PENG Light meets the Event Calculus

Rolf Schwitter
Rolf.Schwitter@mq.edu.au
Motivation

• Research question:
  How to extend an existing controlled natural language so that we can specify knowledge about events and their effects (= periods during which states hold)?

• Approach:
  We look at a scenario where we can observe and describe a sequence of events as they unfold.

• Basically, "eye-witness" reports.
Motivation

• Research question:
  How to extend an existing controlled natural language so that we can specify knowledge about events and their effects (= periods during which states hold)?

• Approach:
  We look at a scenario where we observe and describe a sequence of events as they unfold.

• Basically, "eye-witness" reports.

• Think about a spy who observes and reports events as they unfold – if that helps.
PENG Light

- Simple sentences:
  - subject + verb + (complement)* + (modifier)*
- Complex sentences:
  - coordination (and, or)
  - subordination (if ... then, who, that, ...)
  - quantification (every, all, for every, ...)
  - negation (no, is not, does not, ...)
- Questions:
  - wh-questions
  - yes/no-questions
- Anaphoric references
PENG Light

- PENG Light sentences are translated incrementally during the writing process into first-order logic via discourse representation structures.
- During parsing the following activities occur in parallel:
  - anaphoric expressions are resolved
  - a discourse representation structure is generated
  - a paraphrase is produced
  - lookahead information is generated.
- A model builder tries to generate a finite model.
A Scenario

- John arrives at 10:10 with Flight AZ1777 in Palermo.
- John gets on a bus at the airport.
- The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
- John gets off the bus in Trapani.
A Scenario

- John arrives at 10:10 with Flight AZ1777 in Palermo.
- John gets on a bus at the airport.
- The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
- John gets off the bus in Trapani.

- The temporal structure of these sentences is simple.
A Scenario

- John arrives at 10:10 with Flight AZ1777 in Palermo.
- John gets on a bus at the airport.
- The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
- John gets off the bus in Trapani.

The temporal structure of these sentences is simple. 

Temporal expressions only occur as modifiers of events.
A Scenario

- John arrives at 10:10 with Flight AZ1777 in Palermo.
- John gets on a bus at the airport.
- The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
- John gets off the bus in Trapani.

- Temporal structure of these sentences is simple.
- Temporal expressions only occur as modifiers of events.
- These events have a linear temporal ordering.
A Scenario

- John arrives at 10:10 with Flight AZ1777 in Palermo.
- John gets on a bus at the airport.
- The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
- John gets off the bus in Trapani.

- For some events, no explicit time point is mentioned.
A Scenario

- The weather is bad.
- The wind is strong and the sea is rough.

- We can speak about states in PENG Light.
A Scenario

- If the weather is good then John boards the hydroplane at 14:10 and arrives on Marettimo Island at 15:35.
- If the weather is bad then Johns stays in Trapani and goes to the Albergo Maccotta.

• We can express conditional statements in PENG Light.
Question Answering

- When does John arrive in Palermo?
- Where does John get on a bus?
- When does the bus leave the airport?
- Who gets off the bus?

- These questions can be answered directly over the resulting knowledge base.
- **No** additional background knowledge is required.
The Scenario

- John arrives at 10:10 with Flight AZ1777 in Palermo.
- John gets on a bus at the airport.
- The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
- John gets off the bus in Trapani.
Question Answering

- Where is John now?
- Where is John at 13:30?
- Why does John stay in Trapani?

- These questions require background knowledge.
Human Observer

• Given:
  - John arrives at 10:10 with Flight AZ1777 in Palermo.
  - John gets on a bus at the airport.
  - The bus leaves the airport at 11:30 ...

• A human observer can easily infer:
  - John is on the bus at 12:00 but not anymore in Palermo at that time.
Human Observer

• Given:
  - John arrives at 10:10 with Flight AZ1777 in Palermo.
  - John gets on a bus at the airport.
  - The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
  - John gets off the bus in Trapani.

• A human observer can (easily) infer:
  - John is in Trapani at 13:30 but not anymore on the bus.
An Event-based Solution

• Thus
  
  the event of John getting on the bus at a given point in time initiates that John is on the bus
  
  and
  
  the event of John getting off the bus at a given point in time terminates that John is on the bus.
Architecture

- JSON
- PROLOG Server
- Language Processor
- Model Builder
Extended Architecture

- Language Processor
- Model Builder
- Event Calculus
Extended Architecture

- JSON
- PROLOG Server
- Language Processor
- Model Builder
- Event Calculus
The Event Calculus (EC)

• A framework for reasoning about events and time.
• Many versions exist (Kowalski, Shanahan, Mueller, ...).
• EC has been used for database updates, planning, explanation, hypothetical reasoning.
• The basic entities are: events, fluents and time points.
• Events which occur at a given time point initiate fluents (= properties, states) that hold until they are terminated by other events at a later time point.
Event Calculus (Simplified Version)

- Only two domain independent clauses are necessary:

  holds_at(F,T2) :-
  happens(E,T1),
  before(T1,T2),
  initiates(E,F),
  \+ clipped(T1,F,T2).

  clipped(T1,F,T3) :-
  happens(E,T2),
  terminates(E,F),
  before(T1,T2),
  before(T2,T3).
Event Calculus (Simplified Version)

- Domain specific clauses:

\[
\text{ initiates}(E, \text{located_at}(X,Y)) :\neg \text{event}(E, \text{arriving}(X,Y)).
\]

\[
\text{ terminates}(E, \text{located_at}(X,Y)) :\neg \text{event}(E, \text{leaving}(X,Y)).
\]
Event Calculus (Simplified Version)

- A particular course of events is represented as a set of `happens/2` and `event/2` clauses:

  `happens(e1,'10:10').`
  `event(e1,arriving(sk1,sk2)).`
  `happens(e2,'11:30').`
  `event(e2,leaving(sk3,sk4)).`
  `before('10:10','11:30').`

- The `before/2` clause keeps track of the temporal ordering.
Speaking about Events and Effects

• The initial scenario:
  ▪ John arrives at 10:10 with Flight AZ1777 in Palermo.
  ▪ John gets on a bus at the airport.
  ▪ The bus leaves the airport at 11:30 and arrives at the port of Trapani at 13:05.
  ▪ John gets off the bus in Trapani.

needs to be augmented with domain specific axioms.
Domain-Specific Axioms

- This can be done directly via an extension of PENG Light, for example:
  - If X arrives at Y then this event initiates that X is located at Y.
  - If X gets on Y then this event initiates that X is located in Y.
  - If X is located in Y and Y leaves Z then this event terminates that X is located at Z.
Domain-Specific Axioms

- We can further restrict the domain and the range:
  - If $X$ gets on $Y$ then this event initiates that $X$ is located in $Y$.
  - If a person gets on a vehicle then this event initiates that the person is located in the vehicle.

- Additionally, we need to specify that:
  - John is a person.
  - Every bus is a vehicle.
Model Generation

• PENG Light texts are translated into the input language of a Satchmo-style model builder.

• For example:
  ▪ John arrives with Flight AZ1777 at the airport of Palermo at 10:10.
  ▪ John gets on a bus at the airport.

results in in the following model ...
named(sk1,john). theta(e1,theme,sk1).
event(e1,arriving).
theta(e1,instrument,sk2). named(sk2,az1777).
theta(e1,location,sk3).
object(sk3,airport). associated_with(sk3,sk4).
named(sk4,palermo).
theta(e1,time,sk5). timex(sk5,'10:10').
theta(e2,agent,sk1).
event(e2,getting_on). theta(e2,theme,sk6).
object(sk6,bus).
theta(e2,location,sk3). theta(e2,time,sk7).
timex(sk7,'11:15').
before('10:10','11:15').
Two things to note:
- terms are wrapped by the predicate \texttt{fact/1}:
  \[
  \texttt{fact}(\texttt{named}(\texttt{sk1}, \texttt{john})).
  \]
- Timestamp for events without temporal modifiers, e.g.:
  - John gets on a bus at the airport.
    \[
    \texttt{theta}(\texttt{e2}, \texttt{time}, \texttt{sk7}).
    \]
    \[
    \texttt{timex}(\texttt{sk7}, \texttt{'11:15'}).\]
Domain-Specific Axioms in PENG Light

- If X arrives at Y then
  this event initiates that X is located at Y.

- If X gets on Y then
  this event initiates that X is located in Y.

- If X is located in Y and Y leaves Z then
  this event terminates that X is located at Z.
Domain Specific Axioms in Prolog

\[
\text{initiates}(E,\text{fluent}(X,\text{located} \_\text{at}, Y)) :\quad \text{initiates}(E,\text{fluent}(X,\text{located} \_\text{in}, Y)) :\
\begin{align*}
event (E, \text{arriving}), & \\
\theta (E, \text{theme}, X), & \\
\theta (E, \text{location}, Y). & \\
\end{align*}
\]

\[
\text{initiates}(E,\text{fluent}(X,\text{located} \_\text{in}, Y)) :\quad \text{terminates}(E,\text{fluent}(X,\text{located} \_\text{at}, Z)) :\
\begin{align*}
event (E, \text{getting} \_\text{on}), & \\
\theta (E, \text{agent}, X), & \\
\theta (E, \text{theme}, Y). & \\
\end{align*}
\]

\[
\text{terminates}(E,\text{fluent}(X,\text{located} \_\text{at}, Z)) :\quad \text{terminates}(E,\text{fluent}(X,\text{located} \_\text{in}, Y), \text{now}) :\
\begin{align*}
event (E, \text{leaving}), & \\
\theta (E, \text{agent}, Y), & \\
\theta (E, \text{theme}, Z), & \\
\text{holds} \_\text{at}(\text{fluent}(X,\text{located} \_\text{in}, Y), \text{now}). & \\
\end{align*}
\]
Interface between the Model and EC

• The following rule:

\[
\text{happens}(E,T) \leftarrow \text{event}(E,\text{Type}), \text{theta}(E,\text{time},X), \text{timex}(X,T).
\]

sets up the interface between the facts in the model:

\[
\text{event}(e1,\text{arriving}). \text{theta}(e1,\text{time},sk5). \text{timex}(sk5, '10:10').
\]

and the domain-independent axioms of the EC:

\[
\text{holds_at}(F,T2) \leftarrow \\
\text{happens}(E,T1), \text{before}(T1,T2), \\
\text{initiates}(E,F), \text{clipped}(T1,F,T2).
\]

\[
\text{clipped}(T1,F,T3) \leftarrow \\
\text{happens}(E,T2), \text{terminates}(E,F), \\
\text{before}(T1,T2), \text{before}(T2,T3).
\]
Reasoning

• Domain-independent axioms:

\[
\text{holds}_{\text{at}}(F,T2) \; : \; - \\
(happens(E,T1), \\
before(T1,T2), \\
\text{initiates}(E,F), \\
\text{clipped(T1,F,T2)}).
\]

\[
\text{clipped}(T1,F,T3) \; : \; - \\
(happens(E,T2), \\
\text{terminates}(E,F), \\
before(T1,T2), \\
before(T2,T3)).
\]
Reasoning

\[
\text{initiates}(E, \text{fluent}(X, \text{located\_at}, Y)) :\neg
\begin{align*}
&\text{event}(E, \text{arriving}), \\
&\theta(E, \text{theme}, X), \\
&\theta(E, \text{location}, Y).
\end{align*}
\]

\[
\text{terminates}(E, \text{fluent}(X, \text{located\_at}, Z)) :\neg
\begin{align*}
&\text{event}(E, \text{leaving}), \\
&\theta(E, \text{agent}, Y), \\
&\theta(E, \text{theme}, Z), \\
&\text{holds\_at}(\text{fluent}(X, \text{located\_in}, Y), \text{now}).
\end{align*}
\]
Reasoning (Model)

named(sk1,john). theta(e1,theme,sk1).
event(e1,arriving).
theta(e1,instrument,sk2). named(sk2,az1777).
theta(e1,location,sk3).
object(sk3,airport). associated_with(sk3,sk4).
named(sk4,palermo).
theta(e1,time,sk5).
timex(sk5,'10:10').
theta(e2,agent,sk1).
event(e2,getting_on). theta(e2,theme,sk6).
object(sk6,bus).
theta(e2,location,sk3).
theta(e2,time,sk7).
timex(sk7,'11:15').
before('10:10','11:15').
Reasoning

• Given this knowledge, we can now answer the questions:
  ▪ Where is John at 11:20?
  ▪ Where is John now?

with the help of the Event Calculus and find that John is in the bus at 11:20 and still in Palermo at that time.
Abductive Reasoning

- The EC can be extended in many interesting ways.
- We can combine the EC with a meta-interpreter for abductive reasoning in order to find explanations for why-questions.
- Abductive reasoning is inference to the best explanation.
- It is reasoning backwards from the consequent to the antecedent.
- Abductive reasoning is not a valid form of reasoning, but it can suggest plausible hypotheses.
Abductive Reasoning

• Let’s assume that we want to know
  ▪ Why does John stay in Trapani?

and we know that
  ▪ If the weather is bad then Johns stays in Trapani and goes to the Albergo Maccotta

and know also that
  ▪ John stays in Trapani and goes to the Albergo Maccotta.

then we can infer via abduction that the weather must be bad.
Future Research: Full Event Calculus

- Use of E-KRHyper as model generator:

\[
\begin{align*}
\text{HoldsAt}(f,t) & \leftarrow \text{Initially}_p(f) \land \neg \text{Clipped}(0,f,t) \\
\text{HoldsAt}(f,t3) & \leftarrow \\
& \quad \text{Happens}(a,t1,t2) \land \text{Initiates}(a,f,t1) \land \\
& \quad t2 < t3 \land \neg \text{Clipped}(t1,f,t3) \\
\text{Clipped}(t1,f,t4) & \leftarrow \\
& \quad \exists a,t2,t3 \ [\text{Happens}(a,t2,t3) \land t1 < t3 \land t2 < t4 \land \\
& \quad \text{Terminates}(a,f,t2) \lor \text{Releases}(a,f,t2)] \\
\neg \text{HoldsAt}(f,t) & \leftarrow \text{Initially}_N(f) \land \neg \text{Declipped}(0,f,t) \\
\neg \text{HoldsAt}(f,t3) & \leftarrow \\
& \quad \text{Happens}(a,t1,t2) \land \text{Terminates}(a,f,t1) \land \\
& \quad t2 < t3 \land \neg \text{Declipped}(t1,f,t3) \\
\text{Declipped}(t1,f,t4) & \leftarrow \\
& \quad \exists a,t2,t3 \ [\text{Happens}(a,t2,t3) \land t1 < t3 \land t2 < t4 \land \\
& \quad \text{Initiates}(a,f,t2) \lor \text{Releases}(a,f,t2)] \\
\text{Happens}(a,t1,t2) & \rightarrow t1 \leq t2
\end{align*}
\]
Conclusions

• PENG Light can be extended in a systematic way to speak about events and their effects.
• PENG Light texts are translated automatically into the input language of a model builder.
• The generated model can be used by the Event Calculus for reasoning about events and time.
• This combinations makes PENG Light an interesting specification language for dynamic domains.