Using constraints for Norm-aware BDI Agents

Felipe Meneguzzi – CMU

Wamberto Vasconcelos – Aberdeen

Nir Oren – Aberdeen

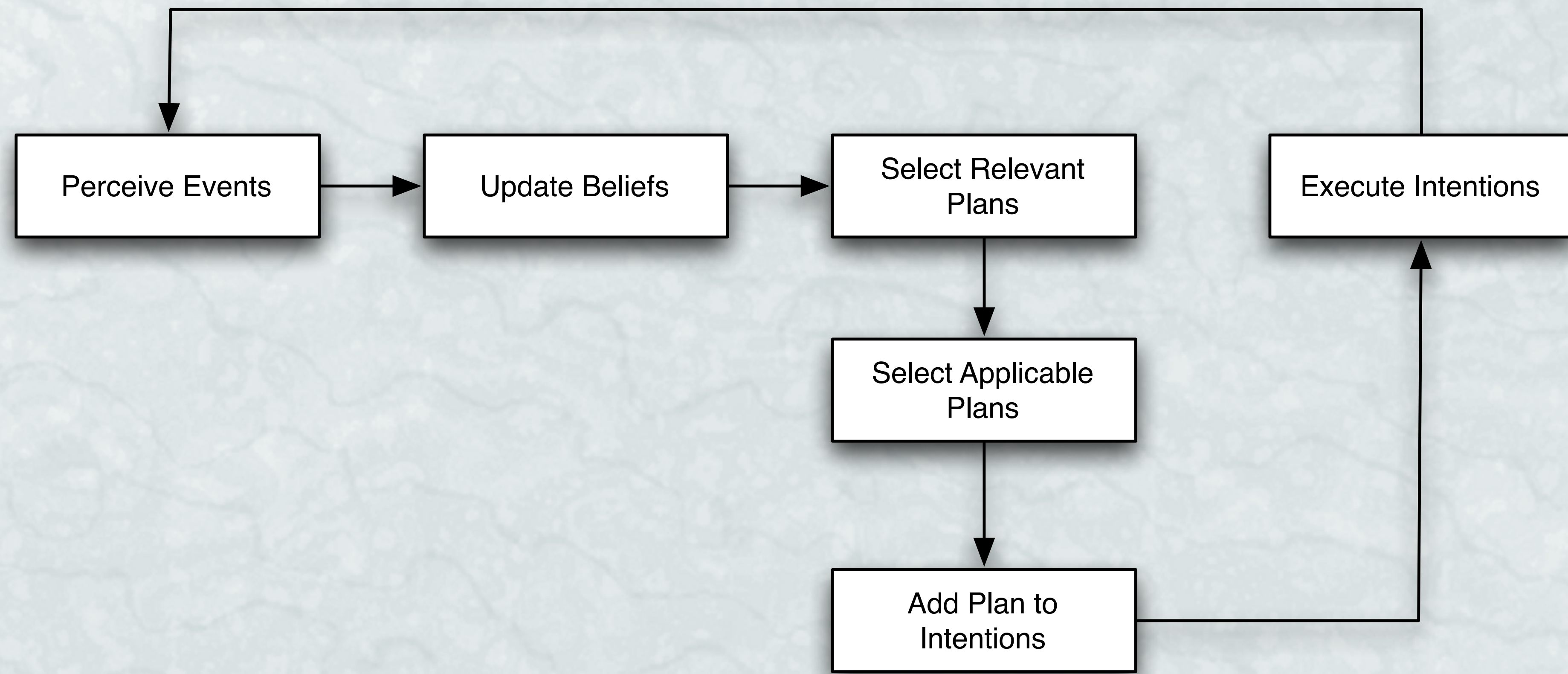
Outline

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

BDI Model

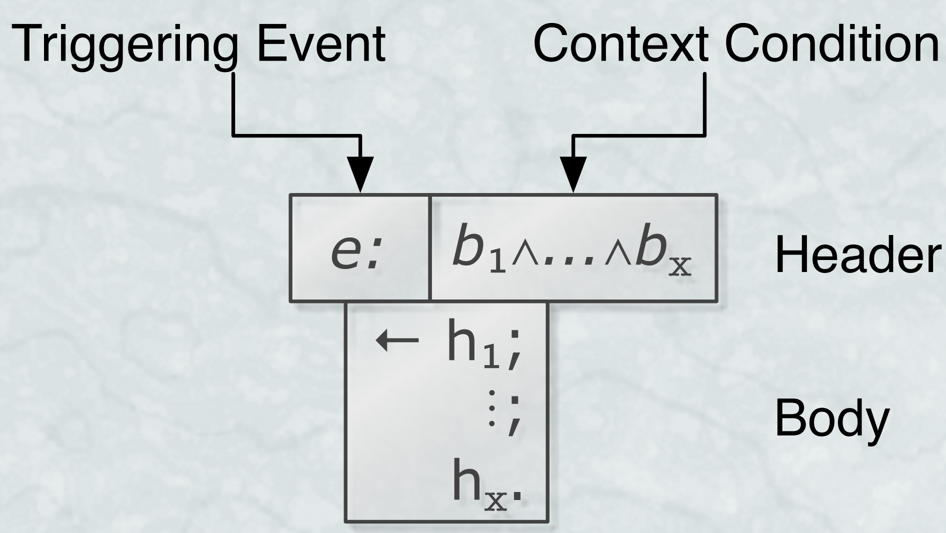
- Beliefs-Desires-Intentions
- Philosophical model of *practical reasoning*
  - Describes how reasoning occurs with *limited resources*
  - Intuitive way of describing reasoning
  - Widely used in the implementation of software agents
  - Has a strong theoretical background
  - Various open implementations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

BDI Reasoning



Plan Selection

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



Plan Example

$$\left\langle +!goTo(C), hasVehicle(V), \begin{bmatrix} getVehicle(V), \\ moveTo(C) \end{bmatrix} \right\rangle$$

New event | **Belief Base**

$$!goTo(london) \quad hasVehicle(airplane)$$

Resulting Plan

$$\begin{bmatrix} getVehicle(airplane), \\ moveTo(london) \end{bmatrix} (C = london, V = airplane)$$

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

$$\langle \nu, Act, Exp, id \rangle$$

Normative formula (ν)

- Annotated deontic formula is of the form

$$\mathrm{X}_{\alpha:\rho}\varphi \circ \Gamma$$

- Where X is the norm type:
  - O – for obligations
  - F – for prohibitions
- φ is the targeted formula (actions in a plan)
- And Γ 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

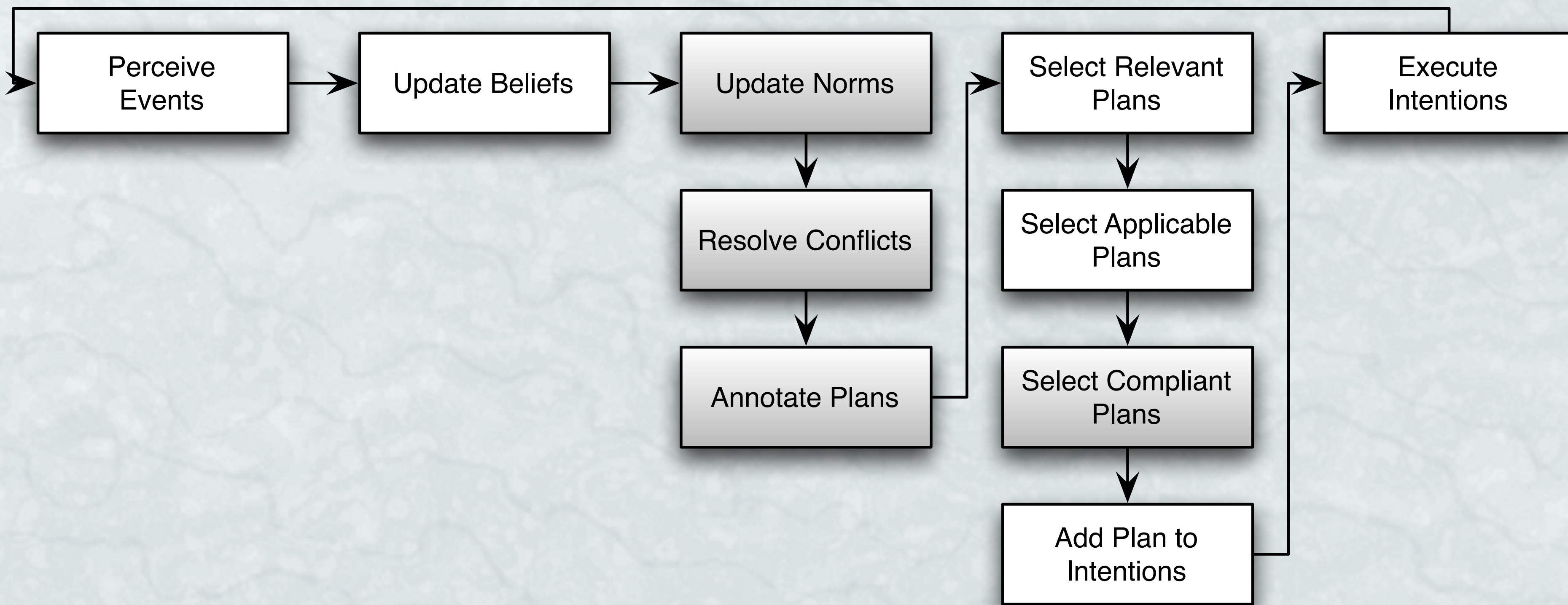
Architecture Desiderata

- 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

$$\left\langle \begin{array}{l} \mathsf{F}_{A:R}\, moveTo(C) \circ C = X, \\ tubeStrike(X), \\ \neg tubeStrike(X), \\ norm1 \end{array} \right\rangle$$

- New event occurs

$$tubeStrike(london)$$

- Specific Norm

$$\left\langle \begin{array}{l} \mathsf{F}_{A:R}\, moveTo(C) \circ C = london, \\ tubeStrike(london), \\ \neg tubeStrike(london), \\ norm1.1 \end{array} \right\rangle$$

- Specific Norm is deleted with event

$$\neg tubeStrike(london)$$

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

$$\left\langle \begin{array}{l} +!goTo(C), hasVehicle(V), \\ \begin{bmatrix} getVehicle(V), \\ moveTo(C) \end{bmatrix} \end{array} \right\rangle$$

- Specific Norm

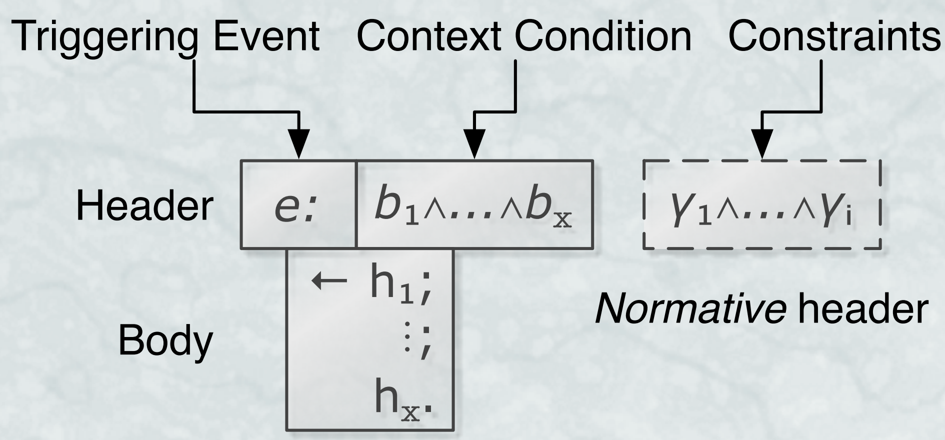
$$\left\langle \begin{array}{l} F_{A:R}\, moveTo(C) \circ C = london, \\ tubeStrike(london), \\ \neg tubeStrike(london), \\ norm1.1 \end{array} \right\rangle$$

- Resulting annotated plan

$$\left\langle \begin{array}{l} +!goTo(C), hasVehicle(V), \\ \begin{bmatrix} getVehicle(V) \circ \mathrm{T}, \\ moveTo(C) \circ C \neq london \end{bmatrix} \end{array} \right\rangle$$

Normative Plan Selection

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



Example Plan Selection

$$\left\langle +!goTo(C), hasVehicle(V), \begin{bmatrix} getVehicle(V), \\ moveTo(C) \end{bmatrix} \circ C \neq london \right\rangle$$

New event **Belief Base**

$$!goTo(london) \quad hasVehicle(airplane)$$

Resulting Plan

$$\begin{bmatrix} getVehicle(airplane), \\ moveTo(london) \end{bmatrix} (C = london, V = airplane)$$

But

$$(C = london \wedge C \neq london) \rightarrow \bot$$

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?