Knowledge Representation Tools for Electronic Commerce

Francesco M Donini

Dipartimento di Studi sulla Comunicazione
Università della Tuscia – Viterbo
P2P EC —what this talk is about?

Peer-to-Peer Electronic Commerce
Peer-to-Peer Electronic Commerce

offers (supplies) requests (demands) services

meet in

Electronic Marketplace + trusted third party
Peer-to-Peer Electronic Commerce

<table>
<thead>
<tr>
<th>offers (supplies)</th>
<th>meet in</th>
<th>Electronic Marketplace</th>
</tr>
</thead>
<tbody>
<tr>
<td>requests (demands)</td>
<td></td>
<td>+ trusted</td>
</tr>
<tr>
<td>services</td>
<td></td>
<td>third party</td>
</tr>
</tbody>
</table>

Marketplace: mostly, Web Site with human interaction
Peer-to-Peer Electronic Commerce

P2P EC — what this talk is about?

Marketplace: mostly, Web Site with human interaction

Renowned example: eBay

http://www.ebay.com

Marketplace + trusted third party

offers (supplies)
requests (demands)
services
Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on www.automobili.com
Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on www.automobili.com
- a room to share in Rome: 851 offers on www.easystanza.com
Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on www.automobili.com
- a room to share in Rome: 851 offers on www.easystanza.com
- a used Notebook PC: 2361 offers on informatica.ebay.it
Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on www.automobili.com
- a room to share in Rome: 851 offers on www.easystanza.com
- a used Notebook PC: 2361 offers on informatica.ebay.it

...how did you choose?
Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on www.automobili.com
- a room to share in Rome: 851 offers on www.easystanza.com
- a used Notebook PC: 2361 offers on informatica.ebay.it

... \textit{which reasoning} did you employed?
P2P is not B2C

- B2C: Business-to-Consumer
- P2P: Peer-to-Peer
P2P is not B2C

- B2C: Business-to-Consumer
  - usually, the seller owns the Web Site

- P2P: Peer-to-Peer
  - the Web Site is of some third party
P2P is not B2C

B2C: Business-to-Consumer
- usually, the seller owns the Web Site
- the seller publishes offers

P2P: Peer-to-Peer
- the Web Site is of some third party
- both parties can publish on the Web Site
P2P is not B2C

- B2C: Business-to-Consumer
  - usually, the seller owns the Web Site
  - the seller publishes offers
  - the client browses...

- P2P: Peer-to-Peer
  - the Web Site is of some third party
  - both parties can publish on the Web Site
  - *Both* parties may take initiative (and browse...)
Thesis of the talk  —why CILC should care?
Semantic Annotation is making Electronic Commerce an arena for Knowledge-based applications
Thesis of the talk — why CILC should care?

Semantic Annotation is making Electronic Commerce an arena for Knowledge-based applications

Knowledge Representation tools can be used in Logic-based Electronic Commerce applications.
Outline of the talk —how I will try to argue?

1. ✓ P2P Electronic Commerce
2. *Enabling technologies*
3. General assumptions
4. Reasoning for Matchmaking
5. Reasoning for Negotiation
6. Languages and expressivity
7. What next?
“The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”
“The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”

— **OWL** - Web Ontology Language Overview
“The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”

— OWL - Web Ontology Language Overview

— DAML - DARPA Agent Markup Language
Semantic Annotation

“The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”

— **OWL** - Web Ontology Language Overview

**DAML** - DARPA Agent Markup Language

Web Services can be described through languages like **DAML-S, OWL-S**, . . .
"On-sale PCs are . . .

home PCs with at most one OS, of type WinX"

< owl:Class rdf:ID = "onSalePC" />
< rdfs: subClassOf >
  < owl:intersectionOf rdf:parseType = "Collection" >
    < owl:Class rdf:ID = "homePC" />
    < owl:Restriction >
      < owl:onProperty rdf:resource = "hasOS" />
      < owl:maxCardinality
        rdf:datatype = "&xsd;nonNegativeInteger" >
        1
      < /owl:maxCardinality >
    < owl: Restriction >
    < owl: allValuesFrom
      rdf:resource = "#winX" />
    < /owl: Restriction >
  < /owl:intersectionOf >
< /rdfs:subClassOf >
< /owl:Class >
An Example in OWL — more precisely, OWL-Lite

"On-sale PCs are ... home PCs with at most one OS, of type WinX"

```xml
<owl:Class rdf:ID="onSalePC"/>
<rdf:sparql:subClassOf>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="homePC"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasOS"/>
      <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
      <owl:allValuesFrom rdf:resource="#winX"/>
    </owl:Restriction>
  </owl:intersectionOf>
</rdf:sparql:subClassOf>
<owl:Class/>
```
“On-sale PCs are ... home PCs with at most one OS, of type WinX”

```xml
<owl:Class rdf:ID="onSalePC"/>
<rdfs:subClassOf>
    <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:ID="homePC"/>
        <owl:Restriction>
            <owl:onProperty rdf:resource="hasOS"/>
            <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
            <owl:allValuesFrom rdf:resource="#winX"/>
        </owl:Restriction>
    </owl:intersectionOf>
</rdfs:subClassOf>
</owl:Class>
```
General Assumptions

Based on Semantic Annotation, we assume that

Offers, requests, services are *logic formulas* $O, R, S, \ldots$
General Assumptions

Based on Semantic Annotation, we assume that

- Offers, requests, services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic theory* $\mathcal{T}$
General Assumptions

Based on Semantic Annotation, we assume that:

- Offers, requests, services are *logic formulas* \( O, R, S, \ldots \)
- The marketplace ontology is a *logic theory* \( \mathcal{T} \)
- an agreement betw. \( O \) and \( R \) is either a...
General Assumptions

Based on Semantic Annotation, we assume that

- Offers, requests, services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic theory* $\mathcal{T}$
- an agreement betw. $O$ and $R$ is either a...
  - model of $\mathcal{T} \cup \{O, R\}$, or a...
General Assumptions

Based on Semantic Annotation, we assume that

- Offers, requests, services are logic formulas $O, R, S, \ldots$
- The marketplace ontology is a logic theory $\mathcal{T}$
- An agreement between $O$ and $R$ is either a...
  - model of $\mathcal{T} \cup \{O, R\}$, or a...
  - set of models of $\mathcal{T} \cup \{O, R\}$, or a...
General Assumptions

Based on Semantic Annotation, we assume that

- Offers, requests, services are *logic formulas* $O, R, S, \ldots$

- The marketplace ontology is a *logic theory* $\mathcal{T}$

- an agreement betw. $O$ and $R$ is either a...
  - model of $\mathcal{T} \cup \{O, R\}$, or a...
  - set of models of $\mathcal{T} \cup \{O, R\}$, or a...
  - formula consistent with $\mathcal{T} \cup \{O, R\}$
Most of the talk — the past is the prologue

1. ✓ P2P Electronic Commerce
2. ✓ Enabling technologies
3. ✓ General assumptions
4. *Reasoning for Matchmaking*
5. Reasoning for Negotiation
6. Languages and expressivity
7. What next?
What’s Matchmaking?

First phase in a Bilateral Commercial Transaction:

1. *Matchmaking* (find counterpart)
2. Negotiation (agree/tradeoff details)
3. Exchange (goods, services, money)
An Example — a cognitive experiment

From Sunday Times, online marketplace

An Example — a cognitive experiment

From *Sunday Times, online marketplace*


- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England
An Example — a cognitive experiment

From *Sunday Times, online marketplace*

- **Request:** Ferrari 430 Coupe/Spider urgently required. Best prices paid. Immediate decision.

- **Offer:** 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

Do they match?
An Example — a cognitive experiment

From *Sunday Times, online marketplace*


- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

*How well* they match? (compared to other offers/requests)
Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a facilitator system

—But: which reasoning?
Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a facilitator system

—But: which reasoning?

Comparer (possibly with deduction)
Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a facilitator system

—But: which reasoning?

- Compare (possibly with deduction)
- Posit missing information
Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a *facilitator* system

—But: which reasoning?

- *Compare* (possibly with deduction)
- *Posit* missing information
- *Revise* conflicting issues
A first classification based on $\models$

An offer $O$ and a request $R$ match...

\textit{exactly} if $\mathcal{T} \models O \equiv R$
A first classification based on $\models$

An offer $O$ and a request $R$ match...

- **exactly** if $\mathcal{T} \models O \equiv R$
- **potentially** if $\mathcal{T} \not\models \neg(O \land R)$
An offer $O$ and a request $R$ match...

- **exactly** if $\mathcal{T} \models O \equiv R$

- **potentially** if $\mathcal{T} \not\models \neg(O \land R)$

- **i.e.**, if $O \land R$ is consistent with $\mathcal{T}$
A first classification based on $\models$

An offer $O$ and a request $R$ match...

- **exactly** if $\mathcal{T} \models O \equiv R$
- **potentially** if $\mathcal{T} \not\models \neg(O \land R)$
- i.e., if $O \land R$ is consistent with $\mathcal{T}$
- **partially** if $\mathcal{T} \models \neg(O \land R)$
A first classification based on $\models$

An offer $O$ and a request $R$ match...

- **exactly** if $\mathcal{T} \models O \equiv R$
- **potentially** if $\mathcal{T} \not\models \neg(O \land R)$
  - i.e., if $O \land R$ is consistent with $\mathcal{T}$
- **partially** if $\mathcal{T} \models \neg(O \land R)$
  - significant if only some details conflict
A first classification based on $\models$

An offer $O$ and a request $R$ match...

- **exactly** if $\mathcal{T} \models O \equiv R$
- **potentially** if $\mathcal{T} \not\models \neg(O \land R)$
  - *i.e.*, if $O \land R$ is consistent with $\mathcal{T}$
- **partially** if $\mathcal{T} \models \neg(O \land R)$
  - significant if only some details conflict
- **“plug-in” (w.r.t. $R$)** if $\mathcal{T} \models R \Rightarrow O$
A first classification based on \( \models \)

An offer \( O \) and a request \( R \) match...

- **exactly** if \( \mathcal{T} \models O \equiv R \)

- **potentially** if \( \mathcal{T} \nvdash \neg (O \land R) \)
  - *i.e.*, if \( O \land R \) is consistent with \( \mathcal{T} \)

- **partially** if \( \mathcal{T} \models \neg (O \land R) \)
  - significant if only some details conflict

- **“plug-in”** (w.r.t. \( R \)) if \( \mathcal{T} \models R \Rightarrow O \)

- **fully** (w.r.t. \( R \)) if \( \mathcal{T} \models O \Rightarrow R \)
Evaluating the match


Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England
Evaluating the match


- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

conflicting info: 430 vs. 360 (different models)
Evaluating the match

- Request: Ferrari 430  *Coupe/Spider urgently required*. Best prices paid. Immediate decision.

- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

In $R$, not in $O$: *Coupe/Spider, urgently required*
Evaluating the match


Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

In $O$, not in $R$: color Argento, Bordeaux Leather seats, 22,700 miles, ...
Abduction (history)

C. S. Peirce (1839–1914)
From $A \Rightarrow B$ and $B$, abduce $A$

Abduction was the first step of scientific reasoning, the other two being
  - Deduction
  - Induction

since Pople [1973] has been used to formalize Diagnosis in AI
Abduction for P2P EC

- Let $\mathcal{L}$ be a logic language
- $R$ a request in $\mathcal{L}$
- $O$ a possible offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology
Abduction for P2P EC

- Let $\mathcal{L}$ be a logic language
- $R$ a request in $\mathcal{L}$
- $O$ a possible offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

find a hypothesis $H$ such that
Abduction for P2P EC

- Let $\mathcal{L}$ be a logic language
- $R$ a request in $\mathcal{L}$
- $O$ a possible offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

- find a hypothesis $H$ such that $H \land O$ is satisfiable in $\mathcal{T}$
Abduction for P2P EC

- Let $\mathcal{L}$ be a logic language
- $R$ a request in $\mathcal{L}$
- $O$ a possible offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

find a hypothesis $H$ such that

- $H \land O$ is satisfiable in $\mathcal{T}$
- $\mathcal{T} \models H \land O \Rightarrow R$
Intuition

When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$. 
Intuition

When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$.

Will $O$ accept $H$?
When \( R \) evaluates its possible transaction with \( O \), before concluding the transaction, \( R \) and \( O \) should agree on \( H \) will \( O \) accept \( H \)?

vice versa for \( O \), with a different \( H' \) such that

\[
T \models R \land H' \Rightarrow O
\]
What Abduction is good for?

- compute a *score* for each counteroffer
What Abduction is good for?

- compute a *score* for each counteroffer
- *e.g.*, number of hypotheses in best $H$
What Abduction is good for?

- compute a *score* for each counteroffer
- *e.g.*, number of hypotheses in best $H$
- *e.g.*, expected utility from $H$'s
What Abduction is good for?

- compute a *score* for each counteroffer
  - *e.g.*, number of hypotheses in best $H$
  - *e.g.*, expected utility from $H$’s
- construct an *explanation* for match suggestions
What Abduction is good for?

- compute a *score* for each counteroffer
  - *e.g.* number of hypotheses in best $H$
  - *e.g.* expected utility from $H$’s

- construct an *explanation* for match suggestions
  - *e.g.* a facilitator that suggests “Offer 213 seems to be the best, supposing your requests color:blue and Credit Card Payment are satisfied”
Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
Best hypotheses

Different criteria:

- **shortest** $H$ — fewer issues to be set
- **maximally ignorant** $H$ — minimal consequences
Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
- *maximally ignorant* $H$ — minimal consequences
- language-specific
Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
- *maximally ignorant* $H$ — minimal consequences
- *language-specific*
  - *e.g.*, minimal conjunctions if $\lor, \neg \notin \mathcal{L}$
Comparing criteria

\[ R = FiatPanda \land radio \land fogLamps \]
Comparing criteria

\[ R = \text{FiatPanda} \land \text{radio} \land \text{fogLamps} \]

\[ O = \text{FiatPanda} \land \text{year2000} \]
Comparing criteria

\[ R = \text{FiatPanda} \land \text{radio} \land \text{fogLamps} \]

\[ O = \text{FiatPanda} \land \text{year2000} \]

\textit{radio} \land \textit{fogLamps} is a \textit{maximally ignorant} \quad H
Comparing criteria

\[ R = \text{FiatPanda} \land \text{radio} \land \text{fogLamps} \]
\[ O = \text{FiatPanda} \land \text{year2000} \]
\[ \text{radio} \land \text{fogLamps} \text{ is a } \text{maximally ignorant} \quad H \]
\[ T = \{ \text{bundleOffer} \Rightarrow \text{radio} \land \text{fogLamps} \land \text{alarm} \} \]
Comparing criteria

- \( R = \textit{FiatPanda} \land \textit{radio} \land \textit{fogLamps} \)
- \( O = \textit{FiatPanda} \land \textit{year2000} \)
- \( \textit{radio} \land \textit{fogLamps} \) is a maximally ignorant \( H \)
- \( T = \{ \textit{bundleOffer} \Rightarrow \textit{radio} \land \textit{fogLamps} \land \textit{alarm} \} \)
- \( \textit{bundleOffer} \) is a shortest \( H \)
Comparing criteria

\[ R = \text{FiatPanda} \land \text{radio} \land \text{fogLamps} \]
\[ O = \text{FiatPanda} \land \text{year2000} \]
\[ \text{radio} \land \text{fogLamps} \text{ is a maximally ignorant } H \]
\[ T = \{ \text{bundleOffer} \Rightarrow \text{radio} \land \text{fogLamps} \land \text{alarm} \} \]
\[ \text{bundleOffer is a shortest } H \]
\[ \text{neither solution is in the other set.} \]
Intermezzo

Abduction could formalize reasoning on missing information for P2P EC
Intermezzo

- Abduction could formalize reasoning on missing information for P2P EC
- what about conflicting information?
Gärdenfors [1988], among many others: Revise Knowledge $\mathcal{K}$ with new info $A$ by:

1. **contracting** $\mathcal{K}$ into $\mathcal{K}_{\neg A}$ such that $\mathcal{K}_{\neg A} \not\models \neg A$
2. **adding** $A$ to $\mathcal{K}_{\neg A}$

Intuition: contract the least
Contraction for P2P EC

Let $\mathcal{L}$ be a logic language

$R$ a request in $\mathcal{L}$

$O$ a possible offer for $R$ in $\mathcal{L}$

$\mathcal{T}$ be a domain ontology
Contraction for P2P EC

Let $\mathcal{L}$ be a logic language

$R$ a request in $\mathcal{L}$

$O$ a possible offer for $R$ in $\mathcal{L}$

$\mathcal{T}$ be a domain ontology

find a pair $\langle G, K \rangle$ ($G$ive up, $K$eep) such that
Let $\mathcal{L}$ be a logic language

$R$ a request in $\mathcal{L}$

$O$ a possible offer for $R$ in $\mathcal{L}$

$\mathcal{T}$ be a domain ontology

find a pair $\langle G, K \rangle$ (Give up, Keep) such that

$\mathcal{T} \models R \equiv G \land K$
Contraction for P2P EC

Let $\mathcal{L}$ be a logic language

$R$ a request in $\mathcal{L}$

$O$ a possible offer for $R$ in $\mathcal{L}$

$\mathcal{T}$ be a domain ontology

find a pair $\langle G, K \rangle$ (Give up, Keep) such that

$\mathcal{T} \models R \equiv G \land K$

$O \land K$ is satisfiable in $\mathcal{T}$
Let $\mathcal{L}$ be a logic language

$R$ a request in $\mathcal{L}$

$O$ a possible offer for $R$ in $\mathcal{L}$

$\mathcal{T}$ be a domain ontology

find a pair $\langle G, K \rangle$ (Give up, Keep) such that

$\mathcal{T} \models R \equiv G \land K$

$O \land K$ is satisfiable in $\mathcal{T}$

$\langle G, K \rangle$ is a contraction of $R$ w.r.t. $O$
Best contractions

Different criteria:

- **shortest** $G$ — fewer issues to give up
Best contractions

Different criteria:

- *shortest* $G$ — fewer issues to give up
- *maximally ignorant* $G$ — minimal consequences
Best contractions

Different criteria:

- **shortest** $G$ — fewer issues to give up
- **maximally ignorant** $G$ — minimal consequences
- maximal knowledge for $K$
Example

\[ R = flat \land (lift \lor firstFloor \lor secondFloor) \]
Example

\[ R = flat \land (lift \lor firstFloor \lor secondFloor) \]

“I want a flat which either has the lift, or it is a low floor”
Example

\[ R = \text{flat} \land (\text{lift} \lor \text{firstFloor} \lor \text{secondFloor}) \]

“\(I\) want a flat which either has the lift, or it is a low floor”

\[ O = \text{firstFloor} \land \text{lastFloor} \land \text{garden} \]
Example

\[ R = \text{flat} \land (\text{lift} \lor \text{firstFloor} \lor \text{secondFloor}) \]

“I want a flat which either has the lift, or it is a low floor”

\[ O = \text{firstFloor} \land \text{lastFloor} \land \text{garden} \]

\[ T = \left\{ \begin{array}{c} \text{firstFloor} \land \text{lastFloor} \equiv \text{house} \\ \text{flat} \equiv \neg \text{house} \end{array} \right\} \]
Example

\[ R = flat \land (lift \lor firstFloor \lor secondFloor) \]

“\text{I want a flat which either has the lift, or it is a low floor}”

\[ O = firstFloor \land lastFloor \land garden \]

\[ T = \left\{ \begin{array}{l}
firstFloor \land lastFloor \equiv house \\
flat \equiv \neg house
\end{array} \right\} \]

\[ G = flat \]
Example

\[ R = flat \land (lift \lor firstFloor \lor secondFloor) \]

“\text{I want a flat which either has the lift, or it is a low floor}”

\[ O = firstFloor \land lastFloor \land garden \]

\[ T = \begin{cases} 
  firstFloor \land lastFloor \equiv house \\
  flat \equiv \neg house 
\end{cases} \]

\[ G = flat \]

\[ K = lift \lor firstFloor \lor secondFloor \]
Logic-based ranking

suppose a buyer enters the marketplace with request $R$
Logic-based ranking

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$
  based on a pair of scores:
Logic-based ranking

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:
  - a score for a best contraction $\langle G, K \rangle$ of $R$ w.r.t. $O_i$
Logic-based ranking

suppose a buyer enters the marketplace with request \( R \)

the facilitator ranks all offers \( O_1, O_2, \ldots, O_n \)
based on a pair of scores:

a score for a best contraction \( \langle G, K \rangle \) of \( R \)
w.r.t. \( O_i \)

a score for a best abduction \( H \) on \( O \) w.r.t. \( K \)
Logic-based ranking

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:
  - a score for a best contraction $\langle G, K \rangle$ of $R$ w.r.t. $O_i$
  - a score for a best abduction $H$ on $O$ w.r.t. $K$
  - an explanation $G, K, H$ of the rank of each offer
Logic-based ranking

- Suppose a buyer enters the marketplace with request $R$.
- The facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:
  - A score for a best contraction $\langle G, K \rangle$ of $R$ w.r.t. $O_i$.
  - A score for a best abduction $H$ on $O$ w.r.t. $K$.
  - An explanation $G, K, H$ of the rank of each offer ← trust!
Alternatives to Belief Revision

Variable-strength preferences [Lukasiewicz & Schellhase KR-06]
Alternatives to Belief Revision

- Variable-strength *preferences* [Lukasiewicz & Schellhase KR-06]

- syntax: $(\alpha > \beta|\phi)[x]$
Alternatives to Belief Revision

- Variable-strength preferences [Lukasiewicz & Schellhase KR-06]

- Syntax: $(\alpha > \beta | \phi)[x]$

- Formula $\alpha$ is preferred to formula $\beta$ in the context $\phi$ with weight $x \in \mathbb{N}$
Negotiation

Second phase in a Bilateral Commercial Transaction:

1. Matchmaking (find counterpart)
2. *Negotiation* (agree/tradeoff details)
3. Exchange (goods, services, money)
Logic-based negotiation

- each agent puts utilities on formulas

e.g., \[
\begin{align*}
U_R(price2000) &= 2 \\
U_R(1YearGuarantee) &= 15
\end{align*}
\]
Logic-based negotiation

- each agent puts utilities on formulas

\[
\begin{aligned}
U_R(\text{price2000}) &= 2 \\
U_R(\text{1YearGuarantee}) &= 15
\end{aligned}
\]

e.g., \text{FiatPanda}

- some formulas are strict requirements
Logic-based negotiation

- Each agent puts utilities on formulas
  
  \[
  \begin{align*}
  U_R(price2000) & = 2 \\
  U_R(1YearGuarantee) & = 15
  \end{align*}
  \]

- Some formulas are strict requirements
  
  e.g., FiatPanda

- Additive utilities
  
  e.g.,
  
  \[
  U_R(price2000 \land 1YearGuarantee) = 2 + 15
  \]
# Example: buyer

## Utilities for $R$:

<table>
<thead>
<tr>
<th>formula</th>
<th>$U_R(\cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiatPanda</td>
<td>strict</td>
</tr>
<tr>
<td>fogLamps $\land$ radio</td>
<td>strict</td>
</tr>
<tr>
<td>price2000</td>
<td>2</td>
</tr>
<tr>
<td>price1000</td>
<td>5</td>
</tr>
<tr>
<td>1YearGuarantee</td>
<td>15</td>
</tr>
</tbody>
</table>
Example: seller

Utilities for $O$:

<table>
<thead>
<tr>
<th>formula</th>
<th>$U_O(\cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$FiatPanda$</td>
<td>strict</td>
</tr>
<tr>
<td>$1YearGuarantee \Rightarrow price2000$</td>
<td>strict</td>
</tr>
<tr>
<td>$price2000$</td>
<td>10</td>
</tr>
<tr>
<td>$price1000$</td>
<td>2</td>
</tr>
</tbody>
</table>
Agreements as models

utilities $U_O(m), U_R(m)$ of a model $m$ of $T$
Agreements as models

Utilities $U_O(m), U_R(m)$ of a model $m$ of $\mathcal{T}$

$= \sum$ utilities of satisfied formulas in $m$
Agreements as models

- utilities $U_O(m), U_R(m)$ of a model $m$ of $\mathcal{T}$
- $= \sum$ utilities of satisfied formulas in $m$
- search for optimal agreements:
Agreements as models

- Utilities \( U_O(m), U_R(m) \) of a model \( m \) of \( \mathcal{T} \)
- \( \mathcal{T} = \sum \) utilities of satisfied formulas in \( m \)

Search for optimal agreements:

- Max-sum: \( \max_m \{ U_O(m) + U_R(m) \} \)
  (welfare)
Agreements as models

- utilities $U_O(m), U_R(m)$ of a model $m$ of $\mathcal{T}$
- $= \sum$ utilities of satisfied formulas in $m$

search for optimal agreements:

- max-sum: $\max_m \{U_O(m) + U_R(m)\}$ (welfare)
- max-product: $\max_m \{U_O(m) \cdot U_R(m)\}$
Example, cntd.: agreement

<table>
<thead>
<tr>
<th>satisfied formulas</th>
<th>R</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FiatPanda</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>fogLamps ∧ radio</strong></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>1YearGuarantee</strong></td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><strong>price2000</strong></td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td><strong>1YearGuarantee ⇒ price2000</strong></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>total utilities</td>
<td>17</td>
<td>10</td>
</tr>
</tbody>
</table>
Preliminary results — see next ECAI-06

Integer Linear Programming can be used, also for max-product
Preliminary results — see next ECAI-06

- Integer Linear Programming can be used, also for max-product
- any other methods provably better?
Preliminary results — see next ECAI-06

- Integer Linear Programming can be used, also for max-product
- any other methods provably better?
- No: finding an optimal agreement is NPO-complete
Preliminary results — see next ECAI-06

• Integer Linear Programming can be used, also for max-product
• any other methods provably better?
• No: finding an optimal agreement is NPO-complete
  • tailored approximation algorithms unlikely to exist, unless APX = NPO
Rest of the talk — do you need a coffee?

1. ✓ P2P Electronic Commerce
2. ✓ Enabling technologies
3. ✓ General assumptions
4. ✓ Reasoning for Matchmaking
5. ✓ Reasoning for Negotiation
6. Languages and expressivity
7. What next?
Which language for P2P EC?

- Propositional
Which language for P2P EC?

- Propositional
  - useful only for theoretical purposes
Which language for P2P EC?

- Propositional
  - useful only for theoretical purposes
- Description Logics (DLs)
Which language for P2P EC?

- Propositional
  - useful only for theoretical purposes
- Description Logics (DLs)
  - OWL-DL *is* a DL
Which language for P2P EC?

- Propositional
  - useful only for theoretical purposes
- Description Logics (DLs)
  - OWL-DL *is* a DL
  - many papers already
Which language for P2P EC?

- Propositional
  - useful only for theoretical purposes

- Description Logics (DLs)
  - OWL-DL *is* a DL
  - many papers already

- why not Logic Programming?
Example, revisited — just an idea

\[ R = \text{FiatPanda, radio, fogLamps} \]

\[ O = \text{FiatPanda, year2000} \]

\[ T = \left\{ \begin{array}{l}
\text{radio} \leftarrow \text{bundleOffer}. \\
\text{fogLamps} \leftarrow \text{bundleOffer}. \\
\text{alarm} \leftarrow \text{bundleOffer}.
\end{array} \right\} \]

\[ \neg R \text{ can be derived from } T \cup \{O\} \text{ if bundleOffer is abducible} \]
$O, R$: conjunctions of atoms
$O, R$: conjunctions of atoms

$\mathcal{T}$: a logic program
ALP 4 P2P EC – can you read this?

- $O, R$: conjunctions of atoms
- $\mathcal{T}$: a logic program
- Find a set of abducibles $H$ such that $\mathcal{T} \cup \{O\} \cup H \vdash R$
$O, R$: conjunctions of atoms
$\mathcal{T}$: a logic program
Find a set of abducibles $H$ such that
$\mathcal{T} \cup \{O\} \cup H \vdash R$
representation & programming in one language
$O, R$: conjunctions of atoms

$\mathcal{T}$: a logic program

Find a set of abducibles $H$ such that

$\mathcal{T} \cup \{O\} \cup H \vdash R$

+ representation & programming in one language

+ enabling technologies exist (RuleML)
$O, \ R$: conjunctions of atoms

$\mathcal{T}$: a logic program

Find a set of abducibles $H$ such that

$\mathcal{T} \cup \{O\} \cup H \vdash R$

+ representation & programming in one language

+ enabling technologies exist (RuleML)

– limited expressivity
Future issues

- agents carry *both* an offer *and* a request
Future issues

- agents carry *both* an offer *and* a request
- “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
Future issues

- agents carry both an offer and a request
- “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
- Dating services
Future issues

- agents carry *both* an offer *and* a request
- “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
- Dating services
- specialized comparisons
Future issues

• agents carry both an offer and a request
  • “Award-winner Chinese calligrapher seeks flat in London” — Sunday Times, August 2002
• Dating services
• specialized comparisons
  • e.g., price, color, delivery time
Future issues

- agents carry *both* an offer *and* a request
  - “Award-winner Chinese calligrapher seeks flat in London” — Sunday Times, August 2002
- Dating services
- specialized comparisons
  - *e.g.*, price, color, delivery time
- epistemic statements
Future issues

- agents carry **both** an offer **and** a request
- “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
- Dating services
- specialized comparisons
  - *e.g.*, price, color, delivery time
- epistemic statements
  - “Best prices paid”
Future issues

- agents carry both an offer and a request
  - “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
- Dating services
- specialized comparisons
  - e.g., price, color, delivery time
- epistemic statements
  - “Best prices paid”
  - “smokers allowed”
Acknowledgements

All people at SisInfLab, Politecnico di Bari
But especially...

- Eugenio Di Sciascio
- Tommaso Di Noia
- Simona Colucci
- Azzurra Ragone
- ... among many others
An invitation — among many other conferences

next *ACM Symposium on Applied Computing* (SAC-2007)

track on Semantic-based Resource Discovery, Retrieval & Composition (SDRC)

papers welcome!