Meta-axioms and Complex Preferences in Evolving Logical Agents

Stefania Costantini
University of L’Aquila, Italy

Pierangelo Dell’Acqua
Linköping University, Sweden

Luis Moniz Pereira
New University of Lisbon, Portugal

Francesca Toni
Imperial College London, UK

LPMAS, Lexington, Kentucky (USA) - July 9th, 2011
Aim

- To develop an application that intelligently and unintrusively help and assist people in performing their daily tasks

- We consider environments that are unknown or for some reason difficult to cope with

- Possible users:
  - elderly people
  - disabled people
To obtain a flexible interaction with the user we devised a multi-layered agent model composed by:

- a base layer PA (for Personal Assistant), and
- a meta-layer MPA (for Meta-Personal Assistant)

PA responsible for the direct interaction with the user

- e.g., behavioral suggestions, appliance manipulations

MPA responsible for:

- supervising and checking the activity of PA, and
- updating PA to correct inadequacies and generate appropriate agent behavior
Agent Model

> Agent Program

- An *agent program* is a tuple $P = \langle PA, MPA, C, CT \rangle$ of software components:
  - $PA$ and $MPA$ are logic programs
  - $C$ is the control component
  - $CT$ contains declarative control information directives
Agent Model

> Evolutionary Semantics

- An agent starts from an initial program $P_0$

- Changes in the environment as well as agent’s own self-modifications are modeled as program transformation steps:

  $$\langle PA_i, MPA_i, C_i, CT_i \rangle \xrightarrow{\mathcal{U}(C_i, CT_i, w_i)} \langle PA_{i+1}, MPA_{i+1}, C_{i+1}, CT_{i+1} \rangle$$

  where $\mathcal{U}$ is the underlying control mechanism and $w_i$ is the external environment


Agent Model

> Evolutionary Semantics

- We thus have a Program Evolution Sequence $PE = [P_0, ..., P_n]$

- We have a corresponding Semantic Evolution Sequence $ME = [M_0, ..., M_n]$ where $M_i$ is the semantic account of $P_i$ according to the specific language and the chosen semantics

- The pair $\langle PE; ME \rangle$ is called the *Evolutionary Semantics* of the agent
Agent Model

> Evolutionary Semantics

- The different languages and formalisms will influence the following key points:
  - when a transition from $P_i$ to $P_{i+1}$ takes place, i.e. which are the external and internal factors that determine a change in the agent
  - which kind of transformations are performed
  - which semantic approach is adopted, i.e., how $M_i$ is obtained from $P_i$

- To describe and semantically account for the dynamic changes that MPA performs on PA we rely upon EVOLLP, an extension of logic programming that allows to model the dynamics of knowledge bases
Agent Model

> Temporal Logic Rules and Meta Rules

- For runtime verification of PA’s activities, MPA exploits temporal logic rules

- Previously, we proposed a framework called Agent-Interval LTL (A-ILTL)
  - an extension of linear time logic (LTL) to include time intervals
  - A-ILTL temporal rules defined upon a logic programming-like set of formulas (where variables implicitly universally quantified)
Agent Model

> A-ILTL operators

<table>
<thead>
<tr>
<th>A-ILTL Op$^k$</th>
<th>OP(m,n;k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau(t)$</td>
<td>NOW$(t)$</td>
</tr>
<tr>
<td>$X^k$</td>
<td>NEXT$(1; k)$</td>
</tr>
<tr>
<td>$X^k_j$</td>
<td>NEXT$(j; k)$</td>
</tr>
<tr>
<td>$F^k$</td>
<td>FINALLY$(1; k)$</td>
</tr>
<tr>
<td>$F^k_m$</td>
<td>FINALLY$(m; k)$</td>
</tr>
<tr>
<td>$G^k$</td>
<td>ALWAYS$(1; k)$</td>
</tr>
<tr>
<td>$G^k_{m,n}$</td>
<td>ALWAYS$(m, n; k)$</td>
</tr>
<tr>
<td>$G^k_{(m,n)}$</td>
<td>ALWAYS$_2$(m, n; k)</td>
</tr>
<tr>
<td>$N^k$</td>
<td>NEVER$(1; k)$</td>
</tr>
<tr>
<td>$N^k_{m,n}$</td>
<td>NEVER$(m, n; k)$</td>
</tr>
</tbody>
</table>

$m$ and $n$ denote the time interval in which the formula must hold
$k$ is the frequency
A-ILTL rules

- The following A-ILTL rule checks that an agent respects the blind commitment to its goals

\[ NEVER(goal(G), \text{deadline}(G, T), \text{NOW}(T_1), T_1 \leq T, \text{not achieved}(G), \text{dropped}(G)) \]

- In order to fulfill their semantic specification, A-ILTL rules must be ground when they are evaluated. A-ILTL rules allow that time instants are variables which are instantiated by an evaluation context

\[ FINALLY(T; F) G :: goal(G), \text{priority}(G, P), \text{timeout}(P, T), \text{frequency}(P, F) \]
A-ILTL rules with Repairs

- During the monitoring process each A-ILTL rule is attempted at a certain frequency.

- If the current state of affairs does not satisfy any A-ILTL rule, some kind of repair action has to be undertaken wrt. the violated rule:

\[
\text{NEVER} \left( \text{not achieved}(G), \text{dropped}(G) \right) :: \\
\left( \text{goal}(G), \text{deadline}(G, T), \text{NOW}(T1), T1 \leq T \right) \div \text{inc\_comt}(T1)
\]

\[
\text{inc\_comt}(T) \leftarrow \text{level}(\text{commitment}, L), \text{increase\_level}(L, L1), \\
\text{assert}(\text{inc\_comt\_at}(T)), \text{assert}(\text{neg}(\text{commitment\_mod}(L))), \\
\text{assert}(\text{commitment\_mod}(L1))
\]
To express that an event must occur before another event, A-ILTL supports the notion of *ordered conjunction*, a conjunction of literals having:

- an explicit timestamp, and

- a linear order

Example:

\[
\text{ALWAYS} \ ( enroll(S, C) \ll attend(S, C) \ll exam(S, C, Grade) ) :: student(S), course(C)
\]
Preferences: Defining a recipe for a dessert

- The construct *icecream > zabaglione* is called a *p-list* (preference list) and states that with the given ingredients one can obtain ice-cream or zabaglione being the former preferred.

- In preparing the dessert, one can employ skim-milk or whole milk. If on a diet, the former is preferred: *skimmilk > wholemilk ← diet*

- Finally, to spice the dessert, one chooses via the *p-set* \{*chocolate, nuts, coconut | less_caloric*\} the less caloric among chocolate, nuts, and coconut.
Preferences: Defining a recipe for a dessert

\[ \text{icecream} > \text{zabaglione} \leftarrow \text{egg, sugar, (skimmilk} > \text{wholemilk} \leftarrow \text{diet)}, \]
\[ \{\text{chocolate, nuts, coconut} \mid \text{less_caloric}\} \]

\[ \text{less_caloric}(X,Y) \leftarrow \text{calory}(X,A), \text{calory}(Y,B), A < B \]

\[ \text{calory(nuts,2)} \]

\[ \text{calory(ciocolate,3)} \]

\[ \text{calory(coconut,3)} \]

- In our framework preferences can change non-monotonically as the agent’s knowledge base evolves over time

- The evolutionary semantics can easily accommodate this kind of preference reasoning
In this paper we propose a further extension to our approach to preferences inspired by the work of Liu and von Wright.

They introduced the concept of complex preference where an agent prefers $\phi$ over $\psi$ if, for any plausible (i.e., presumably reachable) world where $\psi$ holds, there exists a world which is at least as good as this world and at least as plausible where $\phi$ is true.

They write $B(\psi \rightarrow \langle H \rangle \phi)$ where $H$ is a new modality, and the reading is Hopefully $\phi$.

Semantically: if $\mathcal{M}$ is a preference model encompassing a set of worlds $W$ and $s, t \in W$, $\leq$ is a reachability relation meaning at least as plausible and $\preceq$ a preference relation, we have that:

$$\mathcal{M}, s \models H \phi \iff \text{for all } t \text{ with both } s \leq t \text{ and } s \preceq t : \mathcal{M}, s \models \phi$$
Complex Preferences

- To define the $H$ operator in our setting, we allow forms of reasoning to be expressed on *possibility* and *necessity*, analogous to those of modal logic.

- The possible worlds that we consider are the answer sets of an ASP program $\Pi$.

Given atom $A$, we say that:

- $A$ is *possible* if it belongs to some answer set of $\Pi$
  
  $P(A)$ holds whenever $\exists M \in \{M_1, \ldots, M_k\}$ such that $A \in M$

- $A$ is *necessary* if it belongs to the intersection of all the answer sets
  
  $N(A)$ holds whenever $A \in (M_1 \cap \ldots \cap M_k)$
Complex Preferences

- Possibility and necessity can be evaluated within a context

Let $E(\text{Args})$ be either a possibility or a necessity expression

The corresponding *contextual* expression has the form $E(\text{Args}) : \text{Context}$ where \text{Context} is a set of ground facts and rules

$E(\text{Args}) : \text{Context}$ holds wrt. $\Pi$ iff $E(\text{Args})$ holds wrt. $\Pi \cup \text{Context}$

- In this approach, one is able for example to define meta-axioms stating that a proposition $Q$ is plausible w.r.t. theory $T$ if it is *possible* in at least two different worlds given some context $C$:

  \[
  \text{plausible}(T, Q, C) \leftarrow P(T, I, Q) : C, P(T, J, Q) : C, I \neq J
  \]
Complex Preferences

- We can define the $H$ operator explicitly stating which is the aspect that makes a world (in which $\phi$ holds) preferable.

In fact, by $\phi : H(G)$ we mean that, assuming $\phi$, we expect that $G$ will hold in some reachable world, that in our setting is an answer set of $\Pi$.

Thus, $G$ is the reason why reachable worlds in which $\phi$ holds are preferred.

**Definition** Given an answer set program $\Pi$ with answer sets $M_1, \ldots, M_k$, an atom $\phi$ and an atom $G$, the expression $\phi : H(G)$ holds wrt. $\Pi$ whenever the contextual possibility expression $P(G) : \phi$ holds wrt. $\Pi$. 

Complex Preferences

- We can also extend the operator to a form $\phi : H[N](G)$ meaning that, given $\phi$, we expect the hoped-for property $G$ to hold in exactly $N$ different possible worlds.

**Definition** Given an answer set program $\Pi$ with answer sets $M_1, \ldots, M_k$, an atom $\phi$ and an atom $G$, the expression $\phi : H[n](G)$ holds wrt. $\Pi$ iff:

1) there exist $\{v_1, \ldots, v_n\} \subseteq \{1, \ldots, k\}$ such that for every $v_i \in \{v_1, \ldots, v_n\}$ $P(v_i, G) : \phi$ holds, and

2) whenever $P(w, G) : \phi$ holds, then $w \in \{v_1, \ldots, v_n\}$

Note that if $\phi : H[0](G)$ holds, then $G$ does not hold in any possible world.
Complex Preferences

- Note that:
  - The answer set module $T$, where to evaluate the operator $H$, can be explicitly specified $\phi : H(T, G)$ and $\phi : H[N](T, G)$
  - The atom $G$ occurring in the above definitions can be easily generalized to be a conjunction of atoms
Complex Preferences

- We introduce new preference expressions involving the operator $H$

**Definition** Given the atoms $A, B, C$ and the answer set module $T$, the expression $A > B : H(T, C)$ holds if $A : H(T, C)$ holds

Intuitively, we prefer $A$ to $B$ whenever, by assuming $A$, a situation where $C$ holds can be possibly reached
Complex Preferences

- We can compare $A$ and $B$ wrt. how often a state of affairs can be reached.

  Thus, we prefer $A$ over $B$ if by assuming $A$ it is more plausible to reach $C$ than it would have been by assuming $B$.

**Definition** Given the atoms $A, B, C$ and the answer set module $T$, the expression $A > B : \max H(T, C)$ iff $A : H[N_A](T, C)$ and $B : H[N_B](T, C)$ and $N_A \geq N_B$.

- The atoms $A$ and $B$ can have several meanings:
  - they can encode plans like $\text{plan}(p, a_1, \ldots, a_n)$, where $p$ is the identifier of the plan consisting of a sequence $a_1, \ldots, a_n$ of actions
  - here the operator $H$ evaluates plans wrt. some wished-for outcome
Examples

- One may choose to prefer a certain food in the hope that it is healthy:

\[ eat(pasta) > eat(meat) : H(healthy) \]

- A stronger formulation - one chooses to prefer a food that in most of the situations that can reasonably envisaged will procure better health:

\[ eat(pasta) > eat(meat) : maxH(healthy) \]

where one chooses the food that is presumably healthier (i.e., it can be concluded to be healthy in most cases)
Examples

- It can be convenient to include more context in the $H$ operator:

  $$A > B : \text{Context} : H(T, C)$$

  $$A > B : \text{Context} : \text{max}H(T, C)$$

  where $Context$ is a conjunction of atoms to be added to the ASP module before evaluating $H$

- This extension allows for instance:

  $$\text{eat(fruit\_salad)} > \text{eat(cake)} : \text{diabetes} : \text{max}H(\text{healthy})$$
Finally, it will be useful to introduce a generalized form of preference set to allow for the representation below:

This representation takes the varieties of food generated by \( food(F) \) and generates a preference list containing the various kinds of foods ordered according to the degree of healthiness interpreted as the value \( diabetes : H[N]healthy(f) \) for each food \( f \):

\[
\{food(F), eat(F) : diabetes : H(healthy)\}
\]
Complex Preferences

- **Definition** A *modal preference set* is an expression of the form

\[
\{ p(X_1, \ldots, X_n), q(X_1, \ldots, X_n) : B : H(E) \}
\]

where: \( p, q \) are predicates, and \( X_1, \ldots, X_n \) are variables

\( B \) is a conjunction of atoms not involving the \( X_i \)s and involving terms \( Y_1, \ldots, Y_v \)

\( E \) is a conjunction of atoms possibly involving the \( X_i \)s and the \( Y_j \)s

This expression stands for the preference list \( A_1 > \ldots > A_s \) where:

- \( A_i \)s are all the ground atoms \( q(t_1, \ldots, t_n) \) for which \( p(t_1, \ldots, t_n) \) holds

- for each \( A_j, A_k \) in this list, where \( A_j = q(g_1, \ldots, g_n) \) and \( A_k = q(h_1, \ldots, h_n) \), \( A_j \) precedes (is preferred to) \( A_k \) iff the following holds

\[
A_j > A_k : B : \text{max}H(T, E)
\]
The preference expressions given above allows for the definition of a variety of meta-statements.

The following A-ILTL rule states that an agent, having a deadline, always prefers to pursue the goal (adopt the intention) for which there exists a plan such that the goal can with the highest possible confidence be reached within the deadline.

\[
\text{ALWAYS } \text{adopt\_intention}(G, P), \\
\{ \text{plan}(G, P), \text{goal}(G) \text{ : deadlin}(t) : H(\text{reached}(G, P, t)) \}
\]
Future Work

- Even though most of the proposed features have been simulated in DALI, we intend to fully implement an instance of the proposed framework starting from EVOLP, DALI and KGP agents (which are fully-defined and fully-implemented approaches).

- Then, we intend to experiment our setting on practical cases.

- Finally, we aim at exploring a generalization of our setting to the multi-agent case, allowing A-ILTL and preference expressions to be expressed upon several agents.