



Information Society
Technologies

Project number 511731



MODELLing solution for softWARE systems



The MODELWARE Vision

Jan Aagedal, SINTEF

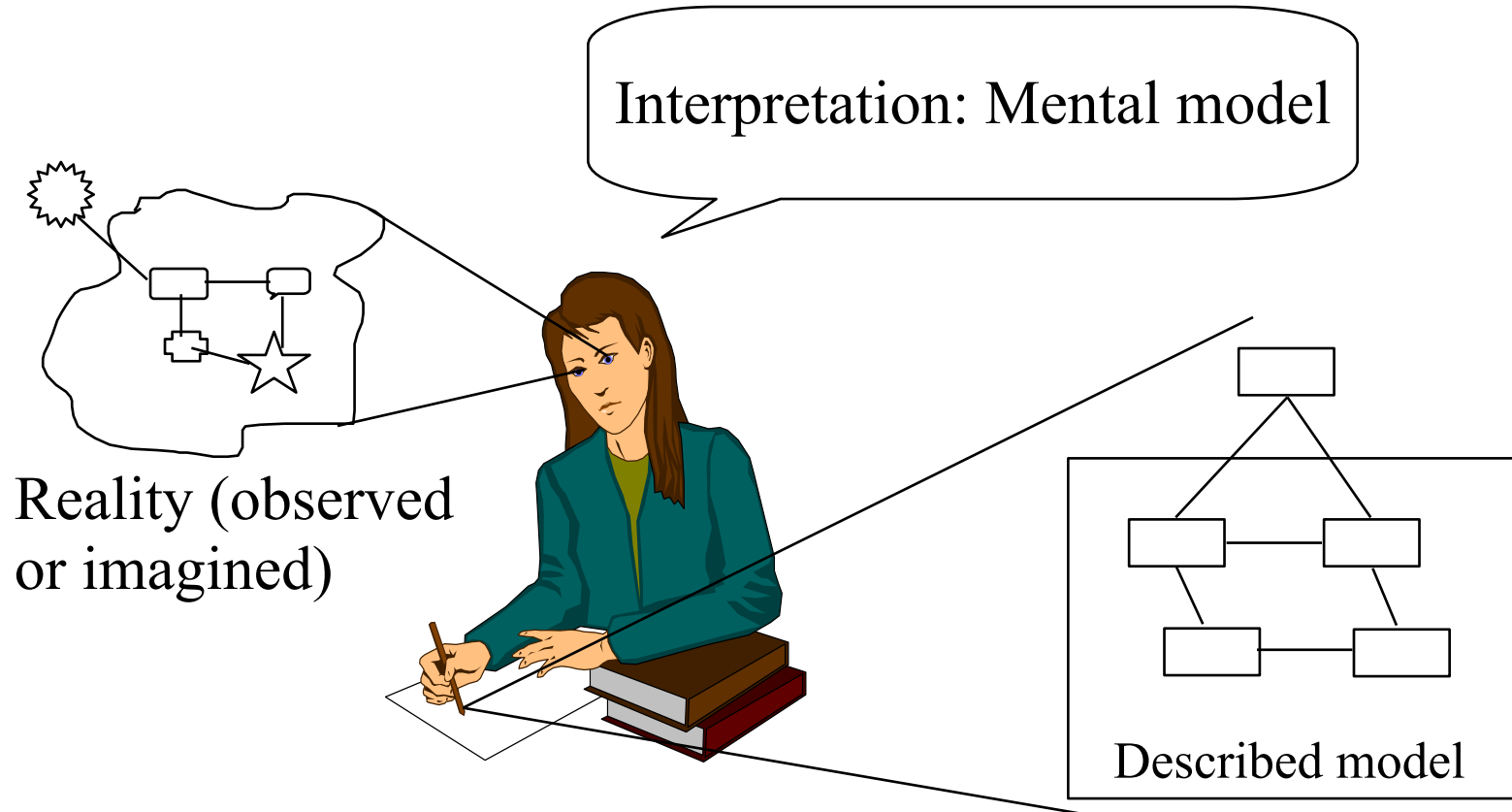
September 2006



- Purpose of presentation
- Why model-driven development
- Why MODELWARE
- Implementing the vision
 - Modelling technologies
 - Process and methodology
 - Industrial trials
 - Successful adoption
- Capabilities of the MODELWARE Open Platform
- MODELWARE solutions targeting different maturity levels



- Purpose of presentation
 - To summarise the vision for the MODELWARE project
 - Vision of model-driven development
 - Added value of MODELWARE
 - To describe how we reach towards the vision
- Audience
 - MODELWARE partners
 - European Commission
 - MDD end users
 - MDD tool vendors
 - Academics and researchers





Two main challenges for software development

- Master complexity
- Manage change

Models meet these by

- Acting as a means of communication between stakeholders
- Dealing with varying levels of detail
 - Abstraction to hide details
 - isolate changes at lower level of abstraction
 - Refinement to introduce details
- Being subject of analysis and other automatic tasks



Principles

Model-Driven Engineering (MDE)

Standards

MDA™
Model-Driven
Architecture
(OMG)

MIC
Model
Integrated
Computing

Software
Factories
(MS)

Other
Standards

Tools

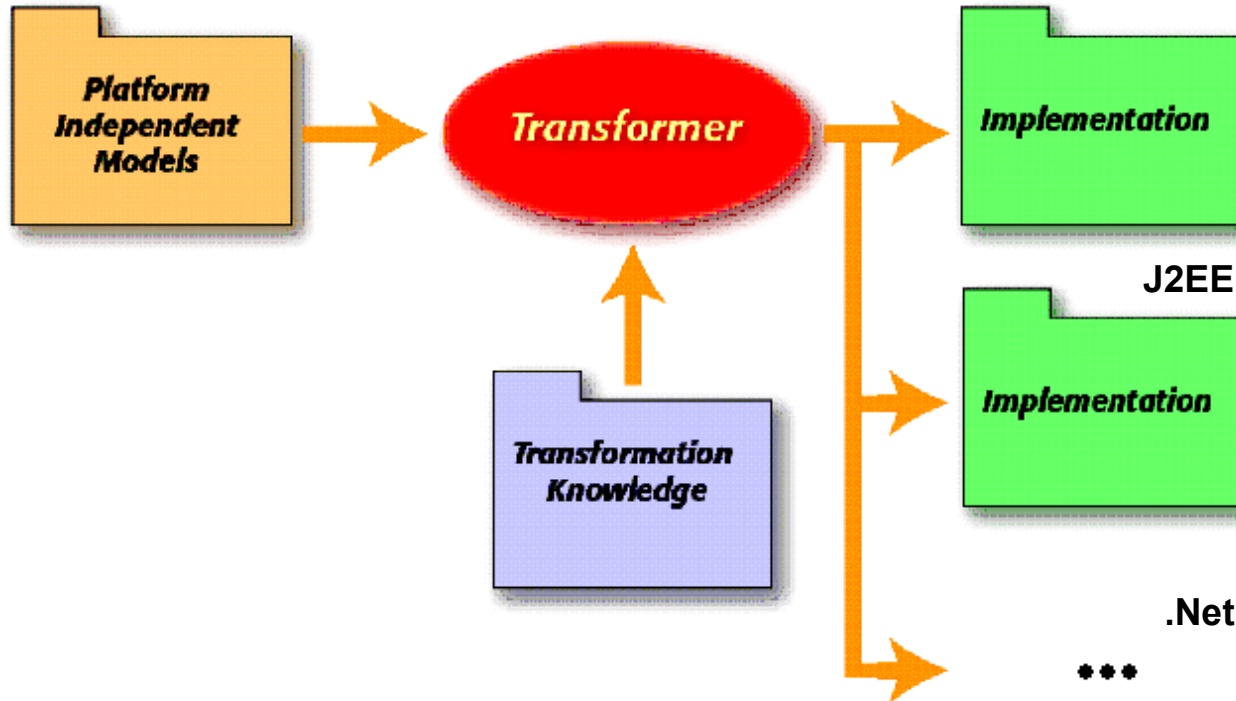
Eclipse
EMF
GMF

GME

Microsoft
Visual Studio
Team system
DSL Tools

Other
Tools





A PIM can be retargeted to different platforms

The concept Platform is relative

- A model can be classified as both PIM and PSM
- Tip: use concrete platform to qualify instead of "PIM" and "PSM"
 - "Component-architecture independent model" instead of PIM, "J2EE-specific model", ".Net-specific model" instead of PSM



Realise the MDD vision

- New development processes and better and more mature tools to support them
 - E.g., tools to allow more automation
- Know-how must be reusable across development projects
 - to allow capitalisation of existing models
- Tool chains must be open to allow interoperability and substitutability between tools
 - to avoid tool vendor lock-in
 - to support the link in of new types of tool ("specialist tools")
 - to allow capitalisation of existing models



MODELWARE aimed at making Model-Driven Development the leading approach in software systems development

MODELWARE had three major objectives:

- In two years, develop a solution that enables a 15-20% increase of productivity in software systems development
- To lead the industrialisation of the solution
- To ensure the successful adoption of the solution by the industry



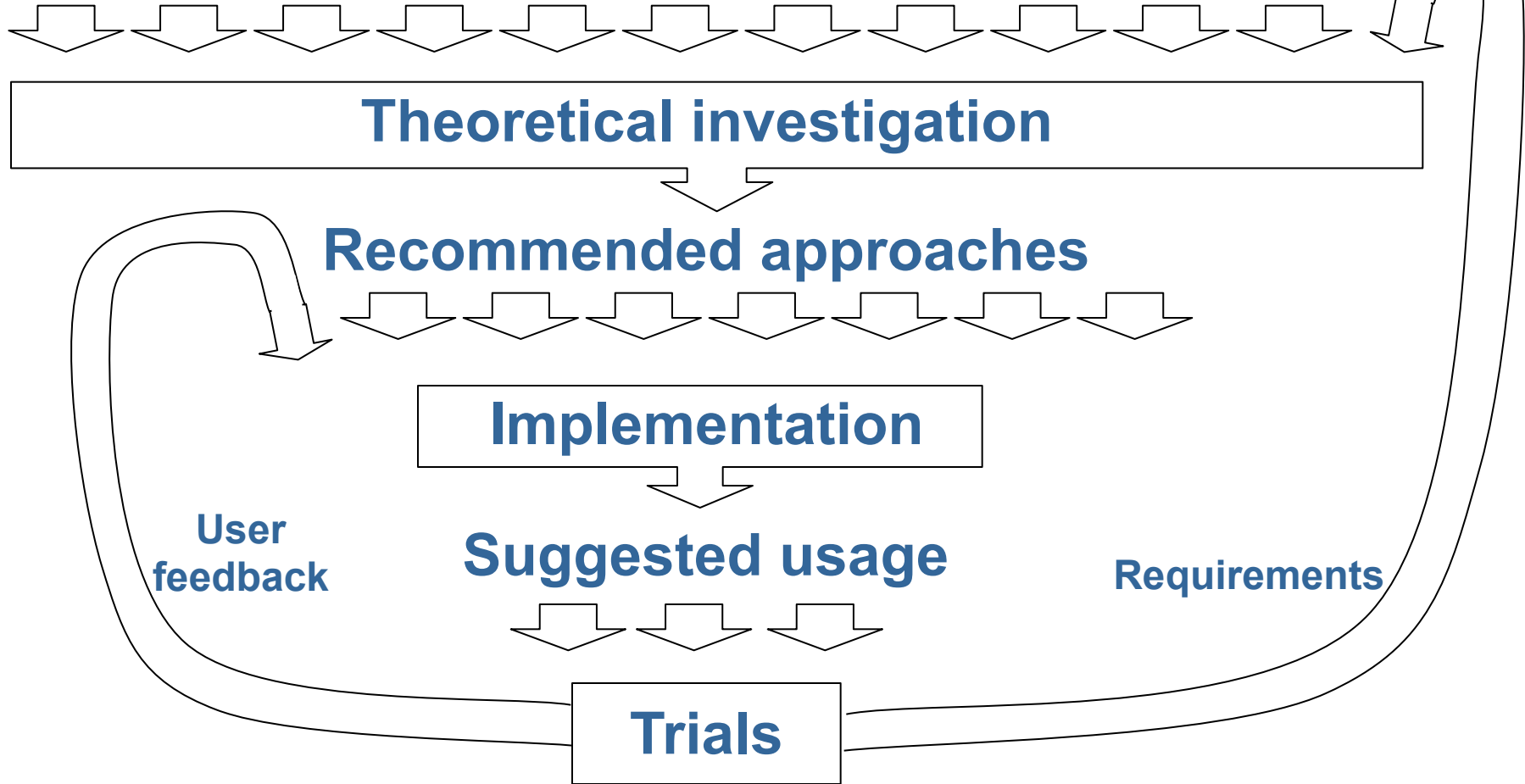


To improve productivity and quality in software development by adopting Model Driven Development

- **Better and more mature tool support for modelling**
 - Powerful, easy-to-use and reusable transformations
 - Model- to-model and model-to-text
 - Development and use of profiles/metamodels
 - Covering the software development lifecycle
- **Improved processes**
 - Process Framework
 - Definition of activities, work products, guidelines to allow MDD to be added to Software Development Processes
 - MDD Maturity Model and technology introduction guidance
- **The packaging of “know-how”**
 - The encapsulation of guidance, tools, process elements in MDA Tool Components
- **Tool substitutability**
 - Allow users to use the tools of their choice within a category
- **Tool interoperability**
 - Allow diverse modelling and non-modelling tools to work together



Potential topics



Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.

September 19, 2006





- **Adding rigor to models**
 - MOF Behaviour semantics
 - Model transformations (to model and to text)
 - Reusable transformations
 - Traceability
 - Architecture Driven Modernisation
- **Specific modelling techniques**
 - UML profiles: QoS, CCM
 - Architecture and platform modelling
 - Viewpoints and mapping to SysML and resource modelling
- **Design of generic model simulator**
 - To be used for any modelling language
 - Tool – generic core with specialization for Activity Diagrams and State Machines
- **Exploring ways to integrate models**
 - Consistency between models
 - Composition of models
 - Approaches and prototypes: Atlas Model Weaver, EML, Glue Generator Tool
- **Proposing a grouping mechanism for modelling technologies**
 - MDA Tool Component allows porting tool customisations

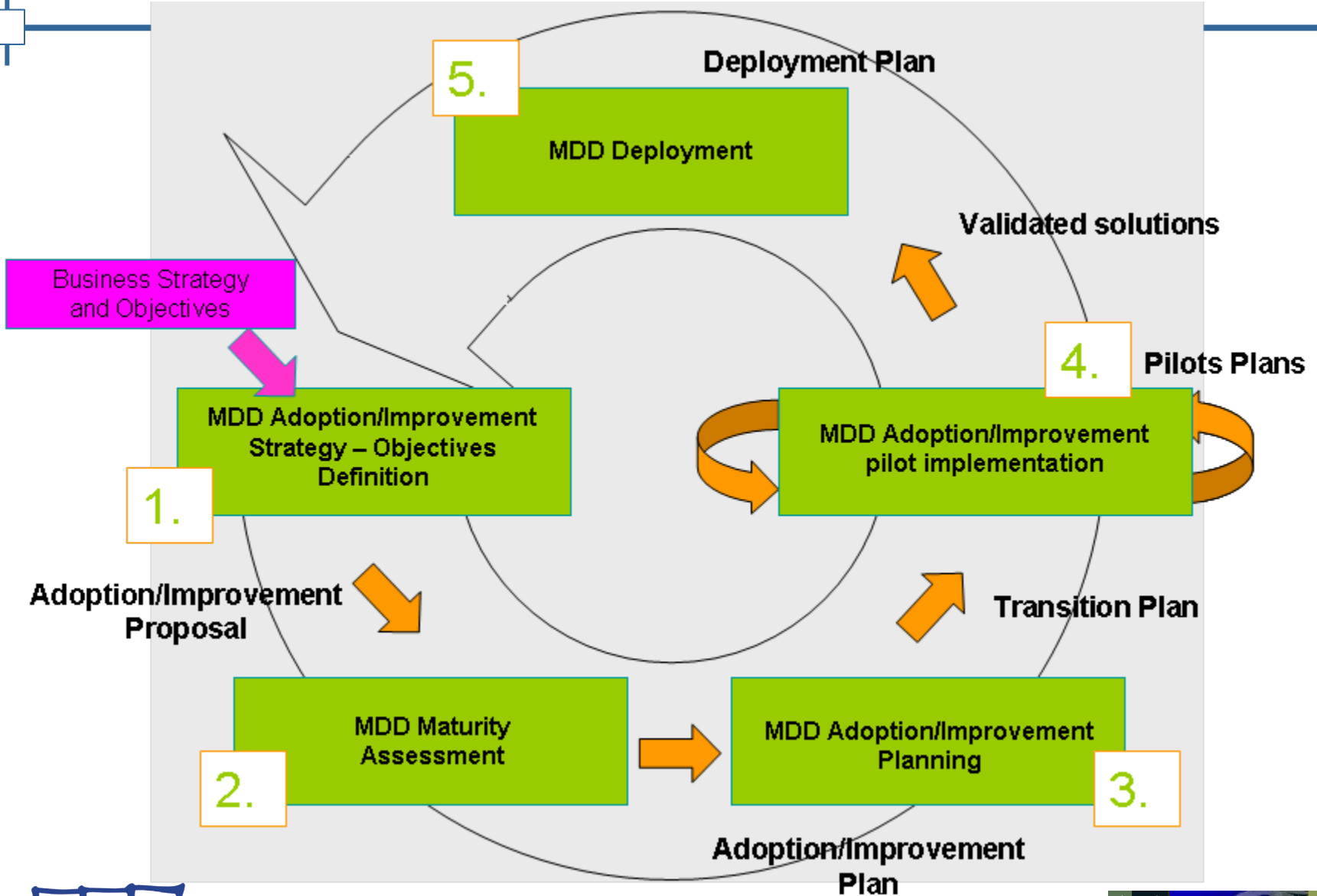




- Guidance for adopting Model Driven Development
 - Technology Change Management
- Determining current state of adoption and goals for enhancement
 - MDD Maturity Model
- Metrics for MDD
 - Business and Engineering Metrics
- Specific Process Elements for MDD
 - MDD Process Framework
 - Artefacts, guidelines, activities etc. to be added to existing methods



MDD Technology Change Management Process



September 19, 2006



Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.



Controlled steps to increase maturity while addressing business objectives and measuring these achievements objectively

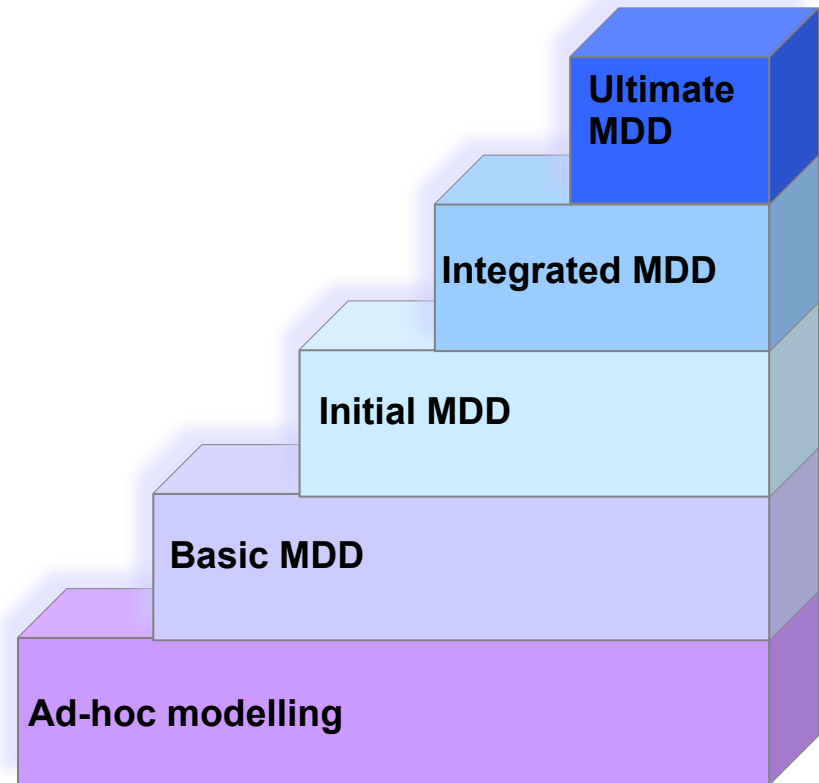
The whole life cycle is model-driven. Domain engineering practices are put in place. All the organisation's knowledge capitalised in models.

The domain concepts are represented by means of a Domain model, and Business models derived from them, introducing reuse. Definition of architecture-centric metamodel. Executable models are developed.

A separation between Business and Technical model is introduced. Implementation issues independent from the business ones. Sharing of models between different stakeholders. Model transformations used.

Basic usage of the models. Models are used for production of code and documentation. Adaptation of MDD tools to project objectives

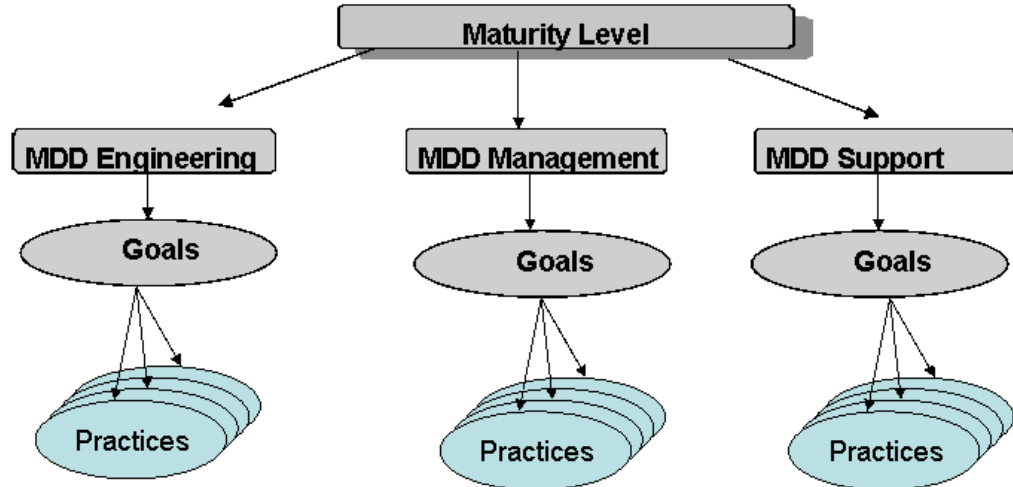
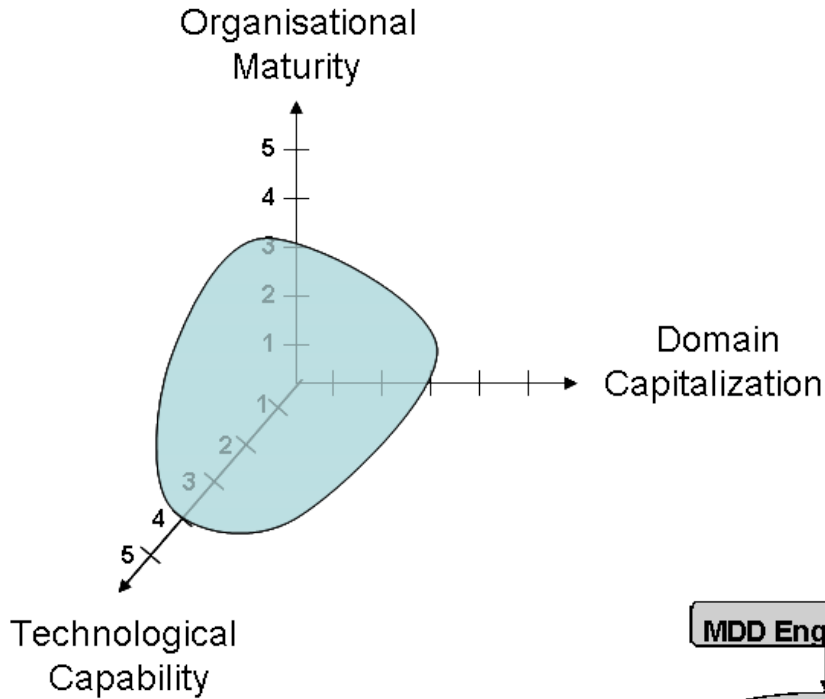
Modelling practices are sporadically used or not used at all



Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.

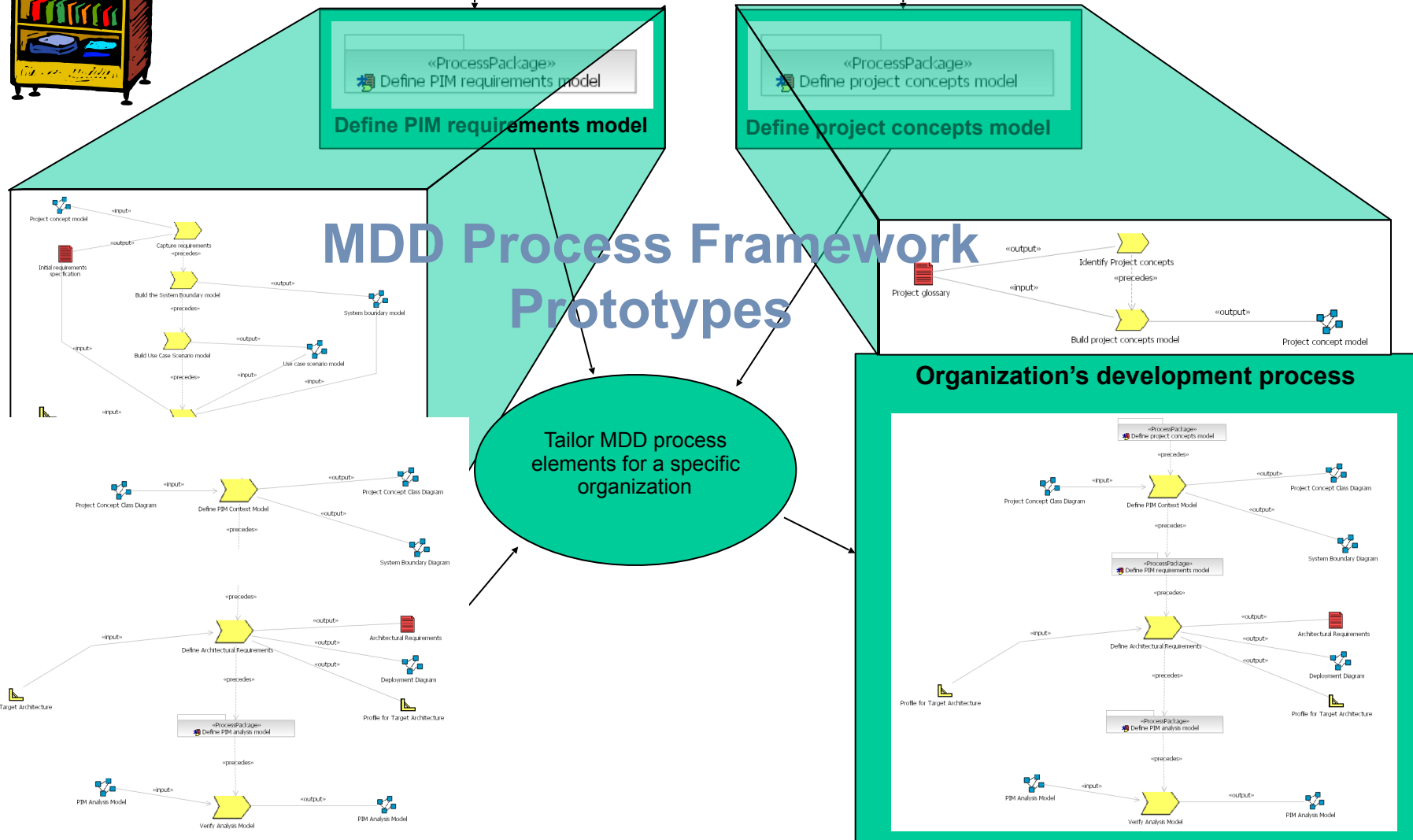
September 19, 2006







composed of MDD process elements

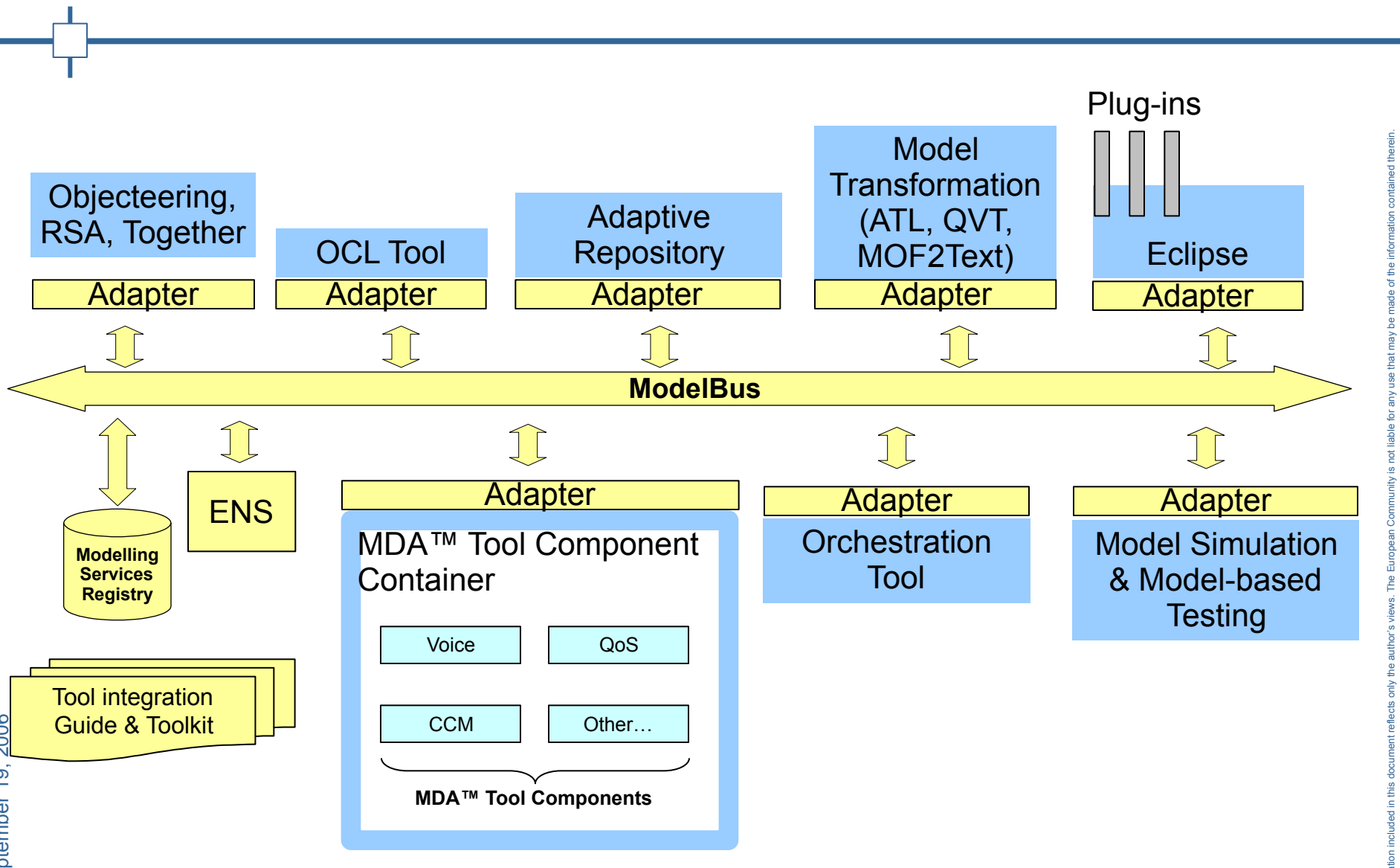


September 19, 2006

Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.

- Many tools involved in software development
- MDD should not require these all to be thrown away and replaced
 - But complemented with modelling capabilities
 - And allow reuse of existing tools and/or artefacts
 - And even for modelling capabilities there needs to be a choice of tooling
- Need to create and manage the models
- And use them to drive the development
- This all requires a tooling platform
 - both specification of its properties (to allow for choice)
 - and implementation (to give people something to use and to prove the vision)

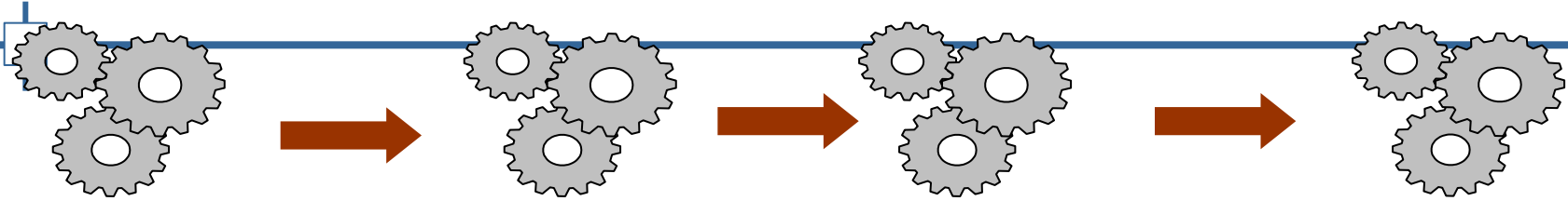




September 19, 2006

Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.

MDD tool chain -> Need for Tool integration



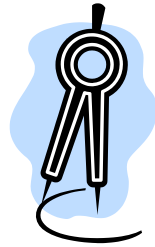
Requirements

- DOORS
- Mantis DB



Design

- Objecteering
- IBM RSA



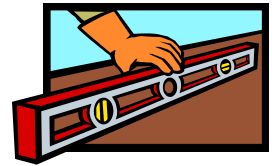
Implementation/ Code Generation

- MOF2Text
- UML2EJB



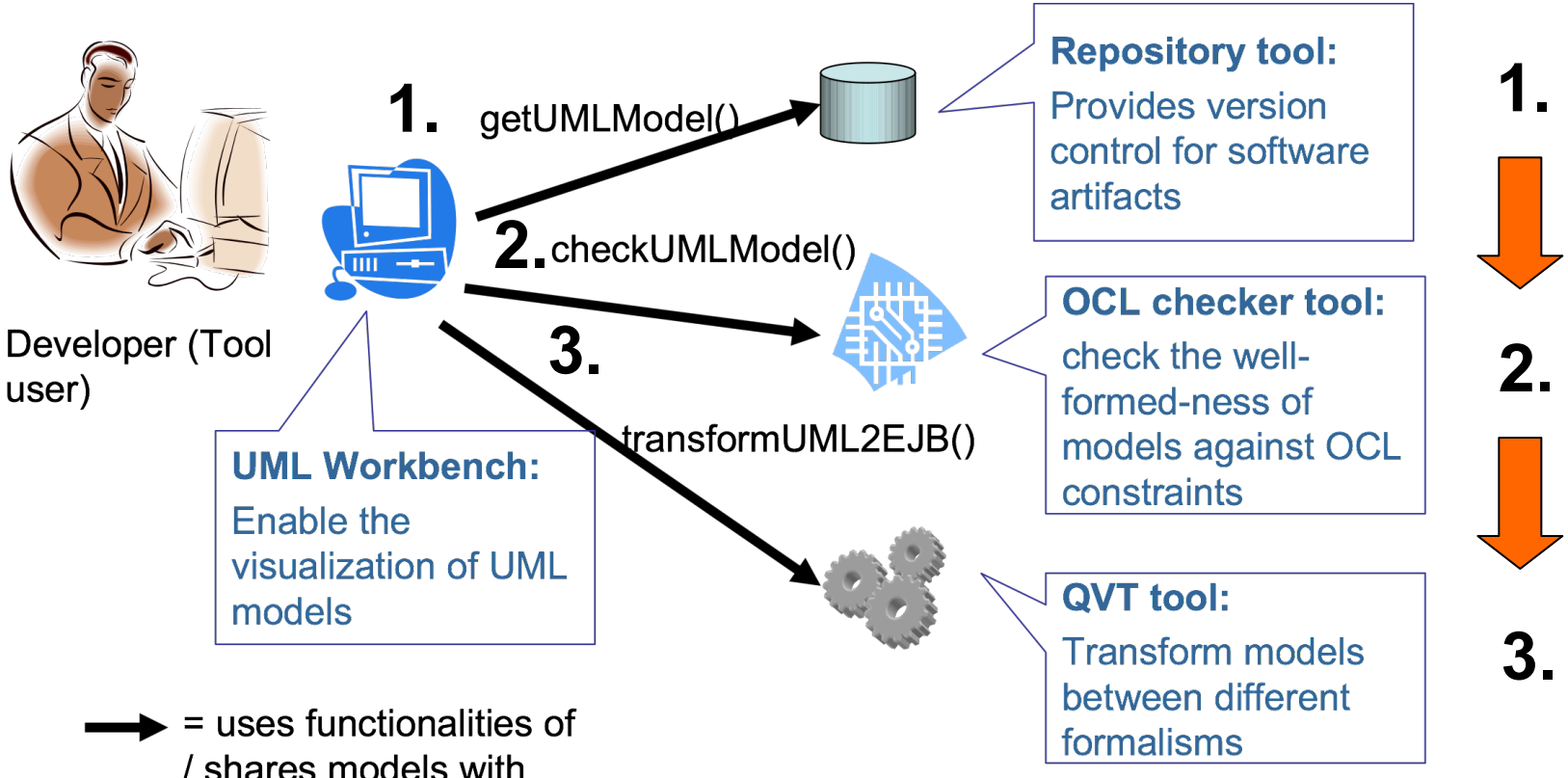
Validation

- Test Generators
- Test Execution



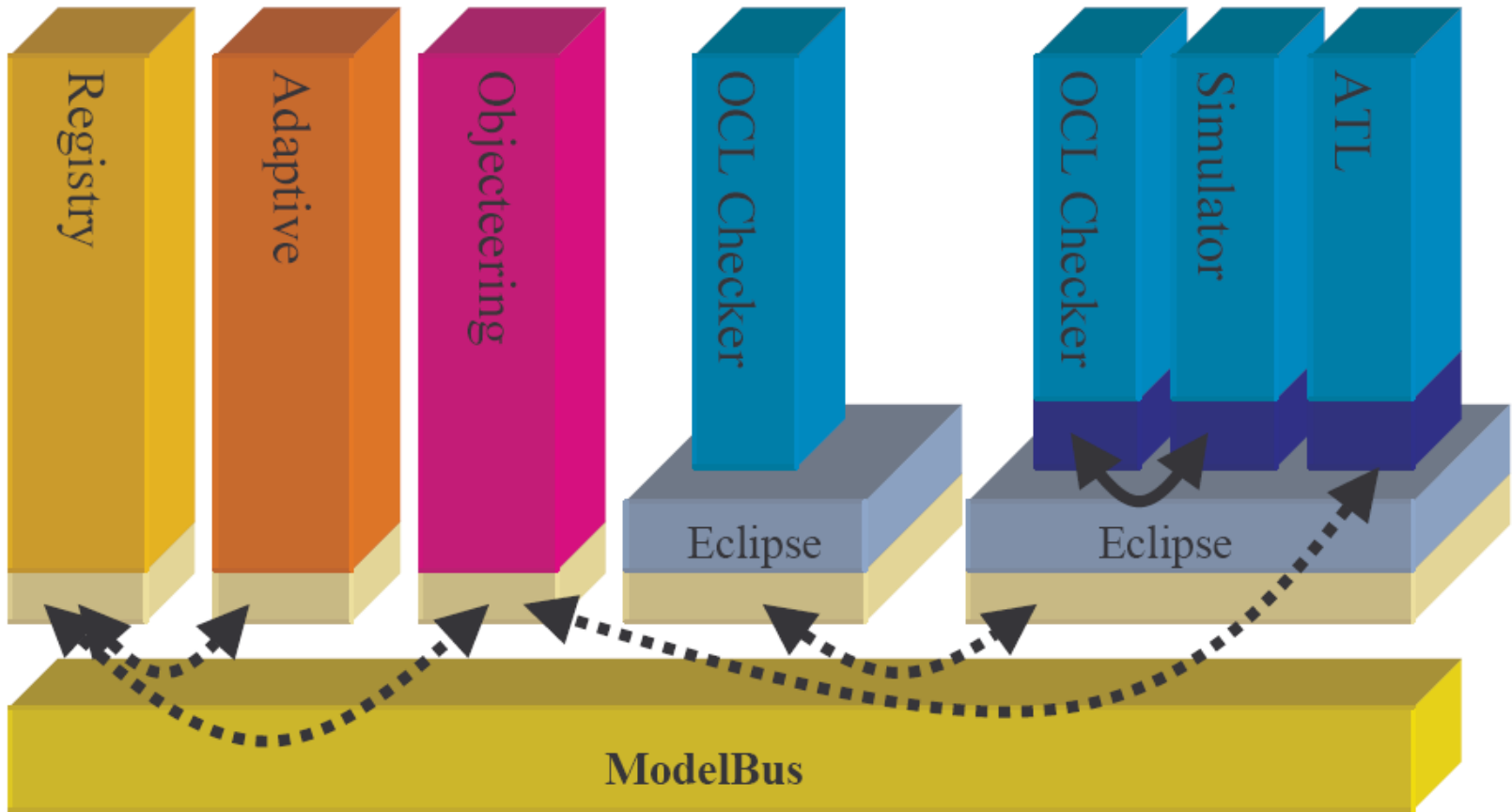
- No universal tool -> Need to plug additional functionalities
- Need to share models between several tools
- Tool substitutability required to avoid vendor lock-in
- Need to support distributed tool chains

Model Driven Development is “orchestration” of **modelling services**
Goal of ModelBus = **Infrastructure** for modelling service integration and interoperability

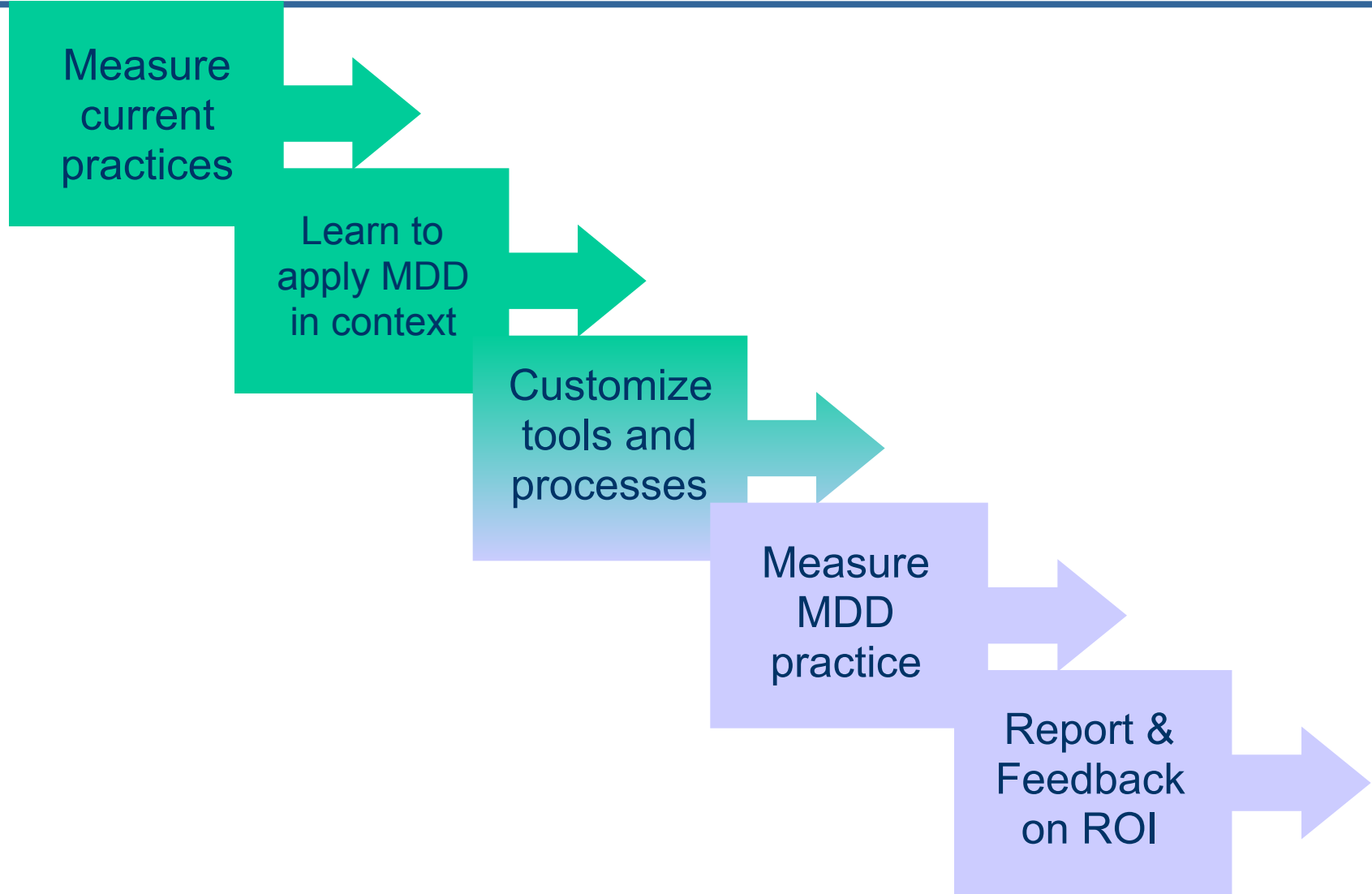


September 19, 2006

Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.



September 19, 2006





- Multiple field-evidence from MDD processes
- MDD returns the investment in modelling at the maintenance phase
- Indifferent at design/implementation phase
- Negative for system engineering at requirement phase
 - But very positive estimates
- Based on knowledge gained
 - Continue to develop and mature appropriate MDD tools
 - Iterate training and deployment
 - Examine the level of modelling in the organization





- Coordinate standardisation
- Federate open communities
 - To get external validation and dissemination
 - End users
 - Tool vendors
 - Open source technology
 - MDDi, GMT, ATL, MOFScript, QVT, EPF
- Disseminate
 - Newsletters, website, conference, scientific papers, white papers, workshops, press releases, events, ...
- Training
 - Public and proprietary
 - ECESIS , videos, classical PPT and exercises





- Multiple projects
 - Including multiple company/outsourcing
- Multiple locations
- Multiple tools
 - including for same task
 - may change over time
- Re-use of models
 - Different platforms
 - Different business applications
 - Different versions
- Process support
 - Blend of current and MDD-specific processes
 - Manage lifecycle of elements





- Non-disruptive
- Integrate and extend existing tools/IDEs
 - Work with but not require Eclipse
- Provide low-cost options
- Not require a lot of new skills
 - Package the technology
 - Embed the process in the technology (wizards)
- Three levels of packaging depending on the user organization:
 - MODELWARE in a Box
 - Intermediate
 - Advanced





*For organizations totally new to MDD and modelling
Just existing development tools*

- Minimal but sufficient set of tools
- Documentation, transformations, wizards etc
- Methods with tool plugins
- Customizable
- Choice of ‘flavors’ (e.g., Eclipse, Java, CCM)
- Upgrade path





For organizations with an environment already in place

Maybe some modeling tools

Patchy usage isolated to sparse projects

No real assets

--

- Library of adapters for tools to be retained (helped via MDDi project)
- Asset management and reuse
- Distributed working
- Automation
- Traceability
- Monitoring and management
- Plug and play services and tools





For organizations experienced/trained in MDD

Or an integrator/consultant with these skills

- For these users, everything is open to extension
- Define/extend:
 - Development processes
 - Transformations
 - Validations
 - Wizards (unless 'closed')
 - Tool pipelines
- Integrate new tools





- **MODELWARE has**
 - advanced state-of-the-art
 - made innovative tools
 - shown the usefulness
 - told the world

- **But significant challenges remain**
 - Model management
 - Model composition and refinement
 - Reverse engineering
 - ...



More information about MODELWARE?



www.modelware-ist.org

September 19, 2006



Information Society
Technologies

Project co-funded by the European Commission under the "Information Society Technologies" Sixth Framework Programme (2002-2006)



Information included in this document reflects only the author's views. The European Community is not liable for any use that may be made of the information contained therein.