Rich Meta Object Facility
Ralf Buschermöhle, Jörg Oelerink

ICFEM 2008
October 27-31, 2008
Kitakyushu, Japan
Workshop UML&FM’08
2 Where does this/do I come from?

Some Facts
OFFIS Institute for Information Technology (Oldenburger Forschungsinstitut für Informatik- Werkzeuge und Systeme)
- Founded 1991 as a not for profit organization
- Closely related to University of Oldenburg
  - Sonderforschungsbereich: AVACS (Automatic Verification and Analysis of Complex Systems)
- Members: State of Lower Saxony, University Oldenburg, Professors of IT and related studies
- Department Transportation, Research Areas
  - Formal Verification & Safety Analysis
  - WCET Analysis
  - Human Centered Engineering
  - Cooperative Mobile Systems
  - Process Analysis & Optimization
- OFFIS Service and Consulting
  - Testing (Automated Test Case Generation)
3 MOF

Some facts ...
- MOF 2.0 developed January 2006 by the OMG
- Differentiation in Essential und Complete MOF
- Meta Language for UML2, SPEM, SysML

What is MOF exactly?
- Language to describe other languages (Syntax)
- Natural language description of Behavior (e.g., operations)
- With XMI possible to read/write models/meta-models with corresponding files directly downloadable from the OMG homepage

Central Usage (right now): Repository
- “Easy” to say an „a“ is related to a „b“ in a language „c“ without
  - Parse Files (XML, XMI, Language Layer)
  - Implementing required data structures
- Positive Project Experience
4 RMOF

- MOF is not enough, why?
  - Not formal
  - No Algorithms

- Why is it interesting to formalize & enrich MOF?
  - Implementation → Formalization → Making decisions
    (e.g., Damm & Harel “LSCs: Breathing Life into Message Sequence Charts”)
  - Are we making the „right“ decisions?
  - Communicate & explore semantically different methodologies
    (e.g., “variable domains”, concurrency, event queues)
  - Drawbacks of different formal methodologies:
    - Time-consuming to understand/apply
    - Different abstraction levels
    - Redundancy
    - No easy way to integrate & explore different approaches with all the positive effects of a MOF repository approach

- RMOF solves these problems
5 RMOF in Short

- Formal Methodology
  - Close to MOF/UML2 Specification
    - Data structures compliant to EMOF / Layer1 to CMOF + Constraints
    - UML2 Enrichments to support Algorithms
  - (Flexible) Symbolic Transition System

- Algorithms
  - State Machines (SM)
    - With different degrees of concurrency (effect, transition, state machine)
    - (Synchronous) Operation calls
    - Constraint SMs
    - Derived Values SMs + Default Derived Union Computations
    - Oberserver SMs
  - Action Language (layer & collection type management, arithmetic operations, …)

- Implementation
  - Used Frameworks (Graphical Representation) vs. our Implementations (Simulation)
  - Layer composition, debugging
6 Formal RMOF::Syntax & Semantics

7.3.33-93 **NamedElement** = (Class, \{(ownedAttribute, (name, qualifiedName, namespace)), (isAbstract, true), (generalization, {Element})\}) is an element in a model that may have a name.

+ name = (Property, \{(aggregation, composite [at]), (type, Kernel::String), (lowerValue, "1 [prev. 0]")\}) of the NamedElement.

Defined in: Core::Abstractions::Namespaces::NamedElement,
Core::Basic::NamedElement, Core::Constructs::NamedElement

\[ \text{down} : \sum M \times M \times 2^N \to 2^N \text{ determining the instance IDs of a set of IDs is defined as follows:} \]

\[ \text{down}_{M,l.n}(N) := \begin{cases} \bigcup_{n.i \in l.n_i} \{n.i\} & \text{if } \exists l.n_i \in M : l.n_i \to l.n \\ \bigwedge \bigcup_{n.i \in l.n_i} \{up(n.i)\} = N & \text{else} \end{cases} \]
Definition 20 (System Semantics) \( \text{Let } \mathcal{M} \text{ be a system. The semantics of } \mathcal{M} \text{ is defined as:} \)

\[
\text{STS}(\mathcal{M}) = (V, \Theta, \rho), \text{ where}
\]

**System Variables:** \( V := \{sc : T_{sc}(\mathcal{M}), \bullet.m.c : Q, \text{sysfail} : \mathbb{B}\} \).

**Initial condition:** \( \Theta := \bullet.m.c = q_0 \land \text{sysfail} = \text{false} \)

**Transition relation:** The intermediate predicate \( \rho_0 \) composes the above introduced sub-predicates and additional conditions on their application within objects’ life-cycle as follows:

\[
\rho_0 := (\neg \text{sysfail} \land \exists(q, \gamma, q') \in \bullet.m.tr : \bullet.m.c = q \land \bullet.m.c' := q' \\
\land (\rho_{\text{assign}}(l) \lor \rho_{\text{guard}}(l)) \land q \neq q_x) \lor q = q_x
\]
8 RMOF::Data Structures for Data Structures

Package
- ownedMemberPackage [0..*] {unique} : PackageableElement
- <s> package [0..1] : Package

Class
- ownedAttribute [0..*] {unique} : Property
- <s> superClass [0..*] {unique} : Class
- ownedOperation [0..*] {unique}

Property
- aggregation : AggregationKind
- isDerived : Boolean
- isDerivedUnion : Boolean
- <s> subsettedProperty [0..*] {unique} : Property
- <s> opposite [0..1] : Property
- <s> class : Class
- <s> defaultValue [0..1] : String
- <s> signature [0..1] : String

MultiplicityElement
- isOrdered : Boolean
- isUnique : Boolean
- lower : Integer
- upper : Integer

Enumeration
- ownedLiteral [1..*] {unique} : EnumerationLiteral

Comment
- <s> body : String
- <s> annotatedElement [0..*] {unique} : Element
9 RMOF::Data Structures for Algorithms

**StateMachine**

- transition [0..*] (unique) : Transition = null
- subvertex [0..*] (unique) : Vertex = null
- operation : Operation = null
- initialActive : Boolean = false
- rtc : runToCompletion = false
- <s> trAttach [0..*] (unique) : TrAttachment = null
- <s> calledOperation [0..1] : Operation = null

**Operation**

- ownedParameter [0..*] (unique) : Parameter = null
- stateMachine : StateMachine = null
- <s> calledFromStateMachine : StateMachine = null
- subOperationCallString [0..1] : String = null
- <s> clazz : Class = null
- effect [0..1] : String = null
- calledParameters [0..*] (unique) : String = null

**Transition**

- <s> source : Vertex = null
- <s> target : Vertex = null
- container : StateMachine = null
- <s> attachment [0..1] : TrAttachment = null

**TrAttachment**

- guard [0..1] : String = null
- <s> container [0..1] : StateMachine = null
- effects [0..*] (unique) : Effect = null
- <s> transition [0..1] : Transition = null

**GlobalStateMachine**

- <s> stateMachines (managed by GlobalStateMachine) [0..*] (unique) : StateMachine = null
- <s> stateMachineHandlers [0..*] (unique) : StateMachineHandler = null
- stateMachineHandlerCount : Integer = 0
- constraintsAllTrue : Boolean = true
- o : Operation = null
- hasReturnParam (Has the current operation a Return Parameter?) : Boolean = false
- paramCount (Parameter Counter) : Integer = 0
- stateExitCount : Integer = 0
- stateExitNo (State Exit Number - total) : Integer = 0
- globaleStateMachine

**globaleStateMachine**
10 RMOF::Action Language

- Directly supported by the platform (on each layer) e.g.,
  - Layer Management
    - up()
    - down()
    - filter()
    - getObjectID()
    - read() & write() object attributes
    - add() & remove() objects
  e.g., filter(down(down(filter(up(*), {{name, StateMachine}}))), {{isActive, true}})
  - Navigation (e.g., a.b.c = 42)
  - Collection Type Management (set, bag, ordered set, sequence e.g.,
    a.b[0].c[3] = 42, remove(s,e), a subBagOf b (flat or deep)
  - Arithmetic Operations (a+b*c = 24, if a = 2, b = 3, c = 4)
  - Simulation related (break(), wait())
  - Coloring of graphical objects
  - … otherwise an Operation is called
11 RMOF::Global State Machine
RMOF::Implementation::
13 RMOF::Implementation

RMOF Specification

RMOF

(Meta-)Model

\(M_k\)

\(M_{k+1}\)

(Meta-)Model

RMOF Implementation (based on EMF)

Ecore

Ecore Meta-Model

Ecore Model

Simulation Environment

Ecore

Ecore Meta-Model

Ecore Model

Ecore

Ecore Meta-Model

Ecore Model

Ecore

Ecore Meta-Model

Ecore Model

Ecore

Ecore Meta-Model

Ecore Model

...
15 Example Instantiation / Instantiation plan

Data Structures

Algorithms & ...

RMOM Core Meta Model

(CMOF Layer)

UML2, SPEM, SysML
16 Summary

► RMOF Specification
  ► Formal Variant of MOF
  ► +Algorithms
    ► (Simple) State Machines, supporting e.g., Operations, Constraints, Derived Value Computations with different degrees of concurrency
    ► Action Language, supporting e.g., Layer & Collection Types Management, Simple Arithmetic Operations, Highlighting

► RMOF Implementation
  ► Based on Java, Eclipse
  ► EMF: Domain Model (Model Access, Persistence, Item Providers)
  ► GMF: Graphical Representation
  ► Enrichment to support arbitrary Modeling Layers
    ► Layer Composition
    ► Ecore/Genmodel Generation
  ► Simulation Environment
    ► Core Layer is RMOF triggered
    ► Simulation/debug/… views
17 Future Work

- Minor Editor Improvements
  - Edit support (e.g., highlighting matching braces)
  - Only relevant editors pop-up during a simulation run
  - Speed things up
- Adding Modeling Layers
  - CMOF (Associations, Constraints, Derived Values)
  - UML2 (Class Diagrams, State Chart Diagrams, Activity Diagrams)
  - SPEM
- Adding Platform Bindings
  - Implementation: C++
  - Analysis: Symbolic (Model Checking), Heuristical & Statistical Analysis Methods
- Modell Import & Export (OMG XMI Metamodels)
- Extensions
  - Model Comparison/Search/Merge
  - Model Storage & Versioning (DB)
  - Distributed Modeling
Thank you! Questions?

That concludes my two-hour presentation. Any questions?

DID YOU INTEND THE PRESENTATION TO BE INCOMPREHENSIBLE, OR DO YOU HAVE SOMETHING OF RARE "POWER-POINT" DISABILITY?

Are there any questions about the content?

There was content?

Complete specs/software please contact me:

buschermoehle@offis.de