Using constraints for Norm-aware BDI Agents

Felipe Meneguzzi – CMU
Wamberto Vasconcelos – Aberdeen
Nir Oren – Aberdeen
Outline

• BDI Reasoning
• Norms
• nu-BDI – Normative BDI Reasoning
  – Updating Norms
  – Actions and Norms
  – Annotating Constraints
  – Selection of Plans
• Conclusions
BDI Model

• Beliefs-Desires-Intentions
• Philosophical model of *practical reasoning*
  – Describes how reasoning occurs with *limited resources*
  – Intuitive way of describing reasoning
  – Widely used in the implementation of software agents
  – Has a strong theoretical background
  – Various open implementations available
Plan Selection

- Key process in BDI architectures
- Filters relevant and applicable plans
- Binds variables to plans in the plan library

\[ e : b_1 \land \ldots \land b_x \leftarrow h_1 ; \::; h_x. \]
Plan Example

\[ +!goTo(C), hasVehicle(V), \left[ \begin{array}{c}
\ \ \ getVehicle(V),
\ \ \ moveTo(C)
\end{array} \right] \]

New event

!goTo(london)  hasVehicle(airplane)

Belief Base

Resulting Plan

\[ \left[ \begin{array}{c}
\ \ \ getVehicle(airplane),
\ \ \ moveTo(london)
\end{array} \right] (C = london,V = airplane) \]
Norms

• Used to define rules of acceptable behaviour in a society

• Through deontic concepts of
  – obligations (must)
  – permissions (may)
  – prohibitions (must not)
Norm representation

• Focuses on the operational aspect of norm compliance
• Norms are defined in the form
  – Normative Formula
  – Activation Condition
  – Expiration Condition
  – Id

\[ \langle v, Act, Exp, id \rangle \]
Normative formula ($\nu$)

- Annotated deontic formula is of the form
  \[ X_{\alpha:\rho} \varphi \circ \Gamma \]

- Where $X$ is the norm type:
  - $O$ – for obligations
  - $F$ – for prohibitions

- $\varphi$ is the targeted formula (actions in a plan)

- And $\Gamma$ is a conjunction of constraints
Previous Normative Systems

• Two extremes of norm processing
  – Blanket plan retractions
    (Normative AgentSpeak)
  – Every norm checked at every plan step
    (BOID)

• Decision about compliance too simplistic
  – Made before real repercussions are known or
  – Non-compliance simply not an option
• We propose something in-between
  – Fine grained
  – Efficient
• Effect of norms calculated at norm receipt
• Decision to comply delayed as much as possible
Reasoning about Norms

• Three key processes:
  – Update norms (Resolve Conflicts)
  – Annotate Plan Library
  – Apply normative restrictions to plans
nu-BDI

Perceive Events ➔ Update Beliefs ➔ Update Norms ➔ Resolve Conflicts ➔ Annotate Plans ➔ Select Relevant Plans ➔ Select Applicable Plans ➔ Select Compliant Plans ➔ Add Plan to Intentions ➔ Execute Intentions
Updating Norms

- Norms can be in two “states”
  - Abstract
  - Specific (or Active)
- When received by agent – abstract norms
- When activation condition holds – new specific norms created
Example Norm Update

- Abstract Norm
  \[ F_{A,R} \text{moveTo}(C) \circ C = X, \]
  \[ \text{tubeStrike}(X), \]
  \[ \neg \text{tubeStrike}(X), \]
  \[ \text{norm1} \]

- Specific Norm
  \[ F_{A,R} \text{moveTo}(C) \circ C = \text{london}, \]
  \[ \text{tubeStrike(}\text{london}\text{)}, \]
  \[ \neg \text{tubeStrike(}\text{london}\text{)}, \]
  \[ \text{norm1.1} \]

- New event occurs
  \[ \text{tubeStrike(}\text{london}\text{)} \]

- Specific Norm is deleted with event
  \[ \neg \text{tubeStrike(}\text{london}\text{)} \]
Annotating Plans

- Plans in the plan library are annotated as specific norms are created
- Normative formula is compared to steps in each plan
- Each step is associated with appropriate normative constraints
Example Plan Annotation

- **Plan**
  \[
  \langle +!\text{goTo}(C), \text{hasVehicle}(V), \text{getVehicle}(V), \text{moveTo}(C) \rangle
  \]

- **Specific Norm**
  \[
  \langle F_{A,R} \text{moveTo}(C) \odot C = \text{london}, \text{tubeStrike}(\text{london}), \neg \text{tubeStrike}(\text{london}), \text{norm1.1} \rangle
  \]

- **Resulting annotated plan**
  \[
  \langle +!\text{goTo}(C), \text{hasVehicle}(V), \text{getVehicle}(V) \odot T, \text{moveTo}(C) \odot C \neq \text{london} \rangle
  \]
Normative Plan Selection

- Similar to original plan selection
- Added check for satisfiability of a normative header
- Constraints from all steps
Example Plan Selection

\[ +!\text{goTo}(C), \text{hasVehicle}(V), \left[ \begin{aligned} \text{getVehicle}(V), \\
\text{moveTo}(C) \end{aligned} \right] \circ C \neq \text{london} \]

New event

\[ !\text{goTo}(\text{london}) \quad \text{hasVehicle}(\text{airplane}) \]

Belief Base

\[ \left[ \begin{aligned} \text{getVehicle}(\text{airplane}), \\
\text{moveTo}(\text{london}) \end{aligned} \right] (C = \text{london}, V = \text{airplane}) \]

Resulting Plan

But

\[ (C = \text{london} \land C \neq \text{london}) \rightarrow \bot \]
Conclusions

• Contributions
  – New norm representation formalism
  – Very fine grained control of normative stipulations
  – Efficient method for processing norms
  – Integrated with practical agent interpreter
Future Work

• Refine norm processing with
  – Deadlines (for obligations)
  – Integrate algorithms for normative conflict detection and resolution
QUESTIONS?