Furthering the Continuous-Change Event Calculus: Providing for Efficient Descriptions of Additive Effects and an Automated Reasoner

Ankesh Khandelwal

Tetherless World Constellation
Rensselaer Polytechnic Institute

29-MAR-2013
Outline

1 Introduction: Logic, Semantic Web, and Event Calculus

2 Contributions
   - Additive Effects in Event Calculus
   - Closed-World Aggregate Summation in FOL
   - Constructing Models for Event Calculus

3 Conclusion
1 Introduction: Logic, Semantic Web, and Event Calculus

2 Contributions
   - Additive Effects in Event Calculus
   - Closed-World Aggregate Summation in FOL
   - Constructing Models for Event Calculus

3 Conclusion
Formal Logic

1. Formal languages for the representation of knowledge with clear semantics

2. Concerned with principles of inference and reasoning
   - reasoning: operation(s) stating validity or consistency of a logical assertion

3. Declarative knowledge representation: express what is valid, the responsibility to interpret this and to decide on how to do it is delegated to an interpreter/reasoner

---

Declarative Knowledge Representation in Logic

If a service is available less than 98% time then its quality is low, otherwise high. And, for low quality services there is $-5\%$ discount and for others $5\%$.\(^2\)

<table>
<thead>
<tr>
<th>Logic Programming</th>
<th>Procedural Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>discount(Service, 5%)</code> :- <code>qos(Service,high)</code>.</td>
<td></td>
</tr>
<tr>
<td><code>discount(Service, -5%)</code> :- <code>qos(Service,low)</code>.</td>
<td></td>
</tr>
<tr>
<td><code>qos(Service,high)</code> :- <code>availability(Service) = 1</code>.</td>
<td></td>
</tr>
<tr>
<td><code>qos(Service,low)</code> :- <code>availability(Service) &lt; 0.98</code>.</td>
<td></td>
</tr>
</tbody>
</table>

**Queries**
- `discount(Service,X)`? All discounts for all service
- `discount(s1,X)`? Discount for “s1”
- `discount(s1,5%)`? Service “s1” → discount 5%?
- `qos(Service,5%)`? All serv. with discount 5%
- `qos(Service,Y)`? All service levels for all services?
- `qos(s1,Y)`? Service level for “s1”?...

**Procedural Programming**

```
boolean getsDiscount(Service s, int value) {
    if (getAvailability(s)==1) & (value==1) return true;
    else if (getAvailability(s)<0.98) & (value<0.98) return true;
    else return false;
}
...
Service getService(int value) {
    for (int i=0; i<getAllServices(); i++) {
        Service s = getService(i);
        if (getAvailability(s)==1) & (value==1) return s;
        else if (getAvailability(s)<0.98) & (value<0.98) return s;
        else return null;
    }
}
int getDiscount(Service s) {
    if (getAvailability(s)==1) return 5;
    else if (getAvailability(s)<0.98) return -5;
    else return 0;
}
... 29
```

\(^2\) Adrian Paschke. *Rules and Logic Programming for the Web*. Lecture in
Semantic Web: Web of Machine-Readable Data

http://lod-cloud.net/

Khandelwal

Defense Talk

6 / 40
Name processes that explain atypical temperature change observed.

http://lod-cloud.net/
Semantic Web Stack

http://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb/
Continuous-Changes in Artificial Intelligence and Knowledge Representation

1. Sowa’s ontology [Sow02] and Upper ontologies [MCR06]
   - No quantitative description

2. PhysSys [BAT97], OntoCAPE [MYM07] eng. ontologies
   - Quantitative descriptions are a black box

3. Hybrid Automaton [Hen96], Halo [GP+10], Cyc [AC02]
   - Non-logic-based

4. Compositional Modeling Language [Fal+94]
   - Qualitative Reasoning

5. Modal temporal logics: computational tree logic [AF01]
   - Only discrete change

6. Fluent [Thi01], Situation [Rei96], and Event [MS96] Calculi
   - Changes as functions of time, except in Event Calculus
Event Calculus is a FOL-based language with many features:

- Ramifications
- Domain constraints
- Triggered actions and concurrent actions
- Actions with non-deterministic effects, etc.

Applications [Mue06]

- Business systems: E.g., payment and online purchase protocols (commitments etc.)
- Workflow modeling
- Natural language understanding: E.g., understanding story and answering questions
- Vision: E.g., recognizing shapes from extracted edges
Contributions

1. Extended Event Calculus for general, concise, and elaboration tolerant descriptions of additive effects using aggregate formulas in FOL.

2. Introduced a novel method for closed-world reasoning for aggregate formulas in FOL.

3. Designed separation of logic and *equations* reasoning for constructing models of Event Calculus descriptions given an initial state and narratives of external actions.
Introduction: Logic, Semantic Web, and Event Calculus

Contributions
- Additive Effects in Event Calculus
- Closed-World Aggregate Summation in FOL
- Constructing Models for Event Calculus

Conclusion
Outline

1. Introduction: Logic, Semantic Web, and Event Calculus

2. Contributions
   - Additive Effects in Event Calculus
   - Closed-World Aggregate Summation in FOL
   - Constructing Models for Event Calculus

3. Conclusion
Describe the result of opening outlets of tanks B and C such that tank A overflows when filled to full capacity.
Layered Tanks Example

State 0

- Tank C
- Tank B
- Tank A
- B above A
- C above A
- Capacity of A = 20
- Capacity of B = 15
- Capacity of C = 15
- Cover of A is open
- Outlet of B is close
- Outlet of C is close
- Amount in A = 0
- Amount in B = 15
- Amount in C = 15
Layered Tanks Example

**State 0**
- Tank C
- Tank B
- Tank A
- B above A
- C above A
- Capacity of A = 20
- Capacity of B = 15
- Capacity of C = 15
- Cover of A is open
- Outlet of B is close
- Outlet of C is close
- Amount in A = 0
- Amount in B = 15
- Amount in C = 15

**State 1**
- ... Cover of A is open
- Outlet of B is open
- Outlet of C is close
- Amount in A initially = 0
- Amount in B initially = 15
- Amount in C = 15

**External action:** Open outlet of B

**Rate of change in B**
\[-K \times \text{amount}(B)\]

**Rate of change in A**
\[+K \times \text{amount}(B)\]
Layered Tanks Example

State 2

... Cover of A is open
Outlet of B is open
Outlet of C is open
Amount in A initially = ...
Amount in B initially = ...
Amount in C initially = 15

Rate of change in B
= -K × amount(B)
Rate of change in C
= -K × amount(C)
Rate of change in A
= K × amount(B) + K × amount(C)

Triggered action: Overflow A

State 3

... Cover of A is open
Outlet of B is open
Outlet of C is open
Amount in A initially = 20
Amount in B initially = ...
Amount in C initially = ...

Rate of change in B
= -K × amount(B)
Rate of change in C
= -K × amount(C)
Rate of change in A = 0

External action: Open outlet of C
### Representation in Event Calculus

<table>
<thead>
<tr>
<th>Type</th>
<th>Type Description</th>
<th>Example</th>
<th>Example Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluent</td>
<td>Boolean-valued temporal property</td>
<td>Open(A)</td>
<td>A is open at top</td>
</tr>
<tr>
<td>Quantity</td>
<td>real-valued temporal property</td>
<td>Amount(A)</td>
<td>amount in A</td>
</tr>
<tr>
<td>Derivative</td>
<td>Derivative of another quantity</td>
<td>$\delta(\text{Amount}(A))$</td>
<td>the rate of change in A</td>
</tr>
<tr>
<td>State of Fluent</td>
<td>the value of fluent F</td>
<td>HoldsAt(F, T)</td>
<td>F holds at T</td>
</tr>
<tr>
<td>State of Quantity</td>
<td>the value of quantity Q</td>
<td>Value(Q, T)</td>
<td>the value of Q at T</td>
</tr>
<tr>
<td>New Relations</td>
<td>Non-Event Calculus predicate</td>
<td>Above(B, A)</td>
<td>B is above A</td>
</tr>
</tbody>
</table>

If outlet $b$ is open then the rate of change in $b = -K \times \text{Amount}(b)$:

$$\text{HoldsAt}(\text{OpenOut}(b), t) \rightarrow \text{Value}(\delta(\text{Amount}(b)), t) = -K \times \text{Value}(\text{Amount}(b), t)$$
## Representation in Event Calculus

<table>
<thead>
<tr>
<th>Type</th>
<th>Type Description</th>
<th>Example</th>
<th>Example Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluent</td>
<td>Boolean-valued temporal property</td>
<td>Open(A)</td>
<td>A is open at top</td>
</tr>
<tr>
<td>Quantity</td>
<td>real-valued temporal property</td>
<td>Amount(A)</td>
<td>amount in A</td>
</tr>
<tr>
<td>Derivative</td>
<td>Derivative of another quantity</td>
<td>$\delta$(Amount(A))</td>
<td>the rate of change in A</td>
</tr>
<tr>
<td>State of Fluent</td>
<td>the value of fluent F</td>
<td>HoldsAt(F, T)</td>
<td>F holds at T</td>
</tr>
<tr>
<td>State of Quantity</td>
<td>the value of quantity Q</td>
<td>Value(Q, T)</td>
<td>the value of Q at T</td>
</tr>
<tr>
<td>New Relations</td>
<td>Non-Event Calculus predicate</td>
<td>Above(B, A)</td>
<td>B is above A</td>
</tr>
</tbody>
</table>

If outlet b is open then the rate of change in b= $-K \times Amount(b)$:

$$HoldsAt(\text{OpenOut}(b), t) \rightarrow Value(\delta(Amount(b)), t) = -K \times Value(Amount(b), t)$$
If outlets of b and c are open then the rate of change in $a = K \times \text{Amount}(b) + K \times \text{Amount}(c)$:

$$[\text{Above}(b, a) \land \text{Above}(c, a) \land b \neq c \\
\land \text{HoldsAt}(\text{OpenOut}(b), t) \land \text{HoldsAt}(\text{OpenOut}(c), t)] \\
\rightarrow \text{Value}(\delta(\text{Amount}(a)), t) = K \times \text{Value}(\text{Amount}(b), t) + K \times \text{Value}(\text{Amount}(c), t)$$

If outlets of b and c and d are open then the rate of change in $a = K \times \text{Amount}(b) + K \times \text{Amount}(c) + K \times \text{Amount}(d)$:

$$[\text{Above}(b, a) \land \text{Above}(c, a) \land \text{Above}(d, a) \land b \neq c \neq d \\
\land \text{HoldsAt}(\text{OpenOut}(b), t) \land \text{HoldsAt}(\text{OpenOut}(c), t) \land \text{HoldsAt}(\text{OpenOut}(d), t)] \\
\rightarrow \text{Value}(\delta(\text{Amount}(a)), t) = K \times \text{Value}(\text{Amount}(b), t) + K \times \text{Value}(\text{Amount}(c), t) \\
+ K \times \text{Value}(\text{Amount}(d), t)$$
For Concise Descriptions of Additive Effects

- If outlets of b and c open then the rate of change in a = $K \times \text{Amount}(b)$
  $+$ $K \times \text{Amount}(c)$:

  $[\text{Above}(b, a) \land \text{Above}(c, a) \land b \neq c$
  $\land \text{HoldsAt}(\text{OpenOut}(b), t) \land \text{HoldsAt}(\text{OpenOut}(c), t)]$
  $\rightarrow \text{Value}(\delta(\text{Amount}(a)), t) = K \times \text{Value}(\text{Amount}(b), t) + K \times \text{Value}(\text{Amount}(c), t)$

- If outlet of b is open then it contributes additively to the rate of change in a with value $K \times \text{Amount}(b)$:

  $[\text{Above}(b, a) \land \text{HoldsAt}(\text{OpenOut}(b), t)]$
  $\rightarrow \text{PartValue}(\delta(\text{Amount}(a)), t, b, K \times \text{Value}(\text{Amount}(b), t))$

- When outlets of B and C are open we have:

  $\text{PartValue}(\delta(\text{Amount}(A)), t, B, K \times \text{Value}(\text{Amount}(B), t))$
  $\text{PartValue}(\delta(\text{Amount}(A)), t, C, K \times \text{Value}(\text{Amount}(C), t))$
For Concise Descriptions of Additive Effects

- If outlets of b and c open then the rate of change in a = $K \times \text{Amount}(b) + K \times \text{Amount}(c)$:

$$[\text{Above}(b, a) \land \text{Above}(c, a) \land b \neq c \land \text{HoldsAt}(\text{OpenOut}(b), t) \land \text{HoldsAt}(\text{OpenOut}(c), t)]$$

$$\rightarrow \text{Value}(\delta(\text{Amount}(a)), t) = K \times \text{Value}(\text{Amount}(b), t) + K \times \text{Value}(\text{Amount}(c), t)$$

- If outlet of b is open then it contributes additively to the rate of change in a with value $K \times \text{Amount}(b)$:

$$[\text{Above}(b, a) \land \text{HoldsAt}(\text{OpenOut}(b), t)]$$

$$\rightarrow \text{PartValue}(\delta(\text{Amount}(a)), t, b, K \times \text{Value}(\text{Amount}(b), t))$$

- When outlets of B and C are open we have:

$$\text{PartValue}(\delta(\text{Amount}(A)), t, B, K \times \text{Value}(\text{Amount}(B), t))$$

$$\text{PartValue}(\delta(\text{Amount}(A)), t, C, K \times \text{Value}(\text{Amount}(C), t))$$
Aggregates in First-order Logic

- Aggregate expression in FOL:

\[ \#\text{sum}\langle r, px.\text{PartValue}(p, t, px, r)\rangle \]

- Value of an *additive* quantity is given by summation of all relevant concurrently active values/effects/changes:

\[ \exists r, px.\text{PartValue}(p, t, px, r) \Rightarrow \text{Value}(p, t) = \#\text{sum}\langle r, px.\text{PartValue}(p, t, px, r)\rangle \]

- But: we have replaced a *close* summation of \( K \times \text{Value}(\text{Amount}(B), t) \) and \( K \times \text{Value}(\text{Amount}(C), t) \) by an *open* (aggregate) summation in a *monotonic* logic with *open world assumption*
Aggregates in First-order Logic

- Aggregate expression in FOL:

\[
\#	ext{sum}\langle r, px.\text{PartValue}(p, t, px, r)\rangle
\]

- Value of an additive quantity is given by summation of all relevant concurrently active values/effects/changes:

\[
\exists r, px.\text{PartValue}(p, t, px, r) \implies \text{Value}(p, t) = \#	ext{sum}\langle r, px.\text{PartValue}(p, t, px, r)\rangle
\]

- But: we have replaced a close summation of \( K \times \text{Value}(\text{Amount}(B), t) \) and \( K \times \text{Value}(\text{Amount}(C), t) \) by an open (aggregate) summation in a monotonic logic with open world assumption
Likewise for Discrete-Change

- Action A discretely changes the value of quantity Q to R:
  \[ \text{BreaksTo}(A, Q, T, R) \]

- Action A changes the value of quantity Q partially by R:
  \[ \text{BreaksPartBy}(A, Q, T, R) \]

- E.g., Scoop reduces the Level by R:
  \[ \text{BreaksPartBy}(\text{Scoop}, \text{Level}, T, R), \text{such that} \]
  \[ \text{change} = \# \sum \langle a, r.\text{BreaksPartBy}(a, \text{Level}, t, r) \land \text{Happens}(a, t) \rangle \rightarrow \text{RightLimit}(\text{Level}, t, \text{Value}(\text{Level}, t) + \text{change}) \]
On Additive Effects since Proposal

- Showed that iterative/recursive summation of additive effects requires artificial ordering additive effects.

- Worked with two other alternatives to PartValue relation to avoid need for novel nonmonotonic reasoning technique, but both alternatives also ridden with similar issues.

- Introduced variants of BreaksPartBy relation to model range restrictions and absolute change.

- Examined ramifications of discrete additive effects, which require the novel nonmonotonic reasoning technique.
Outline

1. Introduction: Logic, Semantic Web, and Event Calculus

2. Contributions
   - Additive Effects in Event Calculus
   - Closed-World Aggregate Summation in FOL
   - Constructing Models for Event Calculus

3. Conclusion
Closed-World Reasoning in FOL with Aggregates

- If sum of all $x$'s s.t. $P(x)$ is true is zero, infer $Q(1)$

$$\#\text{sum} \langle x.P(x) \rangle = 0 \rightarrow Q(1)$$

- Expected model: \{\textit{Q(1)}\}
- Many other unexpected models: \{\textit{P(2)}\}, \{\textit{P(0), Q(1)}\}, etc.
- Implication gives necessary conditions, sufficiency conditions are unknown
- Often, sufficiency conditions are encoded via minimal models
- Unexpected minimal models: \{\textit{P(2)}\}, etc.
- We classify such unexpected models as weak models
- We defined a novel nonmonotonic reasoning technique, $CIRC^A$-transformation to eliminate the weak models
Closed-World Reasoning in FOL with Aggregates

- If sum of all $x$'s s.t. $P(x)$ is true is zero, infer $Q(1)$

$$\#\text{sum} \langle x.P(x) \rangle = 0 \rightarrow Q(1)$$

- Expected model: $\{Q(1)\}$
- Many other unexpected models: $\{P(2)\}$, $\{P(0), Q(1)\}$, etc.
- Implication gives necessary conditions, sufficiency conditions are unknown
- Often, sufficiency conditions are encoded via minimal models
- Unexpected minimal models: $\{P(2)\}$, etc.
- We classify such unexpected models as weak models
- We defined a novel nonmonotonic reasoning technique, $CIRC^A$-transformation to eliminate the weak models
Closed-World Reasoning in FOL with Aggregates

- If sum of all $x$'s s.t. $P(x)$ is true is zero, infer $Q(1)$

\[
\#\text{sum}\langle x.P(x) \rangle = 0 \longrightarrow Q(1)
\]

- Expected model: \{\text{\}$Q(1)$\}
- Many other unexpected models: \{\text{\}$P(2)$\}, \{\text{\}$P(0), Q(1)$\}, etc.
- Implication gives necessary conditions, sufficiency conditions are unknown
- Often, sufficiency conditions are encoded via minimal models
- Unexpected minimal models: \{\text{\}$P(2)$\}, etc.

We classify such unexpected models as \text{weak} models

We defined a novel nonmonotonic reasoning technique, \text{CIRC}^A -transformation to eliminate the weak models
Closed-World Reasoning in FOL with Aggregates

- If sum of all $x$'s s.t. $P(x)$ is true is zero, infer $Q(1)$

$$\#\text{sum}\langle x.P(x) \rangle = 0 \rightarrow Q(1)$$

- Expected model: \{Q(1)\}
- Many other unexpected models: \{P(2)\}, \{P(0), Q(1)\}, etc.
- Implication gives necessary conditions, sufficiency conditions are unknown
- Often, sufficiency conditions are encoded via minimal models
- Unexpected minimal models: \{P(2)\}, etc.
- We classify such unexpected models as weak models
- We defined a novel nonmonotonic reasoning technique, $CIRC^A$-transformation to eliminate the weak models
Showed that the CIRC$^A$-transformation semantics coincides with first-order stable model semantics for *ag-canonical* class of FOL formulas with aggregates.

Used that result to deploy answer-set solvers for logic reasoning for Event Calculus descriptions.
Outline

1. Introduction: Logic, Semantic Web, and Event Calculus

2. Contributions
   - Additive Effects in Event Calculus
   - Closed-World Aggregate Summation in FOL
   - Constructing Models for Event Calculus

3. Conclusion
Automated theorem proving (for FOL): not successful, often requires human assistance [Mue05]

Logic Programming (e.g., Prolog): for discrete-change and continuous-change, described as function of time, Event Calculus, but e.g. cannot handle non-determinism [KS86; Sha00]

Answer-set programming (ASP): for discrete-time and discretized continuous-change Event Calculus [KLP09]
Separation of Logic and Equations Reasoning

- Landmarks: time points at which there is a state transition (i.e. discontinuous change), caused by actions
- Logic reasoning: only at landmarks to infer equations for determination of trajectories and next landmark
- Equations reasoning: to infer next landmark and values at the next landmark
If b is above a and outlet of b is open and rate of change in b is not zero then b is filling a:

$$[Above(b, a) \land HoldsAt(OpenOut(b), t) \land Value(\delta(Amount(b)), t) < 0]$$

$$\iff HoldsAt(Filling(b, a), t)$$
Derivation of New Axioms for Next Landmark

- If a is filling and the amount in it is equal to its capacity then overflow action is triggered:

  \[
  \exists b. \text{HoldsAt}(\text{Filling}(b, a), t) \land \text{Value}(\text{Amount}(a), t) = \text{Value}(\text{Capacity}(a), t) \\
  \rightarrow \text{Happens}(\text{Overflow}(a), t)
  \]

We separate logic and mathematical conditions in the antecedents and construct new axioms

- If b is above a and outlet of b is open then check condition C1(b):

  \[
  \text{Above}(b, a) \land \text{HoldsAt}(\text{OpenOut}(b), t) \\
  \rightarrow \text{CheckCnd}(C1(b), t),
  \]

  where \( C1(b) : \text{Value}(\delta(\text{Amount}(b)), t) < 0 \)

- And, if a is filling then check conditions C2(a):

  \[
  \exists b. \text{HoldsAt}(\text{Filling}(b, a), t) \\
  \rightarrow \text{CheckCnd}(C2(a), t),
  \]

  where \( C2(a) : \text{Value}(\text{Amount}(a), t) = \text{Value}(\text{Capacity}(a), t) \)
Other New Axioms that are Derived

Other axioms are also syntactically derived to:

- Deduce equations for trajectories
- Deduce values (right limit values) of quantities after discontinuous changes
- Detect incomplete specification of a domain, e.g., unknown value after discontinuous change
- Detect unwarranted discontinuous change

We can now assert the default assumption that “quantities are constant by default if continuous”, in addition to the default assumption “quantities are continuous by default”
Other New Axioms that are Derived

Other axioms are also syntactically derived to:

- Deduce equations for trajectories
- Deduce values (right limit values) of quantities after discontinuous changes
- Detect incomplete specification of a domain, e.g., unknown value after discontinuous change
- Detect unwarranted discontinuous change

We can now assert the default assumption that “quantities are constant by default if continuous”, in addition to the default assumption “quantities are continuous by default”
Implementing Logic Reasoning

Different options:
- Automated theorem-provers like SNARK\(^3\)
- Logic programming deductive systems like Prolog\(^4\)
- Answer-set solvers like DLV\(^5\)

\(^3\)http://www.ai.sri.com/~stickel/snark.html
\(^4\)E.g., XSB Prolog: http://xsb.sourceforge.net/
\(^5\)http://www.dlvsystem.com/
Kim, Lee and Palla (2009) showed that Event Calculus can be reformulated as answer-set programs (ASP).

We show that the extended Event Calculus can also be reformulated as ASP.

HEX-program [Eit+05] extends ASP with Higher-Order External atoms, which are required for real number arithmetic and comparisons.

DLVHEX\(^6\) is a prototypical implementation for HEX-programs based in DLV.

[^6]: http://www.kr.tuwien.ac.at/research/systems/dlvhex/
The Model Construction Process

**Input:** FOL Domain Description, initial state, narratives of external actions, max_time

**Output:** Models upto max_time

1. Derive new axioms from the given description;
2. Convert FOL description to ASP in DLV syntax using F2LP tool\(^a\);
3. Convert ASP to HEX-programs, and introduce external atoms;
4. Find initial model(s) using DLVHEX;
5. landmark:= 0;
6. **while** landmark \(\neq\) max_time **do**
   7. Determine discontinuous change at landmark using DLVHEX;
   8. Determine initial values and equations for trajectories and next landmark using DLVHEX;
   9. Compute new landmark and values of quantities at the new landmark using Mathematica;
7. **end**

\(^a\)http://reasoning.eas.asu.edu/f2lp/
Recall: Layered Tanks Example and State 0

State 0

- Tank C
- Tank B
- Tank A
- B above A
- C above A
- Capacity of A = 20
- Capacity of B = 15
- Capacity of C = 15
- Cover of A is open
- Outlet of B is close
- Outlet of C is close
- Amount in A = 0
- Amount in B = 15
- Amount in C = 15
%Initial state

Above(B, A) \land Above(C, A) \quad (1)

InitializedTrue(Open(A)) \quad (2)

InitializedFalse(OpenOut(B)) \land InitializedFalse(OpenOut(C)) \quad (3)

InitialValue(Capacity(A), 20) \land InitialValue(Amount(A), 0) \quad (4)

InitialValue(Capacity(B), 15) \land InitialValue(Amount(B), 15) \quad (5)

InitialValue(Capacity(C), 15) \land InitialValue(Amount(C), 15) \quad (6)

InitialValue(\delta(Amount(A)), 0) \quad (7)

\land InitialValue(\delta(Amount(B)), 0) \land InitialValue(\delta(Amount(C)), 0) \quad (8)

%Narratives of external action occurrences

Happens(UncoverOut(B), 1) \quad (8)

Happens(UncoverOut(C), 2) \quad (9)

%Effects on fluents

Initiates(UncoverOut(b), OpenOut(b), t) \quad (10)
%Non-additive continuous change

\[ \text{HoldsAt}(\text{OpenOut}(b), t) \]  
\[ \longrightarrow \text{Value}(\delta(\text{Amount}(b), t) = -K \times \text{Value}(\text{Amount}(b), t) \]  

%Additive continuous change

\[ [\text{Above}(b, a) \land \text{HoldsAt}(\text{Open}(a), t) \land \text{HoldsAt}(\text{OpenOut}(b), t)] \]  
\[ \land \text{Value}(\text{Amount}(a), t) < \text{Value}(\text{Capacity}(a), t)] \]  
\[ \longrightarrow \text{PartValue}(\delta(\text{Amount}(a)), t, b, K \times \text{Value}(\text{Amount}(b), t)) \]  

%Triggered action

\[ [\text{Above}(b, a) \land \text{HoldsAt}(\text{Open}(a), t) \land \text{HoldsAt}(\text{OpenOut}(b), t)] \]  
\[ \land \text{Value}(\text{Amount}(b), t) > 0 \land \text{Value}(\text{Amount}(a), t) = \text{Value}(\text{Capacity}(a), t)] \]  
\[ \longrightarrow \text{Happens}(\text{Overflow}(a), t) \]  

%Discontinuities caused by actions

\[ \text{Above}(b, a) \longrightarrow \text{Breaks}(\text{UncoverOut}(b), \delta(\text{Amount}(a)), t) \]  
\[ \text{Breaks}(\text{UncoverOut}(b), \delta(\text{Amount}(b)), t) \]  
\[ \text{BreaksTo}(\text{Overflow}(a), \delta(\text{Amount}(a)), t, 0) \]
## Simulation Result

<table>
<thead>
<tr>
<th>Time</th>
<th>B open</th>
<th>C open</th>
<th>Amt B</th>
<th>Amt C</th>
<th>Amt A</th>
<th>$\delta(Amt A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>False</td>
<td>False</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>False</td>
<td>False</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>True</td>
<td>False</td>
<td>2.03</td>
<td>15</td>
<td>12.97</td>
<td>34.06</td>
</tr>
<tr>
<td>2.67</td>
<td>True</td>
<td>True</td>
<td>1.19</td>
<td>8.81</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>True</td>
<td>True</td>
<td>0.04</td>
<td>0.27</td>
<td>20</td>
<td>NA</td>
</tr>
</tbody>
</table>

*K=2*
Introduction: Logic, Semantic Web, and Event Calculus

Contributions
- Additive Effects in Event Calculus
- Closed-World Aggregate Summation in FOL
- Constructing Models for Event Calculus

Conclusion
Summary

1. Extended Event Calculus for general, concise, and elaboration tolerant descriptions of additive effects using aggregate formulas in FOL
   - Additive effects are common in concurrent systems
   - Allows for greater reuse and modular development

2. Introduced a novel method for closed-world reasoning for aggregate formulas in FOL
   - Used it for solving the *frame problem* in extended Event Calculus
   - *Weak* minimal models are general, and the method is widely applicable
Summary

Designed separation of logic and equations reasoning for constructing models of Event Calculus descriptions given an initial state and narratives of external actions

- Implemented a prototype model builder using DLVHEX, through translation into answer-set programs
- Other logic reasoners such as for logic programs can be used instead
- Naturally suited for temporal projection and can be modified for postdiction, etc.

Above enhancements would encourage use of logic formalisms/systems for descriptions of dynamical systems with quantitative descriptions of continuous-changes.
Summary

- Designed separation of logic and equations reasoning for constructing models of Event Calculus descriptions given an initial state and narratives of external actions

  - Implemented a prototype model builder using DLVHEX, through translation into answer-set programs

  - Other logic reasoners such as for logic programs can be used instead

  - Naturally suited for temporal projection and can be modified for postdiction, etc.

- Above enhancements would encourage use of logic formalisms/systems for descriptions of dynamical systems with quantitative descriptions of continuous-changes
Future Works

Implementation:

- Improve the prototypical DLVHEX reasoner. E.g., we could not reason about multiple blocks problem using DLVHEX.

- Extend answer-set solvers to support programs with aggregates with recursive (but stratified) dependencies and external computations.

- Alternatively, try DLV-Complex.

- Implement support for cosine and other functions.

- Deploy a free (and open source) equations reasoner.
Future Works

Extended Event Calculus:

- Design similar extensions to Event Calculus for other FOL-based formalisms like Fluent and Situation Calculi
- Include durative actions and actions with delayed effects in the extended Event Calculus
- Design abductive reasoning algorithms for continuous-change, and the extended, Event Calculus

Process modeling language:

- Realize a process modeling language for the Semantic Web, with tools to support authoring, parsing, etc.
Thanks!

This day would not have been possible if not for:

- Parents: mother for her strengths, father for his “I am with you whatever you decide”
- Father’s elder brother and his wife
- Roommates: Gaurav, Sai, Vaibhav
- Naveen Sundar
- Jesse Weaver
- Lalana Kagal, Fuming Shih
- Rob Miller
- Academics who made this line of research fun: John McCarthy, Ray Reiter, Vladimir Lifschitz, Murray Shanahan
- Claudio Gutierrez, Ravi Palla, Joohyung Lee, Thomas Krennwallner, Peter Schüller, Erik Mueller
- The anonymous reviewer of JAIR
Thank You!
If \( b_1 \) is on \( b_2 \), their relative velocity is zero and \( b_1 \) has a tendency to slide forward relative to \( b_2 \) in response to the applied forces then \( b_1 \) has a net acceleration in the forward direction, the friction is kinetic between them which pushes \( b_1 \) backwards (and \( b_1 \) does not slide backwards):

\[
\begin{align*}
HoldsAt(SR(b_1, b_2), t) & \lor \neg HoldsAt(SR(b_1, b_2), t) \\
HoldsAt(SL(b_1, b_2), t) & \lor \neg HoldsAt(SL(b_1, b_2), t) \\
[ Holds(On(b_1, b_2), t) \land Val(\delta(Pos(b_1)), t) = Val(\delta(Pos(b_2)), t) ] & \\
& \land Holds(SR(b_1, b_2), t)] \\
\rightarrow [ Val(Fr(b_1, b_2), t) = -1 \times Val(CKF(b_1, b_2), t) \times Val(FbBA(b_2, b_1), t) ] \\
& \land Val(\delta(\delta(Pos(b_1))), t) - Val(\delta(\delta(Pos(b_2))), t) > 0 \land \neg Holds(SL(b_1, b_2), t) 
\end{align*}
\]


[Eit+05] Thomas Eiter et al. “A Uniform Integration of Higher-order Reasoning and External Evaluations in Answer-set Programming”. In: *Proceedings of the*

[Fal+94] Brian Falkenhainer et al. 


References


<table>
<thead>
<tr>
<th>Reference</th>
<th>Author(s)</th>
<th>Title</th>
<th>Notes</th>
</tr>
</thead>
</table>
10.1023/A:1010558931274. URL: http://dx.doi.org/10.1023/A%3A1010558931274.