Enlightened Update in a Dialogue Game

Luciana Benotti

TALARIS Team
Loria/INRIA Lorraine, France
Luciana.Benotti@loria.fr
http://www.loria.fr/~benottil/

30 May 2007
What is Enlightened Update?

Enlightened Update (EU) theory studies how common ground is constructed and exploited in coordinated activities (collaborative physical tasks and cooperative conversation).

General Claim: Coordinated activities move forward through:

- The **effects** of an action (or utterance), but also through
- The **assumptions** about the state of the world on which the success of the action depends

Enlightened Update and Tacit Action

It is mutual information that:

- C is a precondition for the action A, and
- An actor S executes A

⇒ S believes C

Tacit actions are executed to make C true (if possible)

The authors of the theory argue that these tacit actions are intended by S when performing A.
Enlightened Update and Tacit Action

It is mutual information that:

- C is a precondition for the action A, and
- An actor S executes A
- C is a precondition for the action A, and
- An actor S executes A
- C is false

⇒ S believes C

⇒
Enlightened Update and Tacit Action

It is mutual information that:

- C is a precondition for the action A, and
- An actor S executes A

\[\Rightarrow S \text{ believes } C\]

- C is a precondition for the action A, and
- An actor S executes A
- C is false

\[\Rightarrow \text{Tacit actions are executed to make } C \text{ true} \quad \text{(if possible)}\]
Enlightened Update and Tacit Action

It is mutual information that:

- C is a precondition for the action A, and
- An actor S executes A
  \[ \Rightarrow S \text{ believes } C \]

- C is a precondition for the action A, and
- An actor S executes A
- C is false
  \[ \Rightarrow \text{Tacit actions are executed to make } C \text{ true (if possible)} \]

The authors of the theory argue that these tacit actions are \textit{intended} by S when performing A.
A New Laboratory for the Theory

- The theory of Enlightened Update was implemented and tested in the conversational agent COREF.
- In COREF, both dialogue participants are assumed to have the same background knowledge.
- We implement and test the theory when one of the participants has incomplete background knowledge.
- We implement enlightened update using planning rather than abduction.
- Our laboratory will be a text adventure game.

The New Laboratory
FrOzA is a text adventure game:

1. FrOzA starts by describing the player location in the game world
2. The player types instructions that are interpreted and executed by the game
3. After executing an instruction, FrOzA describes its effects to the player

FrOzA is an extension of the text adventure engine FrOz developed by Koller, Debusmann, Gabsdil, and Striegnitz. 2004.
The Game Architecture

"Kiss the princess."

"The princess is happy."

Language

Understanding

Language

Generation

Action

Handling

Action Database

GAME SCENARIO

Player Beliefs

Domain Ontology

Inference tools are used to query and modify a game scenario.

- During action handling, the prover Racer and the planner Blackbox collaborate in order to find tacit actions
A Game Scenario: The Actions

- The action database includes all the actions that can be executed in a game scenario.
- The actions are represented in a STRIPS format.

<table>
<thead>
<tr>
<th>action:</th>
</tr>
</thead>
<tbody>
<tr>
<td>take(patient:X)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>preconditions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessible(X)</td>
</tr>
<tr>
<td>takeable(X)</td>
</tr>
<tr>
<td>not(has-loc(X player1))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>effects:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(delete) (\exists Y.\text{has-loc}(X \ Y))</td>
</tr>
<tr>
<td>(add) has-loc(X player1)</td>
</tr>
</tbody>
</table>

- Actions and preconditions are a natural way of thinking about enlightened update.
A Game Scenario: The Knowledge Bases

The world KB contains complete information about the world state.

The world state

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1, couch1)
- apple(apple1)
- has-loc(apple1, couch1)
- drawing-room(room1)
- has-loc(couch1, room1)
- accessible(apple1)
- takeable(apple1)
- room(room2)
- dragon(dragon1)
- has-loc(dragon1, room2)

The player beliefs

- player(player1)
A Game Scenario: The Knowledge Bases

The world KB contains complete information about the world state.

The world state

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- has-loc(apple1,couch1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)
- room(room2)
- dragon(dragon1)
- has-loc(dragon1,room2)

The player beliefs

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- has-loc(apple1,couch1)
- drawing-room(room1)
- has-loc(couch1,room1)
A Game Scenario: The Knowledge Bases

The world KB contains complete information about the world state

The world state

player(player1)
couch(couch1)
brown(couch1)
has-loc(player1,couch1)
apple(apple1)
has-loc(apple1,couch1)
drawing-room(room1)
has-loc(couch1,room1)
accessible(apple1)
takeable(apple1)
room(room2)
dragon(dragon1)
has-loc(dragon1,room2)

The player beliefs

player(player1)
couch(couch1)
brown(couch1)
has-loc(player1,couch1)
apple(apple1)
has-loc(apple1,couch1)
drawing-room(room1)
has-loc(couch1,room1)
accessible(apple1)
takeable(apple1)
A Game Scenario: The Knowledge Bases

The world KB contains **complete information** about the world state

The **world state**

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- has-loc(apple1,couch1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)
- room(room2)
- dragon(dragon1)
- has-loc(dragon1,room2)

The **player beliefs**

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- has-loc(apple1,couch1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)

You are in a brown couch.

There is an apple in the couch.

The couch is in a drawing-room.

> take the apple
A Game Scenario: The Knowledge Bases

The world KB contains **complete information** about the world state

The **world state**

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- has-loc(apple1,couch1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)
- room(room2)
- dragon(dragon1)
- has-loc(dragon1,room2)

The **player beliefs**

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- has-loc(apple1,couch1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)

You are in a brown couch.

There is an apple in the couch.

The couch is in a drawing-room.

> take the apple
A Game Scenario: The Knowledge Bases

The world KB contains **complete information** about the world state

The world state

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)
- room(room2)
- dragon(dragon1)
- has-loc(dragon1,room2)

The player beliefs

- player(player1)
- couch(couch1)
- brown(couch1)
- has-loc(player1,couch1)
- apple(apple1)
- drawing-room(room1)
- has-loc(couch1,room1)
- accessible(apple1)
- takeable(apple1)
The world KB contains complete information about the world state

The world state

player(player1)
couch(couch1)
brown(couch1)
has-loc(player1,couch1)
apple(apple1)
has-loc(apple1,player1)
drawing-room(room1)
has-loc(couch1,room1)
accessible(apple1)
takeable(apple1)
room(room2)
dragon(dragon1)
has-loc(dragon1,room2)

The player beliefs

player(player1)
couch(couch1)
brown(couch1)
has-loc(player1,couch1)
apple(apple1)
has-loc(apple1,player1)
drawing-room(room1)
has-loc(couch1,room1)
accessible(apple1)
takeable(apple1)

You are in a brown couch.
There is an apple in the couch.
The couch is in a drawing-room.
> take the apple
You have the apple.
A Game Scenario: The Knowledge Bases

The world KB contains complete information about the world state

The world state

player(player1)
couch(couch1)
brown(couch1)
has-loc(player1,couch1)
apple(apple1)
has-loc(apple1,player1)
drawing-room(room1)
has-loc(couch1,room1)
accessible(apple1)
takeable(apple1)
room(room2)
dragon(dragon1)
has-loc(dragon1,room2)

We assume that the player KB contains:

- incomplete knowledge about the world state
- mutual information (shared by player and game) about player beliefs

The player beliefs

player(player1)
couch(couch1)
brown(couch1)
has-loc(player1,couch1)
apple(apple1)
has-loc(apple1,player1)
drawing-room(room1)
has-loc(couch1,room1)
accessible(apple1)
takeable(apple1)
Implementing and Testing the Theory
Life WITHOUT Tacit Actions

• Let us see how is life without tacit actions
• Imagine that you are giving instructions not to a computer game but to a person
• The person answers as in the screenshot
• Don’t you find this annoying? This system is not cooperating
Life WITH Tacit Action
(A Cooperative Game)

- A cooperative game would do whatever is needed in order to execute the player instruction.
- The actions executed:
  1. (tacit) stand up
  2. (tacit) take the key
  3. (public) unlock the chest

In FrOzA, the prover Racer and the planner Blackbox interact to find a sequence of tacit actions and perform and enlightened update to the game scenario.
What information should be used in order to discover tacit actions?

1. The **complete information** about the world state stored in the world KB?
2. The **incomplete knowledge** about the world state stored in the player KB?

Let’s consider both possibilities in turn ...
Tacit Action and Complete Knowledge

Suppose that tacit actions are found using the complete information about the world state stored in the world KB.

- The actions executed are:
  1. (tacit) take the golden key
  2. (tacit) unlock the chest with the golden key
  3. (public) open the chest
- But the player doesn’t know which key unlocks the chest!

- To unlock the chest with the golden key cannot be part of the intention of the player.
- Hence, it seems that what we need is the incomplete knowledge ...
Suppose that tacit actions are found using the incomplete information about the world state stored in the player KB.

- For this example no plan is found
- and this is just what we wanted.
- The player just does not have enough knowledge to leave this action tacit

But, what if ...
What if the golden key was stolen by a thief (without the player knowing) and the player says ‘Unlock the chest with the golden key’?

- In the player KB the key is accessible (it is on the table), then FrOzA decides to execute:
  1. (tacit) take the golden key
  2. (public) unlock the chest with the golden key
- But, this sequence of actions is not executable in the world because the key is no longer accessible (the thief has it)

We need both KBs:

- Tacit actions are inferred using the player KB
- Their execution is verified using the world KB
And there is more!

Incomplete Knowledge Leads to Learning ...
Incomplete Knowledge leads to Learning ...

Consider the following situation:

- The player doesn’t know which key unlocks the chest until she *tries* to unlock it with the right key.
- After the action succeeds, the player *knows* (and the game knows that the player knows) which key fits into the chest.

*Remember:* If C is a precondition for the action A, and S executes A publicly then S believes C.
Tacit Action and Learned Knowledge

Later in the game ...

The chest is again locked. This time the player just needs to make public her intention to open the chest.

- And the game infer the tacit action unlock:
  1. (tacit) take the golden key
  2. (tacit) unlock the chest with the golden key
  3. (public) open it

As knowledge grows, more actions can be left tacit.
Summary

- Enlightened update is an **indispensable ingredient** for a cooperative system.
- We have viewed:
  - Enlightened Update and Complete Knowledge
  - Enlightened Update and Incomplete Knowledge
  - Enlightened Update and Learned Knowledge
- Tacit actions should be inferred using **mutual information** about the beliefs of the actor, but their executability have to be verified wrt **complete information**.
- Tacit actions are **dynamic**: As the actor learns about the world, more actions can be left tacit.
One Further Challenge

Handling deeper incomplete knowledge:

- COREF and FrOzA assume that the activity participants share exactly the same action schemas.
- If this assumption is dropped, how can participants coordinate their action schemas?
- How do we abstract action schemas interacting with the world?
  - (sensible) *Hang the coat on the upright vacuum cleaner*
  - (not sensible) *Hang the coat on the upright cup*
- How do we know that the second one should fail and the first one might work?
Thanks for your attention!

Questions?