Contextual ontologies

Sources:

Klarman, S., & Gutiérrez-Basulto, V. (2011). Two-Dimensional Description Logics for Context-Based Semantic Interoperability. In Proc. Twenty-Fifth AAAI Conference on Artificial Intelligence.

Aljalbout S., Buchs D., Falquet G. (2019) Introducing Contextual Reasoning to the Semantic Web with OWL$\mathcal{C}$. In Proc. Intl Conf. on Conceptual Structures – Graph-Based Representation and Reasoning. ICCS 2019. Lecture Notes in Computer Science, vol 11530. Springer.

G. Falquet

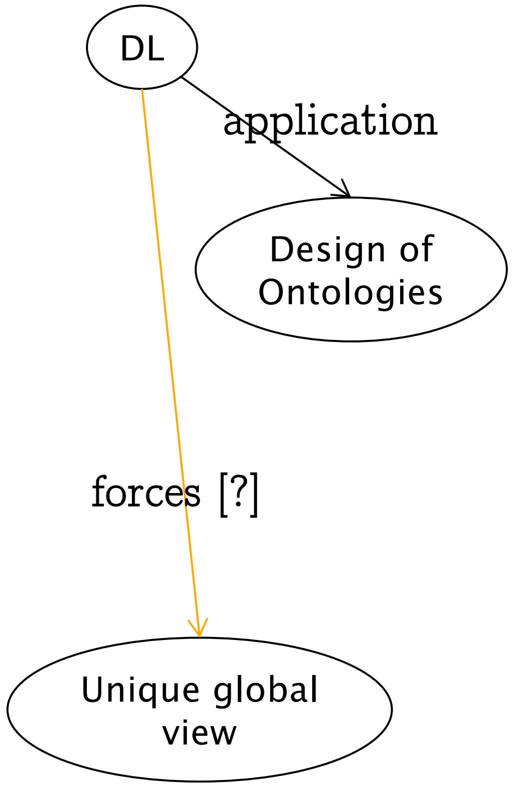
e.g. semantic indexing of texts

[...]

Description Logics (DLs) are popular knowledge representation formalisms, whose most prominent application is the design of ontologies

[...]

Under the standard Kripkean semantics, a DL ontology forces a unique, global view on the represented world, in which the ontology axioms are interpreted as universally true.



McCarthy theory of contexts

replace logical formulas φ, as the basic knowledge carriers, with

ist(c,φ) stating that φ is true in c,

Postulates

1. Contexts are formal objects.
   - anything that can be denoted by a first-order term

2. Contexts have properties and can be described.
   - allows $\forall x(C(x) \rightarrow \text{ist}\ (x, \varphi))$,
     expressing that φ is true in every context of type C .

3. Contexts are organized in relational structures.
   - contexts are often accessed from other contexts. Formally, this can be captured by allowing nestings of the form $\text{ist}(c, \text{ist}(d, \varphi))$.

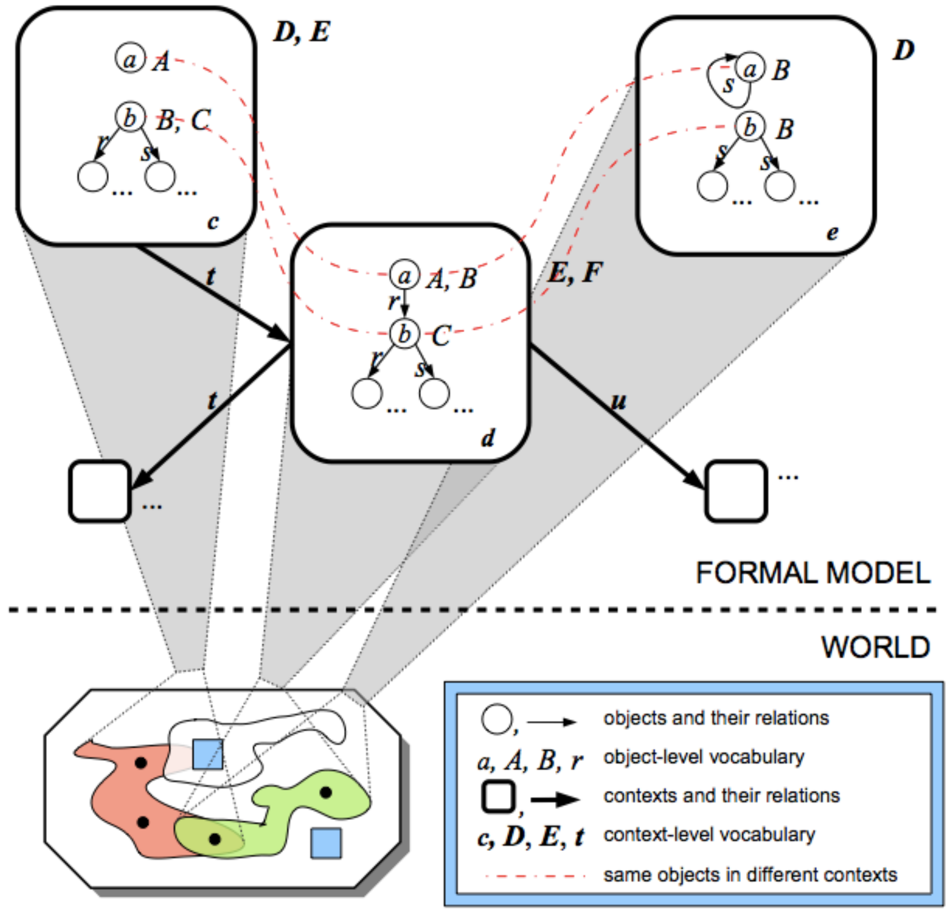


Fig. 1. A formal model of an application domain complying to McCarthy's principles.

Description Logic of Context

A Description Logic of Context C(LC,LO) consists of

- a DL context language LC supporting context descriptions
- an object language LO equipped with context operators for representing object knowledge relative to contexts.

The *context language* is a DL language LC over the vocabulary $\Gamma = (M_C$, M_R , M_I), (concepts, roles, individuals)

with a designated subset $M_I{}^\star \subseteq M_I$ (the context names) .

Object language

The object language of C(LC, LO) is the smallest language containing LO and closed under the constructors of LO and one of the two types of concept-forming operators:

- $\langle r.C \rangle\ D \mid [r.C]\ D \qquad (F1)$
- $\langle C \rangle D \mid [C]D \qquad (F2)$

where C is a concept and r a role of the context language and D is a concept of the object language.

Meaning of the contextual expressions

$\langle C \rangle$ D

all objects which are D in some context of type C,

[C] D

all objects which are D in every such context.

$\langle r.C \rangle$ D

all objects which are D in some context of type C accessible from the current one through r.

[r.C] D

all objects which are D in every such context.

$$\langle \text{EuropeanCountry} \rangle \text{ Citizen}$$

The concept Citizen in some context of type EuropeanCountry.

$$\langle \text{neighbor.Country} \rangle \text{ Citizen}$$

The concept Citizen in some context of type Country accessible through the neighbor relation from the current context.

Knowledge base

A knowledge base is a pair $K = (\mathbf{C}, O)$, where

- $\mathbf{C}$ is a set of axioms of the context language
- O is a set of formulas of the form:

$$c{:}\varphi \quad | \quad C{:}\varphi$$

where

φ is an axiom of the object language (subClass, instance, role axiom)

$c \in M_I{}^{\star}$ (a context)

C is a concept of the context language. (a context class)

meaning of the assertions

$c : \varphi$ states that the axiom φ holds in the context denoted by the name c.

1969 : CanVote $\sqsubseteq$ Aged21orMor

1857 : Professor(Saussure)

California : LegalForRecreationalUse(cannabis)

meaning of the assertions

$C : \varphi$ assert the truth of φ in all contexts of type C

$$\text{EUCountry} : \text{Prohibited}(\textit{remote\_biometric\_identification})$$

in every EU country *remote biometric identification* is prohibited.

$$\text{Country} : \langle \text{neighbor.Country} \rangle \text{Citizen} \sqsubseteq \text{NoVisaRequirement}$$

in every Country a person who is a citizen in some neighboring country does not need a visa

formal interpretation of the language

In standard DL an interpretation is made of

- a domain
- an interpretation function $.^I$ for concepts, properties, and individuals

  $C \rightarrow C^I$, $R \rightarrow R^I$, $o \rightarrow o^I$

In a contextual DL an interpretation is made of

- a contextual domain **C**
- for each context i in **C**

  an interpretation function $.^{I[i]}$ for concepts, properties, and individuals

  $C \rightarrow C^{I[a]}, C^{I[b]}, C^{I[c]}, \ldots$

Contextual interpretation

Table 1. OWL 2 DL^{C}_{core} direct model theoretic semantics

Abstract syntax	CDL syntax	Semantics (interpretation in context k)
IntersectionOf(C_1 ... C_n)	$C_1 \sqcap ... \sqcap C_n$	$C_1^{\mathcal{I}[k]} \cap ... \cap C_n^{\mathcal{I}[k]}$
UnionOf(C_1 ... C_n)	$C_1 \sqcup ... \sqcup C_n$	$C_1^{\mathcal{I}[k]} \cup ... \cup C_n^{\mathcal{I}[k]}$
ComplementOf(C)	$\neg C$	$(\neg C)^{\mathcal{I}[k]} = \Delta^{\mathcal{I}[k]} \setminus C^{I[k]}$
R SomeValuesFrom(C)	$\exists(R.C)$	$x \vert \exists y : (x, y) \in (R)^{\mathcal{I}[k]} and \quad y \in (C)^{\mathcal{I}[k]}$
R AllValuesFrom(C)	$\forall(R.C)$	$x \vert \forall y : (x, y) \in (R)^{\mathcal{I}[k]} \rightarrow y \in (C)^{\mathcal{I}[k]}$
OneOf(a_1 ...a_n)	a_1 ... a_n	$(a_1)^{\mathcal{I}[k]}, ..., (a_n)^{\mathcal{I}[k]}$

Table 2. Semantics of the contexts-based concept forming operators.

Abstract syntax	CDL	Semantics
ConceptValuesFromSomeContext(C [K])	$\langle K \rangle\, C$	$x \in \Delta \mid \exists y \in K^{\mathcal{J}} : x \in C^{\mathcal{I}[y]}$
ConceptValuesFromAllContext(C [K])	$[K]C$	$x \in \Delta \mid \forall y \in K^{\mathcal{J}} \rightarrow x \in C^{\mathcal{I}[y]}$
ConceptValuesFromThisContext(C [k])	$\{k\}\, C$	$x \in \Delta \mid x \in C^{\mathcal{I}[k^{\mathcal{J}}]}$
PropertyValuesFromSomeContext(R [K])	$\langle K \rangle\, R$	$(x, z) \in \Delta \times \Delta \mid \exists y \in K^{\mathcal{J}} : (x, z) \in R^{\mathcal{I}[y]}$
PropertyValuesFromAllContext(R [K])	$[K]R$	$(x, z) \in \Delta \times \Delta \mid \forall y \in K^{\mathcal{J}} \rightarrow (x, z) \in R^{\mathcal{I}[y]}$
PropertyValuesFromThisContext(R [k])	$\{k\}\, R$	$(x, z) \in \Delta \times \Delta \mid (x, z) \in R^{\mathcal{I}[k^{\mathcal{J}}]}$

Example: geographic context

$$
\begin{array}{lll}
\mathcal{C}: & \boldsymbol{Country(germany)} & (1) \\
 & \boldsymbol{neighbor(france, germany)} & (2) \\
\mathcal{O}: & \boldsymbol{germany} : \exists hasParent.Citizen(john) & (3) \\
 & \boldsymbol{Country} : \exists hasParent.Citizen \sqsubseteq Citizen & (4) \\
 & \boldsymbol{france} : \langle \boldsymbol{neighbor.Country} \rangle Citizen \sqsubseteq NoVisaRequirement & (5)
\end{array}
$$

consequence: john is an instance of NoVisaRequirement

Example: context = ontology

$$
\begin{array}{lll}
\mathcal{C}: & \boldsymbol{HumanAnatomy} \sqsubseteq \boldsymbol{Anatomy} & (1) \\
\mathcal{O}: & \top : \langle \boldsymbol{HumanAnatomy} \rangle Heart \sqsubseteq [\boldsymbol{Anatomy}] HumanHeart & (2) \\
 & \boldsymbol{Anatomy} : Heart \sqsubseteq Organ & (3)
\end{array}
$$

- An ontology of human anatomy is an ontology of anatomy
- In any ontology the concept HumanHeart corresponds to the concept Heart in an ontology of the human anatomy
- In every anatomical ontology the heart is an organ

Example: Describing alignments

concept alignment: $\top : \langle \boldsymbol{A} \rangle C \sqsubseteq [\boldsymbol{B}] D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in every ontology of type $\boldsymbol{B}$

semantic importing: $\boldsymbol{c} : \langle \boldsymbol{A} \rangle C \sqsubseteq D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in ontology $\boldsymbol{c}$

upper ontology axiom: $\boldsymbol{A} : C \sqsubseteq D$
axiom $C \sqsubseteq D$ holds in every ontology of type $\boldsymbol{A}$

Implementation on the SW

- representing individual axioms
- representing class/role axioms
- reasoning rules
- extension to context sets

Element E of the language	**Triples and quadruples in $\tau(E)$**	**Main Node of $\tau(E)$**
$\langle K \rangle D$	`_:x owlc:onClass` $\tau(D)$ `_:x owlc:inSomeContextOf` $\tau(K)$	`_:x`
$[K]D$	`_:x owlc:onClass` $\tau(D)$ `_:x owlc:inAllContextOf` $\tau(K)$	`_:x`
$K : C(a)$	$\tau(a)$ `rdf:type` $\tau(C)$ $\tau(K)$	
$K : R(a,b)$	$\tau(a)$ $\tau(R)$ $\tau(b)$ $\tau(K)$	
$K : C \sqsubseteq D$	$\tau(C)$ `rdfs:subClassOf` $\tau(D)$ $\tau(K)$	

$K : R \sqsubseteq S$	$\tau(R)$ `rdfs:subPropertyOf` $\tau(S)$ $\tau(K)$	
$\langle r.K \rangle D$	`_:x owlc:onClass` $\tau(D)$ `_:x owlc:inSomeContextOf` $\tau(K)$ `_:x owlc:linkedThrough` $\tau(r)$	`_:x`
$[r.K]D$	`_:x owlc:onClass` $\tau(D)$ `_:x owlc:inAllContextOf` $\tau(K)$ `_:x owlc:linkedThrough` $\tau(r)$	`_:x`

Adaptation of the OWL2 RL Rules

Table 3. OWL^C: Extended entailment rules for the core language

	IF	THEN
cls-com $\neg C$	T(?c_1, owl:complementOf, ?c_2) Q(?x, rdf:type, ?c_1, ?k) Q(?x, rdf:type, ?c_2, ?k)	false
cls-int1 $C \sqcap D$	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn] Q(?y, rdf:type, ?c1, ?k) Q(?y, rdf:type, ?c2, ?k) ... Q(?y, rdf:type, ?cn, ?k)	Q(?y, rdf:type, ?c, ?k)
cls-int2 $C \sqcap D$	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c, ?k)	Q(?y, rdf:type, ?c_1, ?k) Q(?y, rdf:type, ?c_2, ?k) ... Q(?y, rdf:type, ?c_n, ?k)
cls-uni $C \sqcup D$	T(?c, owl:unionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c_i, ?k)	Q(?y, rdf:type, ?c, ?k)

cls-svf1-1 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, ?p, ?v, ?k) Q(?v, rdf:type, ?y, ?k)	Q(?u, rdf:type, ?x, ?k)
cls-svf1-2 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) T(?u, ?p, ?v) Q(?v, rdf:type, ?y, ?k)	Q(?u, rdf:type, ?x, ?k)
cls-svf1-3 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, ?p, ?v, ?k) T(?v, rdf:type, ?y)	Q(?u, rdf:type, ?x, ?k)
cls-avf-1 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) Q(?u, ?p, ?v, ?k)	Q(?v, rdf:type, ?y, ?k)
cls-avf-2 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) T(?u, ?p, ?v)	Q(?v, rdf:type, ?y, ?k)

cls-avf-1 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) Q(?u, ?p, ?v, ?k)	Q(?v, rdf:type, ?y, ?k)
cls-avf-2 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) T(?u, ?p, ?v)	Q(?v, rdf:type, ?y, ?k)
cls-avf-3 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) Q(?u, ?p, ?v, ?k)	T(?v, rdf:type, ?y)

Table 4. OWL^C: entailment rules for the context-based concept forming operators

	IF	THEN
cxt-svf ($\langle K \rangle D$)	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inSomeContextOf, ?k) Q(?x, rdf:type, ?d, ?y) T(?y, rdf:type, ?k)	T(?x, rdf:type, ?e)
cxt-avf ($[K]D$)	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inAllContextOf, ?k) T(?x, rdf:type, ?e) Q(?x, rdf:type, ?d, ?y)	T(?y, rdf:type, ?k)
cxt-ov ($\{K\}\ D$)	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inThisContext, ?k) Q(?x, rdf:type, ?e)	Q(?x, rdf:type, ?d, ?k)