Agent Protocols for Social Computation

Michael Rovatsos, Dimitrios Diochnos, Matei Craciun
University of Edinburgh
MFSC@AAMAS, 3rd May 2015
Introduction

• Social computation is growing massively
• Platforms resemble multiagent systems
• Yet agent technologies are not used by them
  – Existing agent platforms too complex?
  – Benefits not obvious outside agent community?
Contribution

• Framework for using agent protocols without the need for point-to-point communication
• Based on standard architecture of the Web
  – More likely to achieve uptake
  – Exploit scalability, robustness, simplicity
• Semantics for translating protocols to data-driven view of communication
• Implementation and preliminary empirical evaluation of scalability/robustness
Disclaimer

• Everything the proposed communication framework can do can be done without it
• Formal analysis only serves to show correctness, no provable superior properties
• Advantages have to be proven in the longer term, only initial empirical evidence here
Agent protocols
Protocol graphs

\[
\{(I, \{I \cap I\}), (G, \{G \cap G\})\}
\]

\[
\{(\text{wants}_p, \{G\}), (s_p, \{I\})\}
\]

NO_SOLUTIONS(o, p, I, G, C)

REQUEST(p, o, I, G, C)

INFORM_TASK(o, p, t)

AGREE(p, o, t)

REJECT(p, o, t)

INFORM_INVALID(o, p, t)
Simulation results

![Graph showing matchmaking duration vs number of groups for different agent counts](image-url)

- 6 agents/group
- 9 agents/group
- 12 agents/group
Simulation results

![Graph showing negotiation duration vs. number of groups for different numbers of agents per group](image-url)
Simulation results
Related work

• Unlike blackboard systems, Linda etc we do *not* aim to replace but utilise agent protocols
• Unlike electronic institutions etc we do *not* aim to hand over control to central entities
• Unlike “agents-based web services” our semantics includes non-communicative action
• Important: instead of using bespoke middleware, we rely only on Web architecture
Future work

• Evaluate practical benefits of different communication framework more extensively
• Exploring more different types of protocols and degrees of computational distribution
• Develop methods to produce Web APIs automatically from protocol descriptions