REVISION OF ISO 24707 (COMMON LOGIC)

MICHAEL GRÜNINGER, MEGAN KATSUMI, AND TILL MOSSAKOWSKI

Contents

Date: April 7, 2013.

1. What Is Common Logic?

Common Logic (published as "ISO/IEC 24707:2007 — Information technology Common Logic : a framework for a family of logic-based languages") is a language based on first-order logic, but extending it in several ways that ease the formulation of complex first-order theories. [**?**] discusses in detail the motivations and philosophical background of Common Logic, arguing that it not only is a natural formalism for knowledege representation in the context of the Web, but that it also constitutes a natural evolution from the canonical textbook notation and semantics of first-order logic (FOL), dispensing with some deeply entrenched views that are reflected in FOL's syntax, in particular the segregation of objects, functions, and predicates.

1.1. **First-Order Logic.**

First-order logic is widely used as the basis for knowledge representation and automated theorem proving in artificial intelligence and the Semantic Web. It's utility arises from its expressiveness (which allows users to write expressions for a wide range of commonsense domains such as time, space, and process). In addition, first-order logic has strong logical properties which guarantee that consistency (which is a syntactic notion) is equivalent to the existence of a satisfying interpretation (which is a semantic notion).

The set of logical symbols in the syntax of first-order logic extends the basic propositional connectives – conjunction, disjunction, negation, implication, and biconditional – with the notions of quantifiers and variables. A universal quantifier allows one to state that a formula holds for all instantiations of a variable, while an existential quantifier allows one to state that a formula holds for some instantiation of a variable.

First-order sentences also contain what are known as nonlogical symbols, which denote relations, functions and individuals within some domain of discourse. In the semantics of first-order logic, an interpretation assigns a set of n-tuples to each relation.

Common Logic extends first-order logic in two distinctive ways: (1) any term can be used as function or predicate, and (2) sequence markers allow for talking about sequences of individuals directly, and in particular, provide a succinct way for axiomatising functions and predicates with varying numbers of arguments.

1.2. **How Is Common Logic Used?**

Common Logic is used as a standard logic for knowledge representation and automated reasoning. In this capacity, Common Logic can either be used directly as a neutral representation which can then be reused by different automated reasoning tools, or Common Logic can be used as the underlying logic for integrating the knowledge that has been specified in the representation associated with particular software applications.

1.2.1. *Ontologies.*

Many tasks require correct and meaningful communication and integration among intelligent agents and information resources. A major barrier to such interoperability is semantic heterogeneity: different applications, databases, and agents may ascribe disparate meanings to the same terms or use distinct terms to convey the same meaning. Even when software applications use the same terminology, they often associate different semantics with the terms. This clash over the meaning of the terms prevents the seamless exchange of information among the applications. It has been recognized that the emerging technology concerned with the development and application of ontologies will play a central role in achieving semantic integration ([16]). An ontology is a computer-interpretable specification that is used by an agent, application, or other information resource to declare what terms it uses, and what the terms mean. Ontologies support the semantic integration of software systems through a shared understanding of the terminology in their respective ontologies.

Much of the work in the design and application of ontologies has been in the context of the Semantic Web and biomedical domains. These ontologies have been written using the Web Ontology Language (OWL), which is based on logics which are restrictions of first-order logic. Nevertheless, there are are many domains which require a logic that is at least as expressive as first-order logic, and Common Logic is the predominant language for the specification of ontologies in these domains. Section 2 and Section 3 discuss the current work in using Common Logic for the specification of ontologies.

1.2.2. *Semantic Interoperability.* Common Logic specifies a family of logic languages designed for use in the representation and interchange of information and data among disparate computer systems. Languages in the family share the same declarative semantics, so that it is possible to understand the meaning of expressions in these languages without appeal to an interpreter for manipulating those expressions. Languages in the family are logically comprehensive at its most general, they provide for the expression of arbitrary first-order logical sentences. These features enable software applications to exchange Common Logic expressions in such a way that the intended semantics of the terminology in the expressions is preserved.

1.3. **Additional Resources.** Research papers, presentations, and tutorials about Common Logic can be found at

`cl.tamu.edu`

Material related to current discussions and proposals for extending Common Logic is at

`ontolog.cim3.net/cgi-bin/wiki.pl?CommonLogic`

Issue tracking and other project management information is hosted at

`https://github.com/gruninger/Common-Logic`

2. COLORE: The Common Logic Ontology Repository

The objective of the Common Logic Ontology Repository (COLORE) project is to construct an open repository of first-order ontologies that will serve as a testbed for ontology evaluation and integration techniques, and that can support the design, evaluation, and application of ontologies in first-order logic. All ontologies are specified using Common Logic (ISO 24707).

The COLORE project is led by Dr. Michael Gruninger in the Department of Mechanical and Industrial Engineering at the University of Toronto. COLORE is also a participant in the Open Ontology Repository (OOR) Initiative. The charter of OOR (`oor.net`) is to the promote the global use and sharing of ontologies by:

- establishing a hosted registry-repository;
- enabling and facilitating open, federated, collaborative ontology repositories;
- establishing best practices for expressing interoperable ontology and taxonomy work in registry-repositories.

As of April 3, 2013, there are 1132 ontologies in COLORE, which can be found at

`code.google.com/p/colore/`

At the lowest level are theories of general mathematical structures, such as algebraic structures (e.g. semigroups, groups, rings, vector spaces), and combinatorial structures (e.g. orderings, lattices, graphs). These ontologies serve as the basis for the representation theorems for generic ontologies currently within the repository, such as processes, time, shape, mereotopology, and geometry. At the top level are ontologies for manufacturing standards and specific sofware applications.

2.1. **Building Ontologies.** An ontology repository supports the design of modular ontologies, New ontologies are constructed by extending existing ontologies within the repository. The `cl-imports` construct in Common Logic is the primary means for this notion of extension. In addition, there are cases where the ontology designer may want to restrict the domain of quantification for axioms in the ontology which is being imported. For example, an ontology of time may import an ontology of orderings, but restrict this ordering to timepoints. The first edition of Common Logic supported this technique through the `cl-module` construct.

2.2. **Relationships among Ontologies.** COLORE is organized into sets of ontologies, referred to as hierarchies. Ontologies within the same hierarchy use the same terminology, and they all extend a unique root theory which formalizes the fundamental ontological commitments that are shared by all ontologies in the hierarchy.

Relationships between ontologies in different hierarchies are specified by ontology mappings known as translation definitions. Given a source ontology O and a target ontology T, the mappings provide a definition of the terminology in T using only the terminology in O. We say that the ontology O interprets the ontology T if we can deduce the axioms in T from the axioms of O together with the translation definitions. An important aspect of translation definitions is that they form what is known as a definitional extension of the source ontology – we should not be able to infer any new statements about O from the translation definitions. If we could, then we would really be extending O with new ontological commitments rather than specifying a mapping to ontology T.

3. Relationship to Other Standards

3.1. **Standards with Ontologies in Common Logic.**

3.1.1. *PSL Ontology.* The Process Specification Language (PSL) is a Common Logic ontology for process modelling which enables the interoperability of manufacturing process descriptions between manufacturing engineering and business software applications such as process planning, scheduling workflow, business process reengineering, and project management.

PSL has been demonstrated in several software interoperability scenarios ([3],[4],[5]) and several companies have built commercial software tools to manipulate PSL process descriptions. In 2005, PSL was officially published as an International Standard with the name ISO 18629 by the Joint Working Group 8 of Sub-committee 4 *Industrial data* and Sub-committee 5 *Manufacturing integration* of Technical committee ISO TC 184, *Industrial automation systems and integration* of the International Organization of Standardization.

3.1.2. *Date-Time Vocabulary.* The objective of the Date-Time Vocabulary (DTV) project within the Object Management Group (OMG) is to provide an ontological foundation for concepts related to time in the specification of business rules. For example, such business rules often contain conditions that events must occur every month, and business processes such as order entry require concepts such as "shipping date". The current version of DTV is axiomatized within Common Logic, and can be found in the COLORE repository.

3.1.3. *UML.* fUML (Foundational UML) is a subset of the Unified Modelling Language (UML) version 2 defined by the Object Management Group (OMG). The OMG has specified a foundational execution semantics for fUML using CL (see `http://www.omg.org/spec/FUML/`).

3.1.4. *Semantic Web Services Framework.* The Semantic Web Services Framework (SWSF) has been proposed to support richer semantic specifications of Web services, based on a comprehensive representational framework that spans the full range of service-related concepts. Such a framework enables fuller, more flexible automation of service provision and promotes the use of semantically well-founded reasoning about services. For example, richer semantics can support greater automation of service selection, composition, invocation, and verification.

SWSF includes an axiomatized ontology of service concepts (known as FLOWS, First-order Logic Ontology for Web Services [**?**], [**?**]), which used Common Logic to extend the PSL Ontology to provide the conceptual framework for describing and reasoning about services. This ontology was originally proposed to address the shortcomings of earlier approaches such as OWL-S, whose description logic specification could not axiomatize the intended semantics of its terminology.

In September 2005, SWSF was published as a W3C Member Submission for the World Wide Web Consortium.

3.2. **OntoIOp.** ISO/TC37/SC3/WG3: ISO WD 17347 - Ontology Integration and Interoperability (OntoIOp) Part 1: The Distributed Ontology Language (DOL) is a proposal for an ontology metalanguage that is able to specify relationships between ontologies and relationships between ontology languages. OntoIOp supports translations between Common Logic and other ontology representation languages such as OWL and RDF.

4. Software Support for Common Logic

Current software tool support for Common Logic is still ad hoc, and is primarily restricted to parsers and translators between CLIF and the TPTP exchange syntax for first-order logic. Nevertheless, there are a number of projects under development which are driving the application of Common Logic. In this section, we give an overview of a variety of these projects within industry and academia.

4.1. **HETS.** The Heterogeneous Tool Set (HETS) is an open source software providing a general framework for formal methods integration and proof management. One can think of HETS as acting like a motherboard where different expansion cards can be plugged in, the expansion cards here being individual logics (with their analysis and proof tools) as well as logic translations. The HETS motherboard already has plugged in a number of expansion cards (e.g., theorem provers like SPASS, Vampire, LEO-II, Isabelle and more, as well as model finders).

HETS has been extended with the following kinds of tool support for Common Logic:

- a parser for the Common Logic Interchange Format(CLIF) and the Knowledge Interchange Format (KIF);
- a connection of CL to well-known first-order theorem provers such as SPASS, Darwin and Vampire, so that logical consequences of CL theories can be proved;
- a connection of CL to the higher-order provers Isabelle/HOL and LEO-II in order to perform induction proofs in theories involving sequence markers;
- a connection to first-order model finders such as Darwin that allow one to find models for CL theories;
- support for proving interpretations between CL theories to be correct;
- a translation that eliminates the use of CL modules. Since the semantics of CL modules is specific to CL, this elimination of modules is necessary before sending CL theories to a standard first-order prover;
- a translation of the Web Ontology Language OWL to CL.

4.2. **Samian Platform.** Kojeware Corporation (based in Waterloo, Ontario) is developing software environments to support Common Logic applications. The following is taken from `http://www.kojeware.com/`

> Kojeware specializes in open software infrastructure and software tooling for "heavyweight" semantic systems. The Samian Platform integrates state-of-the-art technologies from the field of Semantic Computing into a unified and open environment for developing semantic systems. The Samian Platform may be used within solutions for system interoperability, semantically assisted search, and knowledge-based decision support.
>
> The $SamianODE^{TM}$ Plug-in for Eclipse provides a feature rich Ontology Development Environment (ODE) for creating first-order theories, ontologies and knowledge-bases represented in Common Logic.
>
> A CLIF aware text editor helps ensure CL texts conform to the ISO 24707 standard. The editor also provides CLIF syntax highlighting and parenthesis matching capabilities.

4.3. **MACLEOD.** MACLEOD is a Common Logic Environment for Ontology Development that will effectively support the development of an ontology across the entire ontology development lifecycle. Although several ontology development tools (such as Protege and TopBraid Composer) exist for ontologies specified in the Web Ontology Language (OWL), there are few tools that provide compehensive support for ontology development in Common Logic.

The work that occurs in the design phase of ontology development requires the ability to add, modify, and remove axioms in the ontology. For documentation purposes, the system must keep a record of all axioms and their revisions, accompanied by a specification of the (re-)design rationale where possible

The system should facilitate versioning of the ontology, each axiom, and any associated modules. The system should also support the techniques of model exploration suggested in the design phase - it must support the use of any first-order logic model generator. The system should store any models generated, providing relevant information such as an association to the version of axioms that were used in the input file.

The verification phase of ontology development focuses on the use of an automated theorem prover to evaluate the semantic requirements; the system must support the use of a first-order logic automated theorem prover to accomplish this. A more subtle aspect of verification is the translation of the semantic requirements into a set of tests for the theorem prover. The system must generate the appropriate input file for the theorem prover, depending on the semantic requirement being tested; it should allow the user to select subsets to be used instead of the axiomatization of the entire ontology; it should allow for the inclusion of lemmas; and it should document all of these choices with the test itself.

The development of MACLEOD as a software environemnt is being organized through a project on github.

4.4. **OntologyPortal.** SUMO (Suggested Upper Merged Ontology) is a large upper ontology, covering a wide range of concepts. It is one candidate for the "standard upper ontology of IEEE working group 1600.1. While SUMO has originally been formulated in the Knowledge Interchange Format (the predecessor to Common Logic), SUMO-CL is a CL variant of SUMO produced by Kojeware. The SUMO-KIF version with all the mid-level and domain ontologies that are shipped with SUMO consists of roughly 32,000 concepts in 150,000 lines of specification.

Additional information about software tool support related to SUMO can be found at `www.ontologyportal.org`:

> "The Sigma knowledge engineering environment is an system for developing, viewing and debugging theories in first order logic. It works with Knowledge Interchange Format (KIF) and is optimized for the Suggested Upper Merged Ontology (SUMO).
>
> Sigma includes a number of useful features for knowledge engineering work, including term and hierarchy browsing, the ability to load different files of logical theories, a full first order inference capability with structured proof results, a natural language paraphrase capability for logical axioms, support for displaying mappings to the WordNet lexicon and a number of knowledge base diagnostics."

4.5. **RECON.** Researchers at the National Institute of Standards and Technology are currently on a project called RECON (Restricted English for Constructing Ontologies) that supports the translation of business rules from natural language into the IKRIS Knowledge Language (IKL), which is an extension of Common Logic. The project also includes the implementation of an automated theorem prover which can reason directly with Common Logic ontologies, without the need to first translate into the syntax of an existing theorem prover.

5. Proposed Revisions to Common Logic

The applications of Common Logic and software implementations for Common Logic support that we have discussed in Sections 2,3, and 4 have led to the identification of several shortcomings in the current version of ISO 24707. As a result, users and technical experts have submitted a set of possible changes and extensions to the standard. The proposal for the Second Edition of ISO 24707 (Common Logic) encompasses both syntactic and semantic issues, which are summarized below; they are discussed in more detail in the attached documents.

5.1. **Syntactic Issues.** Syntactic issues within the scope of the revision include:

(1) Fixing the list of syntactic errors that have already been identified in Defect Report
(2) Correction and completion of the XML syntax in Annex C
(3) More general approach to annotation of cl-texts

5.1.1. *Defect Report.* The implementation of parsers for the concrete syntax of CLIF (Common Logic Interchange Format) led to the identification of errors in the specification of the syntax. These errors are summarized in the Defect Report.

5.1.2. *XML Syntax for CL.* A concrete XML syntax (referred to as XCL) for Common Logic was specified in Annex C in the original edition of the standard. Tara Athan (Athan Services) has identified several problems with the syntax. In addition, proposed changes to the semantics of Common Logic will lead to changes in this concrete syntax.

5.1.3. *Annotation of cl-texts.* The original edition of the standard provides for simple comments, but projects such as COLORE and HETS have demonstrated that a more sophisticated approach to the annotation of Common Logic ontologies is required. For example, the integration of ontology verification with automated theorem provers would be greatly assisted by providing explicit names for each axiom in an ontology.

5.2. **Semantic Issues.** Semantic Issues which are modifications of the existing standard:

(1) Modification of semantics to allow the existence of definitional extensions in CL
(2) Circular imports
(3) Semantics of cl-module
(4) Questions about segregated dialects and interoperability
(5) Clarification of conformance conditions

Semantic issues which constitute new features for the Second Edition:

(1) Namespacing
(2) Numerical quantifiers

5.2.1. *Definitional Extensions.* We saw above that a crucial tool for specifying ontology mappings in Common Logic is through translation definitions, which form a conservative extension of the target ontology. However, work done by Fabian Neuhaus (NIST) and Till Mossalowski (Universität Bremen) showed that the semantics of Common Logic prevented the existence of definitional extensions of ontologies. The revision of Common Logic will include the modification to the semantics which addresses this problem.

5.2.2. *Circular Imports.* Although modular ontologies can be constructed through importing ontologies via `cl-imports`, Tara Athan () and Michael Grüninger noticed that one can construct cases in which one can specify ontologies that import each other. The semantics of `cl-imports` in the original edition of Common Logic is unable to correctly handle this situation.

5.2.3. *Semantics of cl-module.* The semantics of `cl-module` is not treated uniformly across the first edition of the standard. Moreover, Fabian Neuhaus and Pat Hayes have identified several problems with the way in which the local domain of discourse is treated within modules.

5.2.4. *Segregated Dialects and Interoperability.* One of the distinctive features of Common Logic is that its syntax does not make a distinction between relations, functions, and constants – all are in the domain of quantification. This is in contrast with the traditional approach to first-order logic, which segregates relation and functions from constants. Although the original edition of Common Logic allows both the unsegregated and segregated dialects, there are still open questions about the correct semantics of ontologies that combine both dialects through importation.

5.2.5. *Clarification of Conformance Conditions.* One of the original motivations for Common Logic was to support semantic interoperability. Given the variety of ontology specification languages currently in use, it has been recognized that not all are exactly semantically conformant with Common Logic. The conformance conditions specified in Clause 7 of the original standard need to be clarified to fully account for semantic sub-dialects that are partially conformant.

5.2.6. *Namespacing.* It can be readily seen in ontology repositories such as COLORE that multiple ontologies often use the same terms but in very different contexts. For example, a term such as *between* appears in both time ontologies as well as spatial ontologies and mathematical theories of geometry. Although in best practice, each relation in an ontology would be named by URI, an extension of Common Logic which supported namespaces would enhance the design of integrated ontologies. Furthermore, translation definitions among the ontologies in the above examples are more easily treated in the context of namespaces.

5.3. **Additional Aspects.** New Informative Annexes are also proposed for the Second Edition:

(1) Informative Annex that includes the axiomatization of useful 'structural' axioms eg arity of relations, allDifferent, etc.. which are ontologically neutral
(2) Informative Annex on CL proof theory

The following issues are considered to be Optional for the Second Edition:

(1) Additional concrete syntax to include in Annexes (e.g. infix)
(2) Additional connectives
(3) Informative Annex that provides a reserved vocabulary for structural descriptive names (in the sense of Tarski)
(4) Query language for Common Logic

6. Summary

Common Logic is a family of logical languages which is widely used to support a wide range of applications in knowledge representation and automated theorem proving, both in artificial intelligence and the Semantic Web. The implementation of software tools for Common Logic and repositories of Common Logic ontologies have led to the identification of shortcomings and errors in the first edition of the ISO 24707 standard. Given the scope of the changes that are being proposed, we recommend that the development of a Second Edition of ISO 24707 (Common Logic) be initiated.

Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada, gruninger@mie.utoronto.ca

Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada, gruninger@mie.utoronto.ca

DFKI GmbH Bremen, mossakowski@uni-bremen.de