UML 2
Ideas, Standardisation, and Practical Application

Prof. Mario Jeckle
Fachhochschule Furtwangen
mario@jeckle.de
http://www.jeckle.de
Content of this Presentation

- Ideas
  - more precise
  - smaller
  - executable

- UML 2
  - Proposals
  - Standardisation
    - Roadmap
    - Interdependencies
  - Metamodel
  - Practical Application
    - new Diagramms
    - enhanced Diagramms
In the Early Days there where just *Method War*
UML …

Some Backgrounds on a Standardisation Odyssey

Many working for all

Some working for many

Few working for few

... after that there where just Standardization War
UML ... A classical *Second System*
... Why do we Need a New Major Release?

**Evolution**
- Market has generated ...
  - new programming languages (e.g. C#, Python, PHP)
  - new application domains
    (e.g. server programming, real-time applications)

**Experience**
- UML has proven to be inadequate for some modelling tasks ...
  - sometimes there are too less constructs
  - sometimes there are too many of them
  - sometimes there are some of them, but practically no one can decide if they are the right ones

**Eradication**
- programming languages disappeared (e.g. C++)
- some ideas formerly closely linked with UML are now considered completely separate (e.g. process models)
UML 2

...Ideas is there really more than just evolution in it?

- The good thing about definitions is, there are so many of them to choose from, especially when they are contradictory
- Some diagrams lack precision

- UML is too hard to grasp for newbies to OO
- UML is too hard to implement by tool vendors
- UML needs to be tailored before usage in practical projects

- Some parts of UML are just there for visualization
- Some parts are not precise and expressive enough
- Vague semantics
UML 2
The Imperatives

• **Smaller**
  - Reduce number of modelling constructs
  - Reduce number of underlying concepts
  - Re-use core concepts

• **More precise**
  - Reformulate the meta model
  - Use OCL
  - Re-use core concepts w/o modification wherever possible

• **Executable**
  - Enhance state machines
  - Establish a closer link between static and dynamic parts
  - Re-use existing concepts currently outside UML
1. Language features which are not implemented by major tool vendors
2. Language features which are not popularised by major methods
3. Language features which are method specific or specific to an implementation language
4. Language features which lack precise semantics
UML 2
Retire or Deprecate Model Elements

1. Language features which are not implemented by major tool vendors
2. Language features which are not popularised by major methods
3. Language features which are method specific or specific to an implementation language
4. Language features which lack precise semantics

Hardly any method addresses the whole UML, i.e. every possible diagram with all specialities
UML 2
Retire or Deprecate Model Elements

1. Language features which are not implemented by major tool vendors
2. Language features which are not popularised by major methods
3. Language features which are method specific or specific to an implementation language
4. Language features which lack precise semantics

<table>
<thead>
<tr>
<th>ClassA</th>
<th>ClassB</th>
</tr>
</thead>
<tbody>
<tr>
<td>some : int</td>
<td>even : String</td>
</tr>
<tr>
<td>really : bool</td>
<td>more : short</td>
</tr>
<tr>
<td>important : float</td>
<td>important : float</td>
</tr>
<tr>
<td>stuff : byte</td>
<td></td>
</tr>
</tbody>
</table>

«friend»
UML 2
Retire or Deprecate Model Elements

1. Language features which are not implemented by major tool vendors
2. Language features which are not popularised by major methods
3. Language features which are method specific or specific to an implementation language
4. Language features which lack precise semantics

Object1 : ClassA

- some = 1
- really = true
- important = 3.1
- stuff = 42

«copy»

Object2 : ClassA

- some = 1
- really = true
- important = 3.1
- stuff = 42

«become»
UML 2
Introducing: UML Layering

Level 3  Complete
Level 2  Intermediate
Level 1  Basic
Level 0  Foundation

• Idea borrowed from SQL standardisation
• Operationalises notion of UML support
• Even less UML can still be UML
Approaching UML 2

Unified Modeling Language 2.0

- UML is no longer one single language, if this ever was the case
- Four separate RFPs will create four separate but highly interrelated responses
Approaching UML 2
Proposals

• Various proposals for all four RFPs
  • **Superstructure**: 37 Letters of Intent (LOIs); 5 initial submissions by 28 companies
  • **OCL**: 30 LOIs; 4 initial submissions by 10 companies
  • **Infrastructure**: 36 LOIs; 5 initial submissions by 28 companies
  • **Diagram Interchange**: 6 LOIs; 3 initial submissions by 6 companies

• Submitted by single companies, and consortia

• Focussing on single aspects of UML v1.x to improve, proposing new diagrams, or even suggesting to abolish parts of the UML at all
Some complex things should definitely become easier ...
Some cluttered diagrams should definitely become more readable ...
Approaching UML 2 Proposals

Some boundaries have definitely to be tumbled down ...
Approaching UML 2 Proposals

- Superstructure and Infrastructure:
  Most complete, widely supported and hence interesting proposal is submitted by the UML 2 Partners group
  - Members:
    Alcatel, Computer Associates, Ericsson, Hewlet-Packard, IONA, Kabira Technologies, Motorola, Oracle, Rational Software, SOFTEAM, Telelogic, and Unisys
  - Supporters:
    Advanced Concepts Center, Ceira Technologies, Commissariat à L’Energie Atomique, Compuware, DaimlerChrysler, Embarcardero Technologies, Enea Business Software, France Telecom, ...
Roadmap
... the Initial One

- **Infrastructure**
  - RFP
  - LOI
  - Initial Submission
  - Revised Submission

- **Superstructure**
  - RFP
  - LOI
  - Initial Submission
  - Revised Submission

- **Object Constraint Language**
  - RFP
  - LOI
  - Initial Submission
  - Revised Submission

- **Diagram Interchange**
  - RFP
  - LOI
  - Initial Submission
  - Revised Submission

Dates:
- 2000-09-15
- 2001-03-02
- 2001-08-20
- 2001-10-22
- 2002-06-03
- 2002-08-19
Roadmap ... as it Stands Today

Proposals to all RFPs should be voted by the ADTF and the AB at time of this presentation, i.e. the RTF starts it's work. After it's completion UML 2 is finalized.
UML 2
What’s in it -- Metamodel

- Metamodel
  - Completely restructured
  - Concepts refined to be more orthogonal and thus enable reuse
Alternatively to the diagrammatic notation for behavioural diagrams now a tabular notation is offered.
UML 2
What’s in it – New Diagrams

- The *Communication Diagram* focuses on the interaction between lifelines.
- Basically it is a variant of the *Sequence Diagram*. 
UML 2
What`s in it – Enhanced Diagrams

Synchronized Producer-Consumer

Produce

Consume
UML 2
What’s in it – Enhanced Diagrams

Synchronized Producer-Consumer

Produce

Consume
UML 2
What’s in it – Enhanced Diagrams

- **Activities** will replace *Activity Graphs*
- **Activities** are based on Petri nets with queuing
- **Activities** are no longer based on state machines
- **Activities** are better suited for parallel flows
- **Activities** are almost directly executable
- **Activities** could be proved to be deadlock free