Knowledge Based Systems

Jacek Malec
Dept. of Computer Science, Lund University, Sweden

February 11, 2016

Programming vs KR

Programming:

- construct an algorithm to solve a problem
- choose a programming language
- implement the algorithm in the language
- run the program to solve the problem

Knowledge Representation and Engineering:

- identify knowledge necessary to solve the problem
- choose a representation language
- formulate knowledge in the language
- check if solution can be derived

Knowledge

There is evidence that people memorize just **the important stuff**.

- objects
- facts, relations
- rules (generic knowledge)
- causality
- meta-knowledge
- procedural knowledge
- knowledge about processes
- ...

Meta-knowledge

knowledge about other's (or our own)

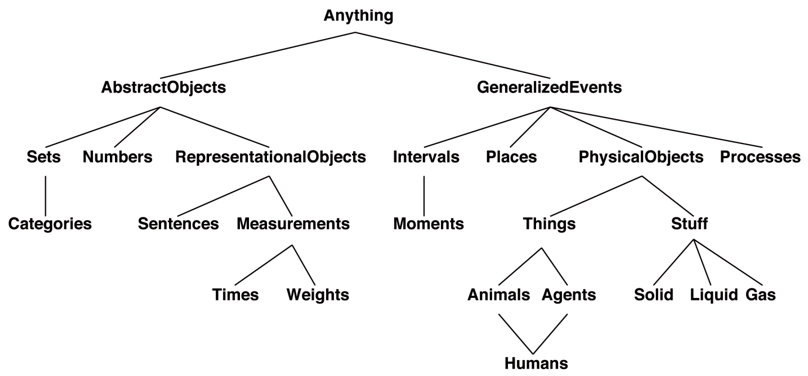
- knowledge (Jacek knows that Earth revolves around Sun)
- beliefs (Jacek believes that his daughter is smart)
- needs (Jacek needs more time for checking assignments)
- goals (Jacek wants to write a paper by Friday)
- intentions (Jacek wants to be ready with checking Assignment 1 by next Thursday)
- capabilities (Jacek can drive)
- ...

Procedural knowledge

How to:

- change a diaper?
- serve in tennis?
- ride a bicycle?
- drive a car?
- ...

Ontology



Anything

AbstractObjects

GeneralizedEvents

Sets

Numbers

RepresentationalObjects

Intervals

Places

PhysicalObjects

Processes

Categories

Sentences

Measurements

Moments

Things

Stuff

Times

Weights

Animals

Agents

Solid

Liquid

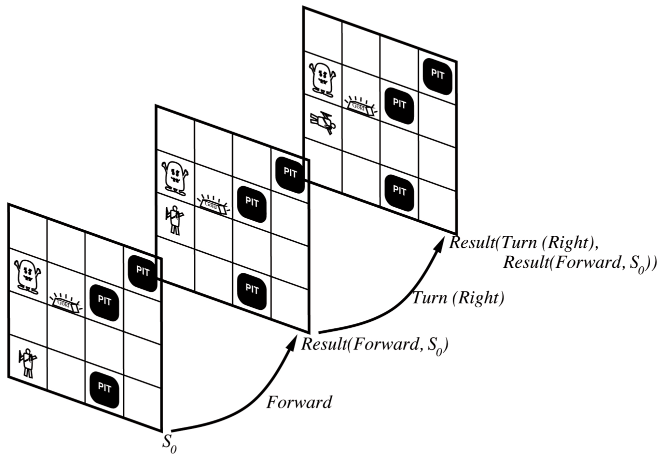
Gas

Humans

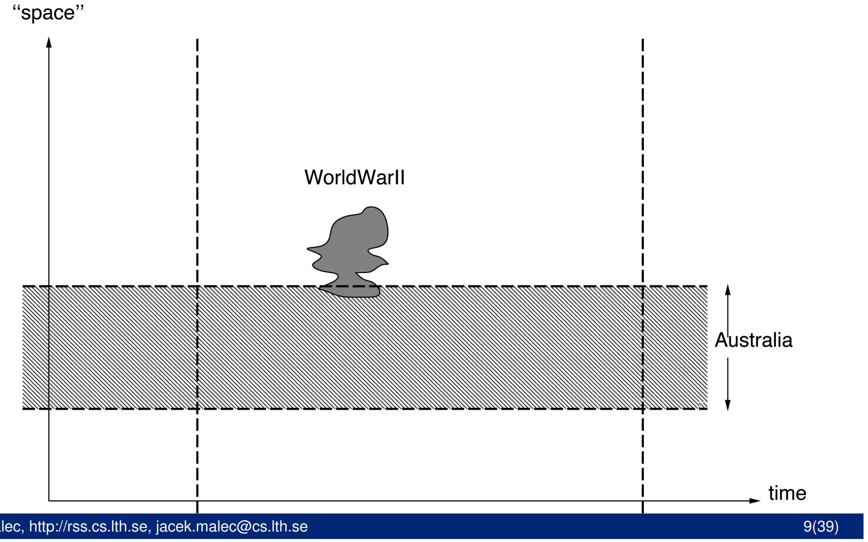
Problems with predicate logic:

- static
- flat
- qualification/ramification/frame problem
- exceptions
- strength
- ...

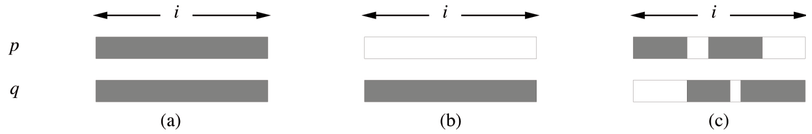
Actions, situations



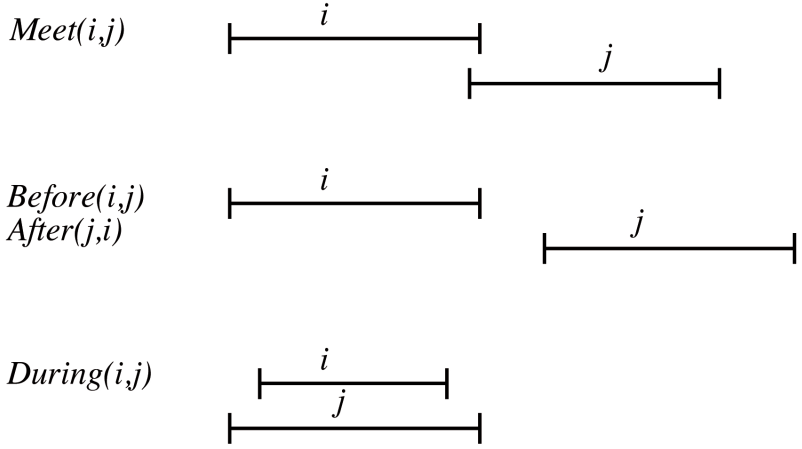
Events



Events



Interval calculus



Representation of exceptions:

$\forall x(\mathit{Bird}(x) \wedge \neg \mathit{Pinguin}(x) \wedge \neg \mathit{Ostrich}(x) \rightarrow \mathit{Flies}(x))$
But if we know $\mathit{Bird}(\mathit{Tweety})$ we can’t say whether it flies!

Qualification Problem:

Unfortunately, many other birds don’t fly either: caged birds, dead birds, birds with broken wing, bird with feathers covered by oil, ...
$\forall x(\mathit{Bird}(x) \wedge \neg \mathit{Pinguin}(x) \wedge \neg \mathit{Ostrich}(x) \wedge \ldots \rightarrow \mathit{Flies}(x))$

Other related problems: frame problem, ramification problem

Solutions:

- non-monotonic logics
- fuzzy logic
- probabilistic logic
- possibilistic logic
- ...

Monotonic vs. non-monotonic

$Th(\Delta)$

monotonic logic:
if $\Delta \subseteq \Delta'$ then $Th(\Delta) \subseteq Th(\Delta')$

non-monotonic logic:
if $\Delta \subseteq \Delta'$ and $Th(\Delta) \supset Th(\Delta')$

Closed World Assumption:

Things that cannot be proven true are probably false

leads to:

Negation as failure

Consider:

$$\begin{aligned}\Delta = \{&A(Stockholm), A(Sturup), A(Copenhagen),\\ &A(Oslo), Conn(Oslo, Copenhagen),\\ &Conn(Stockholm, Oslo),\\ &\forall(x, y, z)(Conn(x, y) \wedge Conn(y, z) \rightarrow Conn(x, z))\}\end{aligned}$$

Because it's not the case that $Conn(Sturup, Stockholm)$, CWA gives us immediately $\neg Conn(Sturup, Stockholm)$.
But let us add $Conn(Sturup, Copenhagen)$ to Δ.
$Conn(Sturup, Stockholm)$ can be proven now, so $\neg Conn(Sturup, Stockholm)$ does not appear as a consequence.
Non-monotonicity!

CWA is syntax-dependent:

If $\Delta = \{\mathit{Single}(\mathit{John}), \mathit{Single}(\mathit{Mary}), \mathit{Clever}(\mathit{Kent})\}$
then CWA gives us:
$\{\neg\mathit{Clever}(\mathit{John}), \neg\mathit{Clever}(\mathit{Mary}), \neg\mathit{Single}(\mathit{Kent})\}$.

But if $\Delta = \{\neg\mathit{Married}(\mathit{John}), \neg\mathit{Married}(\mathit{Mary}), \mathit{Clever}(\mathit{Kent})\}$
then CWA gives us:
$\{\neg\mathit{Clever}(\mathit{John}), \neg\mathit{Clever}(\mathit{Mary}), \neg\mathit{Married}(\mathit{Kent})\}$.

Effective representation of knowledge:

- storing
- retrieval
- modification

What should be represented?

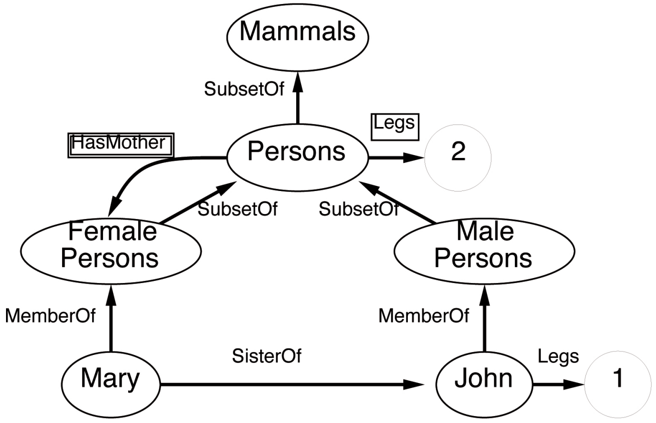
- use logic
- not necessarily FOL

How should it be represented?

- whatever method you find suitable

Semantic networks:

Precursor of Description Logics and Semantic Web Languages (OWL, DAML, OIL, ...)



Reasoning in SN:

- inheritance via SubsetOf (SubClass) and MemberOf (isA) links
- intersection paths
- special meaning associated with some links (like cardinality constraints, etc.)
- classification, consistency, subsumption

May be effective given some restrictions on the logic (DL variants).
Rule-based reasoning on top (RIF and co.)

Knowledge-Based Systems

A generic term, might denote anything that involves encoded knowledge.
Or might mean a system where the knowledge component is *explicit* and manipulable.

Paradigms throughout history of AI:

- Logic-based systems;
- Rule-based systems (expert systems);
- Blackboard systems;
- Semantic web systems.

Rule-based systems

Or *expert systems*.
Promised much. Delivered (too) little. Caused “AI Winter” in the 90s.
Simple architecture:

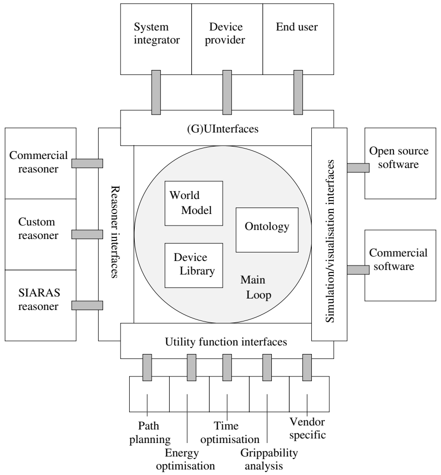
- Facts;
- Rules;
- Inference engine:
  - Matching;
  - Conflict resolution;
  - Rule application.

Blackboard systems

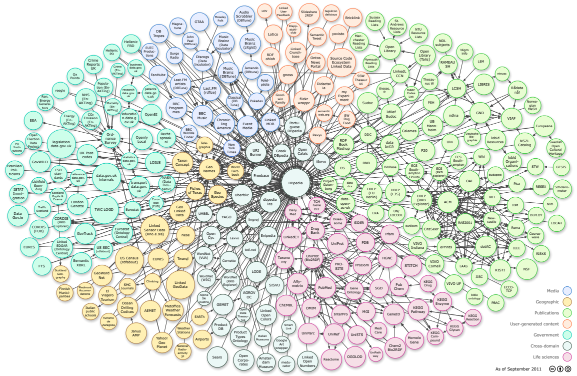
Architecture:

- Knowledge base (blackboard);
- Knowledge sources (expert problem solvers);
- Controller (agenda maintainer).

A blackboard system



Semantic Web



Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. `http://lod-cloud.net/`

Semantic Web

Lots of hype. Lots of acronyms. But some are important!

- URI – Uniform Resource Identifier
- RDF – Resource Description Framework
- RIF – Rule Interchange Format
- SPARQL – SPARQL Protocol and RDF Query Language
- OWL – Web Ontology Language

Open World Assumption!

SPARQL

W3C Recommendation
Queries:

- SELECT (returns a table)
- CONSTRUCT (returns RDF)
- ASK (returns a boolean)
- DESCRIBE (freedom)

Example:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?email
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
}
```

A SPARQL query

What are all the country capitals in Africa?

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
     abc:isInContinent abc:Africa .
}
```

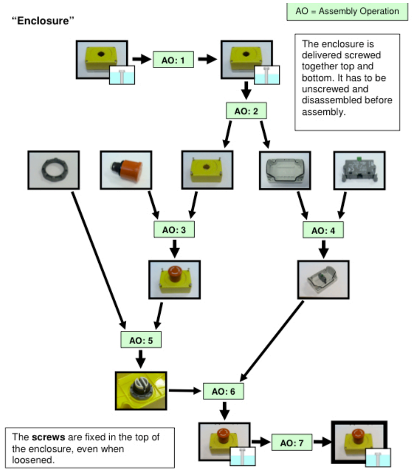
A real SPARQL query

```
http://wiki.dbpedia.org/OnlineAccess
http://kif.cs.lth.se

select ?s where {
  ?s a rosetta:Camera.
}

select distinct ?s ?v where {
?s a rosetta:Camera.
?s ?p ?n.
?n caex-xml:hasName "FocusRange".
?n caex-xml:hasValue ?v.
}
```

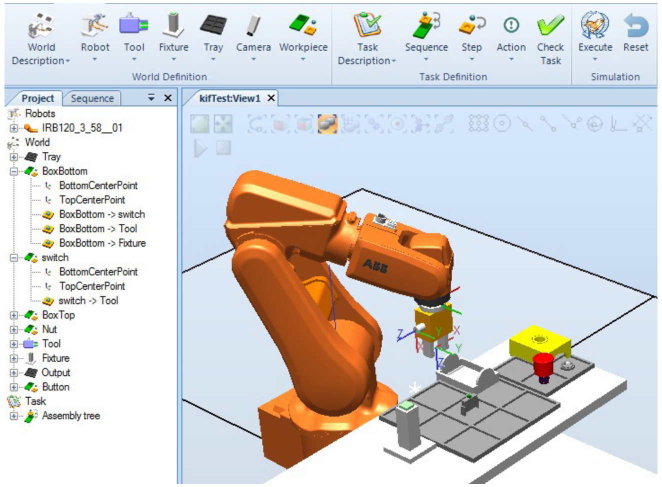
A real case



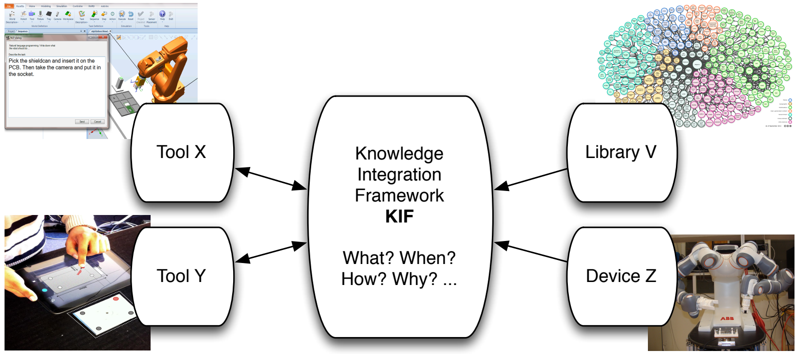
The enclosure is delivered screwed together top and bottom. It has to be unscrewed and disassembled before assembly.

The **screws** are fixed in the top of the enclosure, even when loosened.

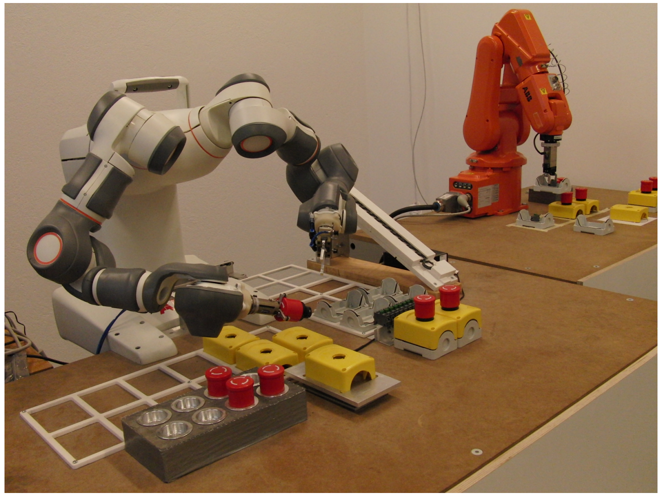
A real case



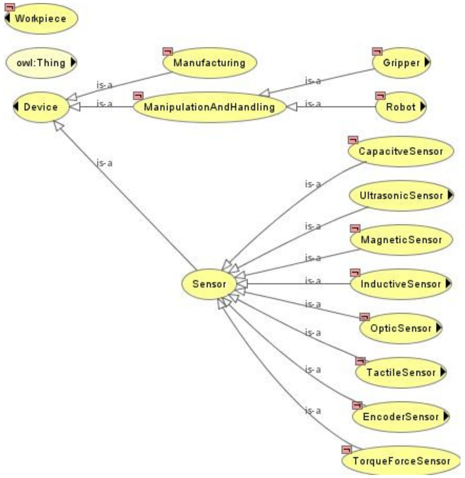
A real case


Pick the shieldcan and insert it on the PCB. Then take the camera and put it in the socket.

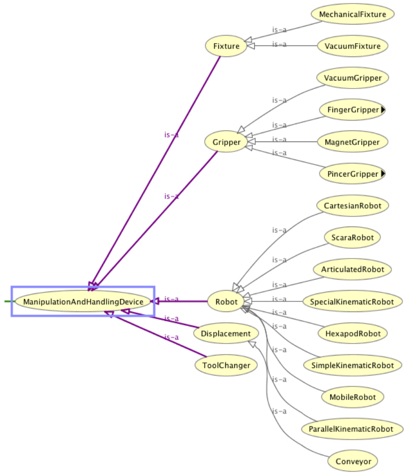
A real case



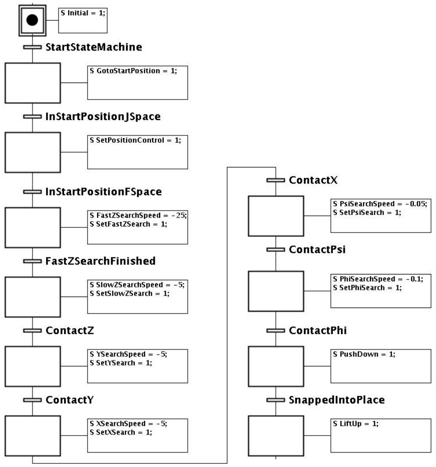
A real case



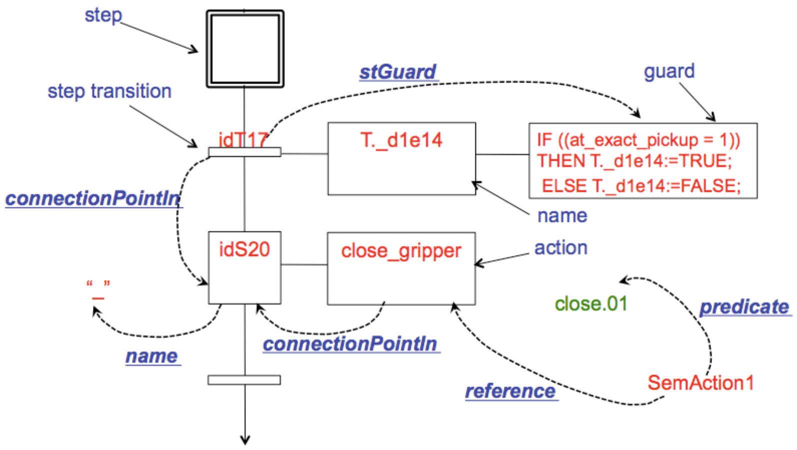
A real case



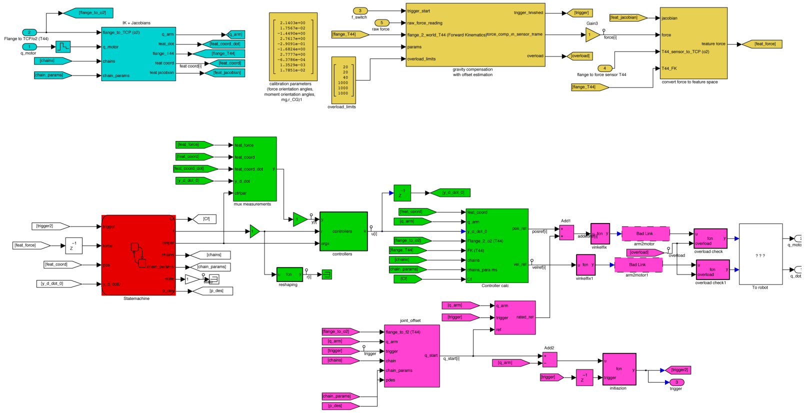
A real case



A real case



A real case



Next period: EDAN70

EDAN70 Project in Computer Science (AKA Intelligent Systems)

- topics proposed by us (PN, EAT, JM)
- but: you may choose your own
- personal supervision
- research-related problems
- workload: normal course, 7.5 hp
- First meeting after Easter, 31/3, 15.15 in E:2116
- but: contact us earlier if you are interested