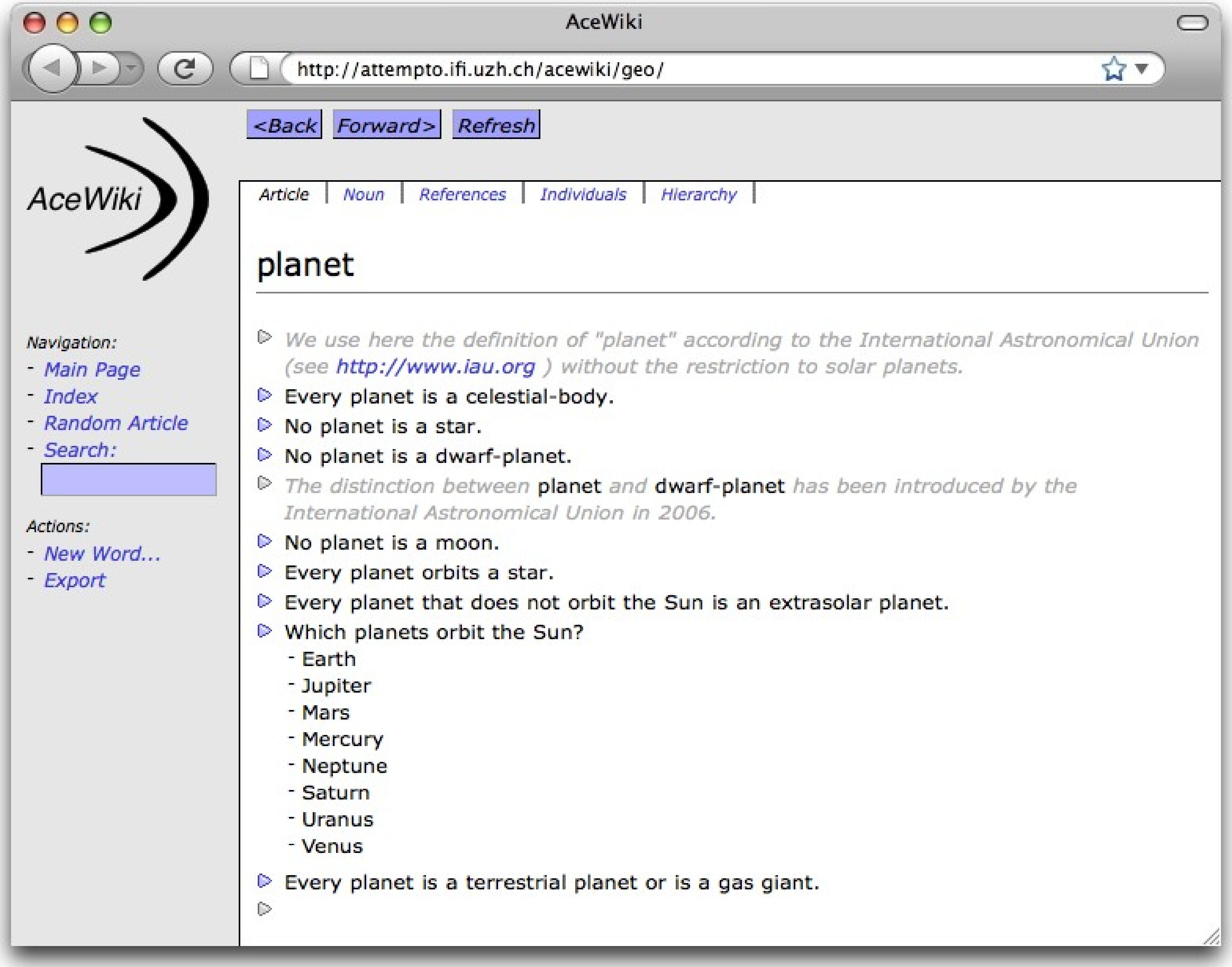
How to Evaluate
Controlled Natural Languages

Tobias Kuhn

Workshop on Controlled Natural Language (CNL 2009),
Marettimo, Italy

8 June 2009

Off Topic: AceWiki



Navigation:
- Main Page
- Index
- Random Article
- Search:

Actions:
- New Word...
- Export

planet

▷ *We use here the definition of "planet" according to the International Astronomical Union (see http://www.iau.org) without the restriction to solar planets.*

▷ Every planet is a celestial-body.

▷ No planet is a star.

▷ No planet is a dwarf-planet.

▷ *The distinction between* planet *and* dwarf-planet *has been introduced by the International Astronomical Union in 2006.*

▷ No planet is a moon.

▷ Every planet orbits a star.

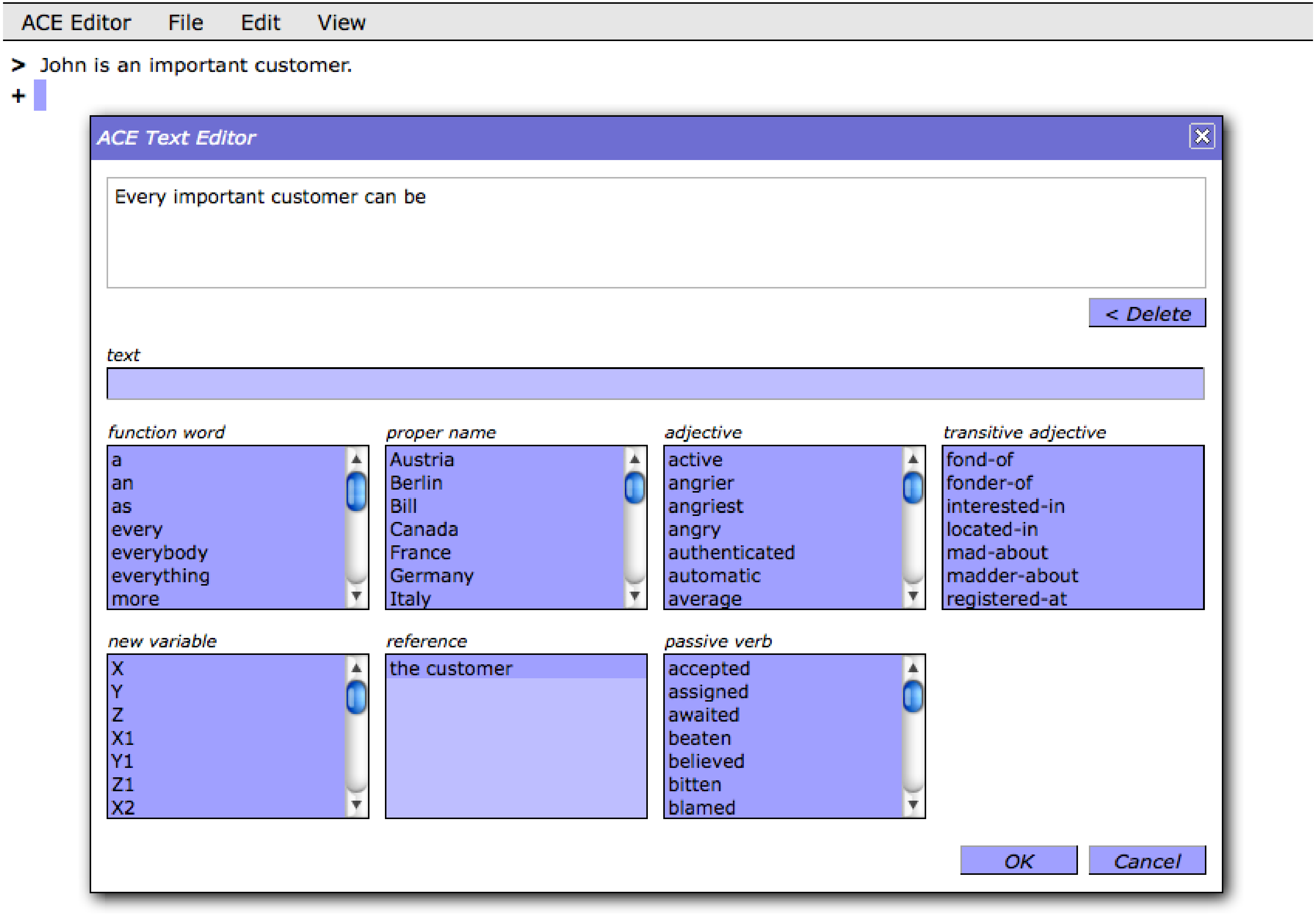
▷ Every planet that does not orbit the Sun is an extrasolar planet.

▷ Which planets orbit the Sun?
- Earth
- Jupiter
- Mars
- Mercury
- Neptune
- Saturn
- Uranus
- Venus

▷ Every planet is a terrestrial planet or is a gas giant.

▷

Off Topic: ACE Editor



> John is an important customer.

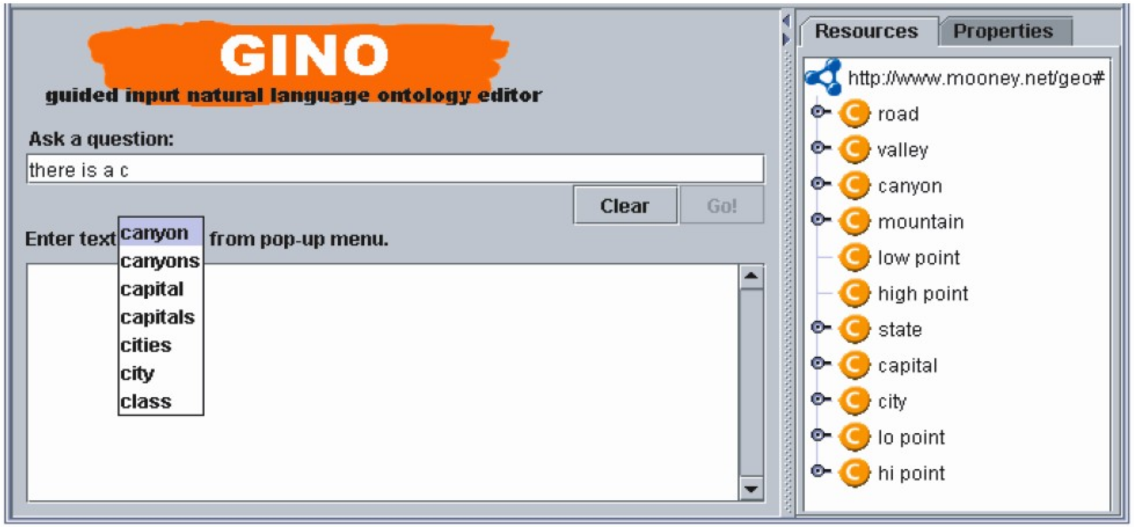
Every important customer can be

Introduction

- (Formal) Controlled Natural Languages (CNL) are designed to be more understandable and more usable by humans than common formal languages.
- But how do we know whether this goal is achieved?
- The only way to find out: User Studies!

Evaluation of CNL Tools

- Many user studies have been performed to evaluate tools that use CNL, e.g. [1].



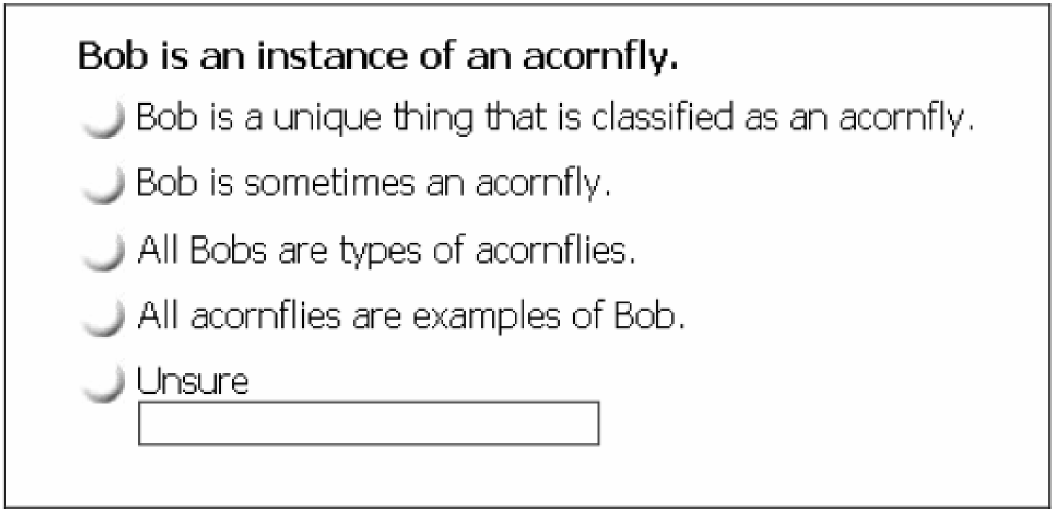
there is a c

- Hard to determine how much the CNL contributes to the understandability
- Hard to compare CNLs to other formal languages because different languages usually require different tools

[1] Abraham Bernstein, Esther Kaufmann. GINO – A Guided Input Natural Language Ontology Editor. ISWC 2006.

Tool-Independent Evaluation of CNLs

- Only very few evaluations have been performed that test a CNL independently of a particular tool.
- [2] presents a paraphrase-based approach: The subjects of an experiment receive a CNL statement and have to choose from four paraphrases in natural English:



Bob is an instance of an acornfly.

○ Bob is a unique thing that is classified as an acornfly.

○ Bob is sometimes an acornfly.

○ All Bobs are types of acornflies.

○ All acornflies are examples of Bob.

○ Unsure

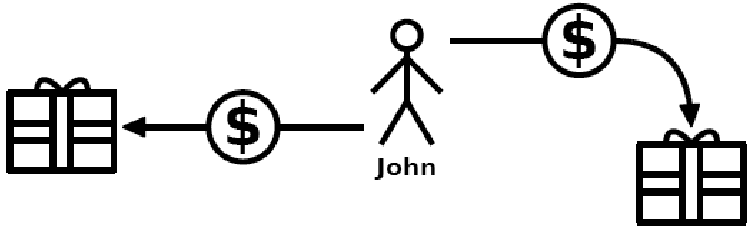
[2] Glen Hart, Martina Johnson, Catherine Dolbear. *Rabbit: Developing a Controlled Natural Language for Authoring Ontologies*. ESWC 2008.

Challenges with Paraphrase-based Approaches

- Ambiguity of natural language
    - One has to make sure that the subjects understand the natural language paraphrases in the right way.
- Does good performance imply understanding?
    - The formal statement and the paraphrases tend to look very similar if both rely on English.
    - One has to exclude that the subjects do the right thing without understanding the statements:
        - Following some syntactic patterns
        - Misunderstanding both – statement and paraphrase – in the same way

My Approach: Ontograph Framework

- Using a simple graphical notation: Ontographs
  - Designed to be used in experiments
  - Idea: Let the subjects perform tasks on the basis of situations depicted by diagrams (i.e. Ontographs).



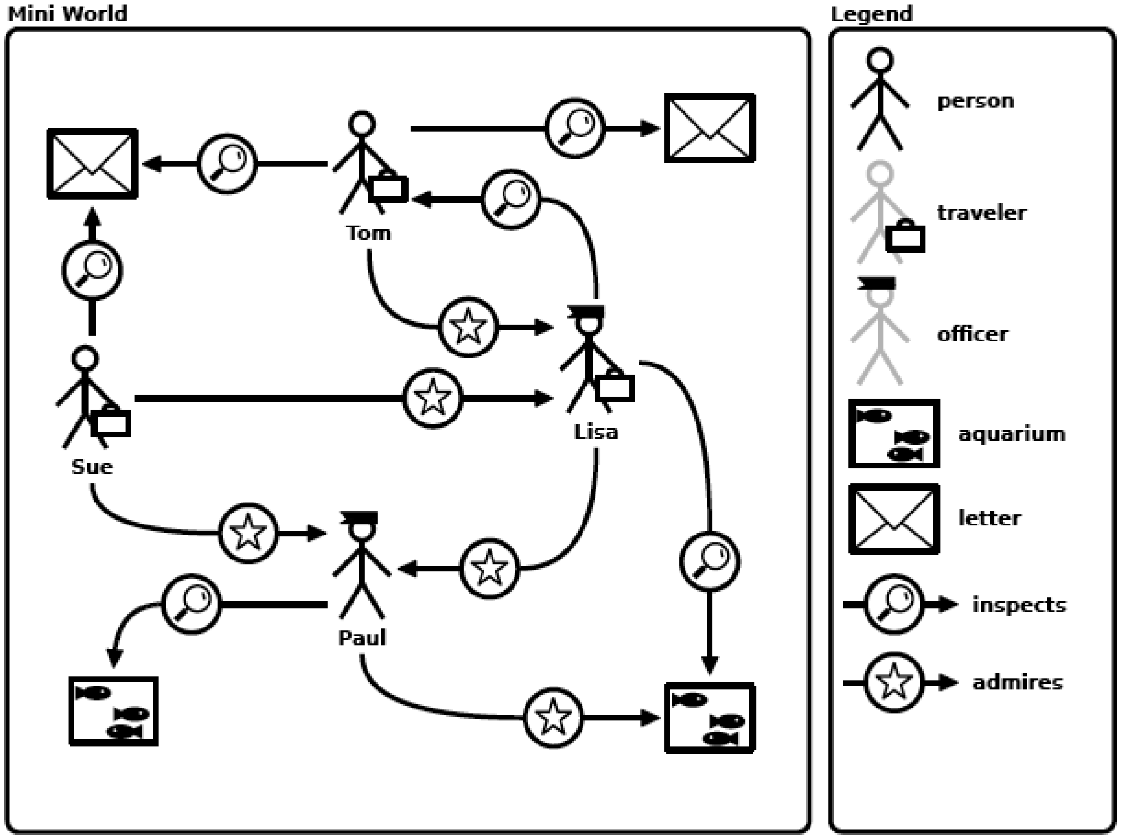
✓ Every present is bought by John.

✗ John buys at most one present.

- Assumption: Ontographs are very easy to understand.

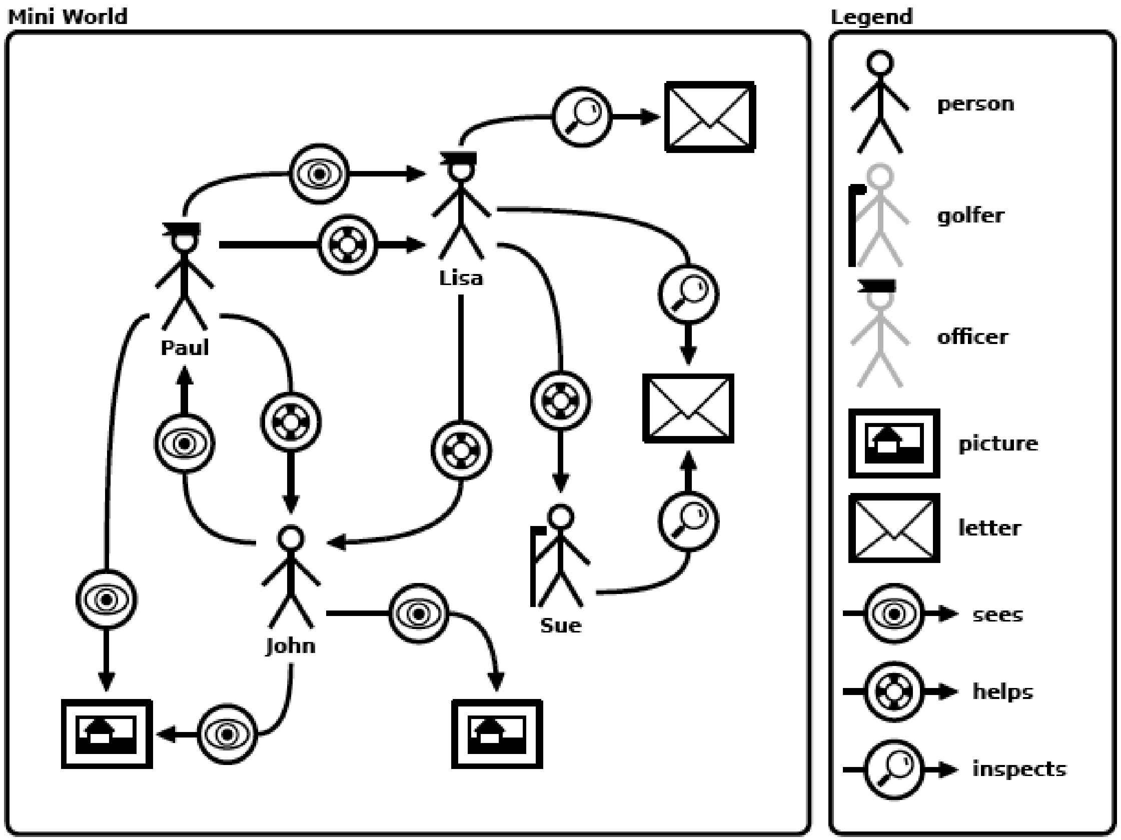
Ontographs

- Ontographs consist of a legend and a mini world.
- The legend introduces types and relations.
- The mini world shows the existing individuals, their types, and their relations.



Ontographs: Properties

- Formal language
- Intuitive graphical icons
- No partial knowledge
- No explicit negation
- No generalization
- Large syntactical distance to textual languages



Experiment: Goal

- The goal of the experiment was to find out whether controlled natural languages are more understandable than comparable common formal languages.
- CNL: Attempto Controlled English (ACE)
- Comparable language: Manchester OWL Syntax [3]:

> »The syntax, which is known as the Manchester OWL Syntax, was developed in response to a demand from a wide range of users, who do not have a Description Logic background, for a “less logician like” syntax. The Manchester OWL Syntax is derived from the OWL Abstract Syntax, but is less verbose and minimises the use of brackets. This means that it is quick and easy to read and write.«

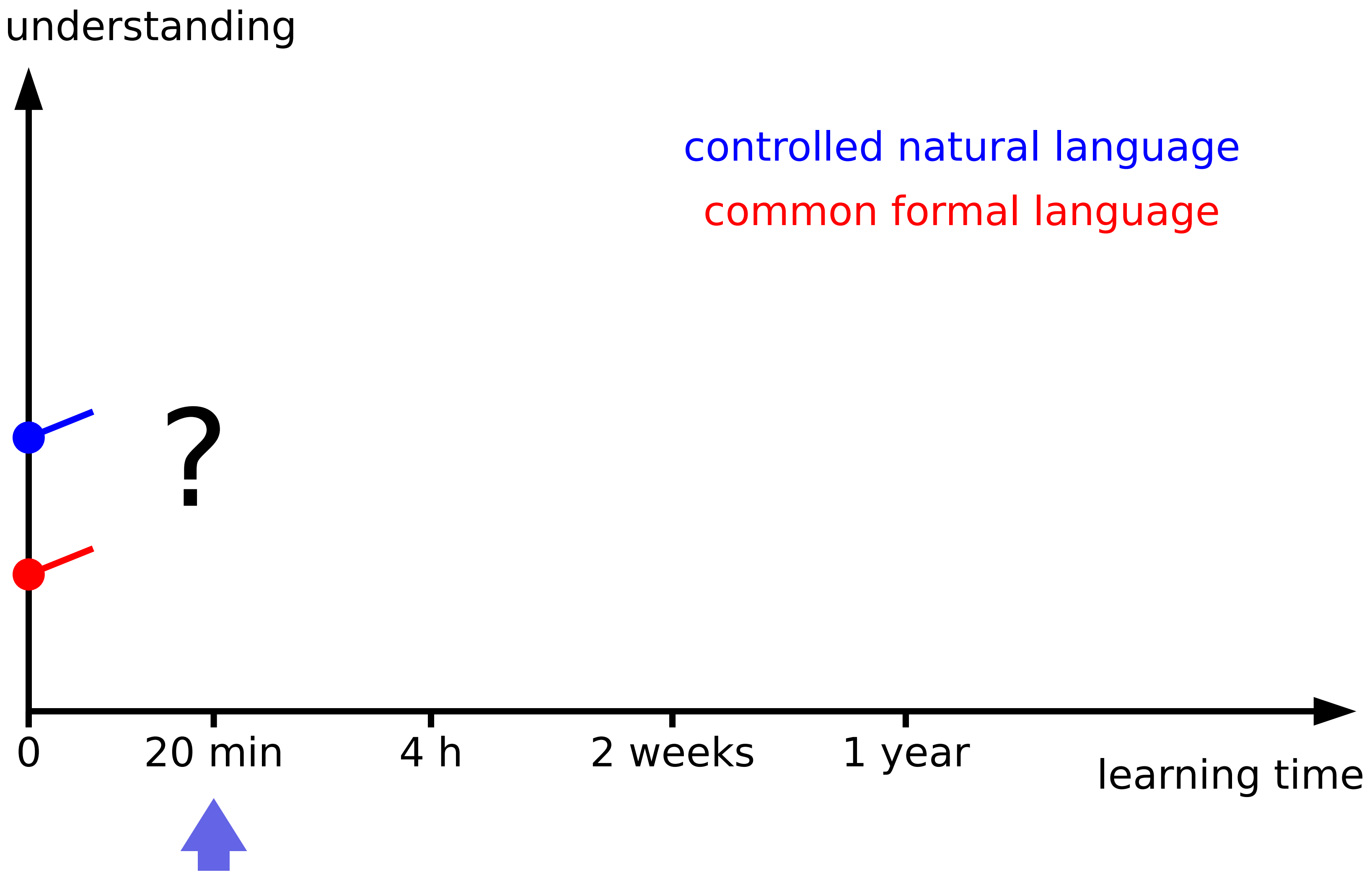
- For a direct comparison, we defined a slightly modified version: MLL (Manchester-like language)

[3] Matthew Horridge, Nick Drummond, John Goodwin, Alan Rector, Robert Stevens, Hai H. Wang. The Manchester OWL Syntax. OWLED 2006.

ACE versus MLL

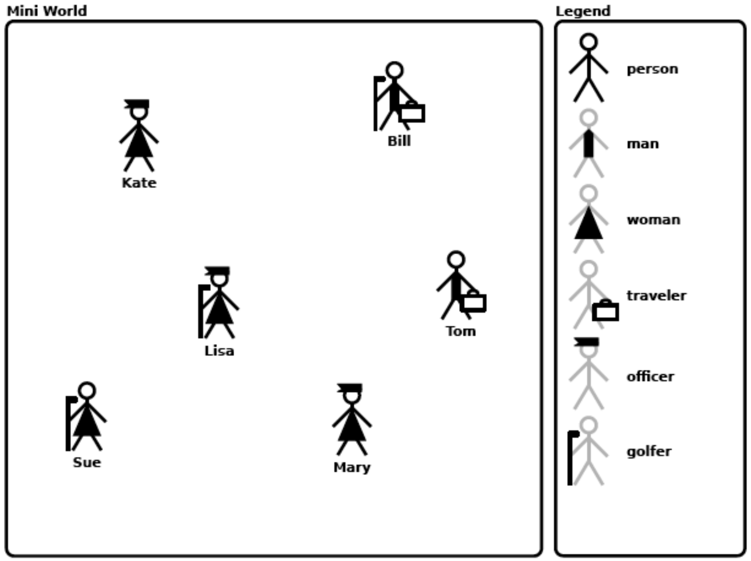
Bill is not a golfer.	Bill **HasType** **not** golfer
No golfer is a woman.	golfer **DisjointWith** woman
Nobody who is a man or who is a golfer is an officer and is a traveler.	man **or** golfer **SubTypeOf** **not** (officer **and** traveler)
Every man buys a present.	man **SubTypeOf** buys **some** present
Lisa helps at most 1 person.	Lisa **HasType** helps **max** 1 person
If X helps Y then Y does not love X.	helps **DisjointWith** **inverse** loves

Learning Time

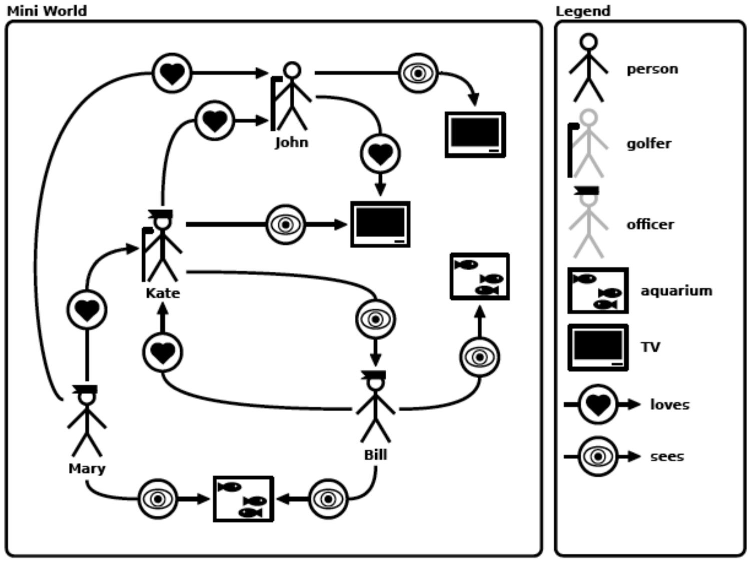


4 Series of Ontographs

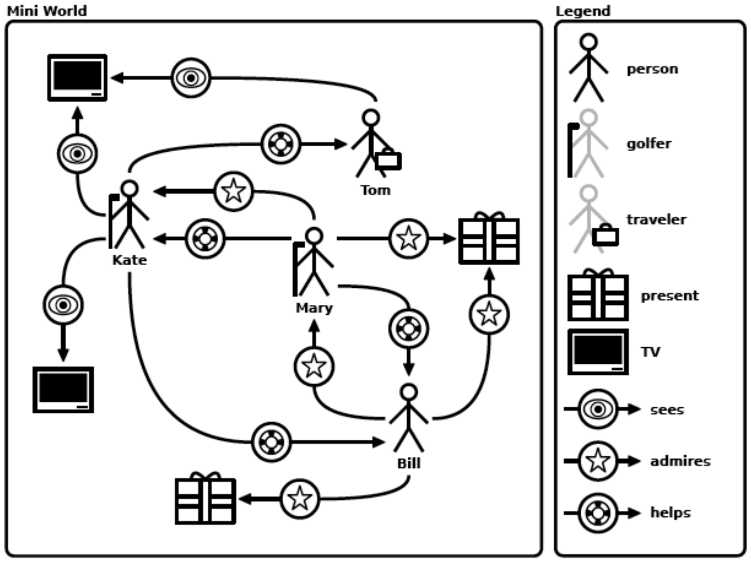
1



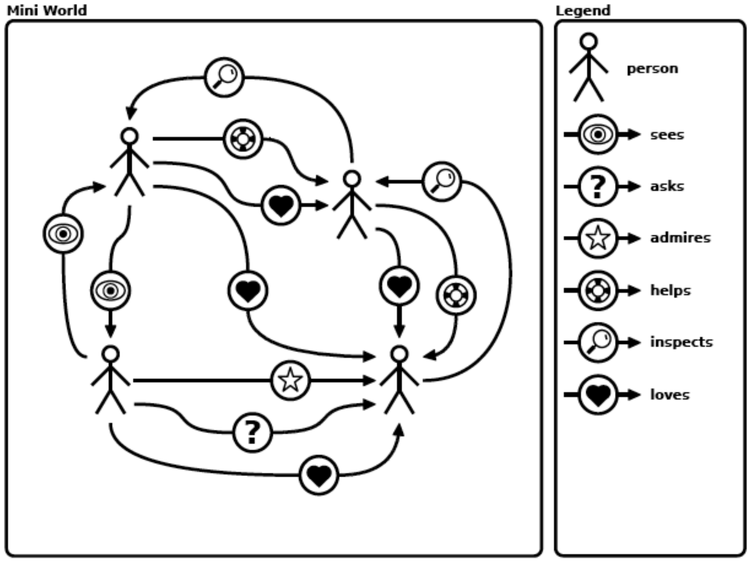
2



3

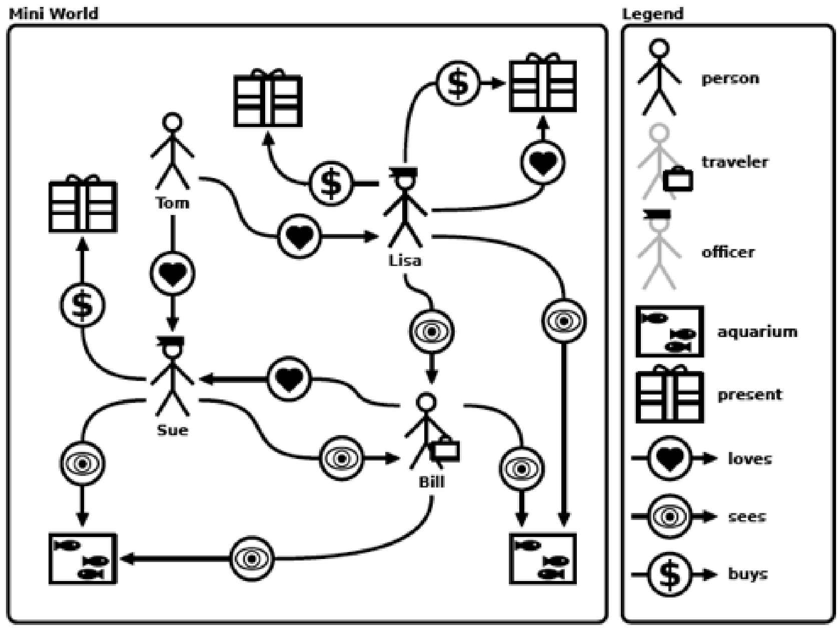


4



Statements in ACE and MLL for each Ontograph

ONTOGRAPH



STATEMENTS

ID	ACE	MLL
1a−	Everything that sees something is an officer.	sees **HasDomain** officer
1b+	Everything that loves something is a person.	loves **HasDomain** person
2a−	Everything that is loved by something is a person.	loves **HasRange** person
2b+	Everything that is bought by something is a present.	buys **HasRange** present
3a−	Everything that loves something is a traveler or is an officer.	loves **HasDomain** traveler **or** officer
3b+	Everything that sees something is an officer or is a traveler.	sees **HasDomain** officer **or** traveler
4a+	Everything that is seen by something is a traveler or is an aquarium.	sees **HasRange** traveler **or** aquarium
4b−	Everything that is bought by something is an aquarium or is an officer.	buys **HasRange** aquarium **or** officer
5a+	Tom loves at least 2 officers.	Tom **HasType** loves **min** 2 officer
5b−	Sue sees at least 2 persons.	Sue **HasType** sees **min** 2 person
6a−	Lisa buys at most 1 present.	Lisa **HasType** buys **max** 1 present
6b+	Bill loves at most 1 person.	Bill **HasType** loves **max** 1 person
7a+	Every traveler sees at least 2 aquariums.	traveler **SubTypeOf** sees **min** 2 aquarium
7b−	Every officer buys at least 2 presents.	officer **SubTypeOf** buys **min** 2 present
8a+	Everything that buys at least 2 presents is an officer.	buys **min** 2 present **SubTypeOf** officer
8b−	Everything that loves at least 2 officers is a traveler.	love **min** 2 officer **SubTypeOf** traveler
9a+	Every officer sees at most 1 aquarium.	officer **SubTypeOf** sees **max** 1 aquarium
9b−	Every person buys at most 1 present.	person **SubTypeOf** buys **max** 1 present
10a−	Everything that is a traveler or that is an officer sees at most 1 aquarium.	traveler **or** officer **SubTypeOf** sees **max** 1 aquarium
10b+	Everything that is an officer or that is a traveler loves at most 1 person.	officer **or** traveler **SubTypeOf** loves **max** 1 person

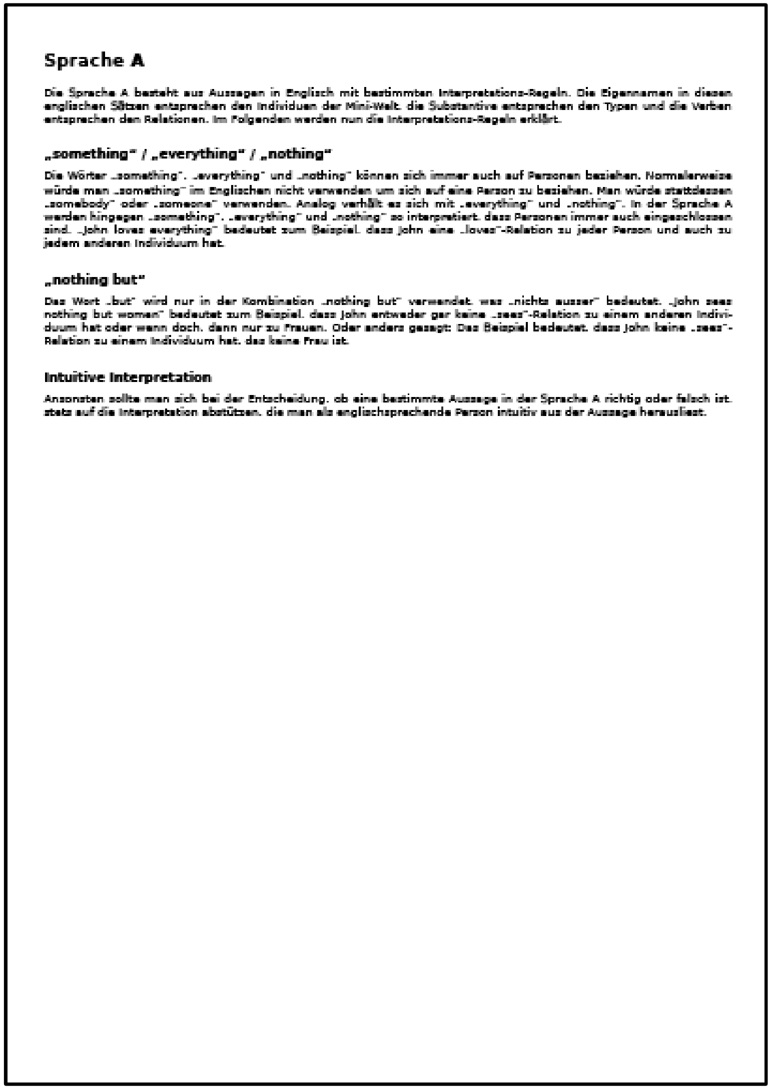
Experiment: Subjects

- Requirements:
  - Students, but no computer scientists or logicians
  - At least intermediate level in written German and English
- Recruitment of 64 subjects:
  - Broad variety of fields of study
  - On average 22 years old
  - 42% female, 58% male
- The subject were equally distributed into eight groups:
  (Series 1, Series 2, Series 3, Series 4) x (ACE first, MLL first)

Experiment: Procedure

- 1. Subjects read an instruction sheet that explains the procedure, the pay-out, and the ontograph notation.
- 2. The subjects answer control questions in order to check whether they understood the instructions.
- 3. During a learning phase that lasts at most 16 minutes, the subjects read a language description sheet (of either ACE or MLL) and see on the screen an ontograph together with 10 statements marked as “true” and 10 marked as “false”.
- 4. During the test phase that lasts at most 6 minutes, the subjects see another ontograph on the screen an have to classify 10 statements as “true”, “false”, or “don't know”.
- 5. The steps 3 and 4 are repeated with the other language.
- 6. The subjects fill out a questionnaire.

Language Instruction Sheets: ACE versus MLL



Sprache A

Die Sprache A besteht aus Aussagen in Englisch mit bestimmten Interpretations-Regeln. Die Eigennamen in diesen englischen Sätzen entsprechen den Individuen der Mini-Welt, die Substantive entsprechen den Typen und die Verben entsprechen den Relationen. Im Folgenden werden nun die Interpretations-Regeln erklärt.

„something" / „everything" / „nothing"

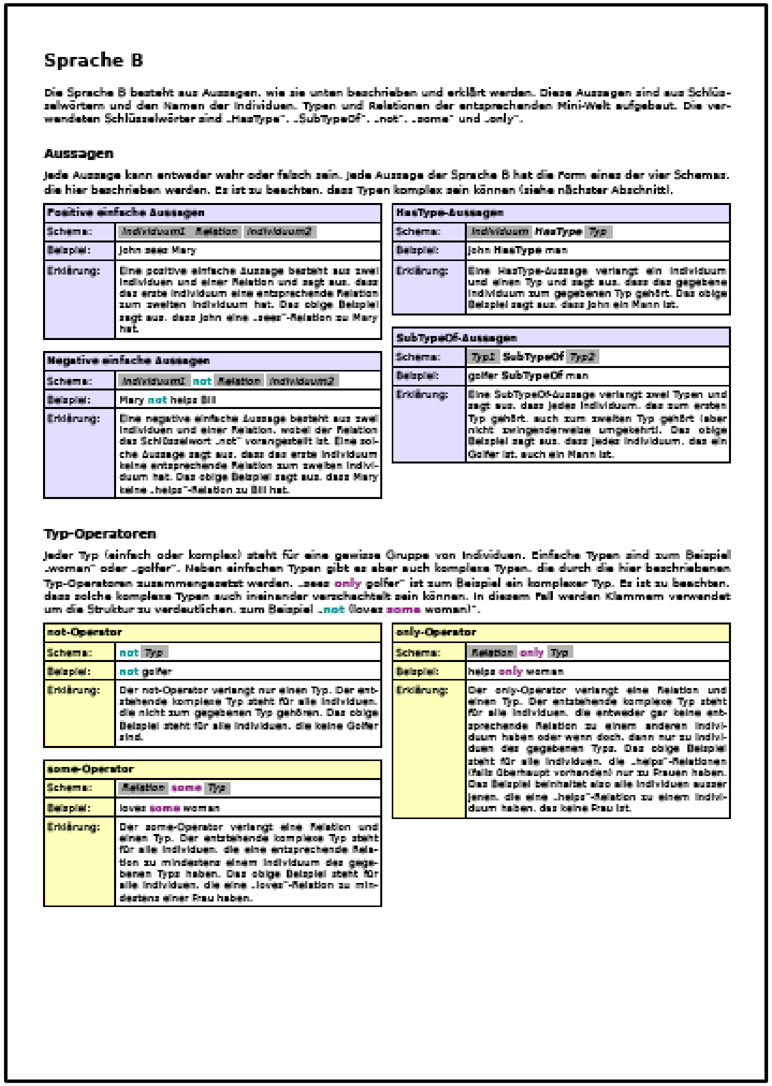
Die Wörter „something", „everything" und „nothing" können sich immer auch auf Personen beziehen. Normalerweise würde man „something" im Englischen nicht verwenden um sich auf eine Person zu beziehen. Man würde stattdessen „somebody" oder „someone" verwenden. Analog verhält es sich mit „everything" und „nothing". In der Sprache A werden hingegen „something", „everything" und „nothing" so interpretiert, dass Personen immer auch eingeschlossen sind. „John loves everything" bedeutet zum Beispiel, dass John eine „loves"-Relation zu jeder Person und auch zu jedem anderen Individuum hat.

„nothing but"

Das Wort „but" wird nur in der Kombination „nothing but" verwendet, was „nichts ausser" bedeutet. „John sees nothing but women" bedeutet zum Beispiel, dass John entweder gar keine „sees"-Relation zu einem anderen Individuum hat oder wenn doch, dann nur zu Frauen. Oder anders gesagt: Das Beispiel bedeutet, dass John keine „sees"-Relation zu einem Individuum hat, das keine Frau ist.

Intuitive Interpretation

Ansonsten sollte man sich bei der Entscheidung, ob eine bestimmte Aussage in der Sprache A richtig oder falsch ist, stets auf die Interpretation abstützen, die man als englischsprechende Person intuitiv aus der Aussage herausliest.



Sprache B

Die Sprache B besteht aus Aussagen, wie sie unten beschrieben und erklärt werden. Diese Aussagen sind aus Schlüsselwörtern und den Namen der Individuen, Typen und Relationen der entsprechenden Mini-Welt aufgebaut. Die verwendeten Schlüsselwörter sind „HasType", „SubTypeOf", „not", „some" und „only".

Aussagen

Jede Aussage kann entweder wahr oder falsch sein. Jede Aussage der Sprache B hat die Form eines der vier Schemas, die hier beschrieben werden. Es ist zu beachten, dass Typen komplex sein können (siehe nächster Abschnitt).

Positive einfache Aussagen	
Schema:	Individuum1 Relation Individuum2
Beispiel:	John sees Mary
Erklärung:	Eine positive einfache Aussage besteht aus zwei Individuen und einer Relation und sagt aus, dass das erste Individuum eine entsprechende Relation zum zweiten Individuum hat. Das obige Beispiel sagt aus, dass John eine „sees"-Relation zu Mary hat.

Negative einfache Aussagen	
Schema:	Individuum1 not Relation Individuum2
Beispiel:	Mary not helps Bill
Erklärung:	Eine negative einfache Aussage besteht aus zwei Individuen und einer Relation, wobei der Relation das Schlüsselwort „not" vorangestellt ist. Eine solche Aussage sagt aus, dass das erste Individuum keine entsprechende Relation zum zweiten Individuum hat. Das obige Beispiel sagt aus, dass Mary keine „helps"-Relation zu Bill hat.

HasType-Aussagen	
Schema:	Individuum HasType Typ
Beispiel:	John HasType man
Erklärung:	Eine HasType-Aussage verlangt ein Individuum und einen Typ und sagt aus, dass das gegebene Individuum zum gegebenen Typ gehört. Das obige Beispiel sagt aus, dass John ein Mann ist.

SubTypeOf-Aussagen	
Schema:	Typ1 SubTypeOf Typ2
Beispiel:	golfer SubTypeOf man
Erklärung:	Eine SubTypeOf-Aussage verlangt zwei Typen und sagt aus, dass jedes Individuum, das zum ersten Typ gehört, auch zum zweiten Typ gehört (aber nicht zwingenderweise umgekehrt). Das obige Beispiel sagt aus, dass jedes Individuum, das ein Golfer ist, auch ein Mann ist.

Typ-Operatoren

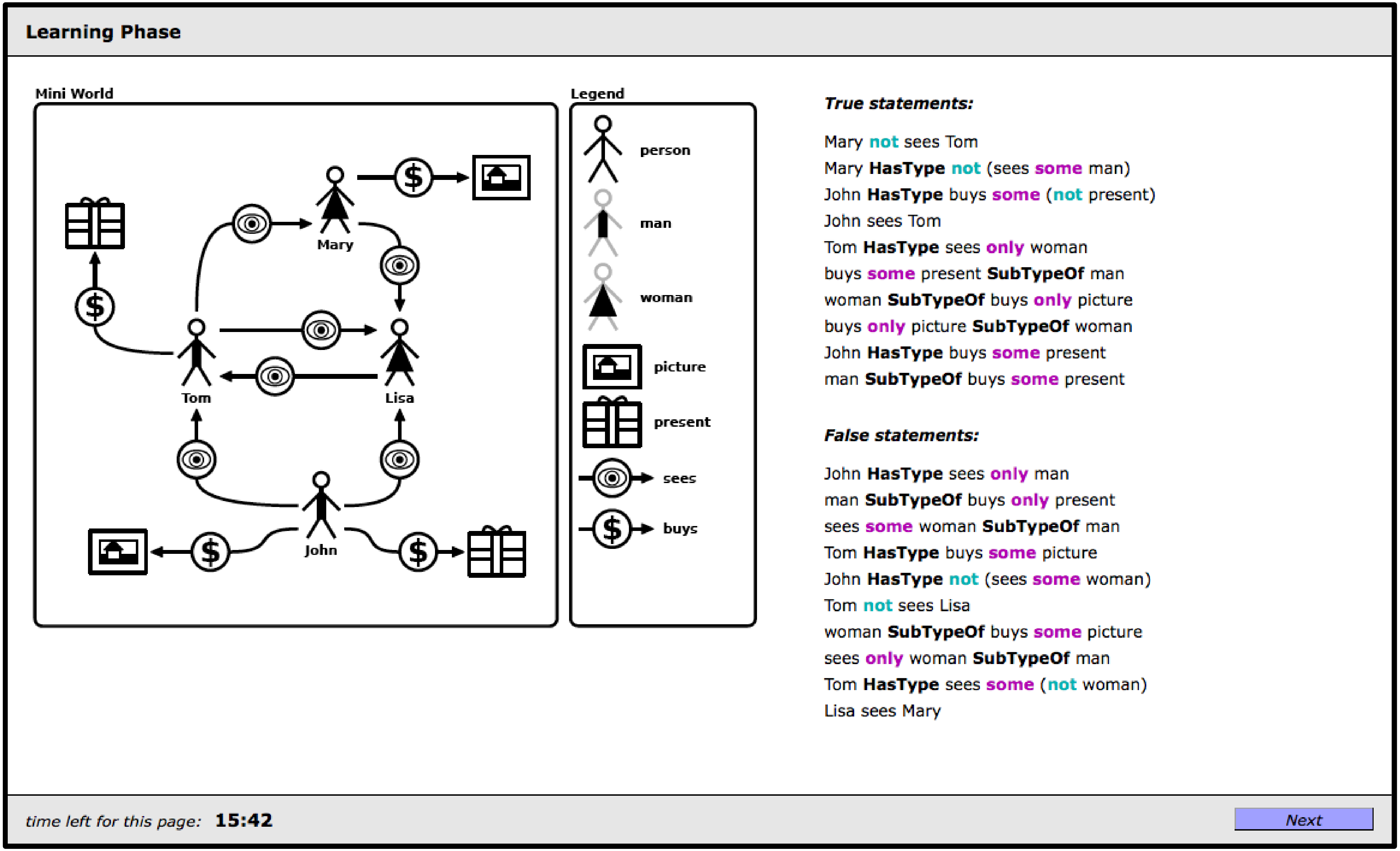
Jeder Typ (einfach oder komplex) steht für eine gewisse Gruppe von Individuen. Einfache Typen sind zum Beispiel „woman" oder „golfer". Neben einfachen Typen gibt es aber auch komplexe Typen, die durch die hier beschriebenen Typ-Operatoren zusammengesetzt werden. „sees only golfer" ist zum Beispiel ein komplexer Typ. Es ist zu beachten, dass solche komplexe Typen auch ineinander verschachtelt sein können. In diesem Fall werden Klammern verwendet um die Struktur zu verdeutlichen, zum Beispiel „not (loves some woman)".

not-Operator	
Schema:	not Typ
Beispiel:	not golfer
Erklärung:	Der not-Operator verlangt nur einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die nicht zum gegebenen Typ gehören. Das obige Beispiel steht für alle Individuen, die keine Golfer sind.

some-Operator	
Schema:	Relation some Typ
Beispiel:	loves some woman
Erklärung:	Der some-Operator verlangt eine Relation und einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die eine entsprechende Relation zu mindestens einem Individuum des gegebenen Typs haben. Das obige Beispiel steht für alle Individuen, die eine „loves"-Relation zu mindestens einer Frau haben.

only-Operator	
Schema:	Relation only Typ
Beispiel:	helps only woman
Erklärung:	Der only-Operator verlangt eine Relation und einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die entweder gar keine entsprechende Relation zu einem anderen Individuum haben oder wenn doch, dann nur zu Individuen des gegebenen Typs. Das obige Beispiel steht für alle Individuen, die „helps"-Relationen (falls überhaupt vorhanden) nur zu Frauen haben. Das Beispiel beinhaltet also alle Individuen ausser jenen, die eine „helps"-Relation zu einem Individuum haben, das keine Frau ist.

Experiment: Learning Phase



True statements:

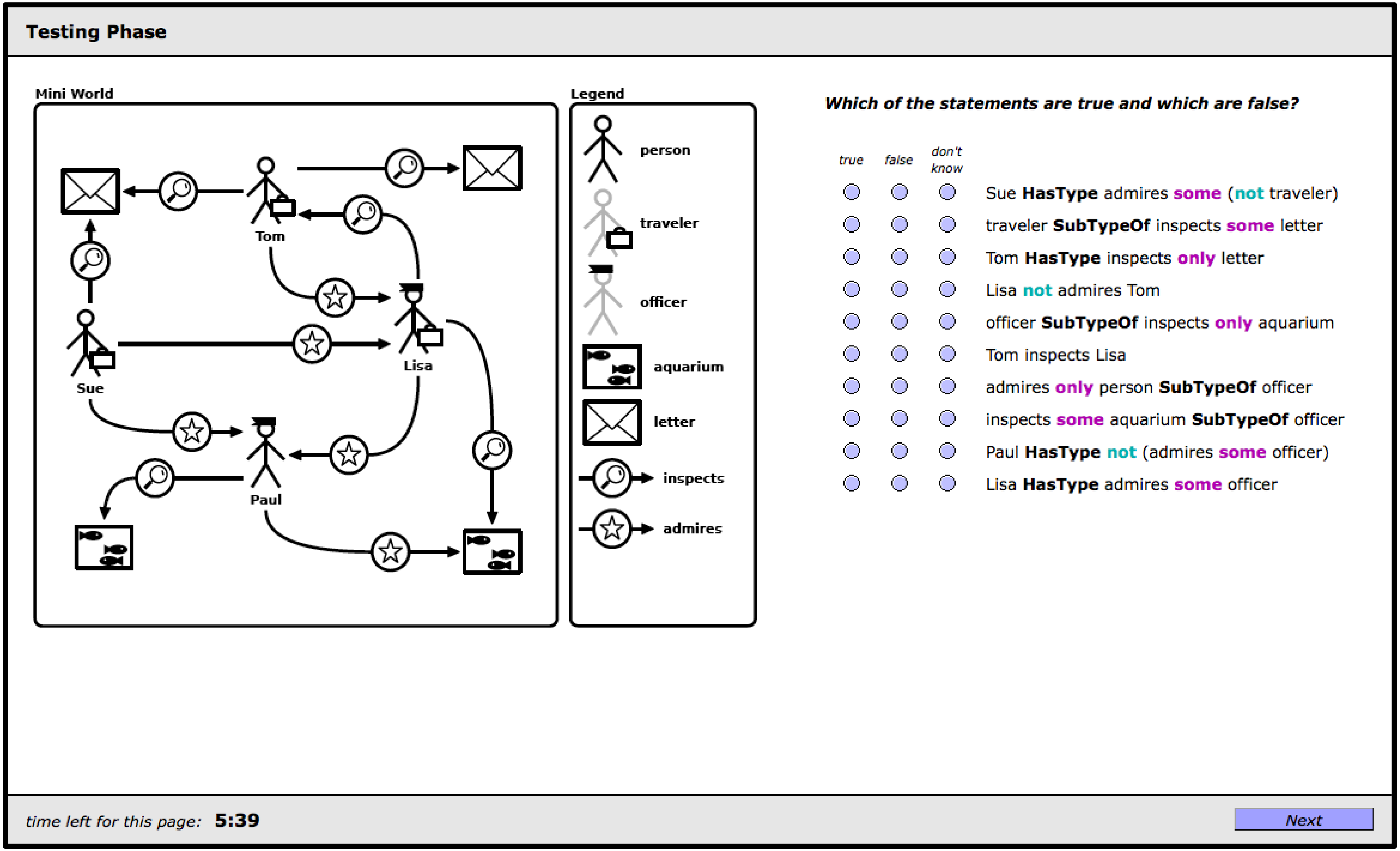
Mary **not** sees Tom
Mary **HasType** **not** (sees **some** man)
John **HasType** buys **some** (**not** present)
John sees Tom
Tom **HasType** sees **only** woman
buys **some** present **SubTypeOf** man
woman **SubTypeOf** buys **only** picture
buys **only** picture **SubTypeOf** woman
John **HasType** buys **some** present
man **SubTypeOf** buys **some** present

False statements:

John **HasType** sees **only** man
man **SubTypeOf** buys **only** present
sees **some** woman **SubTypeOf** man
Tom **HasType** buys **some** picture
John **HasType** **not** (sees **some** woman)
Tom **not** sees Lisa
woman **SubTypeOf** buys **some** picture
sees **only** woman **SubTypeOf** man
Tom **HasType** sees **some** (**not** woman)
Lisa sees Mary

time left for this page: **15:42**

Experiment: Testing Phase



Testing Phase

Which of the statements are true and which are false?

true	*false*	*don't know*	
○	○	○	Sue **HasType** admires **some** (**not** traveler)
○	○	○	traveler **SubTypeOf** inspects **some** letter
○	○	○	Tom **HasType** inspects **only** letter
○	○	○	Lisa **not** admires Tom
○	○	○	officer **SubTypeOf** inspects **only** aquarium
○	○	○	Tom inspects Lisa
○	○	○	admires **only** person **SubTypeOf** officer
○	○	○	inspects **some** aquarium **SubTypeOf** officer
○	○	○	Paul **HasType** **not** (admires **some** officer)
○	○	○	Lisa **HasType** admires **some** officer

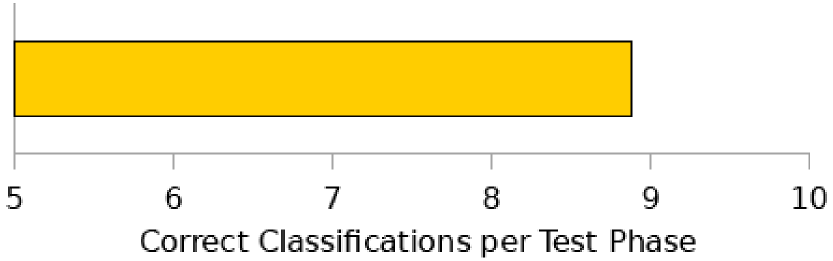
time left for this page: **5:39**

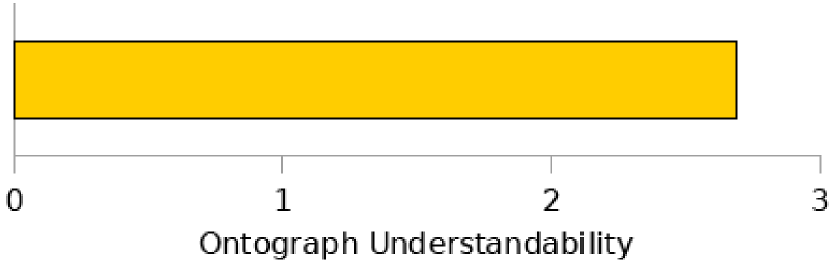
Experiment: Pay-out

- Every subject got 20.00 CHF for participation.
- Furthermore, they got 0.60 CHF for every correctly classified statement and 0.30 CHF for every “don't know”.
- Thus, every subject earned between 20 and 32 CHF.

Evaluation: Ontograph Framework

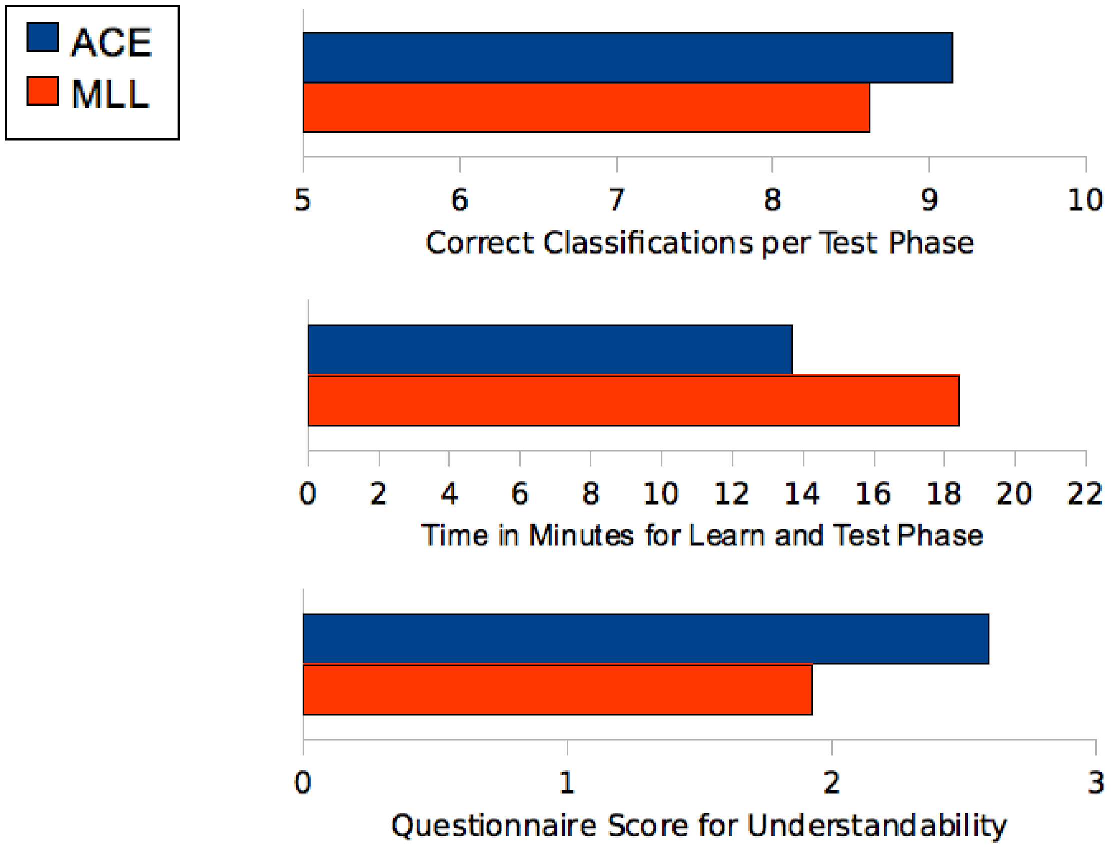
- Did the Ontograph framework work? Answer: Yes!
  - The subjects performed very well in the experiment (8.9 correct classifications out of 10)
  - They found the ontographs very easy to understand (questionnaire score of 2.7 where 0 is “very hard to understand” and 3 is “very easy to understand”)





Evaluation: ACE vs MLL

- Which language performed better?
- Answer: ACE was understood better, within shorter time, and was liked better by the subjects than MLL!



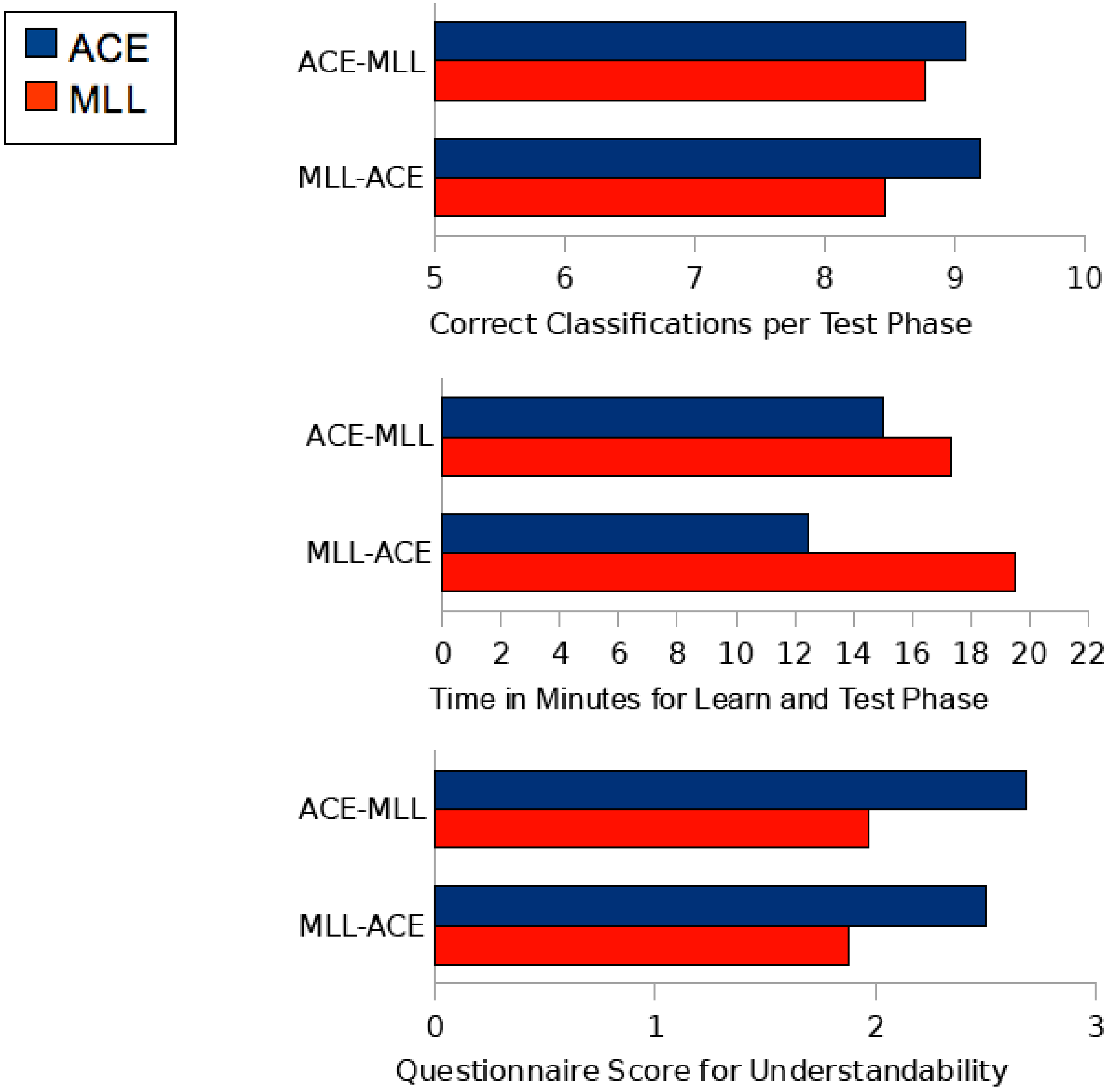
p-values obtained by Wilcoxon singed rank test:

0.003421

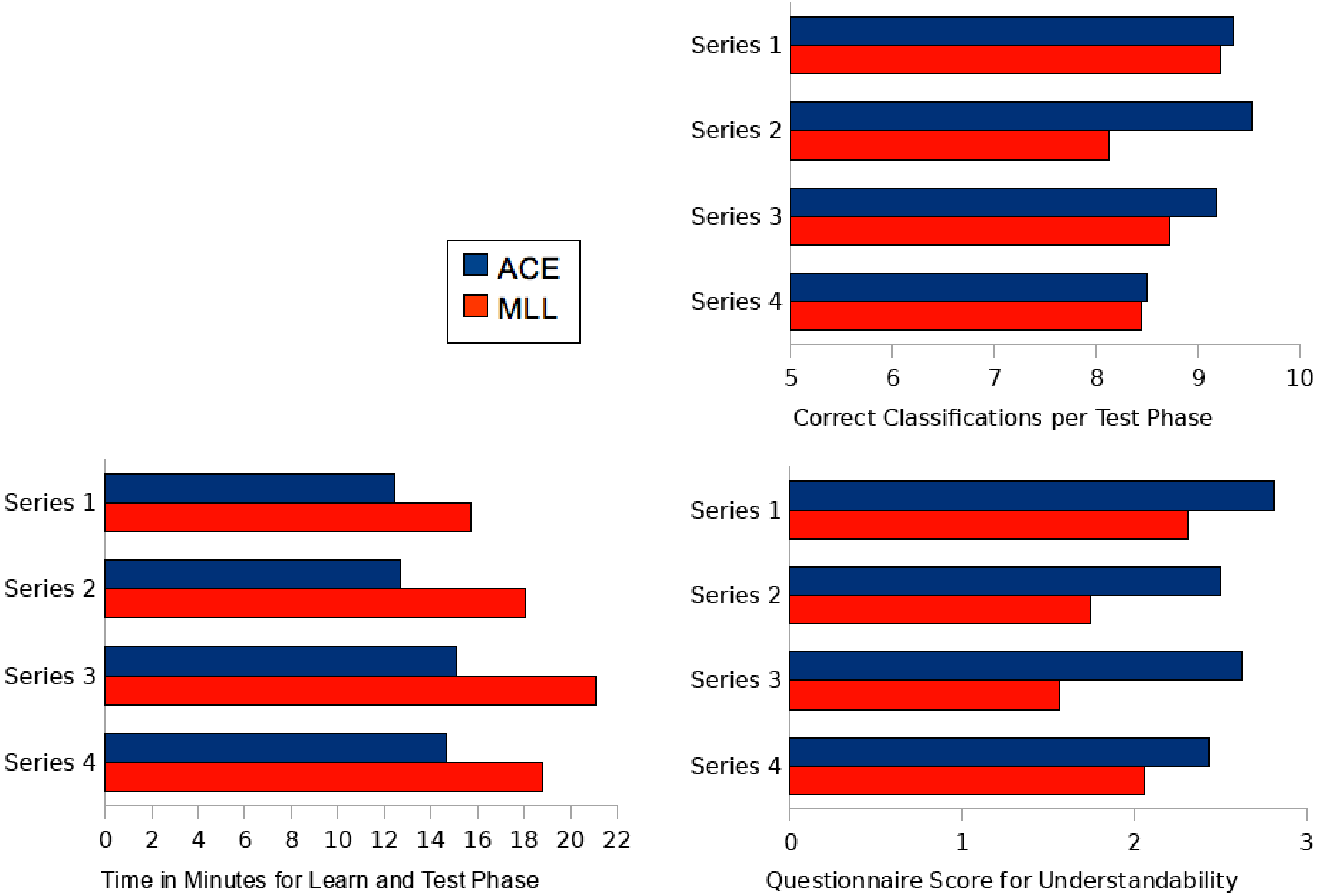
1.493e-10

3.24e-07

Evaluation: First/Second Language



Evaluation: Series 1/2/3/4



Evaluation: Regression

- Regression on the 128 test phase results with the normalized classification score (-5 to 5) as the dependent variable
- Baseline: testing MLL as second language on series 1, male subject of 18 years with good (but not very good) English skills

```
               |               Robust
       sc_norm |      Coef.   Std. Err.      t    P>|t|
---------------|---------------------------------------
           ace |   .5156250   .1800104     2.86   0.006
    first_lang |  -.2187500   .1800104    -1.22   0.229
      series_2 |  -.4802784   .3371105    -1.42   0.159
      series_3 |  -.2776878   .3485605    -0.80   0.429
      series_4 |  -.8795029   .5219091    -1.69   0.097
        female |   .1413201   .2982032     0.47   0.637
  age_above_18 |  -.0724091   .0296851    -2.44   0.018
very_good_engl |   .2031366   .2967447     0.68   0.496
         _cons |   4.302329   .3251371    13.23   0.000
```

Conclusions

- The Ontograph framework seems to be suitable for understandability experiments for CNLs.
- ACE is understood significantly better than MLL.
  - There is no reason to believe that another logic syntax (except CNLs) would have performed better than MLL.
- Furthermore, ACE requires significantly less time to be learned and was liked better by the subjects.

Resources for the Ontograph Framework

- The resources for the Ontograph framework are available freely under a Creative Commons license:
- `http://attempto.ifi.uzh.ch/site/docs/ontograph/`

Thank you for your attention!

Questions/Discussion