Social Performance Measurement for Multiagent Systems

Michael Rovatsos
rovatsos@informatik.tu-muenchen.de

Computer Science Department
Technical University of Munich
Motivation

- Multiagent systems (MAS) find their way to industrial and commercial applications
- Measuring their performance becomes increasingly important
- Agent-oriented Software Engineering (AOSE) provides no methods yet
- Motivation for social performance measurement:
  - challenge: complexity (decentralisation, emergence, micro-macro)
  - opportunity: exploiting high-level communication between agents
- This talk: first steps and initial results
Overview
Introduction
Introduction

Software Engineering is (among other things) concerned with meeting product requirements, such as

- availability,
- modifiability,
- security,
- usability,
- ... 

In Performance Engineering, ideally,

- external attributes should be derived from
- internal attributes
Introduction (II)

In MAS particularly challenging due to

- encapsulation of functionality at agent-level,
- adaptive agents in dynamic environments,
- complex interactions between computational sub-processes,
- emergent (unforeseen) macro-level phenomena,
- potential ignorance of agents’ internal functionalities.

→ lack of work on performance measurement
Introduction (III)

Observation:
MAS exhibit certain properties that are actually advantageous in the development of measures of internal attributes.

1. They are (usually) based on deliberative, knowledge-based (i.e. symbolically operating) agents.

2. Agents in MAS (usually) communicate in high-level languages such as KQML or FIPA-ACL.
Principles of Social Performance Measurement
Social Performance Measurement (I)

Idea of social performance measurement: To exploit
(a) the existing layer of ongoing, symbolic communication in the system which
(b) captures the interactions between agents
(c) and which is comprehensible for humans.

Goals:

- develop simple, easily measurable measures based solely on internal communication data (⇒ social)
- map these to external attributes
Social Performance Measurement (II)

Assumptions:

- MAS exhibit *social properties* that come about from communication processes
- use *communication data* as data material for performance measurement

Communication can be

1. a (textual) message passed between two or more agents or
2. some (physical) action that an agent performs publicly.
Social Performance Measurement (III)

Advantages:

1. By measuring communication we abstract from non-communicative properties of the system → immense reduction of global system complexity
2. We can exploit knowledge-level representations between knowledge-based agents
3. We are able to measure performance in open systems (where internal agent data cannot be accessed)

Prerequisite for defining measures: definition of an underlying MAS model.
A Generic MAS Model
A Generic MAS Model

- MAS consists of a set of agents $\mathcal{A} = \{a, b, c, \ldots\}$ capable of communicating in a high-level ACL (here: FIPA-ACL)
- Semantics of the ACL are common knowledge and obeyed (benevolence is not implied by this)
- MAS has to perform tasks from $\mathcal{T}$ that enter the system at arbitrary points in time
- A real-valued measure

$$d : \mathcal{T} \rightarrow \mathbb{R}$$

for the difficulty of the tasks is available
A Generic MAS Model

Top-level processing cycle:

- task
- negotiation
- agreement
- execution
- completion
- conflict
- failure
Communication (I)

Subset of FIPA-ACL messages of the form

\[ \text{performative}(\text{sender}, \text{receiver}, \text{content}) \]

where

\[ \text{performative} \in \{ \text{inform}, \text{inform\_done}, \text{inform\_refuse}, \text{agree}, \text{accept\_proposal}, \text{request}, \text{cfp}, \text{reject\_proposal}, \text{propose}, \text{failure}, \text{not\_understood}, \text{refuse} \} \]

We measure all quantities wrt total number of messages

\[ TNOM = |M| \]

where the set of all messages is

\[ M = \{ m_1, m_2, \ldots m_n \} \]
Communication (II)

Further definitions:
Partitioning messages according to tasks in

\[ M_T = \{ m_{T_1}, \ldots m_{T_n} \} \]

where \( T \in \mathcal{T}_{curr} \) (set of processed tasks=\( \mathcal{T}_{curr} \subseteq \mathcal{T} \)) such that

\[ M = \biguplus_{T \in \mathcal{T}_{curr}} M_T. \]

Partition tasks into “successfully completed” and “failed” tasks:

\[ \mathcal{T}_{curr} = \mathcal{T}_{succ} \uplus \mathcal{T}_{failed} \]
Defining Social Performance Measures
Basic Performance Measures (I)

Very simple: counting messages and message types

1. \( MPTD = \text{messages per task and difficulty} \):

\[
MPTD = \sum_{T \in \mathcal{T}_{\text{curr}}} \frac{M_T}{d(T)}
\]

2. \( \text{fail-fast version} \ (0 \leq \alpha \ll \beta \leq 1) \):

\[
\text{ff}_-MPTD = \alpha \cdot \sum_{T \in \mathcal{T}_{\text{succ}}} \frac{M_T}{d(T)} + \beta \cdot \sum_{T \in \mathcal{T}_{\text{failed}}} \frac{M_T}{d(T)}
\]
Basic Performance Measures (II)

Less trivial: we can define types of performatives. Assume partition

- \( Type1 \) = \{request\},
- \( Type2 \) = \{inform, inform_done, inform_ref\}
- \( Type3 \) = \{cfp, propose\}
- \( Type4 \) = \{reject_proposal, refuse\}
- \( Type5 \) = \{accept_proposal, agree\}
- \( Type6 \) = \{not_understood, failure\}

\( \rightarrow \) We can count occurrences of these types
Basic Performance Measures (III)

We obtain as further measure the *mean message type usage* $\text{MMTU}$:

$$\text{MMTU}(x) = \frac{1}{|\mathcal{T}_{\text{curr}}|} \sum_{T \in \mathcal{T}_{\text{curr}}} \frac{\{m \in M_i | \text{type}(m) = x\}}{|M_T|}$$

for $x \in \{\text{Type1}, \ldots, \text{Type6}\}$ Trivially $\text{MMTU}$ subsumes *mean performative usage*.

Visualisation: *message type partition chart*
Basic Performance Measures (IV)

Example:

```
Type1
Type2
Type3
Type4
Type5
Type6
```
Basic Performance Measures (IV)

$\textit{MMTU}$ can be refined through parametrisation, e.g.

$$\textit{MMTU}(x, d_1, d_2) = \frac{1}{|\mathcal{T}_{\text{curr}}[d_1 : d_2]|} \sum_{T \in \mathcal{T}_{\text{curr}}[d_1 : d_2]} \frac{|\{m \in M_i | \text{type}(m) = x\}|}{|M_T|}$$

where

$$\mathcal{T}_{\text{curr}}[d_1 : d_2] = \{T \in \mathcal{T}_{\text{curr}} | d_1 \leq d(T) \leq d_2\}$$

Alternatives to difficulty: subsets of agents/agent types, spatial network regions, etc.
Complex Message Patterns (I)

- Simple measures don’t allow for *syntactic analysis* of communication

**We introduce message patterns to abstract from message sequences**

- Use variables $A, B, C \ldots$ for messages or message fields (participant, content) and $\star$ as a wildcard symbol

- **Examples:**
  \[
  p = [\text{accept}(A, B, \text{do}(A, X)), \star, \text{do}(A, X)]
  \]
  \[
  q = [\text{accept}(A, B, \text{do}(A, X)), (\neg \text{do}(A, X))^n]
  \]

- Define $\text{matches}(m, p)$ as a boolean function that matches message sequence $m$ against pattern $p$
Complex Message Patterns (II)

Pattern-based measures:

- Average length of $p$:

$$\text{mean\_length}(p) = \frac{1}{|\mathcal{T}_{\text{curr}}|} \sum_{\text{matches}(m,p) \land m \in M} \text{length}$$

- Task-relative version:

$$\text{mean\_length}(p) = \frac{1}{|\mathcal{T}_{\text{curr}}|} \sum_{\text{matches}(m,p) \land m \in M} \frac{\text{length}}{|M|}$$

- Frequency of occurrence:

$$\text{frequency}(p) = \frac{|\{m \in M | \text{matches}(m,p)\}|}{|M|}$$
Complex Message Patterns (III)

Examples:
Let

\[ p_a = \bigcap_{Q \in A} [\text{propose}(P, Q, X), *, \text{accept}(Q, P, X)], \]
\[ p_c = \bigcup_{Q \in A} [\text{propose}(P, Q, X), *, \text{reject\_proposal}(Q, P, X))] \]

Then mean time to agreement (MTTA) and mean time to conflict (MTTC) can be defined as

\[ MTTA = \text{mean\_length}(p_a), \]
\[ MTTC = \text{mean\_length}(p_c) \]

→ Usefulness of measure definitions in applications depends on interaction protocols
Example
Example (I)

Comparison of the Contract-Net Protocol with the Contract-Net-With-Confirmation Protocol:
Example (II)

- Problem: find internal attribute that best predicts *assigned tasks ratio* (ATR) where ATR(CNP)=0.65 and ATR(CNCP)=1.0
- Evaluation of *TNOM* (20130/30768 resp.) offers no explanation
- Neither does evaluation of *MMTU*:

![Pie charts showing distribution of responses](image)
Example (III)

• Therefore, we measure the frequency of “rejected unconditional proposals”

\[ p = [\text{propose}(A, B, X, \{\}), *, \text{reject}_\text{proposal}(B, A, X)] \]

• Result: \( \text{frequency}(p, \text{CNP}) = 0.35 \), \( \text{frequency}(p, \text{CNCP}) = 0.0 \)

⇒ Superiority of CNCP is due to avoidance of “hasty allocations”

⇒ Complex measure explained what simple measures couldn’t explain
Conclusions & Outlook
Summary

- We suggested social performance measures for MAS based on measuring communication processes.
- Claimed that these aid complexity reduction while correctly predicting external from internal attributes.
- Main feature: abstraction from intra-agent reasoning processes and physical execution.
- Exploiting knowledge-level representations intelligent agents use.
- Defined concrete measures, illustrated usefulness with examples.
Outlook (I)

• Development and critical evaluation of further generic measures (in particular, standardised patterns)
• Further refinement of measures (advanced statistical methods)
• Definition of qualitative semantic measures:
  • classification into processing phases
  • analysing social relationships
  • logical properties - view communication as
    • distributed search
    • constraint satisfaction processes
    • distributed theorem proving
Outlook (II)

- Investigation of relationship external ↔ internal attributes
- Explore combination of social-level and cognitive-level measurements
- Explore more complex social phenomena (group formation, socialisation, power, authority, etc.)
- Develop tools for automatic measurement of complex MAS
Thank you for your attention!