Joint Workshop - Using ISO 15046 Application Scenarios with CSL, RFA & Encoding, 22. September, 0830 - 1200

- 0830-0840 : Introduction, Olaf Østensen
- 0840-0930 : CSL - UML Guidelines (Arne J. Berre)
- 0930-1000 : Rules for Application Schema w/example (Steinar Høseggen)
- 1000-1020 : Coffee Break
- 1020-1100 : Encoding w/example (David Skogan)
- 1100-1200 : Use of 15046 - The goal of Interoperability
  Profiles (Doug O’Brien)
- Summary of req for Use of UML by ISO/TC211 standard developers (David), tool developers (OGC)(Arne), application developers (Steinar)
- Experiences with UML (part Spatial Schema, Spatial Operators. Switzerland, …)
Using ISO 15046 – Modelling and Encoding an Application
A presentation of CSL, RFA and Encoding and relationships to the different parts of ISO 15046

Since the last ISO/TC211 and working group meetings in Victoria there have been discussions among some nations regarding what an application schema is, how this is related to the other parts of ISO/TC 211, and the impact of resolution 55 (regarding the choice of Conceptual Schema Language).
There is a concern that there is not sufficient agreement and understanding within ISO/TC 211 on how Conceptual Schema Language (CSL), Rules for Application Schema (RFA) and Encoding relate to each other, and how these three work items are related to the content of other parts of ISO 15046, including parts 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 20 and perhaps others.

Part of this challenge may be that CSL, RFA, and (to some extent also) Encoding are better understood by information systems specialists than by others who may not have had extensive experience with using CSLs or developing application schemas. At the same time, there is a strong desire within ISO/TC211 to make the ISO 15046 a standard that is as compatible as possible with the IT mainstream.
Another part of the challenge also may be that CSL, RFA, and Encoding are significant topics in themselves and their relationships to the other work items are complex. It simply takes time and effort to communicate and agree upon these ideas. As an approach to meeting these challenges, a workshop will be arranged to give an introduction to CSL Modelling guidelines, RFA, and Encoding. An example of a simple application schema will be used to present the underlying concepts of these three work items, taking into consideration linkages to other parts of ISO 15046, including spatial schema, quality principles, and eventually metadata. The application schema example will also be used to demonstrate encoding using XML.
Another purpose of this workshop will be to propose requirements for those parts of ISO 15046 that are related to, and perhaps impacted by CSL, RFA, and Encoding, and to propose requirements for those environments that wish to convert existing data to the ISO-standard. In doing this, we hope to make a significant contribution to the progress of ISO/TC211 in these areas.
Introduction to Guidelines for use of UML

• Introduction to UML
• The usage of basic data types (OCL-based - covered under encoding presentation)
• The usage of packages
• The usage of classes
• The usage of attributes
• The usage of associations
• The usage of constraints
• Documentation of models
• The use of stereotypes

* Contributions by Spatial schema/operators P Ts by John Herring, Charles Roswell, David Parker & OGC David Case, Cliff Kottman
Resolution on CSL for specifying ISO 15046, adopted by subsequent national vote

ISO/TC211 shall use the *Unified Modeling Language* (UML) static structure diagram with the ISO *Interface Definition Language* (IDL) basic type definitions and the UML *Object Constraint Language* (OCL) as the conceptual schema language for specification of the normative parts of ISO 15046. *This requirement shall be implemented prior to submission of the parts for Committee Draft.*

The reason for this decision is that *the goal of ISO/TC211 is to create a framework to enable syntactic/technical interoperability and to support semantic interoperability, while supporting multiple interchange formats and multiple service implementations.* UML is selected as the conceptual schema language for producing specifications that can support the creation of such a framework.
Impact of Proposed Resolution

Parts of ISO 15046

Usage

UML - Graphical and lexical (Normative)

(UML mappings can be semi-automatic)

Application Schema

Data Interchange

Service Implementations

EXPRESS
UML
INTERLIS
... 
Generic
EXPRESS-based
XML
...

MIDL/COM
IDL/CORBA
SQL/SQL/MM
EXPRESS/SDAI
ODL/ODMG
...
The General Feature Model -> a meta model

Feature Type

- Characterized_by
- Association_described_by

Feature Attribute Type

- Characterized_by
- Behaviour_described_by
- Observes values of

Feature Operation Type

- Depends on
- To_another

Feature Relationship Type

- Characterized_by

Thematic Attribute Type

Spatial Attribute Type

Temporal Attribute Type

Logical Relationship Type

Spatial Relationship Type

Generalisation Relationship Type

Aggregation Relationship Type

Quality Subelement Type

- expressed as
- Relationship from all entities above

Quality_described_by

Described_in_terms_of

Is_coordinate_of
CSMF is described in a language based on Meta-metamodels. Metamodels conforming to Meta-model 1 are used to model schemas. Application models are defined using Application Schema, which are based on Meta-model n. Information bases are then created from these models.
UML Core -> a metamodel
The General Feature Model as an extension to the UML Core metamodel
Evolution of the UML

Submission of UML 1.1 to OMG for adoption, Sept ´97

UML 1.1 (Sept. 1997)

IBM, Taskon, ....

UML 1.0

Public feedback

June ´96 & Oct ´96

UML 0.9 & 0.91

UML Partners’ Expertise

OOPSLA ´95

Unified Method 0.8

Booch ´93

OMT - 2

Booch ´91

OMT - 1

OOSE

Other methods

Submission of UML 1.1 to OMG for adoption, Sept ´97

Evolution of the UML

IBB, Taskon, ....
Introduction to Guidelines for use of UML

• Introduction to UML
• The usage of basic data types (OCL-based - covered under encoding presentation)
  • Basic data types
  • Collection types
• The usage of packages
• The usage of classes
• The usage of attributes
• The usage of associations
• The usage of constraints
• Documentation of models
• The use of stereotypes

In order to use UML as a specification language, we need to create precise UML models -> classes with all public attributes and operations with types, and associations with role names.
Class notation

**GM_Object**

- `+ mbRegion : GM_Object`
- `+ representativePoint : DirectPosition`
- `+ SRS() : SpatialReferenceSystem`
- `+ transform(SRS : SpatialReferenceSystem) : GM_Object`
- `+ equals(object : GM_Object) : Boolean`
- `+ distance(object : GM_Object) : Distance`
- `+ dimension() : Integer`
- `+ dimension(point : DirectPosition) : Integer`
- `+ coordinateDimension() : Integer`
- `+ envelope() : Envelop`
- `+ boundary() : Set(GM_Object)`
- `+ buffer(radius : Distance) : GM_Object`
- `+ convexHull() : GM_Object`
- `+ centroid() : DirectPosition`
- `+ representativePoint() : GM_Point`
- `+ isInComplexes() : Set(GM_Complex)`
- `+ isPartOf(geomCplx : GM_Complex) : Boolean`
- `- universe() : GM_Complex`

**GM_Primitive**

(from GeomPrimitive)

- `+ boundary() : Set(GM_Primitive)`

**GM_Complex**

(from GeomComplex)

**SpatialReferenceSystem**

(from DirectPositioningSchema)
Visibility

ClassName

/ /* derived attribute

+ /* public visibility

# /* protected visibility

- /* private visibility

underline /* class level
Packages in UML

- A mechanism to group related classes of a model.
- Packages dependencies summarize dependencies between classes of different Packages.
- Packages can contain other Packages.
- Can show the Class/Entities found in a given Package.

Geometry Package
(from Logical View)
+ Accuracy
+ TopologicalGeometry
+ RepresentationalGeometry
+ TemporalGeometry
+ Geometry
+ Vector{dimension}
+ SpatialVector
ISO/TC211 Parts Packages

SpatialSchema

Temporal Schema

Portrayal Schema

QualitySchema

DirectPositioningSchema

Gazetter Schema

Feature Catalogue

Metadata Schema
Associations in UML Diagrams

- Purpose: To show relationships between model entities, To define multi-way constraints

```
AssociationName
   ^      ^
  2     1..*
RoleName          RoleName
```

- Multiplicity of an Association

- Generalization
- Dependency (Client-Server)
- Association Notation used to “anchor” at note to a model entity
- Aggregation
- Composition (Strong Aggregation)
- Link Attribute
Associations Between Classes: Cardinality “Decorations”

- UML notations for showing the Multiplicity of Associations:
  - “1” - denotes one and only one (if no Multiplicity is shown the default assumed is one)
  - “0..1” - denotes from zero or one
  - “m..n” - denotes from m to n where m and n are natural numbers
  - “*” - denotes from zero to any positive integer.
  - “0..*” - denotes from zero to any positive integer.
  - “1..*” - denotes from 1 to any positive integer.
  - … or comma separated lists of above (1, 3, 14..50)
UML extensions: Stereotype, tagged values and constraints

**Stereotype**

```
<<interface>>
```

**Constraint and Comment**

{Person.employer = Person.boss.employer}

**Element properties, tagged value**

{author = “Joe Smith”, status = analysis}

<< stereotype >>

Subclass of existing modeling element - may have additional constraints and code generators will treat stereotyped element specially.
### Mandatory, Optional, Conditional

#### <<Feature>>
Building

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>string</td>
</tr>
<tr>
<td>NoOfFloors</td>
<td>short</td>
</tr>
<tr>
<td>Residents</td>
<td>sequence &lt;string&gt;</td>
</tr>
</tbody>
</table>

<<cond>> isLarge : boolean

NoOfFloors > 10 implies isLarge = true
OCL - Object Constraint Language

First order predicate logic, for boolean expressions - included in UML 1.1

Can be used for:
- invariants, value restrictions, pre and post conditions

Expressions with:
- and, or, not, implies, exists, forall,
- collections (select, reject, collect, iterate)

Person
- self.age > 0

Married people are of age >= 18
- self.wife->notEmpty implies self.wife.age >= 18 and
- self.husband->notEmpty implies self.husband.age >= 18 and
Documentation of models

For each class there should be a textual description for describing semantics as follows:

Operations:

Operation: name
Description: <text>
Precondition: <text + OCL>

Input parameters: <description>
Output parameters: <description>
Return value: <description>
Exceptions: <description>
Postcondition: <text + OCL>

Constraints: < any sequencing constraints in the use of this operation with other operations? >

Other Constraints:
Any other constraints to be described. <text + OCL>
Some UML Tools

• Rational Rose: www.rational.com
• Select Perspective: www.selectst.com
• Popkin SA/Object Architect: www.popkin.com (See also see: www.mkp.com/umldg
• WithClass: www.microgold.com/
• VISIO Professional
• Oracle’s Designer (ODD Beta Version)
• **OOram role modeling UML tool with Rose-Link** (freely available from SINTEF/Taskon for ISO/TC211 work -> mail: Arne.J.Berre@informatics.sintef.no)
Interoperability between UML tools - exchange of UML models

Interoperability between tools to be supported by OMG RFP (Stream-based Interchange Format - XMI based on XML) - anticipated for 1999

Until then - exchange with proprietary tool-formats, and documents with diagrams.
Information Modeling - Domain modeling for ISO/TC211 standard developers

Phase 0: Identify scope and context

Phase 1: Identify basic entities

Phase 1b: Is the modeling approach consistent with the approach described in the rules for application schema document

Phase 2: Specify relationships, attributes and possibly operations

Phase 3: Completion of constraints

Phase 4: Model definition harmonisation – with sub-models and other work items
Service modeling for ISO/TC211 standard developers

Phase 0: Identify scope and context – use cases

Phase 1: Identify basic service responsibilities

Phase 2: Specify operations, attributes and service relationships

Phase 3: Completion of constraints on operations

Phase 4: Service definition harmonisation
ISO/ODP overview + Interoperability views

Organisational Interop

Enterprise viewpoint

Domain viewpoint

Computational viewpoint

Engineering viewpoint

Information viewpoint

(Domain Model)

(Distribution structure & patterns)

(Infrastructure mappings)

(Data Interop)

(CORBA, Java APIs, ActiveX, DCOM, SQL/ODBC, ODMG, ...)

Services Interop

(Use Case tools)

(Use Case models)

(Distribution models)

(Distribution Interop)

Technology Interop

(UML + Role-modelling tools)

(UML + Organisation Interop)

(Data Interop)

Use Case tools

Process-modelling tools

Frameworks for distribution concerns

SINTEF Telecom and Informatics
ISO/TC211 and OpenGIS

ISO

Spatial sub schema

Coord Ref. Sys

Meta data

Quality sub schema

General Feature Model

Rules for Application Schema

STD

Abstract Service Spec
(UML w/precision)
(Common Imp.spec)

Direct C++/Java
Portability/Interop Interface

Imp. Spec ODMG/ODL

Imp. Spec SDAI/EXPRESS

Imp. Spec COM/MIDL

Imp. Spec CORBA/IIDL

Imp. Spec ODBC/SQL SQL3/MM

1. Conformance

4) Portability/Interoperability
interface possible

2-Conformance

3. Reverse mapping for
Portability and Interoperability

OGC

Spatial sub schema

Feature rel.ship

Imp. Spec COM/MIDL

Imp. Spec CORBA/IIDL

Imp. Spec ODBC/SQL SQL3/MM

30

OGC

ISO

SINTEF Telecom and Informatics
ISO Requirements on Abstract Service Specifications to be adapted as ISO standard

- In UML (following modeling guidelines) on sufficient level of precision (interfaces, so that:
  - 1) Conformance can be shown with the relevant ISO Parts
  - 2) Conformance between an abstract specification and an implementation specification can be verified. Necessary for potentially new implementation specifications (ODMG/ODL, SDAI/EXPRESS, ...)
  - 3) Interoperability between implementations from different implementation specifications can be enabled, by showing reverse mapping from implementation specification to abstract specification
  - 4) A Portability layer can be made (on the cost of performance)