

Andreas Pleuss, Arnd Vitzthum, Heinrich Hussmann
University of Munich

Integrating Heterogeneous Tools into Model-Centric Development of Interactive Applications





Introduction

- MDD: Systematic, well-structured application development, teamwork, platform-independence, maintainability, etc. ...
- User interface design: Usability of the application, user interface elements and interaction paradigms for new devices, aesthetics of the user interface, etc. ...
- Common practice in UI design: User-Centered Design
- Find out user's tasks, needs and properties to provide an optimized UI
- Achieved by permanent user feedback and user tests
- Based on various prototypes of different granularity
- Examples:
 - Mock-ups for first ideas and basic requirements
 - Lo-fi prototypes like paper prototypes for first tests of task flow and interaction
 - Hi-fi prototypes like click-dummies for more detailed interaction



Tools Used by User Interface Designers

- Support creativity
- Support visual, direct manipulation
- Various heterogeneous tools for different development steps
- Examples:
 - Drawing and image editing tools, like *Photoshop*, *Illustrator*, etc., e.g. used for creation of Mock-Ups
 - Authoring tools like *Flash*, e.g. used for creation of click-dummies
- Visual tools also used for final implementation; in particular for complex (non-standard) user interfaces
 - Multimedia user interfaces often implemented by authoring tools like *Flash* or *Director*
 - 3D user interfaces implemented by 3D graphics tools like *3D studio max*

Problem

- MDD for efficient high-quality application development
 - Systematic and abstract
 - Problem: Low acceptance and understandability for people without technical background
 - User-Centered Design for high-quality user interface development
 - Creative and concrete
Involves graphical designers, users and customers
- 

Clash of cultures between abstract models of software engineers and concrete and creative tools/artifacts in user interface design

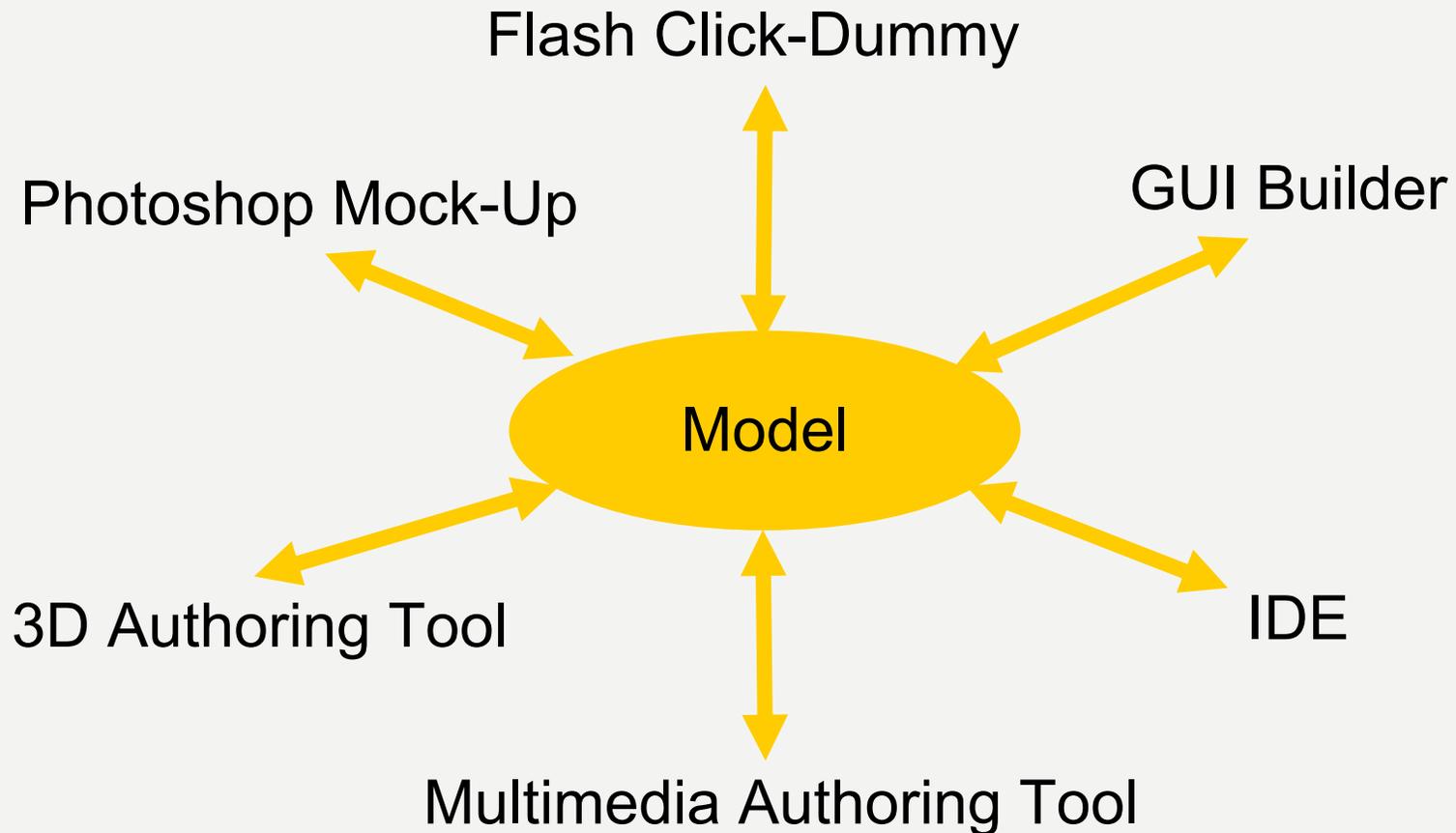


Proposed Solution

- Abstract model adopts role to integrate the heterogeneous artifacts and tools
- Extract (automatically) the relevant information from artifacts from user interface design into the model
- Create skeletons for creative tools from the models
- Advantages:
 - Integration and coordination of interdisciplinary work
 - Increased consistency
 - Efficiency

→ **Models as “central hub” in the development process**

Models as “Central Hub” in the Development Process





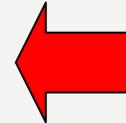
Automatic Extraction of Information: Conventions

- Sometimes necessary: Certain basic conventions for designers to enable automatic analysis of their results
- Naming conventions to specify the semantics of elements
- Simple structure conventions
- But: User Interface Designers already used to conventions and guidelines (e.g. usability guidelines)

Outline for the Remaining talk: Concrete Examples

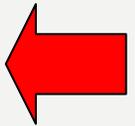
1. From Prototypes to Models:

1. Flash Click-Dummies to Models
2. Photoshop Mock-Ups to Models

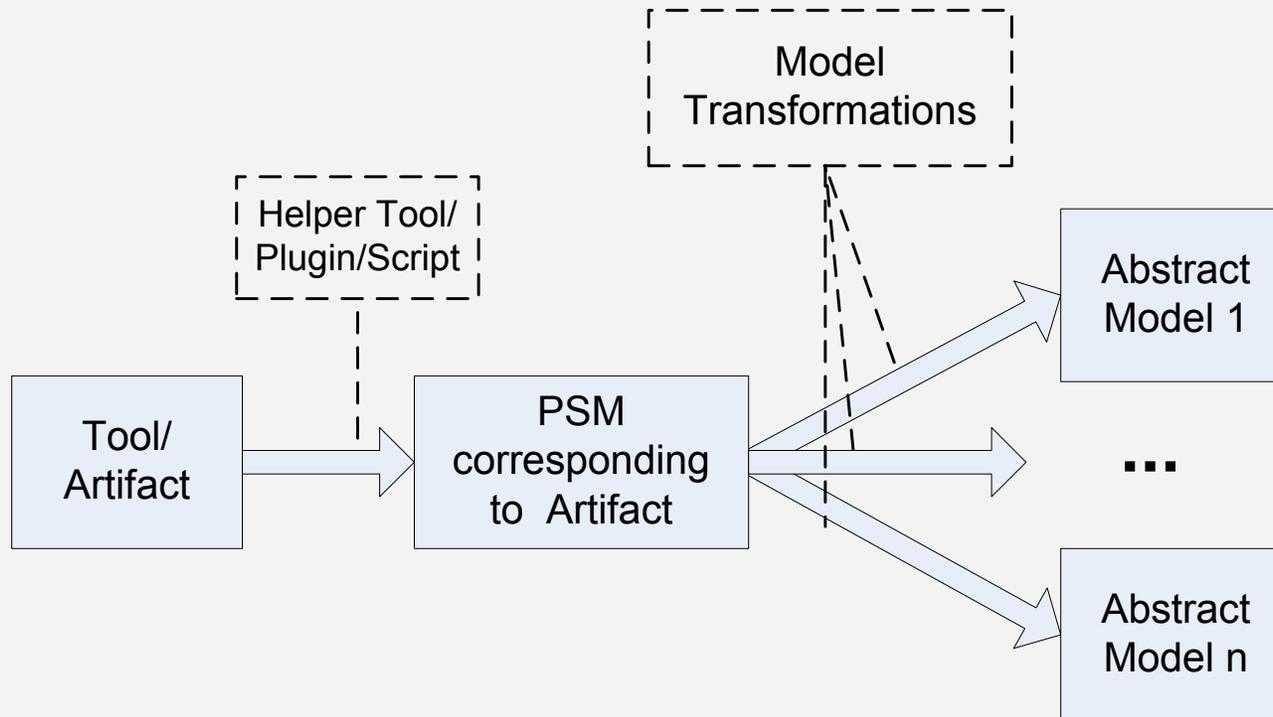


2. From Models to Code Skeletons

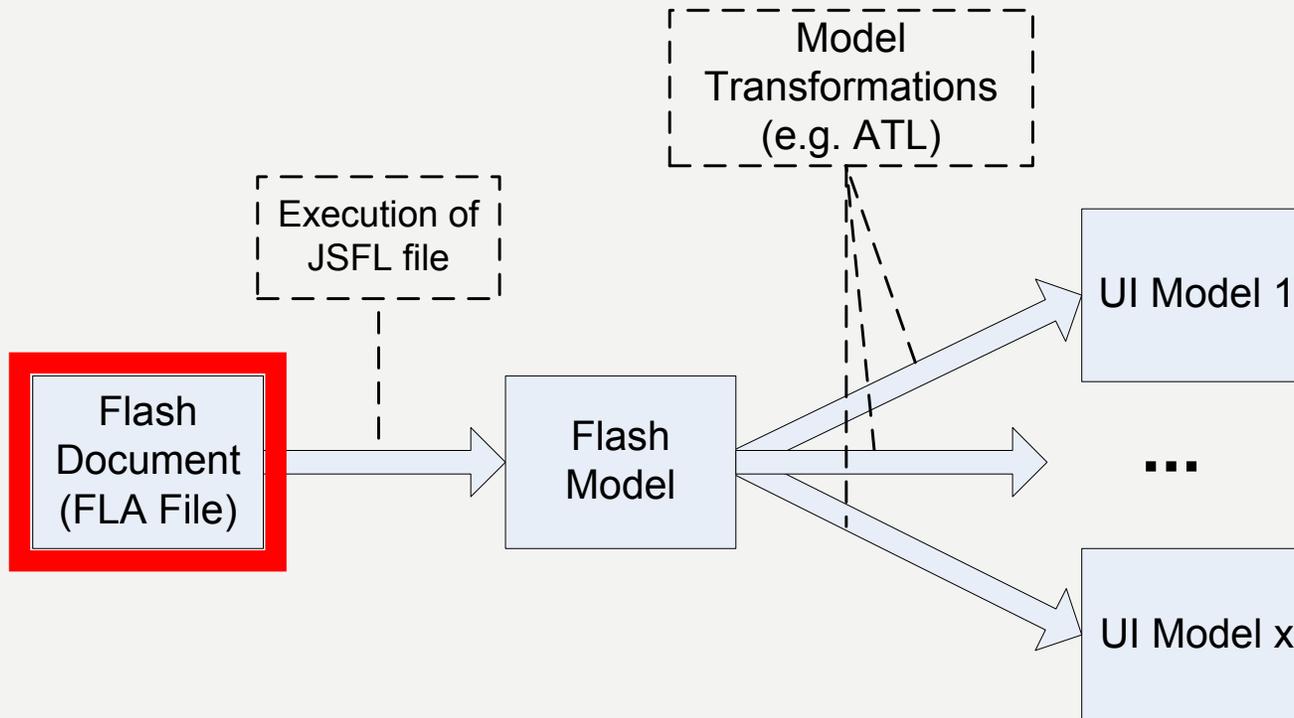
1. From MML Models to Flash
2. From SSIML Models to 3D Authoring Tools



Transition: From Tools to Models



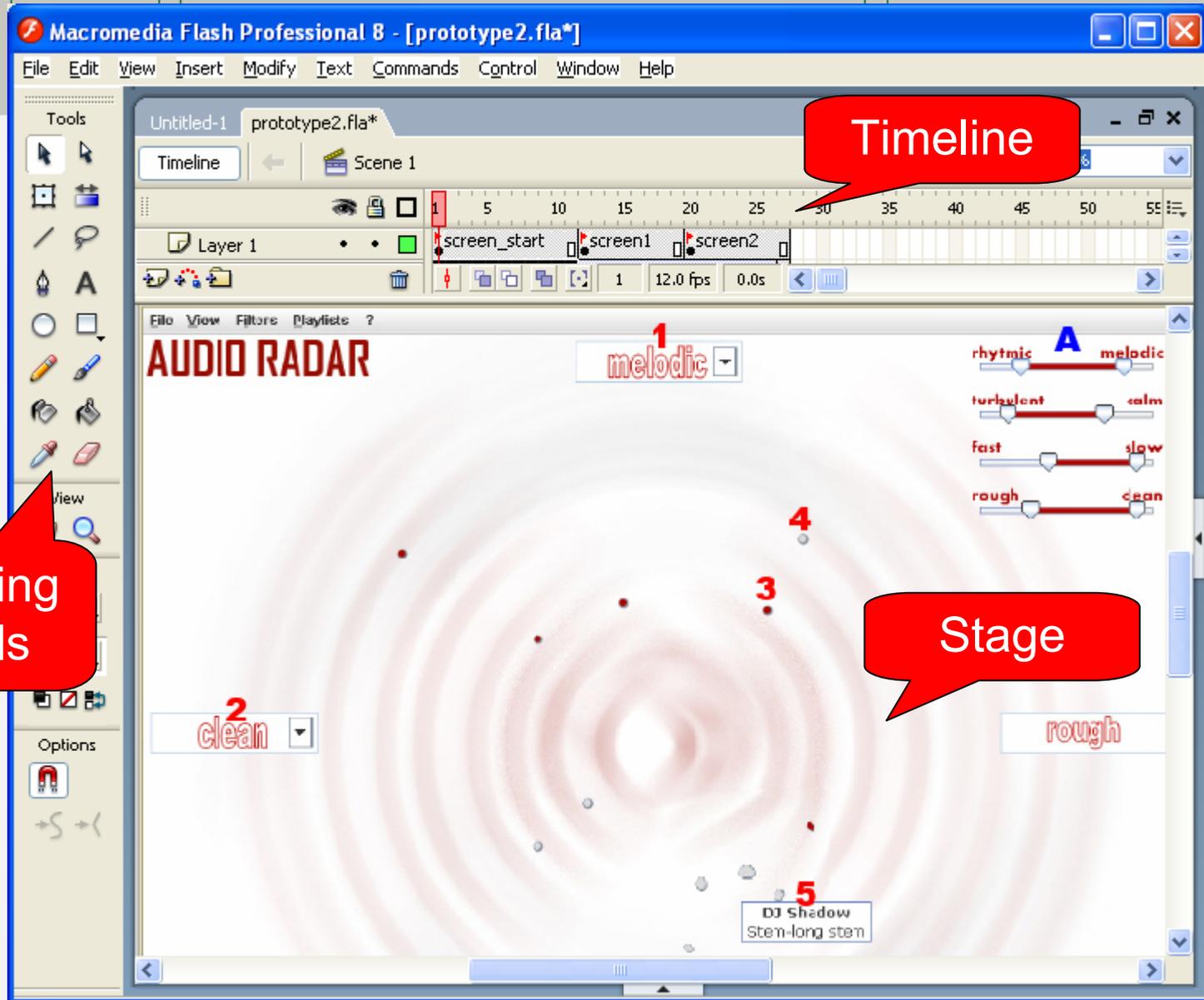
From Flash Click-Dummies to Models





Adobe Flash

- Popular professional multimedia authoring tool
- Supports integration of different media objects and creation of 2D graphics and animation
 - File format: proprietary, binary *FLA* files
- Integrated scripting language *ActionScript*
 - Either embedded within FLA document or as separate class files (*.as)



Macromedia Flash Professional 8 - [prototype2 fla*]

File Edit View Insert Modify Text Commands Control Window Help

Tools

Timeline

Scene 1

100%

Layer 1

screen_start screen1 screen2

20 12.0 fps 1.6s

File View Filters Playlists ?

AUDIO RADAR

calm melodic

clean slow

fast rough

The Servant
Cells

50cent
In da club

DJ Shadow

Stem-long stem

4 5

1 2 3

A B

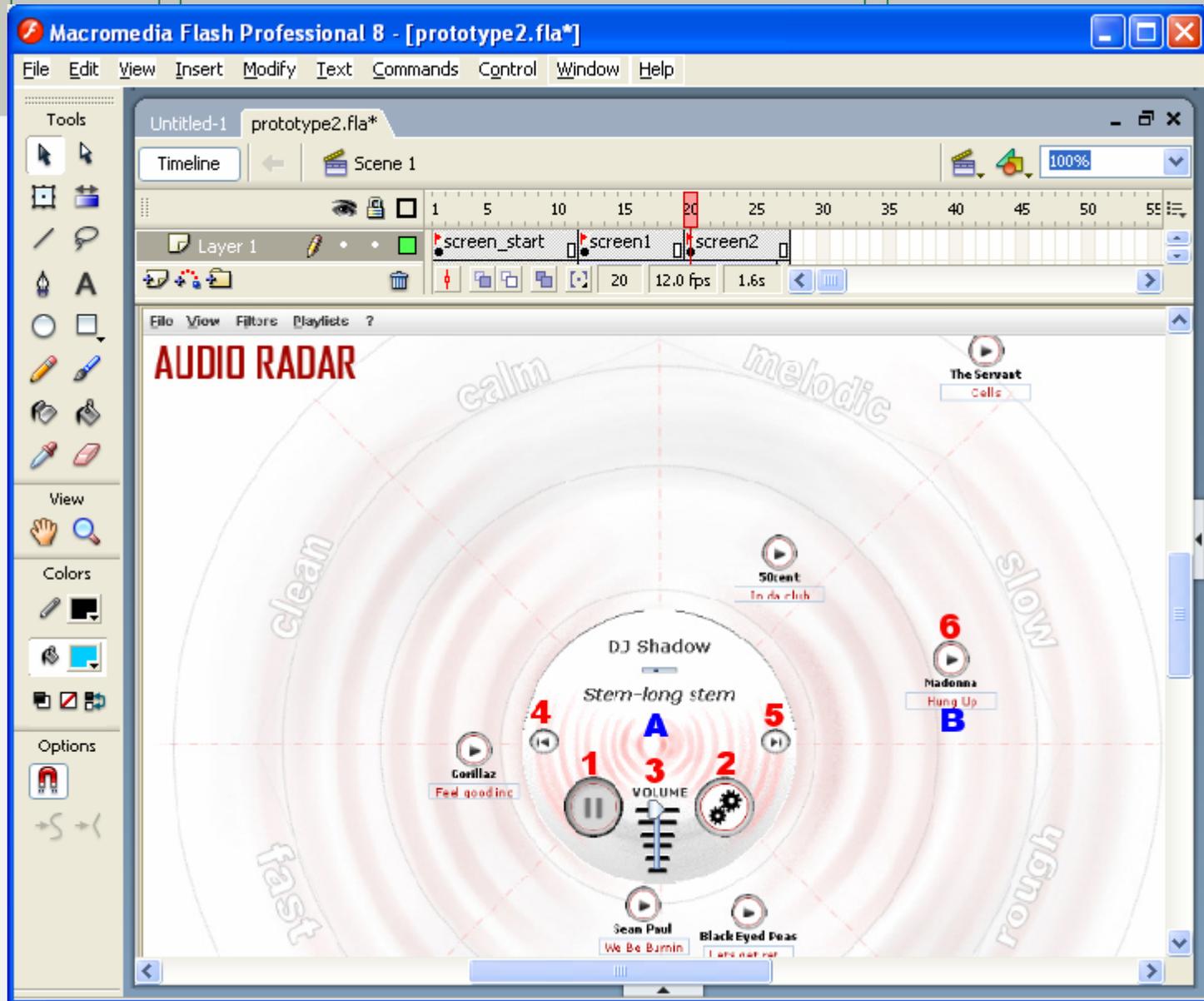
VOLUME

Corllaz
Feel goodinc

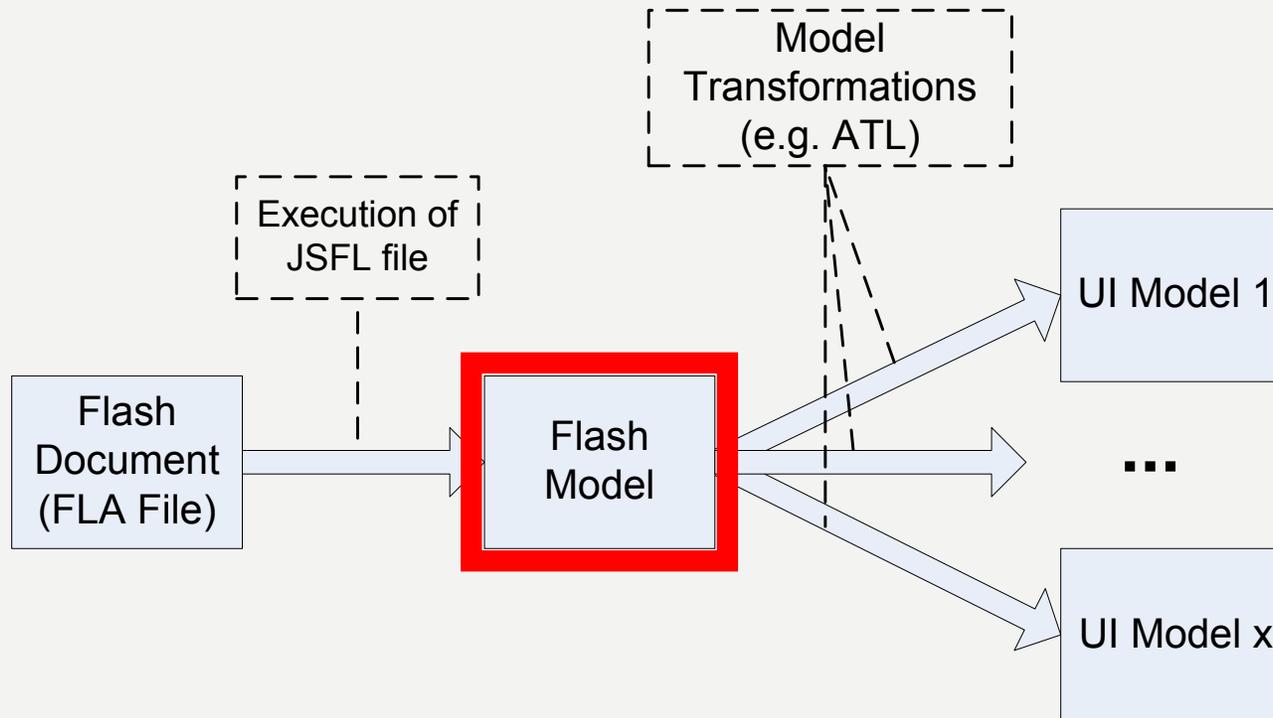
Sean Paul
We Be Burnin'

Black Eyed Peas
I am not a no

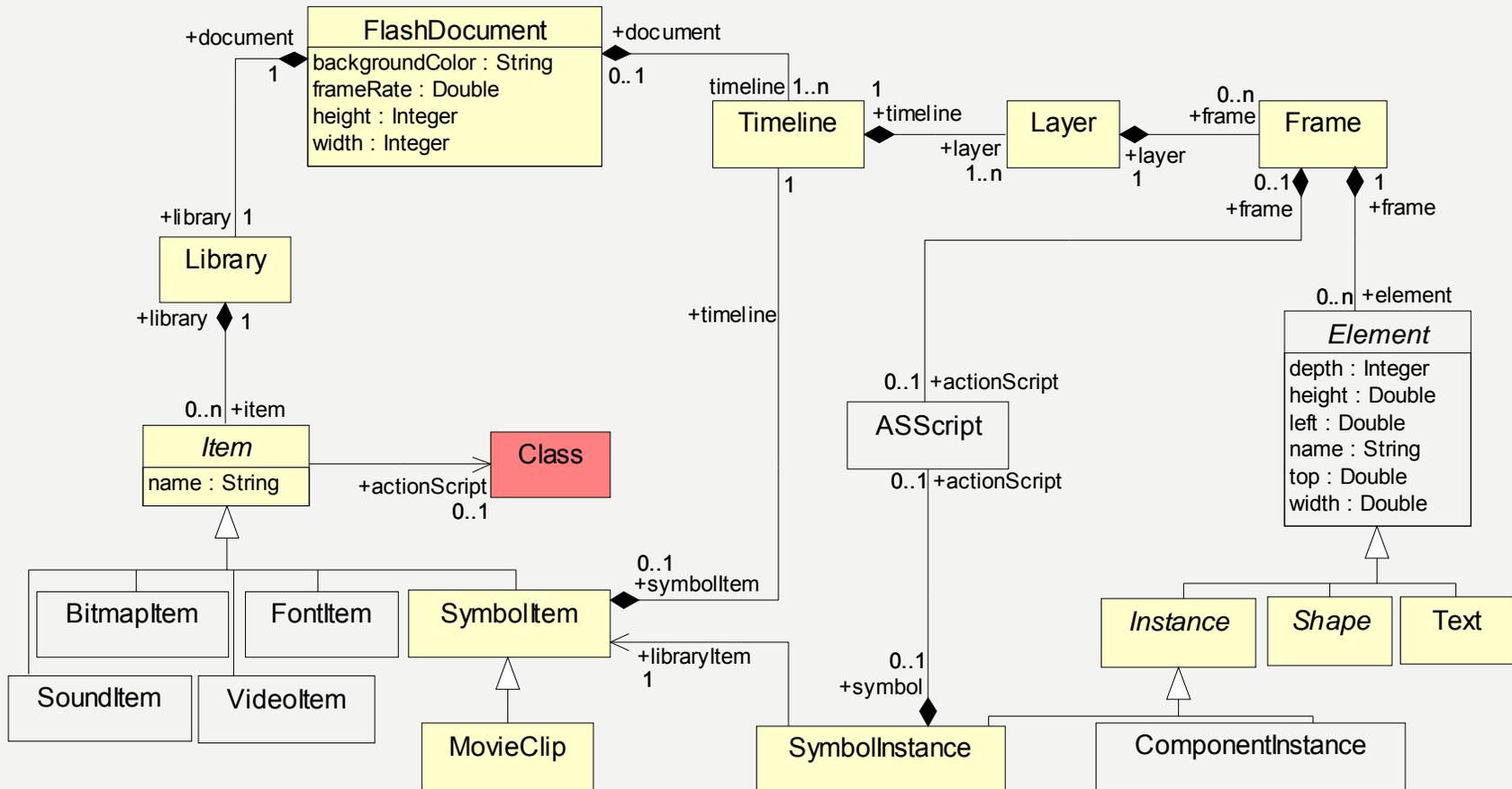
Madonna
Hung Up



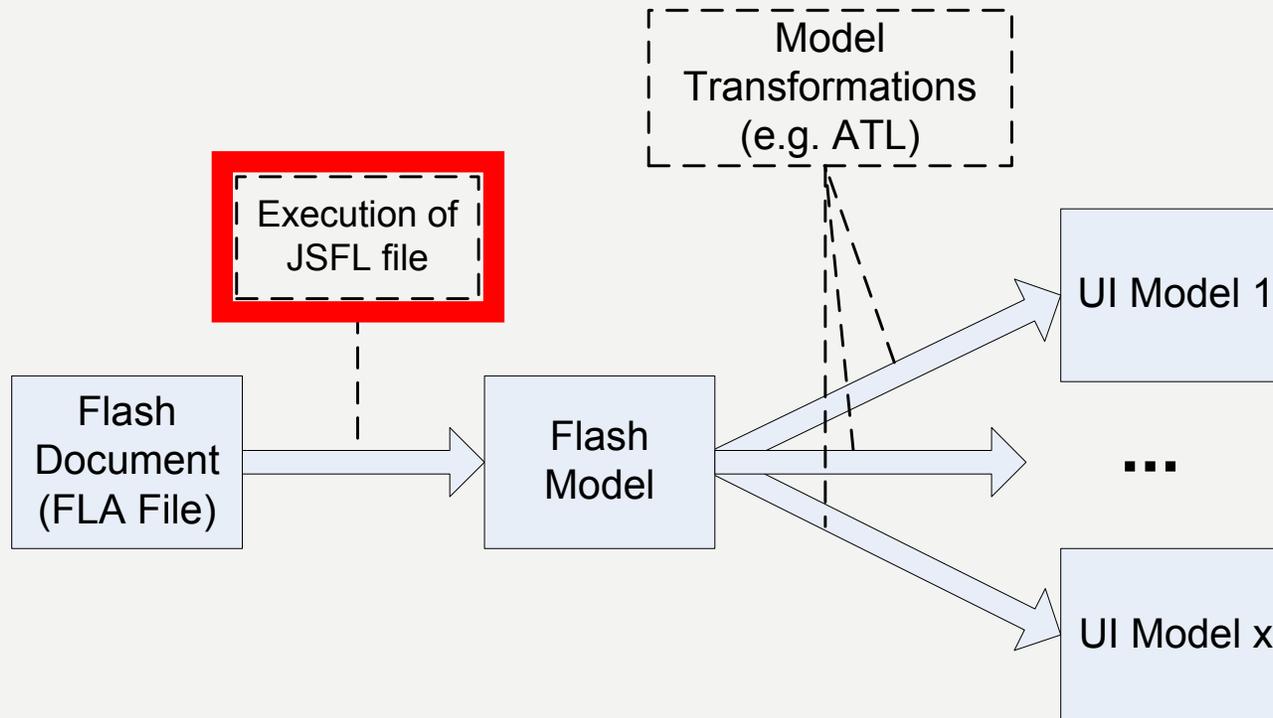
From Flash Click-Dummies to Models



Flash Metamodel (Simplified Extract)



From Flash Click-Dummies to Models

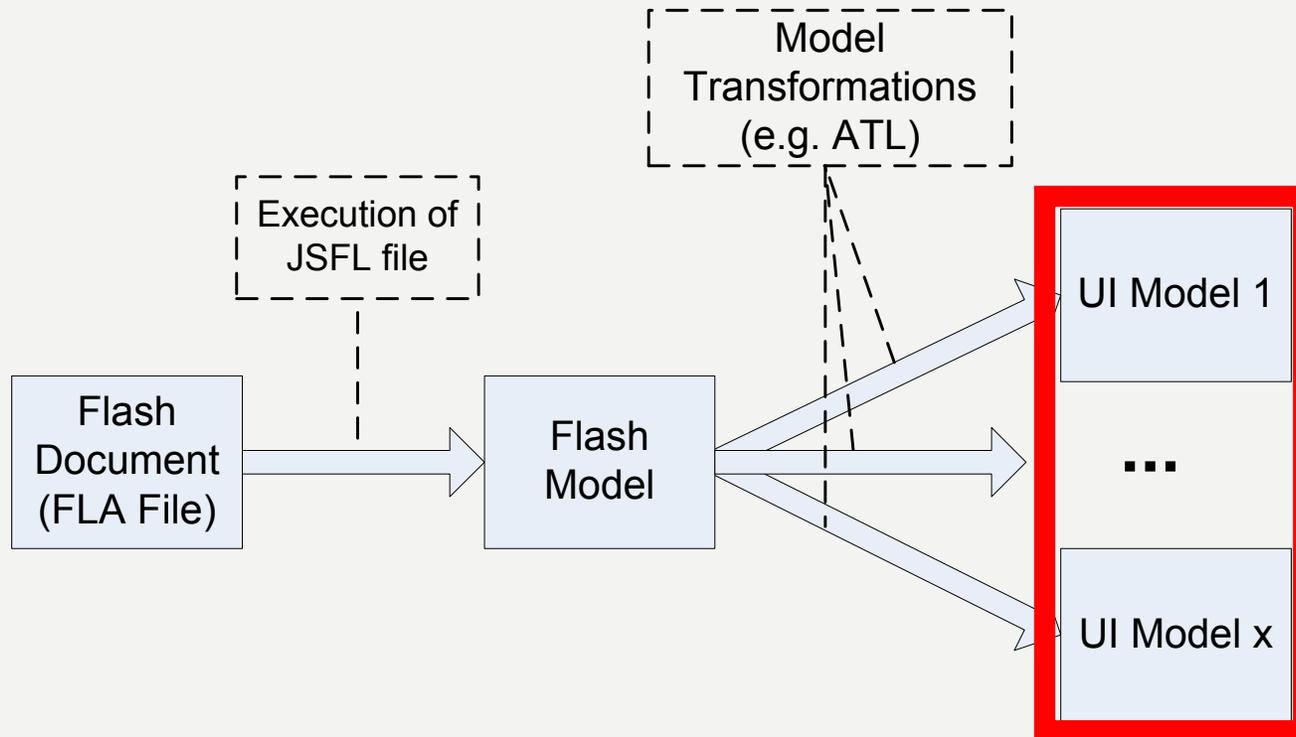




Extracting information

- How to extract information from proprietary, binary Flash files?
- Available: Flash authoring tool provides a plug-in mechanism:
 - Plug-ins are scripts written in JSFL (Java Script for Flash)
 - JSFL provides kind of document object model for Flash documents
- Solution for our purpose: JSFL script browses through document and outputs XML for each found element
- Result: “Flash model” conform to the Flash metamodel which we have specified (implemented with EMF)
- In addition: Parser for ActionScript code (implemented with JavaCC)

From Flash Click-Dummies to Models

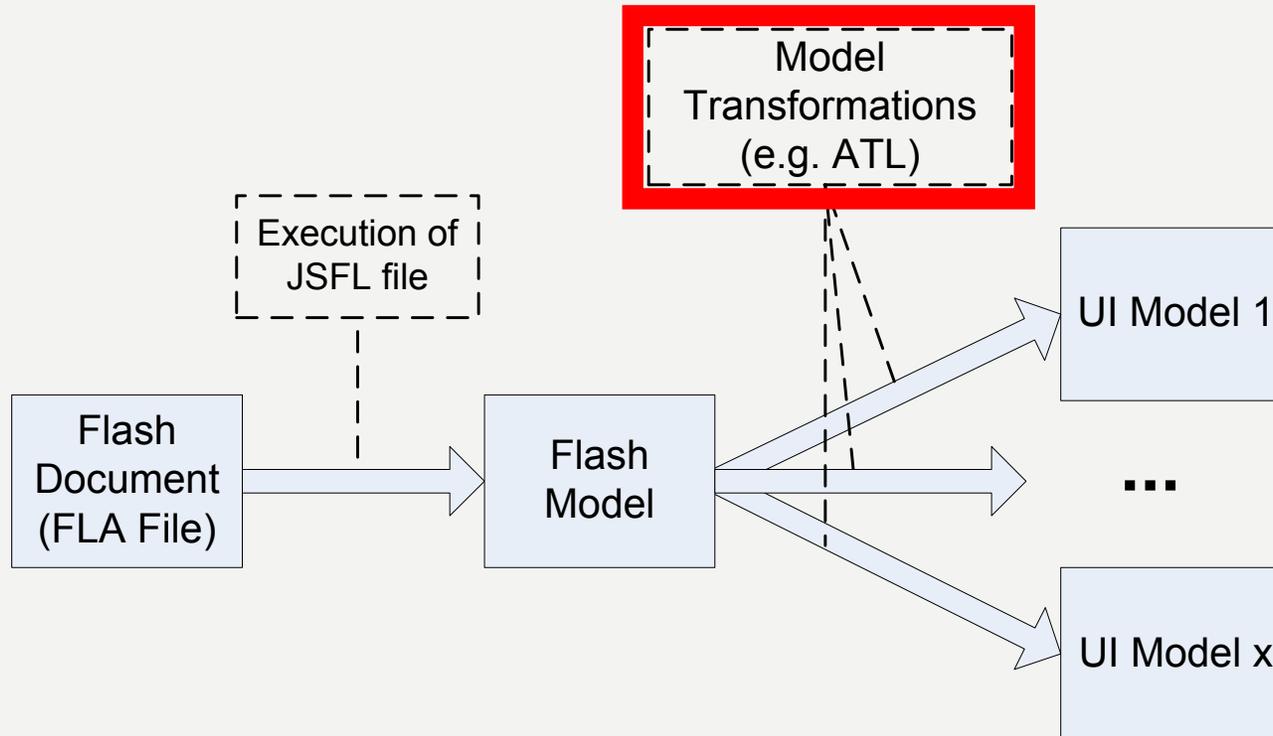




Abstract Models

- Must include information on the user interface
- E.g. modeling languages for model-driven user interface development, like UsiXML, DynamoAid, TERESA, and many others
 - See e.g. MODELS workshops on “Model Driven Development of Advanced User Interfaces“
- Main concepts of such languages:
 - *Presentation Units* (i.e. „screens“ in case of a visual UI)
 - Abstract user interface elements, like *input component*, *output component*, *selection component*, etc.
 - Interaction, Navigation
 - Relationships to domain model (i.e. application logic)

From Flash Click-Dummies to Models





Transformation (Sketch)

- Frames from timeline mapped to presentation units
- Widgets mapped to corresponding abstract UI elements
- Jumps between screens (action script command *gotoAndStop()*) mapped to navigation



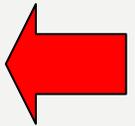
Outline for the Remaining talk: Concrete Examples

1. From Tools to Models:

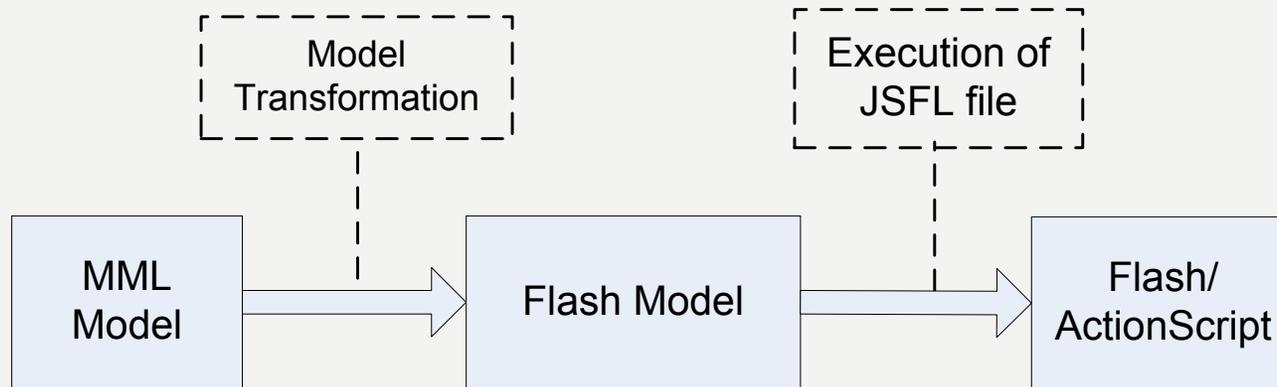
1. Flash Click-Dummies to Models
2. Photoshop Mock-Ups to Models

2. From Models to Code Skeletons:

1. From MML Models to Flash
2. From SSIML Models to 3D Authoring Tools



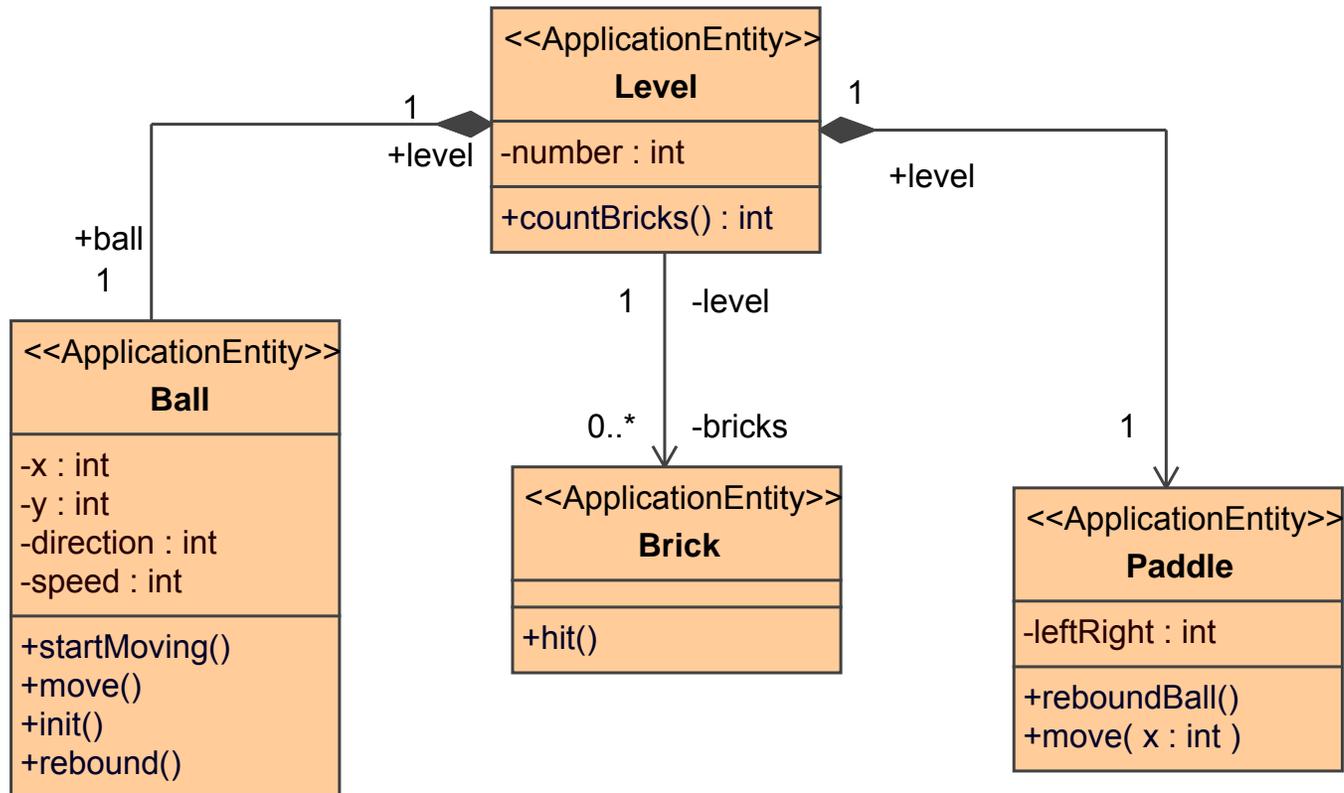
Transition: From MML Models to Flash Code Skeletons



Example: Blockout Gaming Application

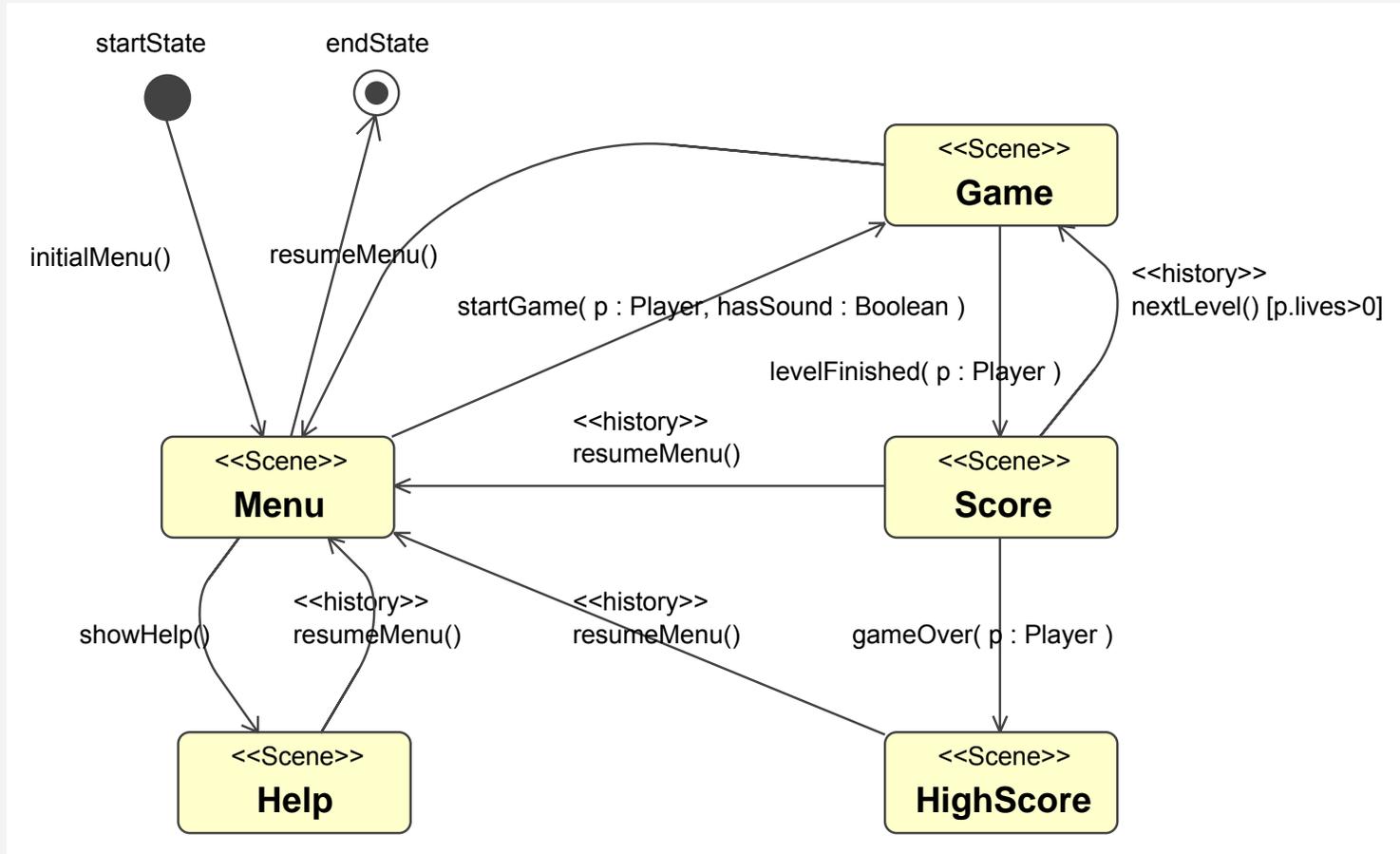


Application Logic: MML Structure Diagram (extract)



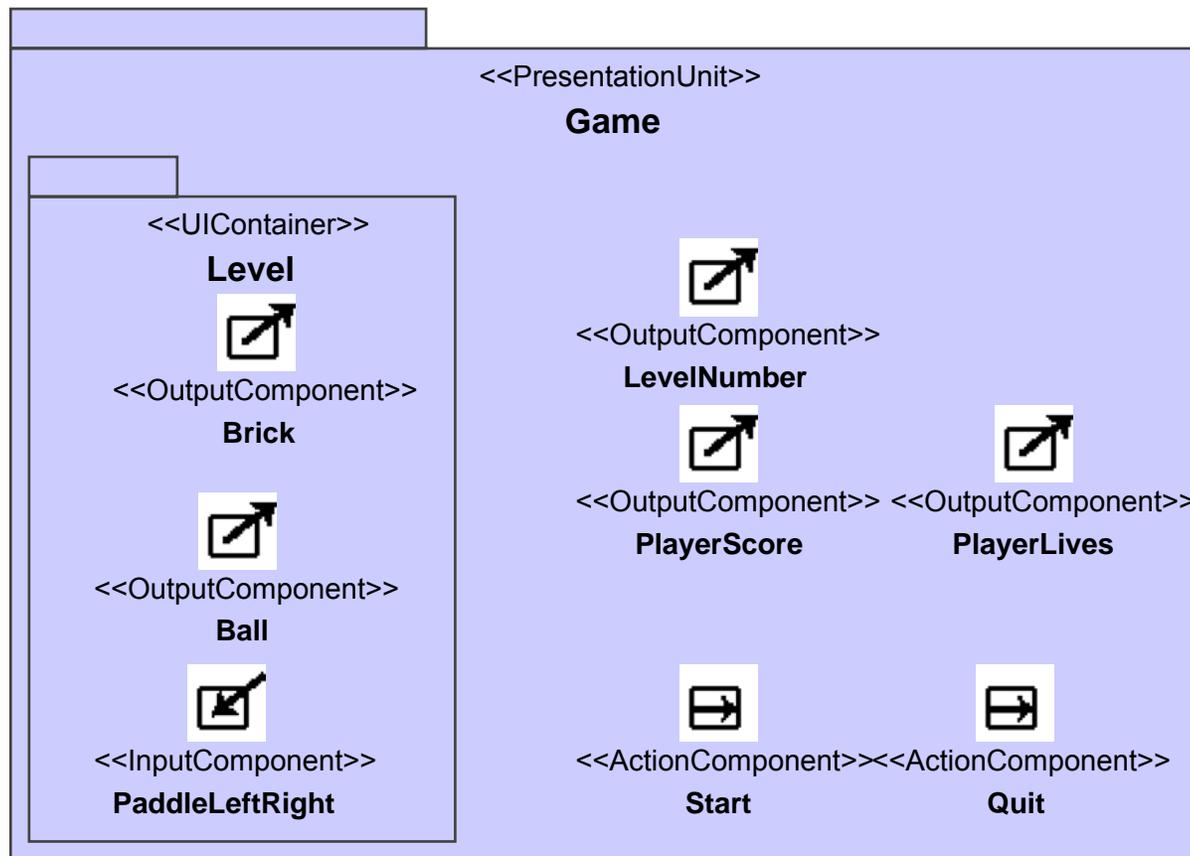


MML Scenes



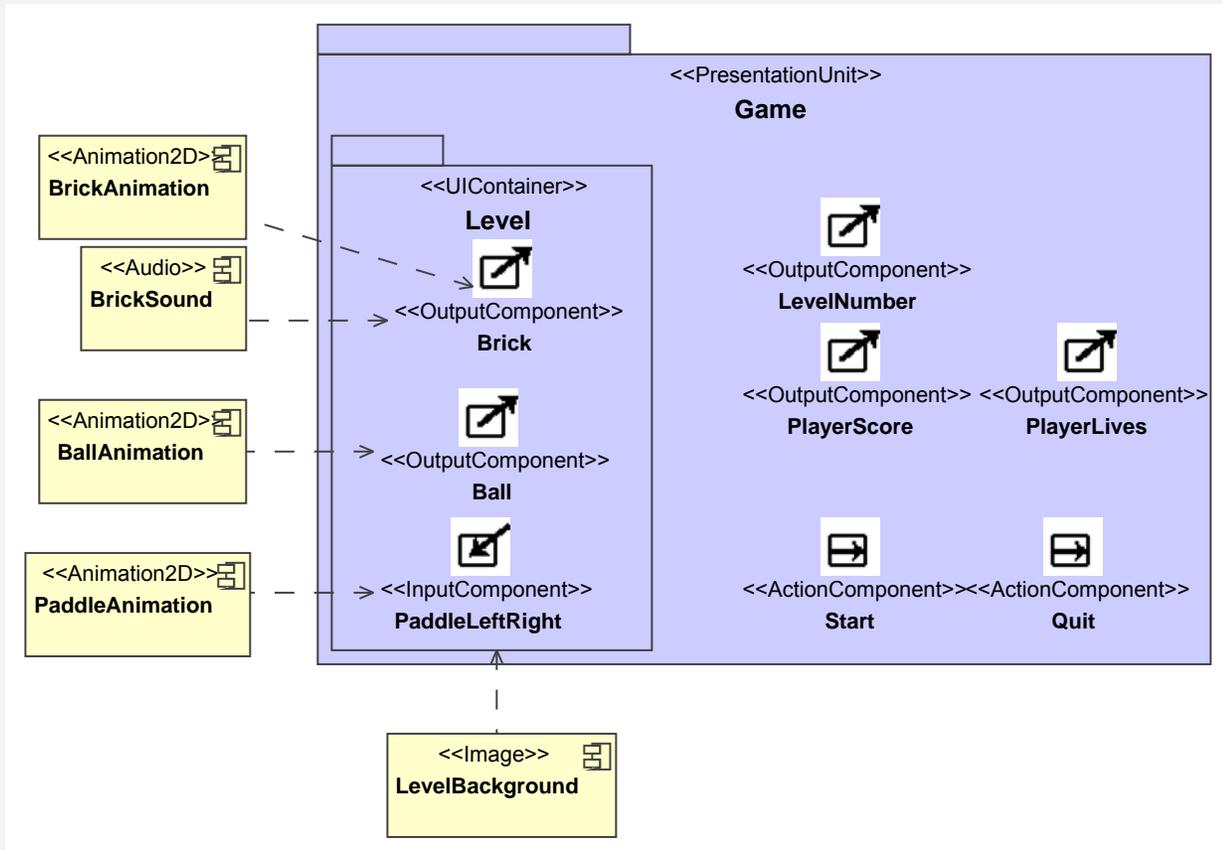


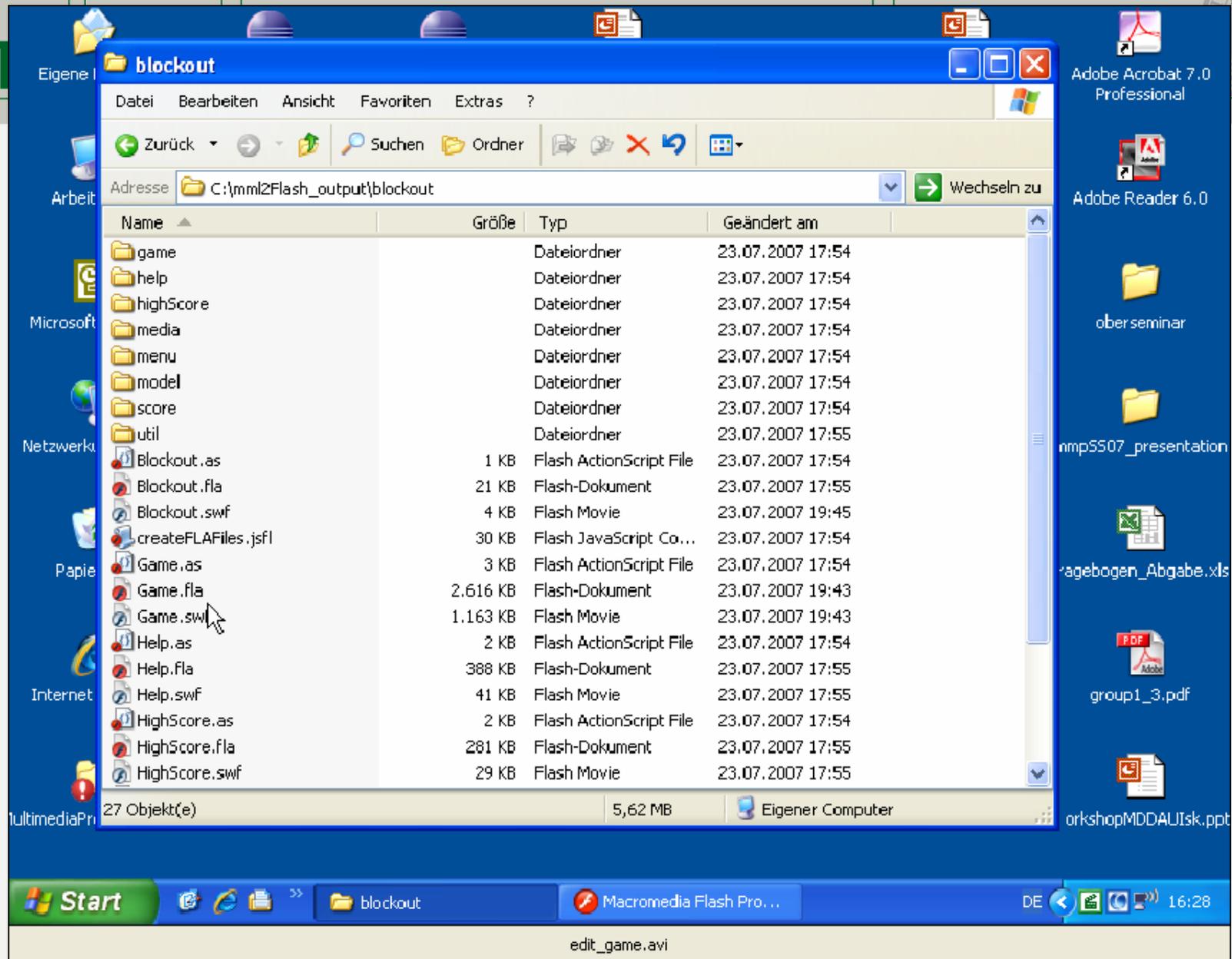
MML Abstract UI of Scene "Game"

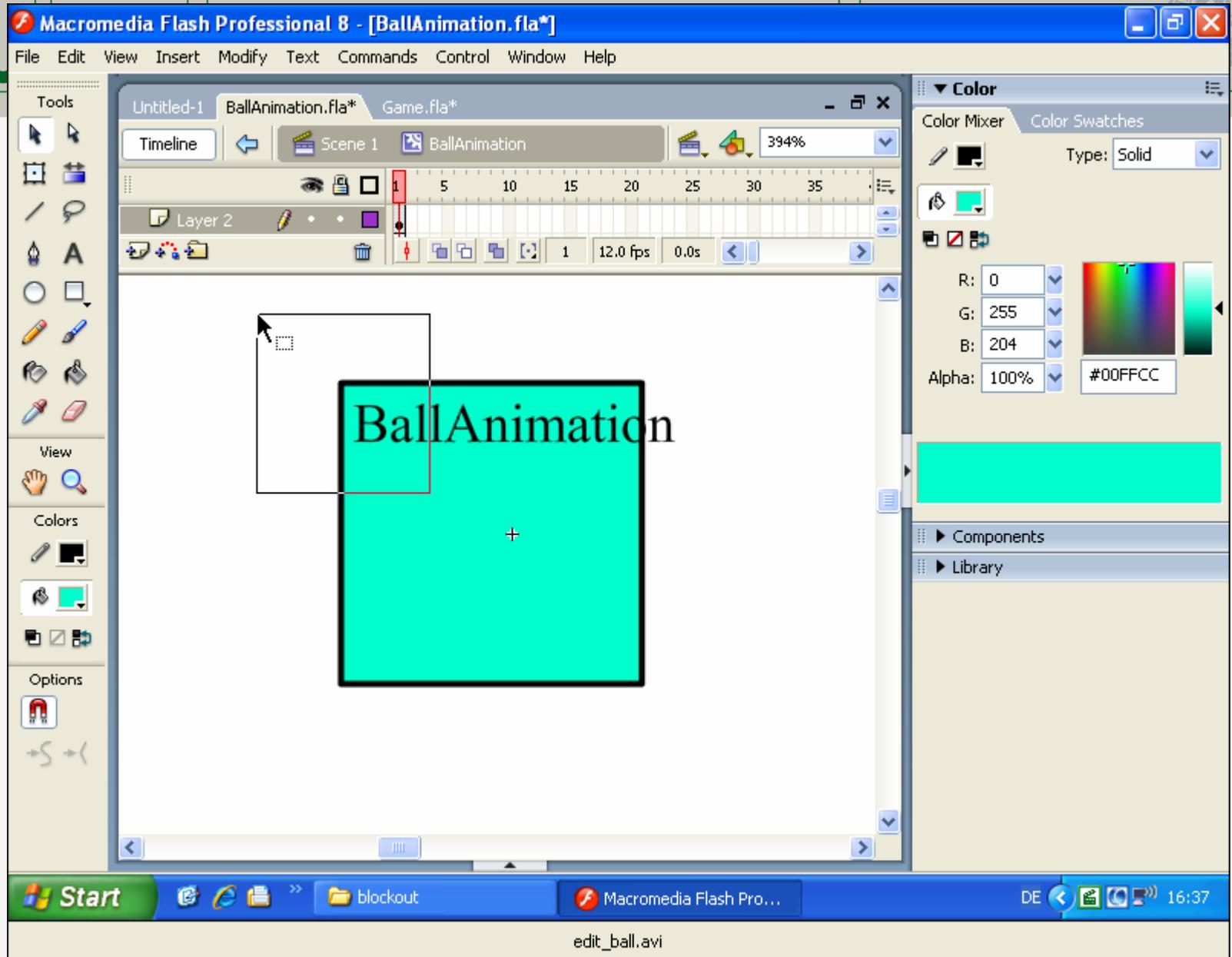


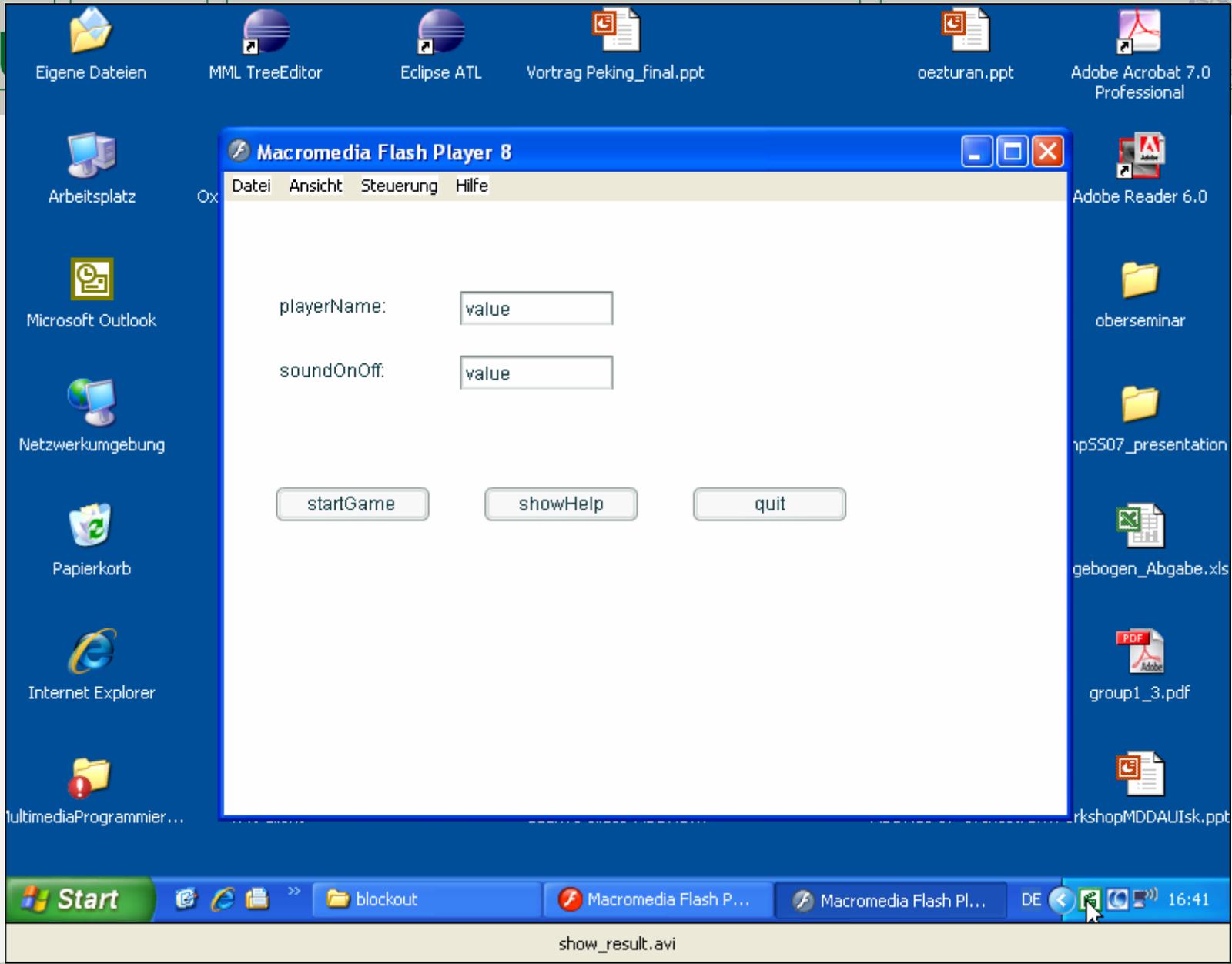


Abstract UI: Partially Realized by Media Components











Summary and Conclusion

- Model as “Central Hub” to integrate creative user interface design and MDD
- Long-term goal: Seamless transitions across different abstraction levels and different tools
- Two-step approach for transitions
- Similar tendencies in commercial tools (e.g. Microsoft Expression)
- Still much work to be done...