KR: Tools and Services

COMP30411
Sean Bechhofer
sean.bechhofer@manchester.ac.uk

Practical KR

- So far, we've talked about some of the *theoretical* aspects of Knowledge Representation
  - Philosophical background
  - First Order Logic (propositional/predicate)
- KR is also a *practical* discipline
  - We want to represent knowledge so that we can actually put it to some use.
  - This requires tools and services
- During the second half of the semester we will look in more detail at some of the services needed to support KR

Ontologies and Services

- There is sometime confusion as to what an ontology actually is:
  - A collection of axioms
  - The particular syntactic rendering
  - The file that contains the syntactic rendering
  - The consequences or entailments.
- Ontologies aren't just static objects
- Rather, an ontology is something that we can interact with, or use within the context of some application

The Right Tool for the Right Job

- There are different activities that we might be involved in relating to knowledge representation
  - Different stages in the lifetime of our knowledge
- Knowledge Acquisition
  - What is it that we are going to model?
- Development
  - Building our models
- Maintenance
  - Controlling and dealing with update or change
- Deployment
  - Using the representation in an application
  - Query and Retrieval
- Each of these stages might require tools or services to support it

Logical vs ExtraLogical

- We can consider two different classes of service:
- Logical Services
  - Those that are concerned with the underlying semantics/interpretation of the ontology
- Extra-logical Services
  - Additional services that are useful, generally in the context of some application or task.

Reasoning

- Reasoners provide services that allow us to compute with our representations
- There are a number of different services that can be useful to us
  - Satisfiability Checking
  - Classification
  - Instance Retrieval
- Each of these services is grounded in terms of operations on the underlying logic.
  - Then we know what those services are actually doing.
- How we actually provide these services, and the trade offs in doing so will be covered in the next few lectures

Interpretations Revisited

- We are normally interested in working within the context of some knowledge base that captures our assumptions about the domain we are modelling.
- Recall that a knowledge base K is a collection of axioms

$$K = \{A_1, \ldots, A_n\}$$

- We then define satisfiability with respect to this knowledge base:

$$K \models B$$

- Iff for any interpretations under which all the A_i are true then B is true.
- The intuition here is that B is a consequence of the axioms in K. If K is an accurate approximation of the way the world is, then B should also hold.

COMP30411: Knowledge Representation 7

Interpretations Revisited

- In Description Logics, we have restrictions on the predicates that are allowed.
- Classes: Unary Predicates
  - Interpreted as classes or sets of objects from the domain
- Properties: Binary Predicates
  - Interpreted as sets of pairs of objects in the domain
- Individuals: Constants
  - Interpreted as particular objects in the domain
- The reasoning tasks that we can define are usually described in terms of these interpretations

COMP30411: Knowledge Representation 8

TBox and ABox

- For historical reasons, Description Logic models are often characterised in terms of a *TBox* and an *Abox*
- The TBox is primarily concerned about the definitions of classes and properties in the model.
  - Class definitions
  - Class constraints
  - Property characteristics
- The ABox is concerned with “facts” about individuals in our domain
  - Class membership
  - Relationships with other individuals
- In expressive languages, we can’t actually separate the two, but it is a useful distinction to bear in mind.

Reasoning Tasks: Satisfiability

- A class is said to be satisfiable iff there is *some* model such that the interpretation of that class is non-empty
- Conversely, a class is unsatisfiable iff the interpretation of the class is empty in *all* models.
- Unsatisfiable classes can indicate that there is some inconsistency in our ontology. Or perhaps we have over-constrained the conditions for a class.
- Satisfiability Checking boils down to checking whether the ontology contains contradictory information about particular classes
  - “all cars are red”;
  - “all cars are blue”; ⇒ Contradiction!
  - “red and blue are different”

Reasoning Tasks: Subsumption

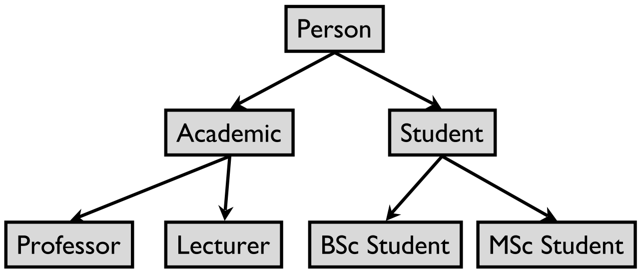
- A class A is said to subsume a class B when it is necessarily the case that everything in B is in A.
- More formally, we can say that A subsumes B iff the interpretation of B is contained in the interpretation of B in *all* models.
- Explicit assertions:
  - All Persons are Animals
- Implicit information
  - All Lecturers are Academics
  - All Academics are Persons
  - We can now make an inference that all Lecturers are Persons
- Subsumption is a formal characterisation of the "kind of" relationship that we often see.

Reasoning Tasks: Equivalence

- Two classes A and B are equivalent when the interpretations of A and B are the same in *all* models.
- For example, if we are modelling polygons, the two classes:
  - Shapes with three sides
  - Shapes with three angles
- Are the same: anything in the interpretation of one must be in the interpretation of the other.

Reasoning Tasks: Classification

- Subsumption Reasoning gives us a mechanism for determining when one class is subsumed by another.
- If we take all the primitive classes in our model and determine which ones are subsumed by others, we can build a classification hierarchy or taxonomy.



Reasoning Tasks: Classification

- Classification may be based on simple assertions (A Lecturer is a Person, a Person is an Animal and so on)
- Classification may also be based on more complex reasoning:
  - Anyone who drives a vehicle must be over 18.
  - Adults are people over 18.
  - A bus is a vehicle
  - Thus, anyone who drives a bus must be an adult.
- This use of definitions and constraints to drive the construction of the hierarchy can be very useful.
  - Particularly if we are trying to maintain a large, complex and possible multi-axial hierarchy.

Why Reasoning?

- Design/Maintenance Phase
  - Warn the user about meaningless or contradictory statements
  - See the consequences of statements made
  - Spot redundancies
  - Support the construction of hierarchies.
- Integration
  - Identifying relationships between alternative conceptualisations
- Use
  - Classification and retrieval of individuals

Query

- The services described help support tasks like construction and maintenance of our knowledge representations
  - They are principally concerned with answering questions about the T-Box.
- If we want to use or deploy this knowledge in applications, it's likely that we need to be able to pose (and answer) queries about the individuals that we are describing: the A-box
  - Who are all the lecturers?
  - Which lecturers teach on third year courses?
- This isn't database lookup…..
  - Queries could be posed relating to incomplete or partial knowledge.
- In terms of implementations, query in Description Logic systems is somewhat less advanced than services relating to the classes or concepts.

Explanation/Debugging

- An additional class of service which is required is explanation
- When models become large and complex, it can be difficult to see why inferences have been drawn.
  - E.g. what were the axioms that led to an inconsistency?
- Providing good, coherent, useful explanations is still an active research area.
- Ontology engineering is much like software engineering.
  - Debugging tools are needed to help us determine what's gone wrong and why

Non Standard Inferences

- In addition to services such as satisfiability checking, subsumption and satisfiability, there are also a number of so-called non standard inferences that have been investigated
- Approximation
  - A translation into a less expressive language with a minimal loss of information
- Least Common Subsumer
  - A description that subsumes a collection of concepts and is the least description to do so
  - Useful in trying to extract commonalities between concept definitions
  - Uninteresting if the language has an *or* operator…

Extra-Logical Services

- Not all useful services are related to the formal semantics of the ontology.
- Lexical services, that map concepts to words or terms are also useful.
  - Providing alternative renderings of concepts, e.g. alternative languages
  - Providing mechanisms for finding concepts.
  - Useful if ontologies are ultimately going to be presented to users.
- Annotations on concepts may also be useful for applications
  - Annotations may not impact on the meaning of concepts or definitions, but supply useful information, e.g. human readable descriptions
  - Identify concepts that are of use for a particular application perspective

COMP30411: Knowledge Representation 19

Extra-Logical Services

- Evaluation Metrics
  - How do we know when our ontology is a *good* ontology?
  - Metrics may help us evaluate the quality of our ontology, but these are often contextual
  - Existing metrics are also often based on structural or syntactic concerns, rather than the semantics.
- OntoClean is a methodology that supports the analysis of ontologies and can help in producing consisent models
  - This is not necessarily about logical consistency in the sense of ensuring that our theory has a model, rather about justifying the
  - We'll see more on OntoClean later

COMP30411: Knowledge Representation 20

Extra-Logical Services

- Query the ontology
  - Considering the ontology as a structure itself
  - When editing or presenting the ontology to the user
  - Determining what has been asserted and what is inferred.

Versioning

- Ontology construction and maintenance are dynamic tasks
- Things change
- We often want to know what has changed and why.
- With logical representations, this may be quite difficult to identify.
  - For example, the syntactic rendering of the ontology may have changed, but the underlying semantics, e.g. collection of entailments, may be exactly the same.
  - Particularly the case when we have RDF-based representations

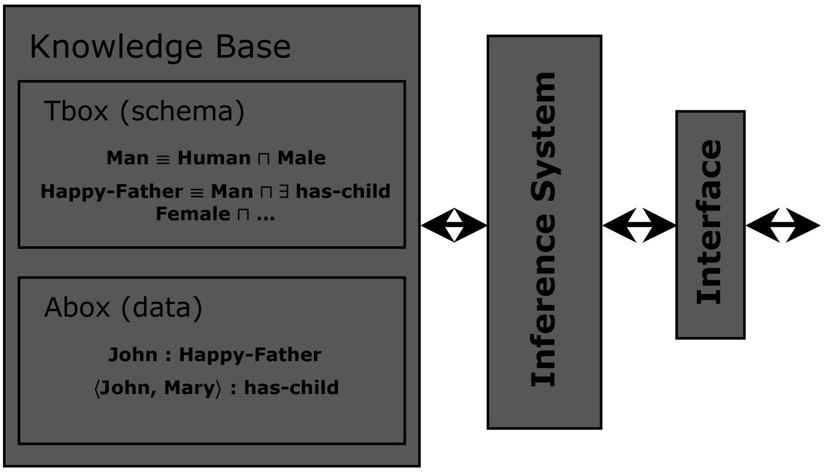
Mapping and Alignment

- In any realistic use of knowledge representation, we are likely to have a number of different models or ontologies describing a domain
- These may differ in
  - Granularity
  - Expressivity
  - Language/Terms used
  - Ontological Perspective
- Services that support the integration and alignment of different ontologies will be essential.
- These may require both logical and extra-logical aspects
  - Axioms spanning ontologies
  - Matching based on lexical annotations

OWL

- The Web Ontology Language (OWL) is a W3C recommendation (standard)
- It provides a language for describing ontologies (models) that integrates with existing Web standards
- The intention is that OWL will be used to describe the data that forms the Semantic Web
  - We'll return to this in later lectures.
- OWL has at its heart a Description Logic
  - One way we can look at OWL is as a standardised syntax for a DL
- A key benefit of providing a standard is that it allows developers to build tools that will (in principle!) interoperate.
  - There are a variety of tools and services that have been developed to work with OWL.

DL Architecture



COMP30411: Knowledge Representation 25

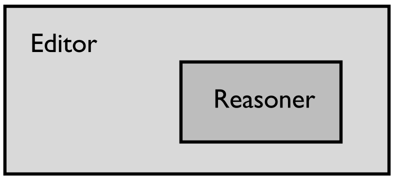
Separation of Concerns

- Reasoning services can be defined independently of the application
  - We see them as useful for tasks such as editing, but they may also be used for other tasks
- Encapsulating a reasoner as a service helps us to separate the responsibilities of components in the bigger picture
- Providing reasoning as a service also helps us in seeing KR as dynamic. It's not just about static representation, but the things that we can do with that representation
- Also facilitates the use of reasoning in other applications

COMP30411: Knowledge Representation 26

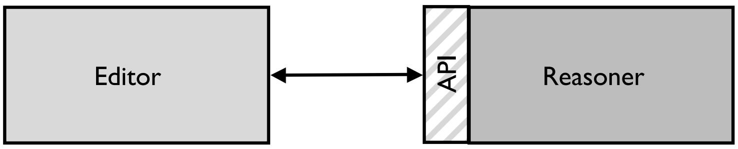
Reasoning Services

- Reasoning embedded within the application



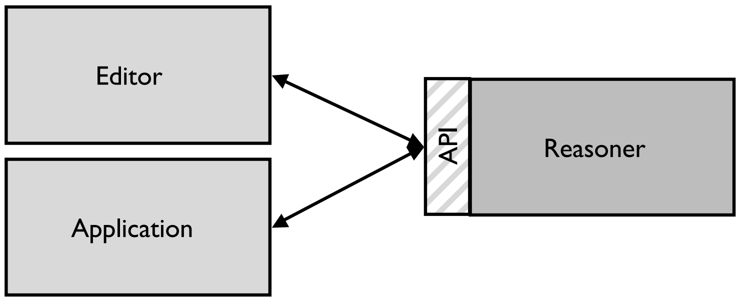
Reasoning Services

- Reasoner available as a separate component via a well defined API
  - E.g. The DIG APIs for DL reasoners



Reasoning Services

- Reasoning services now available to other applications



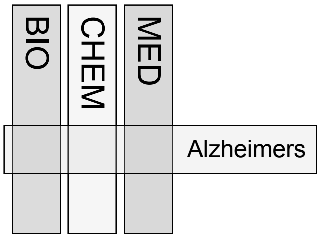
COMP30411: Knowledge Representation 29

Applications

COMP30411: Knowledge Representation 30

Horizontal Integration

- The publisher Elsevier provide a number of information sources organised *vertically* according to journals.
  - Medicine
  - Biology
  - Chemistry
- This makes it difficult to query across different journals, e.g. find information relevant to Alzheimer's, regardless of whether it's in a bio, chemistry or medical journal.
- An ontology helps to resolve the semantic heterogeneity in the data sources.



VO Formation

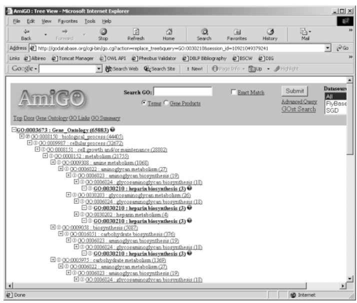
- Virtual Organisation formation is a key aspect in Grid.
- Currently, VOs tend to be specified through an explicit enumeration of the members.
- Semantic technology can allow us to provide a declarative description of the members of a VO and policies concerning their roles.
- We can then use inference to determine whether an actor has access to a particular resource.
- This approach has been explored within the OntoGrid project.

VO Formation

- A richer description of the policies of the VO should allow us to answer a number of questions such as:
  - Given the formal description of the resource X, is it a member of the VO?
  - Which VO resources can the resource X access, and what type of access is it entitled to?
  - What are the capabilities that the resource X must have in order to become a member of the VO?
  - I am a member of the VO. Retrieve all the policies that affect me.
  - Are the set of distributed policies consistent with each other? Is there a resource that cannot be accessed by any member of the VO?

Gene Ontology

- A widely used controlled vocabulary covering functions, processes and locations of gene products.
- Currently delivered as a static, asserted taxonomy.
- But maintenance of such a structure is a difficult task (particularly with multiple inheritance).



Gene Ontology (Next Generation)

- Remodelling **GO** using a richer language (OWL).
- Providing explicit descriptions of concepts.
  - Through examination of the textual descriptions of the terms
  - heparin biosynthesis **defined as**
    biosynthesis and (some acts_on heparin)
  - E.g. biosynthesis that acts on heparin.
  - A **migration path** from simple taxonomy to rich descriptions.
- Use a reasoner to derive inferred subsumption relationships
  - New and "unexpected" relationships are then communicated back to the GO editors.
- Tools for analysis and decomposition of vocabulary terms.
- Reasoning and classification used in the **development** and **maintenance** of the ontology.

GONG

[*chemical*] biosynthesis (GO:0009058)

[i] *carbohydrate* biosynthesis (GO:0016051)

[i] *aminoglycan* biosynthesis (GO:0006023)

[i] *heparin* biosynthesis (GO:0030210)

View 1: Chemicals

View 2: Process

[i] heparin *metabolism* (GO:0030202)

[i] heparin *biosynthesis* (GO:0030210)

[C.J. Wroe, R. Stevens, C.A. Goble, M. Ashburner *A Methodology to Migrate the Gene Ontology to a Description Logic Environment Using DAML+OIL* PSB 2003]

Navigation in the Web

- Commonly, resources are linked by embedding anchors (<**A**> tags) within documents.
  - These links are then traversed while browsing
  - The links between the resources provide utility for both humans and machines.
- Linking has Limitations:
  - Hard coded and Hand crafted.
  - Restricted formats.
  - Requires ownership of the pages.
  - Maintenance and Update.
  - Legacy content can be troublesome
  - Restricted to point-to-point linking

```
<body>
<h1 align=center><a name="Cellular_Respiration">Cellular Respiration</a></h1>
<table border=1 align=right>
<tr><td align=center><b>Index to this page</b></td></tr>
<tr><td>
<ul>
<li><a href="#Mitochondria">Mitochondria</a>
<li><a href="#The_Citric_Acid_Cycle">The Citric Acid Cycle</a>
<li><a href="#The_Respiratory_Chain">The Respiratory Chain</a>
<li><a href="#chemiosmosis">Chemiosmosis in Mitochondria</a>
<li><a href="#36">How many ATPs?</a>
<li><a href="#mtdna">Mitochondrial DNA (mtDNA)</a>
</ul>
</td></tr></table>

<p>Cellular respiration is the process of oxidizing food molecules,
like glucose, to carbon dioxide and water. The energy released is
trapped in the form of <b><a href="../A/ATP.html">ATP</a></b> for use
by all the energy-consuming activities of the cell.</p>

 The process occurs in two phases:
<ul>
<li><b>glycolysis</b>, the breakdown of glucose to pyruvic acid
<li>the complete <b>oxidation of pyruvic acid</b> to carbon dioxide
and water
</ul>

In eukaryotes, glycolysis occurs in the cytosol. (<a
href="../G/Glycolysis.html">Link to a discussion of
glycolysis</a>). The remaining processes take place in
<b>mitochondria</b>.
```

COMP30411: Knowledge Representation 37

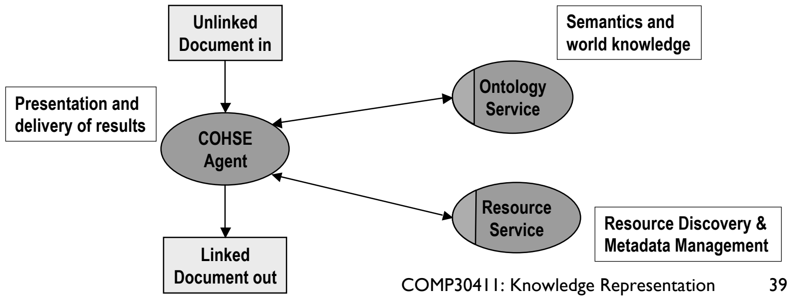
Open Hypermedia

- Open Hypermedia systems seek to address some of these issues
- Links as first class citizens
  - Separation of the links from the source documents
  - Links can be stored, shared, instrumented
- Generic Linking
  - Link sources are determined using some selection, e.g. a word or phrase.
- Specific Linking
  - Link sources are given by addressing a particular document fragment (e.g. using XPointer or some similar mechanism).

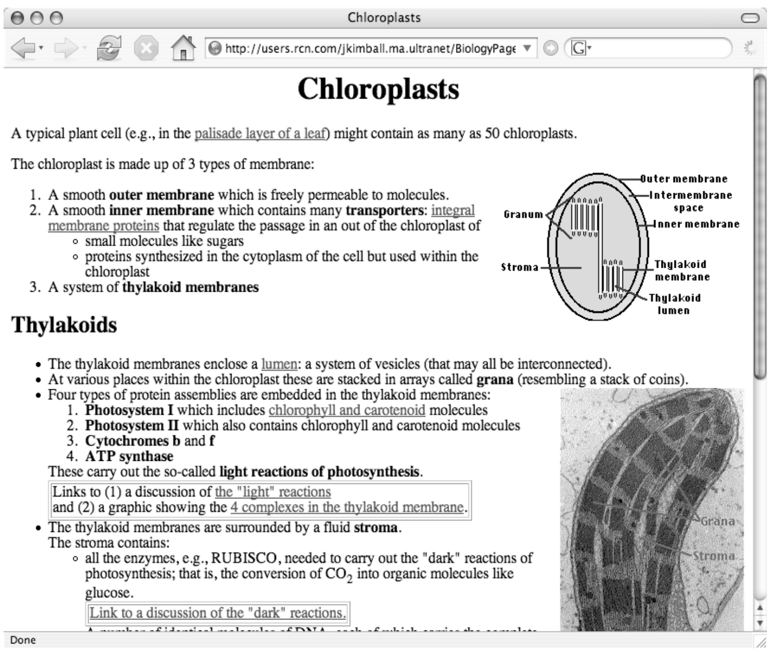
COMP30411: Knowledge Representation 38

Conceptual Open Hypermedia

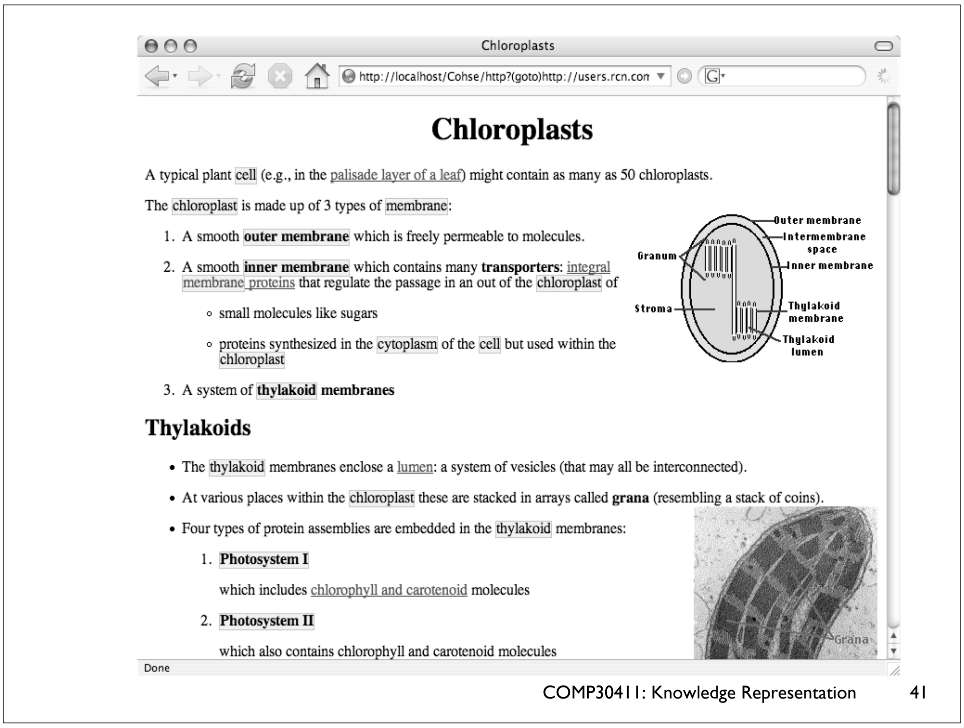
- By adding ontologies into the mix, we can use
  - lexical information in the ontologies to determine link source anchors
  - annotations to determine link targets
  - the ontology structure to aid navigation.

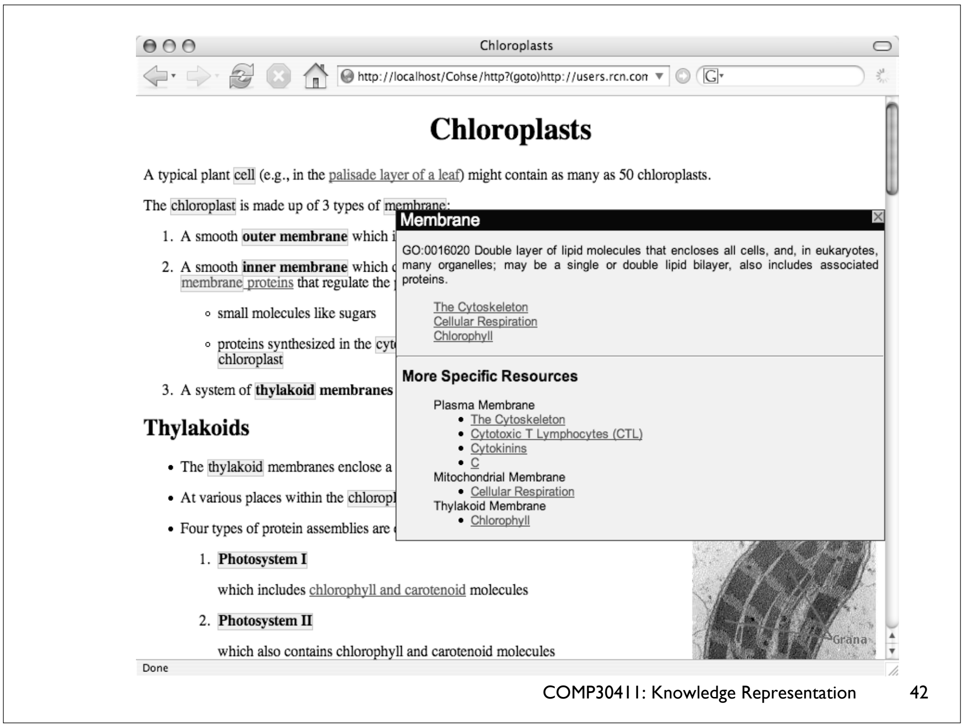


COMP30411: Knowledge Representation 39

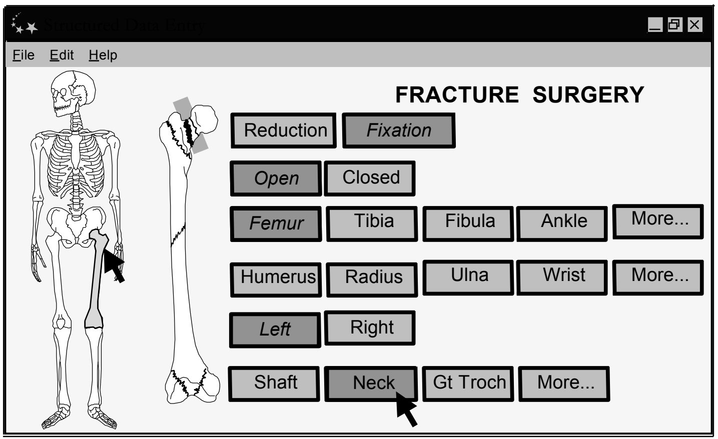


COMP30411: Knowledge Representation 40


Chloroplasts
http://localhost/Cohse/http?(goto)http://users.rcn.con
Chloroplasts
A typical plant cell (e.g., in the palisade layer of a leaf) might contain as many as 50 chloroplasts.
The chloroplast is made up of 3 types of membrane:
1. A smooth outer membrane which is freely permeable to molecules.
2. A smooth inner membrane which contains many transporters: integral membrane proteins that regulate the passage in an out of the chloroplast of
small molecules like sugars
proteins synthesized in the cytoplasm of the cell but used within the chloroplast
3. A system of thylakoid membranes
Outer membrane
Intermembrane space
Inner membrane
Granum
Stroma
Thylakoid membrane
Thylakoid lumen
Thylakoids
The thylakoid membranes enclose a lumen: a system of vesicles (that may all be interconnected).
At various places within the chloroplast these are stacked in arrays called grana (resembling a stack of coins).
Four types of protein assemblies are embedded in the thylakoid membranes:
1. Photosystem I
which includes chlorophyll and carotenoid molecules
2. Photosystem II
which also contains chlorophyll and carotenoid molecules
Grana
Done
COMP30411: Knowledge Representation
41


Chloroplasts
http://localhost/Cohse/http?(goto)http://users.rcn.con
Chloroplasts
A typical plant cell (e.g., in the palisade layer of a leaf) might contain as many as 50 chloroplasts.
The chloroplast is made up of 3 types of membrane:
Membrane
GO:0016020 Double layer of lipid molecules that encloses all cells, and, in eukaryotes, many organelles; may be a single or double lipid bilayer, also includes associated proteins.
The Cytoskeleton
Cellular Respiration
Chlorophyll
More Specific Resources
Plasma Membrane
The Cytoskeleton
Cytotoxic T Lymphocytes (CTL)
Cytokinins
C
Mitochondrial Membrane
Cellular Respiration
Thylakoid Membrane
Chlorophyll
Thylakoids
1. Photosystem I
which includes chlorophyll and carotenoid molecules
2. Photosystem II
which also contains chlorophyll and carotenoid molecules
Grana
Done
COMP30411: Knowledge Representation
42

User Interfaces using Composition



•**Fixation of open fracture of neck of left femur**

COMP30411: Knowledge Representation 43

Protein Classification

- ***A Little Semantic Web Goes a Long Way in Biology***
  - Wolstencroft et.al. ISWC2005
- Given a protein (with a partial description), find the most specific class of proteins that it is an instance of.
- Uses some basic SW technology
  - An OWL Ontology
    - Captures the expert knowledge
  - The Instance Store (IS)
    - Allows storage of a large number of instance descriptions
- Automated methods used to extract the protein data.
  - IS then automates the classification.
- Performs as well as humans.
  - Clear and explicit evidence for the classification

COMP30411: Knowledge Representation 44

Summary

- Ontologies/Representations are dynamic
- We need to consider what we're going to do with them as well as what goes into them
- Logical services relate to the underlying formalism/semantics
  - Classification
  - Satisfiability Checking
- Extra Logical services
  - Annotation
  - Quality
  - Lifecycle