

Requirements Elicitation for MAS with Activity Theory

Rubén Fuentes Fernández, Jorge J. Gómez-Sanz,
Juan Pavón Mestras

Dept. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain

{ruben,jjgomez,jpavon}@sip.ucm.es

12/05/2004

<http://grasia.fdi.ucm.es>



Summary

- ☞ Introduction
 - Requirements Elicitation
 - Social Theories
- ☞ Activity Theory
 - The Activity Checklist
- ☞ Social requirements for MAS
 - Adapting the Checklist
 - Elicitation method
- ☞ An example
- ☞ Conclusions

12/05/2004

<http://grasia.fdi.ucm.es>

2

Requirements elicitation

- ☛ Requirements are about:
 - High-goals of the envisioned system
 - their operationalization into specifications
 - With services and constraints
 - and their assignment to humans or devices.
- ☛ Modelling techniques influence RE.
 - What can be
 - Modelled
 - Observed
 - Discussed

RE with agents

- ☛ Some RE proposals are based on the agent paradigm
 - i*
 - KAOS
 - Albert
- ☛ The agent paradigm proposes
 - A different way of thinking about systems.
 - For paying customers, end-users, and developers.
- ☛ Why not using agents social features for RE?
 - What models can provide us further details?

Social models: Activity Theory

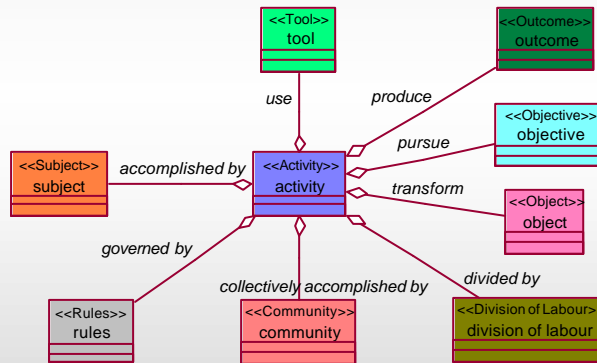
- Activity Theory is a framework to study
 - social activities
 - with a historical-cultural view
 - evolving through contradictions.
- Is it applicable?
 - Human societies and MAS share a common base
 - Social
 - Intentional

Activity Theory for MAS

- Our contribution
 - Formalization of AT concepts.
 - Application of AT to assist AOSE.
 - Support tools.
- This presentation focuses on
 - Requirements Elicitation with Activity Theory.

Activity

- An *activity* is a process.



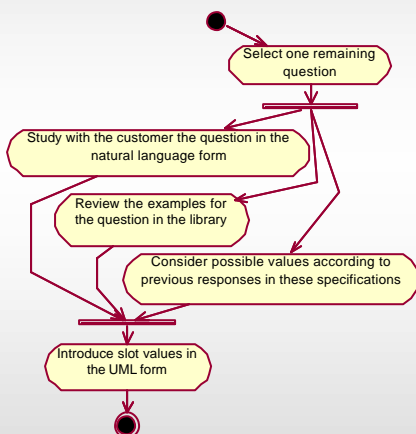
RE with the Activity Checklist (I)

- The Activity Checklist is an AT tool that can be used for RE in MAS.
- The Checklist expresses requirements
 - grouped in *areas*,
 - which contains *aspects*,
 - which are formulated as *questions*
 - in natural language.

RE with the Activity Checklist (II)

- Use our UML formalization for AT to express questions.
- It facilitates further processing.
 - Validation of a specification.
 - Verification of properties.
 - Traceability of requirements.
 - Integration into MAS development tools.

Information gathering



- Collaboration customers-developers.
 - Natural language describes searched information.
 - Library of examples helps to understand the question.
 - Other values in the specification are available.
- Answer recorded in the frame of the question.
 - Less ambiguous.
 - Automated processing.

Example: agents in Robocode

- Robocode
 - <http://robocode.alphaworks.ibm.com>
 - Simulated tank battles between teams.
 - Tanks.
 - Perceive the environment with sensors.
 - Have primitives for movement and shooting.
 - Goal → Survive team.
- Requirements about modelling the team leader.
- Full specification in <http://grasia.fdi.ucm.es/ingenias>

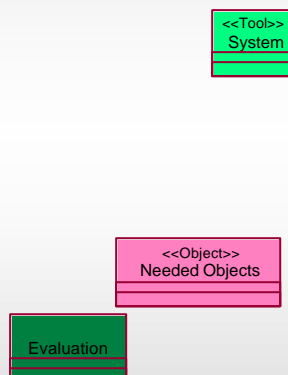
Simulating strategies (I)

- The team leader
 - Conceives the strategy.
 - Gives the commands.
- Requirements.
 - We want to capture aspects of the intelligence of the *team leader* agent.
 - What features we should consider?

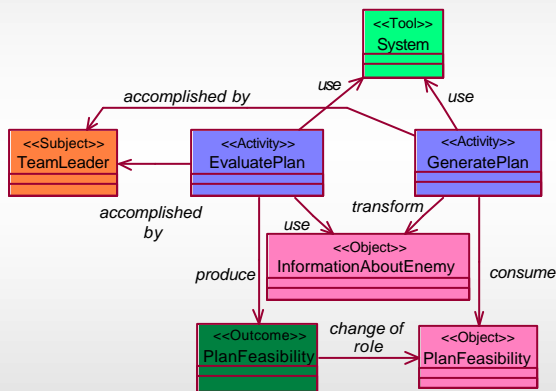
Simulating strategies (II)

- Area → Learning/cognition/articulation.
 - Aspect → Possibilities for simulating target actions before their actual implementation.
 - Question → Should the system analyze that the action can be done before it is really done? For example checking that all the needed resources are available.
 - Question → Does the system provide clues in case of breakdowns/failures that can be used to find a solution?

UML representation of questions

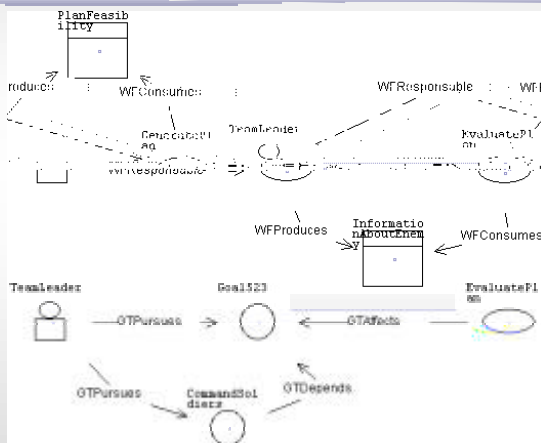


Customer's information



- Answers reusing previous information.
 - TeamLeader
 - GeneratePlan
- Partial answers
 - System
- New entities.
 - PlanFeasibility
 - EvaluatePlan

Translation to INGENIAS



- Automated translation to the agent paradigm.
 - Mappings.
- New requirements generates pending new entities and relations.
 - Goal523
 - GTDepends

Conclusions

- AT is proving its utility in several development tasks for MAS.
 - Requirements elicitation in this case.
 - But also contradictions, properties discovering, development management...
- Activity Checklist as a tool to manage social properties.
 - Social concepts known for users and developers.
 - Dual form
 - Process.
- Results are being integrated within the Ingenias I DK
 - <http://ingenias.sourceforge.net>
- Future work:
 - Enrichment of the Checklist.
 - Adaptation to other MAS methodologies.