical interfaces (APIs), constraints, quality of service and cost [TBB03]. In the following we will therefore focus on the levels of service descriptions and service messaging.

In the past, several types of service descriptions emerged, differing in their models, formalisms and described properties. Syntactic descriptions, like the Web Service Description Language (WSDL)[3] include bindings, operations, input and output data types and endpoints. Semantic service descriptions, like OWL-S[4] focus on the behavior, using ontological structures and capabilities. For connecting the semantic and syntactic layer, middle layer descriptions like MicroWSMO, SA-REST or Semantic Annotations for WSDL (SAWSDL) have emerged [Sa13].

On the messaging level, the XML-based SOAP protocol transfers processing and infrastructure information, allowed transactions, security, and reliability requirements [PZL08]. For describing SOAP endpoints, the WS*-Languages like WSDL, SAWSDL, WSMO are used. REpresentational State Transfer (REST) is an architectural style for identifying resources through URIs and manipulating them through the stateless HTTP operations. Service messages in REST contain metadata about the resources and are used to access the resources in a format-agnostic way [PZL08]. For describing REST services, hREST (HTML for RESTful services) [Du15] on the syntactical level, as well as MicroWSMO and WSMO on the semantic level are available.

3 Approaches for Modeling Services

For facilitating communication in business-IT alignment and service management and for reducing the involved complexity, various types of modeling approaches have been developed. We distinguish them by their abstraction levels: on the *enterprise architecture level*, on the *service interaction level*, and on the *service implementation level*. Modeling services on the enterprise architecture level focuses on functional requirements and the

[3] `https://www.w3.org/TR/wsdl/`

[4] `https://www.w3.org/Submission/OWL-S/`

holistic view on the enterprise for aligning business processes and services [La04]. Examples includes frameworks such as ArchiMate, ARIS and Zachman that have been widely adopted in industry and are typically supported by tools, e.g. [PB17].

On the interaction level, service modeling focuses on the technical alignment of services. This can also be part of EA models, depending on the required abstraction. Service interaction modeling has structural and behavioral aspects. The structural aspect targets the structure of the exchanged data entities and components of services. For example, Ed-douibi et al. [Ed19] developed an approach to connect OpenAPI specifications and UML models. The behavioral aspect concerns the execution of interactions. In this context, GraphQL as originally developed by Facebook, is increasingly gaining momentum. Rodriguez-Echeverria [REIC17] recently created and implemented an approach for generating GraphQL specifications from UML and IFML models.

A more implementation-oriented approach is Node-RED[5], aiming at connecting hardware devices, APIs and online services for designing flows between different nodes. MashUp modeling and web service recommendation systems designed for software developers are also an active research topic [Pe20]. Furthermore, UML models based on Web API specifications like OpenAPI [Ed19] and ArchiMate Application Cooperation Viewpoints[6] are actively used for modeling service implementations.

In summary, various service modeling approaches exist on different abstraction levels. What is however missing, is a machine-processable, semantic integration of service modeling that ranges over all abstraction levels, supporting human actors with complex modeling tasks in volatile environments. Machine-processable semantic integration between service descriptions and enterprise knowledge for example, would enable enterprise architects to receive recommendations for new or improved services in business-IT alignment that are automatically discovered on the Web or mined from existing architectures.

4 Novel Research Directions for Service Modeling

Currently available modeling approaches for service descriptions and service messaging on state-of-the-art technology stacks focus mainly on syntactic aspects, although semantic descriptions have been investigated in the past. The content provided by service endpoints is mostly not described in a machine-interpretable manner. Recently Farré et al. proposed a metamodel for GraphQL that has been enriched semantically with RDF annotations and therefore could be a starting point to machine-processing [FVA19].

Building on this idea, further research could aim at taking up concepts from process mining and workflow mining, as well as current approaches on service mining [VDA12] and extend them with semantics of the services' content. On the implementation level, service mining,

[5] https://nodered.org/
[6] https://pubs.opengroup.org/architecture/archimate3-doc/apdxc.html

similar to process mining, aims at discovering, monitoring and improving services based on event logs and representing them in UML. Such logs can be used to transition from the implementation to interaction level, covering in particular the interaction behavior between services.

For aligning business and services on the enterprise architecture level, additional semantics and structural information are necessary, especially for integrating external services and improving existing ones. Potential data sources for such information could be specifications like GraphQL schemas, as well as web service description languages, enriched with semantic information. For modeling the business-service-alignment, ArchiMate seems a suitable modeling language to start from, allowing the visual integration of business process and workflow steps with application components and data objects.

Future research should therefore aim at re-integrating semantic technologies in service modeling approaches, e.g. through enterprise ontologies [Hi16] or enterprise knowledge graphs together with machine learning. This research could improve the service modeling experience, through semi-automated modeling approaches [Re20] and for supporting users by suggesting suitable services based on current contexts, goals, principles and past modeling decisions.

A key challenge in this research direction is data sparsity, caused by shallow web service documentations and descriptions, thus requiring semantic enrichment for further processing. A further challenge on the instance level lies in the correlation of instances, as well as in analyzing services apart from their context [VDA12]. Additionally, we have to consider a constantly evolving technology stack. A third aspect are the rapidly changing internal and external environmental conditions. Internally, because organizations need to become increasingly agile, implementing scaled agile approaches [EP17]. Externally, in terms of data privacy, as well as related laws and regulations.

More concretely, we envision an improved mining of enterprise architectures, integrating semantic and syntactic information over different layers. For example in an Internet-of-Things context, Node-RED infrastructure could be used as source to develop EA models from hardware level to business level, based on an interconnecting enterprise knowledge graph. In service-based contexts without explicit hardware infrastructure like ERP or Case Management Systems, service mining, based on web service descriptions with semantic enrichment for enterprise knowledge graphs, could be used for service mining and connect the results to business processes and workflows, which could be mined themselves, thereby enabling the (semi-)automatic generation of enterprise architecture models in ArchiMate.

5 Conclusion and Outlook

In this paper we reviewed existing approaches for modeling service descriptions and service messaging on three abstraction levels and in light of novel technology stacks such as

GraphQL and REST. Future research directions may include the integration of semantic aspects as it had been done for previous technology stacks and the application of mining techniques for improving the service modeling experience.

References

[BW07] Braun, Christian; Winter, Robert: Integration of IT service management into enterprise architecture. In: Proceedings of the 2007 ACM symposium on Applied computing. pp. 1215–1219, 2007.

[Du15] Dumitru, Roman et al.: WSMO-Lite and hRESTS: Lightweight semantic annotations for Web services and RESTful APIs. Journal of Web Semantics, 31:39–58, 2015.

[Ed19] Ed-Douibi, Hamza; Izquierdo, Javier Luis Cánovas; Bordeleau, Francis; Cabot, Jordi: WAPIml: Towards a modeling infrastructure for Web APIs. In: MODELS-C. IEEE, pp. 748–752, 2019.

[EP17] Ebert, Christof; Paasivaara, Maria: Scaling agile. Ieee Software, 34(6):98–103, 2017.

[FVA19] Farré, Carles; Varga, Jovan; Almar, Robert: GraphQL schema generation for data-intensive web APIs. In: International Conference on Model and Data Engineering. Springer, pp. 184–194, 2019.

[Hi16] Hinkelmann, Knut; Gerber, Aurona; Karagiannis, Dimitris; Thoenssen, Barbara; Van der Merwe, Alta; Woitsch, Robert: A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology. Computers in Industry, 79:77–86, 2016.

[La04] Lankhorst, Marc M: Enterprise architecture modelling—the issue of integration. Advanced Engineering Informatics, 18(4):205–216, 2004.

[Ou17] Oumaziz, Mohamed A.; Belkhir, Abdelkarim; Vacher, Tristan; Beaudry, Eric; Blanc, Xavier; Falleri, Jean-Rémy; Moha, Naouel: Empirical Study on REST APIs Usage in Android Mobile Applications. In: Service-Oriented Computing. Springer, pp. 614–622, 2017.

[PB17] Pittl, Benedikt; Bork, Dominik: Modeling Digital Enterprise Ecosystems with ArchiMate: A Mobility Provision Case Study. In: 5th International Conference on Serviceology. pp. 178–189, July 2017.

[Pe20] Pelekh, I.; Berko, A.; Andrunyk, V.; Chyrun, L.; Dyyak, I.: Design of a System for Dynamic Integration of Weakly Structured Data Based on Mash-Up Technology. In: 2020 IEEE Third International Conference on Data Stream Mining Processing. pp. 420–425, 2020.

[PZL08] Pautasso, Cesare; Zimmermann, Olaf; Leymann, Frank: Restful web services vs."big“web services: making the right architectural decision. In: WWW Conference. pp. 805–814, 2008.

[Re20] Reitemeyer, Benedikt: Automatic Generation of Conceptual Enterprise Models. In: 2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW). IEEE, pp. 74–79, 2020.

[REIC17] Rodriguez-Echeverria, Roberto; Izquierdo, Javier Luis Cánovas; Cabot, Jordi: Towards a UML and IFML Mapping to GraphQL. In: International Conference on Web Engineering. pp. 149–155, 2017.

[RF19] Reitemeyer, Benedikt; Fill, Hans-Georg: Ontology-driven enterprise modeling: A plugin for the protégé platform. In: Enterprise, Business-Process and Information Systems Modeling, pp. 212–226. Springer, 2019.

[Sa13] Sangers, Jordy; Frasincar, Flavius; Hogenboom, Frederik; Chepegin, Vadim: Semantic web service discovery using natural language processing techniques. Expert Systems with Applications, 40(11):4660–4671, 2013.

[Sa18] Sandkuhl, Kurt et al.: From expert discipline to common practice: a vision and research agenda for extending the reach of enterprise modeling. BISE Journal, 60(1):69–80, 2018.

[TBB03] Turner, Mark; Budgen, David; Brereton, Pearl: Turning software into a service. Computer, 36(10):38–44, 2003.

[VDA12] Van Der Aalst, Wil: Service mining: Using process mining to discover, check, and improve service behavior. IEEE Transactions on Services Computing, 6(4):525–535, 2012.