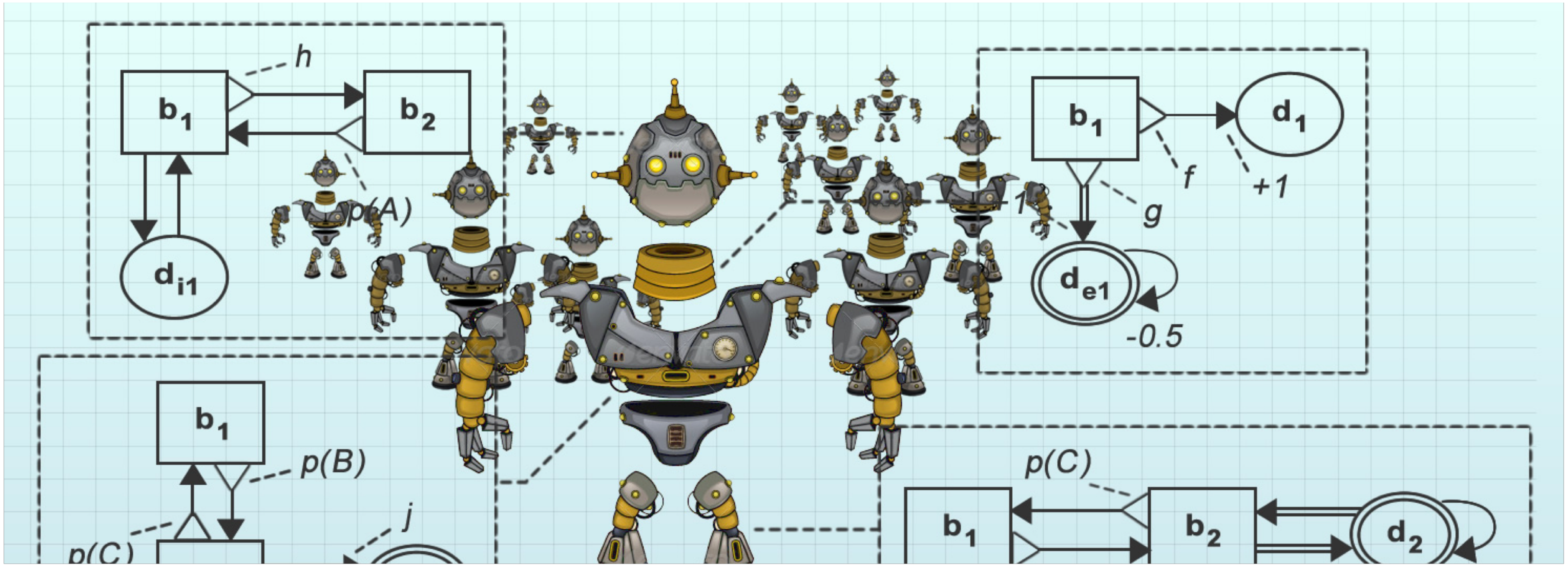
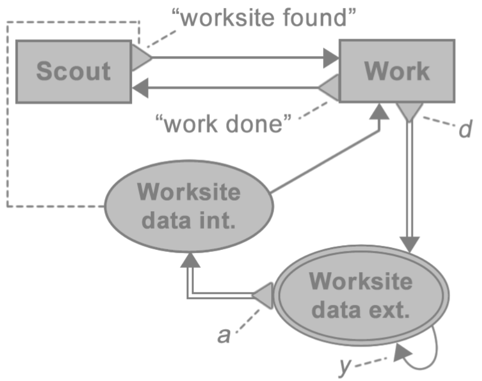
Modeling Multi-Agent Systems with Sketch BDRML

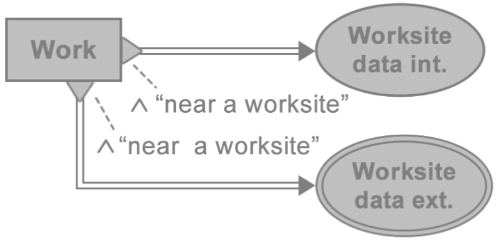
Lenka Pitonakova :: Mar 2019



Why model?

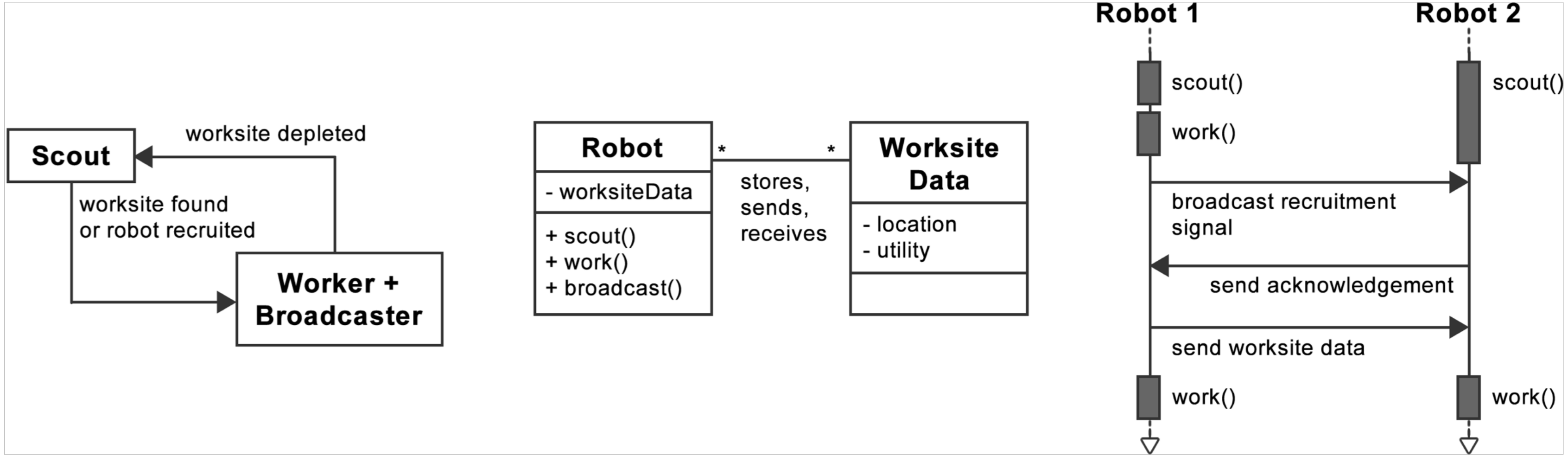
- Planning
  - Behaviours and behaviour transitions easier to draw on paper than directly test in code
  - Data structures: Which ones are needed and when? Is my solution viable?
- Dissemination
  - Image often tells a much clearer story than text
  - A good graphical representation must avoid ambiguities



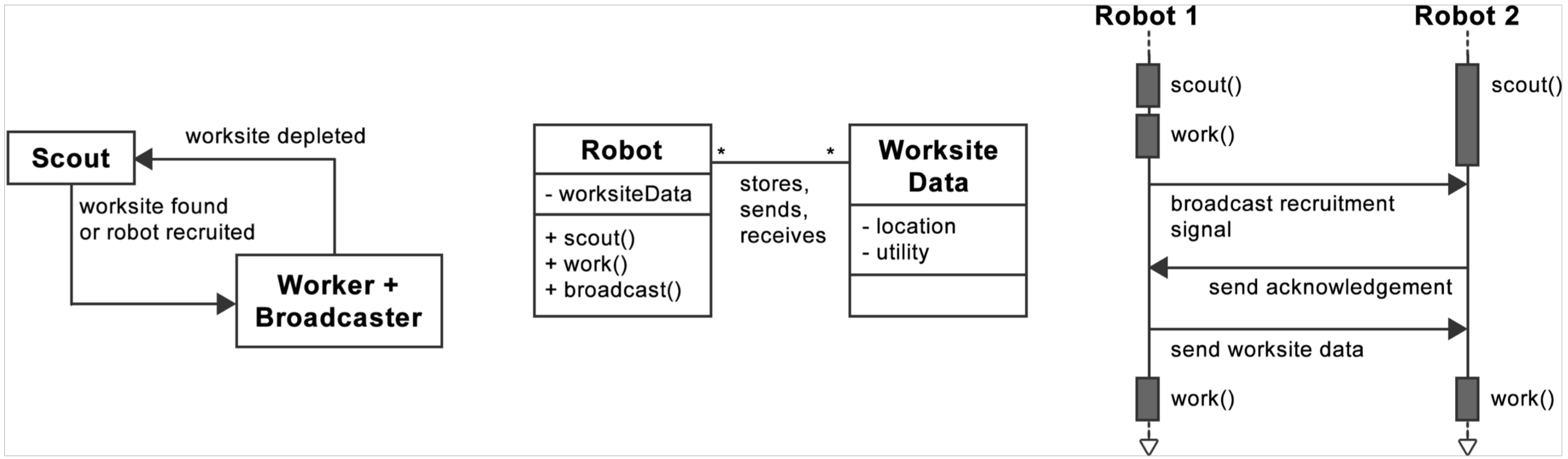


Existing methods

Algorithm: Search for worksites in the environment. Perform work them. Recruit nearby robots while working (e.g. foraging, customer servicing).



Statechart Class diagram Sequence chart



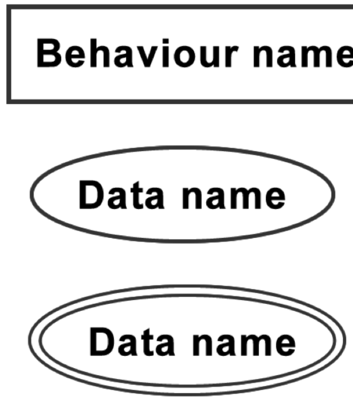
- These methods were invented when programs were simpler and more linear
- Problems for multi-robot systems:
  - Assumptions of finite-state machines with well-defined, predictable interactions
  - No explicit representation of data or of influences external to the system

Behaviour-Data-Relations Modelling Language

- Describes robot **behaviours**, not states.
  - “Work” behaviour versus “Worker” state
  - Model finite-state machines, neural network controllers, behaviour-based controllers, etc.
- Both **behaviours** and **data** are primitives, so they can **relate** to each other
  - Explicit representation of what information is communicated and where it is stored
  - Combines capabilities of statecharts and class diagrams (describe control algorithm) and of sequence charts (describe communication)
- Allows to specify relations between behaviours and **data external** to a robot’s memory
  - Represent communication between robots and interactions with their environment

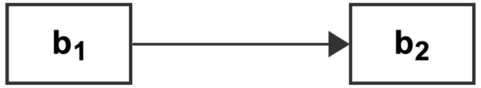
Primitives

- Behaviours
- Internal data structures
- External data structures

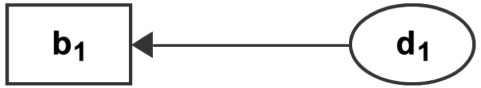


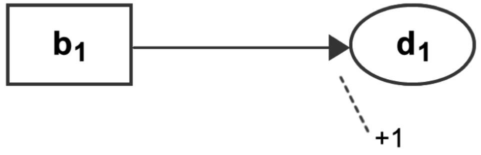
Relations

- Transition

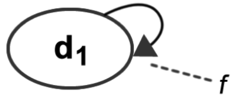


- Read / write

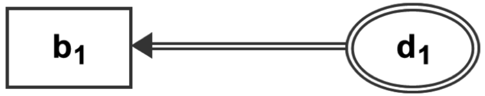


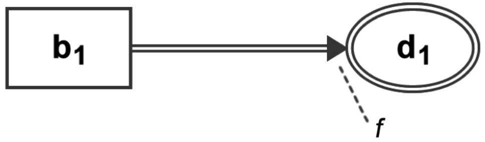


- Update

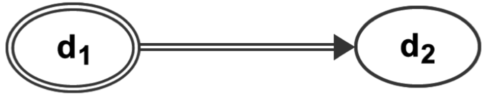


- Receive / send



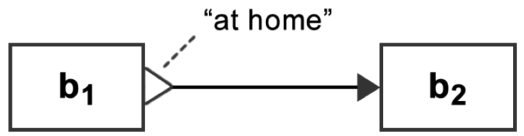


- Copy

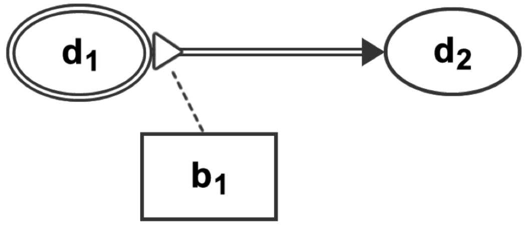


Conditions

- Specify when relations apply
  - A certain fact is true

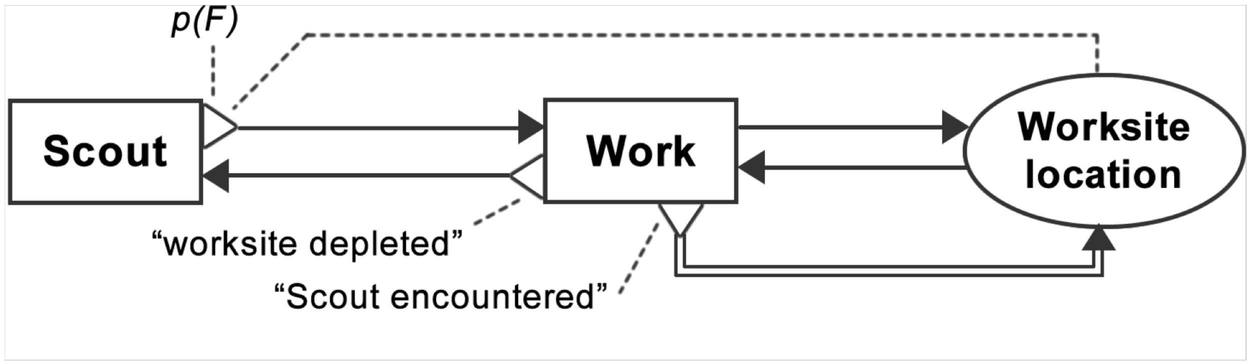


  - Robot is executing a behaviour



  - …

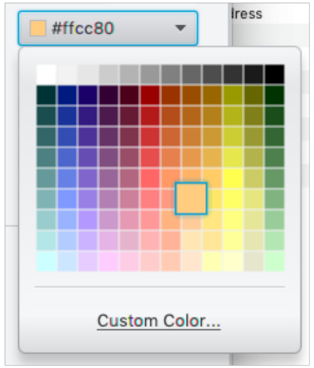
Example: Recruitment

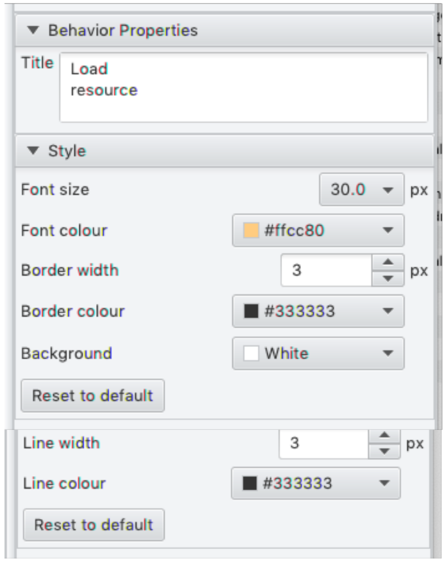


- **Explicit representation of recruitment**:
  - *Send* relation between *Work* and *Worksite location*
  - Conditional transition between *Scout* and *Work*

Styling in Sketch BDRML

- Default styles
  - Set for the whole file
- Component styles
  - Set for individual primitives and relations
  - Possible to reset to Default style
- Advanced colour picker for custom colours and transparency

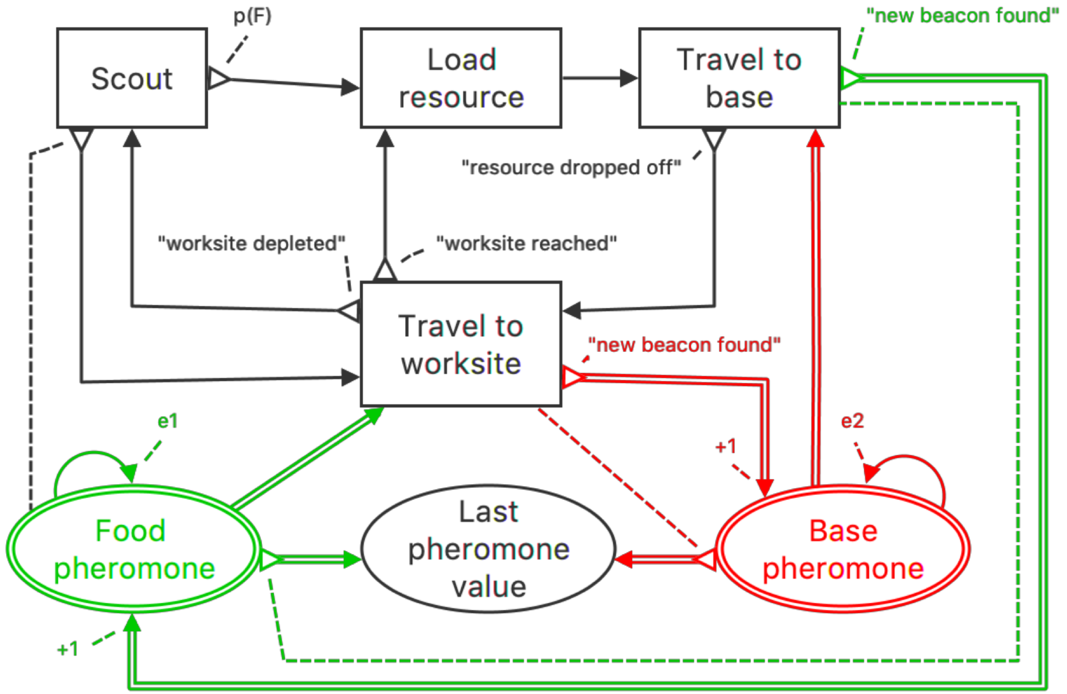




Title: Load resource
Font size: 30.0 px
Font colour: #ffcc80
Border width: 3 px
Border colour: #333333
Background: White
Line width: 3 px
Line colour: #333333

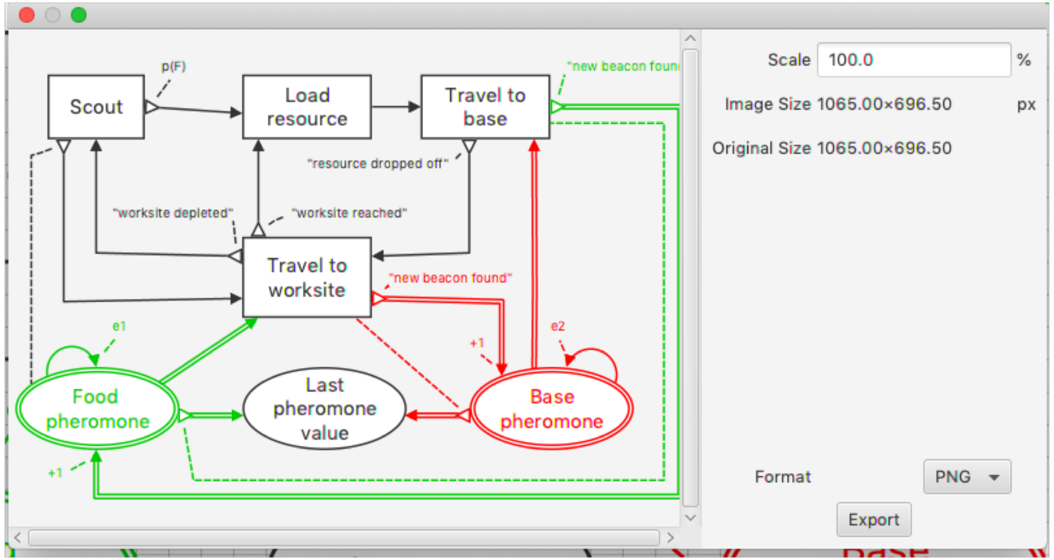
Example: Pheromone Trail Following

- Default black and white styles used on most components
- Different pheromone following behaviours are highlighted



Sharing your work

- Sketch BDRML has its own file format. You can save .bdr files and share them with others
- Use the Export to PNG feature to create images for your papers etc.
  - Shows a preview
  - Image can be re-scaled



Scale 100.0 %
Image Size 1065.00×696.50 px
Original Size 1065.00×696.50

Why Sketch BDRML?

http://swarmdesign.lenkaspace.net/sbdrml/

- Helps you to think about how data should be gathered and used within your system
- Makes it easier to judge feasibility of solutions before they are implemented
- Makes it easier to keep and share algorithm documentation

Sketch BDRML

Acknowledgements

http://swarmdesign.lenkaspace.net/sbdrml/

The following people have been involved with development of BDRML and Sketch BDRML: **Seth Bullock** (University of Bristol), **Ben Rayneau-Kirkhope** (Thales UK), **William Bonnell** (University of Bristol).

Sketch BDRML was funded by the University Bristol and by the EPSRC Early Career Kickster award.

Sketch
BDRML