A First-Order Formalization of Commitments and Goals for Planning

Felipe Meneguzzi\textsuperscript{1}, Pankaj Telang\textsuperscript{2} and Munindar Singh\textsuperscript{2}

\textsuperscript{1}Pontifical Catholic University of Rio Grande do Sul
felipe.meneguzzi@pucrs.br
\textsuperscript{2}North Carolina State University
Motivation

• Commitments have been extensively studied in MAS
  • Encode high-level social relations between agents
  • Define communication protocols among agents

• Previous formalizations
  • Operational semantics for goals and commitments, and their interaction
  • Propositional planning formalization
Commitment Lifecycle

- Formally
  \[ C(\text{Debtor}, \text{Creditor}, \text{antecedent}, \text{consequent}) \]

- E.g.
  \[ C(\text{buyer}, \text{seller}, \text{goods}, \text{paid}) \]
Goal Lifecycle

- Formally
  \[ G(Agent, pg, s, f) \]
- E.g.
  \[ G(buyer, needsgoods,goods,deadline) \]
Relating Commitments and Goals

• Practical Rules relating commitments and goals

• Let $G = G(\text{buyer}, \top, \text{goods}, \bot)$
  and $C = C(\text{buyer, seller, goods, pay})$

• **Entice Rule**: If $G$ is active and $C$ is null, buyer creates $C$


$$\langle G^A, C^N \rangle$$

\[ create(C) \]

• **Motivation**: Buyer can achieve its goals of goods by creating the commitment to pay for them to Seller
Hierarchical Task Network Planning

- Generates a plan by successive refinement of tasks
- Non-primitive Tasks - abstract, high-level tasks to be decomposed
- Primitive Tasks - cannot be further decomposed (operators)
- Multiple implementations (e.g. JSHOP2, SHOP2)
- Abstraction of choice for agent programming languages
HTN Planning for Commitments and Goals

- Formalization of commitment protocols in terms of HTN planning

- Axioms enforcing state transition model for goals and commitments

- Planning Operators describing transitions (e.g. create, suspend, etc.)

- HTN Methods for practical rules (e.g. entice, negotiate, etc.)

- Allows HTN planner to be used to validate commitment protocols
A first-order formalization

- Propositional formalization had several limitations
- Limited expressivity

New First-order formalization:
- Domain independent axioms, methods and operators
- Domain dependent axioms, costs, methods and operators
- Useful patterns of behavior
We now develop the logical rules, operators, and methods in
for piecemeal payments, only the amount received may be
Compensation.

one protocol instance.
for both orders via one check. This is a clear case of flexibil-
close succession, the merchant may ship both of the ordered
(tions would determine whether payments may be split).

Challenge is to capture different ways in which interaction
variables of the antecedent. In order to accomplish this, each
ple, we define rules that compute a commitment's state in

### Commitment Axioms

\[
\begin{align*}
\text{null}(C, Ct, \tilde{C}v) & \leftarrow \neg \text{var}(C, Ct, \tilde{C}v) \\
\text{conditional}(C, Ct, \tilde{C}v) & \leftarrow \text{active}(C, Ct, \tilde{C}v) \land \neg p(C, Ct, \tilde{C}v) \\
\text{detached}(C, Ct, \tilde{C}v) & \leftarrow \text{active}(C, Ct, \tilde{C}v) \land p(C, Ct, \tilde{C}v)
\end{align*}
\]

### Goal Axioms

\[
\begin{align*}
\text{null}(G, Gt, \tilde{G}v) & \leftarrow \neg \text{var}(G, Gt, \tilde{G}v) \\
\text{inactiveG}(G, Gt, \tilde{G}v) & \leftarrow \neg \text{null}(G, Gt, \tilde{G}v) \\
& \land \neg f(G, Gt, \tilde{G}v) \land \neg s(G, Gt, \tilde{G}v) \\
& \land \neg \text{terminalG}(G, Gt, \tilde{G}v) \land \neg \text{suspendedG}(G, Gt, \tilde{G}v) \\
& \land \neg \text{activeG}(G, Gt, \tilde{G}v)
\end{align*}
\]

### Commitment Operators

\[
\begin{align*}
\text{operator} \ \text{!create}(C, Ct, De, Cr, \tilde{C}v), \\
\text{pre}(\text{commitment}(C, Ct, De, Cr) \land \text{null}(C, Ct, \tilde{C}v)), \\
\text{del}(), \ \text{add}(\text{var}(C, Ct, \tilde{C}v))
\end{align*}
\]

\[
\begin{align*}
\text{operator} \ \text{!suspend}(C, Ct, De, Cr, \tilde{C}v), \\
\text{pre}(\text{commitment}(C, Ct, De, Cr) \land \text{active}(C, Ct, \tilde{C}v)), \\
\text{del}(), \ \text{add}(\text{pending}(C, Ct, \tilde{C}v))
\end{align*}
\]

### Goal Operators

\[
\begin{align*}
\text{operator} \ \text{!consider}(G, Gt, X, \tilde{G}v), \\
\text{pre}(\text{goal}(G, Gt, X) \land \text{null}(G, Gt, \tilde{G}v) \land \text{pg}(G, Gt, \tilde{G}v)), \\
\text{del}(), \ \text{add}(\text{var}(G, Gt, \tilde{G}v))
\end{align*}
\]

\[
\begin{align*}
\text{operator} \ \text{!activate}(G, Gt, X, \tilde{G}v), \\
\text{pre}(\text{goal}(G, Gt, X) \land \text{inactiveG}(G, Gt, \tilde{G}v)), \\
\text{del}(), \ \text{add}(\text{activatedG}(G, Gt, \tilde{G}v))
\end{align*}
\]
Domain Dependent Definitions

- Axioms plus Domain-dependent operators

- Commitment Axioms
  \[ p(C,Ct,\vec{C}v) \leftarrow \text{commitment}(C,Ct,De,Cr) \land \varphi \]
  \[ q(C,Ct,\vec{C}v) \leftarrow \text{commitment}(C,Ct,De,Cr) \land \pi \]

- Goal Axioms
  \[ pg(G,Gt,\vec{G}v) \leftarrow \text{goal}(G,Gt,X) \land \omega \]
  \[ s(G,Gt,\vec{G}v) \leftarrow \text{goal}(G,Gt,X) \land \varsigma \]
  \[ f(G,Gt,\vec{G}v) \leftarrow \text{goal}(G,Gt,X) \land \vartheta \]

- Axioms Generated automatically using a compilation tool

- Plus any domain-specific operators (e.g. purchase, ship, etc)
Patterns of Behavior

• Concession Pattern
  2 commitments

• C2 - merchant commits to delivering the goods upon a $20 payment from the customer

• C3 - customer commits to pay $80 upon receiving the goods

• By creating commitments C2 and C3, the customer has one possible way of achieving its goal
Conclusions and Future Work

• A FO formalization of goals and commitment protocols
• Multiple interacting instances of the same goals and commitments
• Piecemeal progress, concession, consolidation and compensation

• Future Work
• Reasoning about probabilities
• Modelling non-cooperative partners
Questions?