Toward G-OWL: A graphical, polymorphic and typed syntax for building formal OWL2 ontologies

Michel Héon
Université du Québec à Montréal
Montréal (Québec)
Canada
heon@cotechnoe.com

Roger Nkambou
Université du Québec à Montréal
Montréal (Québec)
Canada
nkambou.roger@uqam.ca

Christian Langheit
Institut de recherche du Québec
Hydro-Québec Varenne (Québec)
Canada
langheit.christian@ireq.ca

ABSTRACT
The *Web Ontology Language* (OWL-2) aims at offering a family of syntax such as RDF/XML, Manchester Turtle and others, for building ontologies. Ontology engineering is a complex task that requires skills that are rarely accessible to content experts. On the other hand, to model contents pertaining to a specific domain, graphical modeling is a technique that is often used to offer a knowledge representation tool to content experts that are not well acquainted with the process of formal ontology design. In this paper, we present the way in which the usage of polymorphism and symbol typing of graphical vocabulary have allowed us to design the G-OWL syntax, a graphical syntax that aims to graphically represent domain-specific knowledge using the OWL-2.

Keywords
OWL-2, ontology, visual modeling, graphical ontological syntax, graphical syntax, graphical ontology.

1. INTRODUCTION
Graphical syntax are sometimes used during the system design stage, to promote brainstorming and knowledge transfer in organizations [1]. The construction of an OWL2 ontology is not a straightforward activity for a content expert who is not familiar which the OWL. Conversely graphical modeling (such as: *Mind Mapping* or *Concept Mapping*) is a solution that is often considered to allow content experts to graphically represent informal or formal knowledge with simple representational guidance [2]. Several works have allowed the development of graphical syntax to build an ontology. Some based on the specialization of existing language like UML [3], and others on the complete overhaul of the syntax [4-5]. To remain formal, each of these syntax use a "one to one matching" approach where single graphical symbol (either an entity or a relation) is matched to a specific OWL2 symbol (entity for subject and object, and relation for predicate). The originality of the design of G-OWL (acronym for *Graphical OWL*) lies in the assumptions that the availability of a limited number of symbols facilitates the modeling activity. G-OWL syntax implements this principle by using the polymorphism and typology properties of symbols. y offering such mechanisms for reducing the number of symbols without losing the expressiveness, G-OWL differs from existing graphical ontology design tools [3-4-5]. This advantage allows models to highlight the representations associated with the domain-knowledge rather than focus on the syntactic management of the representation expressiveness of OWL.

2. HYPOTHESIS
For the design of G-OWL, our main hypothesis is that, it is possible to reduce the number of symbols required to design a G-OWL model while retaining the expressiveness of the OWL2. To do this, it is necessary to impose a polysemy on G-OWL symbols. Polysemy is the association of a finite number of meaning to a sign. Two techniques allowed us to increase G-OWL's polysemy:

1) **Polymorphism** permits the attribution of several meanings to a single graphical symbol. The symbol can later be disambiguated via its topological usage context (Ex.: see SLink in table 2).
2) **Typology** assignment limits the number of symbols by assigning them a type. This technique attributes a type to the entities and relations of the syntax. This technique distinguished graphic elements belonging to the syntax and elements related to the knowledge domain.

WWW 2016 Companion, April 11-15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org. 2048/10.1145/2872518.288
9377

3. G-OWL SPECIFICATIONS
G-OWL's specification defines the graphical vocabulary of the G-OWL syntax (see [6] for the full specification). It identifies the typed entities and the typed relations that form the vocabulary's alphabet. As presented in table 1 the entities are grouped together at two levels of abstraction – the *abstract level* and the *factual level*. The abstract level refers to the idea of the Terminological ox (T- ox) and Role ox (R- ox) in Description Logic (DL). As for the factual level, it refers to the Assertion ox (A- ox) idea in DL. In table 2, the relations are grouped in five types where most of them are semantically overloaded. As an example, the SLink can be used to specify the hierarchy expression between two Classes or Two Properties. The disambiguation of the SLink meaning (`subClassOf` or `subPropertyOf`) is made by the application of the *disambiguation rules*.

4. TESTING G-OWL'S EXPRESSIVENESS
G-OWL's expressiveness (to make sure it is comparable to OWL2) has been evaluate in two ways. At first, as it has been demonstrating in the chap 6 to 9 of [6], for each expressiveness element cites in the official W3C OWL2 Primer [7] a G-OWL representation can be associated (Table 1 - 2). Doing this ensures that there is at least a G-OWL representation for every semantic element of the OWL2. Secondly, to highlight G-OWL's representational quality, we have chosen to compare the representation of the fragment of the *wine.owl* ontology with its graphical representation, and its representation in G-OWL (see Fig. 1). As we can see, G-OWL uses less entities and relations than the other representations.

5. CONCLUSION
We have briefly presented G-OWL, a graphical, polymorphic and typed knowledge-representation syntax which offers an increased usability compared to a textual syntax such as OWL2. G-OWL has been completely specified and implemented. We are currently working on its validation in order to ensure its consistency, completeness and usability. The consistency and completeness of the syntax can be formally validated through an editing software we are developing for G-OWL. Its usability will be validated by

Table 1 (Partial): Graphical vocabulary of G-OWL entities

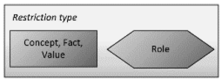
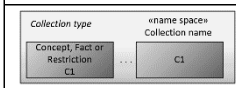
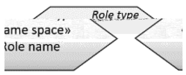
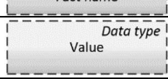
	Graphical Alphabet	Meaning	Typed disambiguation	Polysemy in OWL
Abstract level		The rectangle depicts the « what » of things		`owl:Class`
		The containing rectangle depicts a universal or existential *Restriction* or its value or cardinality.	∃	`owl:someValuesFrom`
			∀	`owl:allValuesFrom`
			∍, ≤, ≥, =	*`plus others …`*
		The containing rectangle is also used to represent a collection of declarative knowledge	∩	`owl:intersectionOf`
			∪, [ ], ¬, ≠	`owl:unionOf` *plus others…*
		The hexagon is used for representing a role that defines the property between abstract or factual entities.	*`if codomain is a data`*	`owl:DatatypeProperty`
			`if codomain is a fact`	`owl:ObjectProperty`
			`T, S(`*`symmetric`*`),` `F(`*`func.`*`),` *plus others …*	`owl:TransitivProperty` *plus others …*
Factual level		The dotted-line rectangle depicts a fact.		*OWL individual*
		The dotted-line depicts data of the type *integer, real, s,* etc.	*`Bool, String, Int, Float`*	`xsd:Boolean` *plus others …*

Table 2 (Partial): Graphical vocabulary of G-OWL relations

Type	Meaning	disambiguation rule	Polysemy in OWL
`--- S -->` SLink	The *specialization link* associates two knowledge items of the same type of which the first is a specialization of the second.	*`if SLink between two concepts`*	`rdfs:subClassOf`
		`if SLink between two roles`	`rdfs:subPropertyOf`
`<-- S -->` LinkDS (LinkS with double orientation)	The *synonymy link* associates two knowledge items of the same type at the abstract level or two facts. It indicates that the first knowledge item is the equivalent (or synonym) of the second.	*`if DSLink between two concepts`*	`owl:equivalentClass`
		`if DSLink between two roles`	`owl:equivalentProperty`
		`if DSLink between two facts`	`owl:sameAs`
`-- A -->` ALink	The *attribution link* associates an attribute to a concept, a restriction or a collection to specify the image or domain of a property.	*`if source is concept and destination is role`*	`rdfs:domain`
		`if source is role and destination is concept`	`rdfs:range`
`--`*`name`*`->` Non Typed Link	The *Non Typed link* associates a *predicate* between a fact and a knowledge item. The name of the predicate is associated to an existing attribute via the *RoleName.*	*`if source is a fact and destination is a fact`*	*Predicate*
`-- I -->` ILink	The *Instantiation link* associates a concept with a fact which designates an instance of this knowledge item.	*`if source is a fact and destination is a concept`*	`rdf:type`

using it as a supporting tool for knowledge elicitation and brainstorming, as well as for building more domain ontologies.

6. REFERENCE

[1] G. Paquette, M. Léonard, J. asque, and . Pudelko, "Modeling for Knowledge Management in Organizations," in *Visual Knowledge Modeling for Semantic Web Technologies: Models and Ontologies*, ed: IGI Global, 2010, pp. 393-413.

[2] D. D. Suthers, "Representational guidance for collaborative inquiry," in *Arguing to Learn* J. Andriessen, M. aker, and D. Suthers, Eds., ed Dordrecht/ oston/London: Kluwer, 2003, pp. 27-46.

[3] D. Gašević, D. Djurić, and V. Devedžić, *Model Driven Architecture and Ontology Development*. New York, Inc.: Springer-Verlag, 2006.

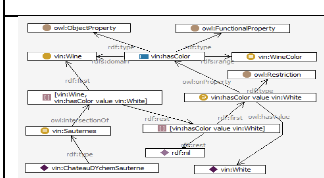
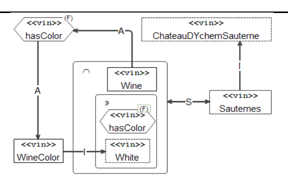
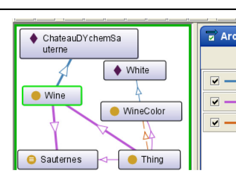
[4] R. Falco, A. Gangemi, S. Peroni, D. Shotton, and F. Vitali, "Modelling OWL Ontologies with Graffoo," in *The Semantic Web: ESWC 2014 Satellite Events*. vol. 8798, V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, and A. Tordai, Eds., ed: Springer International Publishing, 2014, pp. 320-325.

[5] S. Krivov, R. Williams, and F. Villa, "GrOWL: A tool for visualization and editing of OWL ontologies," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 5, pp. 54-57, 2007.

[6] M. Héon, *Web sémantique et modélisation ontologique (avec G-OWL): Guide du développeur Java sous Eclipse*, Editions ENI, 2014.

[7] P. Hitzler, *et al.* (2012, 23-10-2013). *OWL 2 Web Ontology Language Primer (Second Edition)*. Available: http://www.w3.org/TR/owl2-primer/

Fig. 1. Partial ontology of the Château d'Yquem Sauterne in graphical representation and G-OWL (extracted from *wine.owl*).

Graph representation	G-OWL Representation	Protégé OntoGraph	Criteria Ent/link	Graph	G-OWL	OntoGraph
			Number of types	6/8	5/3	6/3
			Total number	13/13	9/5	6/7