Business Rules for Law in Different Forms: Comparing the Applicability of the Semantic Web and Relation Algebra for Implementing Legislative Policies

Pim BOS [a], Lloyd RUTLEDGE [a,1], Rieks JOOSTEN [b] and Stef JOOSTEN [a]
[a] *Open Universiteit, The Netherlands*
[b] *TNO, The Netherlands*

Abstract. Laws and policies are forms of rules captured in natural language. This suggests that they are suitable for further specification in forms of formal logic. Such formalizations include the Ampersand application of relation algebra and the Semantic Web. Ampersand generates information system interfaces from data constraints defined with relation algebra. The Semantic Web, on the other hand, defines the inferencing of new data based on data models and rules defined with Horn clauses. This study examines whether these two forms can apply to implementing the laws and policies regarding a given legal text case study, how the two implementations differ from each other and how they can complement each other.

Keywords. Semantic Web, SWRL, business rules, relation algebra, Ampersand, legal texts

Introduction

Business rules define agreements regarding human interaction in business contexts. Several logical formalisms and computer systems exist for defining and enforcing business rules. Laws are essentially rules for human interaction in society. Compared to business agreements, laws have more focus on procedures and the constraints and requirements in interaction they involve.

This work applies to a case study about the Dutch law regarding the Certificate of Good Conduct (*Verklaring Omtrent Gedrag* or VOG). The VOG law [1] and its associated policies [2] prescribe restrictive rules and logical deductions from given information that lead to the decision of whether or not to issue a VOG for a given applicant. We apply these formalized rules here to information systems for the VOG that guide its data input and then processes the data entered.

In earlier work, we apply relation algebra as a constraint-based approach for defining business rules that drive specification of information systems that execute and enforce those rules [3]. The Semantic Web is another approach for rules and data

[1] Corresponding Author.

systems that focusses on inference for defining relations between objects in an information system [4]. That is, while relation algebra specifies what must always be true in relations between data objects, the Semantic Web specifies how to infer new relations from existing relations between data objects.

This work compares the applicability of these two different forms of the formalization of concepts and rules to the VOG. It considers whether these forms are useful for the VOG case and where the two types differ from or complement each other. In particular, we compare for both approaches they logical expressivity and the capabilities of selected tools that implement each. The result of this study may help organizations in similar situations to make choices in how to formalize business data and rules. In addition, the study is a step towards further research comparing forms of formalization of business rules.

1. Related Work

This section discusses approached for implementing business rules that provide a foundation for this work. First, we present related work regarding business rules from other technologies that the ones we apply here. The following subsections then describe publications regarding the technologies this work does apply: constraint-based logic and the Semantic Web.

1.1. Various Business Rules Approaches

The Business Rules Manifesto describes the basics of working with business rules [5]. In addition, Business Rule Management (BRM) describes the activities required to manage business rules fully and effectively for the execution of corporate strategy and improvement of business practices [6]. The BRM describes the importance of expressing business rules in a formal way. Doing so in a clear manner helps to avoid confusion and create value for all parties involved.

The Object Management Group (OMG) made a step toward the formalization of natural language with the standard called Semantics of Business Vocabulary and Business Rules (SBVR), which combines the use of natural and formal language [7]. While SBVR offers structure and meaning to business rules for formulation in the natural language of the user, it is not sufficient for making IT systems.

In addition to these natural languages, there are formalized languages for expressing business rules. The best known is probably Unified Modeling Language (UML) [8], developed by the OMG, which defines the structure of data that parties like businesses process. UML is relatively approachable for the lay end user, but with its semi-formal notation, there is potential for inconsistencies and differences in interpretation. OMG then defined constraints on UML-defined data models with Object Constraint Language (OCL) [9]. While OCL has proven useful for communicating rules between system developers, few implementations process it for validating or executing the rules. The next subsection discusses work on processing business rules constraint that leads to our work with Ampersand.

1.2. Processing Constraint-based Logic

Various formal mathematical notations apply to capturing business rules in formal logic as constraints. The Z notation is an early relational language for specifying data constraints using set notations [10]. The Alloy system builds upon Z with automatic analysis and design-time data model consistency checking [11].

Z and Alloy help with theoretical analysis but do not generate practical information systems from the rule specifications. In earlier work, we apply concepts similar to those in Z and Alloy to build Ampersand. Ampersand processes business rules in relation algebra to help users in checking their validity [3]. Ampersand does so by processing the business rules into functional specifications, data models and a prototype interface for data entry. Ampersand-generated interfaces perform run-time consistency checking of data as the user enters it. Rule designers can then examine these specification, models and interfaces to see if they match their expectations for how the rules should function in practice.

Ampersand uses the language "A Description Language" (ADL), a formal language that is based on relation algebra. The key components for capturing business rules through the Ampersand method are concepts, relations and rules. In addition, Ampersand generates a prototype interface for an information system for data entry that enforces these rules. Examining the generated documentation and interface helps rule designers validate their rules. Oene [12] and Sangers [13] have demonstrated suitability for developing information systems. Sangers elaborates on possible differences between concepts and ontologies and between ontologies themselves [13].

1.3. The Semantic Web

The current World Wide Web is a huge collection of linked documents that machines process for presentation to humans. As an open system, anyone can contribute to the web at any time. There is no guarantee of quality of information or the persistence of a document. The Semantic Web is an expansion of current World Wide Web that allows machines to share and process information. This creates an environment where software agents can be specific about how they process information. This arrangement makes it possible to link data, allowing interoperability between systems.

Computer and information science use the term ontology to mean the conceptual model of the knowledge within a particular domain. On the Semantic Web, ontologies provide a common structure of data to communicate [14]. Ontologies provide a vocabulary of a particular domain. This vocabulary should help to avoid the confusion of informal natural language, which often exists between the various stakeholders in the development of an IT system. When preparing an ontology, it is important that it is logically consistent and can be used by all stakeholders.

The Semantic Web relies heavily on the use of ontologies, which provide the underlying data structure, so that machines can interpret it [14]. The Semantic Web consists of standards (W3C recommendations) such as the RDF, SPARQL, RDFS, OWL en SWRL. The Resource Description Framework (*RDF*) is a standard model for data interchange on the Web [15]. It describes the characteristics of resources as triples in subject-predicate-object form. A subject is the resource that a triple describes. The predicate is the property or attribute of the resource described. The object is the value of that attribute. So-called triple stores provide RDF data capture. The SPARQL Query Language RDF provides such standardized access to RDF triples stores. RDF and SPARQL provide the foundation for placing and querying data on the Semantic Web.

RDF Schema (*RDFS*), an extension of RDF, is a vocabulary for describing properties and classes in RDF, including hierarchical relationships of these classes and properties [14]. The Web Ontology Language (*OWL*) extends RDFS by setting connections between classes, cardinality, data typing for properties and additional characteristics of properties [16]. Together, RDFS and OWL provide the equivalent of data modeling in the Semantic Web context. Specifically, RDFS and OWL define inferencing of new data based on how existing data fits into the data model, or ontological, structure. The Semantic Web community typically uses them to define ontologies.

There are several initiatives for providing business rules and logic in a Semantic Web environment that go beyond RDFS- and OWL-defined ontologies. Eiter et al have investigated (business) rules and ontologies within the Semantic Web [17]. They concluded that currently there is no initiative to set a good standard and that the integration with OWL is not yet good enough.

When the Semantic Web began to develop its rules, these rules had less structure than in existing ontologies [18]. The Rule Markup Initiative has made an effort to close this gap with the Rule Markup Language (*RuleML*). RuleML is a canonical language for rules using XML. The Semantic Web Rule Language (*SWRL*) is an initiative based on OWL and RuleML [19]. SWRL adds Horn clause expressions [20] in terms of OWL descriptive logic capable of implementing a variety of business rules [14]. While each component of RDFS and OWL cause a specific pattern of inferencing, SWRL enables defining new inference patterns. SWRL has thus much more expressive power than RDFS and OWL, but with the added danger of triggering inferences that systems cannot process.

There is not much literature on the subject of comparing the Semantic Web with other means for business rule implementations. Spreeuwenberg and Gerrits offer a comparison between SBVR and the Semantic Web [21]. It provides a high-level comparison, concluding that the two share roots and goals and a similar form but differ in their target audiences and expressive power.

2. Feasibility

This section starts our presentation of the analysis and findings for the VOG case study implementations. The aim of the study is to determine the *feasibility* of implementing the VOG case study with technologies from relation algebra and from the Semantic Web. Later sections *compare* these two forms of implementation.

A Certificate of Good Conduct, or VOG, is a statement issued by the Ministry of Justice about whether criminal records exists for the person that preclude them from a given type of employment. When someone applies for a position that handles confidential information or vulnerable people, the employer can ask for a VOG from the applicant. In addition, some positions, such as teachers and taxi drivers, legally require a VOG for anyone holding them [1]. We present here an implementation of an interface for the execution of VOG applications that applies relation algebra via Ampersand, including an online interface[2] and the source files that generate it[3]. We developed this implementation upon request of the Dutch Ministry of Justice and Security.

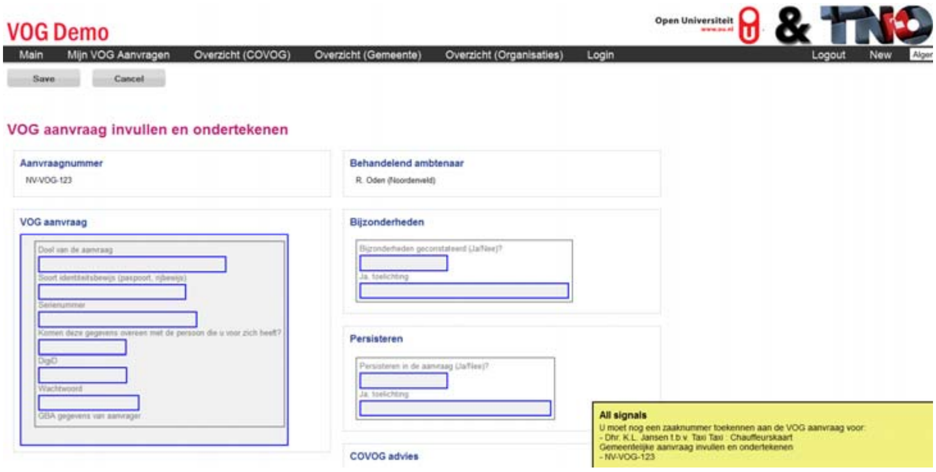

Figure 1. Display from a user interface generated by Ampersand from relation algebra rules for the VOG (in Dutch)[2]. It states that the current VOG request form still requires entry of a business number.

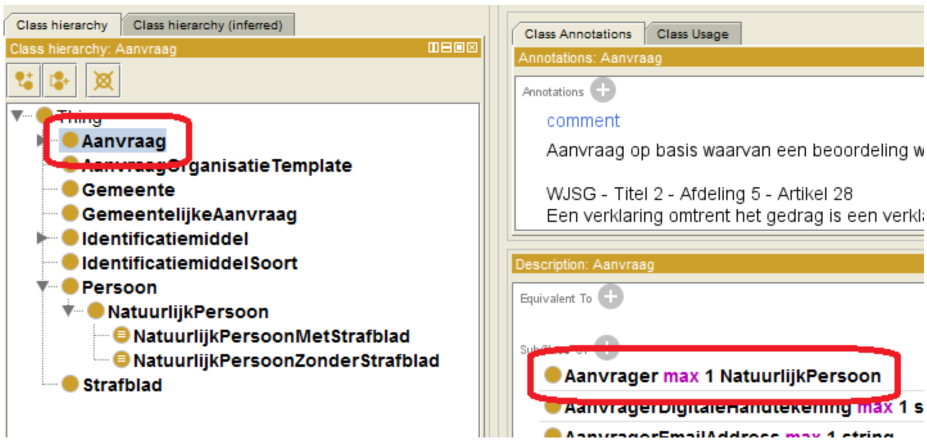

Figure 2. Partial display from the Protégé-OWL implementation of the VOG (in Dutch). It shows the restriction that each application can have a maximum of 1 applicant, who must be a person

---

2 http://sentinel.tarski.nl/ampersand/VOGDemo/ (developed on Firefox)
3 http://is.cs.ou.nl/OWF/index.php5/Bos_2013_resources

Figure 1 shows a display from the interface generated by *Ampersand* for the VOG case study. In this example, a constraint set by the underlying relation algebra rules is broken. Here, the user still must enter a business number for the application. Otherwise, it is cannot entry of that VOG application into the system.

Figure 2 shows an interface display from our implementation with the *Semantic Web* tool Protégé-OWL [22] of the VOG. Both the Semantic Web and Ampersand implementations work for the VOG casus, showing that implementing the VOG casus is feasible with both technologies. The following sections discusses how these implementations differs, and why.

3. Expressive Power

The biggest differences between the Semantic Web and relation algebra approaches to the VOG case study regard expressive power. Ampersand primarily uses *constraints*, which state that something was or not allowed to have occurred, making it retroactive. On the other hand, the primary logical concept on the Semantic Web is *inferencing*, which is proactive by adding new data to make existing data consistent with the rules.

One aspect of the VOG that illustrates this difference regards *unique identification*, such as with the BSN (*burgerservicenummer*, or "citizen service number", the Dutch government's unique identifier for each of its citizens). If two URIs on the Semantic Web share a BSN and if Semantic Web ontological constructs proclaim BSN to be uniquely identifying, the Semantic Web implementation will not consider this a conflict. Instead, the system will regard both URIs as referring to the same resource and infer that all annotations involving either URI apply to that unified resource. Ampersand, on the other hand, takes the typical closed world approach here, by refusing to assign a BSN to a person if another person already has that BSN. Both approaches ensure the same result: each uniquely identified concept occurs just once.

While the Semantic Web has less focus on constraints than Ampersand and typical data systems, it still allows some. Some tools like Protégé-OWL announce *inconsistencies* in Semantic Web data [22]. One example of this in the Semantic Web VOG implementation regards rules about valid applicants. If Protégé-OWL detects no inconsistencies involving the applicant then the user can conclude that he or she is valid. The Ampersand implementation does more or less the same for checking applicant validity. However, although both technologies can detect inconsistencies, the constraint-based nature of Ampersand and the open, inferring nature of the Semantic Web enable inconsistency detection more often in Ampersand.

The VOG requires that an applicant reside in the municipality where the application takes place. Given Ampersand's *Closed World Assumption*, its VOG implementation assumes that residential records are up-to-date and have real-time access. It can thus simply query during the application if the applicant currently lives there. The Semantic Web's *Open World Assumption* makes its VOG implementation of this rule more complex. Our Protégé-OWL implementation must thus query for all information about the history of that person's registration as a resident in the municipality, comparing the dates for each time that person moves in and out to make sure there is no moving out date after that person's last moving in date.

The lack of *negation as failure* in the core Semantic Web ontological standards provided a challenge for its VOG implementation. A VOG certifies that the applicant has no criminal record. However, concluding something from the lack of a given pattern of data requires negation as failure. Semantic Web ontologies cannot infer based on the absence of something because the Open World Assumption means that something may nonetheless be out there somewhere.

We implemented this on Protégé-OWL with a property on VOG applications stating the administrator has confirmed the lack of a criminal record. The administrator can use Protégé-OWL to confirm this with a *SPARQL* query. Unlike the ontological standards, SPARQL can acknowledge the absence of data patterns because SPARQL results are tables of variable bindings instead of inferences feed back into the data store. With this twist, we were able to define an interface that uses Semantic Web technologies to confirm the lack of criminal records.

4. Rule types

Here we analyze the ability of the two formalisms to implement for the VOG rules of particular types given by Loucopoulos and Kardasisa [23]. Their work provides a categorization of business rules, which we apply here for analysis of the categories in legal applications using constraint-based and Semantic Web technologies. Table 1 has the rule types that the VOG implementations require.

Table 1. Types of rules in the VOG case study

Rule type	Description	Example in VOG
Information object rules	Information about general concepts	Each request requires a unique request number
Information assertion rules	Make conclusions based on events and circumstances.	Each request requires payment of a fee
Actor rules	Information on those who carry out actions	No application is complete without confirmed identification of the applicant

Several instances of *information object rules* that we extract from the VOG legal text require unique identifiers for particular types of objects, such as unique request numbers for each VOG application. The Ampersand-generated system maintains this constraint during user interaction: when the user adds an application, Ampersand blocks attempts to enter a previously used application number via constraints.

While Ampersand proactively prevents inconsistent data, the typical Semantic Web approach in such situations is to infer new data retroactively that maintains restrictions. For example, our implementation of the VOG on Protégé-OWL takes any two applications with the same request number and *infers* that both must refer to the same application, and that data about either original object applies to the newly unified application.

Actor rules are similar to information object rules and relate to the integrity and consistency in relation to the actors. The actors in this case study are the applicant and the attending officer. Logical specification of the VOG in both Ampersand and Semantic Web make no logical distinction between information object and actors. The

Ampersand implementations encoding for these rules is similar to its encoding for information object rules. The Semantic Web implementation realizes these rules with SWRL inferencing. An example is the rule requiring confirmation of the applicant's identity. The Protégé-OWL implementation infers the application from the data regarding the means by which he or she identified herself. If such means of identification does not exist for an application then Protégé-OWL will detect an inconsistency. Protégé-OWL detects this inconsistency retroactively when processing all data rather than proactively preventing it as the Ampersand interface does. The Semantic Web implementation's SWRL rules infer and assert the validity of valid applications.

On the other hand, Ampersand's relation algebra-based approach handles such actor rules by constraining the input of new information rather than automatically inferring new data. For example, in the rule "An application may only be made in the municipality in question", The Ampersand implementation derives no new information. Instead, it only indicates when new data breaks the rule.

Ampersand's relation algebra syntax lacks some practical functions that Semantic Web formats have, such as the lack of math, comparisons and numeric measurements of space and time. Comparisons are possible in SWRL, where standard calculation rules are included. In addition, rules in the Semantic Web are unlike those in relation algebra. Whereas the SWRL primarily works with Horn clauses (inferences), relation algebra (in this case Ampersand) works with constraints. This means that Ampersand reports things that may not occur given current data, while Horn clauses infer new data that derives from existing data.

5. Tools

Although we initially compare the syntax of the standards and rules, we also discuss here the ability to use tools implementing these standards. We compare three aspects of each tool: *editing*, *documentation support* and *interface generation*.

Protégé-OWL offers good ontology *editing* capabilities with features such as its display of inference paths. Ampersand, however, lacks such editing support. Development with Ampersand often requires trial and error, by writing in a text editor and then trying the Ampersand generation tool with it.

Given an edited ontology, the two tools offer different support for providing *documentation* of the ontology. Ampersand automatically generates text and diagrams document based on a set of given relation algebra rules, which is useful in helping people confirm the validity of the rules for their given purpose. Oene has shown this documentation to be useful [12]. Protégé-OWL generates no equivalent textual documentation. However, its OntoGraf tab generates data model charts similar to those that Ampersand generates. OntoGraf goes further than Ampersand by providing a GUI interface for additional editing of the graphs.

Ampersand provides better support for *interface generation* than Protégé-OWL and other Semantic Web tools do. Ampersand automatically generates a data entry interface from the underlying rules. This interface guides and limits the user to assure the data entered meets all the rules. Ampersand can do this because relation algebra focusses on constraints. The Semantic Web's emphasis on inference and the Open World Assumption makes it less applicable to constraining aspect of user interface.

In addition, the *target audience* for each implementation is different. Ampersand focuses on communication in system development [3]. In the Semantic Web, the emphasis is much more on the ontology itself, which acts as a foundation for input data to a wide variety of potential systems.

6. Conclusion and Future Work

This work compared the implementation of a legal policy case study with two categories of logic: the Ampersand tool for relation algebra, and the Semantic Web. Both technologies could implement the case study. Semantic Web interfaces tend to be more permissive and adaptive with user data input, with thereby increased risk of problematic data entry. Both technologies enable detection of inconsistencies, although the open nature of the Semantic Web can make them harder to exploit directly. Unlike Ampersand, the Semantic Web cannot implement logical conclusions involving negation. Semantic Web interfaces can work around this by querying the information for the user and letting him or her assert the conclusion explicitly.

Our next planned step for this case study is to map it to the Legal Knowledge Interchange Format (LKIF) [24]. LKIF is a Semantic Web-defined ontology for legal knowledge. Implementation of the VOG in LKIF would provide additional verification of LKIF's applicability. It would also enable comparison of LKIF with Ampersand's constraint-based VOG implementation.

7. Acknowledgements

TNO and the Open Universiteit made the Ampersand implementation of the VOG case study presented here. Its development was upon request of the Dutch Ministry of Justice and Security. The Protégé-OWL implementation and its comparison with the Ampersand implementation come from the Master's Thesis of Pim Bos. Funding for this work comes in part from the Informatics faculty of the Open Universiteit [25].

References

[1] Ministry of Justice of the Netherlands, "Wet justitiële en strafvorderlijke gegevens – Titel 2 – artikel 35," 2012.

[2] Ministry of Justice of the Netherlands, "Beleidsregels VOG-NP-RP 2008," 2008.

[3] G. Michels, J. C. van der Woude and S. M. M. Joosten, "Ampersand: Applying relation algebra in practice," in *12th International Conference of Relational and Algebraic Methods in Computer Science (RAMICS 2011),* Rotterdam, 2011.

[4] T. Berners-Lee, J. Hendler and O. Lassila, *The Semantic Web,* Scientific American, 2001.

[5] R. G. Ross, *Business Rules Manifest,* Business Rules Group, 2003.

[6] A. Coenen, L. Hermans, M. v. Roosmalen and S. Spreeuwenberg, Business Rule

Management, Den Haag: Sdu Uitgevers bv., 2008.

[7] D. Baisley, J. Hall and D. Chapin, *Semantic Formulations in SBVR*, Unisys Corporation, 2005.

[8] J. Rumbaugh, I. Jacobson and G. Booch, "The Unified Modeling Language Reference Manual," 1999.

[9] Object Management Group, "OMG Object Constraint Language (OCL), version 2.3.1," 2012.

[10] J. Spivey, "The Z Notation: A reference manual," Prentice Hall, 1992.

[11] D. Jackson, Software Abstractions: Logic, Language, and Analysis., Cambridge: The MIT Press, 2006.

[12] L. v. Oene, "Zaakgericht werken - Een architectuurbeschrijving met behulp van bedrijfsregels in ADL," Open Universiteit Nederland, 2009.

[13] H. W. Sangers, "Achieving Traceable Compliance using the Ampersand Method," Open University of the Netherlands, 2008.

[14] K. Giri, "Role of Ontology in Semantic Web," *DESIDOC J. Lib. Inf. Technol.,* pp. 116-120, 2011.

[15] F. Manola and E. Miller, "RDF Primer," World Wide Web Consortium, 2012.

[16] I. Herman, "Web Ontology Language," 2007. [Online]. Available: http://www.w3.org/2004/OWL/. [Accessed 14 maart 2012].

[17] T. Eiter, G. Ianni, T. Krennwallner and A. Polleres, "Rules and Ontologies for the Semantic Web," Technische Universitat Wien, Wien, 2008.

[18] H. Boley, S. Tabet and G. Wagner, "Design Rationale of RuleML - A Markup Language for Semantic Web Rules," DFKI GmbH, Eindhoven University of Technology, 2001.

[19] B. Parsia, E. Sirin, B. Cuenca Grau, E. Ruckhaus and D. Hewlett, *Cautiously Approaching SWRL*, Elsevier Science, 2005.

[20] M. O'Connor, H. Knublauch, S. Tu, B. Grosof, M. Dean, W. Grosso and M. Musen, "Supporting Rule System Interoperability on the Semantic Web with SWRL," Stanford University, Stanford, 2005.

[21] S. Spreeuwenberg and R. Gerrits, "Business Rules in the Semantic Web, are there any or are they different?," in *Reasoning web 2006*, Berlin, 2006.

[22] H. Knublauch, R. W. Fergerson, N. F. Noy and M. A. Musen, "The Protégé-OWL plugin: An open development environment for semantic web applications," in *The 3rd International Semantic Web Conference (ISWC2004)*, 2004.

[23] P. Loucopoulos and P. Kardasisa, "Expressing and organising business rules," *Information and Software Technology*, no. 46, p. 701–718, 2 Maart 2004.

[24] R. Hoekstra, J. Breuker, M. Di Bello and A. Boer, "The LKIF Core Ontology of Basic Legal Concepts," *The Legal Ontologies and Artificial Intelligence Techniques Workshop (LOAIT 07)*, pp. 43-63, 4 June 2007.

[25] P. Bos, „Bedrijfsregels in verschillende vormen - Een vergelijking op toepasbaarheid tussen SWRL en Relatie algebra bij wetteksten," Master's Thesis Open Universiteit, Heerlen, Netherlands, 2013.