Knowledge Graph Considered Harmful for Ontology

Seiji Koide[1]

Ontolonomy, LLC., Minami-ku Yokohama 232-0066, Japan,
`koide@ontolonomy.co.jp`,
WWW home page: `http://ontolonomy.co.jp/`

Abstract. The time of knowledge graph has come. Linked Data is now rephrased to knowledge graph, and no one has doubt on the future of it. However, will the time of ontology come in the next step? There exists a long history of ontology ever since the ancient Greece, and now the terminology has become popular with the advent of OWL. Nevertheless, the ontology does not seem to happen so as Linked Data did. There is a serious gap on the semantic representation between the ontology and the knowledge graph, and the gap originates *semantic networks*. In this paper, we pursue the history of knowledge representation, point out the serious semantic gap contained in knowledge graph. We propose an alternative representation language for ontological knowledge in Semantic Webs.

Keywords: New KM, RDF, OWL, knowledge graph, knowledge representation, Frame-based, Case-based

1 Introduction

The purpose of this essay is to cause a stir in the community of Semantics Webs just as Dijkstra did in software engineering.[1] It seems that the time of Linked Open Data (LOD) has come. Ones have become to rephrase "Linked Data" to "knowledge graph", since knowledge graph has been used as technical terminology in the domain of *semantic search* by Google, IBM, etc.. Because the both is roughly the same technology from the technical viewpoint. On the other hand, the term of "ontology" exists ever since the age of ancient Greece, and the engineering of ontologies has been pursued since 1970s as a part of computer science and Artificial Intelligence, then it has become popular today with OWL. However, we have yet a serious gap between the representation of knowledge graph and ontology. More precisely, we have theoretical, practical, and socio-technological difficulties such as how to understand the *subsumption* concept described in RDF Semantics [2] with respect to the ambiguous *IS-A* concept [3, 4], how to represent *structural link* [3, 5] in the form of knowledge graph, and how to discriminate the notion of *class* and *individual* in human mind.

---

[1] See, `https://en.wikipedia.org/wiki/Considered_harmful`

The following is a fallible example which novices easily fall into erros.[2]

```
Wine a owl:Class ;
  rdfs:subClassOf food:PotableLiquid ;
  *madeFromGrape WineGrape .
```

Instead, it should be exactly coded in turtle as follows.

```
Wine a owl:Class ;
  rdfs:subClassOf food:PotableLiquid ,
                  [ a owl:Restriction ;
                    owl:onProperty madeFromGrape ;
                    owl:allValuesFrom WineGrape ] .
```

The cause of this mistake is three fold; firstly, the abstract syntax of RDF (namely RDF graph) does not fit to the description of ontologies; secondly, it it hard to understand the concept of subsumption and property inheritance; thirdly, most people are sluggish to study hard, some one tends to take an easier way even if it is impossible to reach the final goal.

Contrary to the error sentence mentioned above, the following code is completely correct with respect to individuals.

```
ElyseZinfundel a Zinfundel ;
  hasMaker Elyse ;
  hasSugar Dry.
```

In this case, `Zinfundel` is a class, but `ElyseZinfundel`, `Elyse`, and `Dry` are individuals. There is no inheritance on properties. Actually, at the error case shown above, it should be coded with the meanings that every wine (as instance) is made from some wine-grape (as instance), and it must be described at the class level on wine. Otherwise, it is forced to describe the property values on every instances. It should be here noticed that `Wine` and `WineGrape` are classes.

Such a kind of problems was involved in the beginning of *semantic networks*. Woods [3] pointed out the semantical ambiguity of network links and introduced the distinction of *assertional link* and *structural link*. At the final step of KL-ONE family, Clark developed a knowledge representation language, KM (Knowledge Machine) [9, 10]. According to the style in KM-like representation, the error sentence mentioned above may be paraphrased into the followings.

```
Wine is a owl:Class and rdfs:subClassOf food:PotableLiquid .
every Wine is madeFromGrape a WineGrape .
```

Note that these sentences are not English, rather an artificial knowledge representation language. It is a sort of syntax sugar of the turtle syntax and preserves

---

[2] This example is taken from Wine Ontology [6], and where an asterisk is attached to the head of wrong lines.

RDF and OWL semantics. We can translate KM-like sentences into turtle sentences, just as turtle sentences can be translated into RDF/XML format without the loss of information.

It is conceived that Description Logics inherited the heritage of *semantic networks* and *frame systems* [4]. In fact, many features of KL-ONE family came into OWL. In this paper, we pursue the history of knowledge representation, specifically focusing on the development of class notion and subsumption concept, then point out the problem of knowledge graph representation for OWL. Finally, we propose a new representation language, the New KM, a successor of KM by Clark and SWCLOS (Semantic Web processor on top of Common Lisp Object System) [11, 12].

The rest of this paper is organized as follows. In section 2, we present a brief history of the knowledge representation on *semantic networks* and *frame systems*, focusing on class-instance notions and the inheritance concept. section 3 presents an idea of New KM, which is a unified successor of KM by Clark and our own Semantic Web Processor, SWCLOS. Finally, we conclude some remarks and the future work.

2 Historical Views on Knowledge Representation

2.1 Semantic Networks to Description Logic and OWL

The study of *semantic networks* was started as an associative link network at 1966 by Quillian's work [7, 8], for the purpose of making artificial memory for words and meanings. At the time, such an associative link structure was taken as a firm base in modeling human memory. However, many problems that were involved in such simple networks were soon exposed by many researchers.[3] Human epistemology is more complex than the association.

Woods [3] analyzed the semantics of *semantic networks* and introduced the distinction between *structural links*, which present propositional statements on things, and *assertional links* on assertional relation among things. He addressed the following network structure.

```
N12368
    SUPERC   TELEPHONE
    MOD      BLACK
```

The meanings of this sentence may be interpreted in two ways. In OWL, it would be written distinctively as follows, into a proposition in TBox (left side) or an assertion in ABox (right side).

```
N12368                                  N12368 a TELEPHONE ;
  owl:intersectionOf                      hasColor BLACK .
    ( TELEPHONE
```

---

[3] The original by Quillian was not so simple as the successors. The original had devices of type, token, plane, and notions of class, subclass, and modification. [13]

```
[ a owl:Restriction ;
  owl:onProperty hasColor ;
  owl:hasValue BLACK ] ) .
```

The former insists the class of black-colored telephone, but the latter asserts the existence of a telephone whose color is accidentally black. On the idea that the notation must be specified more precisely, he started the discussion on the problem involved in *semantic networks*. However, the issues, in the viewpoints at the present, were confused and spread out widely in distinct levels, from semantics to pragmatics, from denotational levels to logical levels.

Up to the mid of 1970s, semantic networks constitute the primer knowledge representation and many attempts revealed that they never lived up to researcher's expectations. Brackman [13] demonstrated complex semantic networks that contain both conceptual networks and their particulars, and showed special existences a set of links that allows the specification of a concept as a set of attribute definitions in conjunction with a structural interrelation between those attributes, see details in [13].

At 1979, Brackman [14] presented a comprehensive survey on semantic networks. He investigated the work by Quillian, Woods, Brackman, Collins, Carbonell, Winston, expanding to Fillmore, Simmons, Hendrix, Rumelhart, Schank, Heidom, Anderson, Shapiro, Cercone, Phillip Hayes, Norman, and Szolovits, then clarified five levels of characteristics on semantic networks, "implementational," "logical," "epistemological," "conceptual," and "linguistic" levels. Due to limitations of space, we summarize the result at Table 1. Note that we can capture that

Table 1. Characteristics Levels of Semantic Networks

Level	Primitives	description
Implementational	atoms, pointers	A Link is a pointer and a node is a destination.
Logical	propositions, predicates, logical operators	Logical primitives with a structured index like AND, SUBSET, EXISTS
Epistemological	concept types, subpieces, inheritance, structuring relations	Formal structure of conceptual units and interrelationships as them. Independent of any knowledge edge
Conceptual	semantic relations, primitive objects and actions	Language-independent conceptual primitives and case structure.
Linguistic	arbitrary concepts, words, expressions	Networks whose primitives are language-specific.

RDFS and OWL fall into *epistemological level*, which was discovered by Brackman as a missing level in his comprehensive investigation. Usual ontologies that are built using RDFS and OWL are at *conceptual level*. As an instantiation of such an epistemological level, he advocated more elaborated "Structured Inheritance Networks", in which *Role/Filler Description* and *Structural Description* are derived.

Eventually, Brackman [15] published KL-ONE at 1985, and after that many KL-ONE-*ish* systems succeeded. They are, as a whole, called KL-ONE family. All systems of KL-ONE family, except KM [9, 10] is listed at [16].

The class-instance notion gradually emerged in the development of KL-ONE family. In the original KL-ONE, the notion of class was the production of a classifier, and did not provide any explicit primitives for the class-instance indication. This situation is carried over Description Logics and OWL. In CLASSIC [17], the operators for individuals were identical to that for classes, but the function `cl-create-ind` was provided to create an individual under a CLASSIC-description. Moreover, LOOM [18] prepared 17 operators (functions/macros/slots) for instances. Ideas of objects and mixin classes were borrowed from CLOS (Common Lisp Object System) and provided the *mixin-inheritance* functionality.

KM [9, 10] is, which is not regarded so, the last system in the line of KL-ONE family in thought. It has a frame-like syntax as well as CLASSIC and LOOM. However, it presented an easier way to encode the inheritance attributes for instances at classes. The following is actual programming code in KM of Buy event.

```
(every Buy has
  (buyer ((a Agent)))
  (object ((a Thing)))
  (seller ((a Agent)))
  (money ((the cost of (the object of Self))))
  (subevent1 ((a Give with
                (agent ((the buyer of Self)))
                (object ((the money of Self)))
                (rcpt ((the seller of Self))))))
  (subevent2 ((a Give with
                (agent ((the seller of Self)))
                (object ((the object of Self)))
                (rcpt ((the buyer of Self)))))))
```

This code axiomatize that two agents appear as buyer and seller, every event of Buy is accompanied by two distinctive Give events in which one event for a buyer the money is received by the buyer, in the other event for a seller the object is received by seller. In the above example, the event money is instantiated as the cost of the event. KM denotes two fundamental types, instances and classes. A class has the extension of the individuals, and properties of individuals of a class are expressed of the form:

```
(every <class> has
  (<slots1> (<expr11> <expr12> ... ))
  (<slots2> (<expr21> <expr22> ... ))
  ...
```

```
    )
```

Thus, "`(every ...`" form describes properties for individuals of a class to be inherited in accordance with superclass-subclass relation. On the other hand, the form without "`every`" but with "`superclasses`" attribute for classes denotes superclasses of a subjective class with other properties for the class *per se*.

```
  (<class> has
    (superclasses (<superclass1> <superclas2> ... ))
    (<slots1> (<expr11> <expr12> ... ))
    (<slots2> (<expr21> <expr22> ... ))
    ...
    )
```

This grammar greatly reduces the burden of awkward expression in knowledge graph. We propose such a grammar for RDF and OWL in Section 3.

2.2 Frames to RDFS

Minsky [19] published the idea of framework of human cognitive mechanisms at 1974.

> We can think of a frame as a network of nodes and relations. The top levels of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many *terminals* – slots that must be filled by specific instances or data. [...] Collections of related frames are linked together into *frame-systems*. The effects of important actions are mirrored by *transformations* between the frames of a system. [Minskey, 1974]

As Minsky mentioned in his paper, the basic idea of frame is not his invention and his presentation was not complete, but he pointed out several important notions of frame systems such as sharing terminals, a frame and subframes, variables, attachments, default assignment, and so on. He talked the image of frame-based cognition in many scenes, vision, linguistics, memory acquisition, retrieval of knowledge, and control. Minsky's prevision had become the source of many frame systems after that.

KRL and FRL were the first two systems embodied Minsky's idea. Especially, Bobrow's KRL[20] gave some inspiration to KL-ONE family, where the appearance of network disappeared and frame-like forms, e.g., UNIT, appeared. The followings are an example of event description described in KRL.

```
[Event234 UNIT Individual
  <SELF (a Give with
         object = (a Pen)
         giver = (Person2 (which IsHusbandOf Person3)}
         recipient = Person1)>]
```

```
[Give UNIT Specialization
  <SELF (an Event)>
  <object (a Thing)>
  <giver (a Person)>
  <recipient (a Person)>]

[Lawyer UNIT Specialization
  <SELF (a Person)>]

[Pen UNIT Basic
  <SELF (a PhysicalObject)>]

[Lawyer UNIT Specialization
  <SELF (a Person)]

[Person1 UNIT Individual
  <SELF (a Person with firstName = "David")>]

[Person2 UNIT Individual
  <SELF {(a Person with firstName = "Jonathan")
         (which IsHusbandOf Person3)}>]

[Person3 UNIT Individual
  <SELF {(a Person with firstName = "Ellen")
         (a Lawyer)}>]
```

KRL allowed to explicitly express both classes and individuals in a uniform pattern named UNIT. It is obvious that "`(a Pen)`" denotes an instance of class `Pen`. The form "`(a <class> with <slot-name> = <slot-value>)`" indicates an instance of `<class>` has `<slot-value>` to the `<slot-name>` in both class and individual descriptions.

FRL [21, 22] was an implementation for several representation techniques suggested by Minsky and additional functionalities that are, today, conceived common in frame systems. Namely, local procedure attachments, IF-NEEDED, IF-ADDED, and IF-REMOVED were demonstrated in addition to basic functionalities of default value and inheritance along AKO (a kind of). However, the semantics of the inheritance, which should be turned into the subsumption from the viewpoint of RDF Semantics [2], is not formalized, whereas it worked practically.

Schank's Memory Organization Package (MOP) [23] was a very unique package for Case-Based Reasoning (CBR). It is possible to deem it a successor of frame systems, but the most strong impact of MOP on Semantic Webs is that it has made clear the bunch of slots (a set of pairs of property-name and property-value) of an instance define the class of the instance. In other words, the intension (slots) decides the extension (class). Actually, we utilized the algorithm of `slots->mop` in MOP onto building Agriculture Activity Ontology (AAO) [24] with SWCLOS. However, the MOP has no structural difference between classes and instances and the actual deference of MOP between them is just the flag for instance/class is true or false.

Common Lisp Object System (CLOS) [25, 26] is the first genuine class-based object system, in which the instance object structure is completely different from the class object structure, and we need the definition of a class before the instance object creation. We can define `Wine` class in pure CLOS as follows.

```
(defclass Wine (food:PotableLiquid)
  ((madeFromGrape :accessor madeFromGrape :type WineGrape))
  (:metaclass owl:Class))
```

Furthermore, if `rdfs:label` is defined as slot at the metaclass `owl:Class` with the *meta-object protocol* [27], we can add the `rdfs:label` slot of class `Wine` meta-object as follows.

```
(make-instance 'owl:Class :name 'Wine rdfs:label "Wine")
```

3 A Brief Overview of New Knowledge Machine

Along with the progression and popularization of Linked Open Data, more comprehensible and comprehensive tools are required for new entrants from ordinary people. It is easy to write down knowledge graphs at instance levels, but it is difficult to expand the knowledge graph to the ontology. It is obvious that we need new guides who lead starters in LODs to the ontologies. For the purpose of that, we are now tackling to develop a new platform of Semantic Webs that covers from LODs to ontologies. It will be an amalgam of a database for knowledge graph, gently guiding for novices, straight-forward representation of knowledge, etc. In this section, we give rough sketches of several parts of the tools.

3.1 Light Weight Database and Indexing in Knowledge Graph

The new platform has an interface to RDF stores, AllegroGraph, Virtuoso, Stardog, and so on. In addition, we have developed an internal graph memory as cache based on DTREE and the unification[4]. It demonstrated that Princeton WordNet [28] was absorbed on memory for the machine of Intel Core i7-4770, 16GB. In this module, we can perform simple information retrievals on one shot via simple pattern matching that is similar to SPARQL queries.

---

[4] See Paradigms of Artificial Intelligence Programming by Peter Norvig Section 14.8.

3.2 Case-based Hierarchy Adjustment

In addition to domain and range constraints in RDFS, which was already equipped on SWCLOS, we are going to implement flexible hierarchical structure adjustment. Schank's MOP provided the functionality of instance adjustment but not class adjustment. We will expand such adjustment to the class level. The rules of instantiation will work effectively at the class level.

3.3 New KM Language

We are developing a new KM language for Semantic Webs, which imitates the behavior of Clark's KM. Note that RDF semantics and OWL semantics are strictly preserved in the New KM. Especially, it will allows to represent meta-level ontologies [29] like:

```
Species is a owl:Class and rdfs:subClassOf owl:Class .
EndangeredSpecies is rdfs:subClassOf Species .
Eagle is a owl:Class and is a EndangeredSpecies .
Harry is a Eagle .
```

4 Conclusion and Future Work

This paper introduced a new knowledge representation machine, the New KM, which is a successor of SWCLOS, Schank's MOP, and Clark's KM, and more. It will ease the difficulties of entering Linked Open Data and proceeding to ontologies and meta-modeling of ontologies.

This paper pursued the history of knowledge representation from the beginning of *semantic networks* and *frame systems*, focusing on the evolution of class-instance notion and the implementation on representation languages. This paper rediscovered the five levels of *semantic networks*. We found that the highest level, *Linguistic level*, in which we can investigate any concepts and words as networks, is left as unexplored domain. The comprehensive platform based on RDF and OWL will be indispensable to build extensive ontologies from comprehensive web resources like Wikipedia, and others.

References

1. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, Peter Patel-Schneider (eds.): *The Description Logic Handbook*, Cambridge (2003)
2. Hayes, Patrick, Brian McBride: RDF Semantics, `http://www.w3.org/TR/rdf-mt/` (2004)
3. Woods, William A.: What's in a Link: Foundations for Semantic Networks, Representation and Understanding; Studies in Cognitive Science, D. G. Bobrow, A. M. Collins (eds.), 35–82, Academic Press (1975)
4. Nardi, Daniele, Ronald J. Brackman: An Introduction to Description Logics, in [1] 1–40.

5. Sattler, Ulrike, Diego Calvanese, Ralf Molitor: Relationships with other Formalisms, in [1] 137–177.
6. Noy, Natalya F., Deborah L. McGuinness: Ontology Development 101: A Guide to Creating Your First Ontology, `https://protege.stanford.edu/publications/ontology\_development/ontology101-noy-mcguinness.html`, Stanford.
7. Quillian, M. Ross: Semantic Memory, Report AFCRL-66-189, BBN, Cambridge (1966)
8. Quillian, M. Ross: Word Concepts: A Theory and Simulation of Some Basic Semantics Capabilities, *Behavioral Science*, **12**, 410–430 (1967), reprinted in *Readings in Knowledge Representation* (eds.) R. J. Brackman, H. H. Levesque, 98–118, Morgan Kaufmann (1985)
9. Clark, Peter, Bruce Porter: KM - The Knowledge Machine: Users Manual, Technical Report, AL Lab, Univ. Texas at Austin (1999)
10. Clark, Peter, Bruce Porter: KM - The Knowledge Machine 2.0: Users Manual, `http://www.cs.utexas.edu/users/mfkb/RKF/km.html`
11. Koide, Seiji, Masanori Kawamura: SWCLOS: A Semantic Web Processor on Common Lisp Object System, ISWC2004, demo, `http://iswc2004.semanticweb.org/demos/32/` (2004)
12. Koide, Seiji, Hideaki Takeda: OWL-Full Reasoning from an Object Oriented Perspective, ASWC2006, LNCS-4185, 263–277, Springer (2006)
13. Brackman, Ronald J.: What's in a Concept: Structural Foundations for Semantic Networks, Int. H. Man-Machine Studies, **9**, 127–152 (1977)
14. Brackman, Ronald J.: On the Epistemological Status of Semantic Networks, *Associative Networks: Representation and Use of Knowledge by Computers*, N. V. Findler (ed.) 3–50, Academic Press (1979)
15. Brackman, Ronald J., James G. Schmolze: An Overview of the KL-ONE Knowledge Representation System, *Cognitive Science*, **9**, 171–216 (1985)
16. Woods, William A., James G. Schmolze: The KL-ONE Family, *Computers Math. Applic.*, **23**-2-5, 133–177, Pergamon Press (1992)
17. Resnick, Lori A., Alex Borida, Ronald J. Brackman, Charles L. Isbell, Deborah L. McGuinness, Peter F. Patel-Schneider, Kevin C. Zalondek: CLASSIC Description and Reference Manual, `https://cse.buffalo.edu/~shapiro/Courses/CSE663/classicmanual.ps` (1995)
18. Brill, David: Loom Reference Manual Version 2.0, `ftp://ftp.isi.edu/pub/loom/reference2.0-twoside.ps` (1993)
19. Minsky, Marvin: A Framework for Representing Knowledge, *Mind Design*, J. Haugeland (ed.), pp.95–128, MIT Press (1981) Originally published as AI-Memo 306 (1974), `https://dspace.mit.edu/bitstream/handle/1721.1/6089/AIM-306.pdf?sequence=2`
20. Bobrow, Daniel G., Terry Winograd: An Overview of KRL, a Knowledge Representation Language, *Cognitive Science*, **1**-1, 3–46 (1977)
21. Roberts, R. Bruce, Ira P. Goldstein: The FRL Primer, AI Memo 408, MTI (1977)
22. Roberts, R. Bruce, Ira P. Goldstein: The FRL Manual, AI Memo 409, MIT (1977)
23. Riesbeck, Christopher K., Roger C. Schank: *Inside Case-based Reasoning*, Laurense Eribaunm Associates (1989)
24. Joo, Sungmin, Seiji Koide, Hideaki Takeda, Daisuke Horyu, Akane Takezaki, Tomokazu Yoshida: Designing of Ontology for Domain Vocabulary on Agriculture Activity Ontology (AAO) and a Lessons Learned, JIST 2016, 32–46, Springer (2016)
25. X3J13 Document 88-002R: Common Lisp Object System Specification, (1988)
26. Pitmann, Kent: Hyperspec, `http://www.lispworks.com/documntation/Hyperspec/Body/04\_c.htm`

27. Kiczales, Gregor, Jim des Rivières, Daniel G. Boblow: *The Art of the Meaobject Protocol*, MIT (1991)
28. Koide, Seiji, Hideaki Takeda: RDFization of Japanese Electronic Dictionaries and LOD, 2nd Workshop on Linked Data in Linguistics (LDL-3024), (2013)
29. Koide, Seiji, Hideaki Takeda: Inquiry into RDF and OWL Semantics, JIST 2016, 25–31, Springer (2016)