

# Executable Models for Human-Computer Interaction: Meta-Modeling Ubiquitous User Interfaces

Grzegorz Lehmann  
grzegorz.lehmann@dai-labor.de  
www.dai-labor.de

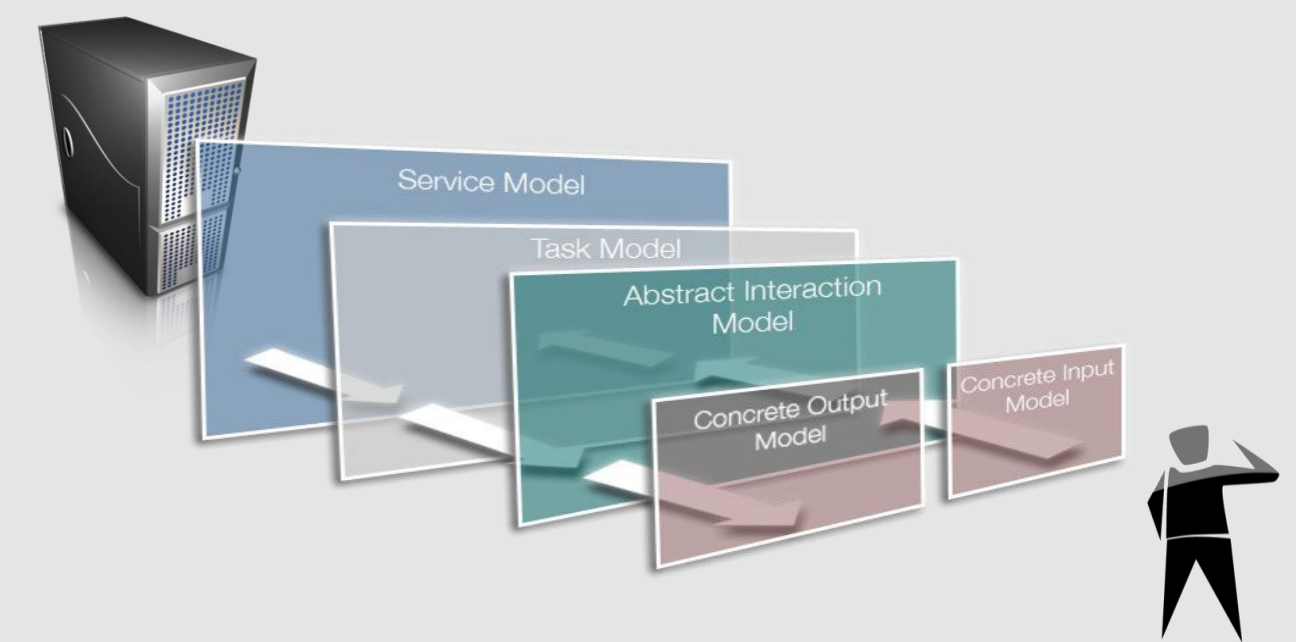
## Motivation

Applications in ubiquitous environments need to adapt dynamically to context of use situations unforeseeable at design time. Ubiquitous user interfaces must not only adapt to multiple situations, but also to multiple users, devices and modalities. Final user interface code generated from models at design time fails to provide the required flexibility, as the design rationale is no longer available at runtime.

## Approach

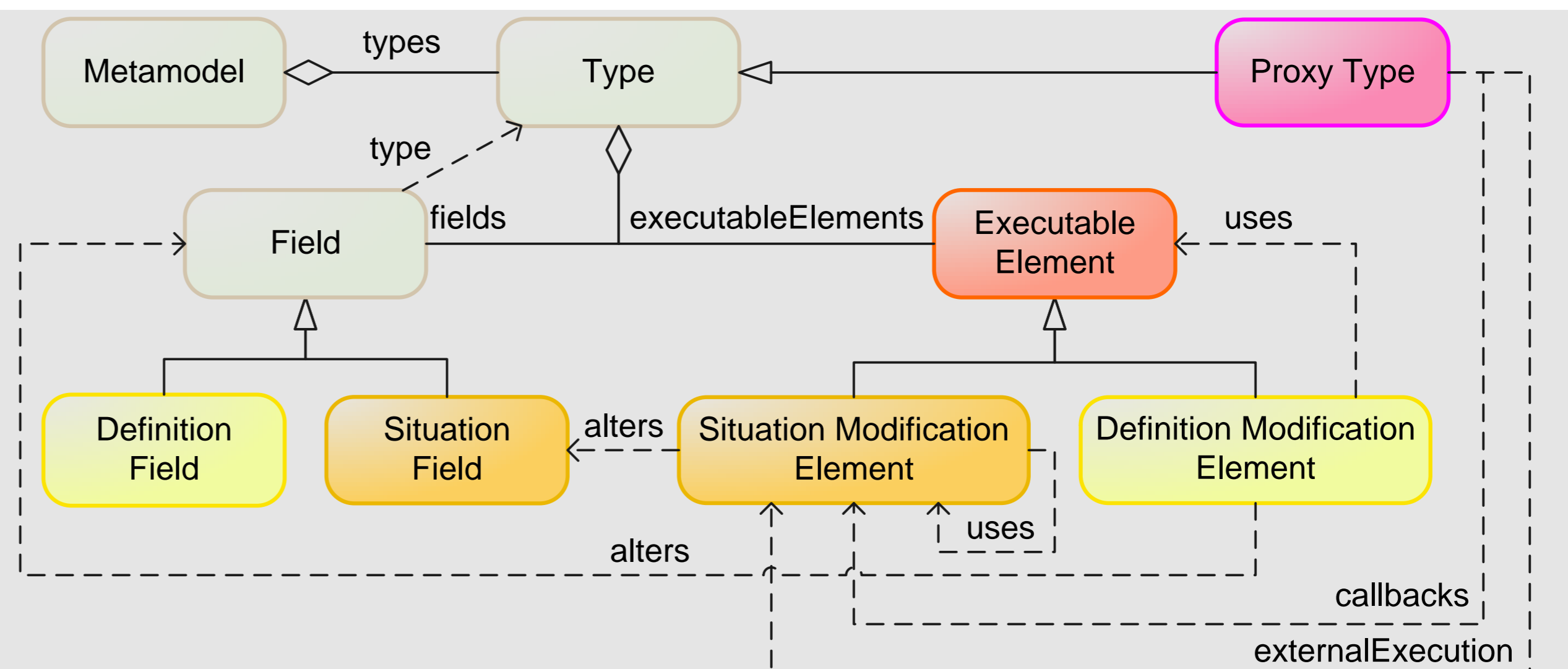
This work deals with the design of executable user interface languages and the utilization of user interface models at runtime. Enhanced with runtime concepts, the user interface models enable the reasoning about the user interface, its state and the decisions of the designer at runtime on a high level of abstraction, as well as provide means for their consistent modification. An explicit definition of the design time rationale, the state information

and the execution logic in the metamodels enables a dynamic adaptation of model-based user interfaces to contexts unforeseeable at design time.



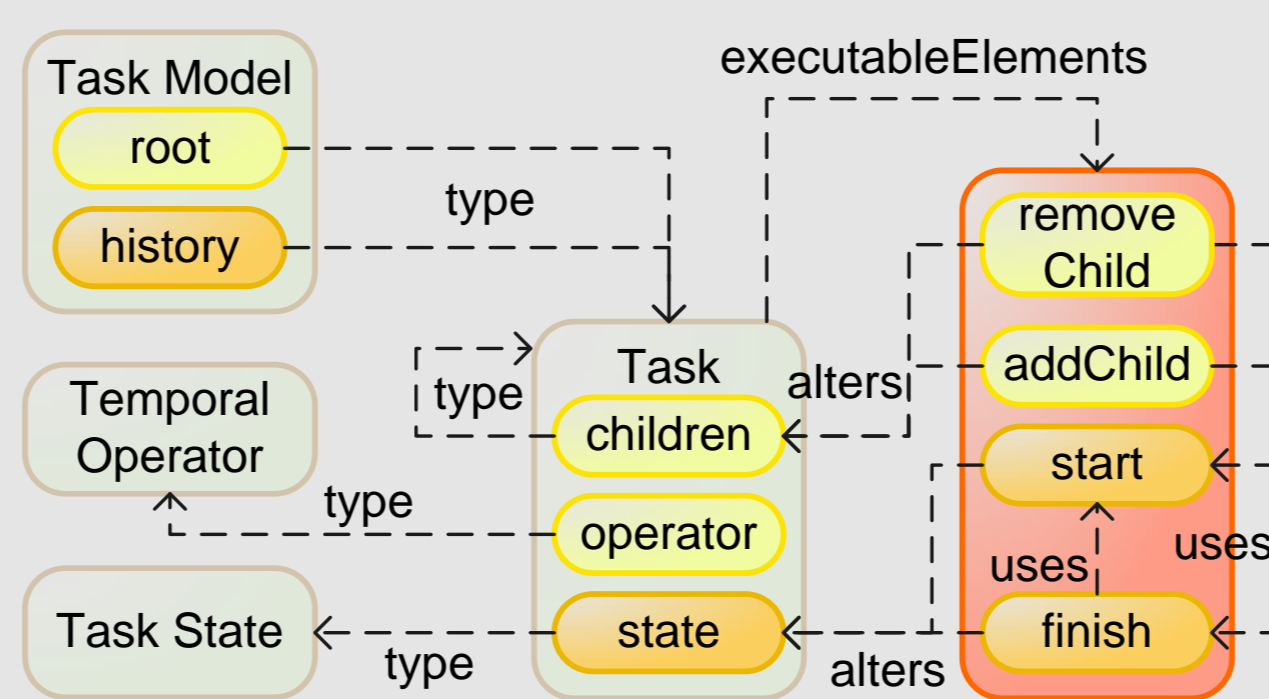
## Goal 1: Meta-Metamodel

The meta-metamodel provides basic building blocks for user interface languages defined in form of conforming metamodels. The constructs of the meta-metamodel distinguish the situation and definition parts of runtime models (*Situation* and *Definition Fields*) and identify their execution and adaptation logic (*Situation* and *Definition Modification Elements*). The *Proxy* type classifies model elements connecting the model with the external world.



## Goal 2: Metamodels

We have applied our meta-modeling process to a set of traditional user interface metamodels. Our most mature results comprise task, layout, service and context metamodels. We have extended them with runtime concepts specified in the meta-metamodel. E.g. the task metamodel has been extended with elements that describe the state of a task tree at runtime and its execution, as well as adaptation logic.

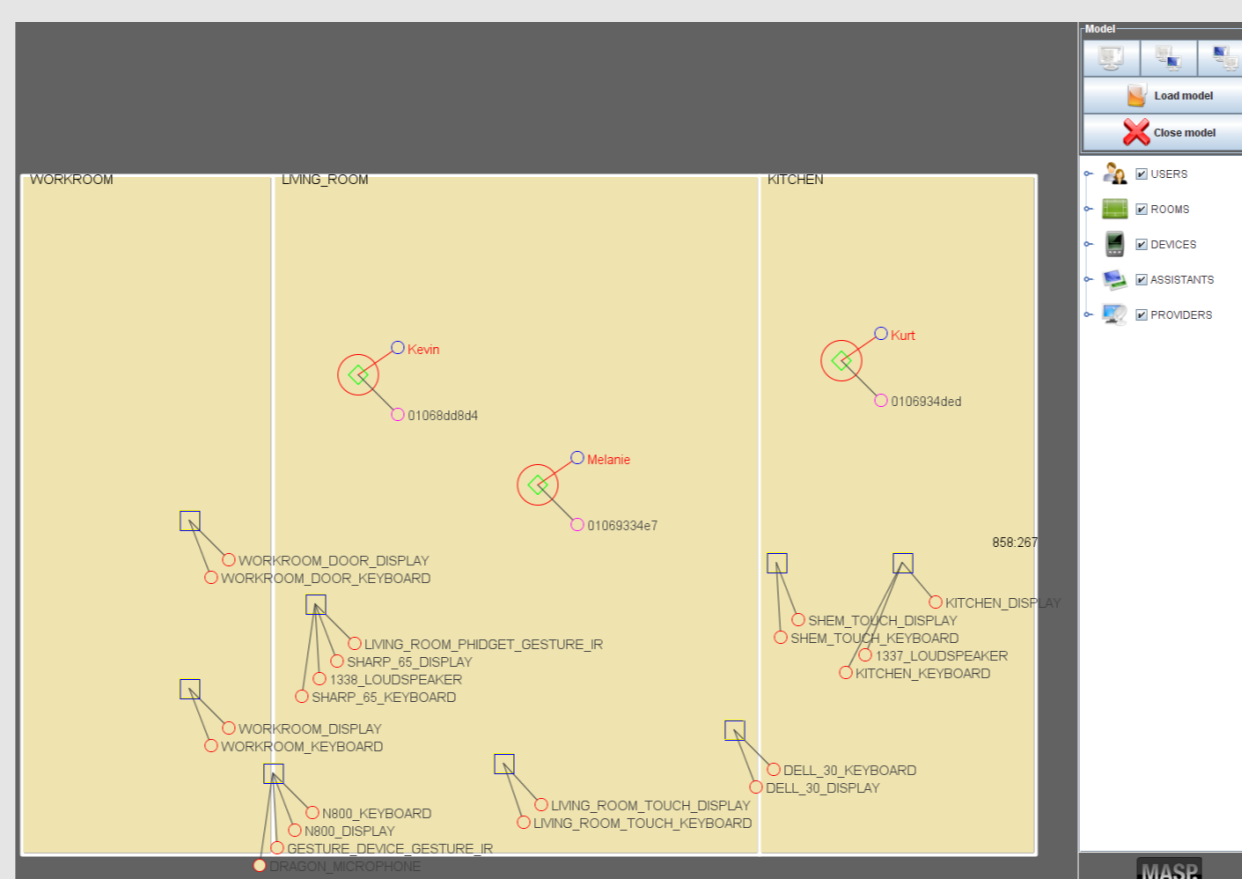


## Outlook

One of the most interesting implications of the runtime models utilization is the closing of the design time – runtime gap. Models that influence the behavior of the modeled system at runtime start acting like code. In the future we plan to further investigate this issue and its impact on the development process of adaptive applications by performing more metamodel and application case studies.

## Goal 3: Tools

To aid the developer during the development process, we have created a special model editor. The editor supports both the meta-modeling and modeling process. Using the editor the developer can create new runtime models, modify and execute them. During the execution the editor enables the inspection of the model's state through a separate runtime view on the situation part of the model.



## Links

MASP  
<http://masp.dai-labor.de>

DAI-Labor  
Distributed Artificial Intelligence Laboratory

