



Vrije Universiteit Brussel

STAR Lab

Towards Semantically Grounded Decision Rules Using ORM⁺

Presenter: Yan Tang (Belgium)

Authors: Yan Tang, Peter Sypns and Robert Meersman

RuleML 2007 (Florida, USA)



Summary

- Background: Ontologies and semantically grounded decision rules
- ORM approach to ontological commitments and its problems
- The solution: ORM⁺
- Discussion
- Conclusion
- Future Work



Background

- Why semantically grounded decision rules?
 - Decision support systems mainly contain *non-sharable* decision rules
 - Decision rules are rarely written in an agreed, formal way
 - Difficult to check the *redundancy* and *similarity* in the decision rule set
 - E.g. if (weather is bad) then (stay at home)
else if(weather is good) then (go for a walk).
 - if not (it's sunny) then (stay at home)
else if (it's sunny) then (go for a walk).
- Use ontology to store the conceptual definition and decision items
 - E.g. “bad weather”
- Ontology
 - Explicit, sharable, formal, conceptual, stored in computers
 - DOGMA (Developing Ontology-Grounded Methods and Applications)
Approach to ontology:
 - *Double articulation*: ontology = lexon base+ commitment (R. Meersman, 1999)



Double articulation principle of ontology engineering

- Lexon (conceptualization):
 - Lexon: plausible binary fact
 - E.g. $\langle \gamma, \text{driver, has, is issued to, drivers license} \rangle$; $\langle \gamma, \text{driving experience, is of, has, driver} \rangle$
- Commitment (axiomatization):
 - Describes particular application views of reality
 - the use of lexons
 - Provides multiple views on stored lexons
 - Needs to be expressed by commitment language

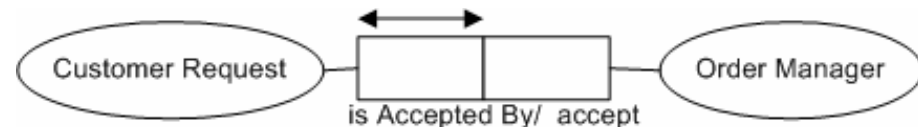


ORM approach to commitment and its problems

- ORM

- Object Role Modeling, Terry Halpin, 1990's
- Intended for modeling and querying DB at a conceptual level
- Why ORM?

- Semantically rich modeling language to model and visualize commitments for non-technical domain experts
 - Expressive capabilities in its graphical notation
 - Verbalization possibilities
- ORM-ML for machines
 - Store ORM graphs
 - Can be mapped to OWL



- Problems

- ORM still lacks several logical operators and connectors for the decision semantics, e.g. implication
- Difficulties to specify some logical operators, e.g. negation

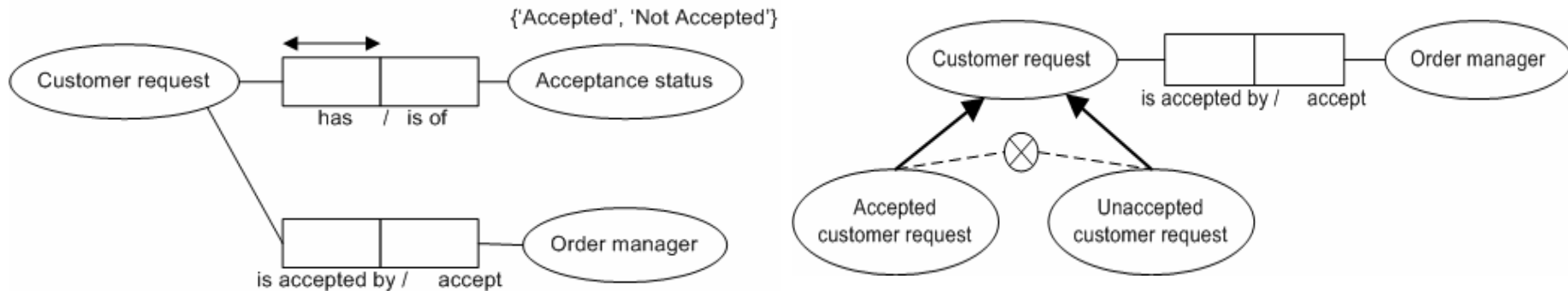


The solution: ORM⁺

- ORM⁺ : an extension of ORM
 - Basic propositional logic connectives
 - Negation
 - Conjunction
 - Disjunction
 - Implication
 - Modality operators
 - Necessity
 - Possibility
 - Process operator
 - Sequence

ORM+ Approach to Commitments – Negation

- ORM: “closed-world” assumption¹⁾ use value type; 2) use exclusion constraint

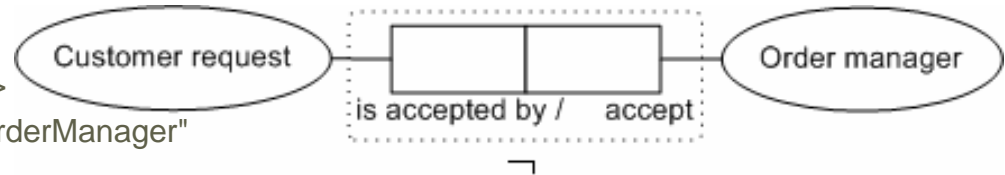


- ORM+: “open-world” assumption
 - because machine needs to know whether the negative situation is taken or not

ORM⁺ Negation Graph & Markup Language

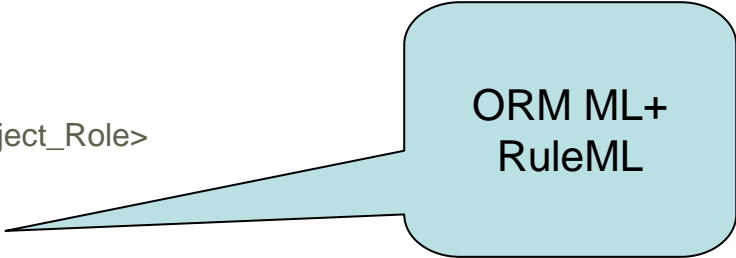
```

<Predicate id="lexon-2">
  <Object_Role ID="lexon2_forward"
  Object="CustomerRequest" Role="isAcceptedBy"/>
  <Object_Role ID="lexon2_backward" Object="OrderManager"
  Role="Accept"/>
</Predicate>
  
```



```

...
<Neg>
  <Atom>
    <Predicate>
      <Object_Role>lexon2_forward</Object_Role>
    </Predicate>
  </Atom>
  <Atom>
    <Predicate>
      <Object_Role>lexon2_backward</Object_Role>
    </Predicate>
  </Atom>
</Neg>
  
```





ORM+ Approach to Commitments – Conjunction

- ORM: doesn't exist
- ORM+

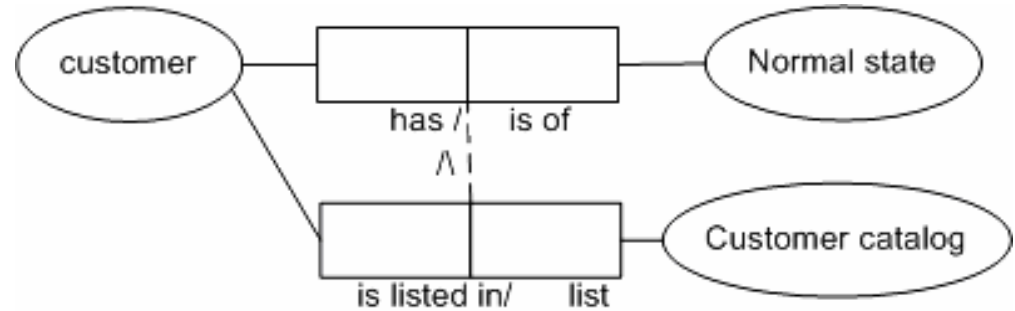
```

...
<Predicate id="lexon-3">
  <Object_Role ID="lexon3_forward"
    Object="Customer" Role="has"/>
  <Object_Role ID="lexon3_backward"
    Object="NormalState" Role="isOf"/>
</Predicate>

<Predicate id="lexon-4">
  <Object_Role ID="lexon4_forward"
    Object="Customer" Role="isListedIn"/>
  <Object_Role ID="lexon4_backward"
    Object="CustomerCatalog" Role="list"/>
</Predicate>

...
<And>
  <Atom>
    <Predicate>lexon-3</Predicate>
  </Atom>
  <Atom>
    <Predicate>lexon-4</Predicate>
  </Atom>
</And>

```



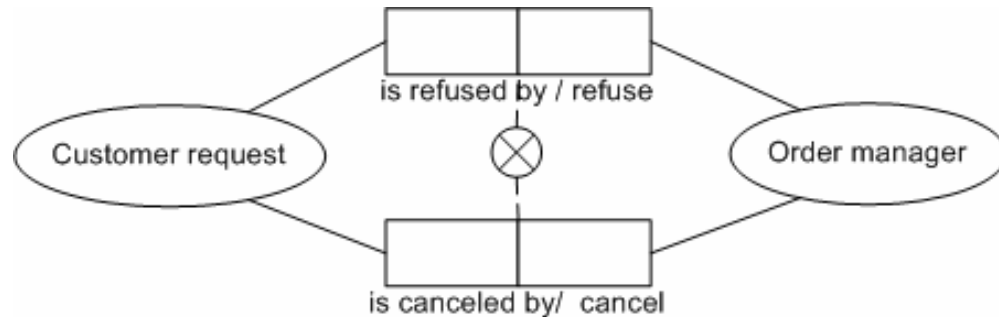


ORM+ Approach to Commitments – Disjunction

- ORM:
 - exclusive-or constraint
 - without (ordinary or)
- ORM+:
 - reuse the exclusive constraint


```
<constraint type="exclusive-or">
  <Object_Role>lexon-5</Object_Role>
  <Object_Role>lexon-6</Object_Role>
</constraint>
```
 - Ordinary or


```
<Or>
  <Atom>
    <Predicate>lexon-5</Predicate>
  </Atom>
  <Atom>
    <Predicate>lexon-6</Predicate>
  </Atom>
</Or>
```
 - ...



ORM+ Approach to Commitments – Implication

- ORM: doesn't exist
- ORM+:

<Rule type="Implies">

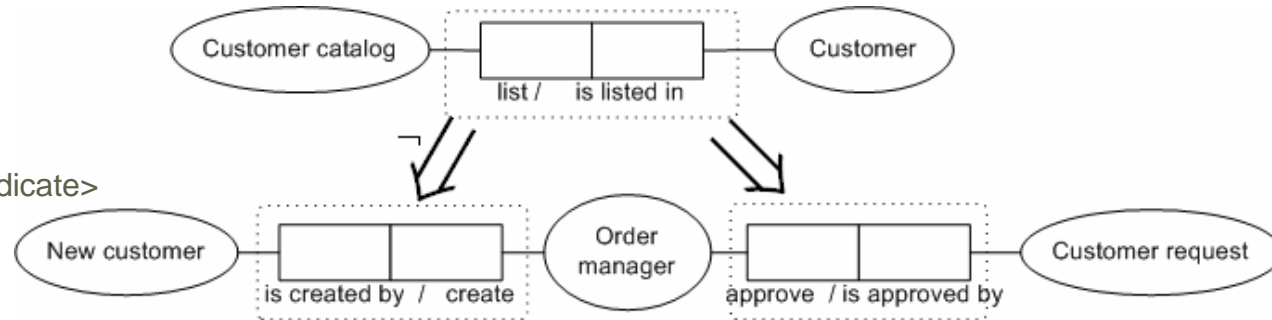
```

<head>
  <Atom>
    <Predicate>lexon-7</Predicate>
  </Atom>
</head>
<body>
  <Neg>
    <Atom><Predicate><Object_Role>lexon8-
forward</Object_Role></Predicate></Atom>
    <Atom><Predicate><Object_Role>lexon8-
backward</Object_Role></Predicate></Atom>
  </Neg>
</body>
</Rule>
  
```

<Rule type="Implies">

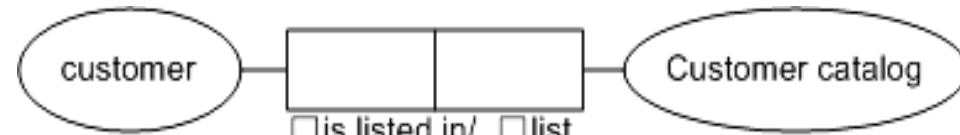
```

<head>
  <Atom><Predicate>lexon-9</Predicate></Atom>
</head>
<body>
  <Atom><Predicate>lexon-8</Predicate></Atom>
</body>
</Rule>...
  
```



ORM+ Approach to Commitments – Necessity

- ORM: verbalization only (ORM2)
- ORM+:



...

```
<Predicate id="lexon-9">
```

```
  <Object_Role ID="lexon9_forward"
    Object="Customer" Role="isListedIn"/>
```

```
  <Object_Role ID="lexon9_backward"
    Object="CustomerCatalog" Role="list"/>
```

```
</Predicate>
```

...

```
<Constraint xsi:type= "Necessity">
```

```
  <Object_Role>
```

```
    lexon9_forward
```

```
  </Object_Role>
```

```
  <Object_Role>
```

```
    lexon9_backward
```

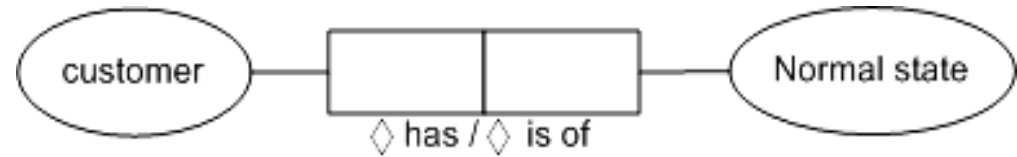
```
  </Object_Role>
```

```
</Constraint>
```

...

ORM+ Approach to Commitments – Possibility

- ORM: verbalization only (ORM2)
- ORM+:



```

...
<Predicate id="lexon-10">
  <Object_Role ID="lexon10_forward"
    Object="Customer" Role="has"/>
  <Object_Role ID="lexon10_backward"
    Object="NormalState" Role="isOf"/>
</Predicate>

```

```

...
<Constraint xsi:type="Possibility">
  <Object_Role>
    lexon10_forward
  </Object_Role>
  <Object_Role>
    lexon10_backward
  </Object_Role>
</Constraint>

```

...

ORM+ Approach to Commitments – Sequence

- ORM: doesn't exist
- ORM+: used to control the process flow

...

```
<Rule xsi:type="Sequence" direction="forward">
```

```
<Atom>
```

```
<Predicate>lexon-11</Predicate>
```

```
</Atom>
```

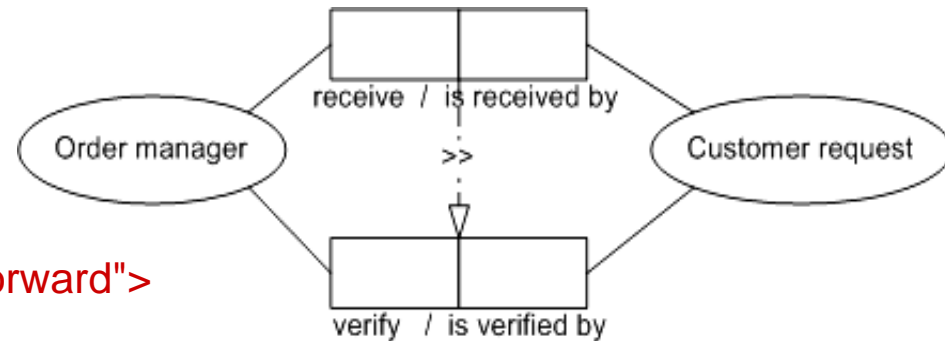
```
<Atom>
```

```
<Predicate>lexon-12</Predicate>
```

```
</Atom>
```

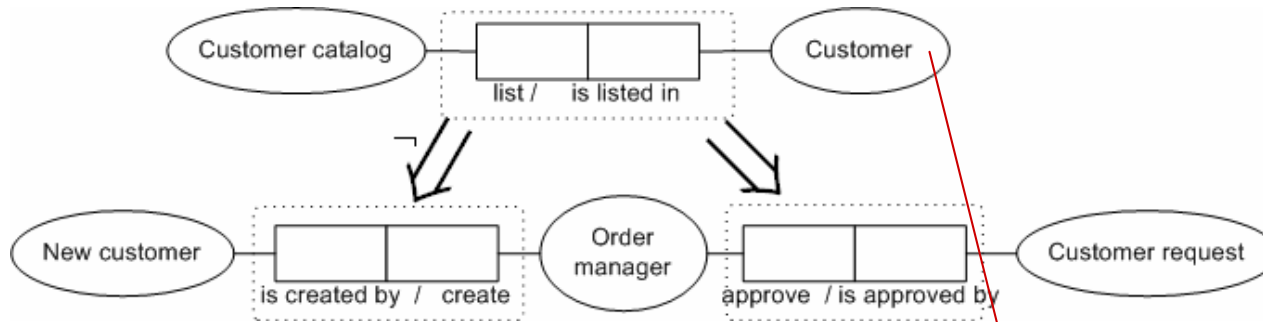
```
</Rule>
```

...

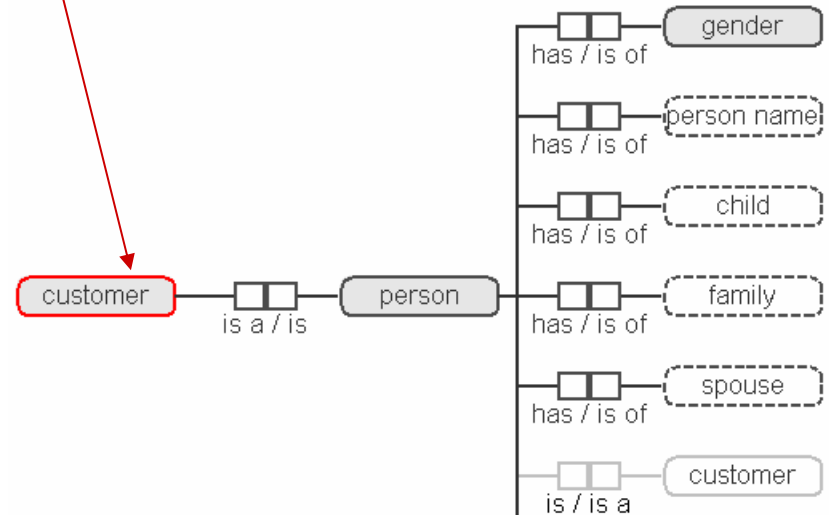




Semantically rich decision rules (example)



- Concepts are defined in domain ontologies
- Decision items need to be annotated with domain ontologies
- Tools: T-Lex, SDT plugin @ STARlab





Conclusion

- Conclusion
 - ORM⁺ is to *model, visualize, store and share* semantically rich decision rules
 - ORM⁺ extends ORM graphical notations
 - ORM⁺ ML = ORM ML + RuleML + others
 - ORM⁺ ML schema reuses **31** type definitions from ORM ML schema, **10** type definitions from *FOL Rule-ML* and introduces **7** new type definitions
 - Can be used by different rule engines
 - InfoSapient
 - JLisa
 - OpenRules
 - OpenLexicon
 - Etc.



Discussion and Future Work

- Discussion:
 - More symbols, more difficult to verbalize the commitments
 - Complexity
- Future work:
 - Mapping ORM+ ML to one inference engine
 - ORM+ visualization tool



Vrije Universiteit Brussel

STAR Lab

Questions?

- Thank you!