Automated Reasoning for Situational Awareness

Peter Baumgartner, Alexander Krumpholz
Supply Chain Integrity Digital Mission

www.data61.csiro.au
Situational Awareness - Systems of Interest

- **Factory Floor**
  - Are the operations carried out according to the schedule?

- **Food Supply Chain**
  - Are goods delivered within 3 hours and stored below 25°C?
  - Why is the truck late?
  - Where did the strawberries/honey come from?

- **Data Cleansing**
  - Does the database have complete, correct, accurate and relevant data?
Situational Awareness - Systems of Interest

• **Factory Floor**
  - Are the operations carried out according to the schedule?

• **Food Supply Chain**
  - Are goods delivered within 3 hours and stored below 25°C?
  - Why is the truck late?
  - Where did the strawberries/honey come from?

• **Data Cleansing**
  - Does the database have complete, correct, accurate and relevant data?

**Build a system that automatically derives such analysis?**
Situational Awareness - Systems of Interest

- **Factory Floor**
  - Are the operations carried out according to the schedule?

- **Food Supply Chain**
  - Are goods delivered within 3 hours and stored below 25°C?
  - Why is the truck late?
  - Where did the strawberries/honey come from?

- **Data Cleansing**
  - Does the database have complete, correct, accurate and relevant data?

**Build a system that automatically derives such analysis?**
Automated Reasoning for Situational Awareness

Events...
- GPS coordinates
- Temp sensor
- Paperwork
- Log DB

Why is the truck late?
Are the tomatoes still fresh?
Automated Reasoning for Situational Awareness

**Why is the truck late?**

**Are the tomatoes still fresh?**

**Events...**
- GPS coordinates
- Temp sensor
- Paperwork
- Log DB

**Why this is hard**
- Incomplete/noisy/erroneous data
- Need domain knowledge ("fresh"?)

[Diagram: Food Production Chain]
Automated Reasoning for Situational Awareness

Why is the truck late?
Are the tomatoes still fresh?

**Why this is hard**
- Incomplete/noisy/erroneous data
- Need domain knowledge (“fresh”?)

**Events...**
- GPS coordinates
- Temp sensor
- Paperwork
- Log DB
Automated Reasoning for Situational Awareness

Why this is hard
- Incomplete/noisy/erroneous data
- Need domain knowledge ("fresh"?)

Our *logic*-based approach
- Domain *modelling* (first principles)
- What-if *reasoning* and *explanations*

*Events...*
- GPS coordinates
- Temp sensor
- Paperwork
- Log DB

*Why is the truck late?*
*Are the tomatoes still fresh?*

Stuck at warehouse / fresh
*OR*
Traffic jam / not fresh
Automated Reasoning for Situational Awareness

Why this is hard
- Incomplete/noisy/erroneous data
- Need domain knowledge ("fresh"?)

Our logic-based approach
- Domain modelling (first principles)
- What-if reasoning and explanations

Implemented in the Fusemate system
Observation: truck is in Sydney at the warehouse
Observation: truck is in Sydney at the warehouse
Observation: tomatoes are loaded
Observation: tomatoes are loaded
Assumption as per schedule: truck is on the road
Assumption as per schedule: truck is on the road
Report: truck is on the road
Report: *truck is on the road*
Conclusion: truck is on the road for too long - tomatoes are no longer fresh
Conclusion: truck is on the road for too long - tomatoes are no longer fresh
Report: actually, at T+1 truck was still in Sydney warehouse
Report: actually, at T+1 truck was still in Sydney warehouse
Conclusion: tomatoes are still fresh at T+2
Conclusion: tomatoes are still fresh at $T+2$
No information at T+3
T+3: What if truck is on the road?
Demo

$T+3$: *What if truck is on the road?*
**T+3:** What if truck is on the road? At Canberra warehouse?
Report: truck at Canberra warehouse
Report: truck at Canberra warehouse
Fusemate in more detail...
Fusemate Model = If-Then Rules

T1

T2

Time
Fusemate Model = If-Then Rules

\[ T_1 < T \leq T_2 \]

Time

NO!
Fusemate Model = If-Then Rules

T1 ≤ T2

NO!

T1 < T ≤ T2

T2

Time
Fusemate Model = If-Then Rules

If $T_1$ and $T_1 < T \leq T_2$ then NO!
Fusemate Model = If-Then Rules

If \( T_1 < T \leq T_2 \) then NO!

If in(T1, B, C) & T1 < T2 NOT exists T s.th. T1 < T \leq T2 & unload(T, B, C) then in(T2, B, C)
Fusemate Model = If-Then Rules

If $T_1$ and $T_1 < T \leq T_2$ then NO!

If in($T_1$, $B$, $C$) & $T_1 < T_2$ NOT exists $T$ s.th. $T_1 < T \leq T_2$ & unload($T$, $B$, $C$) then in($T_2$, $B$, $C$)

default reasoning: "not known" (≠ "known not")
Fusemate Model = If-Then Rules

If $\text{T_1} < T \leq \text{T_2}$ and then NO!

If $\text{in}(\text{T_1}, \text{B}, \text{C}) \& \text{T_1} < \text{T_2}$ NOT exists T s.th. $\text{T_1} < T \leq \text{T_2} \& \text{unload}(T, \text{B}, \text{C})$ then $\text{in}(\text{T_2}, \text{B}, \text{C})$

default reasoning: “not known” (≠ “known not”)

If $\text{at T_1}$ and $\text{at T_2}$ then
revise: $\text{apple} \rightarrow \text{truck}$ or revise: $\text{tomato} \rightarrow \text{truck}$
Fusemate Model = If-Then Rules

If $T_1 < T \leq T_2$ then NO!

If in($T_1$, $B$, $C$) & $T_1 < T_2$ NOT exists $T$ s.th. $T_1 < T \leq T_2$ & unload($T$, $B$, $C$) then in($T_2$, $B$, $C$)

default reasoning: “not known” (≠ “known not”)

If at $T_1$ and at $T_2$ then

revise: at $T_1$ or revise: at $T_2$

A model is a set of if-then rules plus ...
Model = If-Then Rules + Scala Class Hierarchy

abstract class Item { val perishable: Boolean }
abstract class Fruit extends Item { val perishable = true }

abstract class Vehicle { val speed: Int }

case class Truck(id: Int, time: DateTime, load: Set[Item]) extends Vehicle with LogicFact {
  val speed = 80
  val rules = List(
    Truck(id, t, load - item) :- Unload(id, t, item), ...
  )
  def hasPerishableLoad = load contains { _.perishable = true }
}
Model = If-Then Rules + Scala Class Hierarchy

abstract class Item { val perishable: Boolean }
abstract class Fruit extends Item { val perishable = true }

abstract class Vehicle { val speed: Int }

case class Truck(id: Int, time: DateTime, load: Set[Item]) extends Vehicle with LogicFact {
  val speed = 80
  val rules = List(
    Truck(id, t, load - item) :- Unload(id, t, item), ...
  )
  def hasPerishableLoad = load contains { _.perishable = true }
}
Model = If-Then Rules + Scala Class Hierarchy

case class Truck(id: Int, time: DateTime, load: Set[Item]) ... {

... 
@rules(id, time, load)
val rules = List(
    Fail :-
        Unload(id, time, item),
    IF (! load contains items)

    Truck(id, next, load - item) :-
        Step(next, time),
        Unload(id, time, item),
        NOT(t < time, Unload(id, t, item) )
)
}
Model = If-Then Rules + Scala Class Hierarchy

```scala
case class Truck(id: Int, time: DateTime, load: Set[Item]) {

  @rules(id, time, load)
  val rules = List(
    Fail :-
      Unload(id, time, item),
      IF (! load contains items)

    Truck(id, next, load - item) :-
      Step(next, time),
      Unload(id, time, item),
      NOT(t < time, Unload(id, t, item))
  )
}
```

Scala library “set”
Model = If-Then Rules + Scala Class Hierarchy

```scala
case class Truck(id: Int, time: DateTime, load: Set[Item]) ...
{
  ...
  @rules(id, time, load)
  val rules = List(
    Fail :-
      Unload(id, time, item),
      IF (! load contains items)
    Truck(id, next, load - item) :-
      Step(next, time),
      Unload(id, time, item),
      NOT(t < time, Unload(id, t, item) )
  )
}
```

Scala library “set”
Model = If-Then Rules + Scala Class Hierarchy

```scala
case class Truck(id: Int, time: DateTime, load: Set[Item]) ... {
    ...
    @rules(id, time, load)
    val rules = List(
        Fail :-
            Unload(id, time, item),
        IF (! load contains items)
    Truck(id, next, load - item) :-
        Step(next, time),
        Unload(id, time, item),
        NOT(t < time, Unload(id, t, item))
    )
}
```

Macro annotation

```scala
case (Step(next, time1), Unload(id1, time2, item)) if time1 == time && time2 == time && id1 == id
    ! (model exists {
        case Unload(id2, t, item1) if id2 == id &&
            t < time && item1 == item => true
        case _ => false
    }) => Truck(id, next, load - item)
```
Modelling Paradigm Summary

- ✔️ **Inferences** on basis of incomplete information
- ✔️ Derive **multiple** plausible explanations
- ✔️ **Fix** erroneous event data and revise explanations

**Rules**

- ✔️ **Rich data structures**
- ✔️ **Java connectivity**
- ✔️ **Inference engine**

**Infrastructure**

- Declarative
- Infrastruture
**Domain Model** - e.g., If item I is unpacked from a container C at time T then I must have been packed into C at some time S < T

**EPCIS Events** - EPCIS events are sent to fusemate as they become available

**Explanations** - The inference engine derives a set of plausible models consistent with the EPCIS events so far

**Q/A/C** - Where was item I at time T? Item I was unpacked from container C at time T and loc L. Item I has never arrived at L!

**Update** - Plausible models are updated on every new EPCIS event and command provided by user
Case Study
Case Study - Deer Supply Chain

The Use of EPC RFID Standards for Livestock and Meat Traceability

Gary Hartley
New Zealand RFID Pathfinder Group
January 2013
Case Study - Deer Supply Chain

The Use of EPC RFID Standards for Livestock and Meat Traceability

12 events - from farm (NZ) to retailer (DE) encoded in EPCIS
Case Study - Deer Supply Chain

The Use of EPC RFID Standards for Livestock and Meat Traceability

2013

Process Step 4 - Animals arrive at Mountain River Processors’ stun box

Process Step 5 - Cartons of finished Venison cuts packed into cartons at Mountain River processor and moved from the boxing room into chiller room

12 events - from farm (NZ) to retailer (DE) encoded in EPCIS
12 events - from farm (NZ) to retailer (DE) encoded in EPCIS
Case Study - Deer Supply Chain

The Use of EPC RFID Standards for Livestock and Meat Traceability

2013

Gary Hartley
New Zealand RFID Pathfinder Group
January 2013

12 events - from farm (NZ) to retailer (DE) encoded in EPCIS
Case Study - WWWWW

EPCIS defines lower-level events and higher-level “WWWW” concepts


<table>
<thead>
<tr>
<th>Carton-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
</tr>
<tr>
<td>commissioning - active</td>
</tr>
<tr>
<td>Add</td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
</tr>
</tbody>
</table>
Case Study - WWWWW

EPCIS defines lower-level events and higher-level “WWWW” concepts


<table>
<thead>
<tr>
<th>Carton-2</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
<td></td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
<td></td>
</tr>
<tr>
<td>commissioning - active</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td></td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
<td></td>
</tr>
</tbody>
</table>
Case Study - WWWWW

EPCIS defines lower-level events and higher-level “WWW” concepts


<table>
<thead>
<tr>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carton-2</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
</tr>
<tr>
<td>commissioning - active</td>
</tr>
<tr>
<td>Add</td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
</tr>
</tbody>
</table>

Where - Read point
Case Study - WWWW

EPCIS defines lower-level events and higher-level “WWWW” concepts


<table>
<thead>
<tr>
<th>Carton-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
</tr>
<tr>
<td>commissioning - active</td>
</tr>
<tr>
<td>Add</td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
</tr>
</tbody>
</table>

**What**

**Where** - Read point

**Where** - Biz location
Case Study - WWWWW

EPCIS defines lower-level events and higher-level “WWWww” concepts


<table>
<thead>
<tr>
<th>Carton-2</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
<td>Where - Read point</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
<td>Where - Biz location</td>
</tr>
<tr>
<td>commissioning - active</td>
<td>Why - Biz step - Disposition</td>
</tr>
<tr>
<td>Add</td>
<td></td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
<td></td>
</tr>
</tbody>
</table>
Case Study - WWWWW

EPCIS defines lower-level events and higher-level “WWWW” concepts

**WWWww** - What? Where? When? Why?

<table>
<thead>
<tr>
<th>Carton-2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
<td>What</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
<td>Where - Read point</td>
</tr>
<tr>
<td>commissioning - active</td>
<td>Where - Biz location</td>
</tr>
<tr>
<td>Add</td>
<td>Why - Biz step - Disposition</td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
<td>How</td>
</tr>
</tbody>
</table>
## Case Study - WWWWW

**EPCIS defines lower-level events and higher-level “WWWW” concepts**


<table>
<thead>
<tr>
<th>Carton-2</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
<td>Where - Read point</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
<td>Where - Biz location</td>
</tr>
<tr>
<td>commissioning - active</td>
<td>Why - Biz step - Disposition</td>
</tr>
<tr>
<td>Add</td>
<td>How</td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
<td>When</td>
</tr>
</tbody>
</table>
Case Study - WWWW

EPCIS defines lower-level events and higher-level “WWWWS” concepts


<table>
<thead>
<tr>
<th>Carton-2</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
<td>Where - Read point</td>
</tr>
<tr>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
<td>Where - Biz location</td>
</tr>
<tr>
<td>commissioning - active</td>
<td>Why - Biz step - Disposition</td>
</tr>
<tr>
<td>Add</td>
<td>How</td>
</tr>
<tr>
<td>2012-10-25T11:25:53+13:00</td>
<td>When</td>
</tr>
</tbody>
</table>

There are if-then rules for deriving WWWWSs from EPCIS low-level events.
Case Study - Overview

Fusemate output
Tracking “Deer-1” and “Deer-2”

Model representation is in terms of WWWW, Aggregation and Association concepts
Case Study - Overview

Fusemate output
Tracking “Deer-1” and “Deer-2”

Model representation is in terms of WWWW, Aggregation and Association concepts
Case Study - Overview

Fusemate output
Tracking “Deer-1” and “Deer-2”

Model representation is in terms of WWWW, Aggregation and Association concepts
Case Study - Overview

Fusemate output
Tracking “Deer-1” and “Deer-2”

Model representation is in terms of WWWW, Aggregation and Association concepts
Case Study - Overview

Fusemate output
Tracking “Deer-1” and “Deer-2”

Model representation is in terms of WWWW, Aggregation and Association concepts
### Modelling Experiment - Multiple Explanations

<table>
<thead>
<tr>
<th>Carton-2</th>
<th>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</th>
<th>MountainRiverProcessors/NewZealand/CHILLER_ROOM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>commissioning - active</td>
<td>Add</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2012-10-25T11:25:53+13:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Carton-2</th>
<th>PrimeMeat/Germany/DOCK_DOOR</th>
<th>PrimeMeat/Germany/DOCK_DOOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>shipping - in transit</td>
<td>Observe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2012-12-11T22:40:28+13:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Carton-2</th>
<th>Retailer-1/Germany/RECEIVING_BAY</th>
<th>Retailer-1/Germany/IN_STORE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>receiving - sellable_accessible</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>2012-12-12T01:58:34+13:00</td>
<td></td>
</tr>
</tbody>
</table>
## Modelling Experiment - Multiple Explanations

<table>
<thead>
<tr>
<th>Carton-2</th>
<th>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>commissioning - active</td>
</tr>
<tr>
<td></td>
<td>2012-10-25T11:25:53+13:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Carton-2</th>
<th>PrimeMeat/Germany/DOCK_DOOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>shipping - in_transit</td>
</tr>
<tr>
<td></td>
<td>Observe</td>
</tr>
<tr>
<td></td>
<td>2012-12-11T22:40:28+13:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Carton-2</th>
<th>Retailer-1/Germany/RECEIVING_BAY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>receiving - sellable_accessible</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>2012-12-12T01:58:34+13:00</td>
</tr>
</tbody>
</table>

### What is known
Modelling Experiment - Multiple Explanations

What is known
- Carton-2 has arrived at Retailer-1 in Germany
## Modelling Experiment - Multiple Explanations

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Type</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT</td>
<td>commissioning - active</td>
<td>2012-10-25T11:25:53+13:00</td>
</tr>
<tr>
<td>Observe</td>
<td>MountainRiverProcessors/NewZealand/CHILLER_ROOM</td>
<td>shipping - in_transit</td>
<td>2012-12-11T22:40:28+13:00</td>
</tr>
<tr>
<td>Delete</td>
<td>Retailer-1/Germany/RECEIVING_BAY</td>
<td>receiving - sellable_accessible</td>
<td>2012-12-12T01:58:34+13:00</td>
</tr>
</tbody>
</table>

### What is known

- Carton-2 has arrived at Retailer-1 in Germany
- Carton-13 was added to the supply chain like Carton-2 above
Modelling Experiment - Multiple Explanations

What is known
- Carton-2 has arrived at Retailer-1 in Germany
- Carton-13 was added to the supply chain like Carton-2 above
- Carton-13 leaves no trace but it should also have arrived at Retailer-1 in Germany
Modelling Experiment - Multiple Explanations

What is known
- Carton-2 has arrived at Retailer-1 in Germany
- Carton-13 was added to the supply chain like Carton-2 above
- Carton-13 leaves no trace but it should also have arrived at Retailer-1 in Germany

What went wrong?

Fusemate diagnosis scenario - Compute some plausible explanations
Modelling Experiment - Multiple Explanations

We start the diagnosis by telling Fusemate that Carton-13 behaves like Carton-2

trackByCopy("urn:epc:id:sgtin:94130000.01420.2", "urn:epc:id:sgtin:94130000.01420.13")

Result

- **Carton-13**
  - MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
  - Commissioning - active
  - Add
  - 2012-10-25T11:25:53+13:00

- **Carton-2**
  - MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
  - Commissioning - active
  - Add
  - 2012-10-25T11:25:53+13:00

- **ShippingContainer-1**
  - MountainRiverProcessors/NewZealand/EXIT_POINT
  - Shipping - in_transit
  - Observe
  - 2012-10-26T07:53+13:00

- **Carton-13**
  - MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
  - Commissioning - active
  - AggregatedInfo
  - 2012-10-26T07:31:09+13:00

- **Carton-2**
  - MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
  - Commissioning - active
  - AggregatedInfo
  - 2012-10-26T07:31:09+13:00

- **Carton-13**
  - MountainRiverProcessors/NewZealand/CHILLER_ROOM
  - Commissioning - active
  - 2012-10-25T11:25:53+13:00

- **Carton-2**
  - MountainRiverProcessors/NewZealand/CHILLER_ROOM
  - Commissioning - active
  - 2012-10-25T11:25:53+13:00

- **Carton-13**
  - MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
  - Commissioning - active
  - AggregatedInfo
  - 2012-10-26T07:31:09+13:00

- **Carton-2**
  - MountainRiverProcessors/NewZealand/CHILLER_ROOM
  - Commissioning - active
  - AggregatedInfo
  - 2012-10-26T07:31:09+13:00
Modelling Experiment - Multiple Explanations

We start the diagnosis by telling Fusemate that Carton-13 behaves like Carton-2

```
trackByCopy("urn:epc:id:sgtin:94130000.01421.2", "urn:epc:id:sgtin:94130000.01420.13")
```

Result

```
+ Carton-13
| MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT |
| MountainRiverProcessors/NewZealand/CHILLER_ROOM |
| commissioning - active |
| Add |
| 2012-10-25T11:25:53+13:00 |

+ Carton-2
| MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT |
| MountainRiverProcessors/NewZealand/CHILLER_ROOM |
| commissioning - active |
| Add |
| 2012-10-25T11:25:53+13:00 |

+ Carton-13
| MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT |
| MountainRiverProcessors/NewZealand/CHILLER_ROOM |
| commissioning - active |
| AggregatedInfo |
| 2012-10-26T07:31:09+13:00 |

+ Carton-2
| MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT |
| MountainRiverProcessors/NewZealand/CHILLER_ROOM |
| commissioning - active |
| AggregatedInfo |
| 2012-10-26T07:31:09+13:00 |

+ ShippingContainer-1
| MountainRiverProcessors/NewZealand/EXIT_POINT |
| shipping - in_transit |
| Observe |
| 2012-10-26T07:53+13:00 |

+ ShippingContainer-1
| MountainRiverProcessors/NewZealand/ENTRY_GATE |
| shipping - in_transit |
| Observe |
| 2012-10-26T09:13+13:00 |

+ ShippingContainer-1
| PrimeMeat/Germany/DOCK_DOOR |
| shipping - in_transit |
| DisaggregatedFrom |
| 2012-12-11T01:09:46+13:00 |

+ Carton-13
| Retailer-1/Germany/RECEIVING_BAY |
| Retailer-1/Germany/IN_STORE |
| receiving - sellable_accessible |
| Delete |
| 2012-12-12T01:58:34+13:00 |

+ Carton-2
| Retailer-1/Germany/RECEIVING_BAY |
| Retailer-1/Germany/IN_STORE |
| receiving - sellable_accessible |
| Delete |
| 2012-12-12T01:58:34+13:00 |
```
Modelling Experiment - Multiple Explanations

We start the diagnosis by telling Fusemate that Carton-13 behaves like Carton-2.

```
trackByCopy("urn:epc:id:sgtin:94130000.01421.2", "urn:epc:id:sgtin:94130000.01426.13")
```

Result

**Carton-13**
- MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
- MountainRiverProcessors/NewZealand/CHILLER_ROOM
  - commissioning - active
  - Add 2012-10-25T11:25:53+13:00
  - AggregatedInto 2012-10-26T07:31:09+13:00

**Carton-2**
- MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT
- MountainRiverProcessors/NewZealand/CHILLER_ROOM
  - commissioning - active
  - Add 2012-10-25T11:25:53+13:00
  - AggregatedInto 2012-10-26T07:31:09+13:00

**ShippingContainer-1**
- PortOfLyttleton/NewZealand/ENTRY_GATE
  - shipping - in_transit
  - Observe 2012-10-26T09:13+13:00

**Carton-13**
- PrimeMeat/Germany/DOCK_DOOR
  - shipping - in_transit
  - Observe 2012-12-11T22:40:28+13:00
  - DisaggregatedFrom 2012-12-11T01:09:46+13:00

**Carton-2**
- PrimeMeat/Germany/DOCK_DOOR
  - shipping - in_transit
  - Observe 2012-12-11T22:40:28+13:00

**Retailer-1/Germany/RECEIVING_BAY**
- receiving - sellable_accessible
  - Delete 2012-12-12T01:58:34+13:00

**Retailer-1/Germany/IN_STORE**
  - receiving - sellable_accessible
  - Delete 2012-12-12T01:58:34+13:00
Modelling Experiment - Multiple Explanations

We start the diagnosis by telling Fusemate that Carton-13 behaves like Carton-2

trackByCopy("urn:epc:id:sgtin:94130000.01420.13", "urn:epc:id:sgtin:94130000.01420.2")

Result

Now add knowledge of what (not) happened
Modelling Experiment - Multiple Explanations

A phone call confirms that Carton-13 was not seen at the DOCK_DOOR any time after 11/12/2012:
This information is provided to Fusemate (details not shown here)

Plausible Explanation (1)

"Carton-13 has been removed from Container-1 but there is no trace from then on"
Modelling Experiment - Multiple Explanations

A phone call confirms that Carton-13 was not seen at the DOCK_DOOR any time after 11/12/2012:
This information is provided to Fusemate (details not shown here)

Plausible Explanation (1)

“Carton-13 has been removed from Container-1 but there is no trace from then on”
Modelling Experiment - Multiple Explanations

A phone call confirms that Carton-13 was not seen at the DOCK_DOOR any time after 11/12/2012:
This information is provided to Fusemate (details not shown here)

**Plausible Explanation (1)**

"Carton-13 has been removed from Container-1 but there is no trace from then on"
Modelling Experiment - Multiple Explanations

The user asks the system to compute the next plausible model

Plausible Explanation (2)

"Carton-13 has not been removed from Container-1 (still in the container?)"
Modelling Experiment - Multiple Explanations

The user asks the system to compute the next plausible model

Plausible Explanation (2)

“Carton-13 has not been removed from Container-1 (still in the container?)”
Modelling Experiment - Multiple Explanations

The user asks the system to compute the next plausible model

**Plausible Explanation (3)**

"Carton-13 was not loaded into Container-1 in the first place" (And hence cannot be unloaded either as per rules)
Modelling Experiment - Multiple Explanations

The user asks the system to compute the next plausible model

Plausible Explanation (3)

“Carton-13 was not loaded into Container-1 in the first place”
(And hence cannot be unloaded either as per rules)
Modelling Experiment - Multiple Explanations

The user asks the system to compute the next plausible model

Plausible Explanation (3)

"Carton-13 was not loaded into Container-1 in the first place" (And hence cannot be unloaded either as per rules)
Modelling Experiment - Multiple Explanations

The user asks the system to compute the next plausible model

Plausible Explanation (4)

N/A

In particular Fusemate does not generate:
“Carton-13 has been removed from the Container 1 but was not loaded earlier into Container 1”
User Interface - Under Development

OK
command read("demo/deer/data/event-05.xml")
OK
command read("demo/deer/data/event-06.xml")
OK
command read("demo/deer/data/event-07.xml")
OK
command read("demo/deer/data/event-08.xml")
OK
command read("demo/deer/data/event-09.xml")
OK
command read("demo/deer/data/event-10.xml")

command track("urn:uuid:uuid-94130000.01620.2")
query currentModel filter ( isAbout(_, "Carton-2") )
command toDotFile(currentModel, "demo/deer/out/client-tracking1.dot", display = true)
query toDot(currentModel)

Carton-2
MountainRiverProcessors/NewZealand/BONING_ROOM_EXIT - MountainRiverProcessors/NewZealand/CHILLER_ROOM
commissioning - active
Add
2012-10-25T11:25:53+13:00
User Interface - Under Development
Conclusions and Future Work

• Developed Fusemate situational awareness system
  - Fusemate = Logic Programming + Belief Revision + Scala programming language
  - Experimented with Deer supply chain case study
  - Future work
    • User interface
    • More case studies
    • Integration with video tracking
    • Probabilities
    • Temporal logic: $\Box t \cdot \text{shipped}(B) \rightarrow \Diamond s \cdot s \leq t + 5 \land \text{received}(B)$
    • Ontologies