

# CS60461

## Lab & Mini-project Assignment

1

## Mechanics

- Everything is to be submitted through Moodle

– ***Everything must be identified with your library/University ID number and name!***

- **Otherwise you are throwing it away!**
- **Include your library card number in all *filenames* AND in *contents* of each zip file as a read me or with the write up**

2

## Summary - Labs

- The labs are for experimentation and learning. The main assessment will be on the miniproject. However, all but the first lab are marked. You should turn in through moodle:
  - 1 zip file each containing the final ontology(ies) and up to one page of A4 describing your work and problems for each of
    - The Normalised Ontology of People (Lab 2), (Pigeon hole 1: "Card sort Ontology") 5%.
    - The DL reasoning problems from Sean Bechhofer (Lab 3 PM) 10 marks (Pigeon hole 2: "DL Problems") 10%
    - The "check your intuitions lab (Lab 3 AM) and the University Ontology lab (lab 4) 2.5%
    - Upper ontology lab and critique of DOLCE 2.5%
    - Miniproject (Mini project pigeon hole) 30%.
- Course work total: 60% (Labs 30%, Mini project 30%)  
(Exam 40%; total assessment 100%)

3

## Summary - All labs: Commenting, documentation and testing

- All ontologies should be fully commented
  - There should be a paraphrase of the intended meaning for each class and additional comments on any special issues or usages of that class. Any classes that are intended as 'axioms' or 'probes' should be clearly commented.
- Ontologies are software
  - *Software is not complete unless documented and provided with test data and a test mechanism!*

4

## Lab 1(30 Jan 2007)

- Work through the Pizza example to define and get correct classification for vegetarian pizza and protein lover's pizza
- Use the Manchester Pizza Finder to see the results of implementing your ontology
  - Instructions on separate sheet
- No lab to turn in.

5

## Lab 2: Knowledge Acquisition 06 Feb 2007

- Take cards for University ontology to produce an ontology for the university including the personnel department's equal opportunities officer
- Group the cards and form initial hierarchies
  - Separate likely primitives, modifiers, roles, defined concepts and properties, classes and individuals
  - Ladder up to provide abstractions as needed
    - And fill in siblings
  - Propose a normalised ontology
    - Classify it to see that it works correctly
      - Provide probe classes to check both classification and unsatisfiability
        - » TURN IN THIS FILE
  - Download the tangled ontology proposed by the personnel department
    - Untangle it so that it fits the rules for normalised ontologies
      - » TURN IN THIS FILE
  - Zip both files with a report and put in pigeon hole 1: Card Sort Ontology<sup>6</sup>

## Lab 3: 20 Feb 2007 DL Exercises

- A(morning)  
Do the "Checking you Intuitions Lab"
  - If time start on Lab 4. Keep your ontology work to bundle with Lab 4: "Intuitions & University Ontology"
- B)  
(afternoon)Do the DL exercises in the separate lab handout.
  - Submit to Pigeon hole 2: "DL Exercises"

7

## Lab 4: 27 Feb 2007 University Ontology

- Download the simple university ontology from the web
- Create a new value partition for the difficulty of a module
- Re-represent the difficulty of a module as a class to give both a reason and an extent of difficulty. (use the n-ary relations pattern)
- Represent the assessment for a module and its parts – lab, lecture, mini-project (see next slide)
- Represent the rules that an assessment is failed if any of its parts are failed and passed if all of its parts are passed
- Create a set of probe classes to test your ontology
- Submit the resulting ontology and your write up along with the "Check your intuitions" exercise from Lab 3 to Pigeon Hole 3: "Intuitions and University Ontology"

8

## Lab 4: more detail

- Represent the assessment for a module and its parts – lab, lecture, mini-project
  - What we want is to represent the notion of an assessment for a module as a whole and of an assessment for each of the module's parts.
  - Each assessment should have
    - a 'type', e.g. 'exam', 'practical', 'written work', etc.
    - A 'value' - a number of points
    - A 'result' - 'distinction', 'pass', 'fail'.
- Define appropriate classes and properties to achieve the above
  - Including all domains, ranges, any property hierarchy needed, and whether properties are functional, inverse functional or transitive

9

## Lab 5: Upper Ontologies

- Explore the top ontology in ~rector/CS646/Labs/Friday
  - Add (if you have not already done so on Thursday)
    - A new kind of building
    - Lecture theatres
    - Clubs – and perhaps student clubs
    - Staff of rank "Senior lecturer" and "Reader"
    - Features ("qualities") for difficulty of modules, comfort of chairs, and quality of lectures.
    - The parts of the building – rooms, doors, lab-spaces, the front of lab spaces, etc.
    - Parts of modules
    - Express the axiom that a module is failed if any of its parts are failed
- You may not complete all of this. Finish what is practical.
- Write a one-page critique of the upper ontology comparing it with Dolce/Ontoclean
- Turn in your critique plus ontology
- Submit to pigeon hole 4: Upper Ontologies

10

## Mini Project

- The basis of the miniproject is to use what you have learned to create an interesting ontology. It is assumed that you will start from the top ontology in .../~alr/CS646 but you may use another
- Our suggested topic is to extend the ontology to cover other modules and aspects of the University
  - Accommodation – buildings, staff, etc
  - Eating facilities –
  - Study facilities – the library, what's in it, how it is used, who uses it, who works there
  - Greater detail on teaching activities – different kinds of labs, lecturers, etc.
- The assignment is to turn in a single zip file containing:
  - The report explaining
    - Purpose and scope of ontology including limitations and workarounds
    - Meta model for ontology -
      - What upper ontology used? If none, why not?
      - Is it normalised? If not, why not?
    - Special features and problems encountered
    - What inference is used in the ontology? (we expect to see some)
    - How to test the ontology
      - I want to some way to see that it does what it claims to do

11

## Miniproject submission

- 1 zip file for your mini project containing
  - **A report of up to five pages documenting your work.**
    - What you have done
    - What you would like to do but couldn't because it was outside the OWL paradigm
    - What you would like to have done but didn't know how to
    - Bugs and problems with the starting ontology
      - Check regularly for updates. We will try to fix bugs as they are found.
    - A sample of how the ontology might be used to mark up some known resources - informally
  - The ontology itself
  - To pigeon hole 5: Miniproject
- **This will be 50% of the course work mark**

12

## Other Mini-project Alternatives

- Extended pizza ontology to baking and selling pizzas
- Parts and wholes ontology for any engineering or similar activity
- Ontology of software design tasks and methodologies – or similar CS topic
- Enriching/untangling some part of the ontology from Wikipedia or Open Directory
- Extending, enriching some other ontology from the web in OWL DL
- If you have a topic you would particularly like to work on, talk to us. Please get permission before you pick an alternative topic.

13

## Marking the Mini Project

- The write up is primary, I expect to know
  - The scope of the ontology – what it is for.
  - The decisions made and the rationale for them
  - The testing procedures and whether it passed them
    - I would rather see an interesting ontology with sensible test procedures that did not quite succeed than a trivial one with no testing procedures that did
  - A description of how it would be used to annotate web pages or other resources, preferably with specific examples and how concepts or statements using concepts from the ontology would be used to describe them.
    - Preferably in OWL or RDF, but since this has been covered only briefly, an informal version will do.

14

## Marking the Miniproject (2)

- The ontology
  - Should use a principled upper ontology – the one supplied or another
    - feel free to simplify, but state in your write up what you did.
  - Should have a normalised domain ontology
    - i.e. domain primitive entities should form disjoint trees. All multiple classification should be the result of inference
  - Should demonstrate interesting use of the reasoner
  - Ideally, should be satisfiable
    - If not, identify problems and efforts made to trace them to source.
      - You may even find a bug, so don't panic, just identify it!

15

Please remember to put your *name and library card number* on all work

Please be sure that Janet has your email address and library card number so we can reach you if there are problems with any assessments.

**Include your library card number in all filenames**

16

## Appendix: to Download and Install Protégé OWL and related

–Summary – you need

- Protégé 4Alpha – complete installation
- GraphViz – please install in default location
  - Should already be installed on most machines  
From <http://www.graphviz.org/>
- Example ontologies
- “Pizza finder” and Tutorial from <http://www.co-ode.org>
- For some applications FaCT++ from <http://owl.man.ac.uk/factplusplus/>

17

## Protégé Problems

- It is still alpha software
- Save early, save often
  - Save-as each time to a new file
- Also easier to debug
  - If it worked before a change and not after, you know where the problem must lie.

18