Autonomy vs. Uncertainty:
Why agents are different and what we can do about it

Michael Rovatsos

Centre for Intelligent Systems and their Applications

Informatics Jamboree 2005
25th April 2005
Introduction

- Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems
Introduction

- Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems.
- From a traditional computer science/engineering point of view, the AI approach is seen to offer quite some advantages.
Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems.

From a traditional computer science/engineering point of view the AI approach is seen to offer quite some advantages:

- Developing heuristic approaches for hard problems.
Introduction

- Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems.
- From a traditional computer science/engineering point of view the AI approach is seen to offer quite some advantages:
  - Developing heuristic approaches for hard problems
  - Coping with underspecified, poorly understood domains
Introduction

- Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems.
- From a traditional computer science/engineering point of view, the AI approach is seen to offer quite some advantages:
  - Developing heuristic approaches for hard problems
  - Coping with underspecified, poorly understood domains
  - Anthropocentric (since anthropomorphic) design
Introduction

Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems.

From a traditional computer science/engineering point of view, the AI approach is seen to offer quite some advantages:

- Developing heuristic approaches for hard problems
- Coping with underspecified, poorly understood domains
- Anthropocentric (since anthropomorphous) design
- Dealing with uncertainty in the environment
Introduction

- Artificial Intelligence (AI) aims to understand natural intelligence and to replicate intelligent behaviour in artificial (mostly computational) systems.
- From a traditional computer science/engineering point of view the AI approach is seen to offer quite some advantages:
  - Developing heuristic approaches for hard problems
  - Coping with underspecified, poorly understood domains
  - Anthropocentric (since anthropomorphous) design
  - Dealing with uncertainty in the environment
- In recent years building “intelligent agents” has become one of the main concerns of AI research.
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus)
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus)
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) autonomous, reactive & proactive, socially capable computational entities
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) autonomous, reactive & proactive, socially capable computational entities
- No agreed definition (‘agents’ = AI?/agent = thermostat?)
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) *autonomous, reactive & proactive, socially capable computational entities*
- No agreed definition (“agents” = AI? / agent = thermostat?)
- Too broad for a well-defined research area (from RoboCup to electronic auctions via intelligent user interfaces to agent-oriented software engineering)
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) *autonomous, reactive & proactive, socially capable computational entities*
- No agreed definition (“agents”=AI?/agent=thermostat?)
- Too broad for a well-defined research area (from RoboCup to electronic auctions via intelligent user interfaces to agent-oriented software engineering)
- Lots of criticism, quite some of it is justified
Agents are considered to be (according to a rough consensus) autonomous, reactive & proactive, socially capable computational entities.

- No agreed definition ("agents" = AI?/agent = thermostat?)
- Too broad for a well-defined research area (from RoboCup to electronic auctions via intelligent user interfaces to agent-oriented software engineering)
- Lots of criticism, quite some of it is justified
- Is it all just a hype that will soon pass?
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) *autonomous, reactive & proactive, socially capable computational entities*
- No agreed definition ("agents" = AI?/agent = thermostat?)
- Too broad for a well-defined research area (from RoboCup to electronic auctions via intelligent user interfaces to agent-oriented software engineering)
- Lots of criticism, quite some of it is justified
- Is it all just a hype that will soon pass?
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) *autonomous, reactive & proactive, socially capable computational entities*
- No agreed definition ("agents" = AI?/agent = thermostat?)
- Too broad for a well-defined research area (from RoboCup to electronic auctions via intelligent user interfaces to agent-oriented software engineering)
- Lots of criticism, quite some of it is justified
- Is it all just a hype that will soon pass? Maybe, but there still is hope!
Intelligent Agents & Multiagent Systems

- Agents are considered to be (according to a rough consensus) *autonomous, reactive & proactive, socially capable computational entities*
- No agreed definition ("agents" = AI?/agent = thermostat?)
- Too broad for a well-defined research area (from RoboCup to electronic auctions via intelligent user interfaces to agent-oriented software engineering)
- Lots of criticism, quite some of it is justified
- Is it all just a hype that will soon pass? Maybe, but there still is hope! In this talk, I will try to explain why ...
An Alternative Definition

- Instead of debating true nature of agents, consider the following definition:
An Alternative Definition

▶ Instead of debating true nature of agents, consider the following definition:
An Alternative Definition

Instead of debating true nature of agents, consider the following definition:

An agent is a program that interacts with other programs representing different people/organisations in a common computational environment.
An Alternative Definition

► Instead of debating true nature of agents, consider the following definition:

An agent is a program that interacts with other programs representing different people/organisations in a common computational environment

► Replaces (vague, philosophical) notion of autonomy by a simple criterion emphasising the observer perspective
An Alternative Definition

- Instead of debating the true nature of agents, consider the following definition:

  *An agent is a program that interacts with other programs representing different people/organisations in a common computational environment*

- Replaces (vague, philosophical) notion of autonomy by a simple criterion emphasising the **observer** perspective

- Justifies distinction between agents and “ordinary” programs (encapsulation of **purpose** of software rather than its functionality)
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
- Open systems:
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
- Open systems:
  - Changing populations of heterogeneous, opaque agents
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
- Open systems:
  - Changing populations of heterogeneous, opaque agents
  - Potentially self-interested/malicious
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
- Open systems:
  - Changing populations of heterogeneous, opaque agents
  - Potentially self-interested/malicious
  - Very hard to impose restrictions on agent behaviour
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
- Open systems:
  - Changing populations of heterogeneous, opaque agents
  - Potentially self-interested/malicious
  - Very hard to impose restrictions on agent behaviour
  - Impossible to predict global behaviour of the system!
Autonomy vs. Uncertainty

- Fundamental distinction between uncertainty (of a passive environment) and agent autonomy
- Communication replaces direct (physical) manipulation
- Influence exerted on others depends on their expectations
  - “A rock doesn’t care about which robot is trying to move it”
- **Open** systems:
  - Changing populations of heterogeneous, opaque agents
  - Potentially self-interested/malicious
  - Very hard to impose restrictions on agent behaviour
  - Impossible to predict global behaviour of the system!
- In a sense, autonomy is dual to openness
Application Context

- Modern computer applications are increasingly moving towards open systems
Application Context

- Modern computer applications are increasingly moving towards open systems
- Example application areas:
Application Context

- Modern computer applications are increasingly moving towards open systems
- Example application areas:
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, P2P computing
Application Context

- Modern computer applications are increasingly moving towards open systems
- Example application areas:
  - eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, P2P computing
- This is true regardless of our highbrow academic theories of agents, it is happening in the real world!
Modern computer applications are increasingly moving towards open systems

Example application areas:

- eCommerce, Semantic Web, Web Services, Grid computing, mobile/ubiquitous computing, P2P computing

This is true regardless of our highbrow academic theories of agents, it is happening in the real world!

We need methods to deal with this kind of open systems

focus of my research
Outline

Introduction

A framework for expectation-based architectures

Strategic learning of communication patterns

Conclusions
Introduction

A framework for expectation-based architectures

Strategic learning of communication patterns

Conclusions
The ESB Architecture

- Expectation-Strategy-Behaviour
The ESB Architecture

- Expectation-Strategy-Behaviour
- Key ideas:
The ESB Architecture

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

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

- Expectation-Strategy-Behaviour
- 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
- 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 Architecture

- Expectation-Strategy-Behaviour
- 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
- Suitable for integration with the Belief-Desire-Intention (BDI) architecture
Expectations

Definition:

*An expectation is a conditional prediction whose fulfillment will be verified and reacted upon.*
Expectations

Definition:

An expectation is a conditional prediction whose fulfillment will be verified and reacted upon.

Semi-formal description:

We write $(\text{EXP} \ a \ C \ E \ \varphi \ \rho^+ \ \rho^-)$ iff agent $a$ expects $E$ to hold true under condition $C$, and is going to verify this using test $\varphi$. If the expectation is fulfilled he will react with $\rho^+$, otherwise with $\rho^-$. 
Expectations

Definition:

An expectation is a conditional prediction whose fulfillment will be verified and reacted upon.

Semi-formal description:

We write \((\text{EXP} \ a \ C \ E \ \varphi \ \rho^+ \ \rho^-)\) iff agent a expects E to hold true under condition C, and is going to verify this using test \(\varphi\). If the expectation is fulfilled he will react with \(\rho^+\), otherwise with \(\rho^-\).

Example:

\[(\text{EXP} \ A \ a \ (\text{DO} \ A \ \text{request}(A, B, X)) \ (\text{DO} \ B \ X) \ \text{Done}(X) \ \text{nil} \ \text{retract})\]
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?
- Expectations can be
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?
- Expectations can be
  - adaptive (and hence grounded in observation)
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?

- Expectations can be
  - adaptive (and hence grounded in observation)
  - self-referential (which – unlike normal belief – permits agents to change them themselves)
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?

- Expectations can be
  - adaptive (and hence grounded in observation)
  - self-referential (which – unlike normal belief – permits agents to change them themselves)
  - recursive (expectations towards the reasoning agent herself)
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?

- Expectations can be
  - adaptive (and hence grounded in observation)
  - self-referential (which – unlike normal belief – permits agents to change them themselves)
  - recursive (expectations towards the reasoning agent herself)
  - generalised (valid for whole sets of agents/actions, especially in the case of communicative expectations)
Expectations

- Essentially a set of expectations defines a belief revision mechanism, why should this be useful for practical social reasoning?

- Expectations can be
  - adaptive (and hence grounded in observation)
  - self-referential (which – unlike normal belief – permits agents to change them themselves)
  - recursive (expectations towards the reasoning agent herself)
  - generalised (valid for whole sets of agents/actions, especially in the case of communicative expectations)

- This makes them essential for reasoning about open systems!
Strategies

- Any set of expectations implicitly defines a **strategy space**
Strategies

- Any set of expectations implicitly defines a **strategy space**
- Results from space of actions that will lead to (non-)fulfillment of verification conditions
Strategies

- Any set of expectations implicitly defines a **strategy space**
- Results from space of actions that will lead to (non-)fulfillment of verification conditions
- Strategies concerns others’ actions as much as one’s own
Strategies

- Any set of expectations implicitly defines a **strategy space**
- Results from space of actions that will lead to (non-)fulfillment of verification conditions
- Strategies concerns others’ actions as much as one’s own
- Not all different action(s) (sequences) are different strategies, effect on expectations is what matters
Strategies

- Any set of expectations implicitly defines a **strategy space**
- Results from space of actions that will lead to (non-)fulfillment of verification conditions
- Strategies concern others’ actions as much as one’s own
- Not all different action(s) (sequences) are different strategies, effect on expectations is what matters
- Take potential effects on expectations into consideration
Strategies

- Any set of expectations implicitly defines a **strategy space**
- Results from space of actions that will lead to (non-)fulfillment of verification conditions
- Strategies concerns others’ actions as much as one’s own
- Not all different action(s) (sequences) are different strategies, effect on expectations is what matters
- Take potential effects on expectations into consideration
- Strategies define the “vocabulary of behaviours” that may affect expectations so that an assessment of the desirability of these behaviours can follow
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
- As far as own strategies are concerned, the agent can pick a strategy but how about what others will do?
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
- As far as own strategies are concerned, the agent can pick a strategy but how about what others will do?
- No general statements can be made here:
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
- As far as own strategies are concerned, the agent can pick a strategy but how about what others will do?
- No general statements can be made here:
  - Consider only opponents’ most likely/worst-case strategy and adjust own strategy to this
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
- As far as own strategies are concerned, the agent can pick a strategy but how about what others will do?
- No general statements can be made here:
  - Consider only opponents’ most likely/worst-case strategy and adjust own strategy to this
  - The range and temporal scope of validity of a chosen strategy may vary (when will strategies be reconsidered?)
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
- As far as own strategies are concerned, the agent can pick a strategy but how about what others will do?
- No general statements can be made here:
  - Consider only opponents’ most likely/worst-case strategy and adjust own strategy to this
  - The range and temporal scope of validity of a chosen strategy may vary (when will strategies be reconsidered?)
- Outcome of this decision making step: behavioural constraints imposed on the agent and her peers
Behaviours

- After analysing different strategies of others and oneself, agents determine their behaviour (much harder than it sounds)
- As far as own strategies are concerned, the agent can pick a strategy but how about what others will do?
- No general statements can be made here:
  - Consider only opponents’ most likely/worst-case strategy and adjust own strategy to this
  - The range and temporal scope of validity of a chosen strategy may vary (when will strategies be reconsidered?)
- Outcome of this decision making step: behavioural constraints imposed on the agent and her peers
- (Hypothetical) “suspension of autonomy” of others
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.
Outline

Introduction

A framework for expectation-based architectures

Strategic learning of communication patterns

Conclusions
Outline

Introduction

A framework for expectation-based architectures

Strategic learning of communication patterns

Conclusions
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

\[
F = \left\langle \left\langle 5 \rightarrow \text{request}(A_1, A_2, X) \rightarrow 3 \rightarrow \text{accept}(A_2, A_1, X) \rightarrow 2 \rightarrow \text{confirm}(A_1, A_2, X) \rightarrow 2 \rightarrow \text{do}(A_2, X) \right\rangle, \\
\left\langle \{\text{self}(A_1), \text{other}(A_2), \text{can}(A_1, \text{do}(A_1, X))\}, \right. \\
\left\{\text{agent}(A_1), \text{agent}(A_2), \text{action}(X)\}\right\rangle, \\
\left\langle 4 \rightarrow \left\langle [A_1/\text{agent}_1], [A_2/\text{agent}_2] \right\rangle, \\
1 \rightarrow \left\langle [A_1/\text{agent}_3], [A_2/\text{agent}_1], [X/\text{deliver\_goods}] \right\rangle \right\rangle
\]
Frame semantics

- Given a conversation prefix $w$ and a knowledge base $KB$, a set $\mathcal{F} = \{F_1, \ldots, F_n\}$ of frames induces a continuation probability:

$$P(w' | w) = \sum_{F \in \mathcal{F}} P(w' | F, w) P(F | w) = \sum_{F \in \mathcal{F}, ww' = T(F) \vartheta} P(\vartheta | F, w) P(F | w)$$
Frame semantics

- Given a conversation prefix $w$ and a knowledge base $KB$, a set $\mathcal{F} = \{F_1, \ldots, F_n\}$ of frames induces a continuation probability

$$P(w'|w) = \sum_{F \in \mathcal{F}} P(w'|F, w)P(F|w) = \sum_{F \in \mathcal{F}, ww' = T(F)\vartheta} P(\vartheta|F, w)P(F|w)$$

- Define probability of $\vartheta$ proportional to its similarity to $F$:

$$P(\vartheta|F, w) \propto \sigma(\vartheta, F) =$$

$$\sum_{i=1}^{|\Theta(F)|} \frac{\sigma(T(F)\vartheta, T(F)\Theta(F)[i])}{h_{\Theta}(F)[i]} c_i(F, \vartheta, KB)$$
The Framing Process

- Frames represent classes of interactions
The Framing Process

- Frames represent classes of interactions
- Proposed hierarchical decision-making approach:
  1. Select the appropriate frame for a given situation (i.e. classify the situation)
  2. Optimise within the selected frame while disregarding other frames
The Framing Process

- Frames represent classes of interactions
- Proposed hierarchical decision-making approach:
  1. Select the appropriate frame for a given situation (i.e. classify the situation)
  2. Optimise within the selected frame while disregarding other frames
- Apply hierarchical reinforcement learning methods to learn usefulness of frames in a given communication situation
  - Start with an initial set of pre-defined frames ("social rules")
  - Adapt frame models according to observed behaviour (or oneself and of others)
The Framing Process

- Frames represent classes of interactions
- Proposed hierarchical decision-making approach:
  1. Select the appropriate frame for a given situation (i.e. classify the situation)
  2. Optimise within the selected frame while disregarding other frames
- Apply hierarchical reinforcement learning methods to learn usefulness of frames in a given communication situation
  - Start with an initial set of pre-defined frames ("social rules")
  - Adapt frame models according to observed behaviour (or oneself and of others)
- Important: Architecture allows deviation from existing frames on both sides
Relationship to ESB

- The framing mechanism represents an expectation processing mechanism
Relationship to ESB

- The framing mechanism represents an expectation processing mechanism
  - Based on probabilistic model of communicative behaviour and utility-relevant actions
Relationship to 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)
Relationship to 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
Relationship to 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)
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)
- Incorporation of social behaviour in agents’ general planning processes
Relationship to 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)
  ▶ Incorporation of social behaviour in agents’ general planning processes
▶ Successfully applied in complex multiagent negotiation scenarios
Application: A Link Exchange System

Michael Rovatsos
The University of Edinburgh
Without Frame Learning
With Frame Learning

![Agent performance graph](image)

- **Agent utility**
- **Simulation rounds**
- **Average**
- **Minimum**
- **Maximum**
- **Lower benchmark**
- **Upper benchmark**

---

Michael Rovatsos  The University of Edinburgh
Outline

Introduction

A framework for expectation-based architectures

Strategic learning of communication patterns

Conclusions
Outline

Introduction

A framework for expectation-based architectures

Strategic learning of communication patterns

Conclusions
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”
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.
- Concept of expectation can be applied to all three types of mechanisms:
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.

- Concept of expectation can be applied to all three types of mechanisms:
  - Encode assumptions about mental states, deontic frameworks, and agent rationality in expectations.
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.
- Concept of expectation can be applied to all three types of mechanisms:
  - Encode assumptions about mental states, deontic frameworks, and agent rationality in expectations.
- Added flexibility through adaptiveness of expectations.
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.
- Concept of expectation can be applied to all three types of mechanisms:
  - Encode assumptions about mental states, deontic frameworks, and agent rationality in expectations.
- Added flexibility through adaptiveness of expectations:
  - Revise mentalistic assumptions as soon as agent behaviour indicates they are not valid.
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.
- Concept of expectation can be applied to all three types of mechanisms:
  - Encode assumptions about mental states, deontic frameworks, and agent rationality in expectations.
- Added flexibility through adaptiveness of expectations:
  - Revise mentalistic assumptions as soon as agent behaviour indicates they are not valid.
  - Design social laws (e.g. a commitment mechanism) with a focus on handling failure to meet social requirements.
Expressiveness

- ESB does not solve the basic problems of open systems, but it provides a uniform set of abstractions to deal with them.
- Concept of expectation can be applied to all three types of mechanisms:
  - Encode assumptions about mental states, deontic frameworks, and agent rationality in expectations.
- Added flexibility through adaptiveness of expectations:
  - Revise mentalistic assumptions as soon as agent behaviour indicates they are not valid.
  - Design social laws (e.g. a commitment mechanism) with a focus on handling failure to meet social requirements.
  - Drop rationality assumptions in mechanism design if agents behave irrationally.
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
The End

Thank you for your attention!