Task modelling techniques and their assessment

SYBILLE CAFFIAU
PATRICK GIRARD
POITIERS – FRANCE

DOMINIQUE SCAPIN
INRIA – FRANCE

Our contribution to MBUI (1)

- The K-MAD model and the K-MADe tool
  - Result of years of work on the subject (MAD, MAD*, …)
  - A kernel, which has been designed to allow « plugging » extensions for specific needs
  - A hierarchical task model, which includes domain objects and computable conditions (pre-conditions, post-conditions)
  - An enhanced simulation tool, which allows syntaxic and semantic versifications, and manages conditions.

Our contribution to MBUI (2)

- The Dialog Hierarchy model and model-based tools
  - Result of years of work on the subject (Hierarchical Interactors, Diagets, MB Tools, …)
  - A dialogue model (Abstract User Interface), which has been designed to allow a specific description of dialogue
  - A library, which allows easy implementation of such dialogues

Our contribution to MBUI (3) - Applicability

- Assumption: One of the most interesting advantage of task-based model based approach
  - Automating semantic validation of applications
- Case studies
- A proposal for model-based co-design of Task models and Abstract User Interfaces
  - Based of large assessment of models
  - By way of model meta-modelling
  - Based on rules between meta-models

1- Case studies

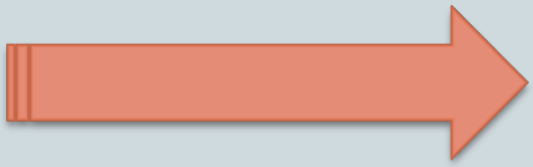
- Several case studies, which allow exploring the limitations of model usage
  - Complete case studies
    - Mastermind
    - Volley-Ball marking sheet
    - Biological analysis laboratory
  - Open case studies
    - Emailer
    - Smartphone
    - …

2- Proposal

- Model-based co-design of Task models and Abstract User Interfaces

1. Development cycles in HCI

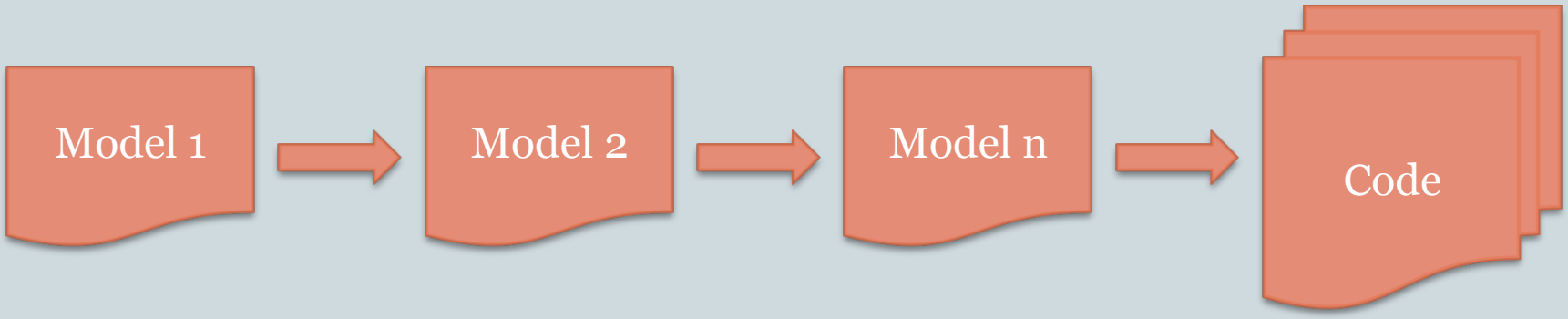
- Prototyping
- Participatory Design
- Evaluation



Iterative cycles

2. Model-Driven Approaches

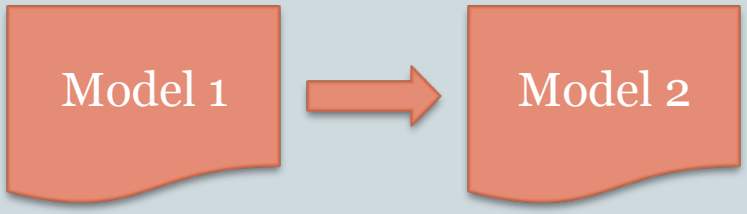
- Promote usage of models in software design
- Main usage of generation



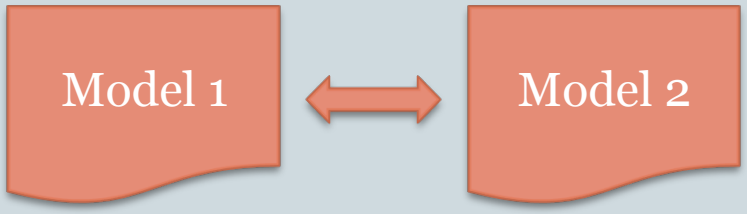
How does it fit with iterative cycles ?

Our proposal

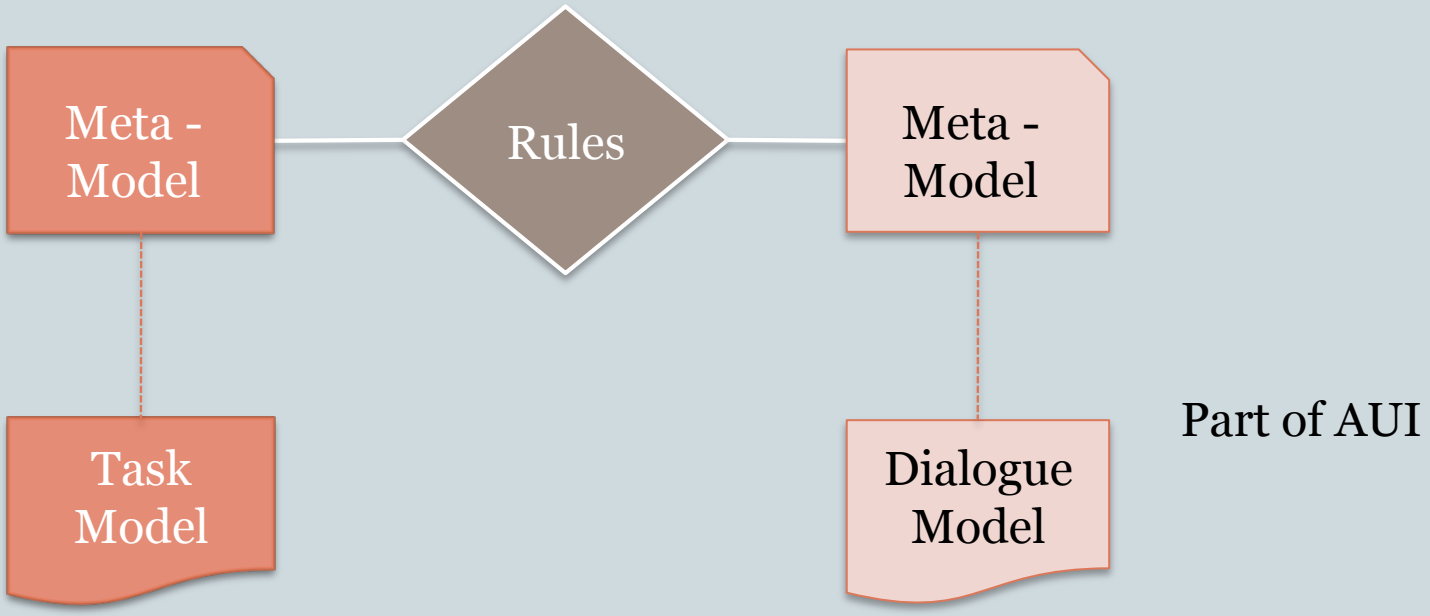
- Instead of building "one way" tranformations…



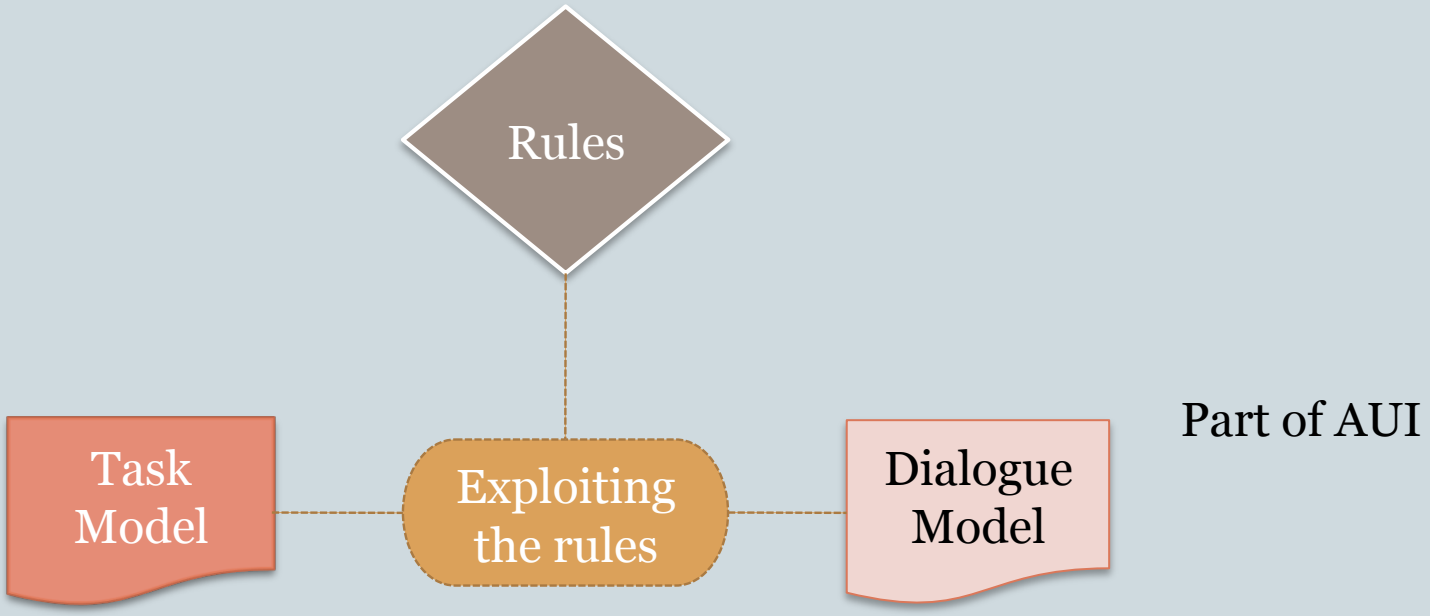
- Elaborating "bidirectional" transformations



Our method: first step



Our method: second step



Summary

- Allows generation
- Allows verifications between models
- Allows iterative design

- Experienced with two specific models
  - K-MAD for task models
  - Hierachical Interactor for dialogue models
- Seems to be usable with other models
  - All hierarchical task models
  - All state-based dialogue models
- To be defined on standards

Future works and open questions

- Future works
  - Implementing rules in tools (Eclipse ?)
  - Study how it can be used with other formalisms
- Open questions
  - Link with the Domain Model ?
  - Complete definition of the Abstract User Interface ?
  - Link with the Concrete User Interface ?