

Editing OWL through generated CNL

Richard Power
Robert Stevens
Donia Scott
Alan Rector

8th June 2009

Table of contents

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

Outline

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

Semantic Web Authoring Tool (SWAT)

- ▶ Aim is to develop an NLG-based tool for viewing and editing knowledge on the semantic web
- ▶ Joint project between OU Computing Department and School of Computer Science at Manchester University
- ▶ EPSRC-funded for 3 years from 1st June 2009



<http://www.swatproject.org>

Partners

Open University
Department of Computing
Natural Language Generation group
Richard Power and Donia Scott



Manchester University
School of Computer Science
Bio-Health Informatics Group
Robert Stevens and Alan Rector



Advisory Board

Siemens Healthcare (medical orders)

World Health Organisation

NHS Connecting for Health

Ordnance Survey

MyGrid (workflows, data provenance)

Stanford University (Protege)

The logo for Siemens, featuring the word "SIEMENS" in a bold, teal, sans-serif font.The logo for NHS Connecting for Health, featuring the letters "NHS" in a blue box above the text "Connecting for Health" in a blue serif font.The logo for Ordnance Survey, featuring a stylized "OS" in a blue and pink box followed by the text "Ordnance Survey" in a blue serif font.

Aims of SWAT

1. To understand current authoring practice for ontologies and other metadata represented in OWL/RDF. This will provide requirements for a Semantic Web authoring tool.
2. To find principles based on these authoring requirements for bridging between natural language and formal ontologies (and other metadata).
3. To provide a format for encoding linguistic resources that supports the mapping of linguistic patterns to ontologies.
4. To produce a tool allowing users to extend the mapping of language to ontology without need of expertise in linguistics.
5. To produce (by combining the above) a tool for authoring ontologies and other metadata by direct manipulation of generated texts, for real-world applications in e-Science, biomedicine, and travel.

Aims of SWAT

1. To understand current authoring practice for ontologies and other metadata represented in OWL/RDF. This will provide requirements for a Semantic Web authoring tool.
2. To find principles based on these authoring requirements for bridging between natural language and formal ontologies (and other metadata).
3. To provide a format for encoding linguistic resources that supports the mapping of linguistic patterns to ontologies.
4. To produce a tool allowing users to extend the mapping of language to ontology without need of expertise in linguistics.
5. To produce (by combining the above) a tool for authoring ontologies and other metadata by direct manipulation of generated texts, for real-world applications in e-Science, biomedicine, and travel.

Aims of SWAT

1. To understand current authoring practice for ontologies and other metadata represented in OWL/RDF. This will provide requirements for a Semantic Web authoring tool.
2. To find principles based on these authoring requirements for bridging between natural language and formal ontologies (and other metadata).
3. To provide a format for encoding linguistic resources that supports the mapping of linguistic patterns to ontologies.
4. To produce a tool allowing users to extend the mapping of language to ontology without need of expertise in linguistics.
5. To produce (by combining the above) a tool for authoring ontologies and other metadata by direct manipulation of generated texts, for real-world applications in e-Science, biomedicine, and travel.

Aims of SWAT

1. To understand current authoring practice for ontologies and other metadata represented in OWL/RDF. This will provide requirements for a Semantic Web authoring tool.
2. To find principles based on these authoring requirements for bridging between natural language and formal ontologies (and other metadata).
3. To provide a format for encoding linguistic resources that supports the mapping of linguistic patterns to ontologies.
4. To produce a tool allowing users to extend the mapping of language to ontology without need of expertise in linguistics.
5. To produce (by combining the above) a tool for authoring ontologies and other metadata by direct manipulation of generated texts, for real-world applications in e-Science, biomedicine, and travel.

Aims of SWAT

1. To understand current authoring practice for ontologies and other metadata represented in OWL/RDF. This will provide requirements for a Semantic Web authoring tool.
2. To find principles based on these authoring requirements for bridging between natural language and formal ontologies (and other metadata).
3. To provide a format for encoding linguistic resources that supports the mapping of linguistic patterns to ontologies.
4. To produce a tool allowing users to extend the mapping of language to ontology without need of expertise in linguistics.
5. To produce (by combining the above) a tool for authoring ontologies and other metadata by direct manipulation of generated texts, for real-world applications in e-Science, biomedicine, and travel.

Outline

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

Semantic Web

- ▶ Gigantic Global Graph
- ▶ Based on Description Logic
- ▶ Standard formalisms (OWL, RDF)
- ▶ No standard ontology
- ▶ Everyone can publish ontologies/data

Obstacles

- ▶ OWL, RDF etc. are unreadable
- ▶ Logic can be misinterpreted
- ▶ Ontologies/datasets can be large
- ▶ Mistakes are hard to detect
- ▶ Users need guidance (what next?)

Objectives

Supporting subject-matter experts:

- ▶ Transparent presentation of ontologies/data
- ▶ Summaries, navigation aids
- ▶ Clarification of logical statements
- ▶ Explanation of errors
- ▶ Advice on options during editing

Controlled Natural Languages

- ▶ Attempto Controlled English
- ▶ Sydney OWL Syntax
- ▶ Rabbit
- ▶ ...

Usual strategy

Users type in sentences, possibly aided by forms interface.

Texts in the CNL can be generated from existing ontologies.

Outline

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

Processable English (PENG)

The screenshot shows a window titled "PENG Demo" with a menu bar (File, Edit, View, Tools, Mode, Help). Below the menu is a "Question:" field. A tabbed interface shows the "Summary[1]" tab. The summary contains:

Title: Department of Computing
Link: <http://www.ics.mq.edu.au>
Lexicon: <http://>

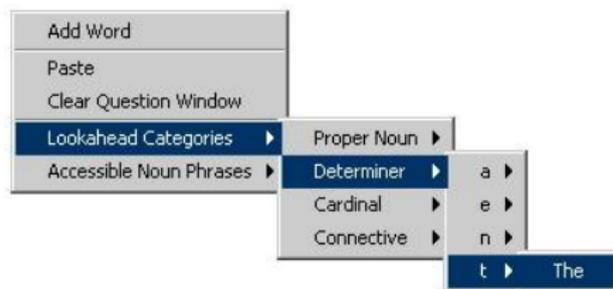
If X is a research programmer then X is a programmer.
If X is a research programmer then X is not a staff member. | [connective :[If]]

DRS:
[drs([], [drs([A, B, C], [pred(A, [be], C)#[0, 2], evt(A, state)#[0, 2], obj([research, programmer], C)#[0, 2], struc(C, atomic)#[0, 2], var(B, C)#[0, 2], struc(C, atomic)#[0, 2]])>drs([], [~drs([D], [pred(D, [be], C)#[0, 2], evt(D, state)#[0, 2], obj([staff, member], C)#[0, 2], struc(C, atomic)#[0, 2]]))), drs([E, F, G], [pred(E, [be], G)#[0, 1], evt(E, state)#[0, 1], obj([research, programmer], G)#[0, 1], struc(G, atomic)#[0, 1], var(F, G)#[0, 1], struc(G, atomic)#[0, 1]])>drs([H], [pred(H, [be], G)#[0, 1], evt(H, state)#[0, 1], obj([programmer], G)#[0, 1], struc(G, atomic)#[0, 1]])])]

<http://web.science.mq.edu.au/~rolfs/peng/writing-peng.html>

Predictive authoring (PENG)

Looking ahead



Accessible NPs



Predictive authoring (ACEWiki)

The screenshot shows the 'Editor' window with a 'Sentence' field containing the text 'if a protein is a subunit of a protein-complex then' followed by a red '1'. Below this is a '< Delete' button. A second field labeled 'add text / filter:' with a red '2' is empty. Below that are three columns of options:

- add function word:** a list with 'a', 'everything', 'every', 'it is false that', 'nothing', 'no', 'something'. A red '3' is next to 'nothing'.
- add proper name:** a scrollable list with 'New Proper Name...', 'Act1', 'Amine-Transport', 'Biosynthesis', 'BLOC', 'Bub1', 'Carbohydrate-Transport', 'Cav1', 'CBF3', 'Cbl', 'Cell-Adhesion', 'Cell-Communication'. A red '4' is next to 'Act1'.
- add reference:** a list with 'the protein', 'the protein-complex'. A red '5' is next to the header.

At the bottom right are 'OK' and 'Cancel' buttons.

http://attempto.ifi.uzh.ch/site/docs/authoring_tools.html

Outline

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

WYSIWYM (Conceptual Authoring)

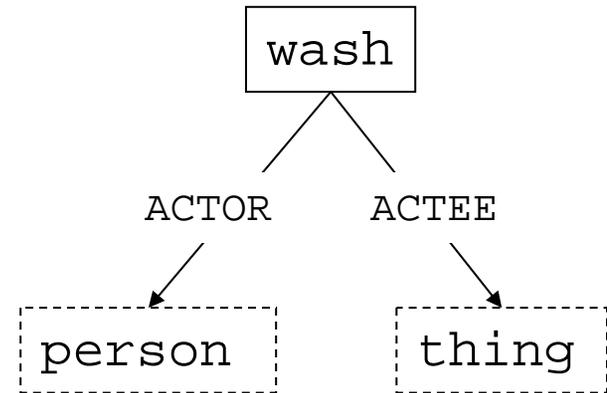
- ▶ Alternative CNL interface
- ▶ Relies entirely on NLG
- ▶ Texts show knowledge so far defined
- ▶ Menus show editing options
- ▶ Used in ≈ 10 projects for ABox editing

Example of ABox editing using WYSIWYM ...

Feedback text

Somebody washes something

What the user sees

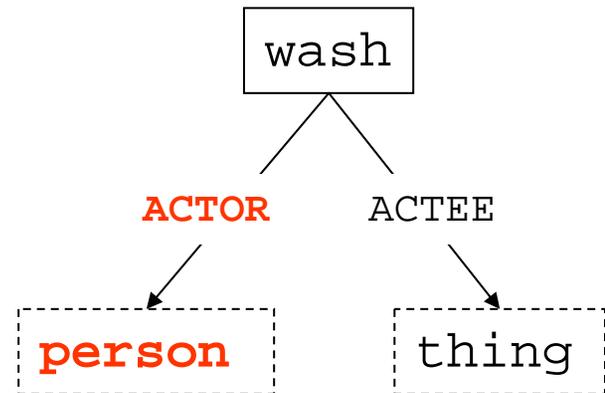


```
wash(e1)  
actor(e1,e2)  
actee(e1,e3)
```

Underlying knowledge

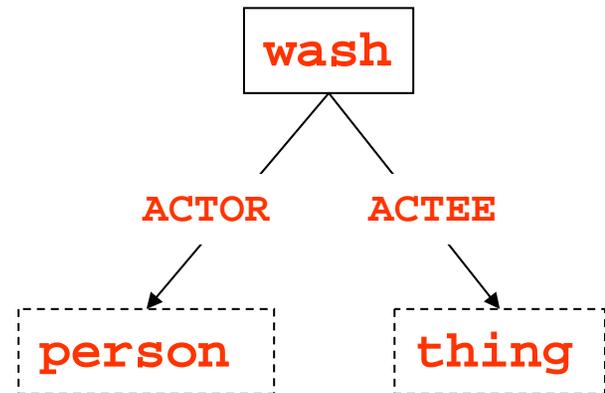
Selecting an entity (1)

Somebody washes something



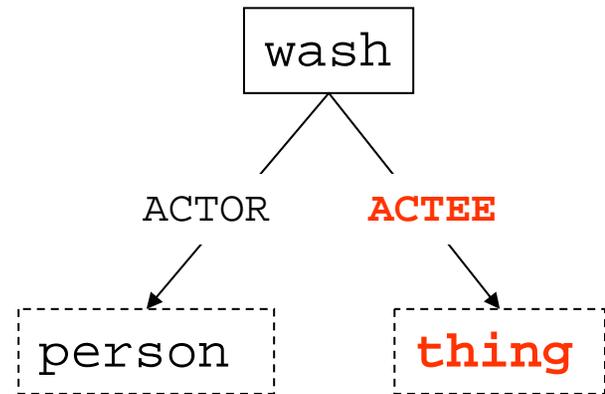
Selecting an entity (2)

Somebody washes something

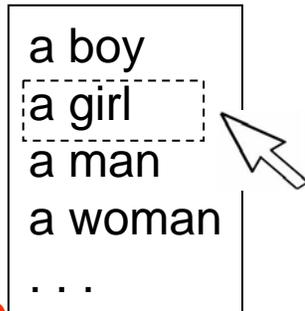


Selecting an entity (3)

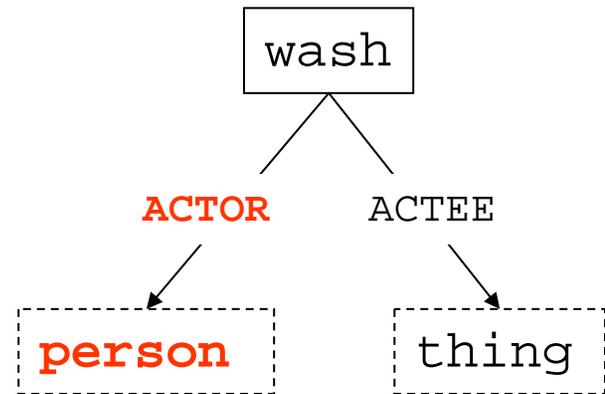
Somebody washes something



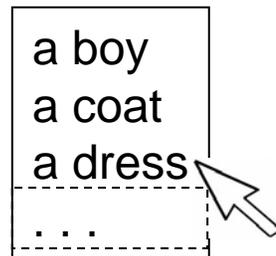
Specifying an entity type (1)



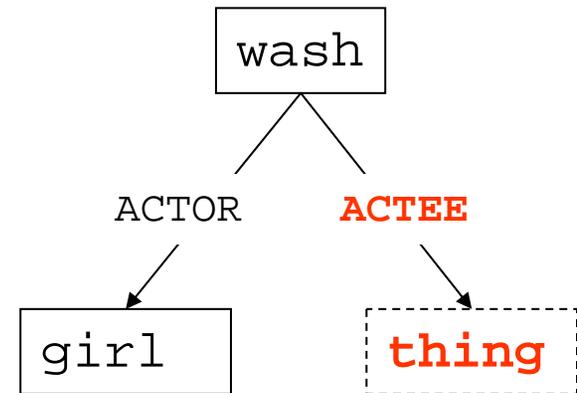
Somebody washes something



Specifying an entity type (2)



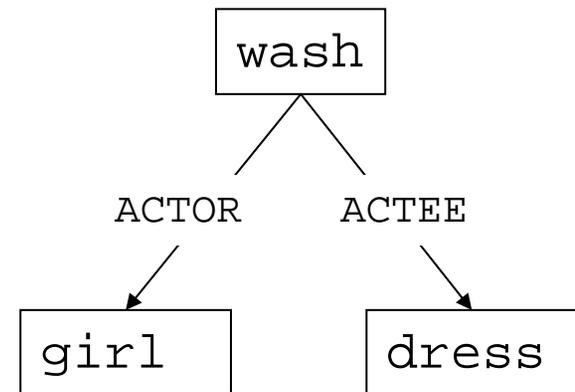
A girl washes **something**



```
wash(e1)  
actor(e1,e2)  
actee(e1,e3)  
girl(e2)
```

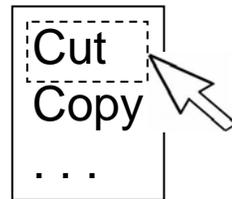
Specifying an entity type (3)

A girl washes a dress

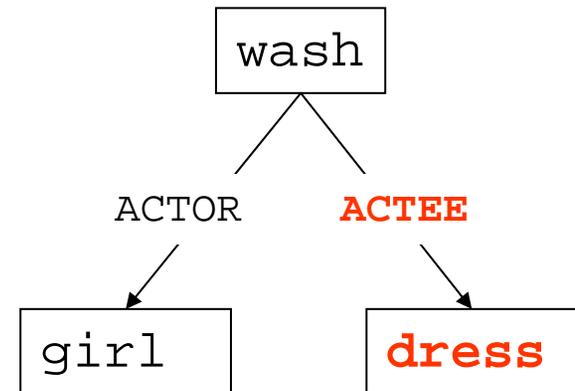


```
wash(e1)  
actor(e1,e2)  
actee(e1,e3)  
girl(e2)  
dress(e3)
```

Cutting an attribute value (1)

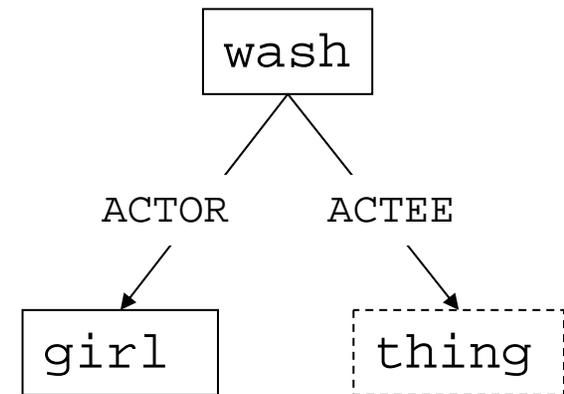


A girl washes **a dress**



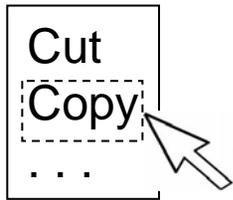
Cutting an attribute value (2)

A girl washes something

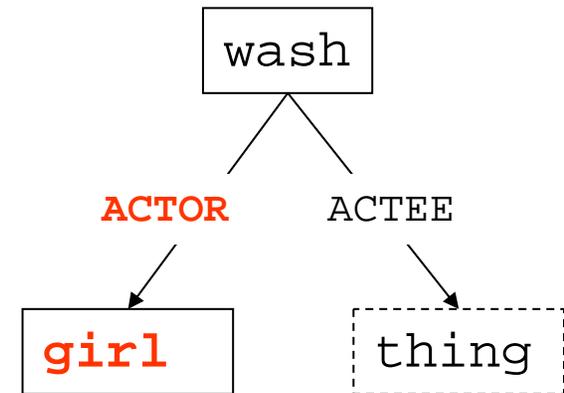


```
wash(e1)  
actor(e1,e2)  
actee(e1,e3)  
girl(e2)
```

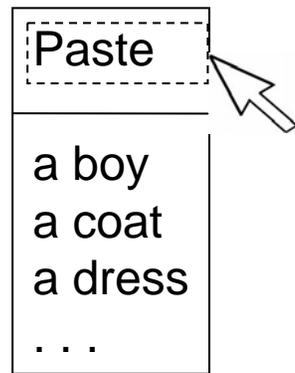
Copy and Paste (1)



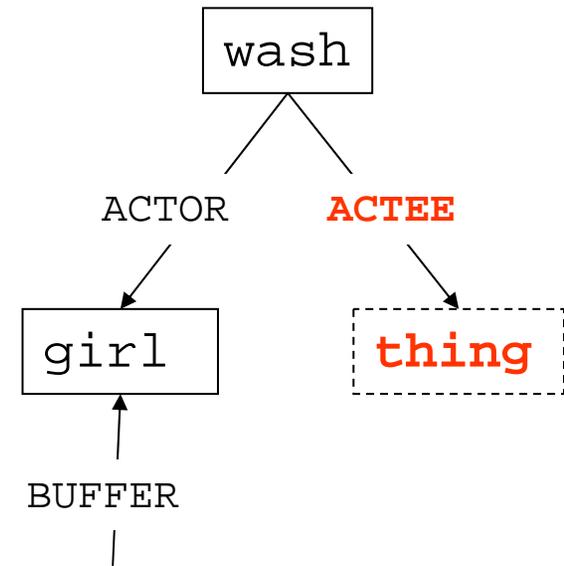
A girl washes something



Copy and Paste (2)

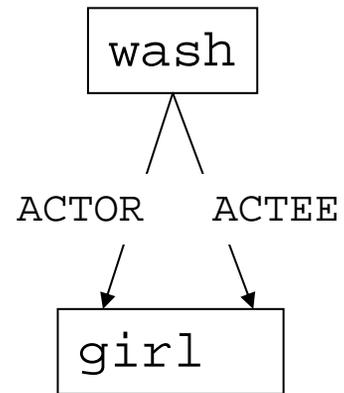


A girl washes **something**



Copy and Paste (3)

A girl washes herself

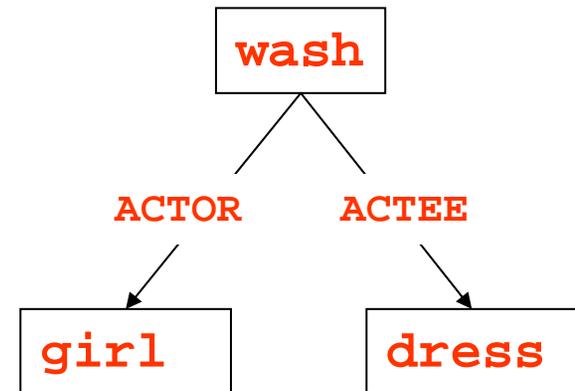


```
wash(e1)  
actor(e1,e2)  
actee(e1,e2)  
girl(e2)
```

Reconfiguring an entity type (1)

A girl washed a dress
A girl will wash a dress
A girl washes a dress somewhere
A girl washes a dress sometime
A girl washes a dress in some way
...

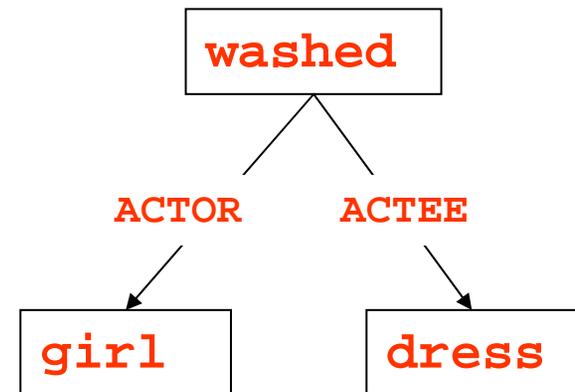
A girl washes an dress



Reconfiguring an entity type (2)

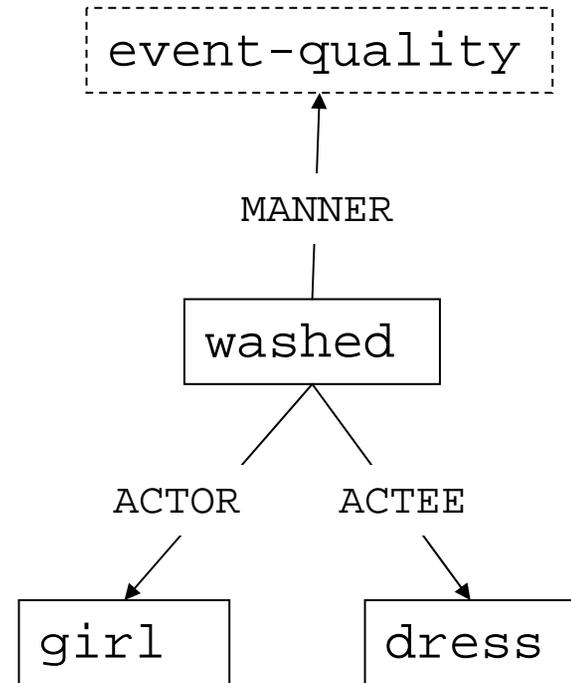
A girl washes a dress
A girl will wash a dress
A girl washed a dress somewhere
A girl washed a dress sometime
A girl washed a dress in some way
...

A girl washed a dress



Reconfiguring an entity type (3)

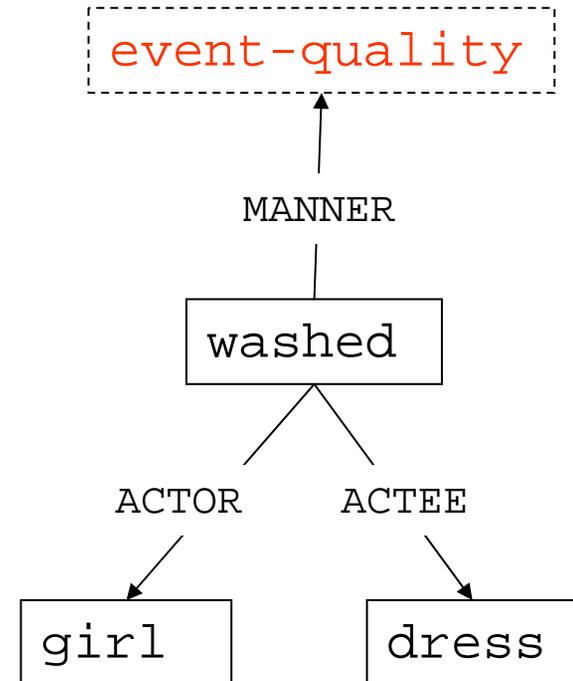
A girl washed a dress in some way



Reconfiguring an entity type (4)

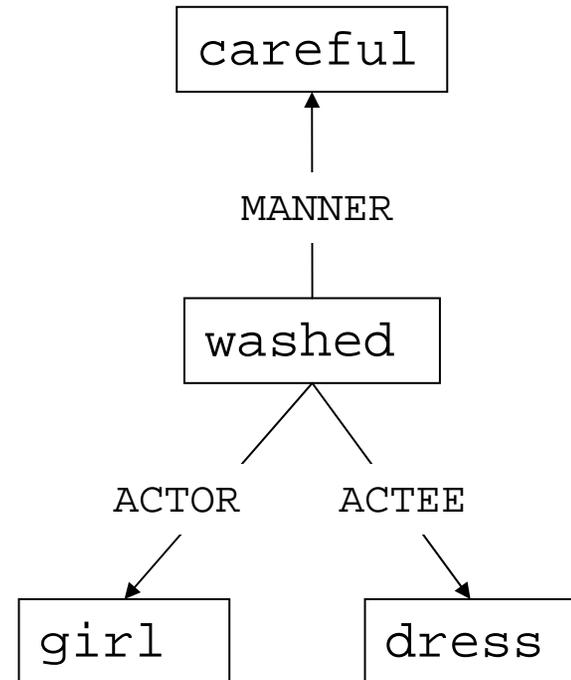
carefully
speedily
thoroughly
...

A girl washed a dress **in some way**



Result of editing

A girl washed a dress carefully



Outline

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

TBox authoring with WYSIWYM

- ▶ Requires extension of WYSIWYM model
- ▶ How should axioms be aligned to text?
- ▶ How should editing operations be presented?

We have a prototype that explores these issues . . .

Aligning axioms to clauses

- ▶ In general, express $C \sqsubseteq D$ by $S[NP, VP]$ where NP expresses C and VP expresses D
- ▶ Individual statements can be fitted to this scheme by using enumerated sets – e.g., $\{a\} \sqsubseteq C$ for $C(a)$

Examples

$dog \sqsubseteq animal$ 'Every dog is an animal'

$dogOwner \sqsubseteq \exists own.dog$ 'Every dog-owner owns one or more dogs'

$\{Pixie\} \sqsubseteq dog$ 'Pixie is a dog'

$\{Mary\} \sqsubseteq \exists own.\{Pixie\}$

Simple CNL fragment

- ▶ Each logical term (individual, class, property) is lexicalised by just one word
- ▶ Individuals are lexicalised by proper names (e.g., Mary, Pixie)
- ▶ Classes are lexicalised by count nouns (e.g., dog, person, thing)
- ▶ Properties are lexicalised by transitive verbs (e.g., own, like) or nouns (e.g., owner)

Editing procedure

- ▶ The knowledge comprises a set of class inclusion axioms ($C \sqsubseteq D$) covering both TBox and ABox
- ▶ The user has controls for adding or removing axioms from the list, and for adding terms to the lexicon
- ▶ New axioms have the form $\top \sqsubseteq \top$ ('every thing is a thing')
- ▶ Editing proceeds by replacing a selected class with another class chosen from a menu of options

Here is a mock-up of how this could work . . .



SWAT Knowledge Base Editor

File

Edit

View

Tools

Feedback text

List of terms



Large empty white text area for feedback text.

Large empty light gray text area for list of terms.

Large empty white text area for feedback text.

Large empty light gray text area for list of terms.



SWAT Knowledge Base Editor

File

Edit

View

Tools

- New KB
- Open KB
- Save KB

List of terms



SWAT Knowledge Base Editor [KB-1]

File



Edit

View

Tools

Feedback text [SWAT]

List of terms

thing

owl:thing



SWAT Knowledge Base Editor [KB-1]

File

Edit

View

Tools

Feedback



- New axiom
- New instance
- New class
- New role
- Copy
- Paste
- Delete

List of terms

thing

owl:thing



SWAT Knowledge Base Editor [KB-1]

File

Edit



View

Tools

Feedback text [SWAT]

List of terms

(1) Every thing is a thing

thing

(1) owl:thing \sqsubseteq owl:thing

owl:thing

SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text [SWAT]		SWAT ✓ ACE Rabbit Sydney	List of terms
(1) Every thing is a thing			thing
(1) owl:thing \sqsubseteq owl:thing			owl:thing



SWAT Knowledge Base Editor [KB-1]

File

Edit



View

Tools

Feedback text [Rabbit]

List of terms

(1) Every thing is a kind of thing

thing

(1) owl:thing \sqsubseteq owl:thing

owl:thing

SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text [Rabbit]	→	List of terms	
(1) Every thing is a kind	<ul style="list-style-type: none">SWATACERabbit ✓Sydney	thing	
(1) owl:thing \sqsubseteq owl:thing		owl:thing	



SWAT Knowledge Base Editor [KB-1]

File



Edit

View

Tools

Feedback text [SWAT]

List of terms

(1) Every thing is a thing

thing

(1) owl:thing \sqsubseteq owl:thing

owl:thing

SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text	<ul style="list-style-type: none">New axiomNew instanceNew classNew roleCopyPasteDelete		List of terms
(1) Every in			thing
(1) owl:thing \sqsubseteq owl:thing			owl:thing



Feedback text [SWAT]

List of terms

(1) Every thing is a thing

thing

New instance name

OWL name

English name



(1) owl:thing

owl:thing



SWAT Knowledge Base Editor [KB-1]

File



Edit

View

Tools

Feedback text [SWAT]

List of terms

(1) Every thing is a thing

Mary
thing

(1) owl:thing \sqsubseteq owl:thing

ns:Mary34
owl:thing

 SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text	<ul style="list-style-type: none"> New axiom New instance New class New role Copy Paste Delete 		List of terms
(1) Every			Mary thing
(1) owl:thing \sqsubseteq owl:thing			ns:Mary34 owl:thing

Feedback text [SWAT]

(1) Every thing is a thing

List of terms

Mary
thing

New class name

OWL name

English name ▼

- periscope
- person
- pest
- pet
- philatelist
- philistine

OK

(1) owl:thin

:Mary34
l:thing

Feedback text [SWAT]

(1) Every thing is a thing

List of terms

Mary
thing

New class name

OWL name

English name ▼

- periscope
- person
- pest
- pet
- philatelist
- philistine



(1) owl:thin

:Mary34
l:thing

 SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]

List of terms

(1) Every thing is a thing



Mary
person
thing
woman

(1) owl:thing \sqsubseteq owl:thing

ns:Mary34
ns:HumanBeing
owl:thing
ns:HumanFemale

Feedback text [SWAT]

(1) **Every thing** is a thing



List of terms

Mary
person
thing
woman

(1) **owl:thing** \sqsubseteq owl:thing

ns:Mary34
ns:HumanBeing
owl:thing
ns:HumanFemale

Feedback text [SWAT]	List of terms
<p>(1) Mary is a thing</p> 	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq owl:thing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

Feedback text [SWAT]	List of terms
1: Mary is a thing	Mary person thing woman
(1) {ns:Mary34} \sqsubseteq owl:thing	ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale



 SWAT Knowledge Base Editor [KB-1]

File  Edit View Tools

Feedback text [SWAT]

List of terms

(1) Mary is a woman

Mary
person
thing
woman

(1) {ns:Mary34} \sqsubseteq ns:HumanFemale

ns:Mary34
ns:HumanBeing
owl:thing
ns:HumanFemale

SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback	<ul style="list-style-type: none"> New axiom New instance New class New role Copy Paste Delete 		List of terms
(1) Mary is :			Mary person thing woman
(1) {ns:Mary34} \sqsubseteq ns:HumanFemale			ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman</p> <p>(2) Every thing is a thing</p> 	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale</p> <p>(2) owl:thing \sqsubseteq owl:thing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every thing is a thing</p>	<p>Mary person thing woman</p> 
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) owl:thing \sqsubseteq owl:thing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every woman is a thing</p> 	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq owl:thing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

 SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every woman is a thing</p>	<p>Mary person thing woman</p> 
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq owl:thing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>



SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]

List of terms

(1) Mary is a woman

(2) Every woman is a person

Mary
person
thing
woman

(1) {ns:Mary34} \sqsubseteq ns:HumanFemale

(2) ns:HumanFemale \sqsubseteq ns:HumanBeing

ns:Mary34
ns:HumanBeing
owl:thing
ns:HumanFemale

 SWAT Knowledge Base Editor [KB-1]

File  Edit View Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every woman is a person</p>	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text	<ul style="list-style-type: none"> New axiom New instance New class New role Copy Paste Delete 		List of terms
<p>(1) Mary is a woman</p> <p>(2) Every woman is a person</p>			<p>Mary</p> <p>person</p> <p>thing</p> <p>woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale</p> <p>(2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p>			<p>ns:Mary34</p> <p>ns:HumanBeing</p> <p>owl:thing</p> <p>ns:HumanFemale</p>

 SWAT Knowledge Base Editor [KB-1]

File  Edit View Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every woman is a person</p>	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text	<ul style="list-style-type: none"> New axiom New instance New class New role Copy Paste Delete 		List of terms
<p>(1) Mary is a woman</p> <p>(2) Every woman is a person</p>			<p>Mary</p> <p>person</p> <p>thing</p> <p>woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale</p> <p>(2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p>			<p>ns:Mary34</p> <p>ns:HumanBeing</p> <p>owl:thing</p> <p>ns:HumanFemale</p>

 SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every woman is a person (3) Every woman is a person</p> 	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq ns:HumanBeing (3) ns:HumanFemale \sqsubseteq ns:HumanBeing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>

SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman</p> <p>(2) Every woman is a person</p> <p>(3) Every woman is a person</p>	<p>Mary</p> <p>person</p> <p>thing</p> <p>woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale</p> <p>(2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p> <p>(3) ns:HumanFemale \sqsubseteq ns:HumanBeing</p>	<p>ns:Mary34</p> <p>ns:HumanBeing</p> <p>owl:thing</p> <p>ns:HumanFemale</p>



 SWAT Knowledge Base Editor [KB-1]

File Edit View  Tools

Feedback text [SWAT]	List of terms
<p>(1) Mary is a woman (2) Every woman is a person (3) Mary is a person</p>	<p>Mary person thing woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq ns:HumanBeing (3) {ns:Mary34} \sqsubseteq ns:HumanBeing</p>	<p>ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale</p>



SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools		
Feedback text [SWAT]			 <ul style="list-style-type: none"> Consistency check Redundancy check Discourse generator 		List of terms
<p>(1) Mary is a woman</p> <p>(2) Every woman is a person</p> <p>(3) Mary is a person</p>					<p>Mary</p> <p>person</p> <p>thing</p> <p>woman</p>
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale</p> <p>(2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p> <p>(3) {ns:Mary34} \sqsubseteq ns:HumanBeing</p>					<p>ns:Mary34</p> <p>ns:HumanBeing</p> <p>owl:thing</p> <p>ns:HumanFemale</p>

SWAT Knowledge Base Editor [KB-1]

File Edit View Tools

Feedback text [SWAT]

- (1) Mary is a woman
- (2) Every woman is a person
- (3) **Mary** is a person

List of terms

Mary
person
thing
woman

Consistency check

Knowledge base KB-1 is consistent



- (1) {ns:Mary34} \sqsubseteq ns:HumanFemale
- (2) ns:HumanFemale \sqsubseteq ns:HumanBeing
- (3) {ns:Mary34} \sqsubseteq ns:HumanBeing

mary34
ns:HumanBeing
owl:thing
ns:HumanFemale

 SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text [SWAT]			Consistency check Redundancy check Discourse generator
<p>(1) Mary is a woman</p> <p>(2) Every woman is a person</p> <p>(3) Mary is a person</p>			List of terms Mary person thing woman
<p>(1) {ns:Mary34} \sqsubseteq ns:HumanFemale</p> <p>(2) ns:HumanFemale \sqsubseteq ns:HumanBeing</p> <p>(3) {ns:Mary34} \sqsubseteq ns:HumanBeing</p>			ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale



File Edit View Tools

Feedback text [SWAT]

- (1) Mary is a woman
- (2) Every woman is a person
- (3) **Mary** is a person

List of terms

Mary
 person
 thing
 woman

Redundancy check

Axiom (3) in knowledge base KB-1 is redundant, since if Mary is a woman and every woman is a person, Mary must be a person.

- (1) {ns:Mary:...
- (2) ns:Huma...
- (3) {ns:Mary:...



ry34
 manBeing
 ng
 ns.HumanFemale

 SWAT Knowledge Base Editor [KB-1]

File	Edit	View	Tools
Feedback text [SWAT]		<div data-bbox="662 249 1100 406" style="border: 1px solid black; padding: 5px;"> Consistency check Redundancy check Discourse generator </div>	List of terms
(1) Mary is a woman (2) Every woman is a person (3) Mary is a person			Mary person thing woman
(1) {ns:Mary34} \sqsubseteq ns:HumanFemale (2) ns:HumanFemale \sqsubseteq ns:HumanBeing (3) {ns:Mary34} \sqsubseteq ns:HumanBeing			ns:Mary34 ns:HumanBeing owl:thing ns:HumanFemale

Feedback text [SWAT]

- (1) Mary is a woman
- (2) Every woman is a person
- (3) **Mary** is a person

List of terms

Mary
person
thing
woman

Discourse generator

Every woman is a person.

Mary is a woman, so Mary is also a person.

- (1) {ns:Mary3
- (2) ns:Human
- (3) {ns:Mary3

y34
nanBeing
ng
nanFemale



Completed knowledge base

- 1: Mary is a woman
- 2: Every woman is a person
- 3: Pixie is a dog
- 4: Every dog is a pet
- 5: Kitty is a cat
- 6: Every cat is a pet
- 7: Every pet is an animal
- 8: Mary owns Pixie
- 9: Mary owns Kitty
- 10: Everything that owns one or more pets is a pet-owner
- 11: Every pet-owner owns one or more pets
- 12: Every pet-owner is a person

Service 1: Coherent discourse

- 1: Mary is a woman
- 2: Every woman is a person
- 3: Pixie is a dog
- 4: Every dog is a pet
- 5: Kitty is a cat
- 6: Every cat is a pet
- 7: Every pet is an animal
- 8: Mary owns Pixie
- 9: Mary owns Kitty
- 10: Everything that owns one or more pets is a pet-owner
- 11: Every pet-owner owns one or more pets
- 12: Every pet-owner is a person

Every cat is a pet; also, every dog is a pet. Every pet is an animal. Every woman is a person; also, every pet-owner is a person. Everyone who owns one or more pets is a pet-owner; conversely, every pet-owner owns one or more pets.

Mary is a woman. She owns Pixie, which is a dog. She also owns Kitty, which is a cat.

Service 2: Answering questions

- 1: Mary is a woman
- 2: Every woman is a person
- 3: Pixie is a dog
- 4: Every dog is a pet
- 5: Kitty is a cat
- 6: Every cat is a pet
- 7: Every pet is an animal
- 8: Mary owns Pixie
- 9: Mary owns Kitty
- 10: Everything that owns one or more pets is a pet-owner
- 11: Every pet-owner owns one or more pets
- 12: Every pet-owner is a person

Is it true that Mary is a pet-owner?

Yes. Mary owns Kitty (8), which is a cat (5). Since every cat is a pet (6), Kitty is a pet, so Mary owns one or more pets. However, everyone who owns one or more pets is a pet-owner (10), so Mary must be a pet-owner.

Feedback through dialogue

1: Every woman is a person

Okay

2: Mary is a woman

Okay

3: Mary is a person

Obviously, since she is a woman and every woman is a person

4: Every man is a person

Okay

5: Every man is a non-woman

Okay

6: Mary is a man

Impossible, since she is a woman and every man is a non-woman

Outline

SWAT project

Background

NLU-based authoring

NLG-based authoring

TBox authoring

Interpretation vs Generation

Comparing interpretation and generation

- ▶ We originally used generation (a) to avoid interpretation errors (b) to avoid need for users to learn the CNL
 - ▶ However, perhaps the same result is achieved by deterministic parsers and predictive authoring
- ▶ Predictive authoring tends to be left-to-right, WYSIWYM tends to be top-down
 - ▶ Unclear which is better – might vary for different users
- ▶ NLG-based methods give more freedom in designing the CNL, since it can use formulations that would be hard to interpret or inconvenient for users to write
 - ▶ For instance, complex boolean expressions can be clarified by punctuation and formatting (bullets)

Presenting complex boolean conditions

Relevant subjects

Patients with the following properties:

- ▶ They are younger than 60 years of age
- ▶ AND
- ▶ They have one of these properties:
 - ▶ They have had bad prognosis myelodysplastic syndrome only for at least six months
 - ▶ OR
 - ▶ They have had acute myelogenous leukaemia caused by bad prognosis myelodysplastic syndrome for at least six months

[...]

(Using the WYSIWYM-based CLEF query tool, medical researchers were able to define queries of this complexity in around 3 minutes.)

Conclusion

- ▶ The SWAT project investigates knowledge authoring for the semantic web using generated CNL
- ▶ Using generation rather than interpretation has implications for the editing process and the design of the CNL
- ▶ Perhaps generation vs interpretation is less important than some other issues that both approaches must confront:
 - ▶ adding new terms
 - ▶ guiding knowledge formation
 - ▶ scaling up to large ontologies/datasets