MDA and Stakeholders in an MDA Process

…and the support for their roles

Model Driven Architecture (MDA)

- An approach:
  - to system development, which increases the rigour of models.
  - It is *model-driven* because it provides a means for using models to direct the course of understanding, design, construction, deployment, operation, maintenance and modification.
- On architecture …
  - The *architecture* of a system is a specification of the parts and connectors of the system and the rules for the interactions of the parts using the connectors.
  - The Model-Driven Architecture prescribes certain kinds of models to be used, how those models may be prepared and the relationships of the different kinds of models.

- Thanks to Mathenge for this overview ☺

SoSyM

- Computation Independent Model (CIM)
  - e.g., business process models

- Platform Independent Model (PIM)
  - e.g., system structure models (class models mainly)

- Platform Specific Model (PSM).
  - e.g., class models tailored to a specific platform

Computation Independent Model (CIM)

SoSyM

- Focuses on the on the environment and the requirements for the system;

- Does not show details of the structure of systems.

- Often called a domain model

- In theory: domain experts can identify (and describe) business concerns in a CIM.

- Focus on the operation, ignoring details necessary for a particular platform.

- Shows that part of the complete specification that does not change from one platform to another.

- May use a general purpose modelling language, or a language specific to the area (application domain) in which the system will be used.

- *In reality* platform independence is relative.

Platform Specific Model (PSM)

- View of a system from the platform specific viewpoint.

- Combines the specifications in the PIM with the details that specify how that system uses a particular type of platform.

- In some senses a more standard view.

SoSyM

Process Overview

- MDA provides mechanisms for formalising the abstraction level of models in software development and for managing transformations among them.

- That is, moving from the Computationally Independent Model (CIM) through Platform Independent Models (PIM) to Platform Specific Models (PSM).

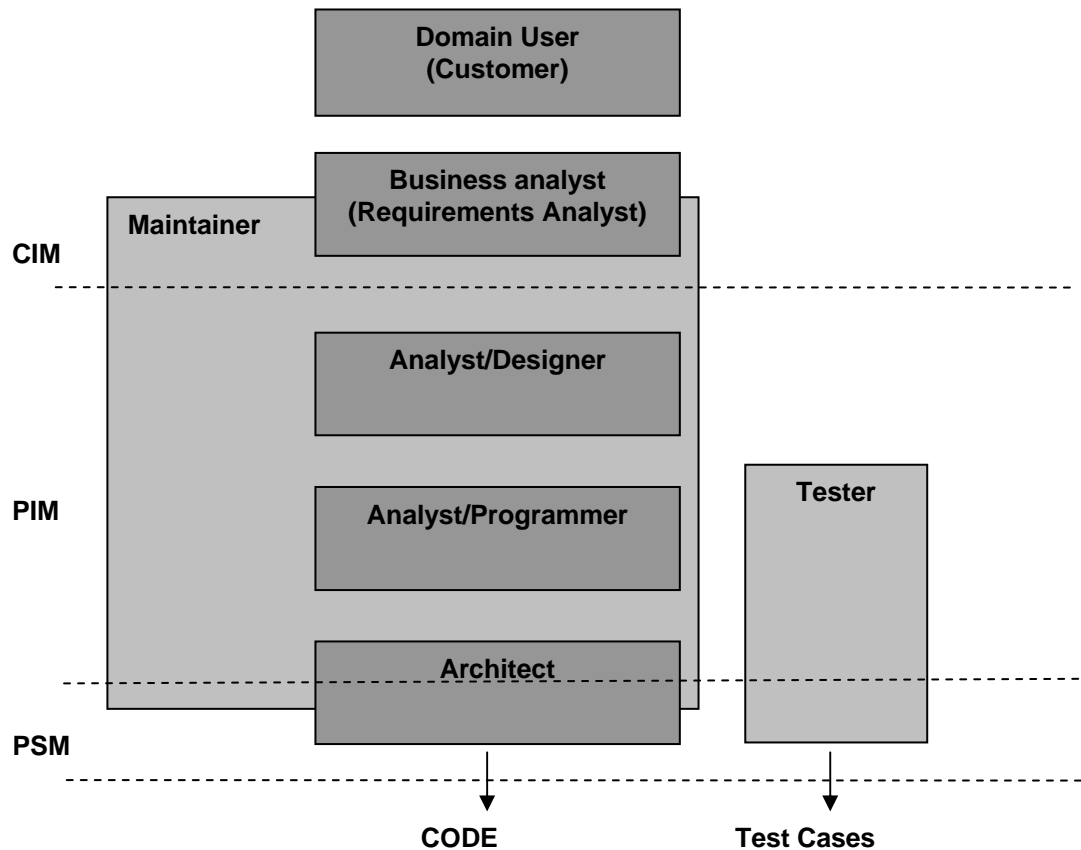*SoSyM*

Intended uses for Levels

- CIM: Business goals, domain understanding, business process & requirements.

- PIM: System Structure (though independent of platform).

- PSM: Derived (via transformations) from PIM

- Code generated (derived) from PSM.

- What about CIM to PIM?

- MDA initiatives tend to focus on modelling notations which are clearly within the domains of those already versed in software development.
    - For example, class diagrams still feature heavily.
- Similarly, tools for MDA tend to support traditional software engineering notations, largely those within the UML, and transformations, downstream, from them.

Stakeholders

- Requirements as "the effects that stakeholders wish to be brought about in the problem domain"
  - Bray, I. *An Introduction to Requirements Engineering.*

- Stakeholders can be identified as anyone that could be materially affected by the implementation of a new system or outcome.
  - Leffingwell, D. and Widrig, D. *Managing Software Requirements: A Use Case Approach.* Boston, US, Addison-Wesley, 2003.

SoSyM

Domain User
(Customer)

Business analyst
(Requirements Analyst)

Maintainer

CIM

Analyst/Designer

PIM

Analyst/Programmer

Tester

Architect

PSM

CODE

Test Cases

Domain User

- The domain user is the end user of the constructed software solution.
- Works for the customer and is an expert in his special domain.
- For example, an insurance salesman knows about his company's offers and legal regulations and is supported by software solutions without the knowledge about technical realisation.
- The domain user normally has no knowledge about business modelling but he can draft the requirements for a software application. In combination with a business consultant the CIM level models can be constructed.
- The language and the graphical representation should be easy to understand so domain users can validate the correctness of the models.
- Since domain users usually use specific vocabulary all tools should supports translations into the domain specific language. The domain user serves as a software tester for acceptance tests, i.e., reviews whether a simulated model performs the expected tasks.
- A domain user has special skills in his fields of work. Often he knows about business economics and enterprise management but has normally only office application skills. Experience in modelling of business processes can not be assumed.

Business Analyst

- Business Analysts advise enterprises on the analysis, conception and implementation of IT solutions.
- They constitute the connection between the customer and the involved IT specialists and need technical as well as social competences.
- A business analyst is one of the main user types envisaged for MDA toolsets. They accomplish interviews with domain users and analyse and model the proposed solution on the CIM level. Since they have knowledge in modelling of business processes as well as technical architectures it should be easy for them to use any supporting MDA tools.
- Business analysts should have a variety of different skills to fulfil their diverse kind of tasks. They should have knowledge about business processes, modelling and controlling, IT concepts and technologies, procedure models, project management and business economics.
- Beyond these technical skills lie social competencies like leadership, team organisation, partner management or knowledge in legal regulations are required. Typical tools used by Business Analyst are Business Rule Management systems and Business Process modelling tools.

Analyst / Designer

- Analysts/Designers are responsible for the conceptual model of business entities and the high level business logic.
- They use design artefacts and models produced by the business analyst and transform them into a design.
- The software designer is also responsible to decide if predefined components may be reused / composed or if they need to be re-implemented.
- The roles of the software designers and developer are often combined especially in smaller development projects and organisations.
- The software designer is a PIM level expert with strong background in conceptual modelling and UML class diagrams.
- The software designer defines the first level of behaviour, but leaves the details to the developer.
- For reusing or composing new applications from pre-existing components the designer Understanding of CIM level artefacts, i.e., the business process model is also required.
- Typical tools for a software designer are graphical modelling tools.

Analyst / Programmer

- The Analyst Programmer is responsible for completing the behavioural modelling that will allow model simulation (i.e., for testing) and the transformation of the models into code.
- The Analyst Programmer is a PIM level expert with a strong background behavioural modelling.
- The Analyst is one of the main users of tools for detailed behavioural modelling.
- The Analyst uses the format that is most appropriate for that task.
- Analysts will also implement components defined by software designers.
- Therefore, they will model the behaviour/business logic of the interfaces that have been designed and also provide the documentations for the components.

Architect

- The architect is an expert in the target platform (for example, Struts) and the programming language (for example, Java) but also has a good understanding of UML.

- An architect is working in application or systems development. The architect is the expert for the PSM level.

- The architect should have knowledge of different target platforms and programming languages.

- Experience in technical system specification and implementation of the proposed solution is mandatory as well as knowledge about programming concepts like software testing methods for quality assurance.

Major tools for MDA

Short word	Full Name	Company
OJ	OptimalJ	Compuware
AS	Arcstyler	Interactive Objects
CT	Constructor	Dot Net Builders
CA	Codagen Architect	Codagen
OG	Objecteering	Objecteering Software [SOFTEAM]
AM	Ameos	Aonix
TA	Together Architect	Borland [Inprise]
XM	XMF Mosaic	Open source
JD	Jamda	Open source
PT	PathMate	IBM
NB	NetBeans	NetBeans & Sun Microsystems
RX	Rational XDE Developer	IBM
EMF	Eclipse Modelling Framework	IBM
WS	Websphere	IBM
Aris	Aris Toolset	IDS Scheer

Feature/Support for	OJ	AS	CT	CA	OG	AM	TA	XM	JD	PT	NB	RX	EMF	WS	Aris
CIM					✓	✓	✓					✓	✓		✓
PIM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PSM	✓		✓	✓	✓			✓	✓		✓	✓	✓*		
UML 2.0	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MOF 2.0	✓	✓		✓	✓	✓		✓			✓	✓	✓	✓	
Action Semantics						✓	✓			✓			✓	✓	
UML profiles		✓		✓	✓	✓									
XMI	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓		
CWM															
QVT								✓							
OCL								✓					✓		
PIM→PSM→Code transform	✓		✓		✓				✓		✓	✓	✓*		
PIM←PSM←Code transform	✓		✓		✓				✓		✓				
PIM→Code transform	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	
PSM→PSM bridge					✓		✓								
Legacy code→PSM transform	✓	✓	✓		✓	✓	✓					✓			
Transform based on patterns	✓	✓			✓	✓	✓	✓	✓						
Traceability of transforms	✓	✓	✓	✓	✓	✓	✓			✓	✓		✓	✓	
Merging of models	✓				✓					✓			✓	✓	
More than one implementation platforms	✓	✓	✓	✓	✓	✓	✓	✓				✓			

Tool Support

- While all tools investigated supported PIM, and the majority PSM, only a minority support CIM, and this is often somewhat limited.

- Furthermore, whilst PIM to PSM transformations are common, moving from CIM to PIM is often far from trivial,
  - perhaps reflecting the hidden paradigm shift that this often requires.

SoSyM

Conclusions

- Whilst there is excellent support for PIM, and good support for PSM, support for CIM is somewhat lacking in the majority of tools.

- Hence, at least two classes of stakeholders, customers (and associated customer representatives) and business analysts are not catered for.

- In addition, whereas transformations from PIM to PSM are supported, moving from CIM to PIM (typically another analyst / designer task) relies heavily on analysts having understanding of both.

- Hence, whilst MDA tools would appear to offer strong support for software development activities from design onwards, the major issue with software development, that is of facilitating requirements, of moving from domain and business models, to software specification and design is largely overlooked.