SeMFIS: A Flexible Engineering Platform for Semantic Annotations of Conceptual Models

Editor(s): Pascal Hitzler and Krzysztof Janowicz
Solicited review(s): N.a.
Open review(s): N.a.

Hans-Georg Fill *
University of Vienna - Research Group Knowledge Engineering, Waehringerstrasse 29, 1090 Vienna, Austria
E-Mail: hans-georg.fill@univie.ac.at

Abstract. In this paper, we present SeMFIS – a flexible engineering platform for semantic annotations of conceptual models. Conceptual models have in the past been used for many purposes in the context of information systems' engineering. These purposes include for example the elicitation of requirements, the simulation of the behavior of future information systems, the generation of code or the interaction with information systems through models at runtime. Semantic annotations of conceptual models constitute a recently established approach for dynamically extending the semantic representation and semantic analysis scope of conceptual modeling languages. Thereby, elements in conceptual models are linked to concepts in ontologies via annotations. Thus, additional knowledge aspects can be represented without modifications of the modeling languages. These aspects can then be analyzed using queries, specifically designed algorithms or external tools and services. At its core, SeMFIS provides a set of meta models for visually representing ontologies and semantic annotations as models. In addition, the tool contains an analysis component, a web service interface, and an import/export component to query and exchange model information. SeMFIS has been implemented using the freely available ADOxx meta modeling platform. It can thus be easily added to the large variety of other modeling methods based on this platform or used as an additional service for other tools. We present the main features of SeMFIS and briefly discuss use cases where it has been applied. SeMFIS is freely available via the OMiLAB.org website at `http://www.omilab.org/web/semfis`.

Keywords: Conceptual Models, Semantic Annotation, Ontologies, Analysis

1. Introduction

Conceptual models have a long tradition in supporting the engineering and analysis of information systems [27,39]. They have been used for example in requirements engineering for enabling the communication between different stakeholders [30], in the context of model-driven software engineering for designing systems and generating code [7], for simulating the behavior of information systems [24], or as runtime models for supporting "dynamic state monitoring and control of systems during execution" [1][p.23].

When creating conceptual models it is reverted to modeling languages, which define the syntax, semantics, and notation of the constructs to be included in the models in a formal way [2,22]. As of today, a large number of modeling languages are available - well-known examples include BPMN [33], UML [32], IDEF [35], EPC [38], or iStar [40]. Although some of these languages are quite extensive and cover a vari-

*E-Mail: hans-georg.fill@univie.ac.at

ety of aspects, it is often required in practice to adapt their semantic scope to individual needs. This may for example result from changes in legal regulations – e.g. due to new reporting obligations that require the documentation of additional information [14] – or, because of new business requirements – e.g. when conducting benchmarks that require particular information structures for enabling industry-wide comparisons [9].

Although the semantic scope of a modeling language could be adapted through a re-design and re-implementation of the language, this is often unfavorable. Especially in industry scenarios where sometimes thousands of models have been created over long time periods and with considerable effort [34], changes in the underlying modeling language may lead to unexpected side effects. Even more so in the case of runtime models, where execution systems interact with the models controlling their behavior.

As a solution to this problem, *semantic annotations* have been introduced for conceptual models [8]. By linking elements of conceptual models to elements in ontologies, additional semantic information can be added without having to modify the underlying modeling language. Thereby, ontologies are chosen because of their nature as an "explicit specification of a conceptualization" [21] that can be shared [31] and that permits to process the annotations by machines [37].

In this paper we present SeMFIS (Semantic-based Modeling Framework for Information Systems) – a meta-model based engineering platform for semantic annotations of conceptual models that supports the representation and analysis of annotations with ontologies, and that can be easily adapted to arbitrary modeling languages and integrated with other tools and services. While previous publications have mainly focused on general requirements [11], on particular scenarios for using SeMFIS [9,10], and technical extensions [18], this paper discusses the modeling, query, and exchange functionalities provided by the most recent version of SeMFIS in detail.

The remainder of the paper is structured as follows. In section 2 we will give an overview of related tools for the semantic annotation of conceptual models. Subsequently, we describe the features of SeMFIS, including the provided modeling editors, query and import/export functionalities, and its extensibility in section 3. Section 4 outlines the architecture of SeMFIS based on the ADOxx meta modeling platform. In section 5 we describe two use cases for applying SeMFIS and discuss limitations of the tool in section 6. The paper is concluded with an outlook on the next steps.

2. Related Tools

Whereas the semantic annotation of textual, video, and audio resources has been discussed in a multitude of publications and has led to the development of several tools – see e.g. [25] for a recent evaluation – approaches and tools for annotating conceptual models are rare and often limited to specific types of conceptual modeling languages. Semantic annotators for textual or data resources such as Apache Stanbol[1], OntoText[2] and the like are not primarily directed towards the use for conceptual models. However, they may be used for specific parts of conceptual models, e.g. to determine concepts from an ontology in the labels of model elements that are expressed in natural language.

Regarding approaches that particularly address the semantic annotation of conceptual models, several tools can be found in the area of semantic business process management. The original motivation for semantic business process management was to combine business process management with semantic web services [23]. For this purpose software tools have been developed that permit to enrich business process models with semantic annotations and subsequently discover web services that match these semantic descriptions. Examples include the Maestro tool [3], an extension of the ARIS modeling toolkit [36], WSMO Studio [5], or Pro-SEAT [29]. However, as these tools are directed towards business processes, they are mostly limited to particular business process modeling and ontology languages. For example, Maestro and WSMO Studio support BPMN and the ARIS extension supports EPC. Maestro, ARIS and WSMO Studio use ontologies expressed in WSML/WSMO format, whereas PRO-SEAT supports OWL ontologies. PRO-SEAT works on models generated by the Metis platform, which permits to define arbitrary modeling languages. However, it is not integrated with the platform but rather an external tool for conducting semantic annotations based on Metis file exports. In addition, not all these tools are freely available thus hampering their use and further development in academic settings. From the mentioned tools only WSMO Studio is currently available under an open source license.

Although also the field of software engineering has discussed semantic annotations of conceptual models, e.g. for UML class diagrams using the XMI format [41] or for SoaML and ODM [26], correspond-

[1] https://stanbol.apache.org/
[2] http://www.ontotext.com

ing tools are to the best of our knowledge not publicly available.

Another direction that can be found in the context of conceptual models with semantic enrichments are tools that represent conceptual models as ontologies. In general, such approaches require the a-priori definition of mappings of a conceptual modeling language to an ontology language. Examples for tools following this direction include ICOM for creating and maintaining ontologies using conceptual models [19] and SemPeT, which provides an ontology for the semantic analysis of Petri nets [6].

In summary, the tools that have been investigated above for conducting semantic annotations of conceptual models are limited to very specific target domains. Furthermore, none of these tools with the exception of PRO-SEAT permits to create semantic annotations for arbitrary types of conceptual modeling languages. However, we could not find an openly accessible version of PRO-SEAT. In addition, none of the tools we found seems to offer interfaces for the programmatic access of the stored information using web interfaces. As far as could be accessed from the corresponding publications, none of the described tools is directed towards supporting industry scenarios with several thousands of models. A core requirement for such scenarios would be for example to provide highly performant storage solutions for the models, ontologies, and annotations.

3. SeMFIS Features

SeMFIS[3] is a flexible platform for engineering semantic annotations of conceptual models. The intention behind SeMFIS was to provide a link in the form of a software platform between the field of conceptual modeling and the field of ontologies. Thereby, profits can be gained from both sides: on the one hand, ontologies provide formal information structures and reasoning mechanisms that can be used to enrich conceptual models. On the other hand, the semi-formal style and visual editors traditionally used for conceptual models facilitate the interaction, in particular for non-technical users. In contrast to other semantic annotation tools for conceptual models, SeMFIS does not require a specific type of modeling language or the modification of an existing modeling language. Due to the decoupling of the semantic annotations in separate annotation models, SeMFIS can be easily added to existing modeling methods without affecting their structure nor behavior. As SeMFIS is implemented based on the industry-driven, freely available ADOxx meta modeling platform [15], it also provides a highly performant persistence layer in the form of a relational database as well as interfaces and a scripting language for programmatic access. In section 3.1 we will describe the model editors provided by SeMFIS. These are used to visually describe semantic annotations and for visualizing ontologies. Section 3.2 gives an overview on the scripting and analysis functionalities in SeMFIS including the programmatic access using a web service interface. In section 3.3 we will present import and export interfaces, in particular the interface to the Protégé platform.

3.1. Model Editors

The semantic annotation approach used in SeMFIS is based on theoretical elaborations discussed in [8]. In order to decouple the annotation information from the underlying conceptual models and ontologies, SeMFIS adds a distinct information structure for describing annotations. In this information structure references to elements in conceptual models and to ontologies are established. These references are then linked using annotator elements, which define the type of annotation. In this way, neither the modeling language for conceptual models nor the specification of ontologies has to be changed.

For representing semantic annotations and ontologies, SeMFIS uses a meta model-based approach. As shown in figure 1 the SeMFIS meta model comprises four model types: the semantic annotation model type, the frames ontology model type, the OWL ontology model type, and the term model type. The semantic annotation model type provides 'model reference' and 'connector reference' elements that can be used to refer to elements in conceptual models. Similarly, it contains the 'ontology reference' element to establish links to ontologies. Via the 'annotator' element connections between the reference elements to conceptual models and the ones to ontologies can be established. In addition, the type of annotation can be specified via the 'annotation type' attribute. In the pre-configured version of SeMFIS the following annotation types are available: 'Is equal to', 'Is broader than', 'Is narrower than', 'Is instance of', 'Is subclass of', 'Is superclass of', 'Is instance using fromClass', 'Is in-

[3] http://www.omilab.org/web/semfis

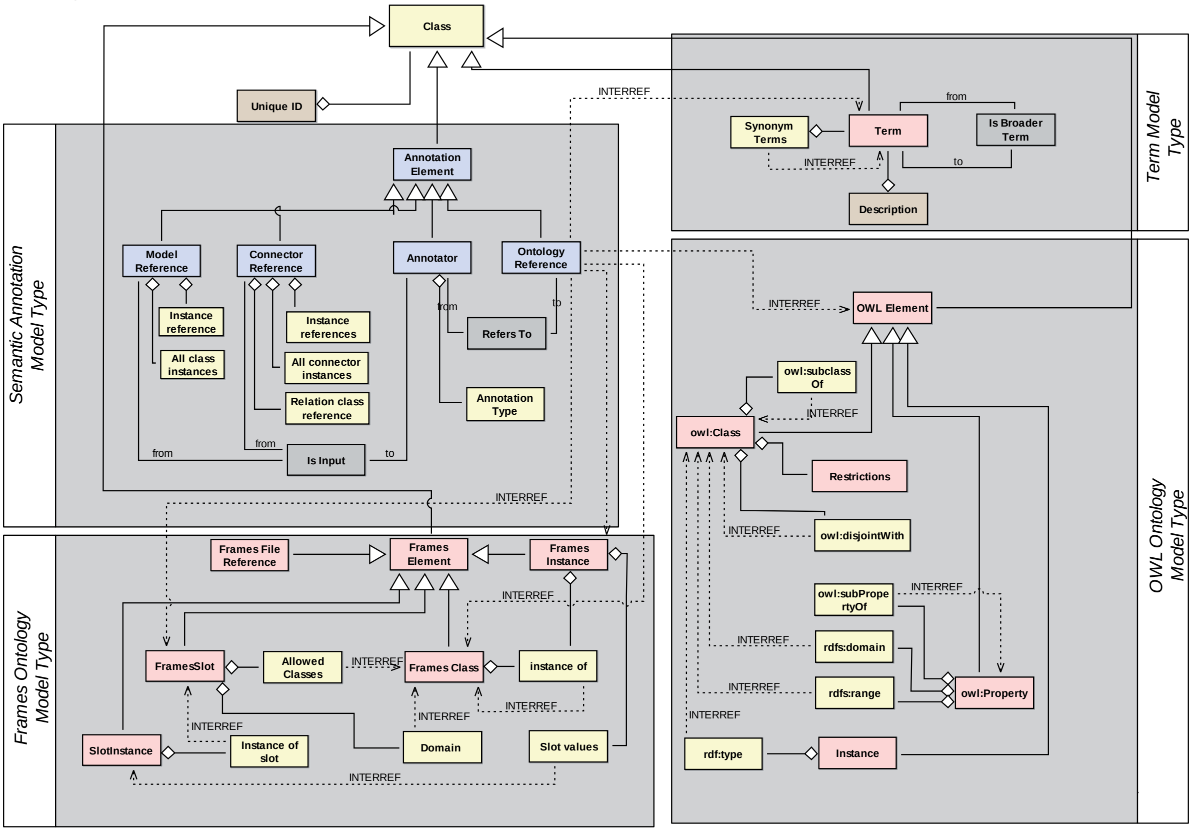

Fig. 1. SeMFIS Meta Model showing the SeMFIS Model Types

stance using toClass'. However, as will be discussed later these types can be easily adapted or extended to fit individual needs.

On the side of ontologies three different types of ontologies can be represented in SeMFIS: OWL ontologies that are today widely used based on the international standard issued by W3C, Frames ontologies as used in Protégé which follow the OKBC protocol [20], and controlled vocabularies in the form of the term model type which permits to express synonym and hypernym relationships between terms. For these ontology types SeMFIS provides the underlying information structures as shown in the meta model. It does however not provide reasoning mechanisms, rule engines, or further constraints regarding the processing of the semantics in the ontologies. If needed, these functionalities can however be easily added to SeMFIS using third-party tools and APIs via the subsequently discussed interfaces and scripting functionalities.

For all the above described model types visual model editors are provided by the SeMFIS modeling component (). In this way the information contained in the SeMFIS model instances is graphically represented and users can interact with the information on a visual level. In the screenshot shown in figure 2 examples for model editors are shown. By double-clicking on the graphical elements, so-called 'notebooks' can be opened for editing the attributes of the model elements.

To establish the linkage between SeMFIS and conceptual models, it can be chosen from two directions. Either, the SeMFIS meta models are added to an existing ADOxx-based modeling method[4]. This has for example been done for the pre-configured version of SeMFIS that is publicly available on the SeMFIS website. As shown in the screenshot in figure 2, SeMFIS has been added to a simplified version of the ADONIS BPMS modeling method [15][p.19]. This also encom-

[4]See the websites `http://www.adoxx.org` and `http://www.omilab.org/web/guest/projects` that offer currently around 40 ADOxx-based modeling methods including methods for UML, BPMN, ER, Petri Nets, iStar and many others.

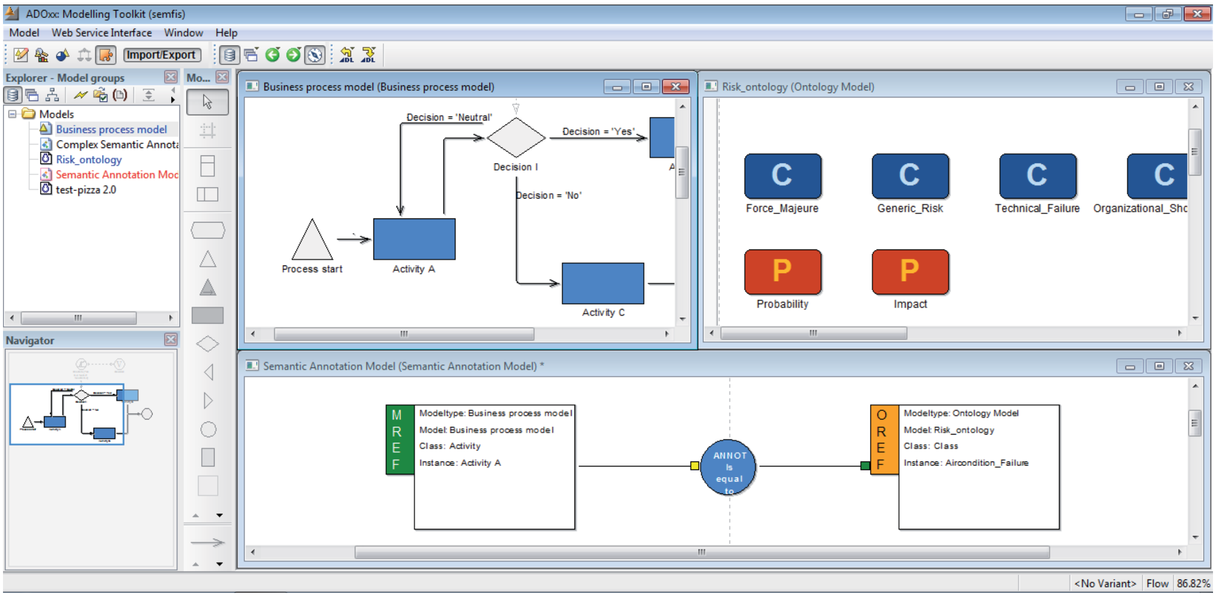

Fig. 2. Screenshot of SeMFIS showing Model Editors for a Business Process Model, an OWL Ontology Model, and a Semantic Annotation Model (from left to right, top to bottom)

passes to inherit functionality from other methods, e.g. the simulation algorithms for BPMS that can be accessed via ( ). Or, as a second direction, SeMFIS is used only as a repository for storing and analyzing information about semantic annotations and ontologies. Then, linkages to elements in conceptual models have to be established using the interfaces and exchange mechanisms below.

3.2. Scripting and Analysis Functionalities

As SeMFIS has been implemented on top of the ADOxx meta modeling platform [15], it can revert to a number of functionalities that are provided by the platform's components. This also includes scripting and analysis functionalities that have been made available for SeMFIS. Via the scripting functionality, statements in the domain-specific ADOscript language can be executed. This language permits to access almost any platform functionality in a programmatic way. It can thus be used for tasks such as constraint checking, model manipulation and analysis, report generation, interaction with third-party tools and APIs, user interaction via a set of pre-defined UI components and many more.

An example where ADOscript has been used to adapt the user interface are the specific menu entries in SeMFIS for starting/stopping the web service access. By adding the ADOscript statement shown in algorithm 1 in the SeMFIS library definition, a user is shown a menu entry in the Import/Export component ( ) to start the web service interface on port 1080 - see also figure 2. Upon the start of the web service, SOAP calls containing ADOscript statements can be made to the platform[5].

ITEM *"∼Start Web Service..."*
importexport: *"∼Web Service Interface"*
CC *"AdoScript"* **SERVICE start port:** *1080*
output: *statusbar*

Algorithm 1. ADOscript Definition for a Web Service Menu Entry in SeMFIS

In addition to analyses with ADOscript, SeMFIS also integrates the analysis component of the ADOxx platform ( ). By using this component queries expressed in AQL (ADOxx Query Language) can be composed and executed. Although this language is currently not as powerful as SQL - e.g. joins are not yet available - it can be used for easily gathering information from the models. An example for a query is given in figure 3. It shows a query definition targeting all ontology reference elements in a semantic annotation model. The results of these queries can then either be

[5]For further information on the web service interface see http://www.adoxx.org/live/aodxx-web-service

interpreted by users, exported in different file formats such as rtf, csv, or html or accessed programmatically with ADOscript for further processing.

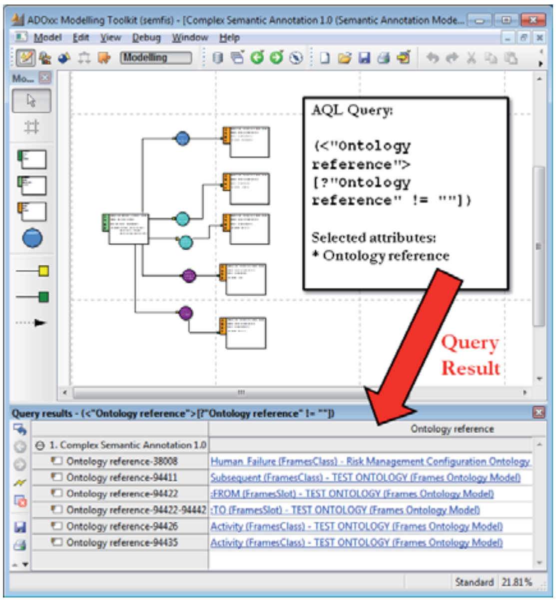

Fig. 3. Example for an AQL Query in SeMFIS retrieving Ontology Reference Elements in a Semantic Annotation Model Instance

3.3. Import and Export Interfaces

SeMFIS provides import and export interfaces for exchanging model information in different file formats. In addition, also an export plugin for the Protégé platform is available. If necessary, these interfaces can also be accessed via ADOscript, e.g. to support additional import/export formats.

3.3.1. Generic XML/ADL Import/Export

The generic XML import/export interface is used to exchange information from arbitrary model types. It is generic in the sense that it does not have to be adapted for specific model types or if modifications are made to a model type. It is therefore well suited for exchanging information with other tools and platforms. Based on the underlying ADOxx platform, SeMFIS also comes with an integrated Java runtime environment and the Saxon XSLT and XQuery Processor. Thereby, transformations to other XML or file formats can be easily realized. Additionally, via the ADL import/export interface model, information in the proprietary ADOxx-ADL format can be exchanged with other ADOxx-based tools that do not offer an XML interface.

3.3.2. SeMFIS Export Plugin for Protégé

In order to facilitate the exchange of ontology information, a plugin for the Protégé ontology management toolkit has been developed. Protégé has been chosen due to its wide-spread adoption in many fields of science and industry, its large user base and its open accessibility and extensibility [20]. The plugin has recently been adapted in a student project to the Protégé 4.x desktop application. As shown by the screenshot in figure 4 the plugin is integrated in the Protégé desktop application in the form of a view extension.

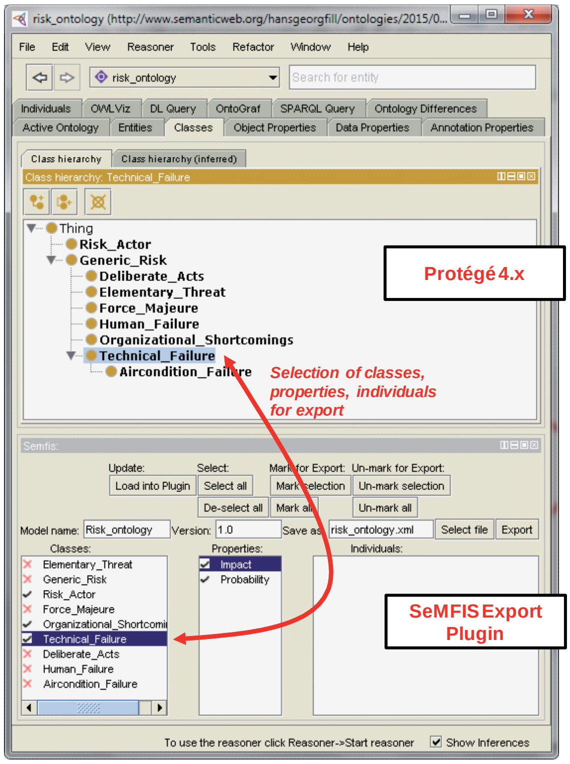

Fig. 4. Screenshot of the SeMFIS Protégé for Selecting Classes, Properties, and Individuals for Exporting them to SeMFIS

The plugin permits to load classes, properties, and instances from OWL ontologies in the plugin environment. Subsequently it can be chosen which of these elements should be exported to SeMFIS. The selected elements are then stored in a SeMFIS-compatible XML file, which can be directly imported in SeMFIS and visualized in the according OWL ontology model editor. The current implementation of the plugin and the SeMFIS meta model for OWL ontologies only support the major OWL constructs. It is planned however for the future to adapt both the meta model and the plugin, in particular also to support constructs from the OWL2 standard.

4. Architecture of SeMFIS

SeMFIS has been realized using the Microsoft Windows-based ADOxx meta modeling platform. ADOxx is professionally developed by BOC Group, a spin-off of the University of Vienna. It has been on the market for more than fifteen years and has since been used for a large number of research and industry projects [15]. The meta modeling approach of ADOxx is also the basis for the commercial ADONIS business process management toolkit[6], the ADOscore strategic management toolkit [28], and the ADOit enterprise architecture management platform[7], which are well recognized in industry [4]. Since a few years, the ADOxx 1.3 version is available free of charge and can be used for the implementation of academic and industrial modeling methods.

A high level overview of the architecture of the current SeMFIS version is shown in figure 5. At the bottom rests the repository with the modeling subsystem and a relational database. Although connectors for several databases are available for ADOxx - including Oracle and DB2 - the standard installation of SeMFIS is based on Microsoft SQLServer. The modeling subsystem is a Microsoft Windows application written in C++ and is responsible for handling the persistence of the models in the database, for executing ADOscript statements and for managing the user authentication. If necessary, the ADOxx platform underlying SeMFIS can also be configured for client-server scenarios. Thereby additional features are made available. These include simultaneous multi-user access to the models from several clients including detailed rights administration as well as a more advanced web service access featuring load-balancing mechanisms and automatically generated web service descriptions in WSDL.

On top of the repository operate the application components for a. *modeling*, which handles the model editors, b. *analysis*, which provides the AQL query component, c. the *web service interface* that accepts SOAP calls containing ADOscript statements, and

[6]http://www.adonis-community.com/
[7]http://www.adoit-community.com/

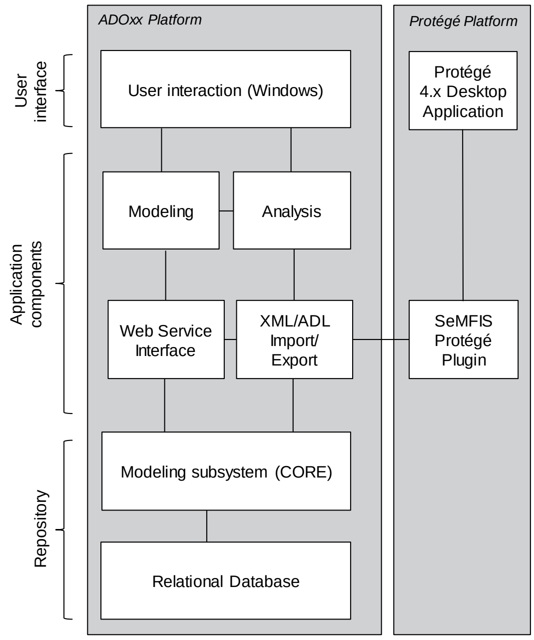

Fig. 5. SeMFIS Architecture based on ADOxx

d. for the *import/export* of model information via XML/ADL. The Protégé plugin is integrated via the XML interface of the import/export application component.

The user interaction is accomplished through a standard Microsoft Windows desktop user interface. Several aspects of the user interface can be customized using ADOscript - as has been shown above with the example for integrating the web service menu. This also includes adding functionalities for intercepting certain user actions, e.g. to enforce constraints when creating elements in models, to prevent the modification of certain models, or for triggering the automated execution of complex modification actions.

A particular feature of the SeMFIS architecture is its focus on extensibility and adaptability. As we will discuss in the following section in the context of the use cases, SeMFIS can be easily integrated with other tools and services. Through the customization features provided by ADOxx – see [15] for a detailed discussion – new model types, model elements, and attributes can be easily added in SeMFIS.

5. Use Cases

SeMFIS has been applied to several use cases and two industry projects in the context of enterprise information systems [13,16,17,8,9,10,11,18]. In the following we will briefly describe two of these use cases to illustrate the possibilities offered by SeMFIS. One use case will be taken from the area of risk management [10] and one from the area of semantic service discovery [18].

In [10] we deployed SeMFIS for the representation, analysis, and simulation of risks in business processes. Based on the semantic annotation approach of SeMFIS, models in a business process modeling language could be annotated with concepts from a risk knowledge base. This risk knowledge base was specified in the form of a frames ontology and contained concepts to describe risks and their impact including details on their probability distributions. By using ADOscript together with the XML export interface of SeMFIS, the annotation data was sent to a specifically created mediator component. This mediator component then processed the data by using the Protégé API for frames ontologies. Together with the Jess tab plugin for Protégé, the data was subsequently processed by a Jess rule engine to determine the effects of the assigned risks on the annotated business process elements.

By reverting to rules it could be easily defined which types of risks should be analyzed, e.g. based on the type of risk as expressed in the frame ontology. The result of the application of the rules were executable statements in ADOscript that specified how a certain modeling element would be affected by an assigned risk – e.g. that the execution time of an activity in a process is extended when the risk materializes. This information was fed back into the SeMFIS platform via ADOscript and used as input for a business process simulation algorithm. As a result, the effects of risks on business processes could be quantitatively assessed in the simulation runs.

The particular advantage of SeMFIS in this use case was twofold. First, it was not necessary to modify the business process modeling language in order to represent the risks and their effects on business processes. Therefore, none of the existing functionalities working on these models had to be modified nor even considered during implementation. Second, the availability of the ADOscript language and the XML interface permitted to easily interact with the third-party tools.

The second use case is positioned in the area of technology-oriented knowledge management and in particular the semantic discovery of web services [18]. In this use case SeMFIS was integrated in a service-oriented architecture that provided a web-based modeling component via Java applets and a web-based client of the Protégé platform. For this purpose the client-server configuration of ADOxx was enabled. Thereby several instances of SeMFIS services were made accessible via a common WSDL interface that distributed incoming calls through a load balancer service. On the client side, users could access the models stored in SeMFIS via a Java-based web modeller. To establish the link to Protégé an ontology/model transformation service was added to the architecture. This service resembled the SeMFIS-Protégé plugin described above. Instead of a manual XML import and export of OWL ontologies from Protégé, this service provided SOAP-based interfaces that directly fed the ontology information exported from Protégé to SeMFIS.

Concerning the meta models, the second use case only re-used the OWL ontology model type of SeMFIS. In addition, a model type for representing an extended version of BPMN diagrams and for representing the contents of WSDL files was added. The BPMN diagram was thereby extended with particular operation elements for representing web service calls. With these elements and the XML export interface, BPEL files could be generated that were executable on a BPEL workflow engine. Instead of using the semantic annotation model type provided by SeMFIS, the operation elements in BPMN were directly linked to the WSDL and the OWL ontology model types. The reason was that the BPMN diagram had already been extended so that the advantage of loose coupling via the annotation model type was not necessary. In this way operations in workflows expressed in BPMN could directly reference a web service call. In addition, requirements for such calls could also be described through linkages to the ontology. Similarly, operations in the WSDL could be linked to ontology concepts. A corresponding semantic discovery mechanism written in ADOscript could then propose WSDL operations whose linked ontology concepts matched those of the BPMN operations.

6. Discussion and Limitations

With the provision of SeMFIS as an open accessible and extendable platform we believe that we can contribute to the further advancement of approaches in the context of the semantic annotation of conceptual models. SeMFIS is thus positioned as a platform for engineering approaches working at the interface of semi-formal, visual representations in the form of conceptual models and rigid, formally defined ontologies.

With the set of meta models, exchange interfaces, and scripting and analysis mechanisms provided by SeMFIS, a wide variety of future applications may be realized. The set of functionalities that is part of the current version of SeMFIS has evolved in a bottom-up fashion from several research and industrial projects. At the same time many theoretical considerations have found their way into the described concepts, including in particular principles from service orientation such as loose coupling, re-usability, composability, and abstraction. SeMFIS is however not a static tool but rather a starting point for innovative endeavors that want to build upon a solid core that can be adapted to individual needs.

Apart from our own research projects, SeMFIS has also been used in several student projects at the University of Vienna. Most recently it was utilized in a course on conceptual models and ontologies at the summerschool on Next Generation Enterprise Modeling 2014 which was funded by the European Union[8]. In this course around 50 master and PhD students were made familiar with SeMFIS and used it in their projects. The feedback we gained from the students was very positive.

Regarding its limitations, one important constraint of SeMFIS is that it is only available as a Microsoft Windows application. This results from the fact that the ADOxx platform currently only supports this operating environment. Although this limitation may be partly overcome by using the web service interface, the visual editors cannot be re-used or need to be reimplemented as described above in the second use case. For users with other operating systems, the most common workaround is to run SeMFIS in a virtual machine that hosts Windows. This may not be an optimal solution but gives full access to the SeMFIS functionalities.

Another limitation of SeMFIS is that it does not yet support the most recent OWL2 standard as well as some axioms of the previous OWL standard including certain restriction types and cardinalities. We are however confident to resolve these issues in the oncoming versions of the platform.

[8] See the homepage of the summerschool: `http://www.omilab.org/web/guest/camp2014/`

7. Conclusion and Outlook

In this paper we have presented SeMFIS – a flexible engineering platform for semantic annotations of conceptual models. SeMFIS provides a set of meta models, exchange interfaces, and scripting and analyis mechanisms for annotating conceptual models with concepts from ontologies. In particular it offers a semantic annotation approach that can be applied to arbitrary modeling languages without requiring their adaptation. SeMFIS has been implemented using the freely available ADOxx platform and is available for free via `http://www.openmodels.at/web/semfis/`. For the future we plan to enhance the coverage of the OWL and OWL2 standards of the meta models provided in SeMFIS. In addition, further functionalities from current research projects are envisaged to be part of future SeMFIS releases. This includes for example the support of social network-based annotations and the integration of natural language processing functionalities [12].

References

[1] G. Blair, N. Bencomo, and R.B. France. Models@Run.Time. *IEEE Computer*, 42(10):22–27, 2009.

[2] D. Bork and H.-G. Fill. Formal Aspects of Enterprise Modeling Methods: A Comparison Framework. In *47th International Conference on System Sciences*. IEEE, 2014.

[3] M. Born, J. Hoffmann, T. Kaczmarek, M. Kowalkiewicz, I. Markovic, J. Scicluna, I. Weber, and X. Zhou. Semantic Annotation and Composition of Business Processes with Maestro. In *5th European Semantic Web Conference, ESWC 2008*. Springer, 2008.

[4] T. DeGennaro, A. Cullen, H. Peyret, and M. Cahill. The Forrester Wave (TM): EA Management Suites, Q2 2013. Technical report, Forrester Research, 2013. URL: `https://www.forrester.com/The+Forrester+Wave+EA+Management+Suites+Q2+2013/fulltext/-/E-RES90501`.

[5] M. Dimitrov, A. Simov, S. Stein, and M. Konstantinov. A BPMO based Semantic Business Process Modelling Environment. In *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007)*. CEUR Workshop Proceedings, 2007.

[6] M. Ehrig, A. Koschmider, and A. Oberweis. Measuring Similarity between Semantic Business Process Models. In *4th Asia-Pacific Conference on Conceptual Modelling (APCCM 2007)*, pages 71–80. ACM, 2007.

[7] P. Fettke and P. Loos. Model Driven Architecture (MDA). *Business and Information Systems Engineering*, 45(5):555–559, 2003.

[8] H.-G. Fill. On the Conceptualization of a Modeling Language for Semantic Model Annotations. In *Advanced Information Systems Engineering Workshops, CAiSE 2011*, pages 134–148. Springer, 2011.

[9] H.-G. Fill. Using Semantically Annotated Models for Supporting Business Process Benchmarking. In *10th International Conference on Perspectives in Business Informatics Research*, volume 90, pages 29–43. Springer, 2011.

[10] H.-G. Fill. An Approach for Analyzing the Effects of Risks on Business Processes Using Semantic Annotations. In *European Conference on Information Systems 2012*. AIS, 2012.

[11] H.-G. Fill. SeMFIS: A Tool for Managing Semantic Conceptual Models. In H. Kern, J.-P. Tolvanen, and P. Bottoni, editors, *Workshop on Graphical Modeling Language Development*. ECMFA, 2012.

[12] H.-G. Fill. On the Social Network based Semantic Annotation of Conceptual Models. In *7th International Conference on Knowledge Science, Engineering and Management*, pages 138–149. Springer, 2014.

[13] H.-G. Fill and P. Burzynski. Integrating Ontology Models and Conceptual Models using a Meta Modeling Approach. In *11th International Protégé Conference*, 2009.

[14] H.-G. Fill, A. Gericke, D. Karagiannis, and R. Winter. Modeling for Integrated Enterprise Balancing (in German). *Business and Information Systems Engineering*, 06/2007:419–429, 2007.

[15] H.-G. Fill and D. Karagiannis. On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. *Enterprise Modelling and Information Systems Architectures*, 8(1):4–25, 2013.

[16] H.-G. Fill and I. Reischl. An Approach for Managing Clinical Trial Applications using Semantic Information Models. In *Business Process Management Workshops - BPM 2009*, pages 581–592, Ulm, Germany, 2009. Springer.

[17] H.-G. Fill and I. Reischl. Stepwise Semantic Enrichment in Health-related Public Management by Using Semantic Information Models. In S. Smolnik, F. Teuteberg, and O. Thomas, editors, *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, volume 195–212. IGI Press, 2011.

[18] H.-G. Fill, D. Schremser, and D. Karagiannis. A Generic Approach for the Semantic Annotation of Conceptual Models using a Service-oriented Architecture. *International Journal of Knowledge Management*, 9(1):76–88, 2013.

[19] P.R. Fillottrani, E. Franconi, and S. Tessaris. The ICOM 3.0 Intelligent Conceptual Modelling tool and methodology. *Semantic Web - Interoperability, Usability, Applicability*, 3(3):1–14, 2012.

[20] J.H. Gennari, M. A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubezy, H. Eriksson, N.F. Noy, and S.W. Tu. The evolution of Protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58:89–123, 2003.

[21] T. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

[22] D. Harel and B. Rumpe. Modeling Languages: Syntax, Semantics and All That Stuff - Part I: The Basic Stuff. Technical Report MCS00-16, The Weizmann Institute of Science, August 22, 2000 2000.

[23] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel. Semantic business process management: A vision towards using semantic web services for business process management. In *IEEE International Conference on e-Business Engineering, 2005. ICEBE 2005*, pages 535–540, 2005.

[24] J. Herbst, S. Junginger, and H. Kühn. *Simulation in Financial Services with the Business Process Management System ADONIS*. Society for Computer Simulation, 1997.

[25] S. Joksimovic, J. Jovanovic, D. Gasevic, A. Zouaq, and Z. Jeremic. An empirical evaluation of ontology-based semantic annotators. In *7th International Conference on Knowledge Capture*, pages 109–112. ACM, 2013.

[26] X. Juicheng, B. Zhaoyang, A.J. Berre, and O.C. Brovig. Model Driven Interoperability through Semantic Annotations using SoaML and ODM. *Information Control Problems in Manufacturing*, 13(0314), 2009.

[27] R. Kaschek. On the evolution of conceptual modeling. In *Dagstuhl Seminar Proceedings*, volume 08181, 2008.

[28] C. Lichka, H. Kühn, and D. Karagiannis. IT based Balanced Scorecard (in German). *wisu*, (7):915–918, 2002.

[29] Y. Lin. *Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability*. PhD thesis, 2008.

[30] J. Mylopoulos. Conceptual Modeling and Telos. In P. Loucopoulos and R. Zicari, editors, *Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development*, pages 49–68. Wiley, 1992.

[31] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W.R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):36–56, 1991.

[32] Object Management Group OMG. OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2, 2007. `http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/` accessed 01-03-2011.

[33] Object Management Group (OMG). Business Process Model and Notation (BPMN) Version 2.0, 2011. `http://www.omg.org/spec/BPMN/2.0/PDF/` accessed 01-03-2011.

[34] M. Rosemann. Potential pitfalls of process modeling: part B. *Business Process Management Journal*, 12(3):377–384, 2006.

[35] Softtech Inc. Integrated Computer-Aided Manufacturing (ICAM) - Architecture Part II - Volume IV Function Modeling Manual ($IDEF_0$). Technical report, 1981. `http://www.dtic.mil/dtic/tr/fulltext/u2/b062457.pdf` accessed 04-09-2012.

[36] S. Stein, C. Stamber, and M. El Kharbili. ARIS for Semantic Business Process Management . In *Business Process Management Workshops*, pages 498–509. Springer, 2009.

[37] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4:14–28, 2006.

[38] W. M. P. Van der Aalst. Formalization and Verification of Event-driven Process Chains. *Information and Software Technology*, 41(10):639–650, 1999.

[39] Y. Wand and R. Weber. Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, 13(4):363–376, 2002.

[40] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos. *Social Modeling for Requirements Engineering*. MIT Press, 2011.

[41] W. Yuxin and L. Hongyu. Adding Semantic Annotation to UML Class Diagram. *2010 International Conference on Computer Application and System Modeling*, 2010.