



MAPPER

Model-based Adaptive Product and Process Engineering

Work Package WP1 – WP6:
The requirement engineering approach

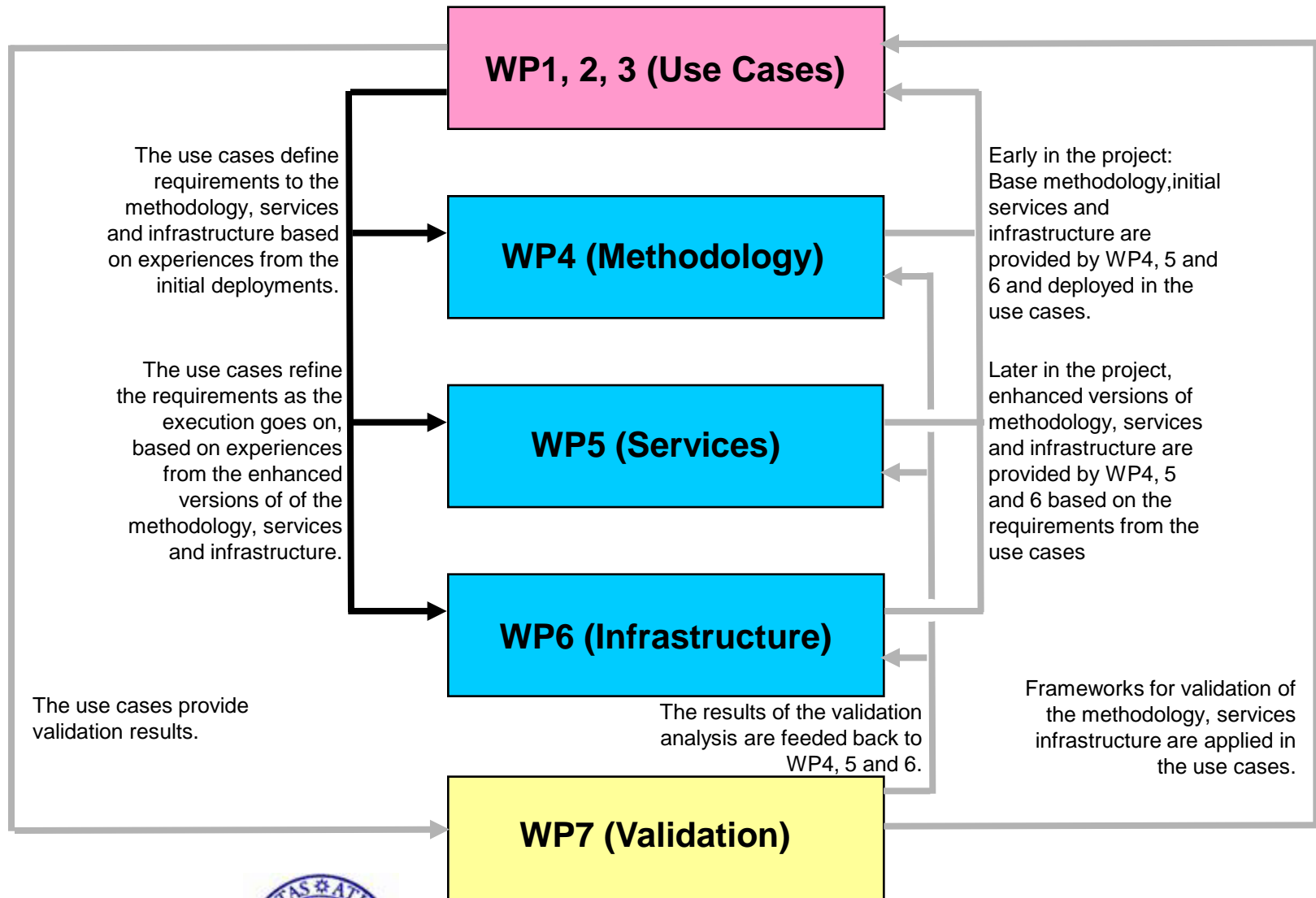
Svein G. Johnsen
Gianni Jaccuci



Outline

- The role of requirement engineering
- The requirement modelling in the use cases
- Extracting requirements for the technical workpackages
 - The Framework for Methodology Requirement Engineering (FMRE)
- Uncovering new requirements
- Including social aspect
- Designing social practice in Mapper
- Conclusions

The flow of requirements



Requirements Modeling in the Use Cases

Starting point:

- Reconfigurable enterprise models capturing best practices and developed for use case execution (solution model)

Objectives

- Identify and elaborate requirements from the individual use case regarding MAPPER infrastructure and methodology
- Capture the requirements as enhancement of the solution model



Requirements Modeling Process in Use Cases

- Use the same roles as in enterprise modeling (facilitator, tool expert, coach, modeler, domain expert)
- Modeling workshop:
 - Agree on AMPS classification (approach, methodology, platform, solution)
 - Walk through solution model task-by-task and identify AMPS requirements
 - capture requirements in task pattern sub-model as enhancement of solution model
- Facilitator: elaborate requirements and revise the model
- Modeling workshop: Recapitulation and refinement
- Facilitator:
 - Add requirements matrix
 - Description of requirements in D4/D5/D6
 - Introduce requirements sub-model



Experiences from Use Case

- AMPS classification
 - Initially was questioned regarding difference between
 - Approach and methodology
 - Platform and solution
 - more and more accepted when requirements got more detailed
- Participatory approach crucial
- Difficult to find adequate granularity level for methodology requirements
- Visual modeling extremely helpful
- Requirements model as enhancement of solution model proved practicable and useful



Extracting requirements for each technical workpackage (The Wp4 as an example)

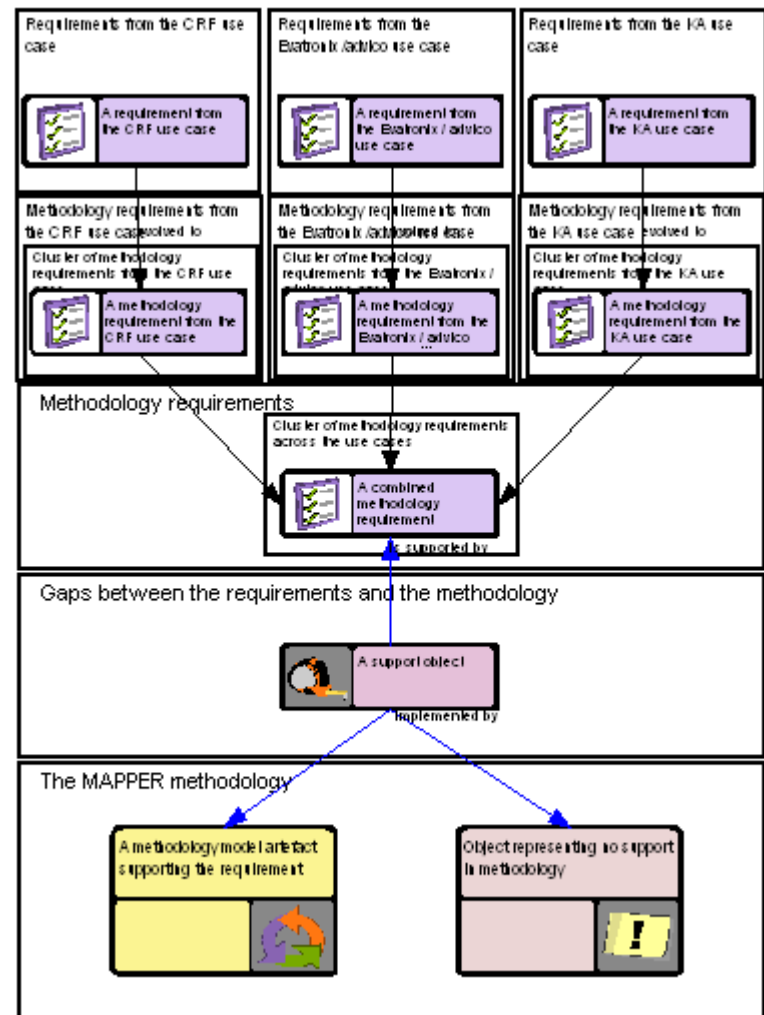
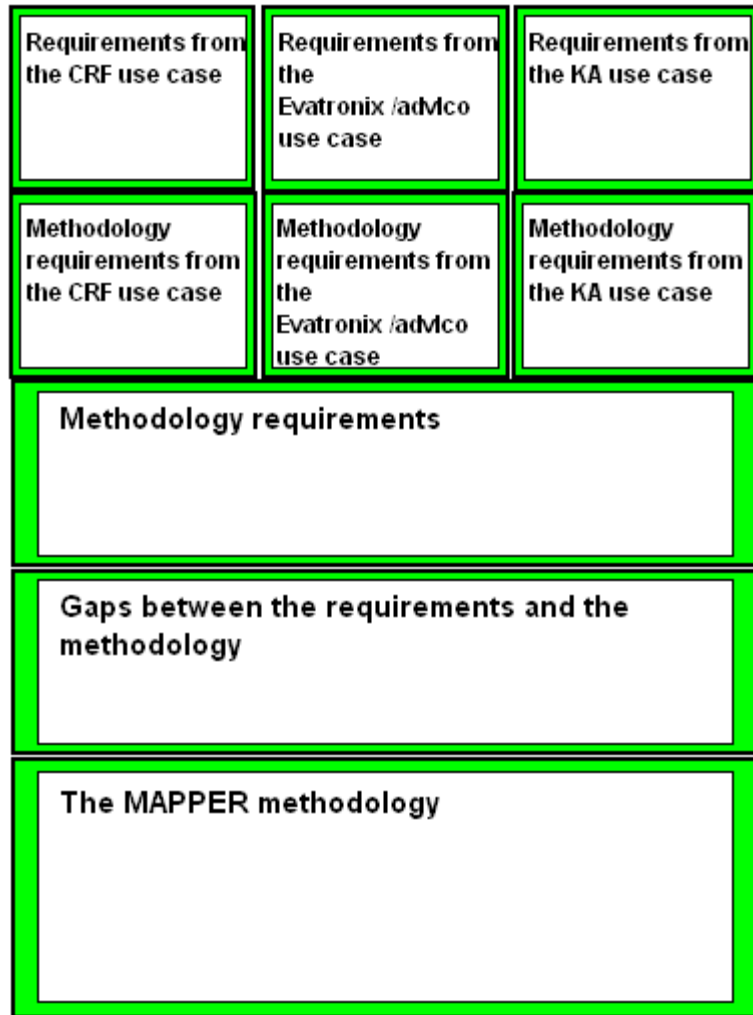
- The methodology requirement engineering was based on the Framework for Methodology Requirement Engineering (FMRE)



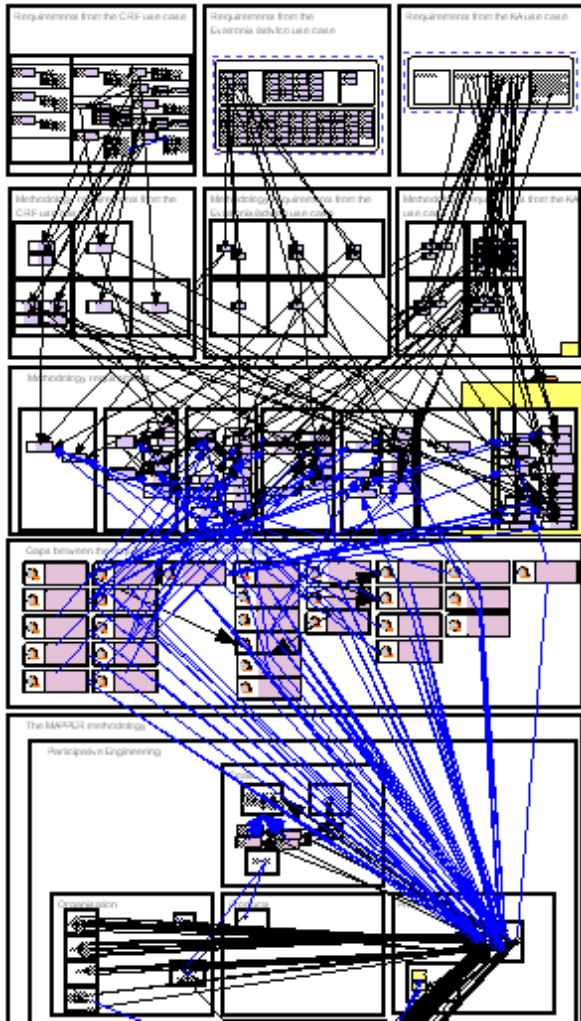
The FMRE

- Dynamic representation organized in a structure of 5 layers:
 - The use case specific requirements
 - The use case specific methodology requirements
 - The combined methodology requirements
 - The assessment of methodology support
 - The methodology

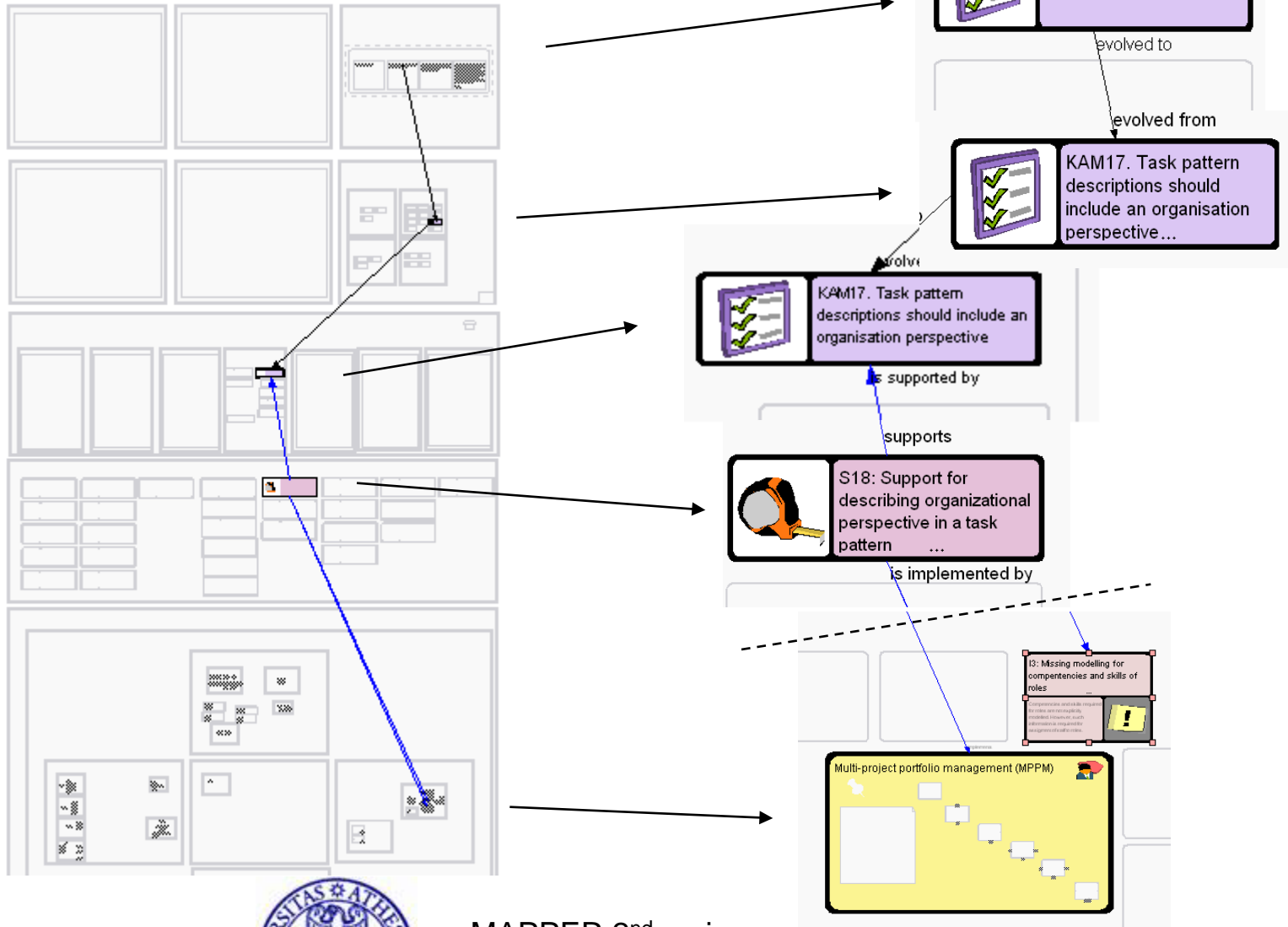
The FMRE



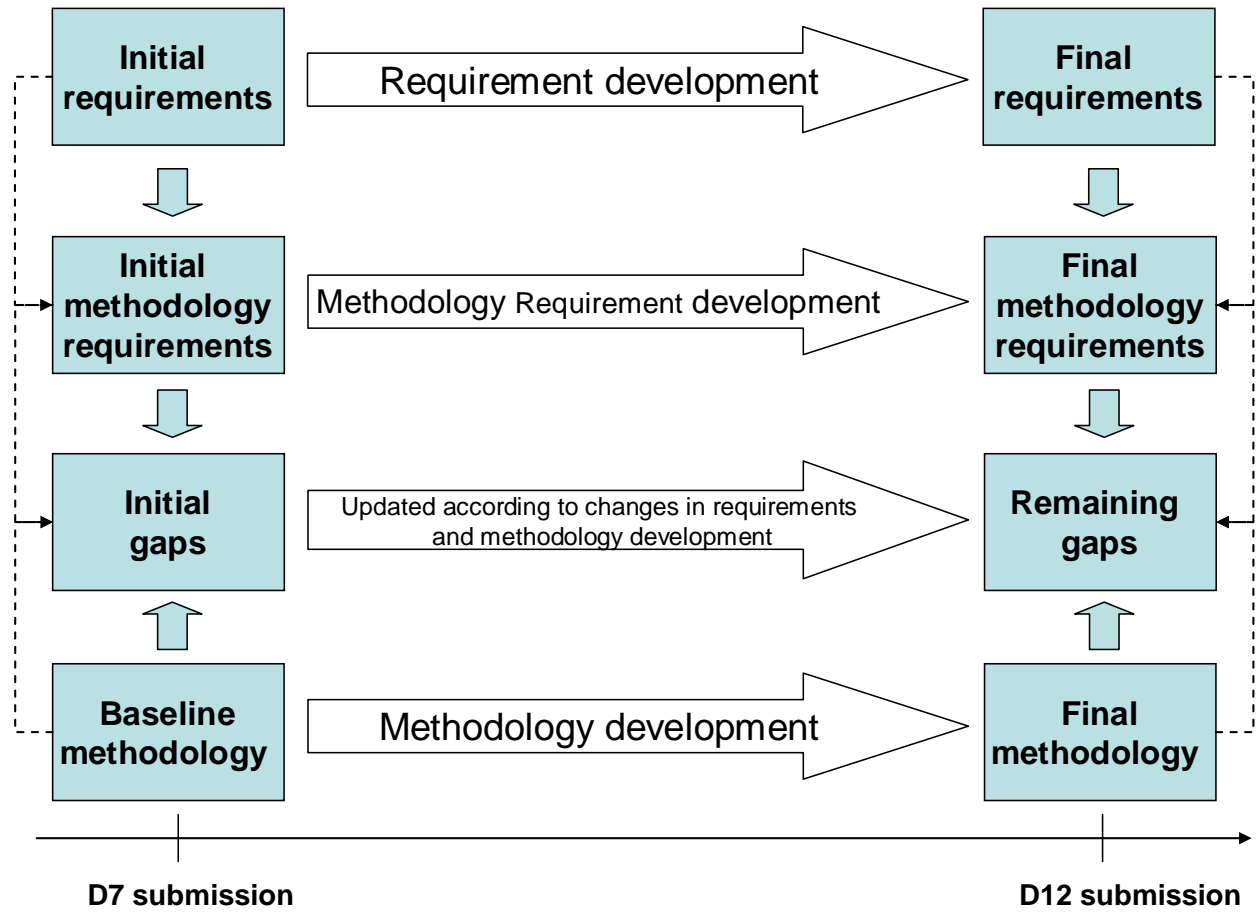
D7 model in FMRE



One example



Evolution over time



Uncovering new requirements

- Piloting, utilizing mode-configured solutions, will uncover concrete requirements.
- This will help us prioritize among existing requirements through direct, concrete dialog between users and developers.

including social aspects

- linking use case practice to requirements, articulating both the modelling approach and the design of social practice
- i.e., a new element: articulation of the sociotechnical approach in MAPPER, *'making place' for IT*, with own method and products
- this yields two better identified streams of activities:
(a) technical (analytic) (b) social (holistic)
- characterised by two types of user involvement
(a) analytic requirement elicitation (b) holistic mock-up co-evolution
- with two different types of products:
(a) requirements / models (b) visions /solutions (mock-ups)
- thus delivering a coherent, stronger, integrated approach and methodology to designing people, computers and work

designing social practice in MAPPER

- a. the ethnographic approach method
- b. examples of visions development by the ethnographic method in MAPPER

a. the ethnographic approach method

1. ethnographic field studies let emerge how questions (by concept development) indicating **goals** to be attained
2. theories of the social provide **principles** guiding social practice towards social goals
3. in the innovation phase, creative design develops **visions** indicating how to attain principle related goals that resolve the 'how' questions

b. examples of visions development by the ethnographic method in MAPPER

- 1. issue lists:** how can we promote participation and overcome barriers, the issue of time, group work theory and membership, innovative vision: my page, tech – individual pages in excel, people - group leadership to empower group members and enforce group trust
- 2. long emails:** how can we better support clients, the issue of asymmetric initiative dialogues, group work theory and effective communication, innovative vision: offering training sessions to users, tech: technology supported learning, people: organise training sessions
- 3. extended logo:** how can we foster the process of innovation, creative design principle of keeping in focus the original concept in innovation design, innovative vision: invent an extended logo populating all work rooms in POI phases, tech: extended logo, people: elicit connection with origin

Conclusions

- The model based requirement engineering approach used in combination with the FMRE has proved to be practicable and useful.
- Ethnography based social practice design adds new requirements regarding social practice execution
- Mock-ups allow the anchoring of solution visions co-constructed with users

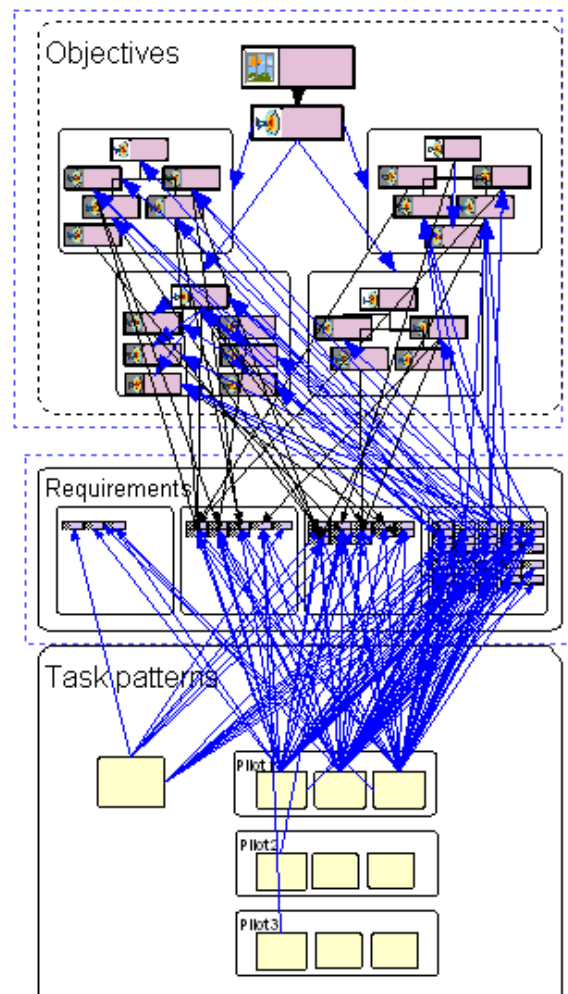
Additional material

- If needed...

The requirement classification

- *Approach* requirements, relevant in cases where several alternative approaches exist for an activity in the use case, e.g. line vs. matrix organisation of projects.
- *Methodology* requirements include any requirement related to methodology use and design, e.g. required extensions or refinements of the baseline methodology.
- *Platform* requirements include requirements with respect to the MAPPER infrastructure, additional IT infrastructure in the use case or information resources (document templates, model templates, instructions, guidelines, etc.).
- *Solution* requirements concern implementation characteristics of the solution for a specific project. An example could be which service to use for a specific project if several equivalent services exist.

The requirements matrix (from D6)



Task pattern definition

- *A task pattern is a self-contained model template with well-defined connectors to application environments capturing knowledge about best practices for a clearly defined task, where*
- *“Self-contained” means that a task pattern includes all perspectives, model elements and relationships between the model elements required for capturing the knowledge reflecting a best practice. Examples for perspectives and model elements are tasks to be performed and their interrelations, roles involved and their competences, and systems and information resources involved and their interfaces.*
- *“Model template” means that the task pattern has to be expressed in a defined modelling language (preferably a (visual) enterprise modelling language) and that no instances are contained in the task patterns, i.e. no real actors, documents or IT systems.*
- *“Connectors” are model elements facilitating the adaptation of the task pattern to target application environments, i.e. only the connectors may be adapted. Other model elements of the task pattern may only be configured, i.e. instances are assigned to the model elements.*
- *“Application environments” currently are limited to enterprise models. It should be investigated, whether this can be extended to process models.*