THE UNIVERSITY of EDINBURGH
informatics

Semantic Web Systems

Ontological Engineering

Jacques Fleuriot

School of Informatics

So far in this course…

- Languages and formalisms for:
  - making assertions about the world

    :JamesDean :playedIn :Giant .
  - defining ontologies

  :Actor rdf:type rdfs:Class .

  :Actor rdfs:subClassOf :Performer .
- But how do we go about developing ontologies?

In this lecture

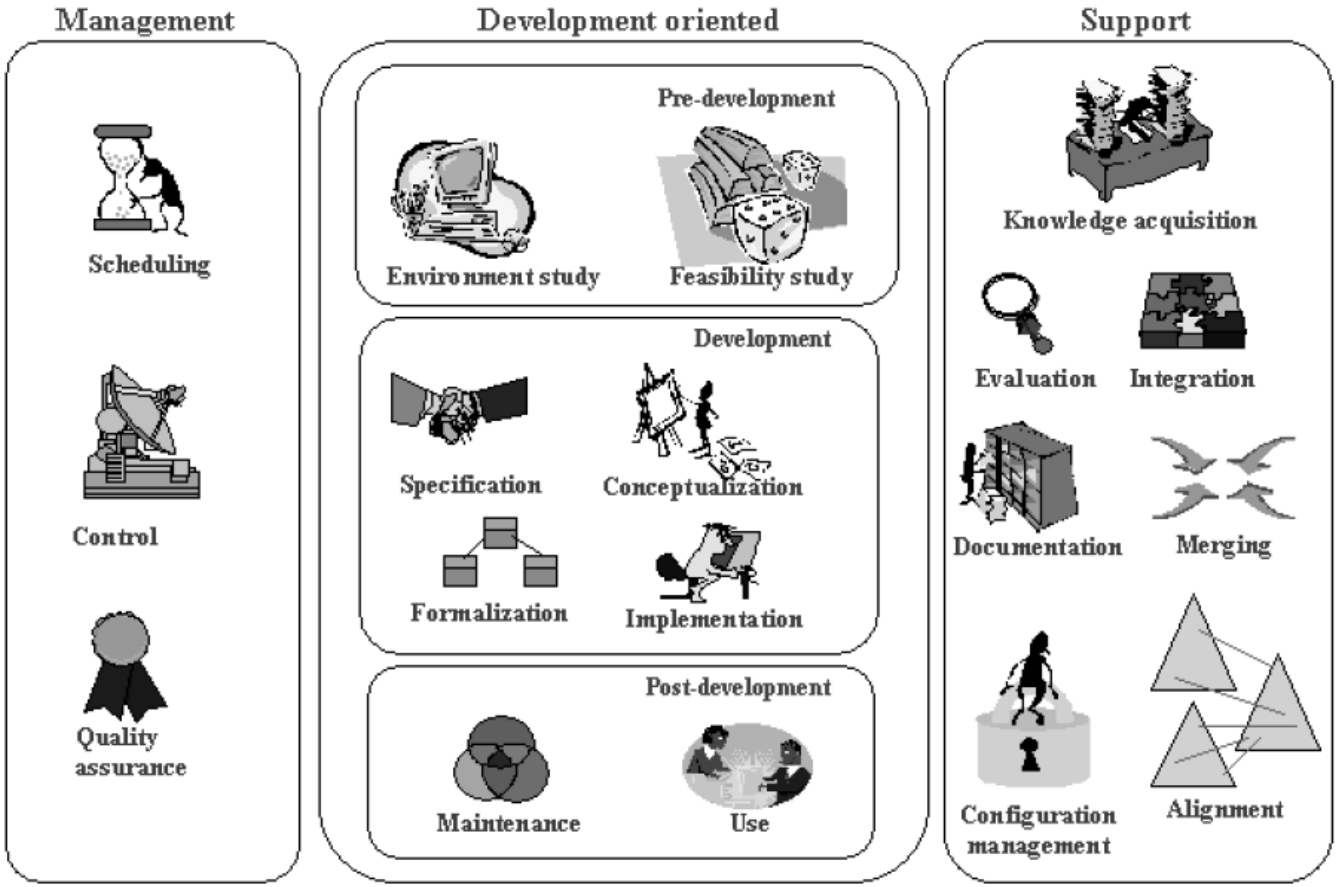
- Methodologies for developing ontologies.
- Tips & considerations.
- Tools.

Software Development Process

- Requirements.
- Design.
- Implementation.
- Testing.
- Maintenance.

Similar story with ontologies!

Ontology Development Process



Source: Ontological Engineering

Methodologies

- Uschold and King's method.
- METHONTOLOGY.
- Grüninger and Fox's methodology.
- On-To-Knowledge.
- Noy and McGuinness.
- NeOn Methodology.
- and many more…

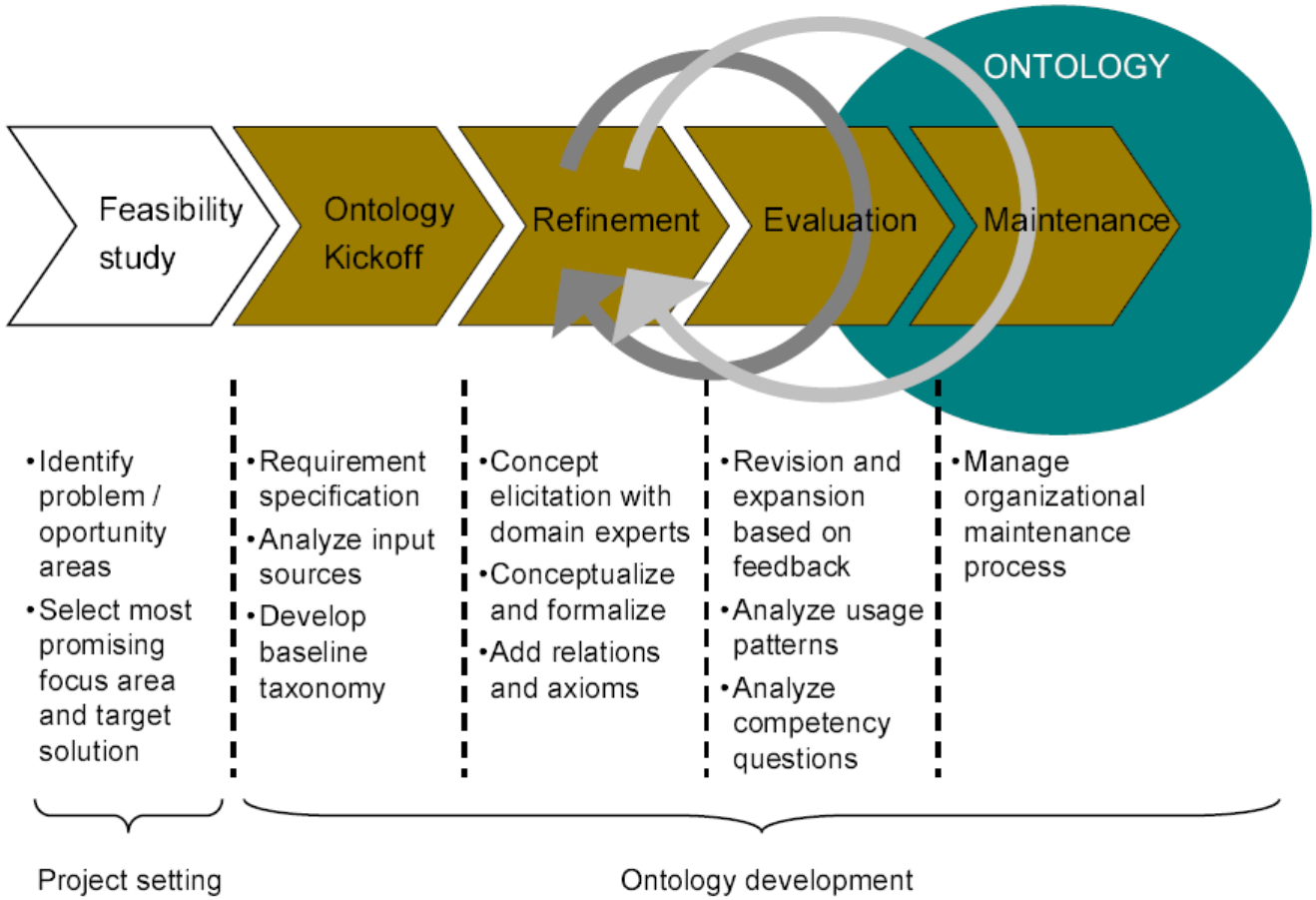
Noy and McGuinness (2001)

1. Determine the domain and scope of the ontology.
2. Consider reusing existing ontologies.
3. Enumerate important terms in the ontology.
4. Define classes and the class hierarchy.
5. Define the properties of classes.
6. Add constraints to the properties.
7. Create instances.

Uschold and King (1995)

- Identify purpose and scope.
- Build the ontology:
  - Capture.
  - Coding.
  - Integrating.
- Evaluation.
- Documentation.

On-To-Knowledge



Source: Staab et al (2001)

Requirement analysis

- Modelling formalism:
  - Is semantic modelling needed/appropriate?
  - Representation based on formal logic reasonable?
  - Which tool/representation language?
- Ontology requirements:
  - Domain? Scope?
  - Granularity?
  - Purpose? Tasks?

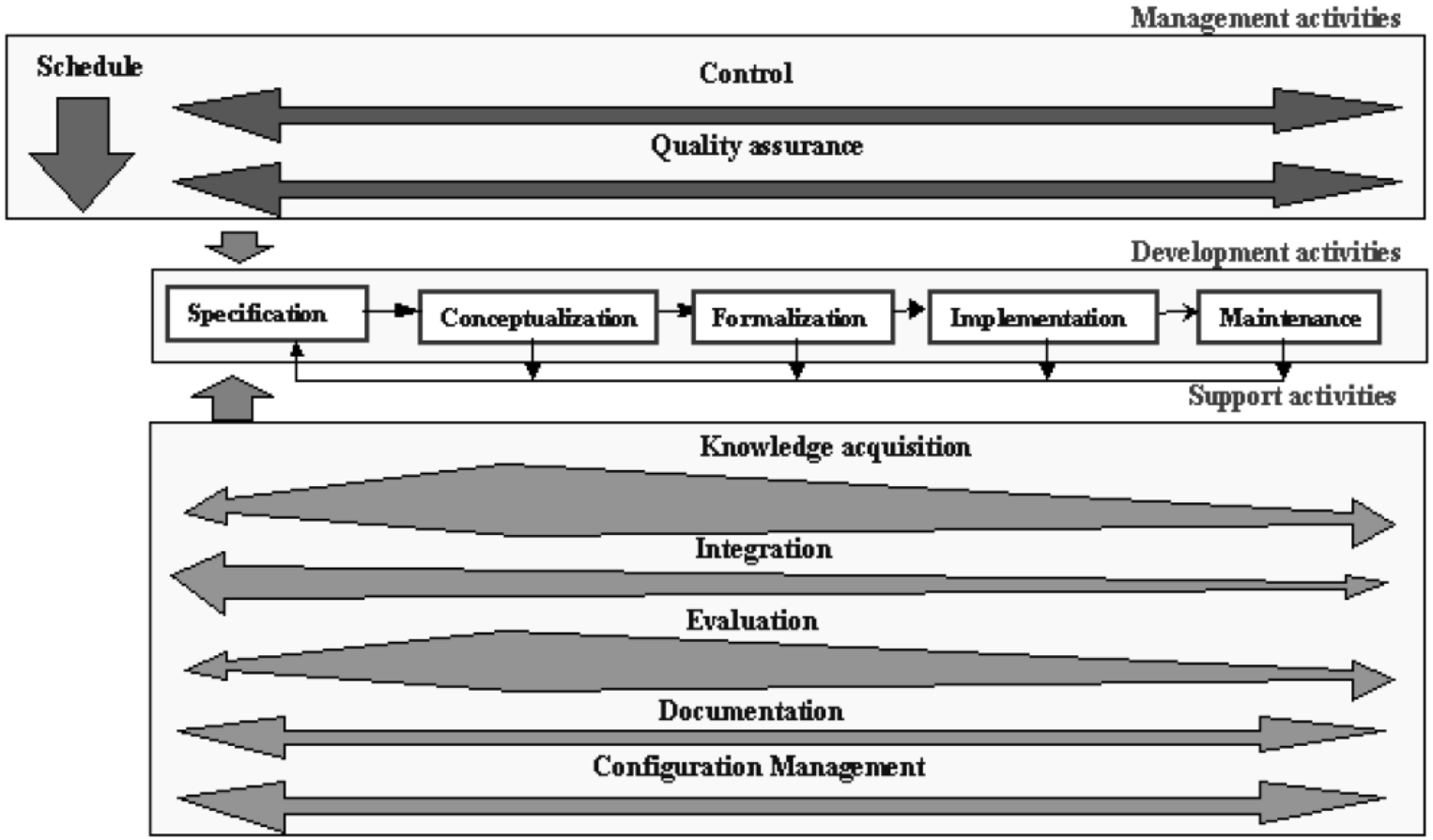
Where is the knowledge?

- Human sources (Domain experts)
  - Interviews, examples, scenarios.
- Unstructured sources (books)
  - Parsing & pronoun resolution, formalisation, integrations with lexical background knowledge.
- Semi-structured sources (websites).
- Structured sources (databases).

Ontology evaluation

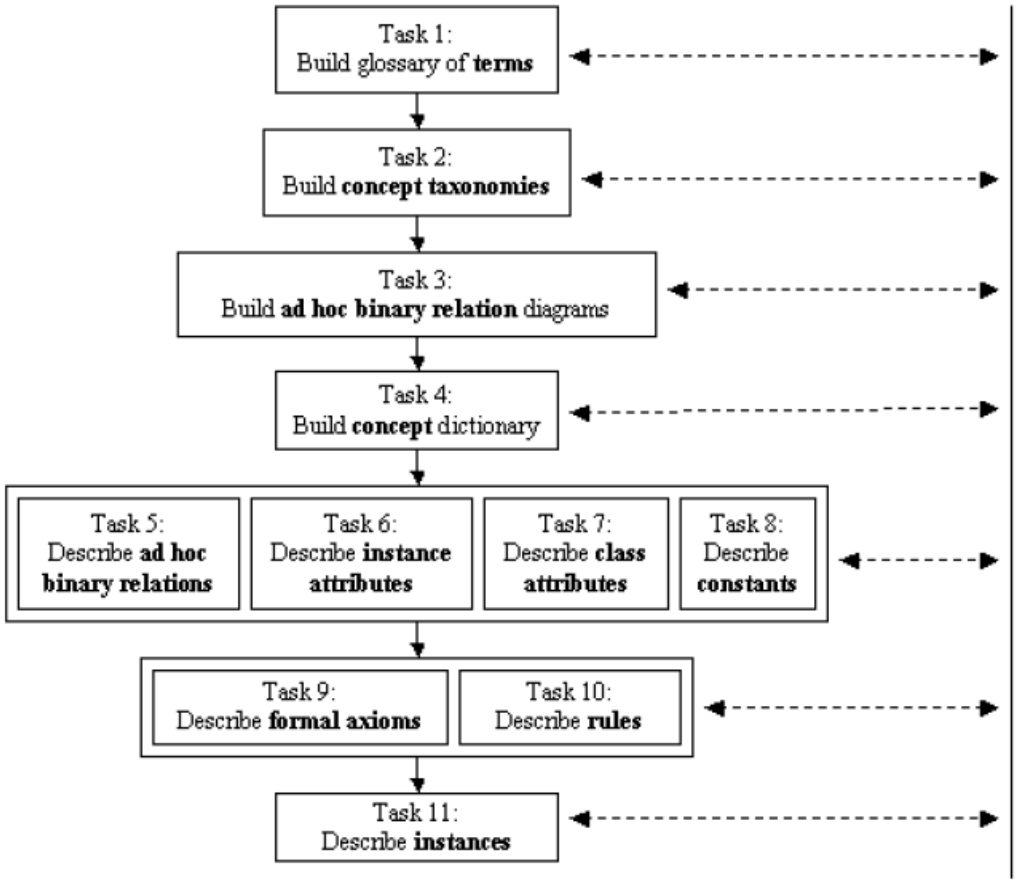
- Evaluation criteria:
  - Usage criteria: Fulfil intended purpose? Its logical consequences agree with reality? Help user accomplish task?
  - Logical criteria: consistency, completeness.
  - Structural and formal criteria: rigidity, identity, unity, dependence.
  - Accuracy: real-world conformance.

METHONTOLOGY



Source: Ontological Engineering

METHONTOLOGY - conceptualisation



Source: Ontological Engineering

The NeOn Methodology for building ontology networks

- A scenario-based methodology
  - supports the collaborative aspects of ontology development and reuse, and
  - the dynamic evolution of ontology networks in distributed environments.
- Main components:
  - A set of nine scenarios for building ontologies and ontology networks.
  - The NeOn Glossary of Processes and Activities.
  - Methodological guidelines for different processes and activities.

http://www.neon-project.org/nw/NeOn_Book

Tips for ontology creation

- Don't forget disjointness:

  $\text{Man} \sqsubseteq \text{Human}$ $\quad \text{Human} \sqsubseteq \text{Man} \sqcup \text{Woman}$

  $\text{Woman} \sqsubseteq \text{Human}$ $\quad \text{Man}(\text{alex})$ $\quad \text{Woman}(\text{amy})$

  $\neg\text{Woman}(\text{alex})$?

- Don't forget role characteristics:
  - Transitive? Inverse? Functional? Symmetric?
- Don't choose too specific domains and ranges.

Tips for ontology creation

- Be careful with quantifiers
  - Birds $\sqsubseteq \exists$ has.Wing "birds have (some) wings"
  - Birds $\sqsubseteq \forall$ has.Wing "birds have only wings"
  - Happy $\equiv \forall$ hasChild.Doctor *vs.*
    Happy $\equiv \forall$ hasChild.Doctor $\sqcap \exists$ hasChild.Doctor
- Don't mistake parts for subclasses
  - Finger $\sqsubseteq$ Hand *vs.* Finger $\sqsubseteq \exists$ part_of.Hand

Tips for ontology creation

- Watch the direction of roles and choose representative names
  - :macbeth :author :shakespeare .
  - :shakespeare :author :macbeth .
- Don't confuse class subsumption and class equivalence
  - $\sqsubseteq$ when necessary condition
  - $\equiv$ when necessary and sufficient condition

Considerations

- When should we introduce a new subclass?
  - Subclass should have additional properties or new property value or participate in different relationships than its superclass
- How many subclasses should a class have?
  - No hard rules. But if just 1 or more than 12, perhaps worth rethinking.
- For a given concept, when should we model it as a class, and when should we model it as an instance?
  - This really depends on the intended use of the ontology

Worth keeping in mind…

- There's no such thing as the "correct way to model a domain".
- Ontology development should be an iterative process

Noy and McGuinness (2001)

Tools

- Ontology editors:
  - Protégé
  - NeOn toolkit
  - SWOOP
  - And many more… http://www.w3.org/wiki/Ontology_editors
- OWL DL reasoners
  - Pellet
  - RacerPro
  - FaCT++

Summary

- Just like in Software Engineering, there is no golden recipe for Ontology Engineering.
- There are methodologies that can guide you throughout this process.
- Question, analyse, justify your approach.
- Ontology Engineering is an iterative process.
- Several tools available.

Reading

- SWWO Ch14

Task / Food for thought

- Revisit the simple hierarchy that you created at the beginning of this course.
- How would you build on it to develop a more comprehensive ontology?
- What considerations would you need to make?
- And how about any assumptions needed?