Towards a real-time BDI model for ROS 2

20TH WORKSHOP “FROM OBJECTS TO AGENTS”
June 26, 2019

Francesco Alzetta
University of Trento
francesco.alzetta@unitn.it

Paolo Giorgini
University of Trento
paolo.giorgini@unitn.it
Motivation

Increasing interest in AI-driven real-time systems

- Adaptive
- Responsive
- Scalable

But

- Hard to design
- Hard to maintain
Motivation

- Design
- Maintenance

Means-end reasoning
- Beliefs
- Desires
- Plans

Real-time execution
- Intention
ROS 2

ROS (Robot Operative System) has several advantages:

- Cross-platform
- Integrated communication system
- Modularity
- Wide adoption

Many improvements made in ROS 2, in particular:

- Real-time compliance and
- System scalability.
Agent architecture

Absolute priority (i.e. classification of desires)

Relative priority among applicable plans
Agent architecture

The time the agent has to accomplish the goal

The maximum time required to complete a plan
Agent architecture

Responsible for:

- **sensing** the environment
- **monitoring** the internal state of the agent
Agent architecture

Responsible for:

- **listening** on belief and goal topics
- **checking** if a new plan instantiation is necessary
- **checking** the status of already instantiated plans
- **perform** rescheduling if necessary
- **send** the new schedule to the Executor_Node
Agent architecture

Responsible for:

- **executing** the intentions step by step
- **updating** belief and goal topics
Experiment

Goal:
Simulate a robot able to take decisions exploiting the human practical reasoning and respecting the real-time constraints.

3 sensors:
- senses cleanliness of the rooms
- monitors the charge of the battery
- monitors the filling level of the dirt tank

Actuators to:
- Move the robot
- Vacuum the dirt
Experiment

3 sensors:
- senses cleanliness of the rooms
- monitors the charge of the battery
- monitors the filling level of the dirt tank

Actuators to:
- Move the robot
- Vacuum the dirt
Results

Legend

- Task start
- Task end

(desire's priority, desire's deadline)

\[ \tau_2 \text{ CleaningSpecific (} p=3, \ d=8 \) \]
Results

\[ \times 10 = \text{Battery total capacity} \]

Legend:
- Task start
- Task end

(task's priority, task's deadline)

\[ \tau_1 \]
\[ \tau_2 \text{ CleaningSpecific (} p=3, \ d=8 \text{)} \]
\[ \tau_3 \]
\[ \tau_4 \text{ RechargeBatterySafe (} p=1, \ d=15 \text{)} \]
\[ \tau_5 \]
\[ \tau_6 \text{ EmptyHalfTank (} p=1, \ d=15 \text{)} \]
Results

Legend

- x 10 = Battery total capacity
- Task start
- Task end

(desire's priority, desire's deadline)

τ₂ CleaningSpecific (p=3, d=8)
τ₄ RechargeBatterySafe (p=1, d=15)
τ₆ EmptyHalfTank (p=1, d=15)
Results

Legend

- **x 10 = Battery total capacity**
- **Task start**
- **Task end**

(task's priority, task's deadline)

\[ \tau_1 \text{ CleaningRoutine (p=2, d=35)} \]

\[ \tau_4 \text{ RechargeBatterySafe (p=1, d=15)} \]
Results

x 10 = Battery total capacity

Legend

- Task start
- Task end

(desire's priority, desire's deadline)

Task request

\[ \tau_1 \] CleaningRoutine \((p=2, \ d=35)\)
\[ \tau_3 \] RechargeBatteryCritical \((p=5, \ d=10)\)
\[ \tau_4 \] RechargeBatterySafe \((p=1, \ d=15)\)
\[ \tau_6 \] EmptyHalfTank \((p=1, \ d=15)\)
Results

\[ \times 10 = \text{Battery total capacity} \]

Legend

- \text{Task start}
- \text{Task end}

(task's priority, task's deadline)

\( \tau_1 \) CleaningRoutine \((p=2, d=35)\)
\( \tau_2 \) CleaningSpecific \((p=3, d=8)\)
\( \tau_3 \) RechargeBatteryCritical \((p=5, d=10)\)
\( \tau_4 \) RechargeBatterySafe \((p=1, d=15)\)
\( \tau_5 \) EmptyFullTank \((p=4, d=6)\)
\( \tau_6 \) EmptyHalfTank \((p=1, d=15)\)
Conclusions and future work

We have proposed an approach to make the design of the means-end reasoning process of robotics devices easier.

The design of the “brain” of the agent follows the human practical reasoning concepts (i.e. the BDI paradigm).

The major novelty consists in expanding the BDI model by providing support for real-time, i.e. allow the agent to take into consideration concepts such as priority and deadline when generating its intentions.

Experimental results show that implementing this approach in ROS 2 is promising, since its structure can easily be exploited to setup a MAS and represent the concepts of the RT-BDI paradigm.
Conclusions and future work

- Formalization of the model
- Creation of a taxonomy redefining BDI concepts in order to suit RT terminology
- Deployment on a real robot and evaluation of performances w.r.t. pure real-time behaviour
- Extension of the framework to support MAS (communication, negotiation, cooperation)
- Implementation of a GUI and design tools to help the designer to build and maintain the code
Thank you