Computational Interaction Frames

Michael Rovatsos

CISA, School of Informatics
The University of Edinburgh

30th September, 2004

Outline

The bottom line (abstract)

Given a set of conversational **interaction patterns**, our method allows agents to **learn** to choose the most appropriate of these in order to maximise their own utility based on past communication **experience**.

The bottom line (abstract)

Given a set of conversational **interaction patterns**, our method allows agents to **learn** to choose the most appropriate of these in order to maximise their own utility based on past communication **experience**.

- Multiagent learning/communication learning

The bottom line (abstract)

Given a set of conversational **interaction patterns**, our method allows agents to **learn** to choose the most appropriate of these in order to maximise their own utility based on past communication **experience**.

- Multiagent learning/communication learning
- More specifically: dialogue management & conversation policy selection learning

The bottom line (abstract)

Given a set of conversational **interaction patterns**, our method allows agents to **learn** to choose the most appropriate of these in order to maximise their own utility based on past communication **experience**.

- Multiagent learning/communication learning
- More specifically: dialogue management & conversation policy selection learning
- Goal: design social reasoning architecture, build agents with these capabilities

The bottom line (abstract)

Given a set of conversational **interaction patterns**, our method allows agents to **learn** to choose the most appropriate of these in order to maximise their own utility based on past communication **experience**.

- Multiagent learning/communication learning
- More specifically: dialogue management & conversation policy selection learning
- Goal: design social reasoning architecture, build agents with these capabilities
- Formal and theoretical underpinnings, but focus on realism

The bottom line (technical)

Agents who use the suggested methods . . .

The bottom line (technical)

Agents who use the suggested methods ...

- Maintain sequences of speech-act like messages $m(s, r, c)$ as models of conversation runs

The bottom line (technical)

Agents who use the suggested methods ...

- Maintain sequences of speech-act like messages $m(s, r, c)$ as models of conversation runs
- Use patterns instead of actual messages, e.g. $m(X, y, C)$

The bottom line (technical)

Agents who use the suggested methods ...

- Maintain sequences of speech-act like messages $m(s, r, c)$ as models of conversation runs
- Use patterns instead of actual messages, e.g. $m(X, y, C)$
- Collect variable values and logical constraints from observed dialogue encounters

The bottom line (technical)

Agents who use the suggested methods ...

- Maintain sequences of speech-act like messages $m(s, r, c)$ as models of conversation runs
- Use patterns instead of actual messages, e.g. $m(X, y, C)$
- Collect variable values and logical constraints from observed dialogue encounters
- Apply reinforcement learning to these "macro-actions" (=patterns+instances+frequencies+constraints)

The bottom line (technical)

Agents who use the suggested methods ...

- Maintain sequences of speech-act like messages $m(s, r, c)$ as models of conversation runs
- Use patterns instead of actual messages, e.g. $m(X, y, C)$
- Collect variable values and logical constraints from observed dialogue encounters
- Apply reinforcement learning to these "macro-actions" (=patterns+instances+frequencies+constraints)
- Employ them in communication given own utility estimates and feedback from the environment

Communication vs. open systems

- Traditional approach to interaction and communication in a MAS:
  - Agent communication languages with speech-act based semantics (KQML/KIF, FIPA-ACL)
  - Communication protocols (CNP, auctions, ...)

Communication vs. open systems

- Traditional approach to interaction and communication in a MAS:
  - Agent communication languages with speech-act based semantics (KQML/KIF, FIPA-ACL)
  - Communication protocols (CNP, auctions, . . .)
- Leads to problems in *open* MAS (heterogeneity, different stakeholders, etc.)

Communication vs. open systems

- Traditional approach to interaction and communication in a MAS:
  - Agent communication languages with speech-act based semantics (KQML/KIF, FIPA-ACL)
  - Communication protocols (CNP, auctions, ...)
- Leads to problems in *open* MAS (heterogeneity, different stakeholders, etc.)
- Question: *If adherence to communication languages and protocols cannot be taken for granted, how can meaningful and coherent communication be ensured?*

Communication vs. open systems

- Traditional approach to interaction and communication in a MAS:
  - Agent communication languages with speech-act based semantics (KQML/KIF, FIPA-ACL)
  - Communication protocols (CNP, auctions, . . .)
- Leads to problems in *open* MAS (heterogeneity, different stakeholders, etc.)
- Question: *If adherence to communication languages and protocols cannot be taken for granted, how can meaningful and coherent communication be ensured?*
- One possible answer: **empirical semantics**

Empirical Semantics

- Meaning of a message is only defined in terms of its consequences (i.e. messages/actions likely to follow it)
  - Immediate reactions of other agents and oneself
  - "Second-order" impact on the expectation structures of any observer

Empirical Semantics

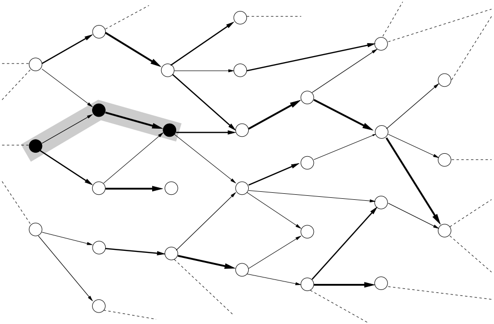
- Meaning of a message is only defined in terms of its consequences (i.e. messages/actions likely to follow it)
  - Immediate reactions of other agents and oneself
  - "Second-order" impact on the expectation structures of any observer
- Knowledge about the effects of messages must be derived from empirical observation

Empirical Semantics

- Meaning of a message is only defined in terms of its consequences (i.e. messages/actions likely to follow it)
  - Immediate reactions of other agents and oneself
  - "Second-order" impact on the expectation structures of any observer
- Knowledge about the effects of messages must be derived from empirical observation
- Meaning can only be constructed through the eyes of an agent, in relation to its goals

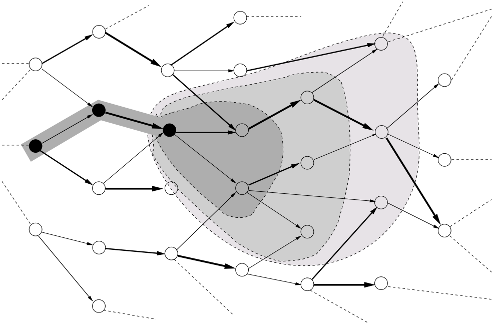
Communication Systems

General way of viewing structure and evolution of communication: **expectation networks**



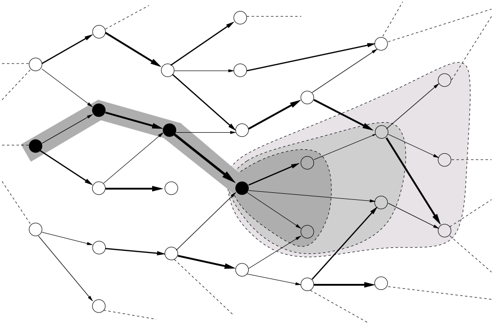
Communication Systems

General way of viewing structure and evolution of communication: **expectation networks**



Communication Systems

General way of viewing structure and evolution of communication: **expectation networks**



Communication Systems

Advantages over "traditional" models of communication semantics:

- No mentalistic assumptions, least commitment approach

Communication Systems

Advantages over “traditional” models of communication semantics:

- No mentalistic assumptions, least commitment approach
- Allows for context-sensitivity and uncertainty

Communication Systems

Advantages over “traditional” models of communication semantics:

- No mentalistic assumptions, least commitment approach
- Allows for context-sensitivity and uncertainty
- Modelling of local and/or global meaning

Communication Systems

Advantages over “traditional” models of communication semantics:

- No mentalistic assumptions, least commitment approach
- Allows for context-sensitivity and uncertainty
- Modelling of local and/or global meaning
- Able to capture evolution of meaning

Communication Systems

Advantages over “traditional” models of communication semantics:

- No mentalistic assumptions, least commitment approach
- Allows for context-sensitivity and uncertainty
- Modelling of local and/or global meaning
- Able to capture evolution of meaning
- *But: how do we get them into agents’ heads (practically speaking)?*

Sociological Foundations

- "Frame" & "Framing" concepts grounded in the sociological theory of Erving Goffman (1922-1982)

Sociological Foundations

- "Frame" & "Framing" concepts grounded in the sociological theory of Erving Goffman (1922-1982)
- A frame is

  > *"the participants' own conceptualisation of the structure within which they are interacting, which may change very quickly as the situation develops"*

  or

  > *"the answer to the question 'what is going on here?' that everyone poses to oneself in an interaction situation"*

Sociological Foundations

- "Frame" & "Framing" concepts grounded in the sociological theory of Erving Goffman (1922-1982)
- A frame is

  > *"the participants' own conceptualisation of the structure within which they are interacting, which may change very quickly as the situation develops"*

  or

  > *"the answer to the question 'what is going on here?' that everyone poses to oneself in an interaction situation"*

- Framing = strategic application of frames

The InFFrA Architecture

- InFFrA = Interaction Frames and Framing Architecture

The InFFrA Architecture

- InFFrA = Interaction Frames and Framing Architecture
- Abstract architecture for social reasoning and learning

The InFFrA Architecture

- InFFrA = Interaction Frames and Framing Architecture
- Abstract architecture for social reasoning and learning
- Uses frames to capture regularities of interaction processes

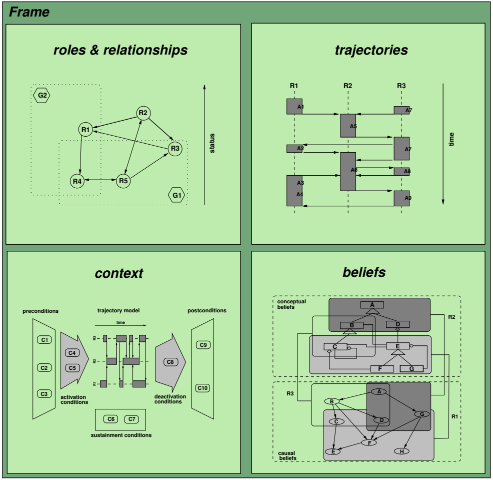
The InFFrA Architecture

- InFFrA = Interaction Frames and Framing Architecture
- Abstract architecture for social reasoning and learning
- Uses frames to capture regularities of interaction processes
- Framing = social reasoning mechanism that builds around frames as central data structure

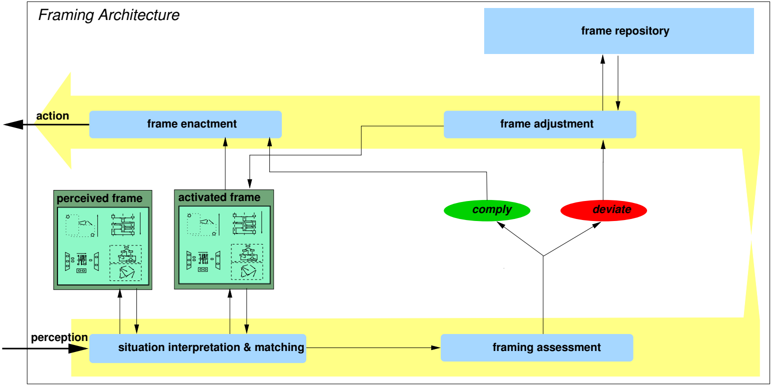
The InFFrA Architecture

- InFFrA = Interaction Frames and Framing Architecture
- Abstract architecture for social reasoning and learning
- Uses frames to capture regularities of interaction processes
- Framing = social reasoning mechanism that builds around frames as central data structure
- Intended to be combined with sub-social reasoning components (e.g. BDI reasoner)

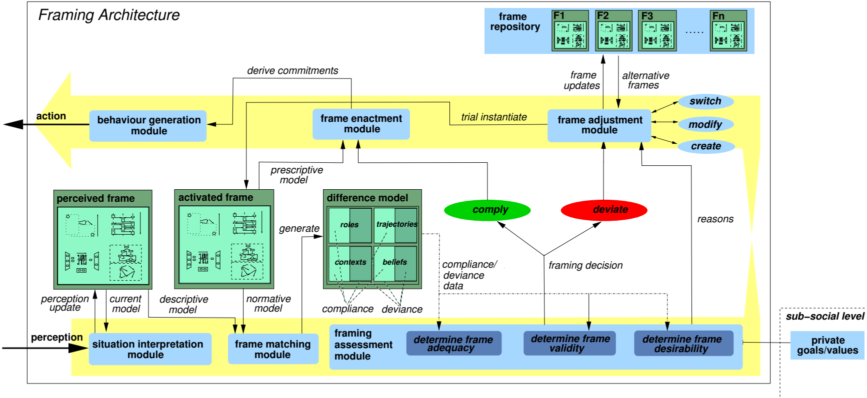
InFFrA – Frames



InFFrA – Framing



InFFrA – Framing



InFFrA – Summary

- Abstract architecture, many possible designs

InFFrA – Summary

- Abstract architecture, many possible designs
- Generic model for agent-level reasoning about interaction

InFFrA – Summary

- Abstract architecture, many possible designs
- Generic model for agent-level reasoning about interaction
- Difference between frames and interaction protocols/conversation policies:
  - Not fixed *a priori*, evolving
  - Include information about context and experience
  - Are vulnerable to manipulation (e.g. deception)
  - Actors move fluidly/rapidly between frames

m^2InFFrA

- m^2InFFrA: an instance of InFFrA for two-party, discrete, turn-taking interactions

m^2InFFrA

- m^2InFFrA: an instance of InFFrA for two-party, discrete, turn-taking interactions
- "Markov-square": two-level hierarchical MDP view of frame-based interaction

m^{2}InFFrA

- m^{2}InFFrA: an instance of InFFrA for two-party, discrete, turn-taking interactions
- "Markov-square": two-level hierarchical MDP view of frame-based interaction
- Frame $F = (T, \Theta, C, h_T, h_\Theta)$

T	a sequence of message patterns, the trajectory
Θ	a list of variable substitutions
C	a list of condition sets (in a propositional language)
h_T	trajectory occurence counter
h_Θ	substitution occurrence counter

An example

$$
\begin{aligned}
F = \Big\langle & \big\langle \xrightarrow{5} \texttt{request}(A_1, A_2, X) \xrightarrow{3} \texttt{accept}(A_2, A_1, X) \\
& \quad \xrightarrow{2} \texttt{confirm}(A_1, A_2, X) \xrightarrow{2} \texttt{do}(A_2, X)\big\rangle, \\
& \big\langle \{\mathit{self}(A_1), \mathit{other}(A_2), \mathit{can}(A_1, \texttt{do}(A_1, X))\}, \\
& \quad \{\mathit{agent}(A_1), \mathit{agent}(A_2), \mathit{action}(X)\}\big\rangle, \\
& \big\langle \xrightarrow{4} \langle [A_1/\mathrm{agent\_1}], [A_2/\mathrm{agent\_2}]\rangle, \\
& \quad \xrightarrow{1} \langle [A_1/\mathrm{agent\_3}], [A_2/\mathrm{agent\_1}], [X/\texttt{deliver\_goods}]\rangle \big\rangle \Big\rangle
\end{aligned}
$$

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 ϑ proportional to its *similarity* to F:

$$P(\vartheta|F, w) \propto \sigma(\vartheta, F) =$$
$$\sum_{i=1}^{|\Theta(F)|} \overbrace{\sigma(T(F)\vartheta, T(F)\Theta(F)[i])}^{\text{similarity}} \overbrace{h_\Theta(F)[i]}^{\text{frequency}} \overbrace{c_i(F, \vartheta, KB)}^{\text{relevance}}$$

Framing in m^{2}InFFrA

- Frames represent classes of interactions

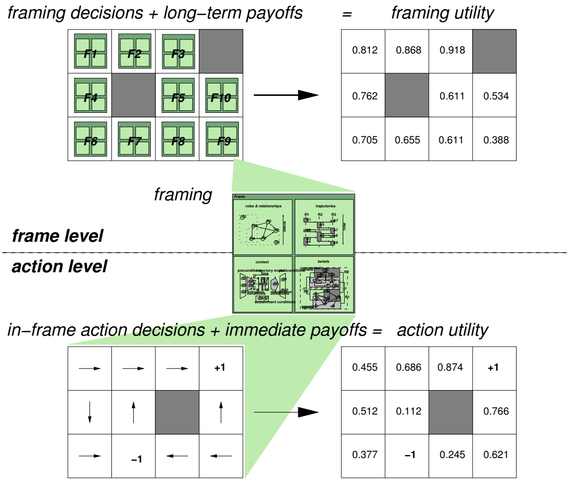
Framing in m^2InFFrA

- Frames represent classes of interactions
- Proposed hierarchical 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

Framing in m^{2}InFFrA

- Frames represent classes of interactions
- Proposed hierarchical 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
- Learning methods can be applied to both levels (frame-level/action-level)

Framing in m^{2}InFFrA



Action-level Optimisation

- Substitution *fixed* by conversation prefix w in frame F:

$$\vartheta_f(F, w) = \mathit{unifier}(w, T(F)[1{:}|w|])$$

Action-level Optimisation

- Substitution *fixed* by conversation prefix w in frame F:

$$\vartheta_f(F, w) = \mathit{unifier}(w, T(F)[1{:}|w|])$$

- Set of substitutions still possible:

$$\Theta_{poss}(F, KB, w) = \left\{\vartheta \middle| \exists\vartheta'.\vartheta = \vartheta_f(F, w)\vartheta' \wedge \exists i.KB \models C[i]\vartheta\right\}$$

Action-level Optimisation

- Substitution *fixed* by conversation prefix w in frame F:

$$\vartheta_f(F, w) = \mathit{unifier}(w, T(F)[1{:}|w|])$$

- Set of substitutions still possible:

$$\Theta_{poss}(F, KB, w) = \left\{\vartheta \middle| \exists\vartheta'.\vartheta = \vartheta_f(F, w)\vartheta' \wedge \exists i.KB \models C[i]\vartheta\right\}$$

- "Own" and "Peer" substitution ϑ_s and ϑ_p

Action-level Optimisation

- Substitution *fixed* by conversation prefix w in frame F:

$$\vartheta_f(F, w) = unifier(w, T(F)[1{:}|w|])$$

- Set of substitutions still possible:

$$\Theta_{poss}(F, KB, w) = \{\vartheta \mid \exists \vartheta'.\vartheta = \vartheta_f(F, w)\vartheta' \wedge \exists i.KB \models C[i]\vartheta\}$$

- "Own" and "Peer" substitution ϑ_s and ϑ_p
- (Private) utility estimate over future message sequences

Action-level Optimisation

- Expected utility of "own" substitution ϑ_s:

$$E[u(\vartheta_s, F, w, KB)] = \sum_{\vartheta_p} P(\vartheta_p | \vartheta_s, F, w) \cdot$$
$$u\big(postfix(T(F), w)\vartheta_f(F, w)\vartheta_s\vartheta_p, KB\big)$$

Action-level Optimisation

- Expected utility of "own" substitution ϑ_s:

$$E[u(\vartheta_s, F, w, KB)] = \sum_{\vartheta_p} P(\vartheta_p | \vartheta_s, F, w) \cdot$$
$$u\big(postfix(T(F), w)\vartheta_f(F, w)\vartheta_s\vartheta_p, KB\big)$$

- Expected utility maximisation to determine optimal action

$$\begin{aligned} \vartheta^*(F, w, KB) &= \arg\max_{\vartheta_s \in \Theta_s} E[u(\vartheta_s, F, w, KB)] \\ m^*(F, w, KB) &= T(F)[|w| + 1]\vartheta^*(F, w, KB) \end{aligned}$$

Action-level Optimisation

- Conditional probability for "peer" substitution estimated from previous instantiations of F:

$$
\begin{aligned}
P(\vartheta_p|\vartheta_s, F, w) &= \frac{P(\vartheta_s \wedge \vartheta_p|F, w)}{P(\vartheta_s|F, w)} = \\
&= \frac{P(\vartheta_f(F, w)\vartheta_s\vartheta_p|F, w)}{\sum_\vartheta P(\vartheta_f(F, w)\vartheta_s\vartheta|F, w)} \\
&\propto \sigma(\vartheta_f(F, w)\vartheta_s\vartheta_p, F)
\end{aligned}
$$

Frame-level Learning

- Reinforcement learning (RL): learning an *optimal* policy π^* in an MDP

Frame-level Learning

- Reinforcement learning (RL): learning an *optimal* policy π^* in an MDP
- Maximisation of the expected profit,
  i. e. $\pi^* = \arg\max_\pi E(\sum_{\tau=1}^{\infty} \gamma^{\tau-1} r_{t+\tau} | s_t = s.\pi)$

Frame-level Learning

- Reinforcement learning (RL): learning an *optimal* policy π^* in an MDP
- Maximisation of the expected profit,
  i. e. $\pi^* = \arg\max_\pi E(\sum_{\tau=1}^{\infty} \gamma^{\tau-1} r_{t+\tau} | s_t = s.\pi)$
- Problem: large number of state/action pairs ("curse of dimensionality")

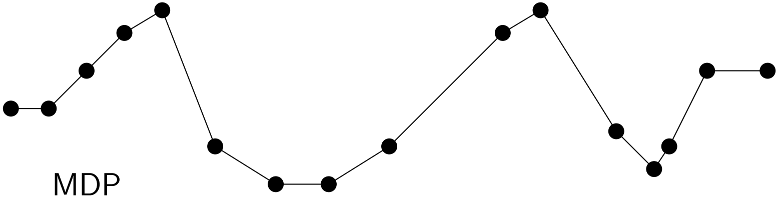
Frame-level Learning

- Reinforcement learning (RL): learning an *optimal* policy π^* in an MDP
- Maximisation of the expected profit, i. e. $\pi^* = \arg\max_\pi E(\sum_{\tau=1}^{\infty} \gamma^{\tau-1} r_{t+\tau} | s_t = s.\pi)$
- Problem: large number of state/action pairs ("curse of dimensionality")
- (One) solution: use "macro actions" to model temporally extended courses of action

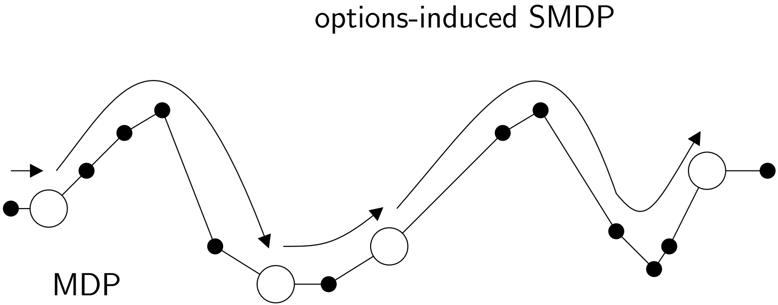
Frame-level Learning

- Reinforcement learning (RL): learning an *optimal* policy π^* in an MDP
- Maximisation of the expected profit, i. e. $\pi^* = \arg\max_\pi E(\sum_{\tau=1}^{\infty} \gamma^{\tau-1} r_{t+\tau} | s_t = s.\pi)$
- Problem: large number of state/action pairs ("curse of dimensionality")
- (One) solution: use "macro actions" to model temporally extended courses of action
- Leads to semi-MDP (SMDP) i. e. state transition probabilities and rewards epend on the *history* of states since the macro has been invoked and to hierarchical RL

SMDPs – Intuitively speaking . . .



SMDPs – Intuitively speaking . . .



Frame-level Learning & Options

- Options: a framework for hierarchical RL, blends nicely with interaction frames

Frame-level Learning & Options

- Options: a framework for hierarchical RL, blends nicely with interaction frames
- "Core" MDP, action set $\mathcal{A}_s$ is augmented by a set of options

Frame-level Learning & Options

- Options: a framework for hierarchical RL, blends nicely with interaction frames
- "Core" MDP, action set $\mathcal{A}_s$ is augmented by a set of options
- *option* $o = (\mathcal{I}, \pi, beta)$

$\mathcal{I} \subseteq \mathcal{S}$	input set
$\pi : \mathcal{S} \times \bigcup_s \mathcal{A}_s \rightarrow [0, 1]$	(intra-option) policy
$\beta : \mathcal{S} \rightarrow [0, 1]$	Terminierungsbedingung

Frame-based Options

For each frame F, define an option $(\mathcal{I}_F, \pi_F, \beta_F)$:

Frame-based Options

For each frame F, define an option $(\mathcal{I}_F, \pi_F, \beta_F)$:

$\mathcal{I}_F$ the set of states (w, KB) such that $T(F)$ "matches" w and the corresponding suffix is executable under KB

Frame-based Options

For each frame F, define an option $(\mathcal{I}_F, \pi_F, \beta_F)$:

$\mathcal{I}_F$ the set of states (w, KB) such that $T(F)$ "matches" w and the corresponding suffix is executable under KB

π_F deterministic policy that chooses the message/action that matches the next step of $T(F)$ while maximises expected suffix utility

Frame-based Options

For each frame F, define an option $(\mathcal{I}_F, \pi_F, \beta_F)$:

$\mathcal{I}_F$ the set of states (w, KB) such that $T(F)$ "matches" w and the corresponding suffix is executable under KB

π_F deterministic policy that chooses the message/action that matches the next step of $T(F)$ while maximises expected suffix utility
"expectation" computed by comparing the (projected) present encounter with past ones stored in $\Theta(F)$ (using σ similarity measure)

Frame-based Options

For each frame F, define an option $(\mathcal{I}_F, \pi_F, \beta_F)$:

$\mathcal{I}_F$ the set of states (w, KB) such that $T(F)$ "matches" w and the corresponding suffix is executable under KB

π_F deterministic policy that chooses the message/action that matches the next step of $T(F)$ while maximises expected suffix utility
"expectation" computed by comparing the (projected) present encounter with past ones stored in $\Theta(F)$ (using σ similarity measure)

β_F determined by $T(F)$, w and KB (as $\mathcal{I}_F$) and by a private desirability measure

Introduction

The Conceptual Level: InFFrA

The Formal Level: m^2InFFrA

Application & Results

- Automated Web Link Exchange
- Experimental Results
- Interest-based Negotiation

Summary & Conclusions

LIESON

- LInk Exchange SimulatiON System

LIESON

- LInk Exchange SimulatiON System
- Objective: increase linkage transparency on the WWW using automated link exchange

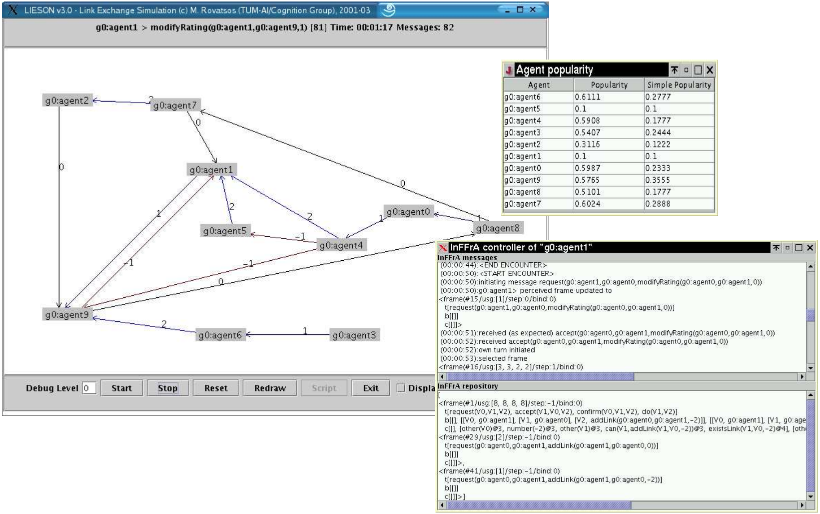
LIESON

- LInk Exchange SimulatiON System
- Objective: increase linkage transparency on the WWW using automated link exchange
- Includes implementation of BDI-like agents with m^2InFFrA engine
  - self-interested agents
  - maximise dissemination of own opinion
  - (highly) boundedly rational

LIESON

- LInk Exchange SimulatiON System
- Objective: increase linkage transparency on the WWW using automated link exchange
- Includes implementation of BDI-like agents with m^2InFFrA engine
  - self-interested agents
  - maximise dissemination of own opinion
  - (highly) boundedly rational
- Experimented with two kinds of negotiation:
  - proposal-based negotiation
  - **interest-based negotiation**

The LIESON System



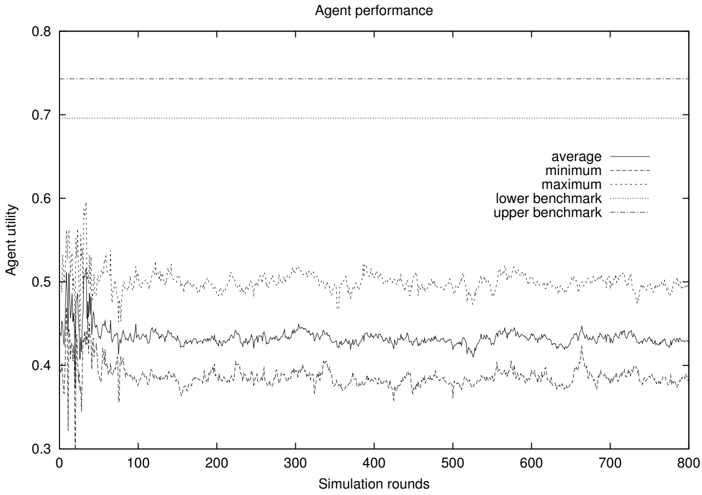
Proposal-based negotiation

$$F_1 = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{accept}(B,A,X) \xrightarrow{0} \texttt{confirm}(A,B,X) \xrightarrow{0} \texttt{do}(B,X)\big\rangle,$$
$$\big\langle can(B,X)@3, effects(X)@4\}\big\rangle$$
$$\big\langle \xrightarrow{0} \langle\rangle\big\rangle\Big\rangle$$

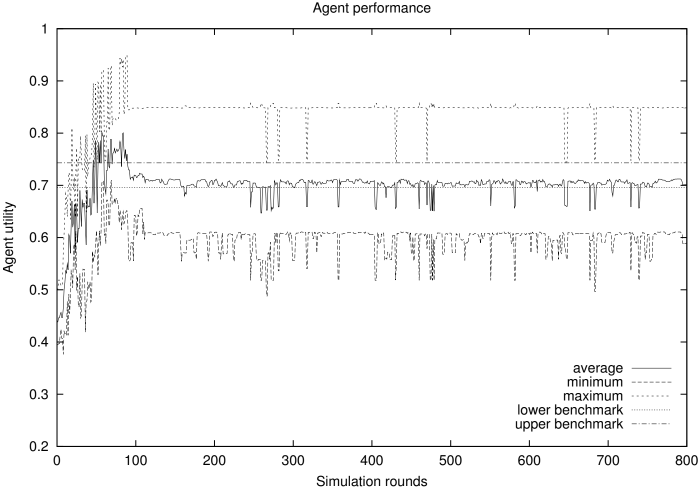
$$F_2 = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{propose}(B,A,Y) \xrightarrow{0} \texttt{accept}(A,B,Y) \xrightarrow{0} \texttt{do}(B,Y)\big\rangle,$$
$$\big\langle \{can(B,Y)@3, effects(Y)@4\}\big\rangle$$
$$\big\langle \xrightarrow{0} \langle\rangle\big\rangle\Big\rangle$$

$$F_3 = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{propose}-\texttt{also}(B,A,Y) \xrightarrow{0} \texttt{accept}(A,B,Y)$$
$$\xrightarrow{0} \texttt{do}(B,X) \xrightarrow{0} \texttt{do}(A,Y)\big\rangle,$$
$$\big\langle \{can(B,X)@3, effects(X)@4, can(A,Y)@4, effects(Y)@5\}\big\rangle$$
$$\big\langle \xrightarrow{0} \langle\rangle\big\rangle\Big\rangle$$

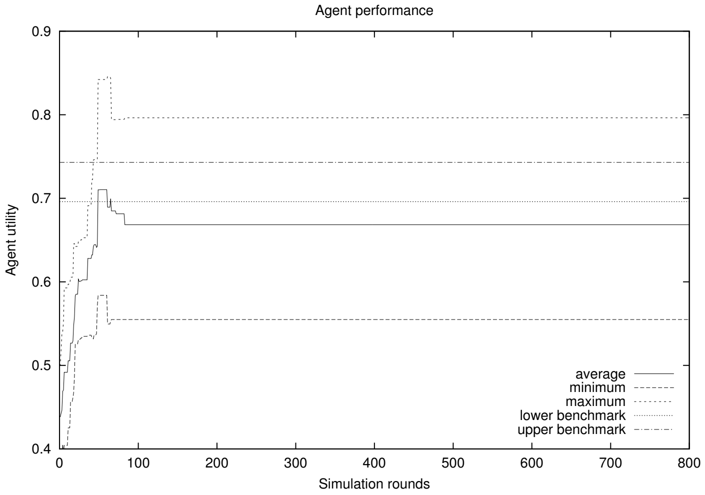
Random agents



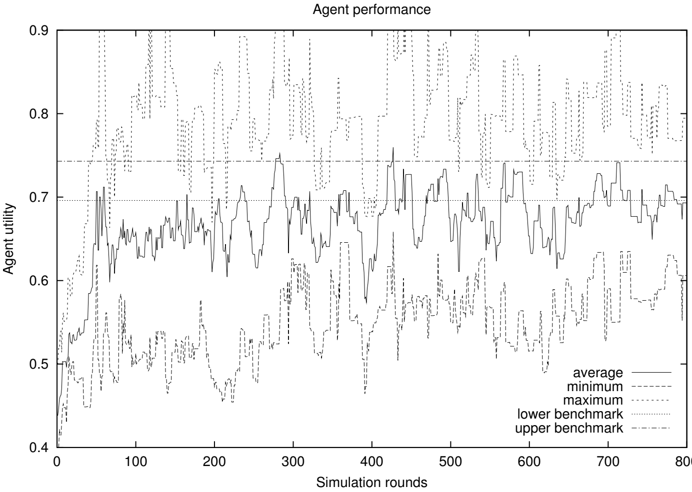
Non-communicating BDI agents



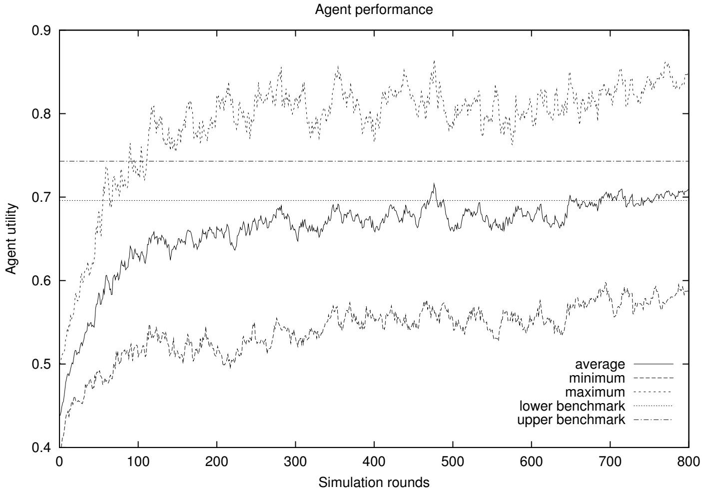
Communicating BDI agents



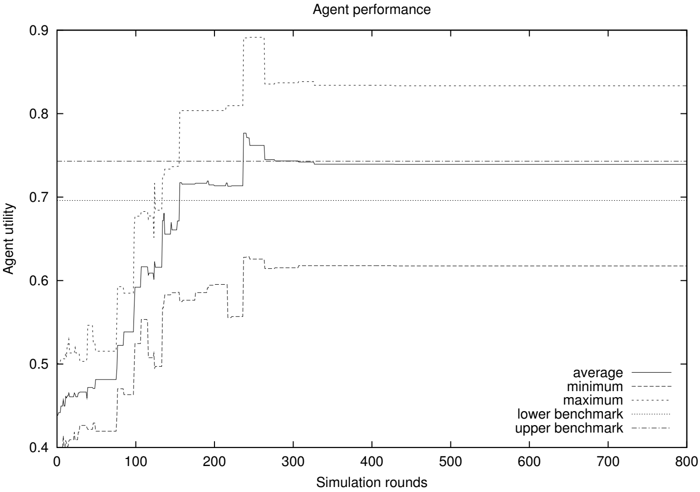
No desirability test (single run)



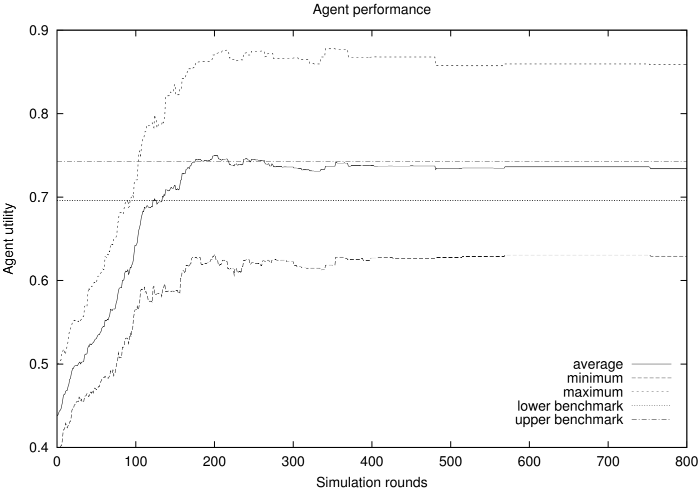
No desirability test (100 runs)



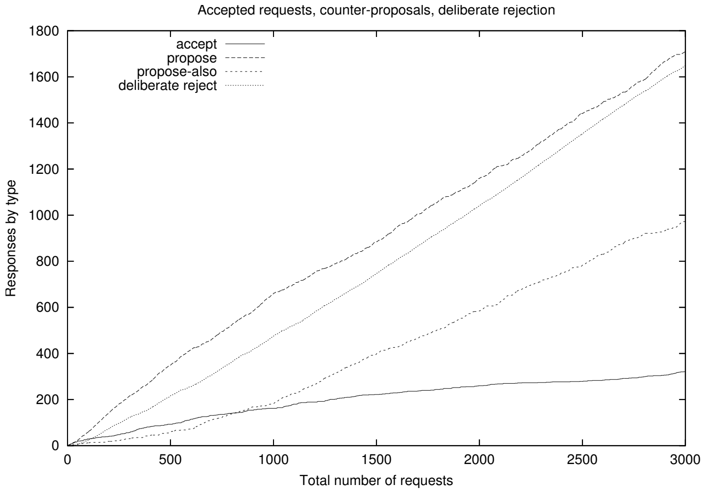
Framing desirability test (single run)



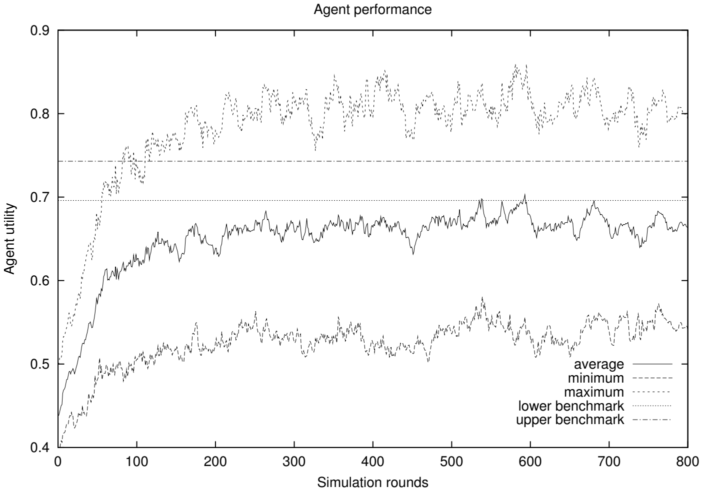
Framing desirability test (100 runs)



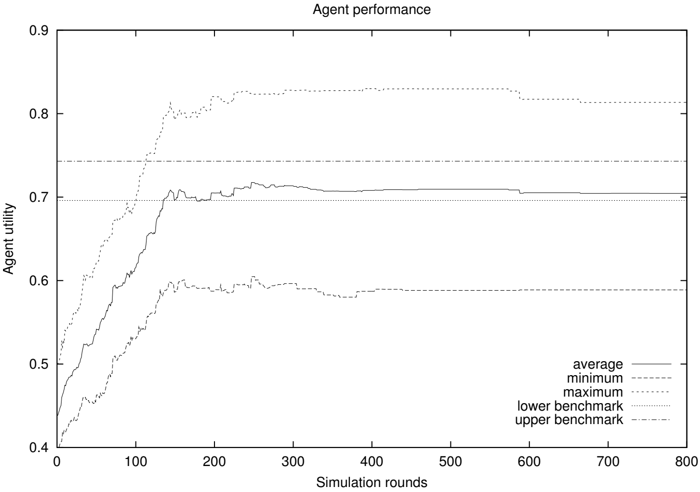
Learning different responses



No desirability, no learning (100 runs)



Desirability test, no learning (100 runs)



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

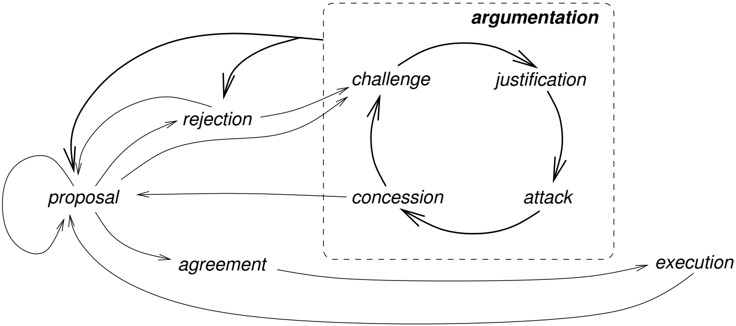
Interest-based Negotiation (IBN)

- 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"

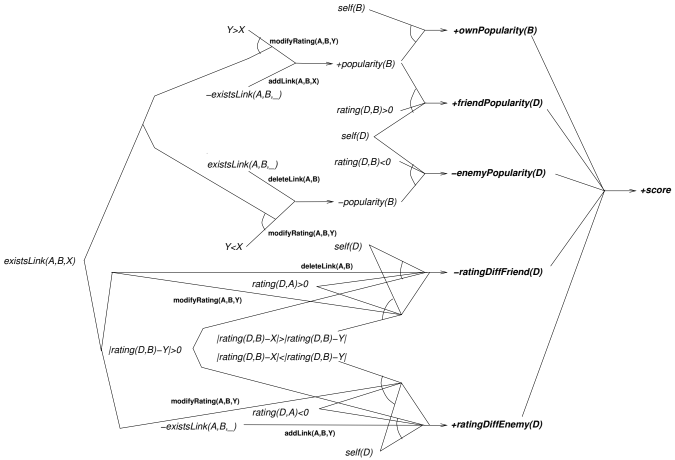
Interest-based Negotiation (IBN)

- 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.

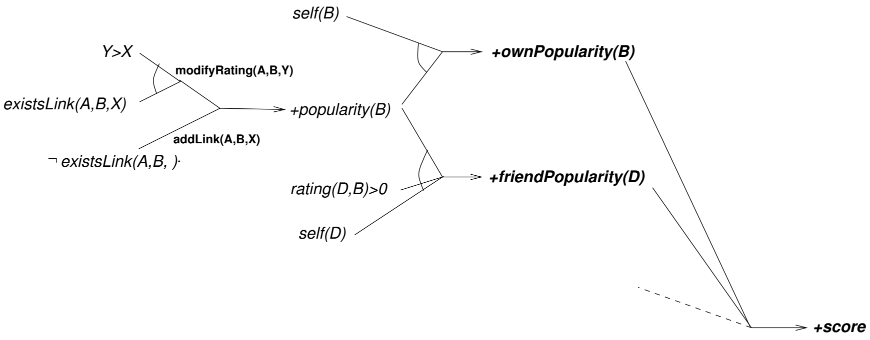
IBN – Dialogue model



IBN – Goal graphs



IBN – Goal graph (detail)



IBN frames – Example

$$
\begin{aligned}
F_{AGM} = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{ask-reason}(B,A,\texttt{request}(X)) \xrightarrow{0} \\
& \quad \texttt{inform-goal}(A,B,G) \xrightarrow{0} \\
& \quad \texttt{attack-goal}(B,A,\textit{alternative-action}(Y)) \\
& \quad \xrightarrow{0} \texttt{concede}(A,B,Y) \xrightarrow{0} \texttt{do}(B,Y)\big\rangle, \\
& \big\langle\{can(B,X), goal(A,G), achieves(X,G), achieves(Y,G), \\
& \quad X \neq Y, can(B,Y)@5, effects(Y)@6\}\big\rangle, \big\langle \xrightarrow{0} \langle\rangle\big\rangle \Big\rangle
\end{aligned}
$$

IBN frames – Example

$$F_{AGM} = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{ask-reason}(B,A,\texttt{request}(X)) \xrightarrow{0} \texttt{inform-goal}(A,B,G) \xrightarrow{0} \texttt{attack-goal}(B,A,\textit{alternative-action}(Y)) \xrightarrow{0} \texttt{concede}(A,B,Y) \xrightarrow{0} \texttt{do}(B,Y)\big\rangle, \big\langle\{\textit{can}(B,X),\textit{goal}(A,G),\textit{achieves}(X,G),\textit{achieves}(Y,G), X \neq Y, \textit{can}(B,Y)@5, \textit{effects}(Y)@6\}\big\rangle, \big\langle \xrightarrow{0} \langle\rangle\big\rangle\Big\rangle$$

- Performance of m^{2}InFFrA agents comparable to that with proposal-based frames

Main Contributions

- Abstract social reasoning architecture based on interaction frames

Main Contributions

- Abstract social reasoning architecture based on interaction frames
- "Open" alternative to rigid protocols (empirical semantics as an alternative to pre-speccied ACL semantics)

Main Contributions

- Abstract social reasoning architecture based on interaction frames
- "Open" alternative to rigid protocols (empirical semantics as an alternative to pre-speccied ACL semantics)
- Bridging the gap between protocol design and agent design

Main Contributions

- Abstract social reasoning architecture based on interaction frames
- "Open" alternative to rigid protocols (empirical semantics as an alternative to pre-speccied ACL semantics)
- Bridging the gap between protocol design and agent design
- Application of machine learning techniques to agent-level communication learning

Main Contributions

- Abstract social reasoning architecture based on interaction frames
- "Open" alternative to rigid protocols (empirical semantics as an alternative to pre-speccied ACL semantics)
- Bridging the gap between protocol design and agent design
- Application of machine learning techniques to agent-level communication learning
- Integration of different components to a practical, implemented system

What this talk did not cover

- Frame merging (generalisation using cluster validation methods)

What this talk did not cover

- Frame merging (generalisation using cluster validation methods)
- Frame concatenation (in a planning sense), iterative interest-based negotiation frames

What this talk did not cover

- Frame merging (generalisation using cluster validation methods)
- Frame concatenation (in a planning sense), iterative interest-based negotiation frames
- Entropy-based desirability criteria

What this talk did not cover

- Frame merging (generalisation using cluster validation methods)
- Frame concatenation (in a planning sense), iterative interest-based negotiation frames
- Entropy-based desirability criteria
- State abstraction in Q-learning (hot topic!)

What this talk did not cover

- Frame merging (generalisation using cluster validation methods)
- Frame concatenation (in a planning sense), iterative interest-based negotiation frames
- Entropy-based desirability criteria
- State abstraction in Q-learning (hot topic!)
- Further applications
  - Opponent classification in multiagent games
  - Deontic autonomy specifications
  - Combination with macro-level communication systems

Future Work

- State and action abstractions for communication to encode the status of a conversation, e.g. in negotiation

Future Work

- State and action abstractions for communication to encode the status of a conversation, e.g. in negotiation
- Meta-communication: negotiating frame conceptions themselves

Future Work

- State and action abstractions for communication to encode the status of a conversation, e.g. in negotiation
- Meta-communication: negotiating frame conceptions themselves
- Looking at other applications, in particular Semantic Web with focus on interaction

Future Work

- State and action abstractions for communication to encode the status of a conversation, e.g. in negotiation
- Meta-communication: negotiating frame conceptions themselves
- Looking at other applications, in particular Semantic Web with focus on interaction
- Leightweight implementation (volunteers?)

Thank you for your attention!
Questions?

Digression: Markov Decision Processes

- Definition (discrete, stochastic MDP):
  $\mathcal{S}$ set of *states*
  $\mathcal{A}_s$ sets of (admissible) *actions* for $s \in \mathcal{S}$
  $p^a_{ss'} = P(s_{t+1} | s_t = s, a_t = a)$
  state transition model
  $r^a_s = E(r_{t+1} | s_t = s, a_t = a)$
  (expected) *reward* if a is executed in s

Digression: Markov Decision Processes

- Definition (discrete, stochastic MDP):
  $\mathcal{S}$ set of *states*
  $\mathcal{A}_s$ sets of (admissible) *actions* for $s \in \mathcal{S}$
  $p^a_{ss'} = P(s_{t+1}|s_t = s, a_t = a)$
  state transition model
  $r^a_s = E(r_{t+1}|s_t = s, a_t = a)$
  (expected) *reward* if a is executed in s
- Markov-property: $p^a_{ss'}$ and r^a_s solely depend on the current state s

Digression: Markov Decision Processes

- Definition (discrete, stochastic MDP):
  $\mathcal{S}$ set of *states*
  $\mathcal{A}_s$ sets of (admissible) *actions* for $s \in \mathcal{S}$
  $p^a_{ss'} = P(s_{t+1}|s_t = s, a_t = a)$
  state transition model
  $r^a_s = E(r_{t+1}|s_t = s, a_t = a)$
  (expected) *reward* if a is executed in s
- Markov-property: $p^a_{ss'}$ and r^a_s solely depend on the current state s
- Agent behaviour modelled using a (discrete, stochastic) *policy* $\pi : \mathcal{S} \times \bigcup_s \mathcal{A}_s \rightarrow [0, 1]$

Digression: Q-Lerning with options

- Q-learning solves the RL problem by learning the value $Q^*(s, a)$ of executing a in s, thereafter following π^*

Digression: Q-Lerning with options

- Q-learning solves the RL problem by learning the value $Q^*(s, a)$ of executing a in s, thereafter following π^*
- Done by updating an approximation of Q^* from sampled state trasitions and rewards

Digression: Q-Lerning with options

- Q-learning solves the RL problem by learning the value $Q^*(s, a)$ of executing a in s, thereafter following π^*
- Done by updating an approximation of Q^* from sampled state trasitions and rewards
- Upate equation for SMDP Q-learning

$$Q(s, o) \leftarrow (1 - \alpha) Q_k(s, o) + \alpha \left[r + \gamma^{\tau} \max_{o' \in \mathcal{O}_{s'}} Q_k(s', o') \right]$$

Digression: Q-Lerning with options

- Q-learning solves the RL problem by learning the value $Q^*(s, a)$ of executing a in s, thereafter following π^*
- Done by updating an approximation of Q^* from sampled state trasitions and rewards
- Upate equation for SMDP Q-learning

$$Q(s, o) \leftarrow (1 - \alpha) Q_k(s, o) + \alpha \left[r + \gamma^\tau \max_{o' \in \mathcal{O}_{s'}} Q_k(s', o') \right]$$

- An optimal policy is then given by

$$\pi^*(s, a) = \arg\max_{a'} Q^*(s, a')$$