Intelligent Agents on the Web – Some Ideas and Challenges

Michael Rovatsos

Centre for Intelligent Systems and their Applications

British University in Dubai
24th May 2005
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- Examples
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems

- Examples
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- Examples
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
- **Open** systems:
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- Examples
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
- **Open** systems:
  - Interaction with components stemming from different people/organisations
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- Examples
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
- Open systems:
  - Interaction with components stemming from different people/organisations
  - Very hard to impose restrictions on others’ behaviour (potentially self-interested/malicious)
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- Examples
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
- Open systems:
  - Interaction with components stemming from different people/organisations
  - Very hard to impose restrictions on others’ behaviour (potentially self-interested/malicious)
  - Impossible to predict global behaviour of the system!
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- Examples
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
- **Open** systems:
  - Interaction with components stemming from different people/organisations
  - Very hard to impose restrictions on others’ behaviour (potentially self-interested/malicious)
  - Impossible to predict global behaviour of the system!
- Communication replaces direct control
Introduction

- Modern computer applications are increasingly moving towards open, decentralised systems
- **Examples**
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, peer-to-peer systems
- **Open** systems:
  - Interaction with components stemming from different people/organisations
  - Very hard to impose restrictions on others’ behaviour (potentially self-interested/malicious)
  - Impossible to predict global behaviour of the system!
- Communication replaces direct control
- A paradigm shift in systems development?
Introduction

- **Intelligent agents** provide a useful metaphor for such systems.
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
Introduction

- **Intelligent agents** provide a useful metaphor for such systems.
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
  - Build artificial systems that exhibit intelligent behaviour
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
  - Build artificial systems that exhibit intelligent behaviour
- In **Distributed Artificial Intelligence**, the focus is on interaction between agents situated in a common environment
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
  - Build artificial systems that exhibit intelligent behaviour
- In **Distributed Artificial Intelligence**, the focus is on interaction between agents situated in a common environment
- Two sub-fields:
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
  - Build artificial systems that exhibit intelligent behaviour
- In **Distributed Artificial Intelligence**, the focus is on interaction between agents situated in a common environment
- Two sub-fields:
  - Distributed Problem Solving (strictly cooperative)
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
  - Build artificial systems that exhibit intelligent behaviour
- In **Distributed Artificial Intelligence**, the focus is on interaction between agents situated in a common environment
- Two sub-fields:
  - Distributed Problem Solving (strictly cooperative)
  - Multiagent Systems (focus on coordination among self-interested agents)
Introduction

- **Intelligent agents** provide a useful metaphor for such systems
- Deeply rooted in the endeavour of Artificial Intelligence, which is to
  - Understand intelligence in natural systems (humans, animals)
  - Build artificial systems that exhibit intelligent behaviour
- In **Distributed Artificial Intelligence**, the focus is on interaction between agents situated in a common environment
- Two sub-fields:
  - Distributed Problem Solving (strictly cooperative)
  - Multiagent Systems (focus on coordination among self-interested agents)
- In this talk, I will discuss how multiagent systems can be used as a suitable technology for open systems using the Web as an example
Outline

Introduction

Agents and the Web

Agents & Multiagent Systems
The (Semantic) Web
The Interaction Perspective
Link Exchange Negotiations

Learning communication patterns

The ESB Architecture

Conclusions
Outline

Introduction

Agents and the Web

Agents & Multiagent Systems
The (Semantic) Web
The Interaction Perspective
Link Exchange Negotiations

Learning communication patterns

The ESB Architecture

Conclusions
What is an agent?

- Most widely accepted definition:
What is an agent?

- Most widely accepted definition:
What is an agent?

- Most widely accepted definition:

  An agent is anything that can perceive its environment (through its sensors) and act upon that environment (through its effectors)
What is an agent?

**Most widely accepted definition:**

An agent is anything that can perceive its environment (through its sensors) and act upon that environment (through its effectors)
What is an agent?

- Most widely accepted definition:

  An agent is anything that can perceive its environment (through its sensors) and act upon that environment (through its effectors).

- Autonomous agent:
What is an agent?

- Most widely accepted definition:
  
  An agent is anything that can perceive its environment (through its sensors) and act upon that environment (through its effectors)

- Autonomous agent:
What is an agent?

▶ Most widely accepted definition:

An agent is anything that can perceive its environment (through its sensors) and act upon that environment (through its effectors)

▶ Autonomous agent:

A computer system that is capable of independent (autonomous) action on behalf of its user or owner
Intelligent agents

- Intelligent agents are usually considered to be
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
  - Reactive (responsive to changes in environment)
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
  - Reactive (responsive to changes in environment)
  - Proactive (able to take initiative for action)
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
  - Reactive (responsive to changes in environment)
  - Proactive (able to take initiative for action)
  - Rational (goal-oriented, striving to optimise its decisions)
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
  - Reactive (responsive to changes in environment)
  - Proactive (able to take initiative for action)
  - Rational (goal-oriented, striving to optimise its decisions)
  - Socially capable (able to communicate and interact with others)
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
  - Reactive (responsive to changes in environment)
  - Proactive (able to take initiative for action)
  - Rational (goal-oriented, striving to optimise its decisions)
  - Socially capable (able to communicate and interact with others)

- (Optional) additional features:
Intelligent agents

- Intelligent agents are usually considered to be
  - Autonomous (capable of independent action)
  - Situated (embedded in an environment)
  - Reactive (responsive to changes in environment)
  - Proactive (able to take initiative for action)
  - Rational (goal-oriented, striving to optimise its decisions)
  - Socially capable (able to communicate and interact with others)

- (Optional) additional features:
  - adaptiveness, mobility, lifelike qualities, real-time behaviour, sensorimotor capabilities, etc.
Controversy

Autonomous, situated in an environment, proactive and “intelligent” (in a way), but is it an agent?
The Web

- The current Web landscape: A collection of files/documents
The Web

- The current Web landscape: A collection of files/documents
  - mostly text, some multimedia, some databases, some (simple) services
The Web

► The current Web landscape: A collection of files/documents
  ► mostly text, some multimedia, some databases, some (simple) services

► HTML: Modest compliance with standards (thanks to robustness of browsers)
The Web

- The current Web landscape: A collection of files/documents
  - mostly text, some multimedia, some databases, some (simple) services
- HTML: Modest compliance with standards (thanks to robustness of browsers)
- Hyperlinks: Annotated with text, sometimes barely understandable even for humans
The Web

- The current Web landscape: A collection of files/documents
  - mostly text, some multimedia, some databases, some (simple) services
- HTML: Modest compliance with standards (thanks to robustness of browsers)
- Hyperlinks: Annotated with text, sometimes barely understandable even for humans
- Capabilities:
The Web

- The current Web landscape: A collection of files/documents
  - mostly text, some multimedia, some databases, some (simple) services
- HTML: Modest compliance with standards (thanks to robustness of browsers)
- Hyperlinks: Annotated with text, sometimes barely understandable even for humans
- Capabilities:
  - Simple information retrieval (scalability?)
The Web

▶ The current Web landscape: A collection of files/documents
  ▶ mostly text, some multimedia, some databases, some (simple) services

▶ HTML: Modest compliance with standards (thanks to robustness of browsers)

▶ Hyperlinks: Annotated with text, sometimes barely understandable even for humans

▶ Capabilities:
  ▶ Simple information retrieval (scalability?)
  ▶ Fairly simple transactions/services (play chess, buy a book)
The Web

- The current Web landscape: A collection of files/documents
  - mostly text, some multimedia, some databases, some (simple) services
- HTML: Modest compliance with standards (thanks to robustness of browsers)
- Hyperlinks: Annotated with text, sometimes barely understandable even for humans
- Capabilities:
  - Simple information retrieval (scalability?)
  - Fairly simple transactions/services (play chess, buy a book)
- All the relevant data is (or will soon be) on the Web, but in a form suitable for human processing only (it seems)
The Problem

This is what my homepage looks like to a machine:

- name & picture
- job title, affiliation
- contact details
- research
- e-mail
- teaching
Example

- We would like the Web to be used for automating more complex tasks:
Example

- We would like the Web to be used for automating more complex tasks:
Example

- We would like the Web to be used for automating more complex tasks:

  Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?
Example

- We would like the Web to be used for automating more complex tasks:
  
  *Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?*

- How can my agent find/parse/extract garage’s free times?
Example

► We would like the Web to be used for automating more complex tasks:

Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?

► How can my agent find/parse/extract garage’s free times?

► Which of my appointments are critical/flexible? Even if I annotated entries, what if the garage’s timetable doesn’t have such a concept?
Example

➤ We would like the Web to be used for automating more complex tasks:

> Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?

➤ How can my agent find/parse/extract garage’s free times?

➤ Which of my appointments are critical/flexible? Even if I annotated entries, what if the garage’s timetable doesn’t have such a concept?

➤ Lots of constraints:
Example

- We would like the Web to be used for automating more complex tasks:
  
  *Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?*

- How can my agent find/parse/extract garage’s free times?

- Which of my appointments are critical/flexible? Even if I annotated entries, what if the garage’s timetable doesn’t have such a concept?

- Lots of constraints:
  - How long will it take to get to the garage?
Example

- We would like the Web to be used for automating more complex tasks:
  
  *Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?*

- How can my agent find/parse/extract garage’s free times?

- Which of my appointments are critical/flexible? Even if I annotated entries, what if the garage’s timetable doesn’t have such a concept?

- Lots of constraints:
  - How long will it take to get to the garage?
  - Would I pay extra if they come to collect the car?
Example

▶ We would like the Web to be used for automating more complex tasks:

\[\text{Why can’t my online calendar and bank account negotiate with my garage’s to arrange a mutually convenient time and price to repair my leaking tyre?}\]

▶ How can my agent find/parse/extract garage’s free times?

▶ Which of my appointments are critical/flexible? Even if I annotated entries, what if the garage’s timetable doesn’t have such a concept?

▶ Lots of constraints:
   ▶ How long will it take to get to the garage?
   ▶ Would I pay extra if they come to collect the car?
   ▶ Can they repair the door lock too?
The Semantic Web

▶ What is the Semantic Web?
The Semantic Web

What is the Semantic Web?
The Semantic Web

What is the Semantic Web?

The idea of representing Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations.
The Semantic Web

- What is the Semantic Web?
  
  The idea of representing Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations

- Semantic Web technologies:
The Semantic Web

- What is the Semantic Web?
  
  The idea of representing Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations

- Semantic Web technologies:
  
  - Explicit meta-data: try to capture the meaning of data by annotating it with information about the content
The Semantic Web

What is the Semantic Web?

*The idea of representing Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations*

Semantic Web technologies:

- Explicit meta-data: try to capture the meaning of data by annotating it with information about the content
- Ontologies: facilitate organisation/navigation & search, bridge gaps between terminologies
The Semantic Web

- What is the Semantic Web?
  
  *The idea of representing Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations*

- Semantic Web technologies:
  - Explicit meta-data: try to capture the meaning of data by annotating it with information about the content
  - Ontologies: facilitate organisation/navigation & search, bridge gaps between terminologies
  - Logic: reasoning about the meta-data using ontological knowledge
What is the Semantic Web?

The idea of representing Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations

Semantic Web technologies:

- Explicit meta-data: try to capture the meaning of data by annotating it with information about the content
- Ontologies: facilitate organisation/navigation & search, bridge gaps between terminologies
- Logic: reasoning about the meta-data using ontological knowledge
- Agents: the programs that are going to use all this
Semantic Web Technologies: The Layer Cake
The Semantic Web

- A lot of progress has been made as concerns basic Semantic Web technologies
The Semantic Web

- A lot of progress has been made as concerns basic Semantic Web technologies
  - Standardisation efforts (esp. SW languages)
The Semantic Web

- A lot of progress has been made as concerns basic Semantic Web technologies
  - Standardisation efforts (esp. SW languages)
  - Inference engines
The Semantic Web

- A lot of progress has been made as concerns basic Semantic Web technologies
  - Standardisation efforts (esp. SW languages)
  - Inference engines
  - Tools
The Semantic Web

- A lot of progress has been made as concerns basic Semantic Web technologies
  - Standardisation efforts (esp. SW languages)
  - Inference engines
  - Tools

- However, the interaction perspective has received fairly little attention so far
The Semantic Web

- A lot of progress has been made as concerns basic Semantic Web technologies
  - Standardisation efforts (esp. SW languages)
  - Inference engines
  - Tools

- However, the interaction perspective has received fairly little attention so far

- In other words: The data is (or will be) out there, but where are the agents that are going to use it?
An Example: Link Exchange Negotiations

Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.
An Example: Link Exchange Negotiations

- Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.

- Automated inspection of other sites + Knowledge about own preferences (i.e. those of one’s owner) = Assessment of own stance of opinions expressed in other sites.
An Example: Link Exchange Negotiations

- Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.
- Automated inspection of other sites + Knowledge about own preferences (i.e. those of one’s owner) = Assessment of own stance of opinions expressed in other sites.
- Goal of each Web site owner (and his agent): Maximal dissemination of one’s own opinion.
An Example: Link Exchange Negotiations

- Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.
- Automated inspection of other sites + Knowledge about own preferences (i.e. those of one’s owner) = Assessment of own stance of opinions expressed in other sites.
- Goal of each Web site owner (and his agent): Maximal dissemination of one’s own opinion.
- This can be achieved by:
An Example: Link Exchange Negotiations

- Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.
- Automated inspection of other sites + Knowledge about own preferences (i.e. those of one’s owner) = Assessment of own stance of opinions expressed in other sites.
- Goal of each Web site owner (and his agent): Maximal dissemination of one’s own opinion.
- This can be achieved by:
  - Maximising the popularity of one’s own site.
An Example: Link Exchange Negotiations

- Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.

- Automated inspection of other sites + Knowledge about own preferences (i.e. those of one’s owner) = Assessment of own stance of opinions expressed in other sites.

- Goal of each Web site owner (and his agent): Maximal dissemination of one’s own opinion.

- This can be achieved by:
  - Maximising the popularity of one’s own site.
  - Increasing the popularity of sites that express similar opinions.
An Example: Link Exchange Negotiations

- Imagine agents representing Web sites are able to conduct inference about the content of other pages provided using Semantic Web methods.

- Automated inspection of other sites + Knowledge about own preferences (i.e. those of one’s owner) = Assessment of own stance of opinions expressed in other sites.

- Goal of each Web site owner (and his agent): Maximal dissemination of one’s own opinion.

- This can be achieved by:
  - Maximising the popularity of one’s own site
  - Increasing the popularity of sites that express similar opinions
  - Decrease the popularity of sites with unfavourable opinions
An Example: Link Exchange Negotiations

- Traffic provides a measure for popularity, and is affected by links between sites
An Example: Link Exchange Negotiations

- Traffic provides a measure for popularity, and is affected by links between sites.
- Links are weighted with numerical "ratings" expressing opinion source site has of target site.
An Example: Link Exchange Negotiations

- Traffic provides a measure for popularity, and is affected by links between sites.
- Links are weighted with numerical "ratings" expressing opinion source site has of target site.
  - In a more advanced system, these would correspond to comments such as "Click here for my favourite site on topic X"
An Example: Link Exchange Negotiations

- Traffic provides a measure for popularity, and is affected by links between sites.
- Links are weighted with numerical “ratings” expressing opinion source site has of target site.
  - In a more advanced system, these would correspond to comments such as “Click here for my favourite site on topic X”.
- Of course, the displayed ratings (actual link weights) can differ from the (private) actual ratings.
An Example: Link Exchange Negotiations

- Traffic provides a measure for popularity, and is affected by links between sites.
- Links are weighted with numerical “ratings” expressing opinion source site has of target site.
  - In a more advanced system, these would correspond to comments such as “Click here for my favourite site on topic X”.
- Of course, the displayed ratings (actual link weights) can differ from the (private) actual ratings.
- Agent goal: maximise opinion dissemination (in terms of some utility measure) through negotiation with other agent about link exchange.
An Example: Link Exchange Negotiations

- Traffic provides a measure for popularity, and is affected by links between sites.
- Links are weighted with numerical “ratings” expressing opinion source site has of target site.
  - In a more advanced system, these would correspond to comments such as “Click here for my favourite site on topic X”.
- Of course, the displayed ratings (actual link weights) can differ from the (private) actual ratings.
- Agent goal: maximise opinion dissemination (in terms of some utility measure) through negotiation with other agent about link exchange.
- System goal: increase linkage transparency on the WWW.
The LIESON System
The LIESON System

- Utility model with interesting properties
The LIESON System

- Utility model with interesting properties
  - No linkage, “full positive” linkage, “full negative” linkage lead to low utility or all agents
The LIESON System

- Utility model with interesting properties
  - No linkage, “full positive” linkage, “full negative” linkage lead to low utility or all agents
  - Full “honest” linkage yields lower utility than “politically correct” linkage (omitting negative links)
The LIESON System

- Utility model with interesting properties
  - No linkage, “full positive” linkage, “full negative” linkage lead to low utility or all agents
  - Full “honest” linkage yields lower utility than “politically correct” linkage (omitting negative links)
- Highly boundedly rational agents!
The LIESON System

- Utility model with interesting properties
  - No linkage, “full positive” linkage, “full negative” linkage lead to low utility or all agents
  - Full “honest” linkage yields lower utility than “politically correct” linkage (omitting negative links)
- Highly boundedly rational agents!
- Main problem: in an open system, it is unclear whether agents will fulfill one’s requests
The LIESON System

- Utility model with interesting properties
  - No linkage, “full positive” linkage, “full negative” linkage lead to low utility or all agents
  - Full “honest” linkage yields lower utility than “politically correct” linkage (omitting negative links)
- Highly boundedly rational agents!
- Main problem: in an open system, it is unclear whether agents will fulfill one’s requests
- Objective: develop methods to learn strategic application of negotiation strategies
The LIESON System

- Utility model with interesting properties
  - No linkage, “full positive” linkage, “full negative” linkage lead to low utility or all agents
  - Full “honest” linkage yields lower utility than “politically correct” linkage (omitting negative links)
- Highly boundedly rational agents!
- Main problem: in an open system, it is unclear whether agents will fulfill one’s requests
- Objective: develop methods to learn strategic application of negotiation strategies
- Two levels of complexity: proposal-based/argumentation-based negotiation
The Interaction Frames Approach

- Goal: learn patterns of agent conversations from experience and apply them strategically in one’s own interactions
The Interaction Frames Approach

- Goal: learn patterns of agent conversations from experience and apply them strategically in one’s own interactions
- Each pattern (interaction frame) consists of
The Interaction Frames Approach

- Goal: learn patterns of agent conversations from experience and apply them strategically in one’s own interactions
- Each pattern (interaction frame) consists of
  - a sequence of message patterns (speech-act like, augmented with variables)
The Interaction Frames Approach

- Goal: learn patterns of agent conversations from experience and apply them strategically in one’s own interactions
- Each pattern (interaction frame) consists of
  - a sequence of message patterns (speech-act like, augmented with variables)
  - pairs of logical conditions and variable substitutions
The Interaction Frames Approach

- Goal: learn patterns of agent conversations from experience and apply them strategically in one’s own interactions
- Each pattern (interaction frame) consists of
  - a sequence of message patterns (speech-act like, augmented with variables)
  - pairs of logical conditions and variable substitutions
  - occurrence counters representing previous enactments
The Interaction Frames Approach

- Goal: learn patterns of agent conversations from experience and apply them strategically in one’s own interactions
- Each pattern (interaction frame) consists of
  - a sequence of message patterns (speech-act like, augmented with variables)
  - pairs of logical conditions and variable substitutions
  - occurrence counters representing previous enactments
- Combine hierarchical reinforcement learning methods, case-based reasoning and clustering techniques to learn “framing”, i.e. strategic use of frames
An example

\[ F = \left\langle \left\langle \begin{array}{c}
5 \rightarrow \text{request}(A_1, A_2, X) \\
3 \rightarrow \text{accept}(A_2, A_1, X) \\
2 \rightarrow \text{confirm}(A_1, A_2, X) \\
2 \rightarrow \text{do}(A_2, X)
\end{array} \right\rangle, \\
\left\langle \{\text{self}(A_1), \text{other}(A_2), \text{can}(A_1, \text{do}(A_1, X))\}, \\
\{\text{agent}(A_1), \text{agent}(A_2), \text{action}(X)\}\right\rangle, \\
\left\langle 4 \rightarrow \langle [A_1/\text{agent}_1], [A_2/\text{agent}_2] \rangle, \\
1 \rightarrow \langle [A_1/\text{agent}_3], [A_2/\text{agent}_1], [X/\text{deliver goods}] \rangle \rangle \right\rangle \]
Proposal-based negotiation

\[ F_1 = \langle \langle 0 \rightarrow \text{request}(A, B, X) 0 \rightarrow \text{accept}(B, A, X) 0 \rightarrow \text{confirm}(A, B, X) 0 \rightarrow \text{do}(B, X) \rangle, \]
\[ \langle \text{can}(B, X)@3, \text{effects}(X)@4 \rangle \rangle \]

\[ F_2 = \langle \langle 0 \rightarrow \text{request}(A, B, X) 0 \rightarrow \text{propose}(B, A, Y) 0 \rightarrow \text{accept}(A, B, Y) 0 \rightarrow \text{do}(B, Y) \rangle, \]
\[ \langle \{\text{can}(B, Y)@3, \text{effects}(Y)@4\} \rangle \]
\[ \langle 0 \rightarrow \langle \rangle \rangle \rangle \]

\[ F_3 = \langle \langle 0 \rightarrow \text{request}(A, B, X) 0 \rightarrow \text{propose}–\text{also}(B, A, Y) 0 \rightarrow \text{accept}(A, B, Y) \]
\[ 0 \rightarrow \text{do}(B, X) 0 \rightarrow \text{do}(A, Y) \rangle, \]
\[ \langle \{\text{can}(B, X)@3, \text{effects}(X)@4, \text{can}(A, Y)@4, \text{effects}(Y)@5\} \rangle \]
\[ \langle 0 \rightarrow \langle \rangle \rangle \rangle \]
Interest-based Negotiation (IBN)

- A special kind of argumentation-based negotiation
Interest-based Negotiation (IBN)

- A special kind of argumentation-based negotiation
- As opposed to proposal-based negotiation, IBN allows agents to
  - obtain information about others’ beliefs and goals
  - point at others’ misconceptions
  - identify/suggest alternatives
Interest-based Negotiation (IBN)

- A special kind of argumentation-based negotiation
- As opposed to proposal-based negotiation, IBN allows agents to
  - obtain information about others’ beliefs and goals
  - point at others’ misconceptions
  - identify/suggest alternatives
- Our goal: not performance improvement, but coping with more complex communication “regime”
Interest-based Negotiation (IBN)

- A special kind of argumentation-based negotiation
- As opposed to proposal-based negotiation, IBN allows agents to
  - obtain information about others’ beliefs and goals
  - point at others’ misconceptions
  - identify/suggest alternatives
- Our goal: not performance improvement, but coping with more complex communication “regime”
- Approach due to Rahwan et al.
IBN – Dialogue model
IBN frames – Example

\[ F_{AGM} = \left\langle \langle 0 \rightarrow \text{request}(A, B, X) \rightarrow 0 \rightarrow \text{ask-reason}(B, A, \text{request}(X)) \rightarrow 0 \rightarrow \right. \right. \]
\[ \text{inform-goal}(A, B, G) \rightarrow \]
\[ \text{attack-goal}(B, A, \text{alternative-action}(Y)) \rightarrow \]
\[ 0 \rightarrow \text{concede}(A, B, Y) \rightarrow 0 \rightarrow \text{do}(B, Y) \rangle, \]
\[ \left\langle \{ \text{can}(B, X), \text{goal}(A, G), \text{achieves}(X, G), \text{achieves}(Y, G), \right. \]
\[ X \neq Y, \text{can}(B, Y) @5, \text{effects}(Y) @6 \} \rangle, \left\langle 0 \rightarrow \langle \rangle \right. \right. \rangle \]
Without Frame Learning

Agent performance

Agent utility

Simulation rounds

average
minimum
maximum
lower benchmark
upper benchmark
With Frame Learning
Outline

Introduction

Agents and the Web

Learning communication patterns

The ESB Architecture

Conclusions
Outline

Introduction

Agents and the Web

Learning communication patterns

The ESB Architecture

Conclusions
The ESB Architecture

- Expectation-Strategy-Behaviour architecture
The ESB Architecture

- Expectation-Strategy-Behaviour architecture
- Generalisation of ideas of interaction frames approach
The ESB Architecture

- Expectation-Strategy-Behaviour architecture
- Generalisation of ideas of interaction frames approach
- Key ideas:
The ESB Architecture

- Expectation-Strategy-Behaviour architecture
- Generalisation of ideas of interaction frames approach
- Key ideas:
  - Models of agents’ interaction behaviour are stored as expectations and updated with new observations
The ESB Architecture

- Expectation-Strategy-Behaviour architecture
- Generalisation of ideas of interaction frames approach
- Key ideas:
  - Models of agents’ interaction behaviour are stored as expectations and updated with new observations
  - Set of current expectations creates a strategy space
Expectation-Strategy-Behaviour architecture

Generalisation of ideas of interaction frames approach

Key ideas:
- Models of agents’ interaction behaviour are stored as expectations and updated with new observations
- Set of current expectations creates a strategy space
- Own behaviour chosen from these strategies in accordance with agent’s goals
The ESB Architecture

- Expectation-Strategy-Behaviour architecture
- Generalisation of ideas of interaction frames approach
- Key ideas:
  - Models of agents’ interaction behaviour are stored as expectations and updated with new observations
  - Set of current expectations creates a strategy space
  - Own behaviour chosen from these strategies in accordance with agent’s goals
- Concept of expectation used to bridge gap between cognitive and social system layer
The ESB Feedback Loop

- Expectations generate strategies, these generate behaviours, and the observation of these behaviours leads to new expectations.
The ESB Feedback Loop

- Expectations generate strategies, these generate behaviours, and the observation of these behaviours leads to new expectations.

- Agent-level (cognitive) vs. system-level (social) views (managing one’s own interactions versus controlling open systems)
The ESB Feedback Loop

- Expectations generate strategies, these generate behaviours, and the observation of these behaviours leads to new expectations.
- Agent-level (cognitive) vs. system-level (social) views (managing one’s own interactions versus controlling open systems).
- A closer look reveals that this nothing but a learning loop for interaction learning.
Interaction Frames and ESB

- The framing mechanism represents an expectation processing mechanism
Interaction Frames and ESB

- The framing mechanism represents an expectation processing mechanism
  - Based on probabilistic model of communicative behaviour and utility-relevant actions
Interaction Frames and ESB

- The framing mechanism represents an expectation processing mechanism
  - Based on probabilistic model of communicative behaviour and utility-relevant actions
  - Scope of prediction: current communicative encounter (conversation)
Interaction Frames and ESB

- The framing mechanism represents an expectation processing mechanism
  - Based on probabilistic model of communicative behaviour and utility-relevant actions
  - Scope of prediction: current communicative encounter (conversation)
  - Expectations will be adapted according to observed behaviour
Interaction Frames and ESB

- The framing mechanism represents an expectation processing mechanism
  - Based on probabilistic model of communicative behaviour and utility-relevant actions
  - Scope of prediction: current communicative encounter (conversation)
  - Expectations will be adapted according to observed behaviour
  - “Second-order” effect of own behaviour taken into account (heuristics for trading off long-term reliability of frames vs. short-term utility maximisation)
Interaction Frames and ESB

- The framing mechanism represents an expectation processing mechanism
  - Based on probabilistic model of communicative behaviour and utility-relevant actions
  - Scope of prediction: current communicative encounter (conversation)
  - Expectations will be adapted according to observed behaviour
  - “Second-order” effect of own behaviour taken into account (heuristics for trading off long-term reliability of frames vs. short-term utility maximisation)

- Successfully applied in complex multiagent negotiation scenarios ➔ a good example for applying the ESB principles
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems

- **Objectivist**: impose some kind of deontic apparatus on the system to regulate agent behaviour
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems

- **Objectivist**: impose some kind of deontic apparatus on the system to regulate agent behaviour
  - Methods abound: commitments and conventions, norms, roles, deontic logics, organisational approaches, electronic institutions
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems

- **Objectivist**: impose some kind of deontic apparatus on the system to regulate agent behaviour
  - Methods abound: commitments and conventions, norms, roles, deontic logics, organisational approaches, electronic institutions
  - Problem: no unifying model, no grounding in agent cognition
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems

- **Objectivist**: impose some kind of deontic apparatus on the system to regulate agent behaviour
  - Methods abound: commitments and conventions, norms, roles, deontic logics, organisational approaches, electronic institutions
  - Problem: no unifying model, no grounding in agent cognition

- **Rationalistic**: devise interaction mechanisms such that system objectives are achieved despite agents’ self-interest
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems

- **Objectivist**: impose some kind of deontic apparatus on the system to regulate agent behaviour
  - Methods abound: commitments and conventions, norms, roles, deontic logics, organisational approaches, electronic institutions
  - Problem: no unifying model, no grounding in agent cognition

- **Rationalistic**: devise interaction mechanisms such that system objectives are achieved despite agents’ self-interest
  - Examples: game-theoretic approaches (mechanism design, etc.)
Unifying Existing Approaches in ESB

- **Mentalistic**: assume a model of mental states of other agents (so that behaviour can essentially be fully predicted)
  - Example: Mentalistic ACL semantics (e.g. in FIPA-ACL)
  - Problem: Not feasible in open systems

- **Objectivist**: impose some kind of deontic apparatus on the system to regulate agent behaviour
  - Methods abound: commitments and conventions, norms, roles, deontic logics, organisational approaches, electronic institutions
  - Problem: no unifying model, no grounding in agent cognition

- **Rationalistic**: devise interaction mechanisms such that system objectives are achieved despite agents’ self-interest
  - Examples: game-theoretic approaches (mechanism design, etc.)
  - Problem: simplification of interaction mechanisms to guarantee properties, “worst-case reasoning”
Challenges

- Improve our understanding of expectation-based systems
Challenges

- Improve our understanding of expectation-based systems
- Develop appropriate representations (rule-based, probabilistic, deontic etc.) and decision-making algorithms
Challenges

- Improve our understanding of expectation-based systems
- Develop appropriate representations (rule-based, probabilistic, deontic etc.) and decision-making algorithms
- Develop evaluation criteria for such architectures (are there “stable” sets of expectations that ensure smooth interaction in the system?)
Challenges

- Improve our understanding of expectation-based systems
- Develop appropriate representations (rule-based, probabilistic, deontic etc.) and decision-making algorithms
- Develop evaluation criteria for such architectures (are there “stable” sets of expectations that ensure smooth interaction in the system?)
- Map existing approaches to a common “ESB language” to compare (and combine?) them
Challenges

- Improve our understanding of expectation-based systems
- Develop appropriate representations (rule-based, probabilistic, deontic etc.) and decision-making algorithms
- Develop evaluation criteria for such architectures (are there “stable” sets of expectations that ensure smooth interaction in the system?)
- Map existing approaches to a common “ESB language” to compare (and combine?) them
- Apply these methods to the development of open systems in real-world applications
Challenges

- Improve our understanding of expectation-based systems
- Develop appropriate representations (rule-based, probabilistic, deontic etc.) and decision-making algorithms
- Develop evaluation criteria for such architectures (are there “stable” sets of expectations that ensure smooth interaction in the system?)
- Map existing approaches to a common “ESB language” to compare (and combine?) them
- Apply these methods to the development of open systems in real-world applications
- Vision: Semantic Web ➔ Strategic Web
The End

Thank you for your attention!