Model-driven approaches to large-scale e-business system development

Steve Cook

IBM Distinguished Engineer

Visiting Professor, UKC

IBM

Agenda

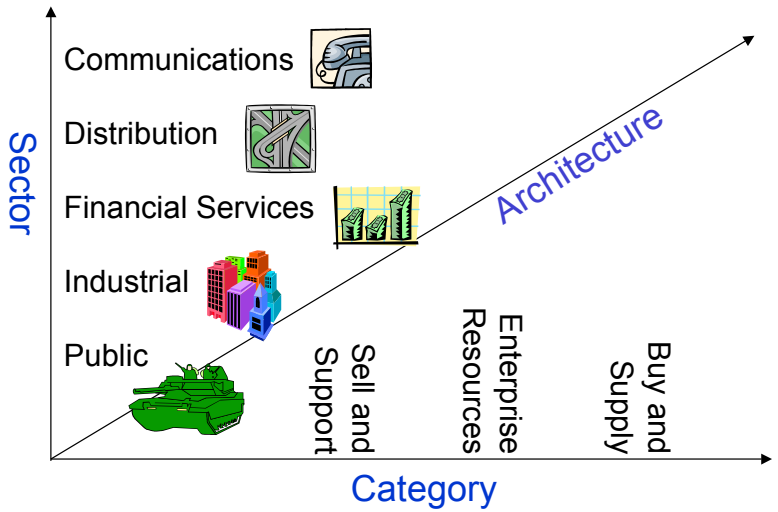
Why models?

The anatomy of models

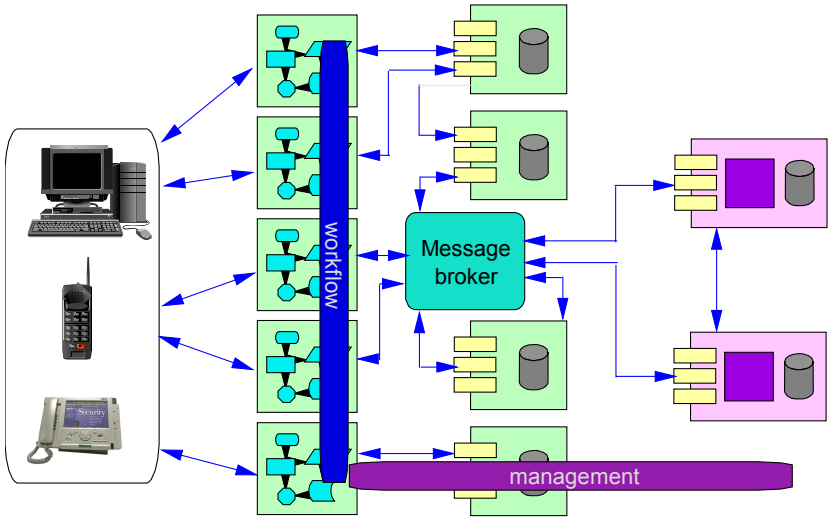
From models to code

Standards and maturity

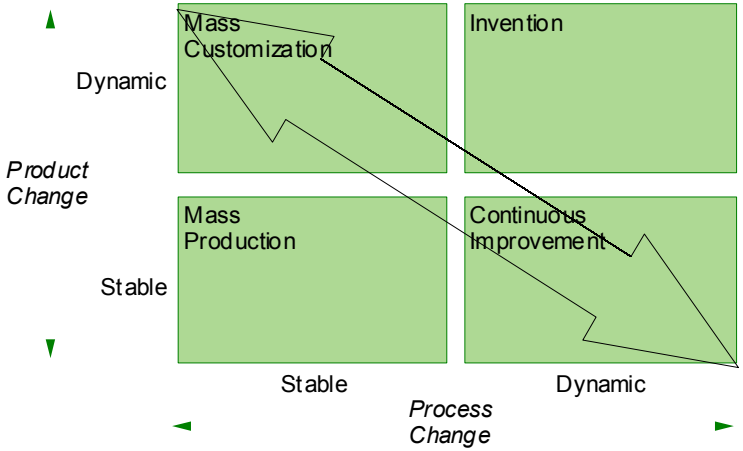
e-business systems are developed in a large, multi-dimensional space



The architecture is complex and layered and almost always involves legacy and package integration



How can we "mass-customize" solutions in this complex space?



Various approaches have been tried

Reusable Code (Objects and Components)

Code is too context-dependent to be very reusable

Level of abstraction (code) is too fixed and physical

It is too difficult to find the part you need

"Knowledge Management"

Representation tends to be in disconnected silos, hierarchical, using text and pictures with no semantics

Stored elements have no semantic foundation, and no notions of refinement or composition

My thesis

There is a "middle way" which has the potential to deliver a much greater degree of reuse

This "middle way" is based on modelling as a fundamental technology

Models:

Are a formal representation of some aspect of a system from a particular viewpoint

Must be precise, abstract and verifiable

Should be easy to understand

Must be capable of composition and refinement

Agenda

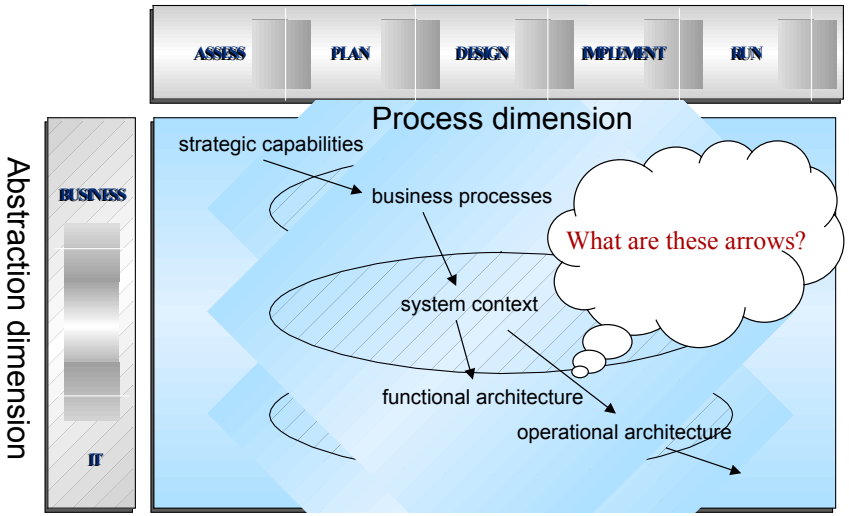
Why models?

The anatomy of models

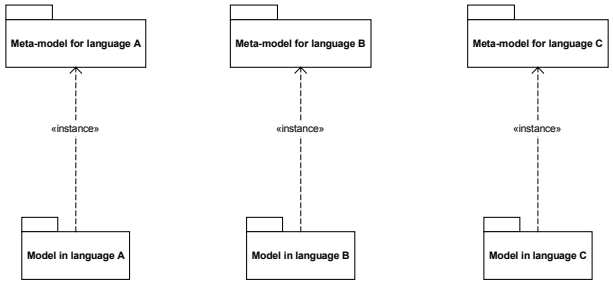
From models to code

Standards and maturity

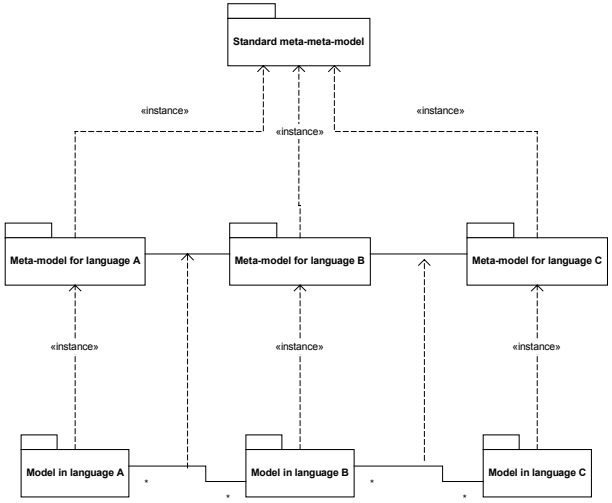
There are many kinds of inter-related model that apply at many levels



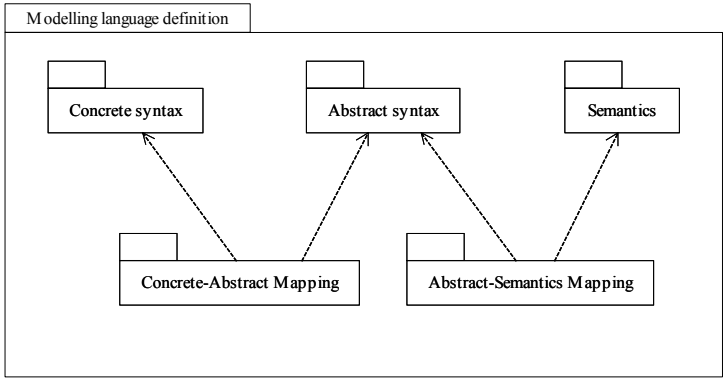
Each model is constructed from elements defined by a meta-model



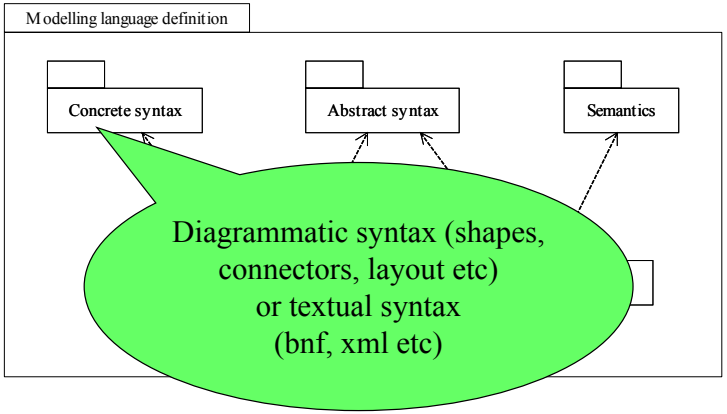
Models can only be related precisely if their languages are related precisely



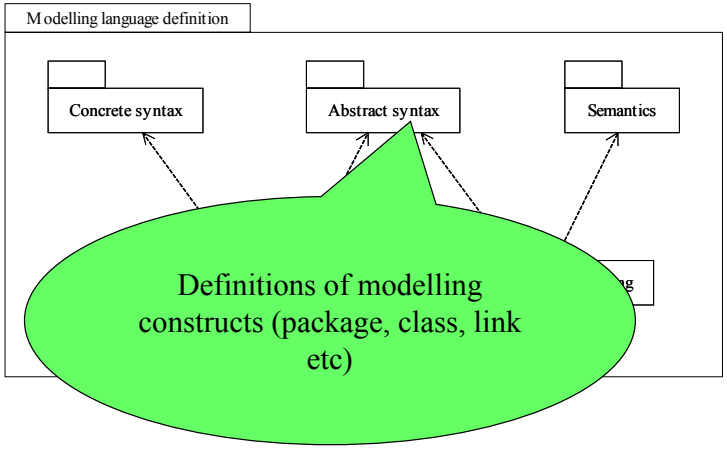
A modelling language has three main aspects



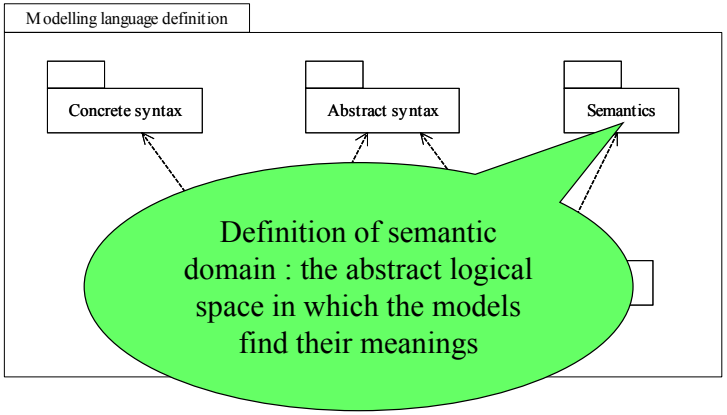
A modelling language (meta-model) has three main aspects



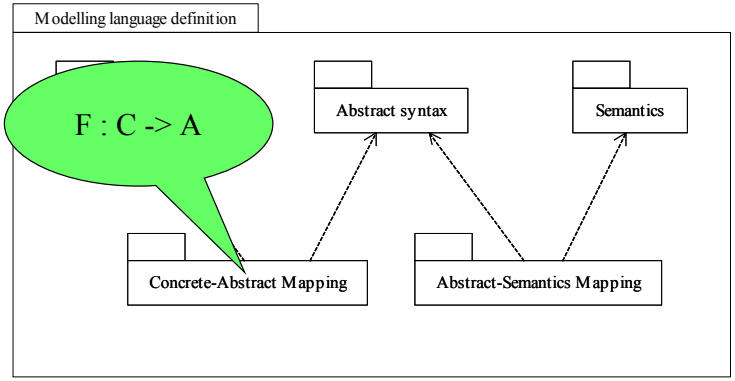
A modelling language (meta-model) has three main aspects



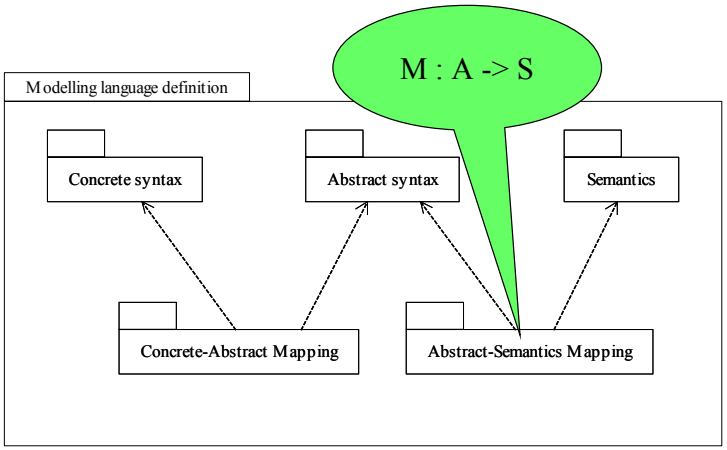
A modelling language (meta-model) has three main aspects



A modelling language (meta-model) has three main aspects



A modelling language (meta-model) has three main aspects



A standard meta-meta-model should fully support the precise definition of modelling languages

Definition of concrete syntax(es) (shapes and layout, physical interchange formats)

Definition of abstract syntax (concepts and relationships, well-formedness rules)

Definition of semantics (semantics domain)

Definition of mappings between domains

Agenda

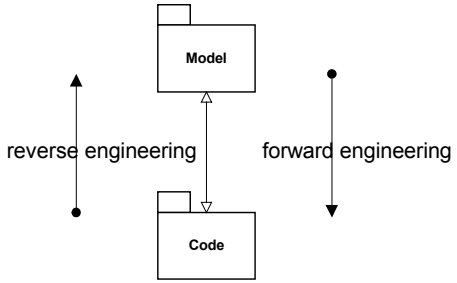
Why models?

The anatomy of models

From models to code

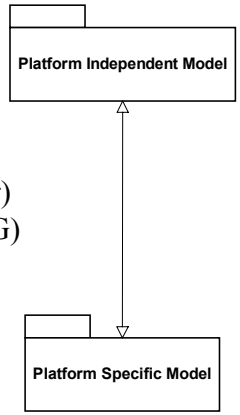
Standards and maturity

Conventional approaches for mapping models to code are much too simplistic

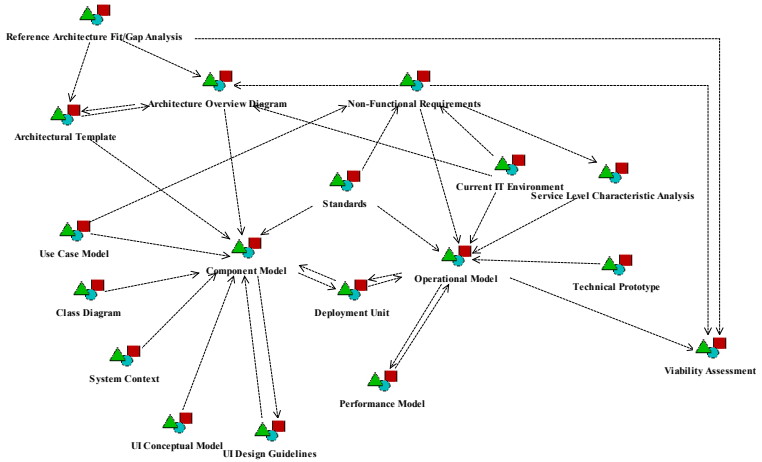


Abstracting platform differences is necessary, but not sufficient

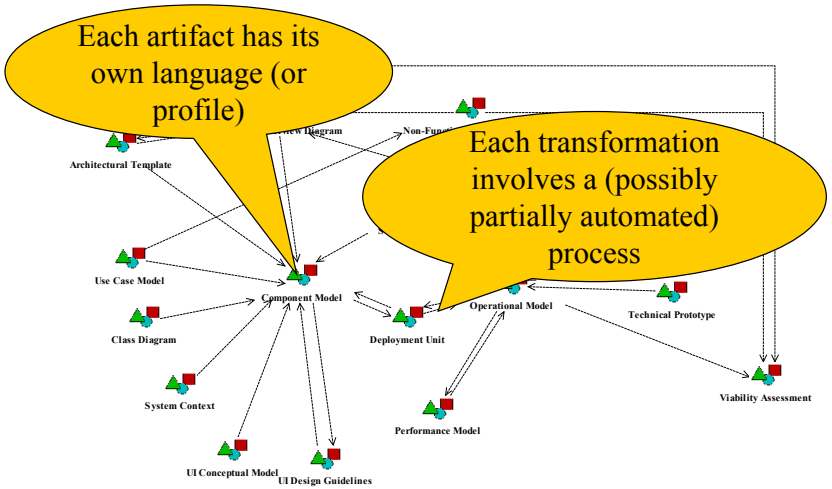
Recursive Design (Shlaer-Mellor)
Model-Driven Architecture (OMG)



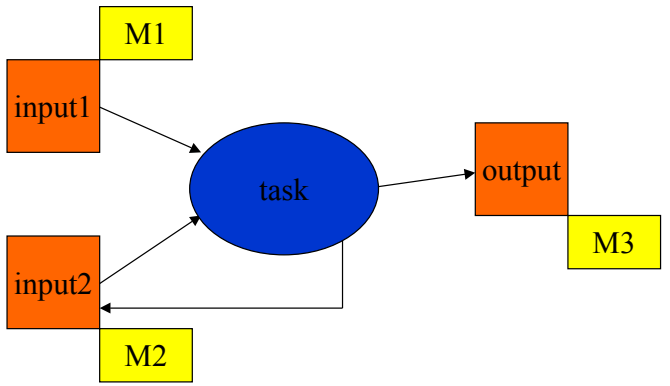
Numerous work products must be produced on the way from requirements to implementation and operation



Numerous work products must be produced on the way from requirements to implementation and operation



A model of the process can be coupled to metamodels for the work products



Agenda

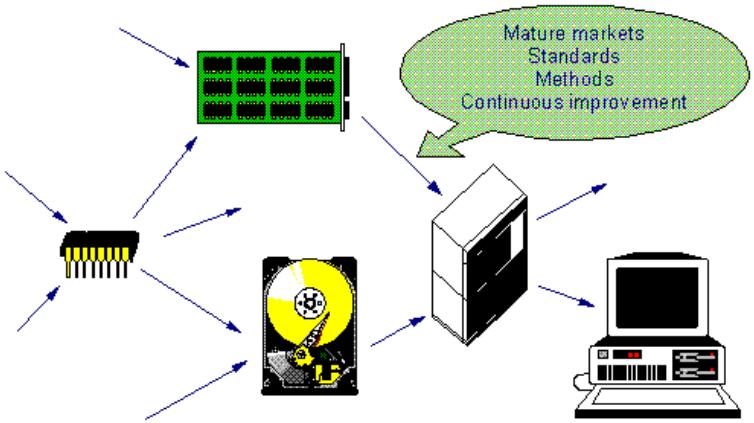
Why models?

The anatomy of models

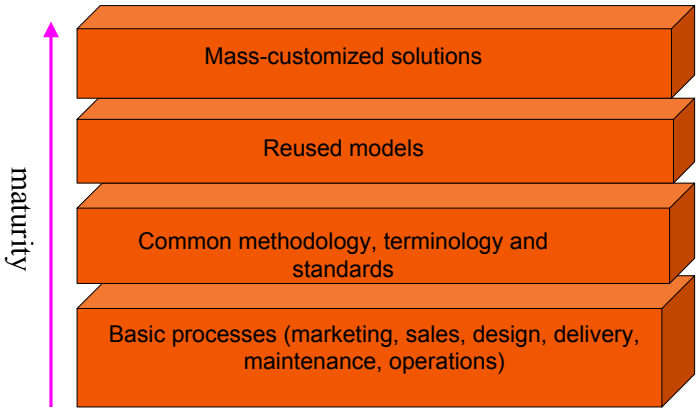
From models to code

Standards and maturity

Mass-customization requires mature value networks in the industry



Mass-customization requires maturity of the development organisation



The emerging standards in the modelling area are UML, MOF, XMI, and CWM

UML : Unified Modeling Language

and its profiles

MOF : Meta-Object Facility

XMI : XML Metadata Interchange

CWM : Common Warehouse Metamodel

UML is:

Notation

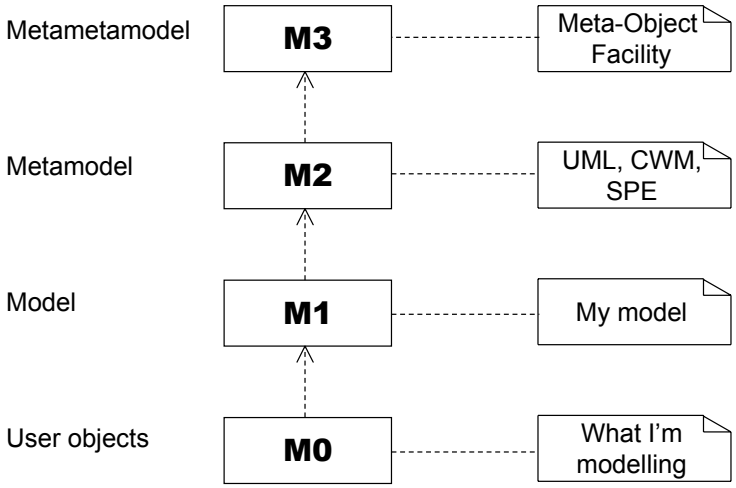
Abstract syntax (metamodel defined using MOF)

Well-formedness rules (Object Constraint Language)

Semantics (natural language)

IDL interface

UML is positioned in the OMG's "4-layer architecture"



MOF is:

A standard language for describing metadata

MOF metametamodel (M3) defined in itself

MOF reflective IDL interfaces for generic manipulation of metadata

MOF to IDL mappings for type-safe manipulation of metamodel specific information

MOF to XMI mapping: OMG XMI (XML Metadata Interchange) specification

MOF to Java mapping: Sun JSR-40, JMI (Java Metadata Interchange)

XMI (XML Metadata Interchange) is:

The standard format for interchanging MOF metamodels and their instances

It uses XML for the transfer syntax and interchange format

Specify XML Document Type Definitions (DTD) to enable transfer and verification of

- UML based models (eg. mymodel.xml, using uml.dtd)
- MOF based metamodels (eg. uml.xml, using mof.dtd)
- Models based on other MOF-based metamodels (e.g. mymodel.xml using cwm.dtd)

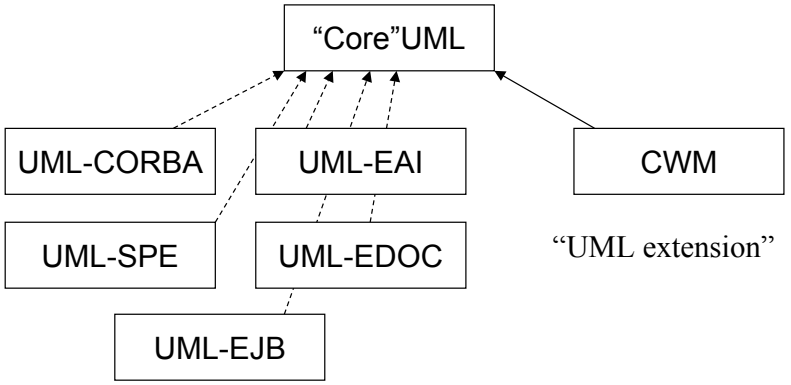
XML schema version is in the works

CWM (Common Warehouse Metamodel) is:

A standard model for data warehouse metadata management

Defined using MOF, interchanged using XMI, and reusing aspects of the UML metamodel

UML is in fact a family of languages, all built using MOF



The Request for Proposals for UML version 2 has been issued and work is in progress

UML 2 Infrastructure

UML 2 Superstructure

UML 2 Object Constraint Language

UML 2 Diagram Interchange

UML 2 Infrastructure calls for:

Architectural alignment and restructuring

- strict alignment with 4-layer model
- make MOF abstract syntax a subset of UML abstract syntax
- restructure the metamodel in order to separate concerns
- identify "semantic variation points"
- backwards compatible with XMI 1.x

Extensibility

- specify profiles
- specify "first class extensions"

So what is missing to achieve the vision of large-scale model-driven development?

Standard modelling languages that cover the entire space of development

Especially architecture description & process description

Precise definition of modelling languages

Theory, practice and standards for composition, refinement and transformation of models

Tools that support modelling languages properly

Integration of tools

including composition, refinement and transformation

Process standardisation

IBM funded a feasibility study by pUML (precise UML group)

See www.cs.york.ac.uk/puml/ for the document "A Feasibility Study in Rearchitecting UML as a Family of Languages using a Precise OO Meta-Modeling Approach", (Clark, Evans, Kent, Brodsky, Cook) and associated tools

The study proposed a new meta-modelling facility (MMF) containing:

Meta-Modelling Language (MML)

Meta-Modelling Tools (MMT): a satisfaction checker - does instance X satisfy constraint C from model M?

- *check that a model satisfies its metamodel*
- *check that a metamodel satisfies the MML rules*
- *check that MML satisfies the MML rules*

Useful links

OMG - www.omg.org

UML forum - www.celigent.com/uml/

pUML group - www.cs.york.ac.uk/puml/