Knowledge Engineering in Rosetta project
European Robotics Forum, Västerås

Jacek Malec, Lund University
Anders Björkelund, Herman Bruyninckx, Klas Nilsson
& many others

April 8, 2011

Questions we want to have answered

Terminology What are task descriptions, action recipes, skills and other primitives, and what are their relationships?

Conventions Are there shared definitions, conventions (e.g., coordinate systems, units), and data structures?

Scene graphs How should data be represented (maps, objects, actions)? What data should be annotated and how? What kind of reasoning is performed or needed?

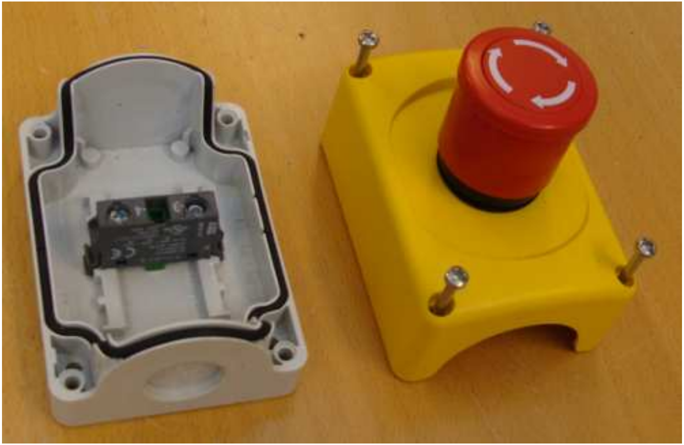
Reuse of knowledge How can a robot decide which knowledge (e.g., map or skill) to reuse in a new situation?

Reuse of tools What existing software modules, algorithms, libraries, or APIs can be reused?

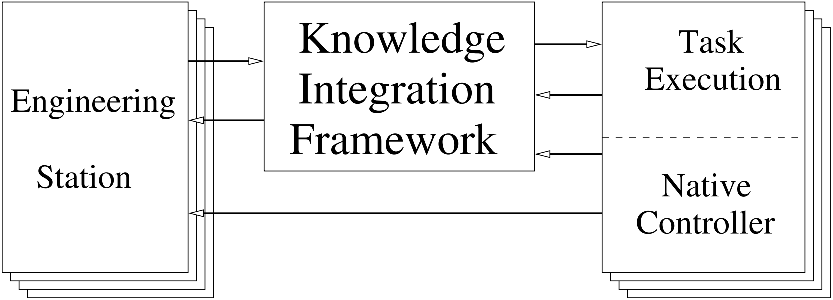
Knowledge engineering How will/should the knowledge base grow? What are the processes bootstrapping a knowledge base useful in real applications?

Rosetta

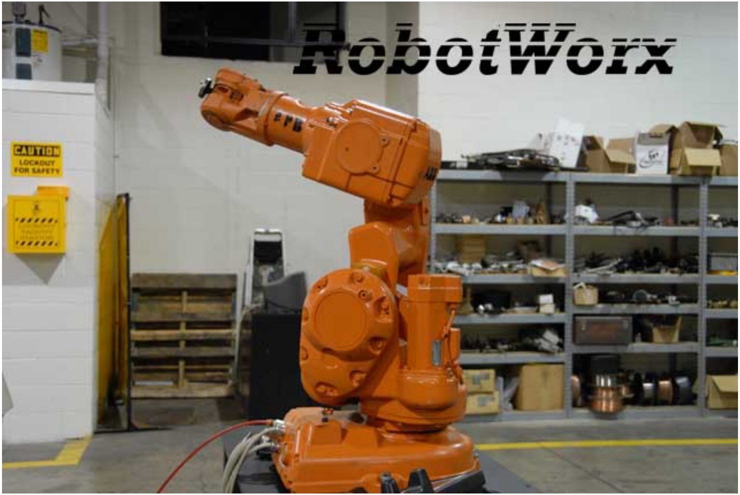
Our aim is assembly:



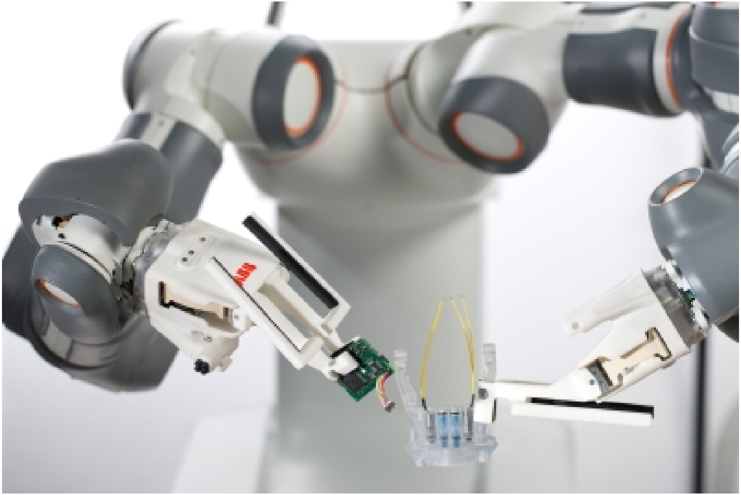
Rosetta


Engineering
Station
Knowledge
Integration
Framework
Task
Execution
Native
Controller

Target robots: IRb-140



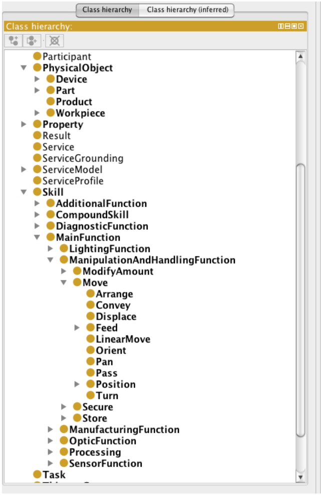
Target robots: FRIDA



Vocabulary

- Activities:
  - Motions
    continuous spacetime activity of a robot finishing on some observable condition (detectable by sensors)
  - Actions
    every other continuous time activity but a motion, e.g. image processing or path-planning
- Skills
  Discretely interconnected set of activities (FSM, with states possessing appropriate structure), may span several levels of complexity (compound skills vs. primitive skills)
- Tasks
  Abstracted skills, providing information about the goal, but not the means; e.g. an assembly graph for the stop button case

Conventions: Rosetta ontology



- Participant
- PhysicalObject
  - Device
  - Part
  - Product
  - Workpiece
- Property
- Result
- Service
- ServiceGrounding
- ServiceModel
- ServiceProfile
- Skill
  - AdditionalFunction
  - CompoundSkill
  - DiagnosticFunction
  - MainFunction
    - LightingFunction
    - ManipulationAndHandlingFunction
      - ModifyAmount
      - Move
        - Arrange
        - Convey
        - Displace
        - Feed
        - LinearMove
        - Orient
        - Pan
        - Pass
        - Position
        - Turn
      - Secure
      - Store
    - ManufacturingFunction
    - OpticFunction
    - Processing
    - SensorFunction
- Task

Origins:

- SIARAS (2006–08, assembly, sensing)
- RoSta (2008–09, manufacturing)

Undergoing modularisation:

- NASA qudt ontology
- OWL-S
- statecharts
- project-specific knowledge

Scene graph

By the *scene graph* we understand all information about the objects in the work-cell related to the scene geometry. In particular, for each object we need to know its shape (possibly as a function of time if it moves or is being deformed), its physical properties, the space it occupies and its relation to other objects.

Overall purpose of scene graph:

- to share a world model between two or more components in the system, so that
- they can provide their piece of information about how the world looks like, and how it is changing.
- they can get information about those parts of the world they are not able to provide their own information for.

Skills

Discretely interconnected set of activities: Finite State Machines, where states correspond to activities

- Motions
- (sensing and processing) Actions

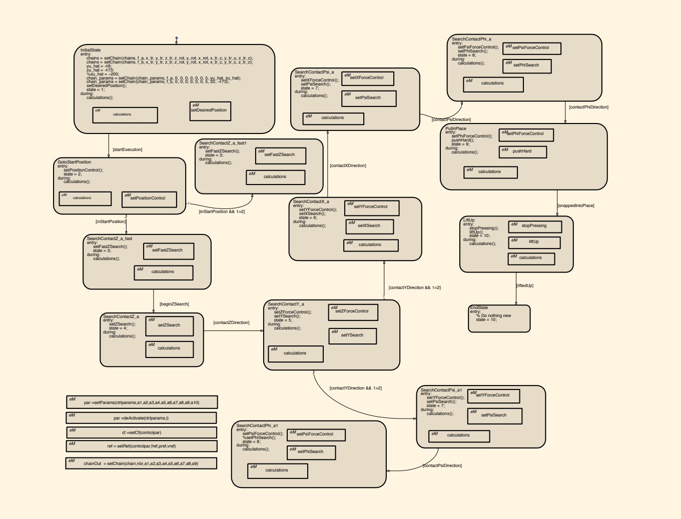
May span several levels of complexity (compound skills vs. primitive skills)

May be treated as services (for task planning/decomposition purposes)

Parametrised (to enable learning)

Contextualized (platform-specific solutions)

An example skill



```
InitialState
entry:
    chains = setChain(chains, f_a, x_tr, y_tr, z_tr, z_rot, y_rot, x_rot, x_tr_c, y_tr_u, z_tr_c);
    chains = setChain(chains, f_b, x_tr, y_tr, z_tr, z_rot, y_rot, x_rot, x_tr_c, y_tr_c, z_tr_c);
    yu_hat = -18;
    zu_hat = -173;
    %zu_hat = -200;
    chain_params = setChain(chain_params, f_a, 0, 0, 0, 0, 0, 0, 5, yu_hat, zu_hat);
    chain_params = setChain(chain_params, f_b, 0, 0, 0, 0, 0, 0, 5, 22, -173);
    setDesiredPosition();
    state = 1;
during:
    calculations();
```

```
GotoStartPosition
entry:
    setPositionControl();
    state = 2;
during:
    calculations();
```

```
SearchContactZ_a_fast1
entry:
    setFastZSearch();
    state = 3;
during:
    calculations();
```

```
SearchContactZ_a_fast
entry:
    setFastZSearch();
    state = 3;
during:
    calculations();
```

```
SearchContactZ_a
entry:
    setZSearch();
    state = 4;
during:
    calculations();
```

```
SearchContactY_a
entry:
    setZForceControl();
    setYSearch();
    state = 5;
during:
    calculations();
```

```
SearchContactX_a
entry:
    setYForceControl();
    setXSearch();
    state = 6;
during:
    calculations();
```

```
SearchContactPsi_a
entry:
    setXForceControl();
    setPsiSearch();
    state = 7;
during:
    calculations();
```

```
SearchContactPhi_a
entry:
    setPsiForceControl();
    setPhiSearch();
    state = 8;
during:
    calculations();
```

```
PutInPlace
entry:
    setPhiForceControl();
    pushHard();
    state = 9;
during:
    calculations();
```

```
LiftUp
entry:
    stopPressing();
    liftUp();
    state = 10;
during:
    calculations();
```

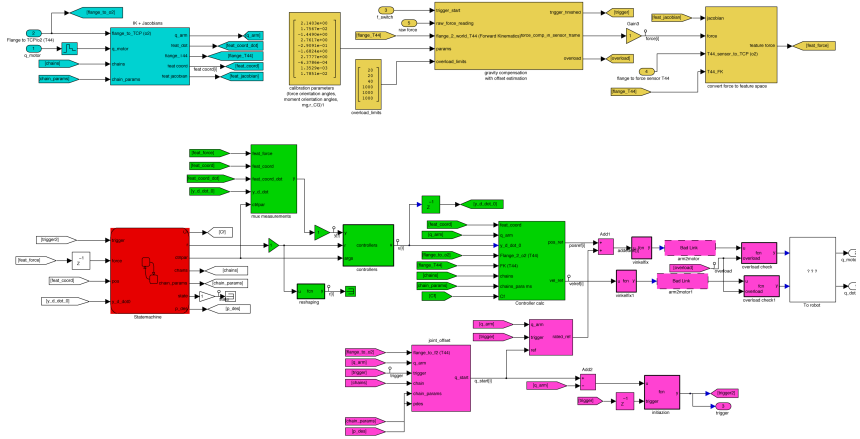
```
EndState
entry:
    % Do nothing new
    state = 10;
```

```
SearchContactPsi_a1
entry:
    setYForceControl();
    setPsiSearch();
    state = 7;
during:
    calculations();
```

```
SearchContactPhi_a1
entry:
    setPsiForceControl();
    %setPhiSearch();
    state = 8;
during:
    calculations();
```

```
par =setParams(ctrlparams,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10)
par =deActivate(ctrlparams,i)
cf =setCf(controlpar)
ref = setRef(controlpar,fref,pref,vref)
chainOut  = setChain(chain,nbr,a1,a2,a3,a4,a5,a6,a7,a8,a9)
```

An example skill



Reusable tools and results

What could we share?

- ontology / ontologies
- skill knowledge
- log data (for learning purposes)

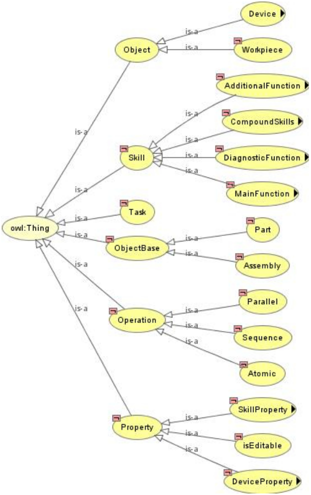
Some of Rosetta data is unfortunately not open: a good policy is needed

Conclusions

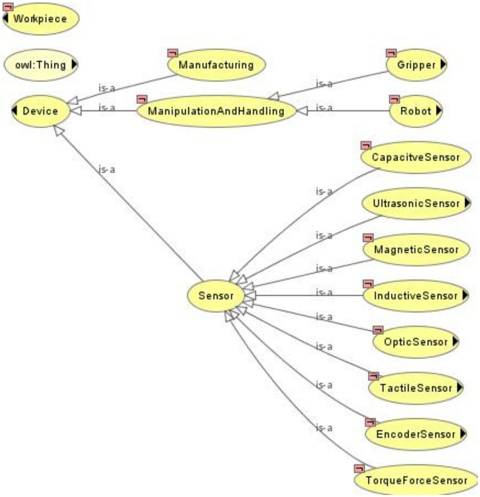
- Challenging questions
- Hard knowledge extraction problem
- Lots has been already done, sometimes under different names
- White paper to be written...

THANK YOU!

Ontology extra 1



Ontology extra 2


Workpiece
owl:Thing
Manufacturing
Gripper
Device
ManipulationAndHandling
Robot
CapacitveSensor
UltrasonicSensor
MagneticSensor
InductiveSensor
OpticSensor
TactileSensor
EncoderSensor
TorqueForceSensor
Sensor
is-a