Object-oriented Modeling of Multimedia Applications

areas which are relevant for multimedia applications (cf. Steinmetz)

<table>
<thead>
<tr>
<th>Modeling Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document structures</td>
</tr>
<tr>
<td>Database</td>
</tr>
<tr>
<td>Operating System</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>technical foundations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Architecture</td>
</tr>
<tr>
<td>Audio</td>
</tr>
</tbody>
</table>
6. UML & OMMMA

Intro: Objectives and Context of OMMMA

- definition of an object-oriented modeling language for multimedia
- based on the UML standard
- development of a multimedia CASE tool
- extension by an execution and/or generator component (programming languages, multimedia standard formats)
Motivation OMMMA

Object-oriented Modeling of MultiMedia Applications

**OMMMA:**
- characteristics of MM applications
- interactive software
- composition of media objects
  1. application structure, media objects
  2. presentation (GUI layout, audio)
  3. user interaction, event handling
  4. predefined (real-)time behavior

**OMMMA-L**
- modeling of MM applications
- incomplete MM development process
- integrated modeling of MM aspects
  1. assignment of aspects to diagram types
  2. pragmatics: usage & combination

**UML:**
- OMG standard general purpose
- pure UML?
  - no:
    - presentation
    - temporal behavior
    - pragmatics

Integrierte, konsistente MM-Modellierung
Development Strategy for OMMMA

OMMMA-L: UML-based diagram language for OMMMA

- development steps according to the 4 tiers of language design
6. UML & OMMMA

Development Strategy for OMMMA

OMMMA-L: UML-based diagram language for OMMMA

- development steps
- concepts
  - identification of characteristics to be modeled and their relationships
  - assignment to multimedia aspects
6. UML & OMMMA

Development Strategy for OMMMA

- OMMMA-L: UML-based diagram language for OMMMA
- development steps
- concepts
- languages (syntax and semantics)
  - assignment of multimedia aspects to diagram types
  - analysis of existing UML language elements; adaption & extension using UML‘s built-in extension mechanisms
Development Strategy for OMMMA

- OMMMA-L: UML-based diagram language for OMMMA
- development steps
- concepts
- languages (syntax and semantics)
- methods (pragmatics)
  - guidelines for using and combination of diagram types
  - integrated & consistent application model
Development Strategy for OMMMA

- OMMMA-L: UML-based diagram language for OMMMA
- development steps
  - concepts
  - languages (syntax and semantics)
  - methods (pragmatics)
  - formalization: refined meta model vs. profile (& definition of semantics)
Development Strategy for OMMMA

- OMMMA-L: UML-based diagram language for OMMMA
- development steps
  - concepts
  - languages (syntax and semantics)
  - methods (pragmatics)
  - formalization: refined meta model vs. profile (& definition of semantics)
- tools
  - syntax-directed editors, consistency checking;
  - code generation; process support
Example Application: Composers’ Encyclopedia

- **Hauptthema**

- **Biographie**
  
  **Mozarts**
  
  - **1718**: Großvater Johann Georg Mozart, Buchbinder in Augsburg, heiratet Anna Maria Suizor.
  
  - **1719**: In Augsburg wird als Sohn dieses Paares **Mozarts Vater Leopold** geboren.
  
  - **1720**: In St. Gilgen, ein Ort am Wolfgangsee bei Salzburg, wird **Anna Maria Pertl**, Mozarts Mutter geboren. Leopold Mozart läßt
Example Application: Composers' Encyclopedia

- **media objects**
  - hypertext, portraits, music samples & scores, video sequences
- **logical structure**
  - content relationships, composition of a composer entry
- **unified layout**
  - reusable
- **interactivity**
  - common dialogue structure, navigation: alphabetic panel, scroll list, buttons, control
- **predefined scenes**
  - temporal behavior, synchronization
6. UML & OMMMA

Software Development: Traditional (?) Approach

problem domain

implementation

program
Software Development: Reality

problem domain

program

program
Software Development: Model-centered Approach

problem domain

analyse and design

abstracts from irrelevant details

model

abstracts from implementation details

code

program
6. UML & OMMMA

Relevance of a Model

- Contract between client and software developer
- Contract between
  - developers within a team
    - horizontally: subsystems
    - vertically: analysis - design - implementation
  - developer and supplier of out-sourced subsystems
  - developer and components („trusted components“)
- Documentation
  - maintenance, extension, adaptation, re-engineering
Requirements for a Model and the Modeling Language

Requirements for a Model

– user-friendly, understandable
– complete
– expressive, adequate
– correct, precise

Requirements for a Modeling Language

• visual, diagrammatic
• structure & behavior
• problem domain-specific
• syntax & semantics
6. UML & OMMMA

Contents

5. Multimedia Framework

- OMMMA: Object-oriented Modeling of Multimedia Applications Based on UML

6.1 Introduction OMMMA

6.2 Unified Modeling Language (UML)

6.3 Language Definition and Extensibility of UML

6.4 OMMMA-L
Goals of UML

- expressive, standardized modeling language
- integration of best practices
- general purpose, not problem domain-oriented language
- extension and specialization mechanisms
- formal syntax and informal semantics
- fixed concrete (graphical) syntax
- CASE tool support
- use within software development processes

- but
  - no fixed process or method
  - no fixed tool
  - no fixed implementation (programming) language
Four Tiers of Language Design

- **Concepts**
  - Languages
  - Methods (Process Model)
  - Tools

UML: union of (at least) 9 different sub-languages
## Classification of UML Diagram Languages

<table>
<thead>
<tr>
<th>Diagrams</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Class / Object Diagrams</td>
<td>Structure Diagrams</td>
</tr>
<tr>
<td>(2) Use Case Diagrams</td>
<td>Behavior Diagrams</td>
</tr>
<tr>
<td>(3) Statechart Diagrams</td>
<td>Interaction Diagrams</td>
</tr>
<tr>
<td>(4) Activity Diagrams</td>
<td></td>
</tr>
<tr>
<td>(5) Sequence Diagrams</td>
<td></td>
</tr>
<tr>
<td>(6) Collaboration Diagrams</td>
<td></td>
</tr>
<tr>
<td>(7) Component Diagrams</td>
<td>Implementation Diagrams</td>
</tr>
<tr>
<td>(8) Deployment Diagrams</td>
<td></td>
</tr>
<tr>
<td>(9) Object Constraint Language (OCL)</td>
<td>Structural Constraints</td>
</tr>
</tbody>
</table>
Usage of UML within Standard Software Development Process

<table>
<thead>
<tr>
<th>Standard Process</th>
<th>Requirements</th>
<th>Specification</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure Diagrams</td>
<td>Class / Object Diagrams</td>
<td>Use Case Diagrams</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Behaviour Diagrams</td>
<td>Statechart Diagrams</td>
<td>Activity Diagrams</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Diagrams</td>
<td>Sequence Diagrams</td>
<td>Collaboration Diagrams</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation Diagrams</td>
<td>Component Diagrams</td>
<td>Deployment Diagrams</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Object Model: Class Diagram

- **classes** define structure (attributes) and behavior
- (operations) of objects of a kind ("who is participating?"")
- **associations** define (possible) relationships between objects of these classes
  - **associations** with role names and multiplicities
  - aggregation and composition
- **inheritance relationships**
- **visibilities** (private, protected, public)
- **explicit integrity constraints**
  - (given in OCL (Object Constraint Language))
Dynamic Model: Statecharts

- origin: state transition diagram
- **UML Statecharts** (proposed by D. Harel)
  - state transition diagram extended by
    - hierarchy concept (complex states)
    - complex state transitions
  - are assigned to classes
- model possible sequences of method calls on objects of this class
  („when is something allowed to happen?“) („object life-cycle“)
Functional Model (1)

- model of functionality („what?“)
- assigned to use cases or operations
- orientated towards different foci:
  - data-flow or control-flow oriented: activity diagrams
  - scenario oriented: sequence diagrams
  - object-structure oriented: collaboration diagrams
Functional Model(2): Activity Diagrams

origin: statechart diagrams

UML activity diagrams

- alternative interpretation of statechart diagrams
  - states represent activities
  - state transitions are automatically triggered

- suited for modeling of procedural control flow
- not suited for flows that are triggered by external events (user input)
Functional Model(3): Sequence Diagrams

• origin: Message Sequence Charts (MSC)

• refinement of use case diagrams („what happens?“)

• description of interaction between objects by exchanging messages over time

• language elements:
  • passive / active objects
  • anonymous/named objects / multi-objects
  • synchronous / asynchronous / conditional message exchange
6. UML & OMMMA

**Functional Model(4): Collaboration Diagrams**

**collaboration diagrams** consist of

- **collaboration**: object (role) diagram
  
  (objects/roles and relationships)

- **interaction**: collaboration extended by the flow of message exchange

- in comparison to sequence diagrams, collaboration diagrams focus on objects and their relationships rather than message sequences in time
Consistency of UML Models

**consistency relationships between diagrams**

consistency between object model and functional resp. dynamic model:

usage of classes and operations as defined in the class diagram

consistency between dynamic and functional model:

conformity of sequences of method calls on each object with possible call sequences defined in statechart
## Contents

5. Multimedia Framework

- OMMMA: Object-oriented Modeling of Multimedia Applications Based on UML

  6.1 Introduction OMMMA
  6.2 Unified Modeling Language (UML)
  6.3 Language Definition and Extensibility of UML
  6.4 OMMMA-L
6. UML & OMMMA

Language Definition

tools
methods
languages
concepts

(process model)
syntax
semantics
UML Language Definition

UML:

- "general purpose" object-oriented modeling language
- syntax definition by meta model ( = meta class diagram)
- context-sensitive rules as "well-formedness rules" in OCL
- semantics definition in natural language

extensibility of UML:

- light-weight extensions of meta model ( "stereotypes"
- extensibility of context-sensitive rules ("constraints"
- additional properties ("tagged values"
- extension and specialization of semantics definition

UML's built-in extension mechanisms
**Language Definition**

4-layer meta model architecture for UML

- **M3: MOF Metametamodel**
- **M2: UML Metamodel**
- **M1: UML Model**
- **MO: UML Objects**

- **Model level**: MO = UML Objects
- **Instance level**: MO = UML Objects

- **Language definition**
  - **Meaning**: M3
  - **Grammar**: M2

**UML notation**:
- Package: MO
- Dependency: MO
Language Definition

Meta
MetaModel

<< metaclass >>
Attribute
name : Name

<< metaclass >>
Class
name : Name

<< metaclass >>
Operation
name : Name

<< instanceOf >>

UML
MetaModel

<< metaclass >>
Attribute
name : Name
initValue: Exp

<< metaclass >>
Class
name : Name
isActive:bool

<< metaclass >>
Operation
name : Name

<< instanceOf >>

UML Model

ReservationContract

Deposit : int = 100
fromPeriod : date
setDeposit ()

<< instanceOf >>

UML Objects

rc22 : ReservationContract

Deposit = 250
fromPeriod = 04092000
6. UML & OMMMA

Meta Model Architecture

Behavioral Elements

- Collaborations
- Use Cases
- State Machines
- Common Behavior
- Activity Graphs

Foundation

- Core
- Extension Mechanisms
- Data Types

UML metamodel

Model Management
6. UML & OMMMA

Language Definition

UML Language Definition

syntax
- context-free
- context-sensitive

semantics
- class diagram
- OCL constraints
- UML metamodel
- natural language

Problem:
no formal semantics
Formal Semantics

**Note:** semantics for individual diagram types exist

**Missing:** formalized integrated semantics

**Approaches:**
- individual approaches
  - (Petri Nets, graph grammars, transition systems, ...)
- precise UML (pUML) group
- OMG task force: Action Semantics
UML-Profiles

• domain- or process-specific extensions of the UML

Business Modeling
Software Processes
Multimedia

core UML
## Contents

5. Multimedia Framework

- OMMMA: Object-oriented Modeling of Multimedia Applications Based on UML
  
  6.1 Introduction OMMMA
  
  6.2 Unified Modeling Language (UML)
  
  6.3 Language Definition and Extensibility of UML
  
  6.4 OMMMA-L
Note:

UML is a general purpose, i.e., problem domain-independent modeling language

Question:

How can UML be adapted, extended or used for modeling of multimedia applications?
multimedia application =

interactive software system,

which coordinates different independent media objects

- discrete media objects (e.g., text, pictures)
- continuous media objects (e.g., video, audio, animations)
Interactive software system =
functionality + user interface

modeled by UML use case resp. class diagrams
refined in analysis and design by UML (behavior) diagrams

no UML user interface model
ad-hoc realization by GUI-builder

missing user interface model

consistency
Objective:

integrated, UML-based model for all aspects of a multimedia application
Aspects of Multimedia Applications

1. logical structure
   - application domain objects
   - associated media objects

2. spatial presentation (layout)

3. predefined temporal behavior
   - continuous media objects
   - real-time requirements

4. interactive control
   - event-driven
   - multimodality

integrated model
interactive software system =
  internal functionality + user interface

interactive system architecture patterns
  Model - View - Controller (MVC)
  Presentation-Abstraction-Control (PAC)
Interactive Systems: Model-View-Controller Pattern

- Model-View-Controller (MVC)
- Constituents
  - Model: core functionality and data
  - View: display information to user
  - Controller: handling user inputs

![Diagram of MVC pattern]
Interactive Systems: Presentation-Abstraction-Control Pattern

- Presentation-Abstraction-Control (PAC) [J. Coutaz, 87]
- Facets
- Presentation: perceivable behavior
- Abstraction: basic functionality and data
- Control: communication
  - between hierarchically organized „Agents“
  - with their respective Abstraction und Presentation facets
Interrelations between Aspects of a Multimedia Application

- MVC (Model - View - Controller) - Pattern

4 interactive Control
Controller

2 spatial presentation
View

1 logical structure
Model

3 predefined temporal behaviour
Model

Model_{static}

Model_{dynamic}
OMMMA: Object-oriented Modeling of MultiMedia Applications

Language Requirements:

- covers all aspects of a MM application
- graphical / visual
- already familiar to software developers

comprehensive

easy to understand

easy to learn

UML
UML Diagram Types vs. Multimedia Aspects

1. Class / Object Diagrams
2. Use Case Diagrams
3. Statechart Diagrams
4. Activity Diagrams
5. Sequence Diagrams
6. Collaboration Diagrams
7. Component Diagrams
8. Deployment Diagrams

(1) logical structure
   - application domain objects
   - associated media objects

(2) spatial presentation

(3) predefined temporal behaviour

(4) interactive control
Class Diagram Example

- UML notation unchanged
- $\text{Model}_{\text{static}}$ of MVC
- frameworks, e.g. application structure and associated media types

```
Media
  └── ContinousMedia
      ├── Audio
      └── Video
  └── DiscreteMedia
      ├── Animation
      ├── Graphics
      └── Image

MMApplication
  ├── ApplicationUnit
  └── ComposerEntry
      ├── TextEntry
      ├── Film
      └── Portrait
          └── MusicPiece
              └── MusicSheet
```

- $1..*\text{ Video}$
- $1..*\text{ Image}$
- $1..*\text{ Text}$
- $1..*\text{ ComposerEntry}$
- $1..*\text{ MusicPiece}$
UML Diagram Types vs. Multimedia Aspects

1. Class / Object Diagrams
   - logical structure
     - application domain objects
     - associated media objects

3. Statechart Diagrams
   - spatial presentation

5. Sequence Diagrams
   - predefined temporal behaviour
   - interactive control
**Statechart Diagram**

- **interactive behavior**
- **overall dynamics of multimedia application (MMA)** composed from dynamics of building blocks, e.g. scenes
- **notation like in standard UML**
- **coupling of event-driven and inherently predefined behavior**
  - *activity* of internal transition evaluates to a predefined temporal action sequence specified by a Sequence Diagram
  - alternatively, *actions* on state transitions or internal to states
UML Diagram Types vs. Multimedia Aspects

(1) Class / Object Diagrams

(1) logical structure
application domain objects
associated media objects

(2) spatial presentation

(3) predefined temporal behaviour

(4) interactive control

3) Statechart Diagrams

(5) Sequence Diagrams
Sequence Diagram Example

Music

int MTitle

KNMn: MusicSheet
KNMa: MusicPiece
Highlight: Marker
Sym42n: MusicSheet
Sym42a: MusicPiece

[MTitle == Mozart1]

[MTitle == Mozart2]
Extensions to the Sequence Diagram: Timing Constraints

Music

int MTitle

T [s]
0
max 0.2
[0.2; 0.5]
2:10
0:25

Sym42n: MusicSheet
Sym42a: MusicPiece

[MTtitle == Mozart2]

[MTtitle == Mozart1]

Summer 2001
Multimedia Software Engineering
Extensions to the Sequence Diagram: Timing Constraints

specification of an upper bound
max 0.2 is semantically equivalent to
[0; 0.2]

analogous specification of a lower bound unsing keyword min
min 0.3 is semantically equivalent to
[0.3; ∞)

specification of lower and upper bound for duration of time interval by an interval

precise specification of duration of time interval
2:10 is semantically equivalent to [2:10; 2:10]
Extensions to the Sequence Diagram: Complex Activations

- complex activation for an application object
- composed activation
  - multiple parallel presentations
- animation activation
  - automatic triggering of a sequence of activations
Extensions to the Sequence Diagram: Dynamic Assignment

- assigning media and presentation objects:
  - to each activation of an application object
    - `<media-object.media-type>`s and presentation objects (e.g. bounding box, audio channel)
  - can be optionally assigned

```
<table>
<thead>
<tr>
<th>:Audio</th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;2525.wav&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

presented in BBox1

```

```
<table>
<thead>
<tr>
<th>:Anim</th>
<th>BBox1</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Bild1.gif&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Bild2.gif&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Bild3.gif&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Bild4.gif&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Bild5.gif&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Bild6.gif&gt;</td>
<td></td>
</tr>
</tbody>
</table>

presented in BBox2

```
Other Extensions to the Sequence Diagram

**media filters**
- time-dependent function for changing a numeric presentation parameter, e.g. volume or brightness

**parameterization**
- starting time
- object links
- execution choice points

**timing tolerance**
- activation delay
e.g. for media preprocessing
- deactivation delay

- fade in
- permanent level 66%
- fade out
  - 0%
  - 100%
Extended Sequence Diagram: Summary

Music
int MTitle

\[ \text{MTitle} = \text{Mozart1} \]

ImageVideoBox

<KNMg:Graphics>

<KNMtrack:Audio>

Highlight: Marker

assigned presentation object

automatic triggering

media filter

assigned media object

composed activation

parameters

activation delays

time constraints
UML Diagram Types vs. Multimedia Aspects

(1) Class / Object Diagrams
   (1) logical structure
       - application domain objects
       - associated media objects
(2) Statechart Diagrams
   (2) spatial presentation
(5) Sequence Diagrams
   (3) predefined temporal behaviour
   (4) interactive control
Extension of UML: OMMMA-L

Object-Oriented Modeling of MultiMedia Applications - the Language

OMMMA-L:
- Class Diagrams
- Statechart Diagrams
- Presentation Diagrams
- Extended Sequence Diagrams

(1) logical structure
- application domain objects
- associated media objects

(2) spatial presentation

(3) predefined temporal behaviour

(4) interactive control
Example

Composer Encyclopedia

Wolfgang Amadeus Mozart was born in Salzburg (a town in the west of Austria) in 1756 in the midst of what was called the Age of Enlightenment (the period when people believed that science could advance human progress.) His father, Leopold, was also a musician who published a book called Violin School in 1756. His mother, Anna Maria, was an intellectual woman whose father, also called Wolfgang, was a bass singer.
Spatial Presentation (Layout)

Composer Encyclopedia

Alphabet

箭头动画

图像框

图像视频框

主题

维也纳第一卷

Wolfgang Amadeus Mozart was born in Salzburg (a town in the west of Austria) in 1756 in the midst of what was called the Age of Enlightenment (the period when people believed that science could advance human progress.) His father, Leopold, was also a musician who published a book called Violin School in 1756. His mother, Anna Maria, was an intellectual woman whose father, also called Wolfgang, was a bass singer.
Elements of Presentation Diagrams

- **visual presentation layout**
  - specified using bounding boxes of different shapes
    - **visualization elements:**
      - purely for presentation
    - **interaction elements:**
      - allow for generation of (user) events
      - **stereotypes for basic GUI elements,** e.g. button
        - hierarchical composition of bounding boxes
        - structuring of GUI by layout views (see below)
- **auditive presentation structure**
  - **iconic representation of audio channels**
6. UML & OMMMA

Structuring & Reuse: Composition of Presentation Diagrams

Presentation Views

- LViewEnc
- LViewCom

Diagram elements:
- Alphabet
- ArrowAnimation
- ImageBox
- ImageVideoBox
- HypertextBox

Navigation controls:
- Prev
- Next
- Play
- Pause
- Rew
- Ffw
- Stop

- Prev1
- Next1

Summer 2001
Multimedia Software Engineering
6. UML & OMMMA

Presentation Diagram Editor of OMMMA-Tool Prototype

tree view of presentation including audio channels

(layered) presentation view

(layered) bounding boxes of a presentation view
OMMMA-L covers all aspects

MVC (Model - View - Controller) - Pattern

(4) Interactive Control
Controller
Statechart Diagram

(2) Spatial Presentation
View
Presentation Diag.

(1) Logical Structure
Model_{static}
Class Diagram

(3) Predefined Temporal Behaviour
Model_{dynamic}
Ext. Sequence Diagram
Model Consistency

MVC (Model - View - Controller) - Pattern

- (4) Interactive Control
- (2) Presentation
- (3) Predefined Temporal Behaviour
- (1) Logical Structure

Model\_static

Model\_dynamic

consistency

Presentation Diagram

Statechart Diagram

Class Diagram

Ext. Sequence Diagram
OMMMA-L: Summary

- object-oriented modeling language for multimedia applications
- based on UML (Unified Modeling Language)
- new features
  - presentation diagram
  - extended sequence diagram
- formally defined abstract and concrete syntax
  - based on UML meta model
- OMMMA-tools available (extensions of Rational Rose)
Comparison

Dimensions of Multimedia Software Engineering

- technical foundations
- system software
- services
- usage
- areas
- model
- view
- controller
- communication
- architecture
- implementation
- design
- analysis
- abstraction

* MM-Framework
* OMMMA
* MM-Modeling
* „MM-Middleware“