

# INFS3101 / 7100

## Ontology and the Semantic Web

### Module 6 Subclasses and Subproperties

## Last lecture: Key Terms



- ❖ Complex objects need both **identity** and **unity**. Identification and unification can be **logical** or **lexical**. **Independent** classes can help identify **dependent** classes. A whole can be identified **metonymically**. Identity and unity depend on **context**. Can be **indexical**.
- ❖ **Countable** types have both identity and unity. **Bulk** types lack one. **Containers** can give **pseudo-identity** to a bulk type.

## Structure of ontology

- ❖ Common to represent ontology as a world of objects.
  - Information Spaces, Bob Colomb, invoice#78912
- ❖ Objects are organised into classes
  - Information Spaces is a book
  - Bob Colomb is a person
  - invoice#78912 is an invoice
- ❖ Objects have properties
  - Publisher of book, age of person, date of invoice

## Metaproperties

- ❖ Need to talk about properties, so need **metaproperties**, or properties of properties
- ❖ How do we know an object is an instance of a given class?
- ❖ It will have a particular value of a particular property
  - MInfTech student has program code = 5182
- ❖ Such a property is called a **rigid** property for that class

## More metaproperties

- ❖ If we know an object is an instance of a class what properties do we know it has?
- ❖ The rigid property, of course
- ❖ But there may be other properties which always have the same value if the object is an instance of that class.
  - MInfTech student is an EPSA student
  - MInfTech student has INFS7905 in study plan
- ❖ Called **essential** properties
- ❖ Having parts can be essential properties
- ❖ Being a whole can be essential, too
- ❖ Rigid properties are necessarily essential

## More metaproperties

- ❖ How do we identify an object?
- ❖ By some property values
  - Keys, compound keys
- ❖ These properties have the metaproperty **identity** for the object.
- ❖ How do we tell which parts belong to which object (last week)?
- ❖ Again by values of properties.
- ❖ Called the unifying relation.
- ❖ Metaproperty is **unity**.

## More Metaproperties

- ❖ Data modeling has distinction between mandatory and optional property.
- ❖ Essential and rigid properties must be mandatory
- ❖ Properties with identity and unity generally are also mandatory.

## Classes have subclass structure

- ❖ Every instance of a subclass is an instance of all superclasses
- ❖ Books
  - Used
    - comics
  - New
    - Fiction
      - Australian Fiction
    - Non-fiction
      - Computer books (eg DA Books)
      - Chess books
      - Religious books
      - Occult books

## Subclasses declared or defined

- ❖ Sometimes one class is declared to be a subclass of another. Knowing the subclass, we can infer the superclass.
  - A UQ BInfTech student is a UQ Student
- ❖ Sometimes a subclass is defined by a predicate on a superclass (SQL statement eg).
  - A higher-year student has completed 8 courses
  - Can reason from properties of instance of superclass to subclass membership
- ❖ Superclass **subsumes** subclass

## Properties interact with subclasses

- ❖ Property restrictions on a superclass can never be weakened in a subclass, because if I is an instance of the subclass I is also an instance of the superclass.
  - EG an optional property of a superclass can be mandatory in a subclass, or necessarily absent in a subclass.
  - We could have e-mail address as an optional property of student, and two subclasses
    - E-mail student, where e-mail address is mandatory
    - Non-email student, where e-mail address is absent

## No problem for defined subclasses

- ❖ Instances of a defined subclass must satisfy both the subclass and superclass definition predicates, so can never weaken property restrictions.
  - An e-mail BInfTech student is a BInfTech student
- ❖ But must take care with declared subclasses
  - Clyde is-a elephant
  - Elephant is-a species
  - But is Clyde a species?

## Subsumption and metaproperties

- ❖ Metaproperties can be inherited or introduced
- ❖ A type **supplies** a property if introduced there
- ❖ It **carries** a property if inherited
  - MInfTech student supplies property program = 5182
  - Carries property student name

## Designation of metaproperties

- ❖ Metaproperties designated
  - Identity I
  - Unity U
  - Essential E
  - Rigid R
- ❖ If A is a metaproperty of a type, then
- ❖ +A necessarily used for that purpose for all objects of that type
  - + means necessary
- ❖ -A not necessarily used for that purpose for all objects of type
  - - means not
- ❖ ~A necessarily not used for that purpose for all objects of type
  - ~ means anti-

INFS3101/INFS7100 week 5, 22 November, 2005

Bob Colomb

13

## For BInfTech students

- ❖ Program = 2235 +R, -I
  - (rigid and anti-identity)
- ❖ Name anti-rigid, not identity (-R, -I)
- ❖ Pass INFS3101 not essential (-E)
- ❖ Student number -R (therefore -E), +I
  - Anti-rigid, necessarily identity
- ❖ Faculty = EPSA +E, -R (necessary but not rigid)
- ❖ Course attendance -U

INFS3101/INFS7100 week 5, 22 November, 2005

Bob Colomb

14

## Subsumption constraints

- ❖ Necessarily stronger than possible
  - + stronger than -
- ❖ Necessarily inconsistent with necessarily not
  - + in superclass inconsistent with ~ in subclass
- ❖ So a type +A cannot subsume -A, -A
- ❖ But type -A can subsume +A
- ❖ And -A can subsume +A
- ❖ A metaproperty can be strengthened but not made inconsistent or weakened

INFS3101/INFS7100 week 5, 22 November, 2005

Bob Colomb

15

## Examples

- ❖ +R property applies to all subclasses
- ❖ Program code = 2230 is +R for BInfTech Student
- ❖ Is BBusMan/BInfTech dual degree Student a subclass of BInfTech Student?
- ❖ Has program code = 2222, so no.
- ❖ E-mail address -I for Student
- ❖ But could be +I for Currently Enrolled Student

INFS3101/INFS7100 week 5, 22 November, 2005

Bob Colomb

16

## Example: Set/Instance

- ❖ Clyde identified by name as an elephant
  - Name carries identify for elephants
- ❖ Elephant identified by species name as a species
  - Species name carries identity for species
- ❖ Jumbo, Dumbo and all other elephants have the same species name property.
- ❖ Elephants have different identity properties from species, so subsumption fails
- ❖ Semantics of is-a in this case is instance-set, not subsumption.
- ❖ **Set-instance is not superclass/subclass.**

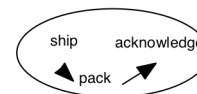
INFS3101/INFS7100 week 5, 22 November, 2005

Bob Colomb

17

## Example: Part/Whole

- ❖ Part not necessarily a subclass of whole
- ❖ A whole may have an essential property which a part cannot have.
- ❖ If rigid property of Activity is completion of instances of Acknowledge, Pack and Ship, none of them can have that property as essential.



Activity

INFS3101/INFS7100 week 5, 22 November, 2005

Bob Colomb

18

## Example: View

- ❖ A view can be a subclass of a table if:
- ❖ View excludes only optional attributes
- ❖ In this case an instance of the view is also a valid instance of the table.

## Properties can have subproperties

- ❖ studyrecord : student -> set of enrolments
  - is a property
- ❖ studyrecordS1.04 is a subproperty
  - Enrolments in semester 1, 2004
- ❖ So is studyrecordINFS
  - Enrolments in INFS courses
- ❖ friend : person -> set of person is a property
- ❖ bestfriend is a subproperty

## Application: Simple Interoperation

- ❖ Purchaser A, Supplier B
- ❖ A has products in table ProdA, id pnoA (+1 for A)
- ❖ B has products in table ProdB, id pnoB (+1 for B)
- ❖ Need correspondence ProdAB, id pnoA x pnoB (+1 for AB)
- ❖ ProdA pnoA      ProdB pnoB
  - 123                      abc
  - 437                      xsq
- ❖ Achieve ProdAB subclass of both ProdA and ProdB
- ❖ Created and maintained by either A, B or both

## Application: Exchange Level

- ❖ Many members, A, B, C, ...
- ❖ Each have product subclass ProdAB, ProdAC, ...
- ❖ Can the exchange make a unified product catalog?
- ❖ Can take union of all Prodx
  - ProdA pnoA    ProdB pnoB    ProdC pnoC
    - 123                      abc                      5d2
    - 437                      xsq
    - 990    7n2
- ❖ Instances of unified catalog identified by equivalence classes of correspondences.
- ❖ Product in unified catalog only if appears in at least one correspondence.
- ❖ Results in a strange declared subclass structure

## Application: Exchange Level

- ❖ More common to have something like ISBN
  - Uniform identity relation
- ❖ Requires work to do this (speech acts)
- ❖ Gives subclass structure depending on who buys or sells what. Still declared.
- ❖ Similar story with members of the exchange
- ❖ Subclasses Purchasers, Suppliers
- ❖ Subclasses not disjoint
- ❖ But still declared. Intersection is defined.
- ❖ Called horizontal partitioning in distributed databases

## Application: Aircraft Example

- ❖ Role - essential attributes
  - Manufacturer - serial#, model#, bill of materials
  - Airline - serial#, config#, bookable seats
  - Production - model#, local#, processes completed
  - Safety inspector - serial#, subsystem history
- ❖ How to integrate? No attributes in common.
- ❖ Is there something outside all systems?

## Application: Can be difficult

- ❖ Order Dept: Customers - can place orders
  - Some customers pay cash
- ❖ Accounts Dept: Accounts - owe money
  - Bad debtors blacklisted
- ❖ Difficult to integrate Customers and Accounts
- ❖ Need a much more complex common superclass

## Application: Aircraft Example

- ❖ For aircraft case, physical airplane is brute fact
- ❖ All systems involve institutional facts where aircraft counts as something in a context
- ❖ Players can agree on an anti-essential property as universal identifier
- ❖ If no property other than identifier is mandatory can consider the different views as subclasses

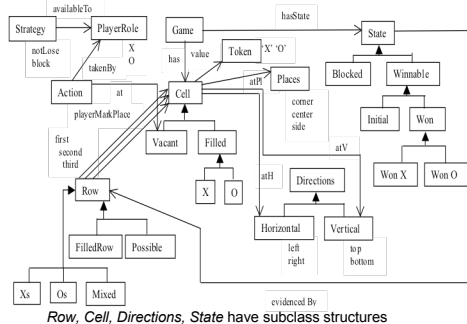
## Semantic Web

- ❖ Semantic web applications can have thousands of players with many roles
- ❖ Often have complex declared subclass and subproperty structures based on which player is responsible for what.
- ❖ Distributed systems generally have much simpler (or at least more coherent) defined subclass structures.

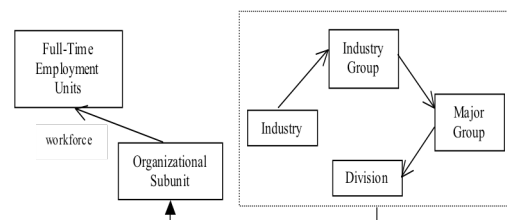
## Example

- ❖ Assume books have *subject* property
  - Assigned by US Library of Congress, say
- ❖ PhilosophyBooks.com sells all and only philosophy books.
  - Defined subclass.
- ❖ DatabaseBooks.com sells only database books, but only those relevant to its clientele.
  - Partially defined (necessary condition only)
  - Partially declared
- ❖ OntologyBooks.com sells books its management thinks will be interesting to ontology developers
  - Declared subclass

## Tic Tac Toe comment



## SIC comment



SIC is part-whole system generating subclass structure for Organisational Subunit

## SNOMED comment

- ❖ SNOMED facets are a whole with parts
- ❖ Each facet is a system like SIC, a whole-part system organising a subclass system for medical interventions.

## Periodic Table, Dimension

- ❖ Period and Group systems of defined subclasses
- ❖ Several other defined subclass systems based on physical properties.
- ❖ Dimensions based entirely on part-whole relationship
  - Distance is not a subclass of SI system
  - Metre is not a subclass of distance
  - Centimetre is not a subclass of Metre

## Summary: Key Terms



- ❖ Common to represent things as **objects** in classes. Objects have **properties**. Superclass **subsumes** subclass, superproperty **subsumes** subproperty. Properties have **metaproperties rigid, essential, identity, unity** which govern subsumption. Subclasses can be **defined** or **declared**.

## Resources

- ❖ **Essential**
  - Notes chapter 6 Subclasses and Subproperties
- ❖ **Further**
  - Guarino and Welty, (2002) Evaluating Ontological Decisions with OntoClean, *Communications of the ACM*, 45(2) on web site.