Sigma: An Integrated Development Environment for Logical Theories

Adam Pease and Christoph Benzmüller*

Articulate Software, Angwin, CA, USA

IKBET-2010, August 16-17, 2010 Lisbon, Portugal

*supported by DFG grant BE 2501/6-1

What is Sigma?

Sigma supports the development of formal ontologies (logical theories) in various ways

- browsing and display
- analysis and debugging
- inference
- mapping, merging and translation

What is Sigma?

Sigma supports the development of formal ontologies (logical theories) in various ways

- browsing and display
- analysis and debugging
- inference
- mapping, merging and translation

Sigma has been created to support the development of the Suggested Upper Ontology SUMO

However, it is not restricted to SUMO!

Sigma supports the development of (1) formal ontologies (logical theories) in various ways

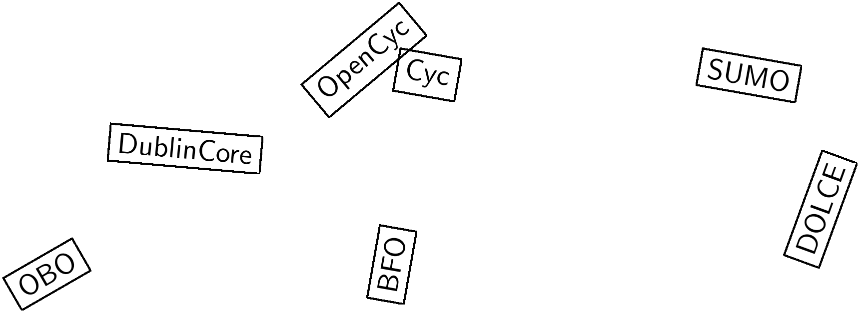
- (2) browsing and display
- (3) analysis and debugging
- (4) inference
- (5) mapping, merging and translation

Sigma has been created to support the development of the Suggested Upper Ontology (1) SUMO

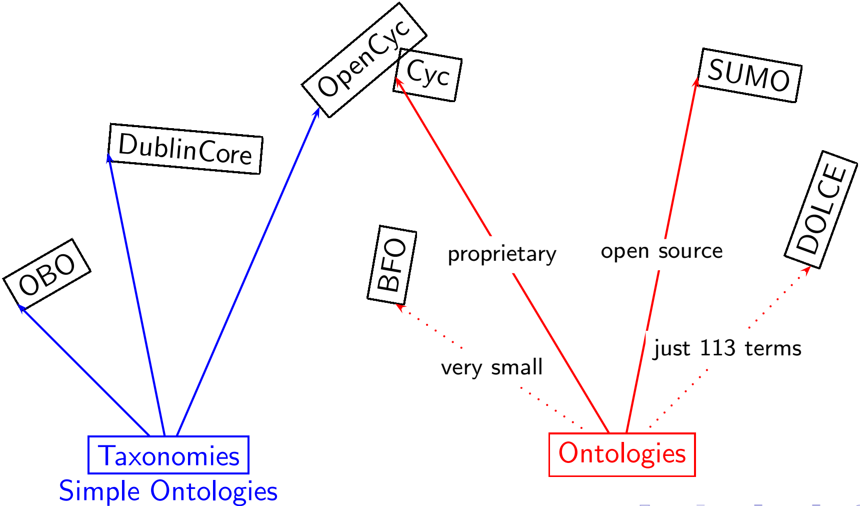
However, it is not restricted to SUMO!

(Formal) Ontologies and SUMO

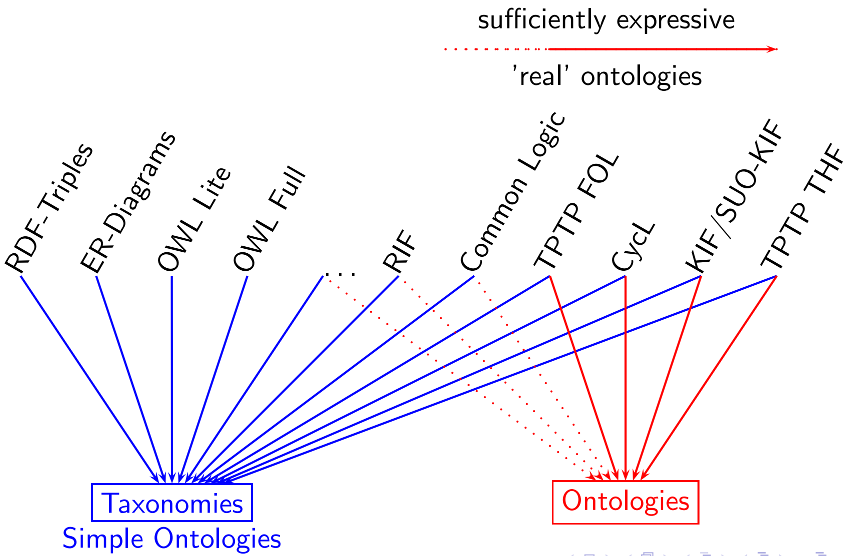
(Formal) Ontology: An Overstretched Notion?



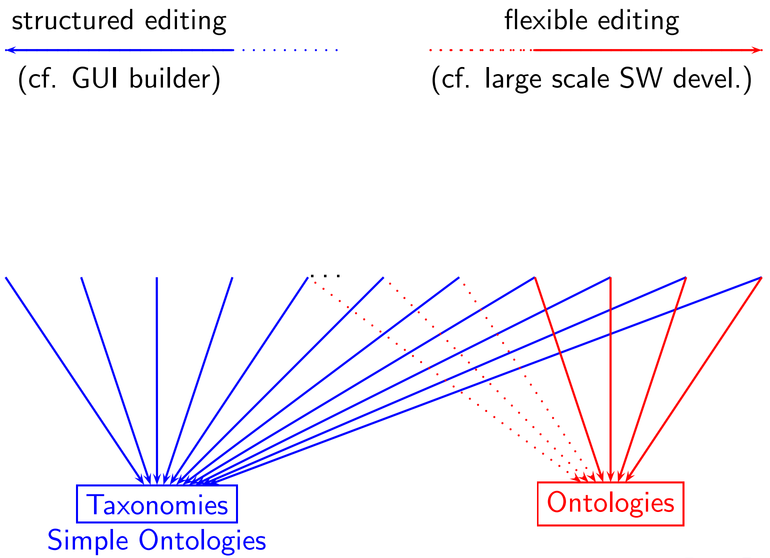
Ontology: An (Over-)Stretched Notion?



Ontology: An (Over-)Stretched Notion?



Sigma is not a graphically-based authoring tool!



Taxonomy versus Ontology: Examples

Simple Taxonomy (excerpt from SUMO's base taxonomy)

```
(subclass Physical Entity)
(subclass Abstract Entity)
(partition Entity Physical Abstract)

(subclass Object Physical)
(subclass Process Physical)
(partition Physical Object Process)


...
```

Taxonomy versus Ontology: Examples

The SUMO Ontology is more than just a Taxonomy

```
(subclass Physical Entity)
(subclass Abstract Entity)
(partition Entity Physical Abstract)

(subclass Object Physical)
(subclass Process Physical)
(partition Physical Object Process)
```

```
(<=>
    (instance ?PHYS Physical)
    (exists (?LOC ?TIME)
        (and
            (located ?PHYS ?LOC)
            (time ?PHYS ?TIME))))
```

Taxonomy versus Ontology: Examples

The SUMO Ontology is more than just a Taxonomy

```
(subclass Physical Entity)
(subclass Abstract Entity)
(partition Entity Physical Abstract)

(subclass Object Physical)
(subclass Process Physical)
(partition Physical Object Process)
```

```
(<=>
    (instance ?PHYS Physical)
    (exists (?LOC ?TIME)
        (and
            (located ?PHYS ?LOC)
            (time ?PHYS ?TIME))))
```

```
(subclass ObjectAttitude IntentionalRelation)

(=>
    (and
        (instance ?REL ObjectAttitude)
        (?REL ?AGENT ?THING))
    (instance ?THING Physical))
```

The SUMO Ontology: Some Facts

- developed since: > 10 years
- original motivation: education support (e.g. government training applications)
- open source: www.ontologyportal.org
- SUMO versus SUMO:

  SUMO: Suggested Upper-level Ontology
  MILO: Mid-level Ontology
  Specific Domain-level Ontologies
  } SUMO

- representation language: SUO-KIF (adaptation of the Knowledge Interchange Format KIF)
- logic: mainly first-order; some higher-order extensions

- developed since: > 10 years
- original motivation: education support (e.g. government training applications)
- open source: www.ontologyportal.org
- SUMO versus SUMO:

  SUMO: Suggested Upper-level Ontology
  MILO: Mid-level Ontology
  Specific Domain-level Ontologies
  } SUMO

- representation language: SUO-KIF (adaptation of the Knowledge Interchange Format KIF)
- logic: mainly first-order; some higher-order extensions

The SUMO Ontology: Some Facts

- developed since: > 10 years
- original motivation: education support (e.g. government training applications)
- open source: www.ontologyportal.org
- SUMO versus SUMO:

  SUMO: Suggested Upper-level Ontology
  MILO: Mid-level Ontology
  Specific Domain-level Ontologies
  } SUMO

- representation language: SUO-KIF (adaptation of the Knowledge Interchange Format KIF)
- logic: mainly first-order; some higher-order extensions

The SUMO Ontology: Some Facts

- developed since: > 10 years
- original motivation: education support (e.g. government training applications)
- open source: `www.ontologyportal.org`
- SUMO versus SUMO:

  SUMO: Suggested Upper-level Ontology
MILO: Mid-level Ontology
Specific Domain-level Ontologies } SUMO

- representation language: SUO-KIF (adaptation of the Knowledge Interchange Format KIF)
- logic: mainly first-order; some higher-order extensions

The SUMO Ontology: Some Facts

- developed since: > 10 years
- original motivation: education support (e.g. government training applications)
- open source: `www.ontologyportal.org`
- SUMO versus SUMO:

  SUMO: Suggested Upper-level Ontology
  MILO: Mid-level Ontology
  Specific Domain-level Ontologies
  } SUMO

- representation language: SUO-KIF (adaptation of the Knowledge Interchange Format KIF)
- logic: mainly first-order; some higher-order extensions

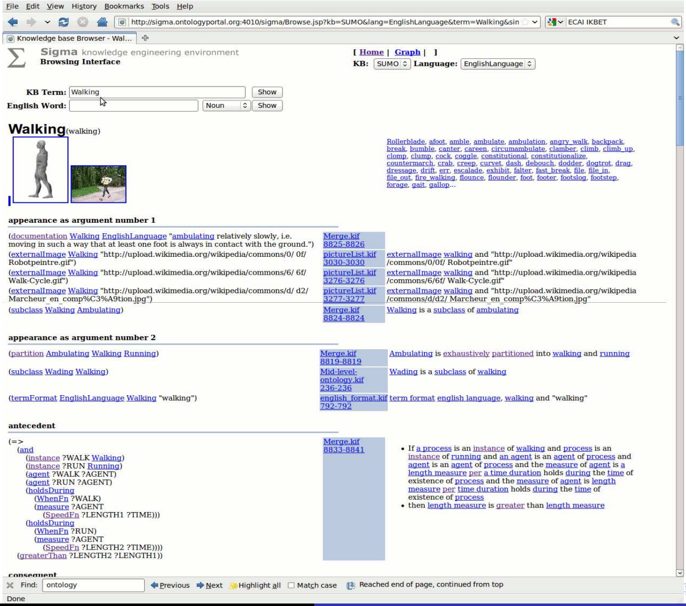
The SUMO Ontology: More Facts

SUMO			
Total Terms	relations	Total Axioms	Rules
1173	353	4741	932
MILO			
Total Terms	relations	Total Axioms	Rules
1662	159	5116	1183
Domain ontologies			
Total Terms	relations	Total Axioms	Rules
17312	708	77974	2041
Total for all ontologies			
Total Terms	relations	Total Axioms	Rules
20147	1220	87831	4156

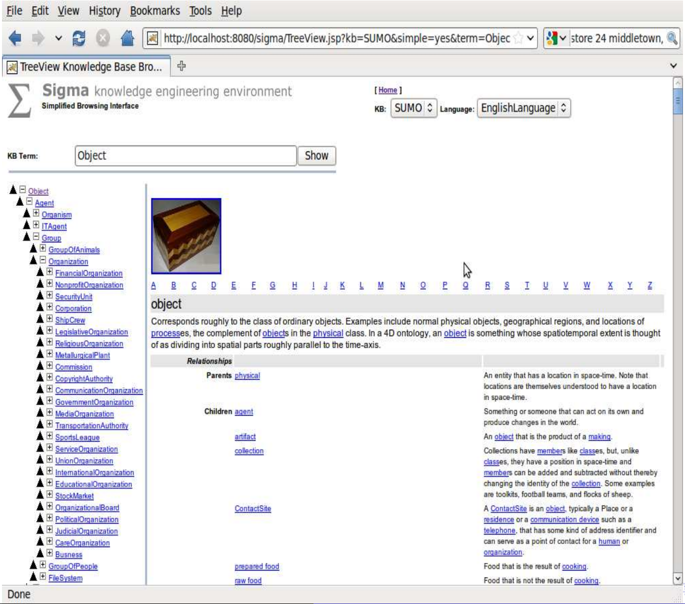
Table 1: SUMO term and axiom statistics

SIGMA: Browsing and Display

SIGMA: Browsing and Display — Hyperlinked Text



SIGMA: Browsing and Display — Hyperlinked Text



SIGMA: Analysis and Debugging

SIGMA: Analysis and Debugging — Inconsistencies

Position: Inconsistencies are just a fact of life! Finding them improves quality of SUMO. However, 'global consistency' is not pre-requisite for usefulness of SUMO.

Sigma provides both

- general purpose tools:
  - first-order theorem provers → next part
  - (new) higher-order theorem provers → ARCOE-2010 talk
- special purpose tools for error detection:
  - check for terms without root in subclass hierarchy
  - check for term with parents that are defined disjoint
  - (indicative) check for terms lacking documentation
  - (indicative) check for terms that do not occur in any rules

SIGMA: Analysis and Debugging — WordNet Mappings

Diagnostics tool for SUMO–WordNet mappings

- finds WordNet synsets without SUMO term
- finds WordNet synsets not available in current KB
- taxonomy comparison component (hierarchy mismatch not always an error!)

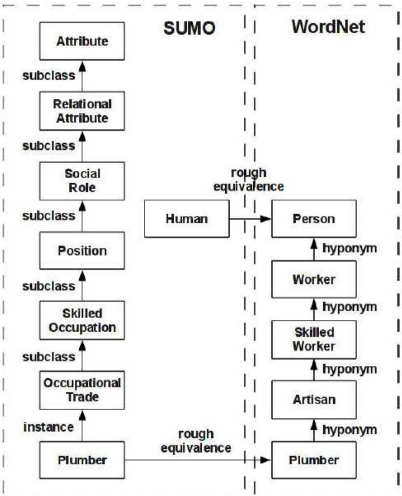


Figure 3: Comparing hierarchies of SUMO and WordNet

SIGMA: Inference

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- KIFVampire
- SiNE relevance filter
- SystemOnTPTP FOL ($>$ 40 reasoning systems)

SIGMA: Inference

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- KIFVampire
- SiNE relevance filter
- SystemOnTPTP FOL ($>$ 40 reasoning systems)

Higher-Order

- LEO-II
- SystemOnTPTP THF (soon; 6 reasoning systems)

SIGMA: Inference

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- KIFVampire
- SiNE relevance filter
- SystemOnTPTP FOL ($>$ 40 reasoning systems)

Higher-Order

- LEO-II
- SystemOnTPTP THF (soon; 6 reasoning systems)

Mainly used for

- question anwering
- analysis and debugging

SIGMA: Inference

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- KIFVampire
- SiNE relevance filter
- SystemOnTPTP FOL ($>$ 40 reasoning systems)

Higher-Order (see ARCOE-2010 talk!)

- LEO-II
- SystemOnTPTP THF (soon; 6 reasoning systems)

Mainly used for

- question anwering
- analysis and debugging

SIGMA: Inference with FO-ATPs in Question Answering

Sigma knowledge engineering environment
SystemOnTPTP Interface

[Home | Graph | Prefs] KB: SUMO Language:

(instance ?X PrimaryColor)

Local SystemOnTPTP   Remote SystemOnTPTP   System: EP---0.999
Maximum answers: 2   Query time limit: 300
TPTP Proof   IDV-Proof tree   Hyperlinked KIF
Ask   Tell

(SZS Status Theorem)
Answer 1. [definite] ?X6 = Red

1. (exists (?X6) (instance ?X6 PrimaryColor))	[Query]
2. (instance Red PrimaryColor)	[KB]
3. (instance Red PrimaryColor)	2
4. (not (instance ?X6 PrimaryColor))	1
5. (not (instance Red PrimaryColor))	4
6. True	3 5

(SZS Status Theorem)
Answer 1. [definite] ?X = Blue

1. (instance Blue PrimaryColor)	[KB]
2. (exists (?X1) (instance Blue PrimaryColor))	[Instantiated Query]
3. (not (exists (?X1) (instance Blue PrimaryColor)))	2
4. (not (instance Blue PrimaryColor))	3
5. (instance Blue PrimaryColor)	1
6. (not (instance Blue PrimaryColor))	4
7. True	6 5
8. True	7
9. True	8

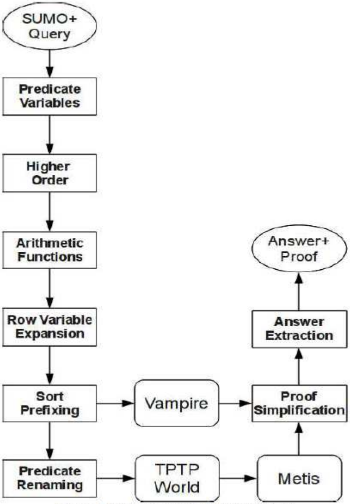


Figure 4: Sigma pre- and post-processing steps

Further Reading:

[PeaseSutcliffe, ESARLT-2007]
[TracSutcliffePease, PAAR-2008]
[PeaseEtAl., AICom, 2010]

SIGMA: Inference with FO-ATPs

Predicate Variables

```
(instance part TransitiveRelation)

(<=>
  (instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (?REL ?INST1 ?INST2)
        (?REL ?INST2 ?INST3))
      (?REL ?INST1 ?INST3))))
```

SIGMA: Inference with FO-ATPs

Predicate Variables

```
(holds instance part TransitiveRelation)

(<=>
  (holds instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (holds ?REL ?INST1 ?INST2)
        (holds ?REL ?INST2 ?INST3))
      (holds ?REL ?INST1 ?INST3))))
```

Approach A: global 'dummy' predicate

SIGMA: Inference with FO-ATPs

Predicate Variables

```
(holds instance part TransitiveRelation)

(<=>
  (holds instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (holds ?REL ?INST1 ?INST2)
        (holds ?REL ?INST2 ?INST3))
      (holds ?REL ?INST1 ?INST3))))
```

Approach A: global 'dummy' predicate

⟶ worsens prover performance (indexing of `holds`)

SIGMA: Inference with FO-ATPs

Predicate Variables

```
(instance part TransitiveRelation)

(<=>
  (instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (?REL ?INST1 ?INST2)
        (?REL ?INST2 ?INST3))
      (?REL ?INST1 ?INST3))))
```

Predicate Variables

```
(instance part TransitiveRelation)

(<=>
  (instance part TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (part ?INST1 ?INST2)
        (part ?INST2 ?INST3))
      (part ?INST1 ?INST3))))
```

Approach B: predicate instantiation

⟶ good performance

⟶ but semantics of ?REL modified

Higher-Order: Embedded Formulas

```
(believes John (likes Mary Jeff))
```

Query:

```
(believes John (likes Mary ?X))
```

Higher-Order: Embedded Formulas

```
(believes John '(likes Mary Jeff)
```

Query:

```
(believes John '(likes Mary ?X))
```

Approach: Quoting of embedded formulas

SIGMA: Inference with FO-ATPs

Higher-Order: Embedded Formulas

```
(believes John
  '(and
    (likes Mary Jeff)
    (likes Bill Sue)))
```

Query:

```
(believes John '(likes Mary ?X))
```

Approach: Quoting of embedded formulas

⟶ connectives in embedded formulas loose their semantics

Higher-Order: Embedded Formulas

```
(believes John
  '(and
    (likes Mary Jeff)
    (likes Bill Sue)))
```

Query:

```
(believes John '(likes Mary ?X))
```

Approach: Quoting of embedded formulas

⟶ connectives in embedded formulas loose their semantics

⟶ solution with higher-order provers proposed

[BenzmüllerPease, PAAR-2010],[BenzmüllerPease, ARCOE-2010]

SIGMA: Inference with FO-ATPs

Arithmetic Expressions

```
(<=>
  (lessThanOrEqualTo ?NUMBER1 ?NUMBER2)
  (or
    (equal ?NUMBER1 ?NUMBER2)
    (lessThan ?NUMBER1 ?NUMBER2)))
```

Arithmetic Expressions

```
(<=>
  $lesseq(?NUMBER1,?NUMBER2)
  (or
    (?NUMBER1 = ?NUMBER2)
    $less(?NUMBER1,?NUMBER2)))
```

Approach: Translate into native symbols as required by provers

SIGMA: Inference with FO-ATPs

Arithmetic Expressions

```
(<=>
  $lesseq(?NUMBER1,?NUMBER2)
  (or
    (?NUMBER1 = ?NUMBER2)
    $less(?NUMBER1,?NUMBER2)))
```

Approach: Translate into native symbols as required by provers
⟶ more and more FO-ATPs provide support for basic arithmetic

SIGMA: Inference with FO-ATPs

Row variables

```
(=>
  (and
    (subrelation ?REL1 ?REL2)
    (?REL1 @ROW))
  (?REL2 @ROW))
```

Row variables

```
(=>
  (and
    (subrelation ?REL1 ?REL2)
    (?REL1 ?ROW1))
  (?REL2 ?ROW1))

(=>
  (and
    (subrelation ?REL1 ?REL2)
    (?REL1 ?ROW1 ?ROW2))
  (?REL2 ?ROW1 ?ROW2))
```

Approach: expand up to maximum arity of 7

SIGMA: Inference with FO-ATPs

Row variables

```
(=>
  (and
    (subrelation ?REL1 ?REL2)
    (?REL1 ?ROW1))
  (?REL2 ?ROW1))

(=>
  (and
    (subrelation ?REL1 ?REL2)
    (?REL1 ?ROW1 ?ROW2))
  (?REL2 ?ROW1 ?ROW2))
```

Approach: expand up to maximum arity of 7

⟶ few axioms with 2 row variables (max. 49 expanded axioms)

⟶ limits semantics of row variables

Sort Prefixing

```
(=>
  (and
    (instance ?TRANSFER Transfer)
    (agent ?TRANSFER ?AGENT)
    (patient ?TRANSFER ?PATIENT))
  (not
    (equal ?AGENT ?PATIENT)))
```

SIGMA: Inference with FO-ATPs

Sort Prefixing

```
(=>
  (and
    (instance ?TRANSFER Transfer)
    (agent ?TRANSFER ?AGENT)
    (patient ?TRANSFER ?PATIENT))
  (not
    (equal ?AGENT ?PATIENT)))
```

```
(domain agent 2 Agent)
(domain patient 2 Object)
```

Sort Prefixing

```
(=>
  (and
    (instance ?AGENT Agent)
    (instance ?PATIENT Object))
  (=>
    (and
      (instance ?TRANSFER Transfer)
      (agent ?TRANSFER ?AGENT)
      (patient ?TRANSFER ?PATIENT))
    (not
      (equal ?AGENT ?PATIENT))))
```

Approach: extract SUMO type information and add

Predicate Renaming

```
(instance instance BinaryPredicate)
```

Predicate Renaming

```
(p_instance t_instance BinaryPredicate)
```

Approach: decouple predicates by renaming

Predicate Renaming

Approach: decouple predicates by renaming
$\longrightarrow$ information loss for some predicates; here for `p_instance`

SIGMA: Mapping, Merging and Translation

SIGMA: Mapping, Merging and Translation

Translation

- Sigma provides (some restricted) support for reading and writing OWL format
  - has been be applied to SUMO:
    $\longrightarrow$ lossy translation
    $\longrightarrow$ bulk of axioms become informative comments for users
- Sigma supports translations to
  - TPTP FOL (classical first-order logic)
  - TPTP THF (classical higher-order logic)
- export to PROLOG
- export (prototype) to SQL
- SUO-KIF is expressive: similar translation to and from other less expressive formats should be possible

Mapping and Merging

- simple algorithm developed
  - used for creating an initial alignment with the lightweight Open Biomedical Onologies OBO
  - however, bulk of effort still spent by human in selecting accurate matches (due to massive number of false positives when trying it fully automatically)

Conclusion

- Sigma is a toolkit and testbed for (rich) ontology development and application
- Sigma has co-evolved with SUMO
- Sigma is open source: `www.ontologyportal.org`

Ongoing and future work

- package a formal release (no formal release for >2 years)
- semantics for higher-order aspects of SUMO
- integration of higher-order reasoning systems
- support layer in Sigma for integrated, heterogeneous reasoning support
- inbuilt first-order theorem prover