OWL Simplified English*

Richard Power
Open University, UK

* A finite-state language for ontology editing

Semantic Web Authoring Tool

(EPSRC 2009-2012)

Open University (Department of Computing)

Richard Power
Sandra Williams
Allan Third
Tu Anh Nguyen

Manchester University (School of Computer Science)

Robert Stevens
Alan Rector
Fennie Liang

Sussex University (Department of Informatics)

Donia Scott

Objectives

Theoretical

Clarify relationship of formal languages (OWL) to natural languages (English)

Practical

Develop tools for viewing and editing OWL ontologies in natural language

SWAT Natural Language Tools

http://swat.open.ac.uk/tools/

SWAT
SemanticWebAuthoringTool

SWAT Natural Language Tools

Analyse your OWL ontology, build a lexicon from it, or convert it into English sentences or definition paragraphs.

Select the ontology file

Choose File no file selected

File format should be OWL/XML (.xml) or OWL/RDF (.owl).

Select the output format

- (•) Alphabetical English glossary (class, individual and property definitions)
- ( ) English sentences (one sentence per OWL axiom)
- ( ) Prolog terms (translate OWL to Prolog)
- ( ) Lexicon (lexical entries for class, individual and property names)
- ( ) Axiom patterns (frequency counts)
- ( ) OWL/XML, with verbalisations as "SWATDescription" annotations.

Explanations of these outputs are given here.

ADULT (class)

Examples Elderlies, and drivers are adults.

Distinctions No adult is a young.

ANIMAL (class)

Description An animal eats a thing.
If X has as pet Y then Y is an animal.
If X eats Y then X is an animal.

Examples The following are animals: tigers, sheep and people, and so on (6 items in total).

ANIMAL LOVER (class)

Definition An animal lover is defined as a person that has as pet at least three things.

BICYCLE (class)

Typology A bicycle is a vehicle.

BONE (class)

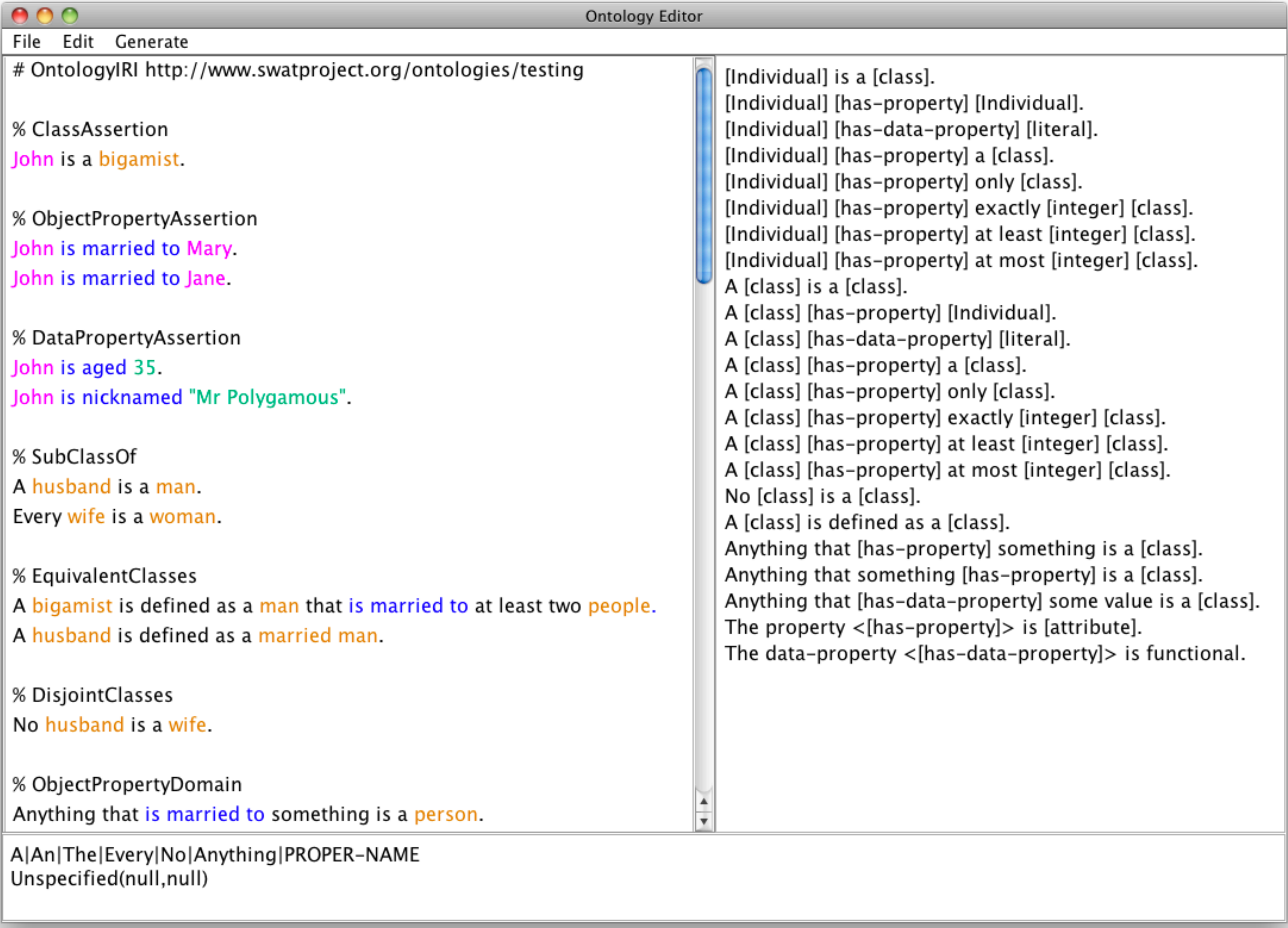
Other No information.

BRAIN (class)

Other No information.

BROADSHEET (class)

Typology A broadsheet is a newspaper.


Ontology Editor
File Edit Generate
OntologyIRI http://www.swatproject.org/ontologies/testing
% ClassAssertion
John is a bigamist.
% ObjectPropertyAssertion
John is married to Mary.
John is married to Jane.
% DataPropertyAssertion
John is aged 35.
John is nicknamed "Mr Polygamous".
% SubClassOf
A husband is a man.
Every wife is a woman.
% EquivalentClasses
A bigamist is defined as a man that is married to at least two people.
A husband is defined as a married man.
% DisjointClasses
No husband is a wife.
% ObjectPropertyDomain
Anything that is married to something is a person.
[Individual] is a [class].
[Individual] [has-property] [Individual].
[Individual] [has-data-property] [literal].
[Individual] [has-property] a [class].
[Individual] [has-property] only [class].
[Individual] [has-property] exactly [integer] [class].
[Individual] [has-property] at least [integer] [class].
[Individual] [has-property] at most [integer] [class].
A [class] is a [class].
A [class] [has-property] [Individual].
A [class] [has-data-property] [literal].
A [class] [has-property] a [class].
A [class] [has-property] only [class].
A [class] [has-property] exactly [integer] [class].
A [class] [has-property] at least [integer] [class].
A [class] [has-property] at most [integer] [class].
No [class] is a [class].
A [class] is defined as a [class].
Anything that [has-property] something is a [class].
Anything that something [has-property] is a [class].
Anything that [has-data-property] some value is a [class].
The property <[has-property]> is [attribute].
The data-property <[has-data-property]> is functional.
A|An|The|Every|No|Anything|PROPER-NAME
Unspecified(null,null)

Download editing tool

`http://mcs.open.ac.uk/rp3242/editor/`

Requires Java runtime environment

Outline

- Motivation
- Demonstration
- Language
- Coverage
- Conclusion

Previous work

Attempto Controlled English (ACE)
Sydney OWL Syntax (SOS)
Rabbit

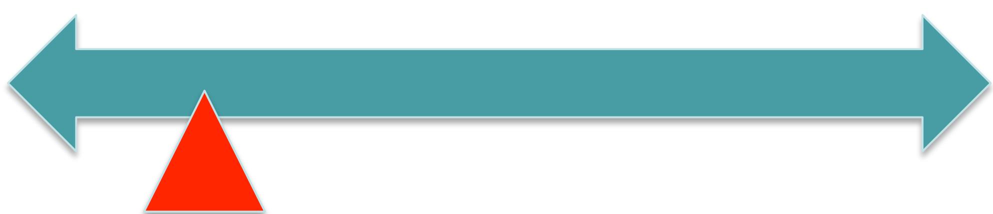
Controlled Natural Languages

ACE Wiki
ROO (Rabbit to OWL Ontology construction)
RoundTrip Ontology Authoring

Ontology Editing Tools

OWL Simplified English

- Very simple rules for forming sentences
- Little or no effort required to build lexicon
- Disallows structurally ambiguous sentences
- Can be interpreted by finite-state transducer
- Coverage limited in theory, adequate in practice

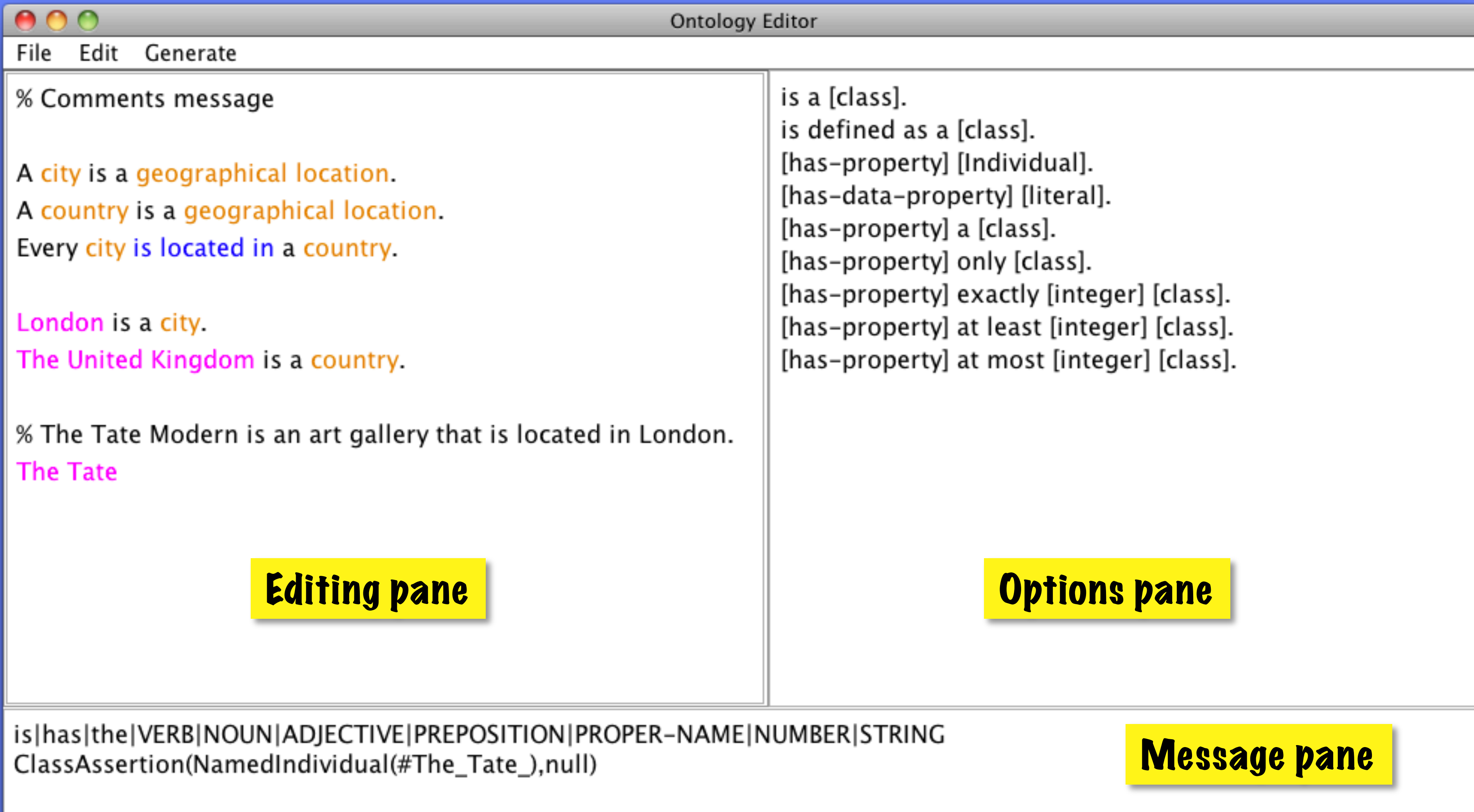
SIMPLICITY  COVERAGE

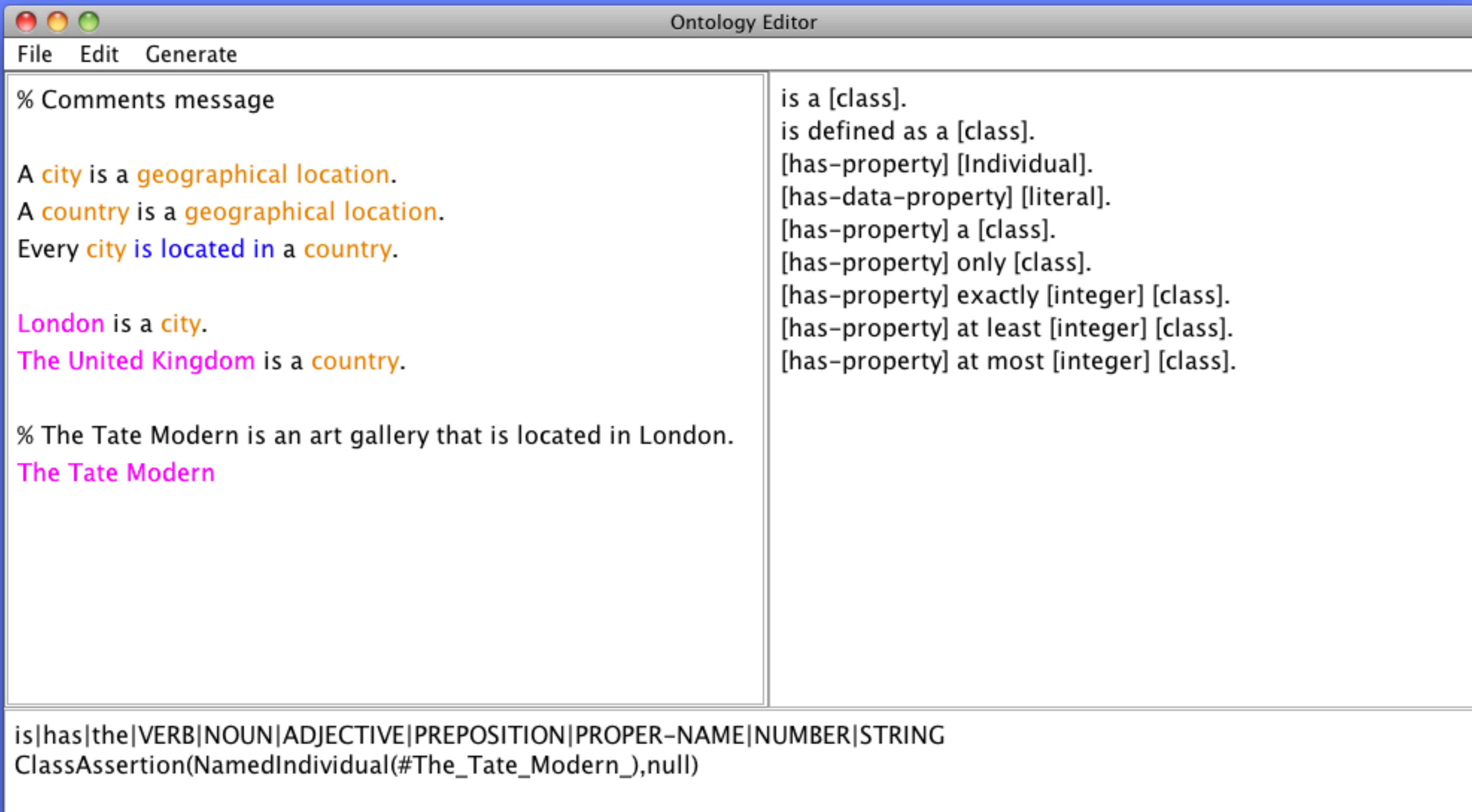
Editing tool

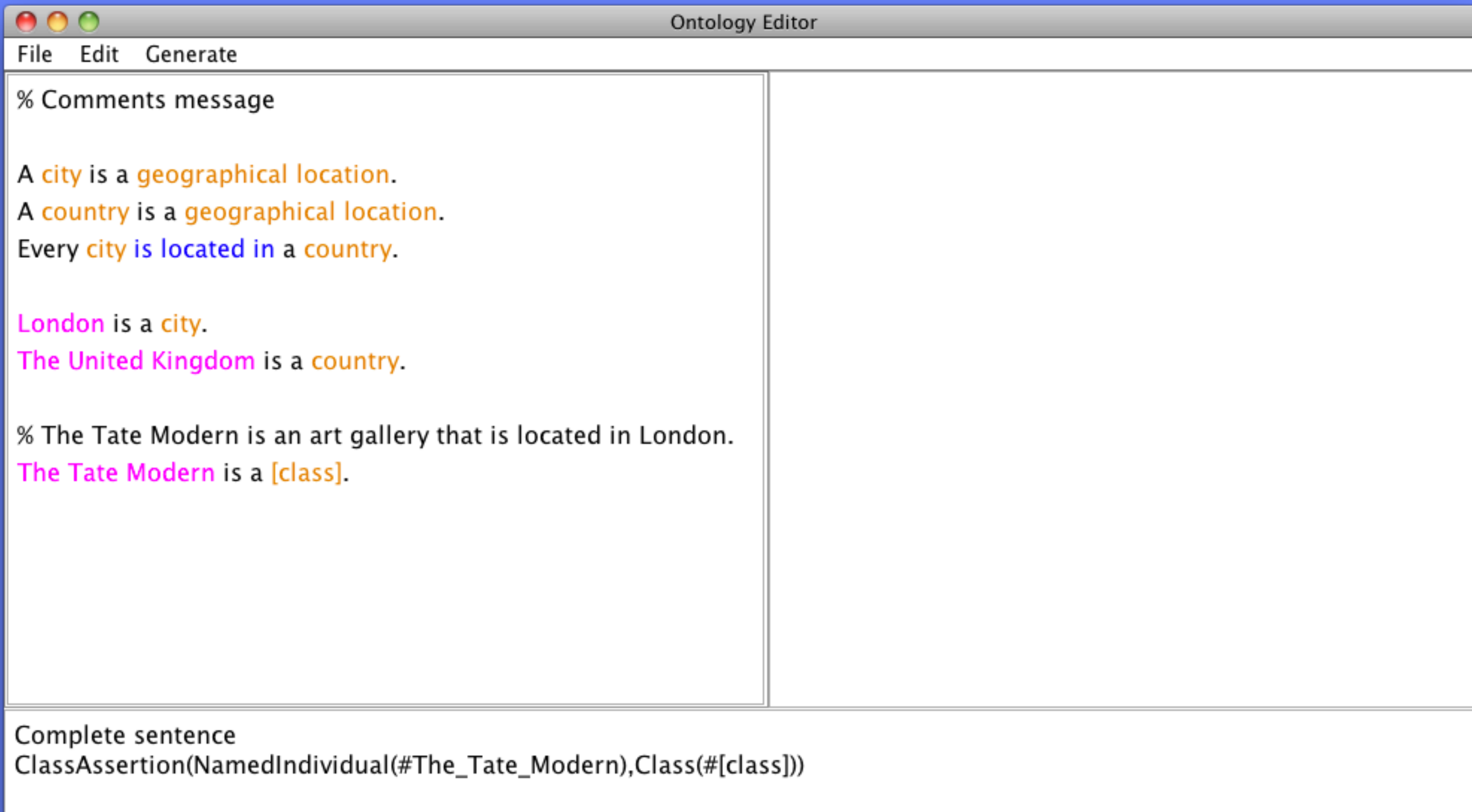
- User edits text as in predictive authoring
- Patterns for a complete sentence are offered as in WYSIWYM
- Patterns contain anchors for entity names (individual, class, property) for which options are computed from the current text
- Efficient implementation is much easier if the grammar is finite-state

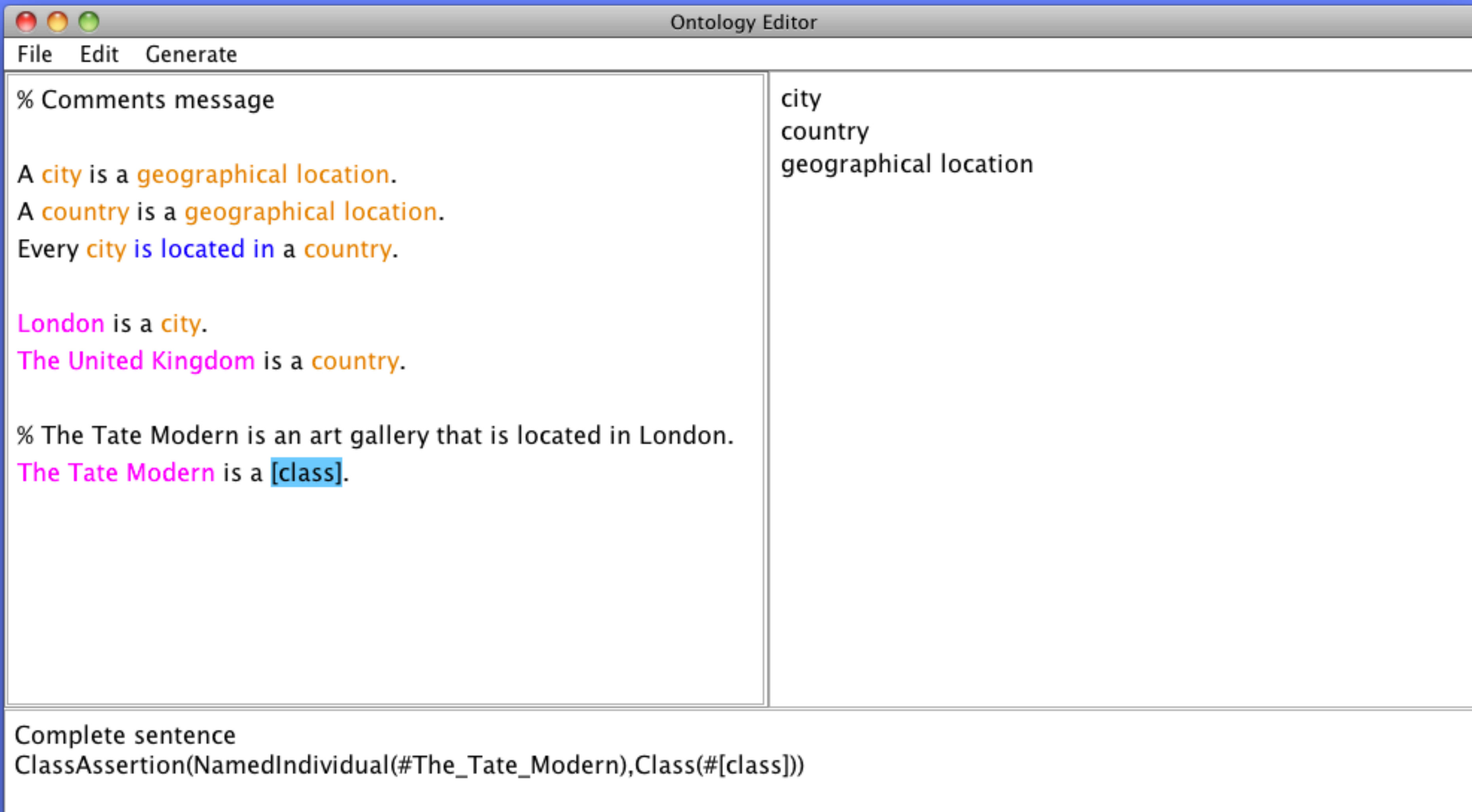
Outline

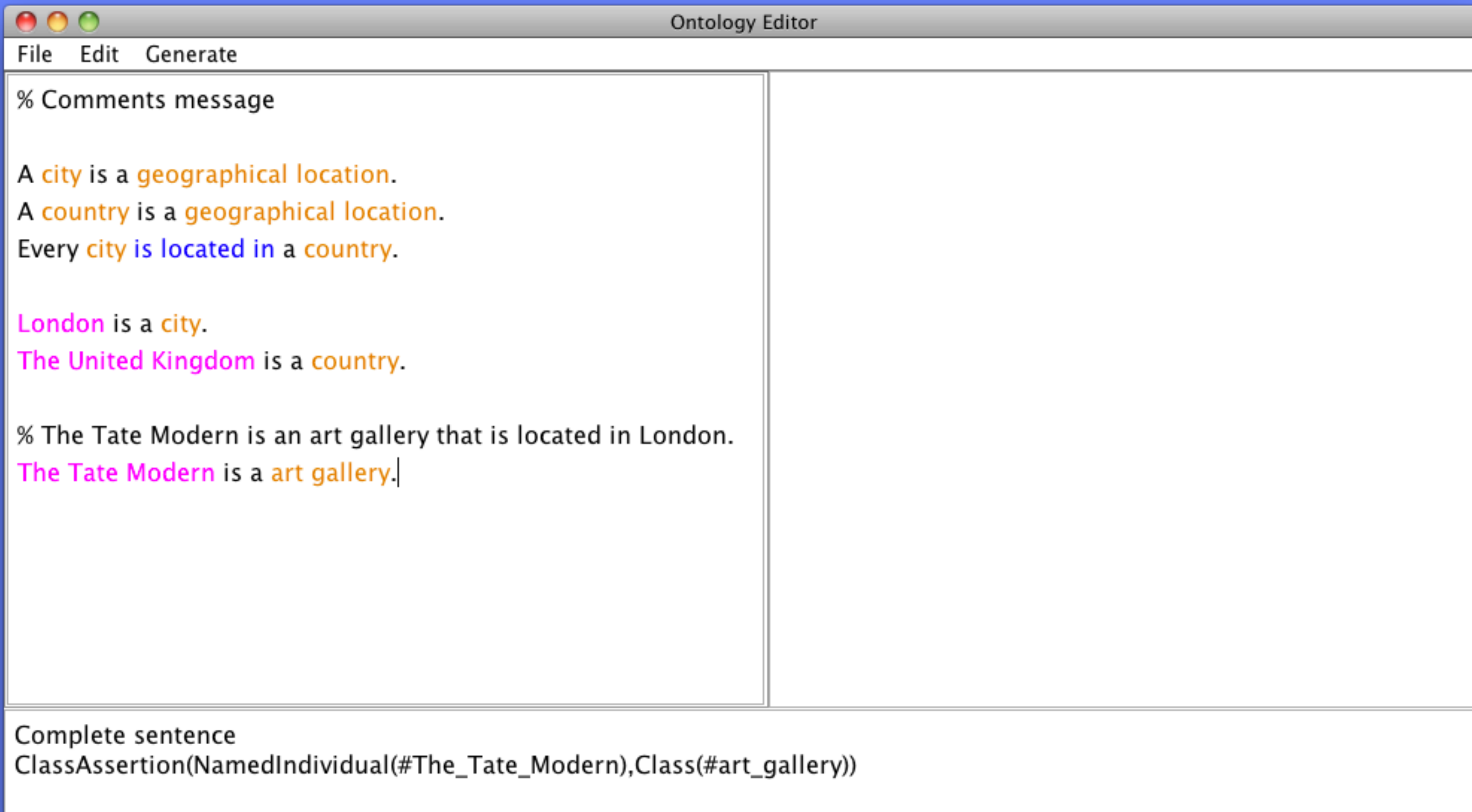
- Motivation
- Demonstration
- Language
- Coverage
- Conclusion

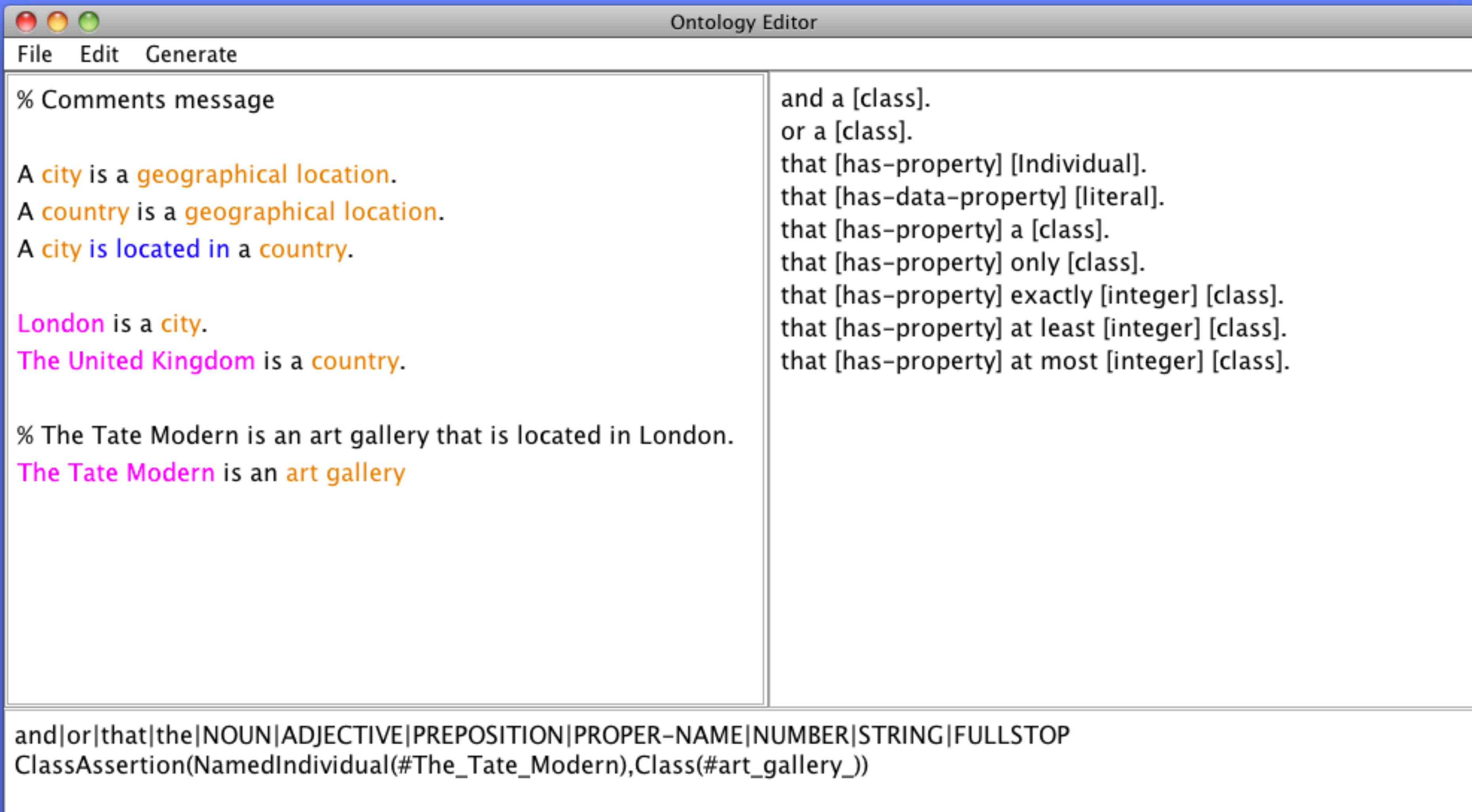

Ontology Editor
File
Edit
Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
Every city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate
Editing pane
is a [class].
is defined as a [class].
[has-property] [Individual].
[has-data-property] [literal].
[has-property] a [class].
[has-property] only [class].
[has-property] exactly [integer] [class].
[has-property] at least [integer] [class].
[has-property] at most [integer] [class].
Options pane
is|has|the|VERB|NOUN|ADJECTIVE|PREPOSITION|PROPER-NAME|NUMBER|STRING
ClassAssertion(NamedIndividual(#The_Tate_),null)
Message pane

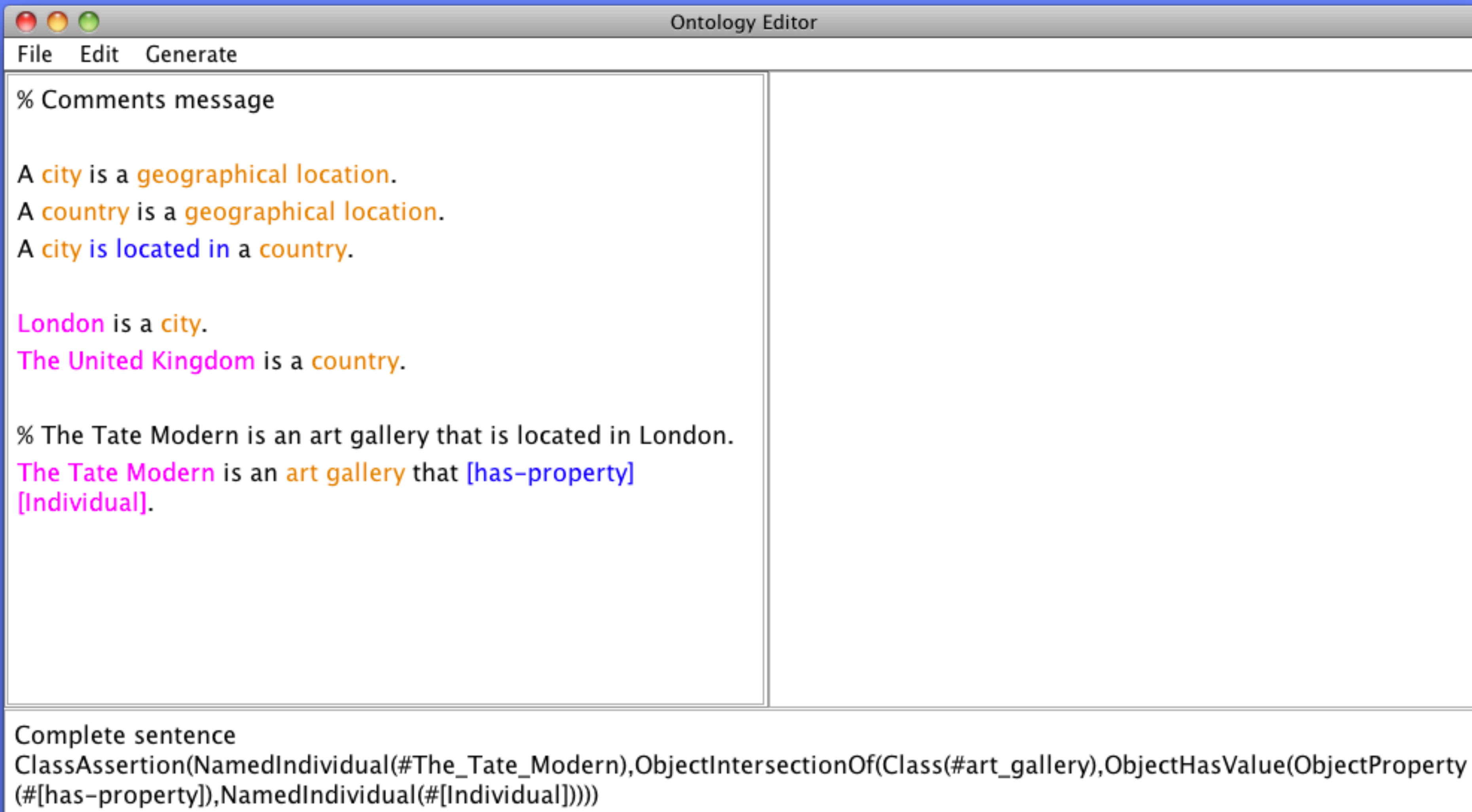

Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
Every city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern
is a [class].
is defined as a [class].
[has-property] [Individual].
[has-data-property] [literal].
[has-property] a [class].
[has-property] only [class].
[has-property] exactly [integer] [class].
[has-property] at least [integer] [class].
[has-property] at most [integer] [class].
is|has|the|VERB|NOUN|ADJECTIVE|PREPOSITION|PROPER-NAME|NUMBER|STRING
ClassAssertion(NamedIndividual(#The_Tate_Modern_),null)

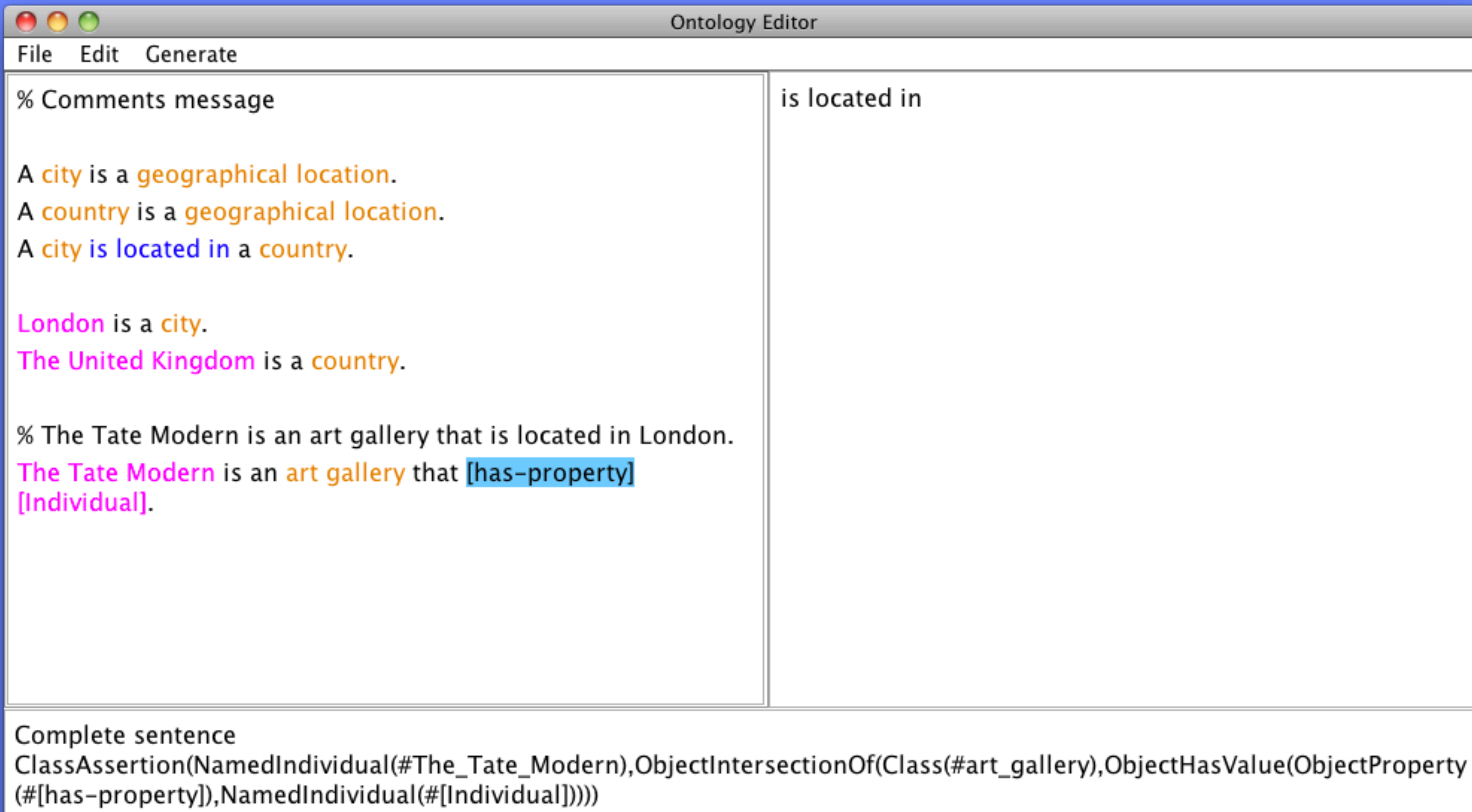

Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
Every city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is a [class].
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),Class(#[class]))

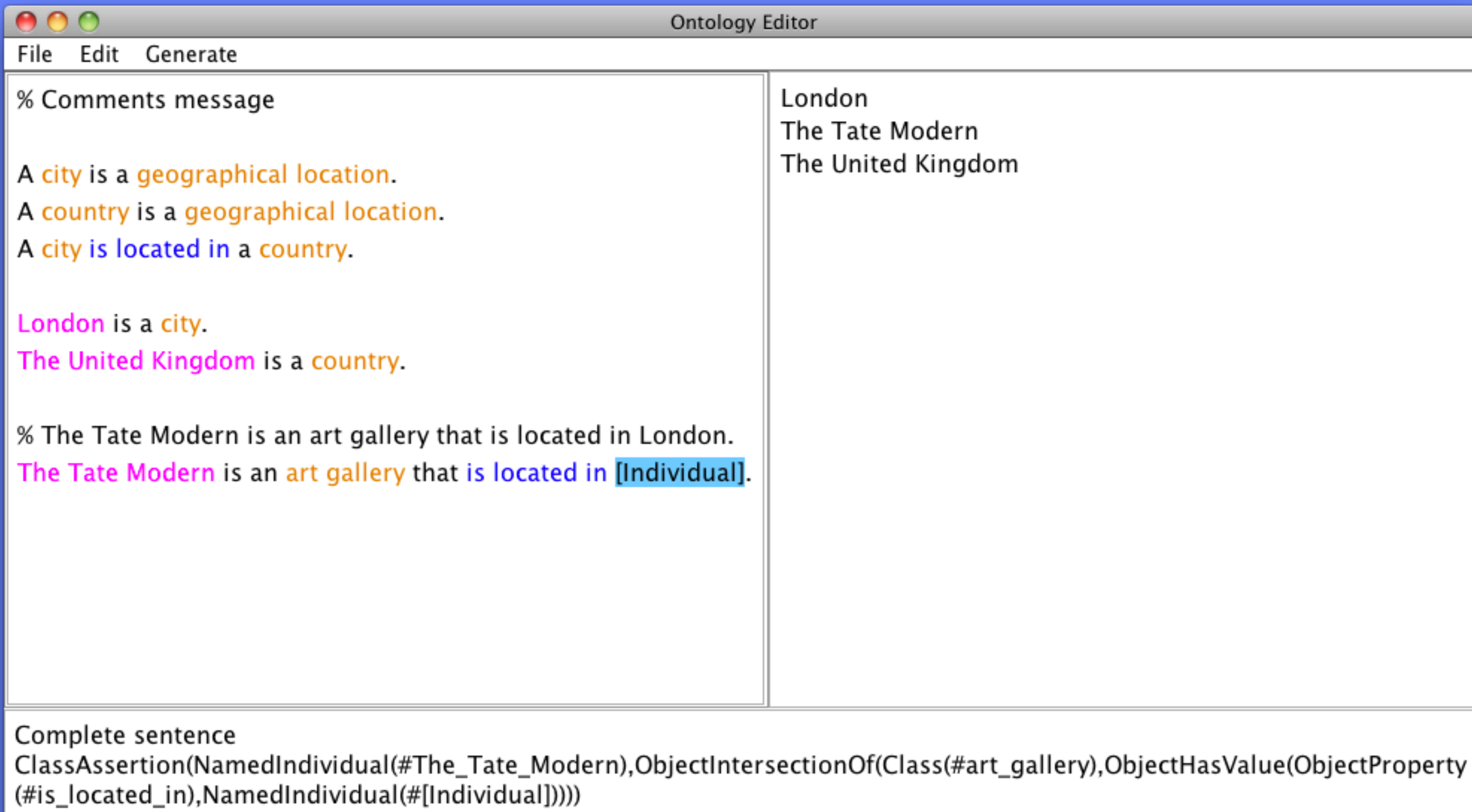

Ontology Editor
File
Edit
Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
Every city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is a [class].
city
country
geographical location
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),Class(#[class]))

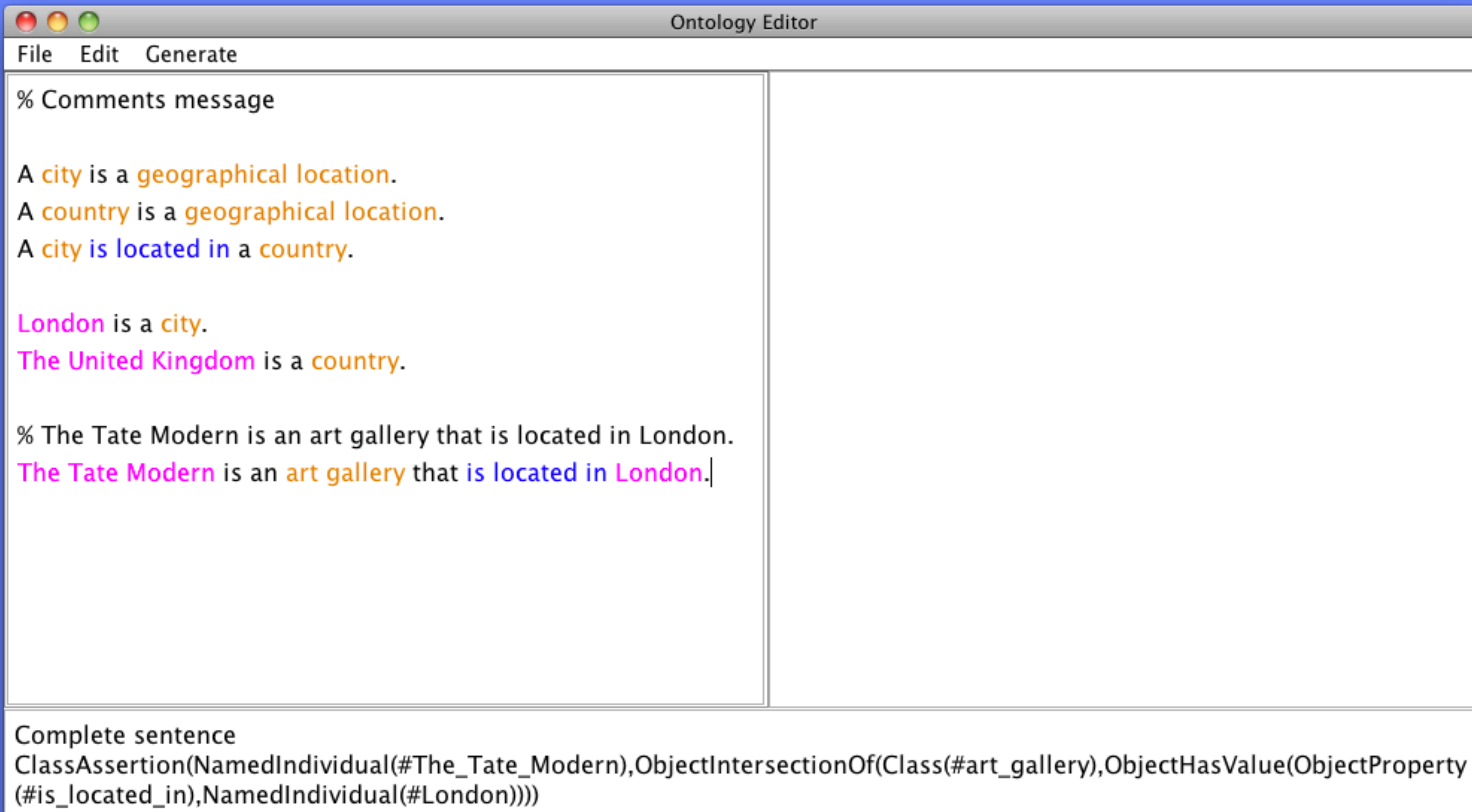

Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
Every city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is a art gallery.
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),Class(#art_gallery))


Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
A city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is an art gallery
and a [class].
or a [class].
that [has-property] [Individual].
that [has-data-property] [literal].
that [has-property] a [class].
that [has-property] only [class].
that [has-property] exactly [integer] [class].
that [has-property] at least [integer] [class].
that [has-property] at most [integer] [class].
and|or|that|the|NOUN|ADJECTIVE|PREPOSITION|PROPER-NAME|NUMBER|STRING|FULLSTOP
ClassAssertion(NamedIndividual(#The_Tate_Modern),Class(#art_gallery_))


Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
A city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is an art gallery that [has-property] [Individual].
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),ObjectIntersectionOf(Class(#art_gallery),ObjectHasValue(ObjectProperty(#[has-property]),NamedIndividual(#[Individual]))))


Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
A city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is an art gallery that [has-property] [Individual].
is located in
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),ObjectIntersectionOf(Class(#art_gallery),ObjectHasValue(ObjectProperty(#[has-property]),NamedIndividual(#[Individual]))))


Ontology Editor
File Edit Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
A city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is an art gallery that is located in [Individual].
London
The Tate Modern
The United Kingdom
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),ObjectIntersectionOf(Class(#art_gallery),ObjectHasValue(ObjectProperty
(#is_located_in),NamedIndividual(#[Individual]))))


Ontology Editor
File
Edit
Generate
% Comments message
A city is a geographical location.
A country is a geographical location.
A city is located in a country.
London is a city.
The United Kingdom is a country.
% The Tate Modern is an art gallery that is located in London.
The Tate Modern is an art gallery that is located in London.
Complete sentence
ClassAssertion(NamedIndividual(#The_Tate_Modern),ObjectIntersectionOf(Class(#art_gallery),ObjectHasValue(ObjectProperty
(#is_located_in),NamedIndividual(#London))))

Outline

- Motivation
- Demonstration
- Language
- Coverage
- Conclusion

Axiom in OSE and OFS

London is a city that is capital of the United Kingdom and is divided into at least 30 boroughs.

```
ClassAssertion(Class(#London),
  ObjectIntersectionOf(Class(#city),
    ObjectHasValue(ObjectProperty(#capitalOf),
      NamedIndividual(#UK))
    ObjectMinCardinality(30,
      ObjectProperty(#dividedInto),
      Class(#borough))))
```

Restricted words

London is a city that is capital of the United Kingdom and is divided into at least 30 boroughs.

ENTITY NAMES

- Individual name
- Class name
- Property name

a/an, and, or, that, not, anything, something, every, no, least, most, only, exactly, ...

Some words are used only as scaffolding, and cannot be included in an entity name

Word categories

Syntactic sugar	every, no, a/an, and, or, that, ...
Number	two, 365, 3.14, ...
String	“Pride and Prejudice”, “XY123”, ...
Verb (present)	is, has, takes, participates, ...
Proper noun	John, X23, London, ...
Preposition	of, by, in, from, ...
Noun/other	person, taken, yellow, slowly, ...

How words are categorised

Syntactic sugar	a/an, and,...	Listed in program
Number	two, 365, ...	Number words, digits
String	“XY123”, ...	Double quotes
Verb (present)	takes, ...	Listed by USER
Proper noun	John, ...	Upper-case letter
Preposition	of, by, ...	Listed in program
Noun/other	person, ...	Lower-case letter

Entity names

Entity	Opening	Continuation
Individual	**Proper noun, ‘the’**	**Proper noun, ‘the’, Number, String, Preposition, Noun/other**
Class	**Proper name, ‘the’, Number, String, Preposition, Noun/other**	**Proper name, ‘the’, Number, String, Preposition, Noun/other**
Property	**‘is’, ‘has’, Verb (present)**	**Noun/other, Preposition**
Literal	**Number, String**	

Reason for these rules

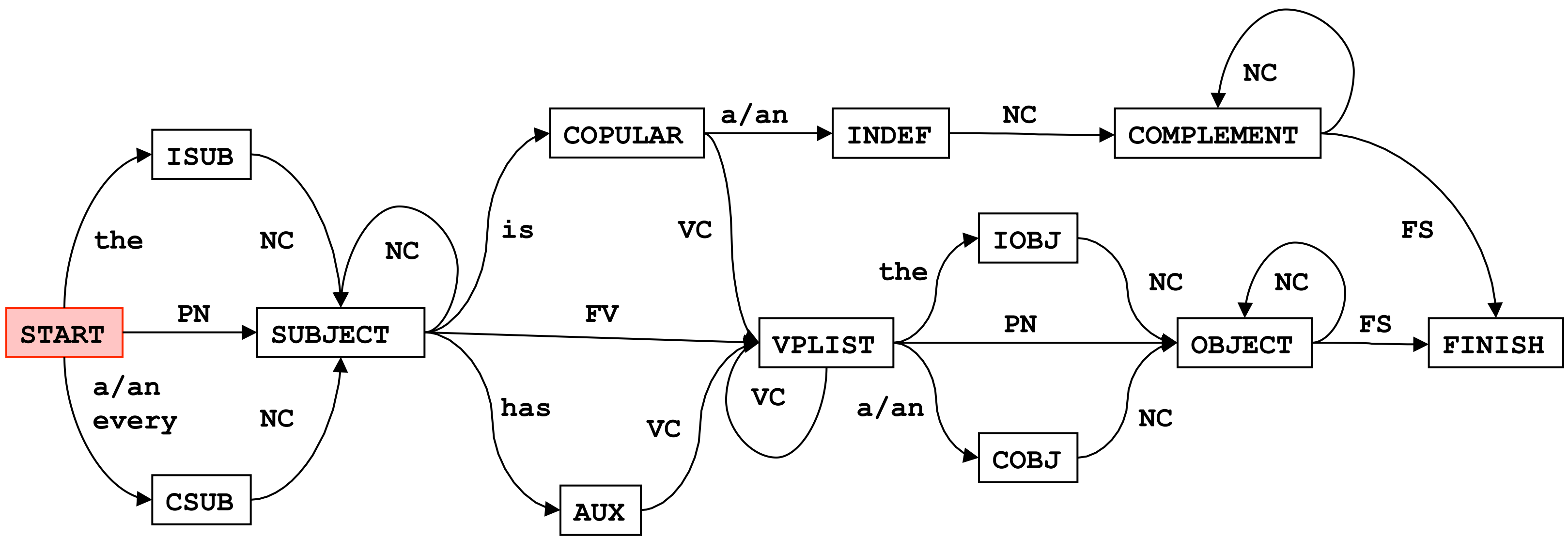
The author is not required to define names for individuals, classes and properties in advance, so the system must infer when they start and end.

`The queen is a woman that lives in Buckingham Palace and is married to a Greek that is named "Phillip".`

■ (purple)	Individual
■ (orange)	Class
■ (blue)	Property
■ (green)	Literal

1

PN Proper noun
NC Noun-phrase continuation (PN, the, Number, String, Prep, Noun/other)
VC Verb-phrase continuation (Prep, Noun/other)
FV Tensed verb
FS Full stop

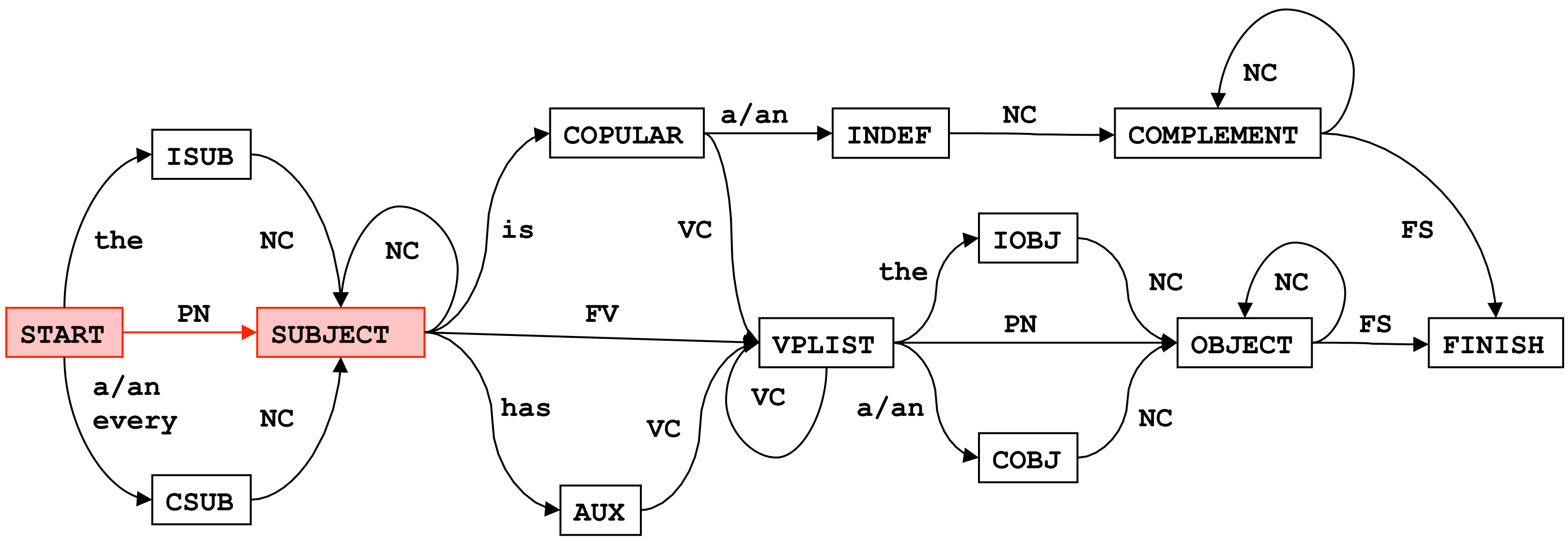


Tony Blair is married to a lawyer.

Unspecified(null,null)

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

2

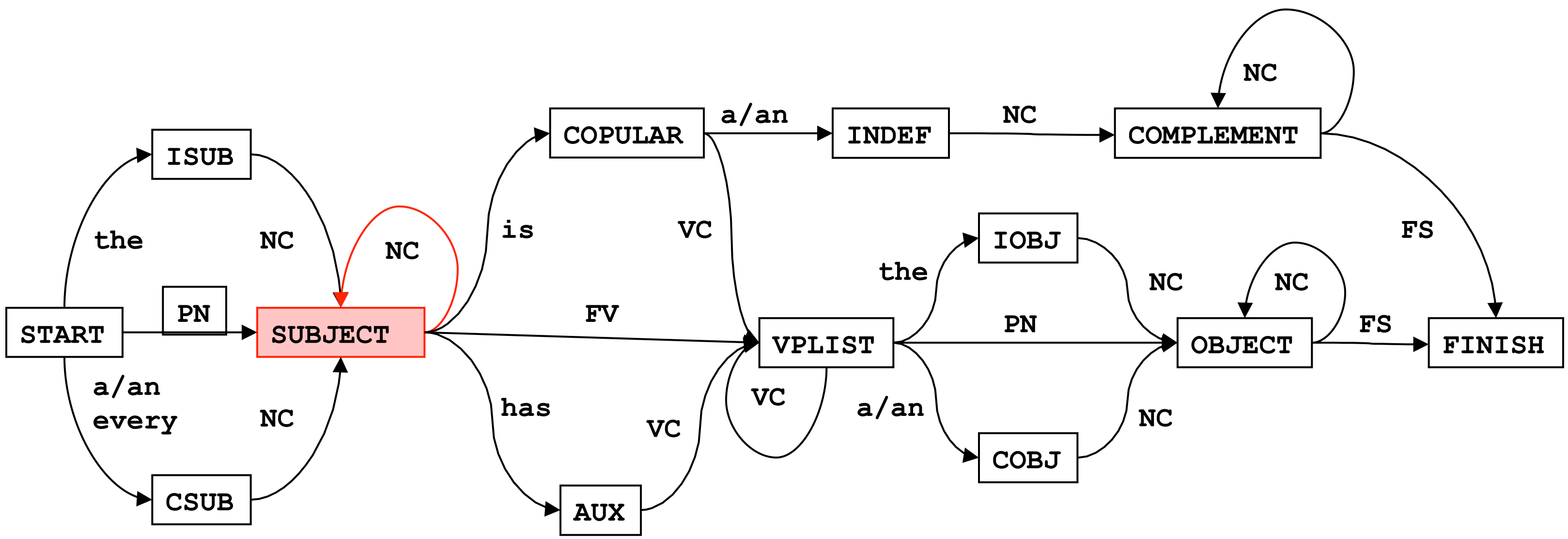


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony),null)

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

3

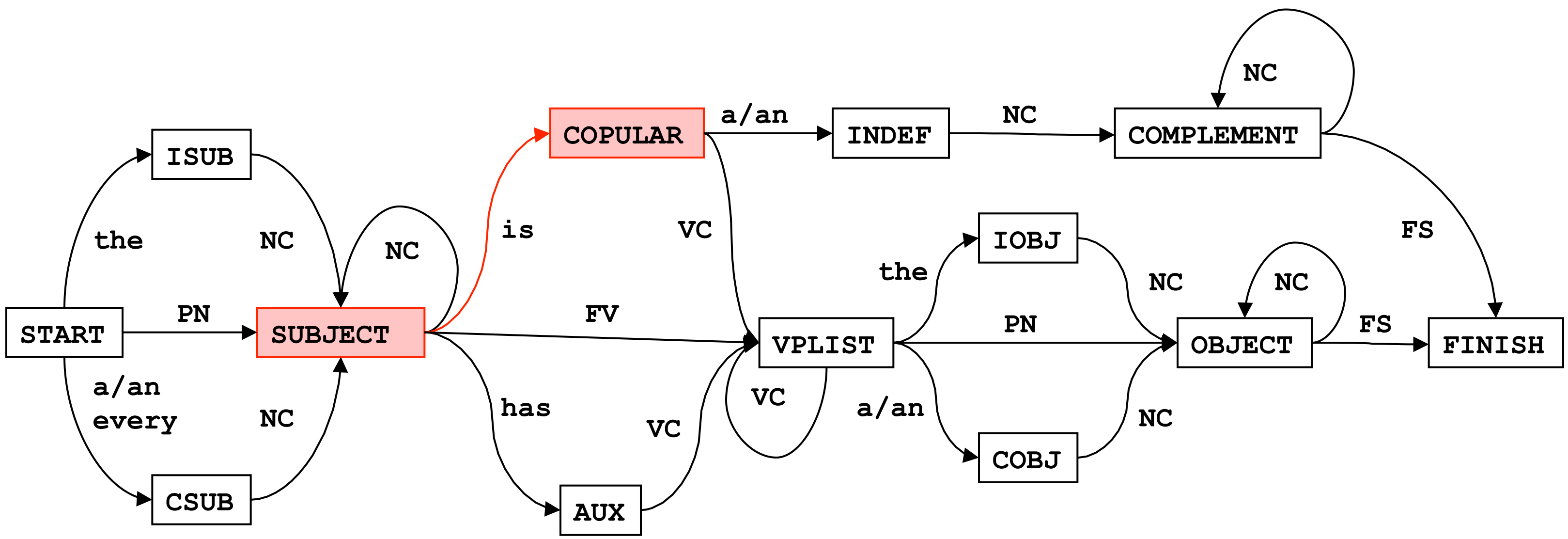


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),null)

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

4

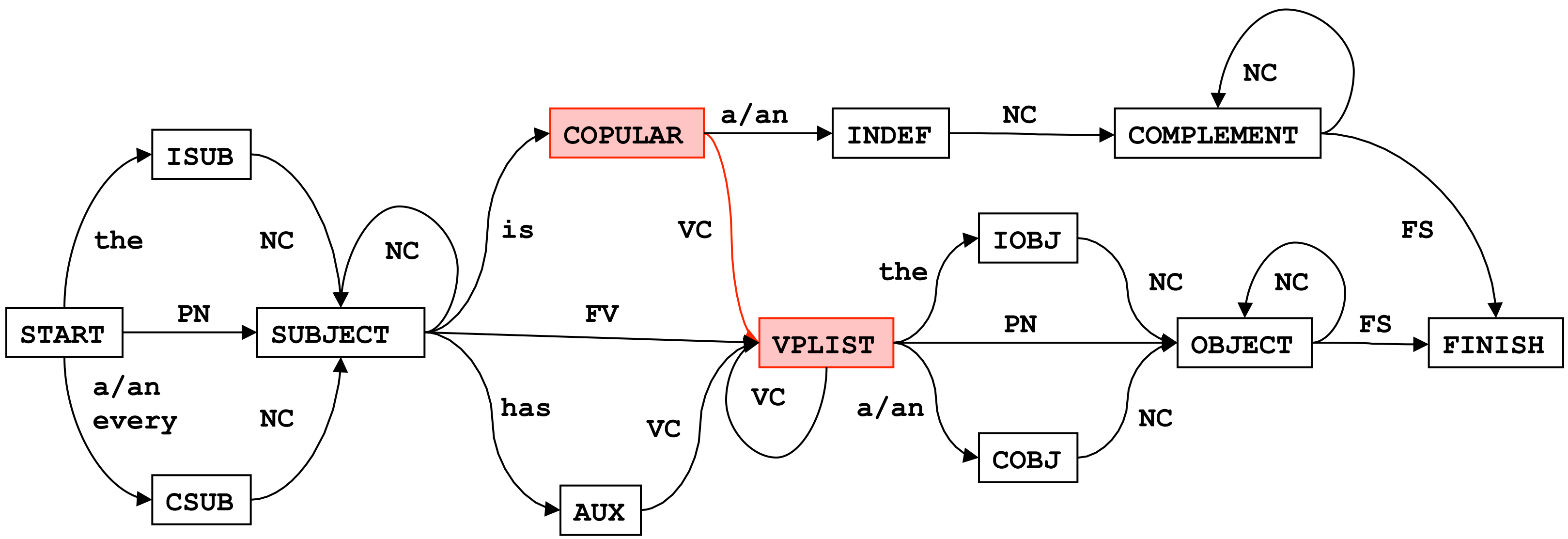


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),null)

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

5

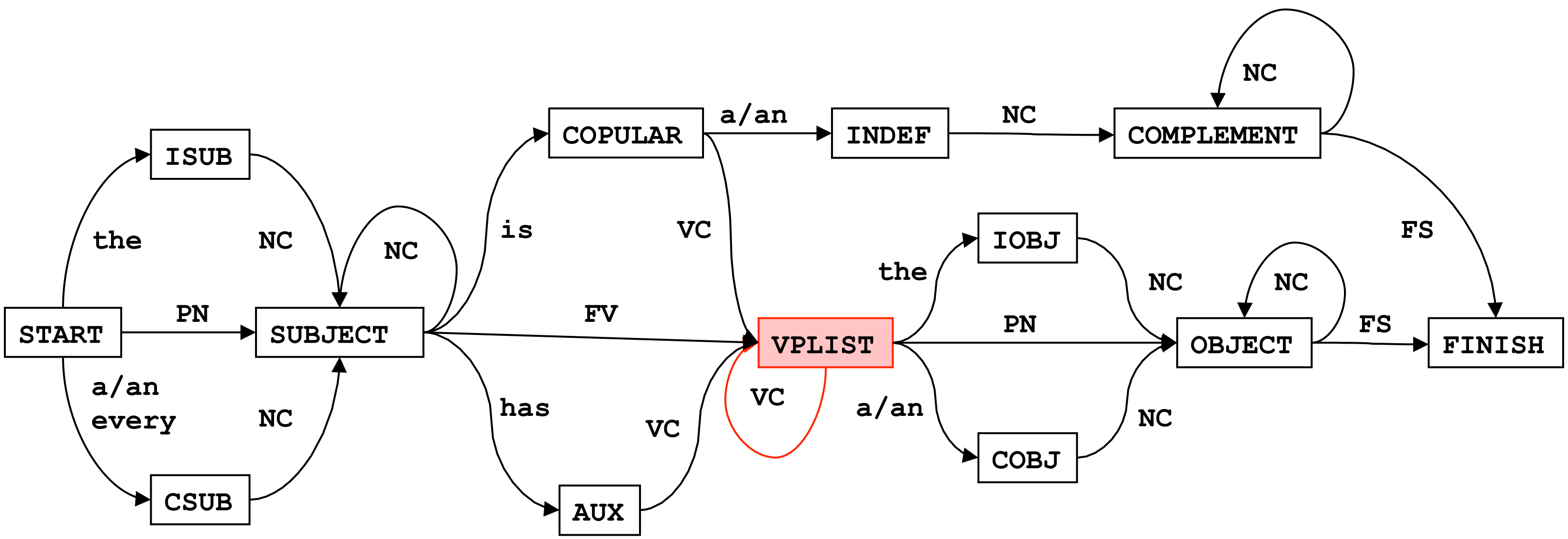


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),
UnspecifiedRestriction(ObjectProperty(#is_married),null))

6

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

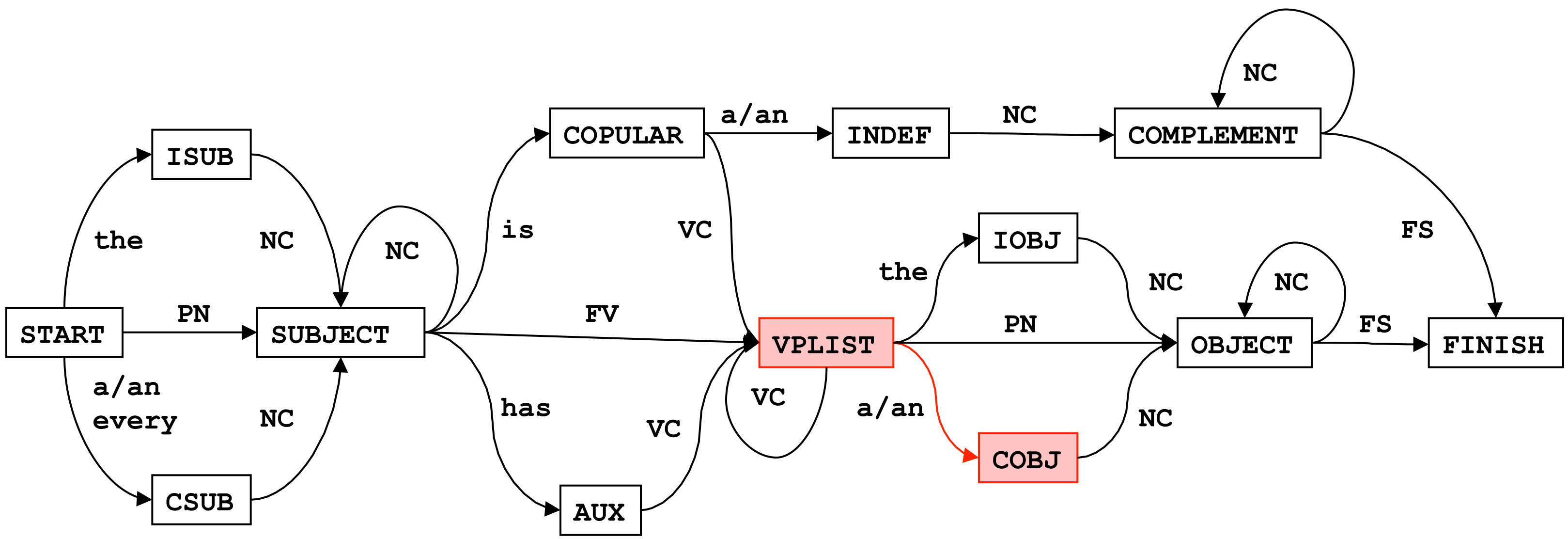


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),
UnspecifiedRestriction(ObjectProperty(#is_married_to),null))

7

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

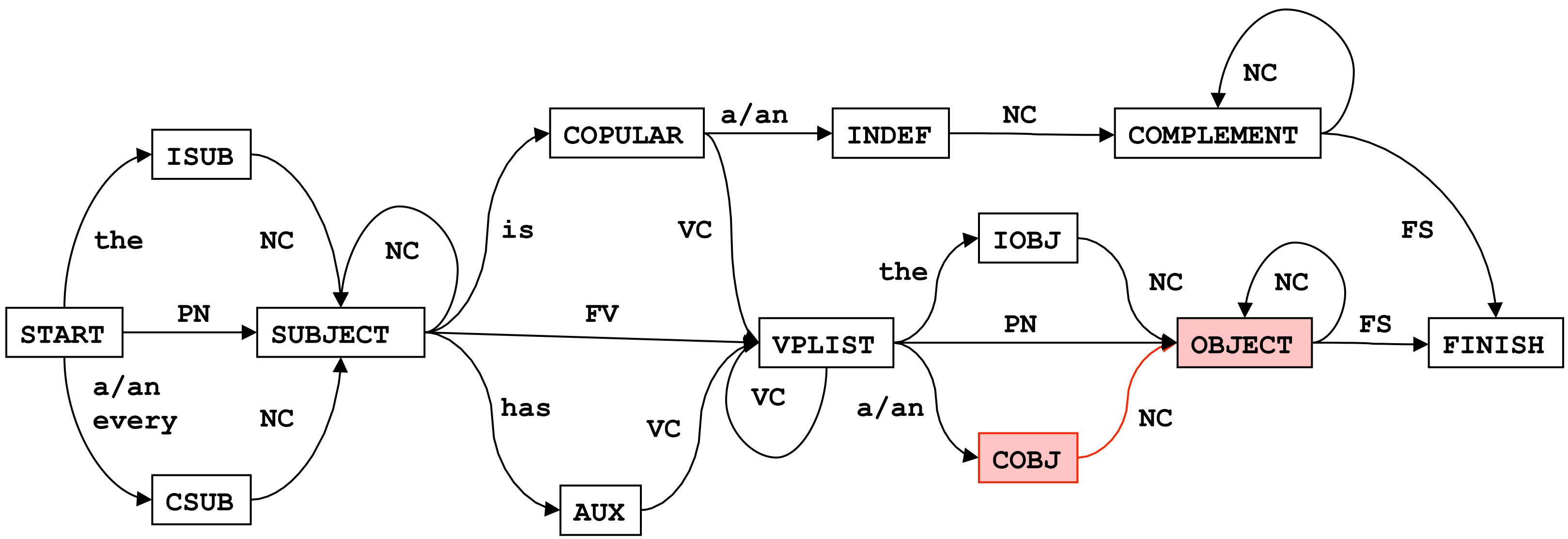


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),
ObjectSomeValuesFrom(ObjectProperty(#is_married_to),Class()))

8

PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop

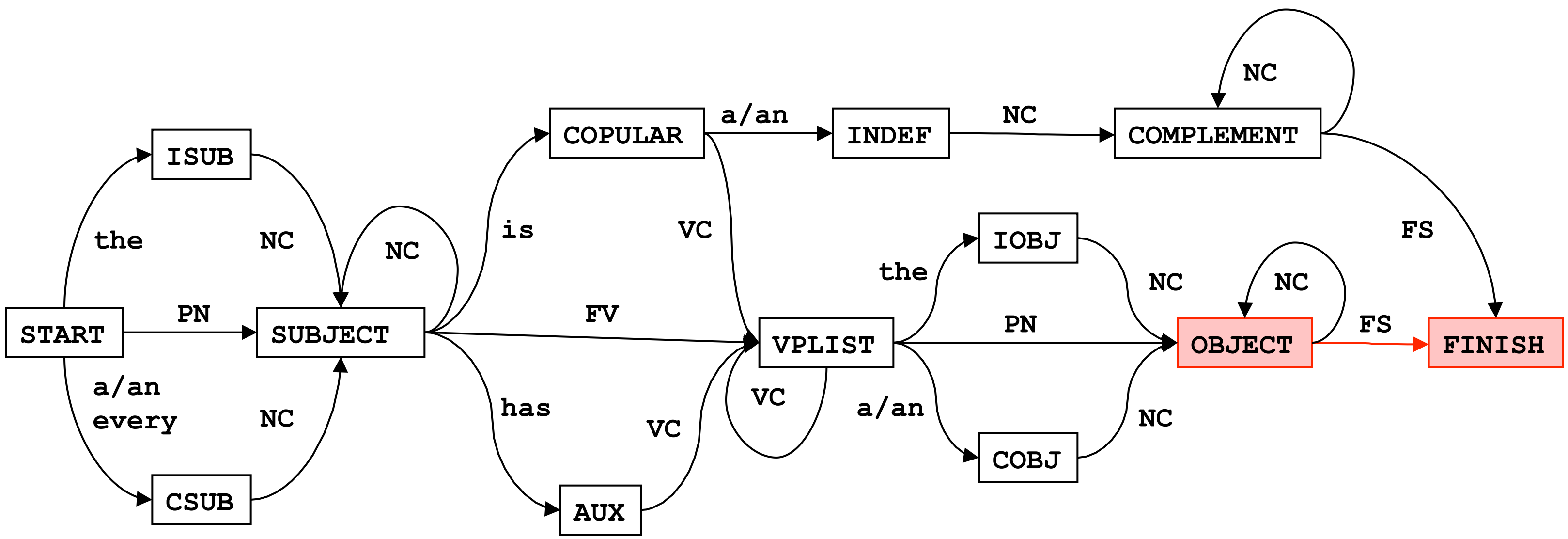


Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),
ObjectSomeValuesFrom(ObjectProperty(#is_married_to),Class(#lawyer)))

9

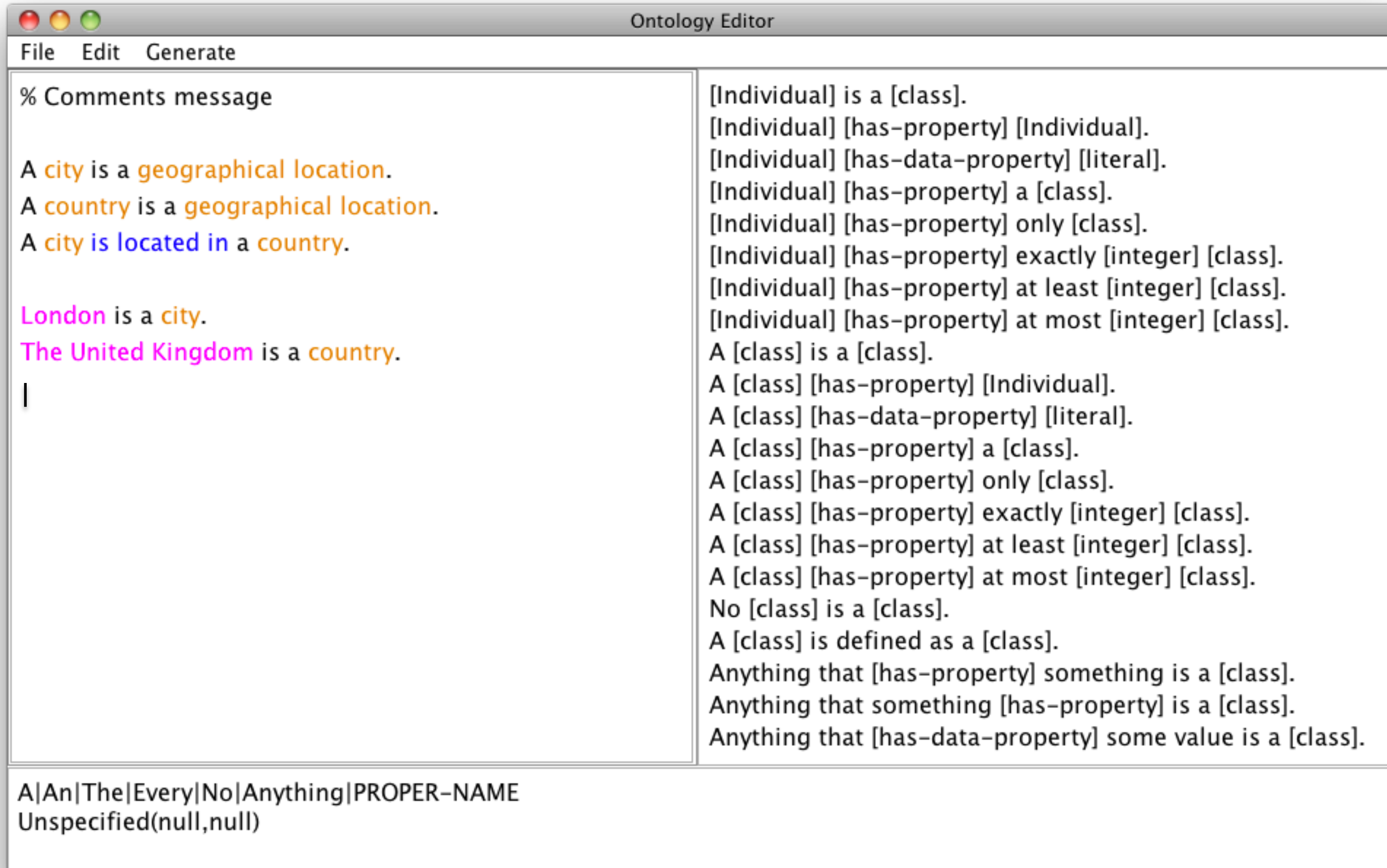
PN Proper noun
NC Noun-phrase continuation
VC Verb-phrase continuation
FV Tensed verb
FS Full stop



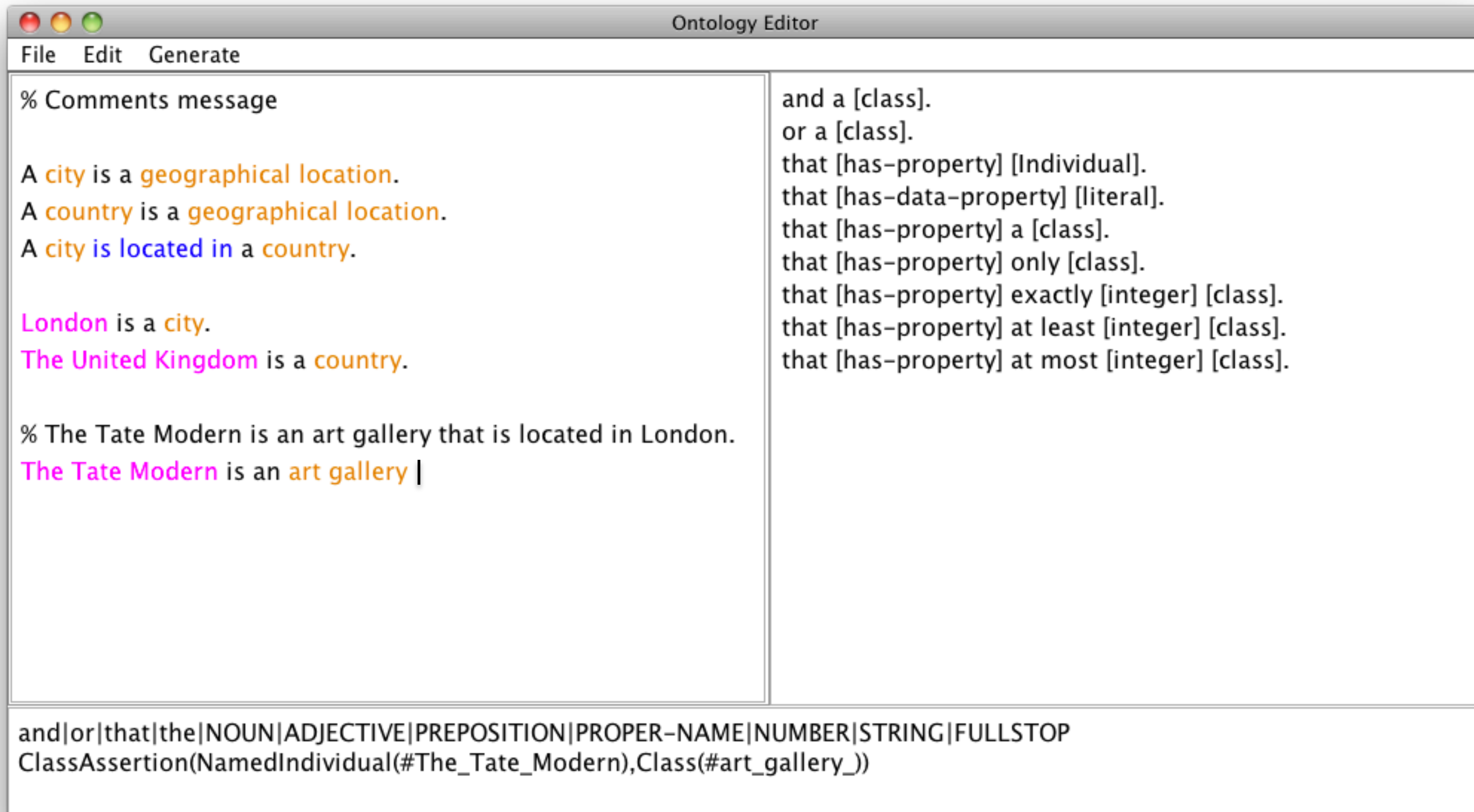
Tony Blair is married to a lawyer.

ClassAssertion(NamedIndividual(#Tony_Blair),
ObjectSomeValuesFrom(ObjectProperty(#is_married_to),Class(#lawyer)))

Basic sentence patterns



Sentence continuations



Sentence structure

Sentence = Subject Predicate

Subject = [Individual]
Subject = *A*|*Every*|*No* [Class]

Predicate = *is* NPList *that* VPList *that* VPChain

NPList = *a* [Class] *and a* [Class] …

VPList = [Props] *a* [Class] *and* [Props] …

VPChain = [Props] *a* [Class] *that* [Props] …

Outline

- Motivation
- Demonstration
- Language
- Coverage
- Conclusion

Complex axiom patterns

	Predicate pattern	Frequency
1	$C \sqcap P.C$	1100
2	$P.(C \sqcap P.C)$	473
3	$P.(C \sqcup C)$	434
4	$C \sqcap (P.C \sqcap P.C)$	248
5	$P.(C \sqcap (P.(C \sqcap P.C)))$	164
6	$P.(C \sqcup (C \sqcup C))$	105
7	$C \sqcap (P.C \sqcap (P.C \sqcap P.C))$	78
8	$P.C \sqcap P.C$	59
9	$C \sqcap (P.C \sqcap (P.C \sqcap (P.C \sqcap P.C)))$	47
10	$P.(C \sqcup (C \sqcup (C \sqcup C)))$	43
11	$C \sqcap (P.(C \sqcap P.C))$	39
12	$P.(C \sqcup (C \sqcup (C \sqcup (C \sqcup C))))$	35
13	$P.(P.C)$	33
14	$P.\neg C$	28
15	$P.(C \sqcup (C \sqcup (C \sqcup (C \sqcup (C \sqcup C)))))$	26
16	$P.C \sqcup P.C$	25
17	$P.C \sqcap (P.C \sqcap P.C)$	20
18	$C \sqcap (P.C \sqcap (P.C \sqcap (P.C \sqcap (P.C \sqcap (P.C \sqcap P.C)))))$	20
19	$\neg P.C$	19
20	$P.(C \sqcup (C \sqcup (C \sqcup (C \sqcup (C \sqcup (C \sqcup C))))))$	17

Results from corpus of over 550 ontologies

99.8% of axioms had simple subject term

All the top 20 complex predicate patterns are within the constraints of OWL Simplified English

If axiom patterns were created randomly we would expect just 2-3 to lie within our constraints

Three fundamental patterns

- Genus-Differentia (Aristotle)
  - *A pet-owner is a person that owns a pet*
- Restriction list
  - *A pet-owner owns a pet and cleans a cage*
- Alternative role-fillers
  - *A pet-owner owns a cat or a dog or a canary*

Measuring practical coverage

- Enumerate all possible complex class expressions up to a given complexity level
- Apply a criterion to determine which expressions yield ambiguous sentences
- Count the expected frequency of ambiguous sentences if all complex class expressions were equally likely
- Compare with the observed frequency for complex class expressions in an ontology corpus

Structural ambiguity

$$C \sqsubseteq \exists P.(C \sqcap C)$$

A N Vs a N and a N.

A child has as parent a mother and a father.

$$C \sqsubseteq \exists P.(C \sqcap \exists P.C \sqcap \exists P.C)$$

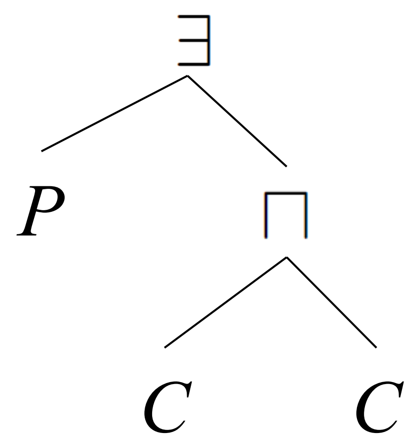
A N [Vs a N that Vs an N] and Vs a N.

A N Vs a N that [Vs an N and Vs a N].

A queen appoints a minister that governs a country and wears a crown.

Enumerating complex classes (1)

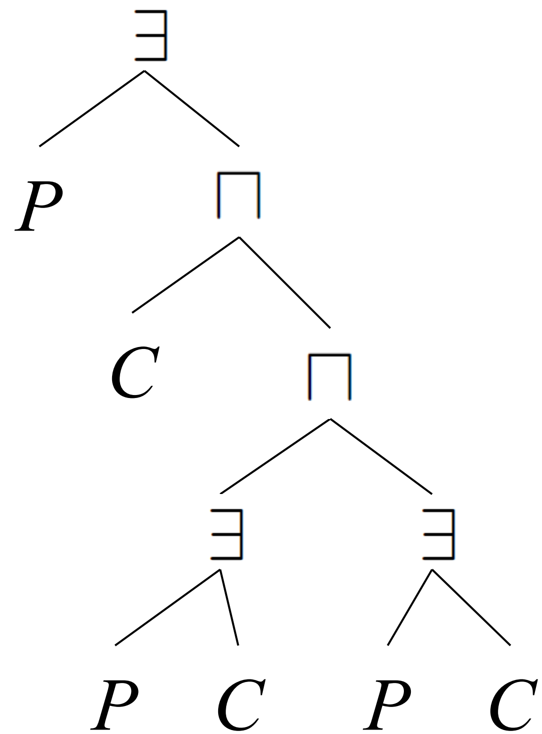
$$C \sqsubseteq \exists P.(C \sqcap C)$$



Complexity = 2 (number of non-terminal nodes)

Enumerating complex classes (2)

$$C \sqsubseteq \exists P.(C \sqcap \exists P.C \sqcap \exists P.C)$$



Complexity = 5 (normalised to binary tree)

Observed vs Expected

	Non-ambiguous	Ambiguous
Observed	5439	84
Expected	3107	2416

The **expected** frequency of complex axioms yielding ambiguous verbalisations was 2416/5523 or 43.7%.

The **obtained** frequency was 84/5523 or 1.5%.

Outline

- Motivation
- Demonstration
- Language
- Coverage
- Conclusion

Conclusions on complexity

- Overwhelmingly ontology authors favour complex class constructions that are not structurally ambiguous when verbalised
- Therefore, if we restrict sentence patterns to avoid structural ambiguity, almost all axioms found in our corpus could be formulated
- Probably many of the remaining axioms could be refactored

Main ideas in OSE

- Editing tool combines predictive authoring and WYSIWYM
- Finite-state controlled language favours efficient implementation and prevention of structural ambiguity
- Language requires minimal lexical input from user (verb list)
- Language allows but does not impose correct English

Will it work

User studies still pending ...

Questions?

... but thanks for your attention