



Empirical Software Engineering for Agent Programming

M. Birna van Riemsdijk – II, TU Delft, The Netherlands

Acknowledgement: Koen Hindriks

10/22/12



Does analytical research suffice to advance state-of-the-art in agent programming?

- Complex real life issues involving **humans and their interaction with technology** cannot be answered through a solely analytical approach
- **Empirical research** needs to complement theoretical studies to advance state-of-the-art in agent programming



Empirical Software Engineering (1)

Basically: letting people **use** the technologies and see what happens ...

... in a **principled and systematic** way!

Empirical Software Engineering (2)

Gather **data** for:

1. getting a better understanding of **how** software developers **use** the technology (problems and possible solution patterns)
2. **within technology comparison**: demonstrate that one variant of (using) the technology is better than another
3. **improving** the technology based on these findings
4. **across technology comparison**: demonstrate that the technology is better than some other technology

Research Questions (1)

- how do programmers use agent-oriented languages?
 - which constructs do they use?
 - what is expressed using which constructs?
 - what kind of patterns do they use?
 - which problems do they experience while programming?
 - which aspects of the languages do they find difficult or easy to understand?
- which ways of using agent-oriented languages improve agent software quality?
 - how should constructs be used?
 - which patterns and anti-patterns can be distinguished?

Research Questions (2)

- how does the use of an agent programming language compare with other paradigms?
 - do similar programming patterns emerge?
 - how does the speed of software development compare?
 - how do the resulting programs compare with respect to software quality measures like efficiency, maintainability and readability?
- which features should an Integrated Development Environment for agent programming have?
 - how do programmers use existing IDEs?
 - to what extent do requirements for an IDE for MAS engineering differ from those for mainstream software development?
 - which approaches for debugging are needed in the context of agent programming?

What do we mean by “better” software?

ISO/IEC 9126 Standard: Software Quality Model

- Functionality – interoperability, functionality compliance, ...
- Reliability – fault tolerance, recoverability, ...
- Usability – understandability, learnability, ...
- Efficiency – time behavior, ...
- Maintainability – analyzability, testability, ...
- Portability – adaptability, co-existence, ...

How to measure software quality (ISO/IEC 9126 standard)?

- For each of the sub-characteristics: define a set of **attributes** that can be **verified or measured** in the software product
- Define a set of **quality metrics** which evaluates the degree of presence of quality attributes
- Internal metrics (static), external metrics (running software), quality in use metrics (use in real conditions)

=> attributes and metrics **vary** between technologies/
software products!

Translation to engineering MAS

- Which ISO software quality characteristics are **suitable for MAS**?
- Which ISO software quality characteristics are particularly **important** (problematic or strength) **in MAS**?
- Are there any **MAS-specific** software quality **characteristics**?
- Which **MAS-specific attributes and metrics** can be defined for measuring the characteristics?

How to gather and analyze data: empirical research methodology

- Quantitative research
 - test pre-determined hypotheses and produce generalizable results (statistics)
 - answering mechanistic 'what' questions
 - test hypotheses
- Qualitative research
 - illumination and understanding of complex psychosocial issues
 - answering humanistic 'why?' and 'how?' questions
 - identifying possible hypotheses

Case Study Research

- qualitative
- range from very ambitious and well organized studies in the **field**, to small **toy examples** that claim to be case studies
- criticized for being of less value, impossible to generalize from, being biased by researchers etc.
- applying **proper research methodology practices** as well as considering that knowledge is more than statistical significance

P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14(2):131–164, 2009.

Existing Work

- A few isolated papers
 - Van Riemsdijk, Hindriks (PRIMA'09, PRIMA'10)
 - Benfield et al. (AAMAS'06)
 - Gemrot et al. (CAVE'12)
 - Jordan & Collier (ProMAS'10)
 - Sollenberger & Singh (AAMAS'10)
 - Telang & Singh (AAMAS'12)
 -?

How to build a culture of empirical research in the AAMAS community?

- **community** should agree on/develop empirical research methodologies for engineering MAS
- **researchers** should apply and explain research methodology and aims of research in paper
- ensure that **reviewers** are aware and use the established agreement in community on what counts as good empirical research for engineering MAS

Take Home Messages

- We should increase use of **empirical methods** in research on engineering MAS to complement theoretical studies

=> Development based on **data**!

- As a **community** we need to come to an agreement on what counts as good empirical research