Towards a “Strategic” Web

Michael Rovatsos
(joint work with Alexandros Belesiotis, George Christelis, Matt Crosby, and Iain Wallace)

Centre of Intelligent Systems and their Applications
School of Informatics, University of Edinburgh
mrovatso@inf.ed.ac.uk
Background

Visit www.cisa.inf.ed.ac.uk/agents for details
Background

• What is special about agents? Interaction in a common environment
• To make agents intelligent and autonomous, we need to automate such interaction
• Interested in knowledge-based reasoning about interaction
• Reasoning about interaction is by definition practical reasoning
• Vision: given a specification of the interaction problem, automatically synthesise behaviour
Background

• We are interested in building systems, not only specifying them formally
• Rational agents need to synthesise action sequences to operate autonomously
• We want to tell them what to achieve, not how, abstraction desirable
• This suggests using knowledge representation techniques
• Planning is the interface between KR methods and practical reasoning
• But multiagent planning underdeveloped, no simple common framework
The “Strategic” Web

• Many interaction on the Web are strategic, i.e. involve potentially divergent views and objectives of users
• Currently, very little support for this on the Web (with exception of some eCommerce applications)
• Applications rely on hardcoded policies, large-scale data mining, or manual user intervention
• Vision: represent knowledge about interests of users to be able to reason about them
• Warning: Same dangers as Semantic Web (standards, burden of annotation, scalability issues, etc)
• But also opportunity to exploit what is really different about agents (different entities with different goals)
Examples

• Two examples, representative of different types of interaction planning problems:
  – A buyer-seller negotiation in an online shop (complex agent models, partial knowledge, large-scale player and action sets, vagueness)
  – A negotiation for sharing computational resources on a peer-to-peer network (complex resource constraints, simple actions but potentially complex sequencing)
Examples

**BUYER-SELLER**

B: I would like an art history book.
S: Good art history books range from $35-$55.
B: I would like something cheaper.
S: There’s “Art for Kids” at $15.
B: I want a book for adults.
S: There’s “Art History for Dummies” at $25.
B: Great, I’ll take that.

*(execution follows, including payment, delivery, etc)*

**PEER-TO-PEER**

P: I’d like to stream a music concert in high quality tomorrow night.
Q: Who will be performing?
P: It’s a “best-of” transmission from a festival.
Q: I don’t like watching concerts unless I know what bands are playing.
P: Could I still borrow your bandwidth?
Q: OK, if you grant me prioritised access to yours for seven days after that.

*(execution follows, including settings to preference in P2P system, actual streaming actions, etc)*

Towards a "Strategic" Web
The dialogue metaphor

• Examples deliberately looked like conversations, a simple, intuitive way of thinking about Strategic Web

• **Dialogue planning metaphor** covers synthesis, negotiation, and execution aspect
  – If communication actions are interpreted in a planning-based way, we should be able to plan them just like physical actions
  – But hard to decide about communication strategy before having synthesised collaborative plans
  – Actions planned for deception detection ahead of execution may affect suggested deals
The technology

Towards a "Strategic" Web
Challenges

• Languages for describing strategic interaction situations on the Web
• Tractable (approximate?) inference and plan synthesis algorithms
• Preference elicitation and content presentation techniques
• Human-centric & interdisciplinary approach required
Why not game theory?

• Game-theoretic methods very popular currently and address the problem of reasoning about interaction

• Information in real-world domains available in relational terms (e.g. on the Web), not enumerated state actions as assumed in game theory

• **Non-incremental**: unable to express how a game changes when we incrementally change background knowledge

• Knowledge-based methods might be useful in lifting overly restrictive assumptions (full rationality, perfect knowledge, etc)

• Intuition: many large-scale games might be actually “easier” than we think *(this is speculative)*
Previous work

• Three examples of our current work on knowledge-based reasoning about interaction:
  - Macro-level: Automated norm synthesis
  - Meso-level: Argumentation-based conflict resolution
  - Micro-level: Practical social reasoning architectures

• Address general multiagent systems problems:
  – Setting up social laws to avoid undesirable states
  – Exchanging information to align divergent views
  – Reasoning about others from an agent’s point of view

• From a general computer science point of view:
  – Designer-level specification of system constraints
  – Integration of distributed sources of data
  – Process-level view of environment behaviour
Automated norm synthesis in a planning environment

- Norms ensure global conflict states are never entered by prohibiting actions in certain states.
- At the same time agents’ private goals should remain achievable.
- Automated synthesis of such norms is NP-hard in enumerated state systems.
- Existing methods don’t exploit abstractions of propositional/first-order domain theories.
- Our method: find “detours” around conflict states by local search in generalised state spaces.
Automated norm synthesis

Iterated process of forward-backward search around conflict state specification:

- Not better than full state-space search in the worst case but often get lucky
- With simple additional pruning techniques search can often be cut down drastically
- Currently working on synthesising sanctions
Example

• Tunnel world example:

  Agents entering tunnels have to leave them out the opposite end immediately (so on entering tunnel, future crash not avoidable)

  Our algorithm solves this by computing a general norm
  \( \{ \text{at}_1(N), \text{at}_2(N'), \text{tunnel}(T), \text{conn}(N,T), \text{conn}(T,N')\}, \text{move}_1(N,T) \)  

  Note that we ignore extra cost caused to agent that has to take a detour to reach her goal when adhering to the norm
Argumentation-based conflict resolution in planning

• **Argumentation** is a method for determining the status of propositions in the presence of conflicting information.
• Different acceptability-based semantics and protocols that implement these.
• Rarely used for reasoning about action, our intuition is that this can be done more efficiently due to domain structure.
• Suggest framework for acceptable planning:
  A plan $P$ is acceptable wrt (potentially conflicting) knowledge bases $KB_1$ and $KB_2$
  iff $KB_1 \models P$ and $KB_2 \models P$
Argumentation-based conflict resolution

- Plan proposal generated by single agent (with any planner)
- Validation based on simple plan projection
- Dispute in case of disagreement, argumentation follows
- Ends in successful defence of initial proposal or rejection
- An alternative to generating one $P$ that works under both $KB$s
Example

- Robot gridworld domain

\[
\begin{aligned}
\text{PRO} & \quad \text{OPP} \\
2 & \quad 2 \\
1 & \quad 1, 2 \\
o & \quad t & \quad t \\
r & \quad o & \quad r \\
1 & \quad 2 & \quad 1, 2
\end{aligned}
\]

\[
\begin{aligned}
\text{At} & \quad \text{Pickup} \quad \Rightarrow \quad \text{Drop} \\
\text{PRO} & \quad \text{OPP} \\
\text{At} & \quad \text{Pickup} \\
\text{Proponent moves} & \quad \text{Opponent moves}
\end{aligned}
\]

(1): \( P \quad \Rightarrow \quad \text{Pickup} \quad \Rightarrow \quad \text{Drop} \)

(2): \( GNA \quad \text{At} (o, 2, 2) \)

(3): \( HE \quad \text{Pickup}, \text{Hold}(r, o), s_3 \)

(4): \( HI \quad \text{Have}(r, o) \)

(5): \( DA \quad \text{Pickup}, \text{At}(o, 1, 2), s_1 \)

(6): \( HI \quad \text{At}(o, 1, 2) \)

(7): \( \text{Belief} \)
Practical social reasoning architectures

- Practical reasoning architectures like BDI do not specifically consider social interaction.

- **Social reasoning** = reasoning about other agents and social mechanisms governing the system (i.e. hidden system properties).

- Assumption: any social reasoning mechanism can be formalised as a set of update rules regarding constraints concerning hidden system properties.

- **Expectation-Strategy-Behaviour (ESB)** architecture as a general computational framework.
The ESB framework

- **Expectations** express assumptions about other agents’ mental states or behaviours
- Their specification includes rules for how to update beliefs with relevant observations
- **Strategies** restrict the way potential future expectations are projected (think of a restricted expectation graph)
- **Behaviours** condition own behaviour (e.g. belief change at BDI level) on constraints verified against expectation graph
- Formal semantics, easily combined with state-of-the-art model-checkers
- An ESB engine can be easily combined with a normal BDI interpreter (in our implementation, Jason/AgentSpeak)
ESB reasoning engine

**ESB Specification**
- **Expectation Table**
  - Set of Expectation tuples
- **Strategy**
  - Selected pre-defined strategy
- **Behaviour Table**
  - Condition + "Action" pairs

**ESB Engine**
- **E-graph Generator**
  - Creates the expectation graph
- **S-Graph Generator**
  - Updates the strategy graph
- **Behaviour Condition Checker**

**BDI Interpreter**
- **Beliefs**
- **Belief Revision Function**

**Environment**

Towards a "Strategic" Web 21
Planning Games

• Examples illustrate use of knowledge-based methods for reasoning about interaction
• But so far not concerned with strategic interaction
• Currently trying to look at more general framework of strategic multiagent planning
• Why planning? At the frontline of what is possible in terms of scalability while maintaining “knowledge-level” flavour
• Introduce notion of coalition-planning game (reward for goal, cost for plan, no action = 0)
• Solution stable if no set of agents can increase utility by jointly adopting other plan
• Formally: plan $\pi$ stable for iff no plan $\pi$ exists for any subset $\Phi'$ of agents $\Phi$ such that $u_\phi(\pi') > u_\phi(\pi)$ for all $\phi$ in $\Phi'$
• Present an algorithm for computing stable plans, but complexity issues (enumeration of strategies necessary)
Interesting problems

Three general problems seem interesting:

• How to **compute acceptable plan** given a solution criterion (in particular adapting existing planning heuristics)

• How to **search plan space incrementally** for generating proposals during negotiation

• How to **use background knowledge** to guide plan recognition and optimal response generation
Examples

• Parcel delivery: the simplest (?) domain which raises interesting issues

  • Fundamental question: how can domain structure help here?
Conclusions

• Argued for “Strategic” Web as an interesting field for agent applications
• Personal view: automated reasoning about strategic interaction is key contribution of agent technology
• Long-term challenge, hard and risky
• Examples of previous work indicate practical reasoning algorithms are possible
• Current goal is to develop similar methods for settings of strategic interaction
• A lot of scope for doing things in a multiagent planning setting, very little previous work
• If this can be used to do strategic dialogue planning, this would provide a key building block of future Strategic Web technologies
Thank you. Questions?

Material based on
Christelis & MR @ AAMAS 2009
Belesiotis, MR & Rahwan @ ArgMAS 2009
Wallace & MR @ AAMAS 2009

Find out more/get involved at
http://www.cisa.inf.ed.ac.uk/agents