A Modeling Environment for Visual SWRL Rules based on the SeMFIS Platform*

Hans-Georg Fill[1], Benedikt Pittl[2], and Gerald Honegger[2]

[1] University of Bamberg, 96047 Bamberg, Germany
[2] University of Vienna, 1090 Vienna, Austria

Abstract. The representation and processing of semantic information can today be accomplished using a wide range of formalisms. Rule-based approaches are not only a well-known but also easy to use technique. Most approaches rely on a textual specification of rules that can be processed by an according rule engine. For simplifying the specification and understanding of rules by domain experts, we present a visual model editor for rules based on the W3C SWRL recommendation. The goal of this approach is to provide a means for a visual interaction with rule-based systems, while at the same time preserving full expressiveness. The visual language for SWRL rules has been implemented on the SeMFIS platform. In addition, serialization and de-serialization mechanisms have been added for OWLXML and SWRLXML formats. As there are currently no official conformance tests available for SWRL, the approach has been evaluated using W3C sample set for SWRL rules.

1 Introduction

Rule-based formalisms have been used in the past as a foundation for creating expert systems [15] and have recently come to a revival in the context of web-based applications and linked data [14]. In the context of business information systems, rules have amongst their many applications been used for detailing the branching in business processes, allocating resources, or constraint and compliance checking [2, 13]. They are thus an established and sound technique for the declarative specification of decisions based on given facts. Furthermore, rules are one of the main pillars for realizing the semantic web [11]. The rule standardization effort of the W3C resulted in the SWRL (Semantic Web Rule Language) recommendation with the special aim on inferring new knowledge from existing knowledge bases in the form of ontologies [11].

For creating and modifying SWRL rules it is common to use text based editors like the Protégé SWRL tab [11]. An alternative approach is the use of visual languages [9]. In this way, rule languages can be integrated in enterprise modeling environments, thus leveraging the burden from business users to express rules in a textual syntax. We thus decided to create a visual modeling language for rules according to the most recent SWRL standard [5] and implement it in the form of a prototype.

* Springer Link: https://link.springer.com/chapter/10.1007/978-3-319-59144-5_30

The remainder of the paper is structured as follows: In section 2 the design of the modeling language is described followed by elaborations on it significance to research and practice in sections 3 and 4. A first evaluation of the language is presented in section 5. The paper ends with a conclusion in section 6.

2 Design

In this section we describe the motivation for the design of our prototype, potential use cases, intended user groups as well as the list of features. To the best of our knowledge, a visual language for a complete representation of SWRL rules is still missing. The visual language for SWRL proposed in [7, 8] took a first step in this direction. However, it did not provide explicit support for the different types of built-ins available in SWRL and neither dealt with the import of SWRL rules as models. Also the implementation introduced in [1] is incomplete as it misses the representation of atoms and OWL constructs. It neither includes export mechanisms. The Protégé plugin Axiome visualizes SWRL rules in the form of graphs [4]. However, the plugin is a pure visualization tool so that modifications of rules in the graphical representation are not possible. It thus seems favorable to develop a modeling language for SWRL that covers all aspects of this language. Such a modeling language can be used both for use cases dealing with the analysis of existing SWRL rules, e.g. to enhance the understanding of a rule, as well as for use cases where SWRL rules have to be specified by non-technical experts, e.g. to conduct compliance checking by business users [13]. As the user group for which we intended to provide the visual modeling language for SWRL were people familiar with enterprise modeling but not necessarily with the modeling of rules, the goal was to realize a language that includes all necessary constructs in a visual form. We thus realized the following features in our prototype: (i) Visual modeling of all atoms as defined in the SWRL standard (ii) Explicit representation of variables and data values (iii) Linkage to visually represented OWL ontologies using the previously developed SeMFIS modeling language [3] (iv) Import and export of visual SWRL models to SWRL XML and OWL XML formats (v) Implementation of the modeling language on the SeMFIS platform The underlying metamodel of our visual language can not pretend in detail due to the space constraints. However, the most important classes are shown in table 1. For the design of the visual notation we took into account the principles of Moody [10]. For example, to ensure semiotic clarity we mapped each core concept of SWRL to one modeling element. Therefore, each atom defined in the SWRL standard is represented with a single modeling element in our visual language and vice versa. We also applied the dual theory which enforces that graphical elements are enriched with a textual description [10]. This resulted in the visual notation shown in table 1. The textual information is dynamically added at run time as shown in figure 1. An example for a model instance based on the SWRL modeling language is shown in figure 1. It is taken from the W3C SWRL specification and would be specified in traditional textual syntax as follows: `hasStatus(?customer, Gold)` $\wedge$

Antecedent	The antecedent (condition) of a rule	A C	Class atom, e.g. Person(?x)	A ≠	Different Individuals Atom, e.g. differentFrom(?x,?y)
Consequent	The consequent of a rule	A i	Individual-valued Property Atom, e.g. hasRisk(?x,?y)	A =	Same Individuals Atom, e.g. sameAs(?x,?y)
IndividualConstant	Constant representing a data value	A d	Datavalued Property Atom, e.g. hasName(?x,?y)		MetaInfo for describing meta information about a rule, e.g. about imports
IndividualVariable	Variable for storing individual values	A D	Datarange Atom with datatype	TERM	Built-In Term used as reference for composing built-in parameters
DataVariable	Variable for storing data values	A DL	Datarange Atom with literal list		Relation for connecting atoms to antecedents and consequents
DataConstant	Constant representing a data value	A B	Built-In Atom, e.g. greaterThan(?age, 17)		Relation for connecting antecedents and consequents

Table 1. Excerpt of the Elements Used for the SWRL Modeling Language

`hasTotalPurchase(?customer, ?total)` ∧ `swrlb:greaterThanOrEqual (?total, 500) -> hasDiscount(?customer, 10)`. In addition to the atoms connected to the antecedent and consequent elements, also the separate elements for representing variables, data values and builtIn-terms are shown.

3 Significance to Research

Our work has the following two main contributions to research. First, it extends previous research on visual languages for SWRL rules - e.g. in [8, 1, 4] - by providing a visual representation of SWRL rules that includes all elements of the W3C recommendation in a formal manner. Second, the prototypical implementation will be integrated in the SeMFIS platform and provided as open-source to the scientific community via the OMiLAB initiative [6]- `http://semfis-platform.org`. In this way other researchers will be able to use the modeling language or extend it for their purposes.

4 Significance to Practice

From an industry perspective, the simplification of user interfaces for interacting with complex formalisms is considered as essential for supporting non-technical business users. This is also the case for the described prototype that enables users to specify SWRL rules by using a set of pre-configured elements instead of having to compose rules in a textual syntax. Furthermore, the described prototype includes mechanisms for importing and exporting SWRL rule models to the SWRL XML and OWL XML formats. This enables the interoperability with applications such as Protégé that build upon these standards, which is one of the core requirements of industrial applications.

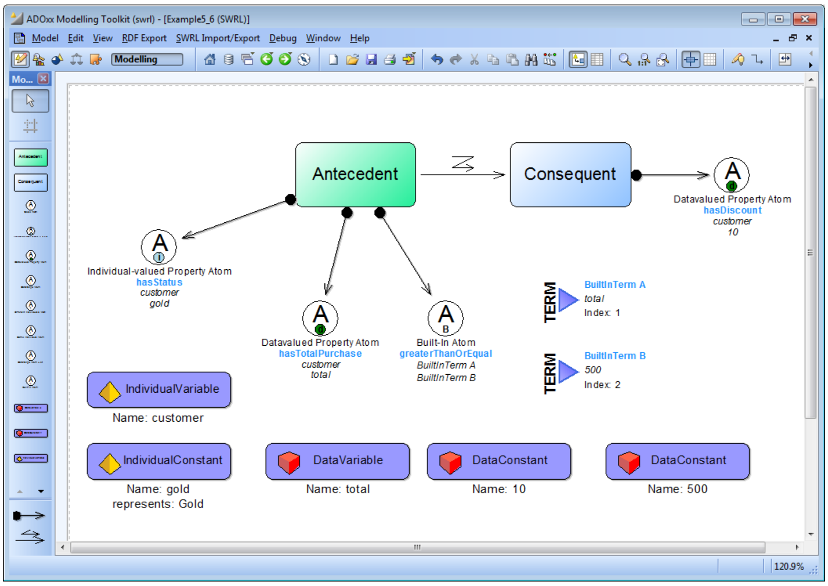

Fig. 1. Example Model for a SWRL Rule from the W3C Sample Set in the SeMFIS Modeling Environment

5 Evaluation

For the evaluation of artifacts in design-oriented and engineering research, it can be chosen from several paths [12]. In the case of our prototype, the most important aspect was to assess whether the models created with it conform to the recommendation by W3C for SWRL[3]. Therefore, we evaluated the completeness of our approach by creating the sample models specified in the W3C recommendation[4] and compared their XML serialization to the original code samples. Thereby, it could be verified that all examples were correctly represented. However, SeMFIS' OWL language requires an update in order to represent datatype attributes within restriction elements (see example 4 in the specification). For evaluating the completeness of our visual language we compared the supported language concepts to the concepts described in the SWRL specification. Currently, the support of annotation elements is limited. We will add this function-

[3] A screencast that illustrates the usage of the model editor and the import/export mechanisms can be found here: `http://semfis-platform.org/swrl/Screencast/`.

[4] The examples can be found at: `https://www.w3.org/Submission/SWRL/#5.1`

ality within the next SeMFIS release. Further evaluations will include aspects such as the performance of the transformation algorithms as well as user-related aspects, e.g. to assess the ease of use of the modeling language.

6 Conclusion and Outlook

In this paper we introduced a visual modeling language for SWRL. The language was implemented as a prototype using the SeMFIS platform. This allows us to create SWRL models complying to the W3C recommendation. In our further research we plan to support additional serialization formats such as RDF-XML.

References

1. Bak, J., Nowak, M., Jedrzejek, C.: Graph-based Editor for SWRL Rule Bases. In: RuleML@ChallengeEnriched 2013. CEUR (2013)
2. Feldkamp, D., Hinkelmann, K., Thoenssen, B.: KISS - Knowledge-intensive Service Support for Agile Process Management. In: SBPM Workshop 2007. CEUR (2007)
3. Fill, H.G.: SeMFIS: A Flexible Engineering Platform for Semantic Annotations of Conceptual Models. accepted for Semantic Web (SWJ) (2016)
4. Hassanpour, S., O'Connor, M.J., Das, A.K.: A Rule Management and Elicitation Tool for SWRL Rule Bases. In: RuleML Challenge (2009)
5. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML (2004)
6. Karagiannis, D., Buchmann, R.A., Burzynski, P., Reimer, U., Walch, M.: Fundamental conceptual modeling languages in omilab. In: Domain-Specific Conceptual Modeling, pp. 3–30. Springer (2016)
7. Leutgeb, A., Utz, W., Woitsch, R., Fill, H.G.: Adaptive processes in e-government - a field report about semantic-based approaches from the eu-project fit. In: ICEIS'2007. pp. 264–269. INSTICC (2007)
8. Leutgeb, A.: The business rules method : a modeling method for adaptive processes - Master Thesis. University of Vienna, Wien (2007)
9. Lukichev, S., Wagner, G.: Visual rules modeling. In: International Andrei Ershov Conf. on Perspectives of System Informatics. pp. 467–473. Springer (2006)
10. Moody, D.: The physics of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering 35(6), 756–779 (2009)
11. O'Connor, M., Knublauch, H., Tu, S., Musen, M.: Writing rules for the semantic web using SWRL and Jess. Protégé With Rules WS, Madrid (2005)
12. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2007)
13. Pham, T., Le Than, N.: Checking the Compliance of Business Processes and Business Rules Using OWL 2 Ontology and SWRL. In: AECIA Conference 2015. pp. 11–20. Springer (2016)
14. Stadtmueller, S., Speiser, S., Harth, A., Studer, R.: Data-fu: A language and an interpreter for interaction with read/write linked data. In: World Wide Web Conference. pp. 1225–1236 (2013)
15. Studer, R., Benjamins, R., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25, 161–197 (1998)