M@RT Breakout-Sessions:

- Adaptation & Reasoning (Ben)
- What‘s a M@RT? (Katmandu)
  - Executable, interpreted; relationship between model & system?
- Goal-driven, high-level models (Everest)
- Applications of M@RT
  - Uses; Killer-Apps; Mapping to techn(?)
- To do:
  - Catch phrase of state of art
  - What we know
  - What we don‘t know
  - One key question for M@RT

M@RT Breakout-Sessions: Applications of M@RT

- Applications of M@RT
  - Uses; Killer-Apps; Mapping to techn(?)

- Catch phrase of state of art

- What we know
  - M@RT should use models to make certain aspects of a system flexible and adaptable.
  - Flexibility depends on the kind of system and its uses.
  - Thus, we have many different kinds of models and purposes @ RT.
- What we don‘t know
  - What‘s the difference between flexibility & evolution?
  - What‘s the difference between flexibility & variability?
  - How these fit together in a sound and robust way?

  - Will models become useful assets on their own now?

- One key question for M@RT
  - How to make QM?
  - Applicable to physical / organisational/… systems other than software?

M@RT

- 2 ways for M@RT:
  - A) At start/restart of the system: read model as configuration
  - B) While system is running: adapt the model
    - B1) Adapt through user
    - B2) Self-adaptation according to changing circumstances

- Applications now (varying adaptivity…):
  - DBMS (know nothing about their data structures)
  - Protocols, WebContainers etc… (configured by models?)
  - SAP;

- Applications in Future, examples:
  - Using Design Models, eg. For buildings/city infrastructure to create management models for the RT physical system
  - Operative (workflow) models to monitor/guide activities e.g. construction, production, bureaucracy ,