

Model Driven Service Creation for eGovernment Services

Dr. Olaf Kath

kath@ikv.de

Dr. Marc Born

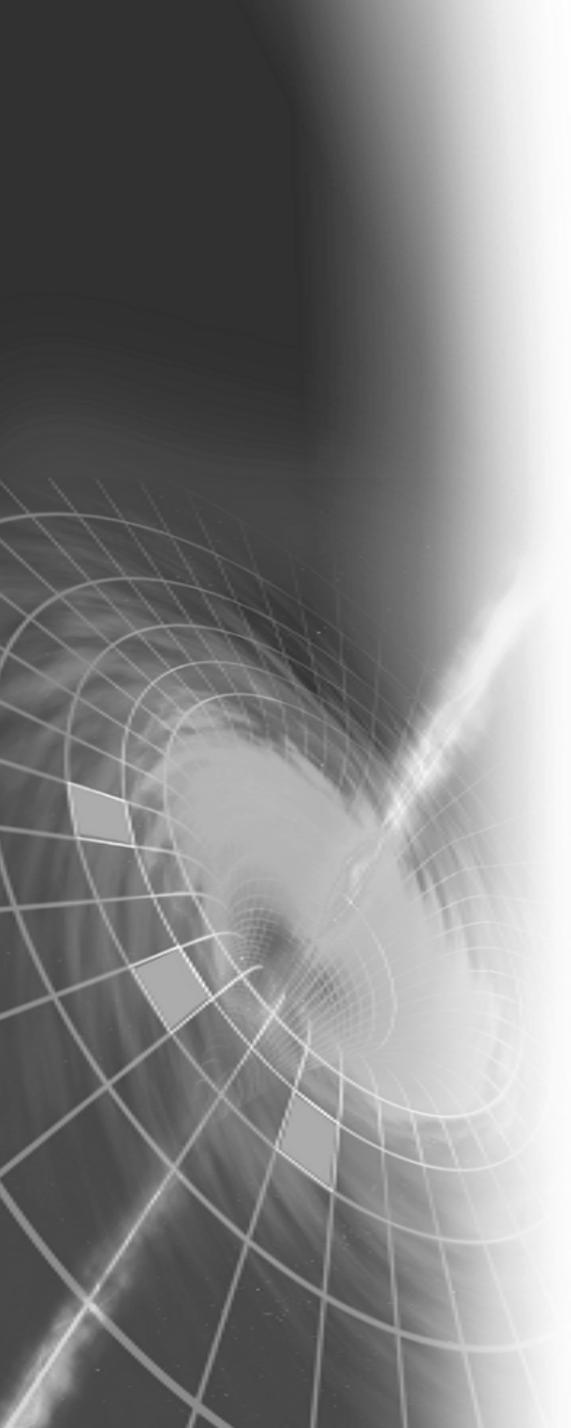
born@fokus.fraunhofer.de

Dr. Osamu Iwaki

iwakio@nttdata.co.jp

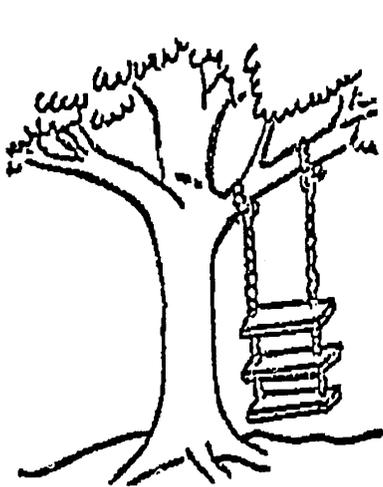
Shingo Kamiya

kamiyas@nttdata.co.jp

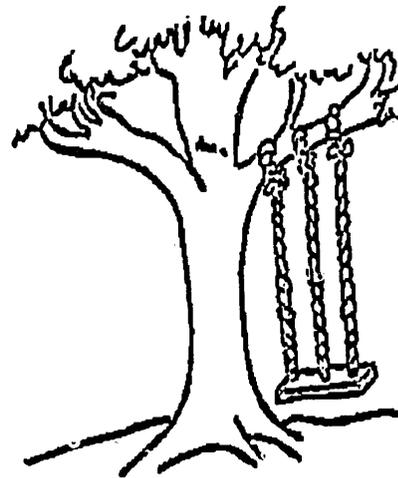


Why Model Driven Service Creation?

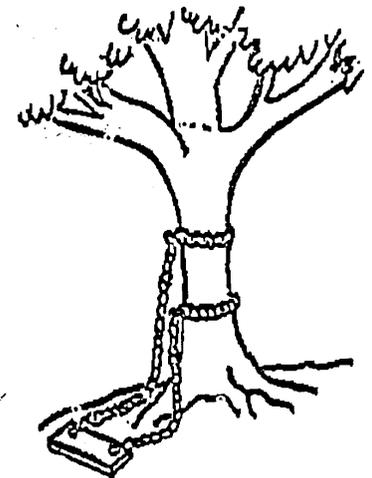
***Support for systems engineering
from requirements analysis down
to running solution***



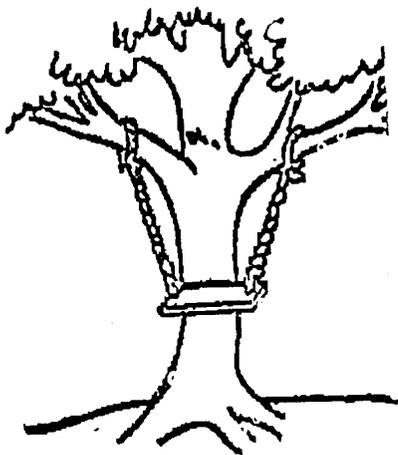
As Management requested it



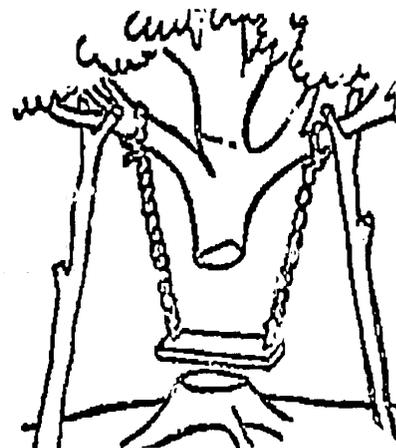
As the Project Leader defined it



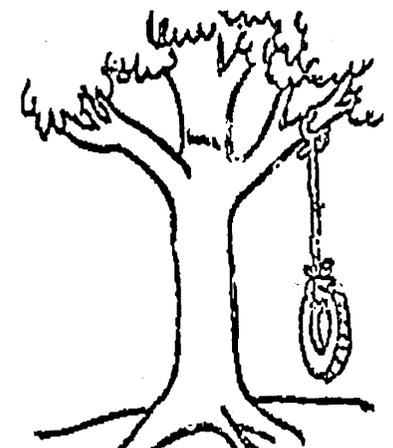
As Systems designed it



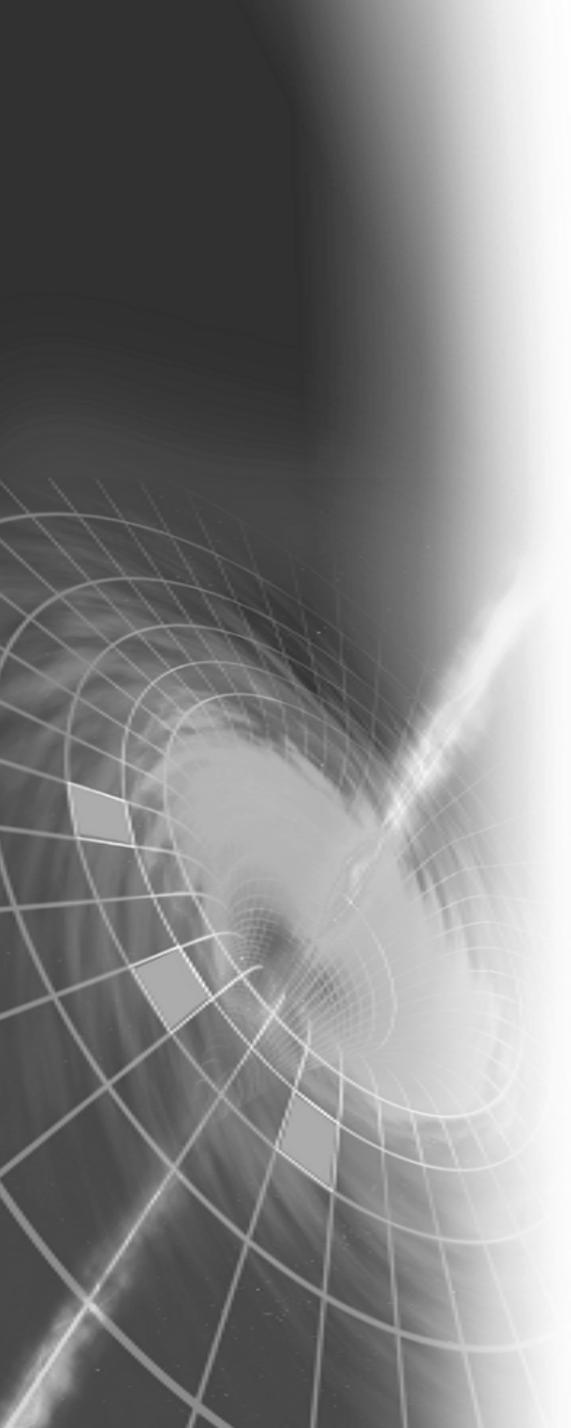
As Programming developed it



As Operations installed it



What the customer wanted



The vision of the MDSC project:

***System development tools
running in an environment that
consistently connects the output
of one with the input of another***

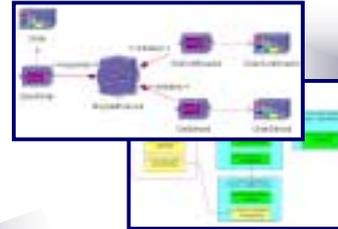


System Requirements

system development



technology independent model (abstract)



platform-independent modeling



technology-independent model (refined)



platform-specific modeling



technology specific model

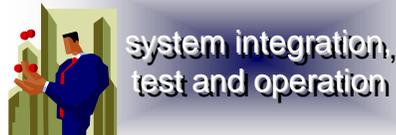


integrated-development environment

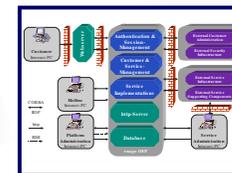


system-components

system operation

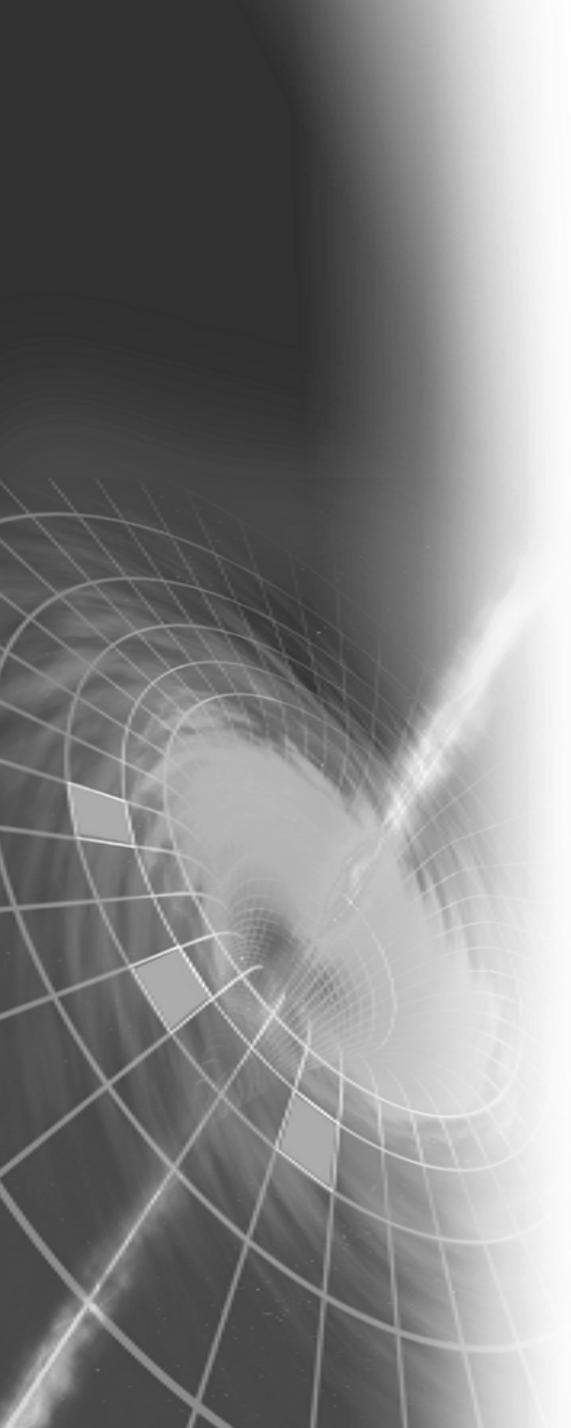


system integration, test and operation



Service platform for integration and operation

system operation

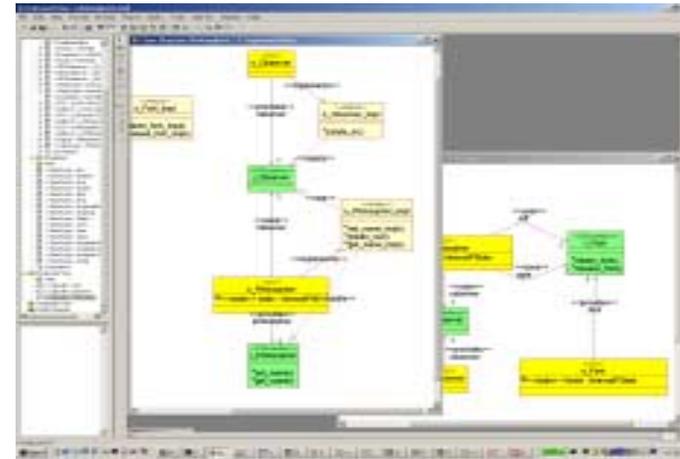


***Project history - Assessment and
Experimentation Results:***

3 Fallacies

Fallacy 1 - One Tool Fits All

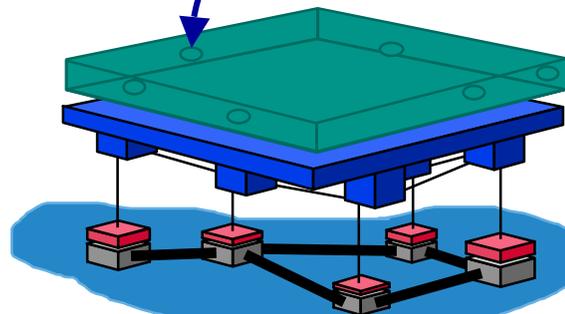
- System Construction is too complex
 - There's no ONE tool, that fits to ALL tasks in the process
 - The “We've integrated DOORS in our UML tool” isn't the solution either
- There must be **integration of different tools** for specific purposes



tool for modeling

one stop code generation

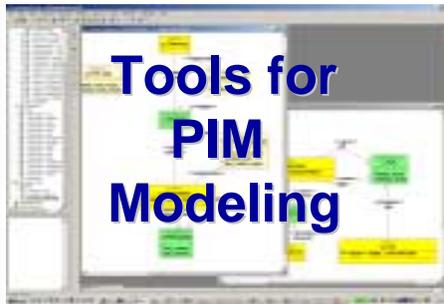
Platform Components



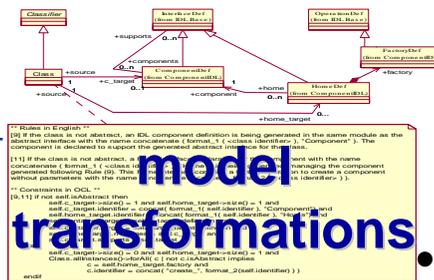
Fallacy 2 – UML is MDA is UML

- UML is a general purpose modeling language
- However, UML is not a modeling language for any purpose
- UML Profiles are not sufficient either
 - Define the semantics of business processes on top of the UML Metamodel? Why?
- There must be other techniques to
 - Define the semantics of elements to model e.g.
 - Business processes
 - Platform concepts
 - Define transformation rules between models
 - Define different notations for the same concepts

Fallacy 3 – XML is the Solution for Tool Integration

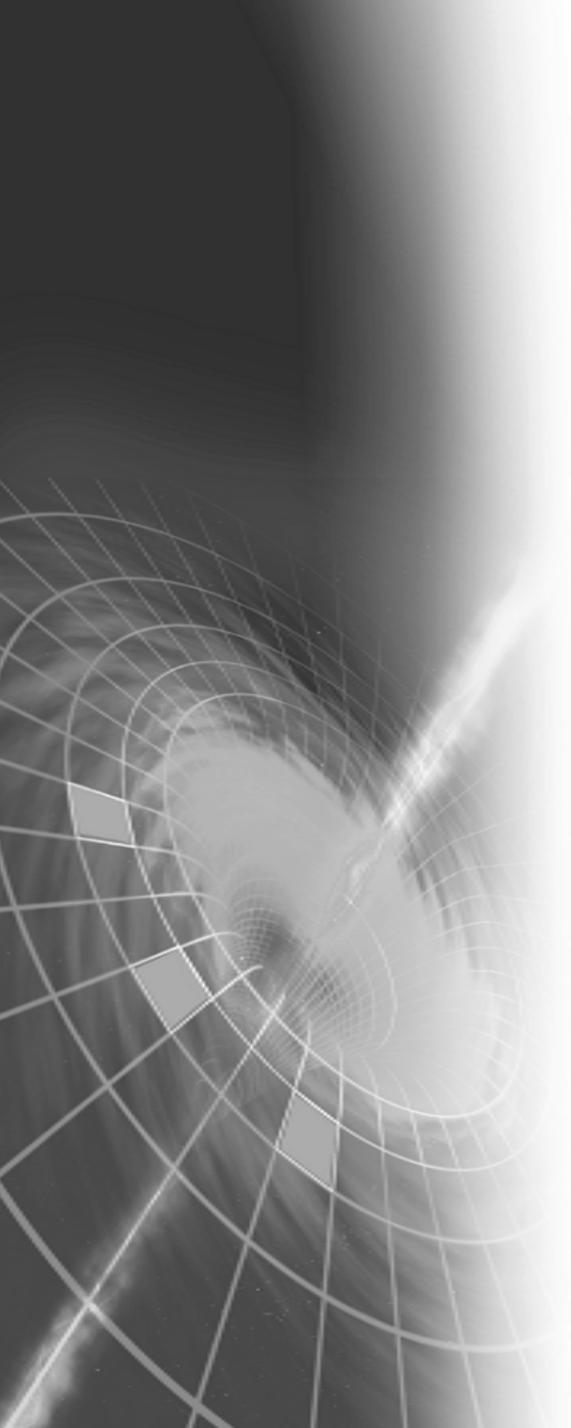


Tools for
PIM
Modeling



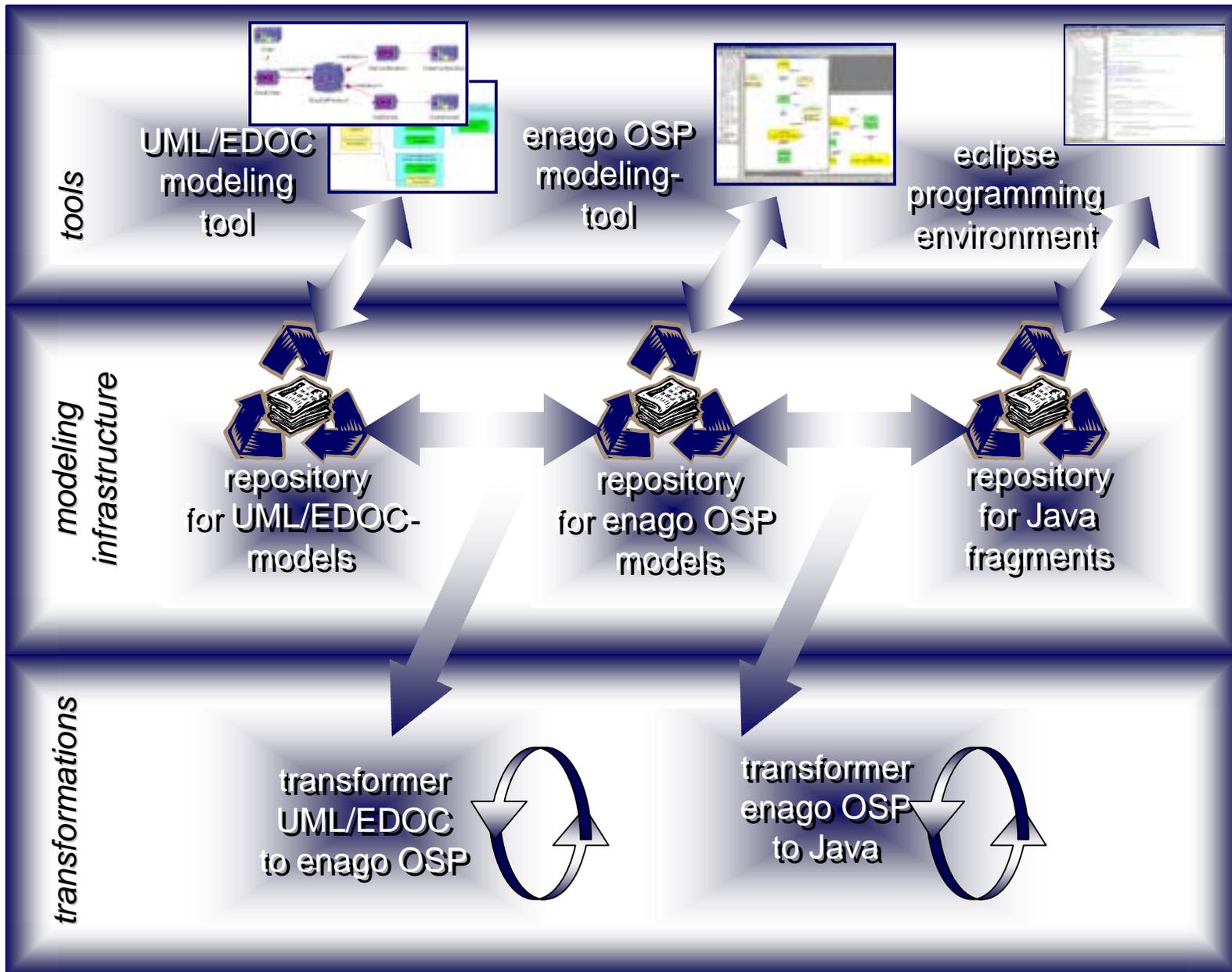
Tools for
PSM
Modeling

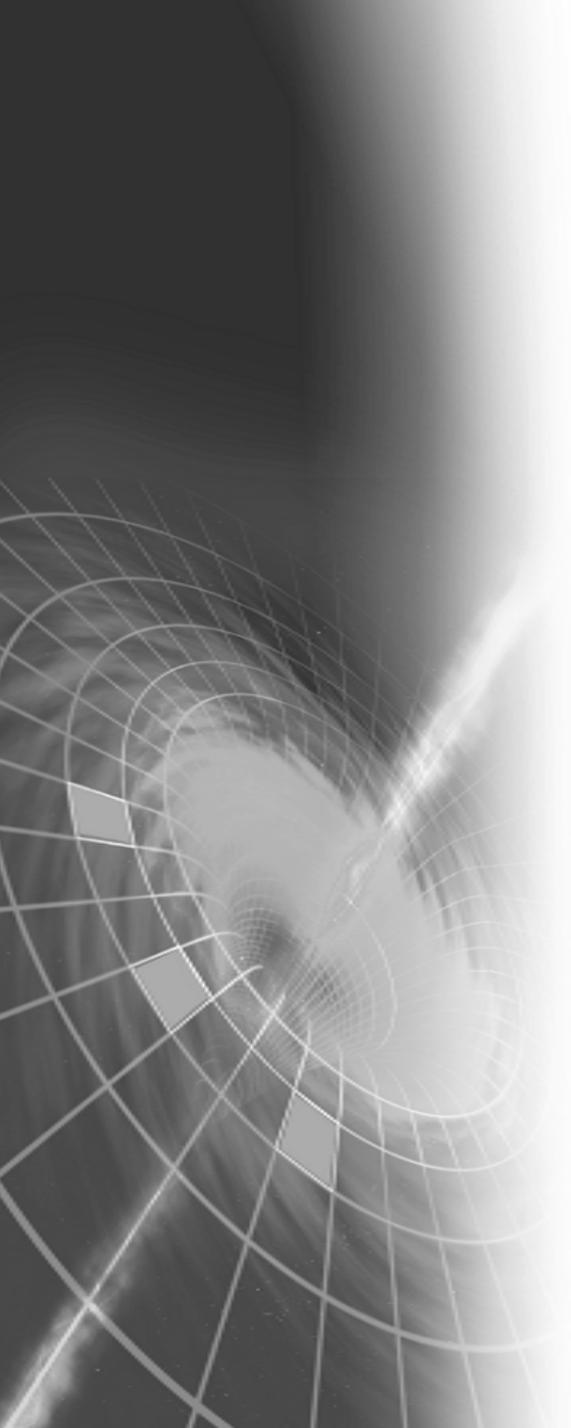
- XML is a technique to exchange any kind of models between tools
 - specializations for e.g. UML, MOF, CORBA, ...
- It can really only be used for this purpose!
 - The “we use XML and XSLT for transformations between models” won’t work
 - The “we store all models as XML” won’t work either
- There must be other techniques to manage e.g.
 - traceability
 - consistency
 - forward/backward transformations



The approach to the vision:

The MDSC Big Picture





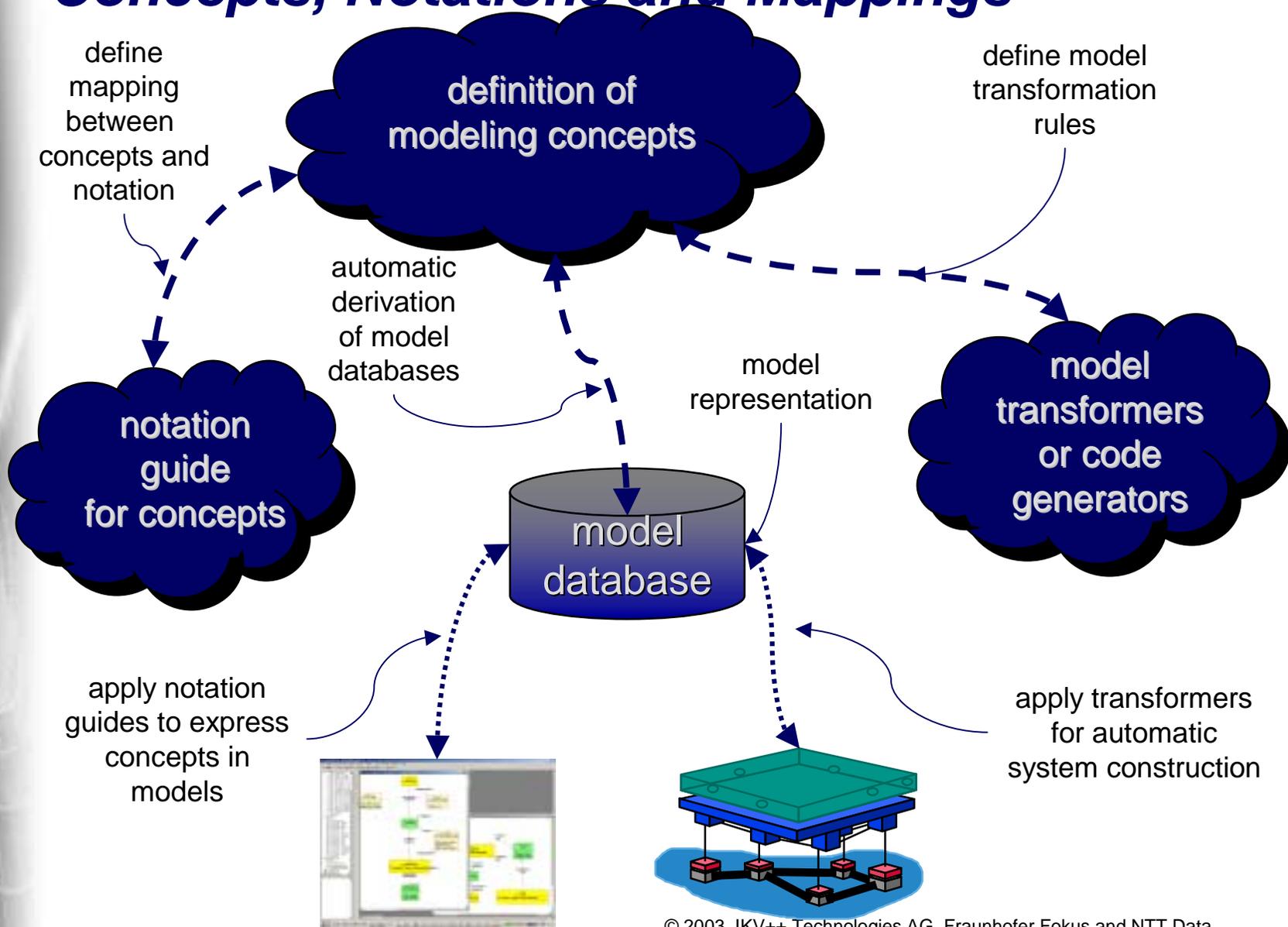
Some aspects behind the Big Picture:

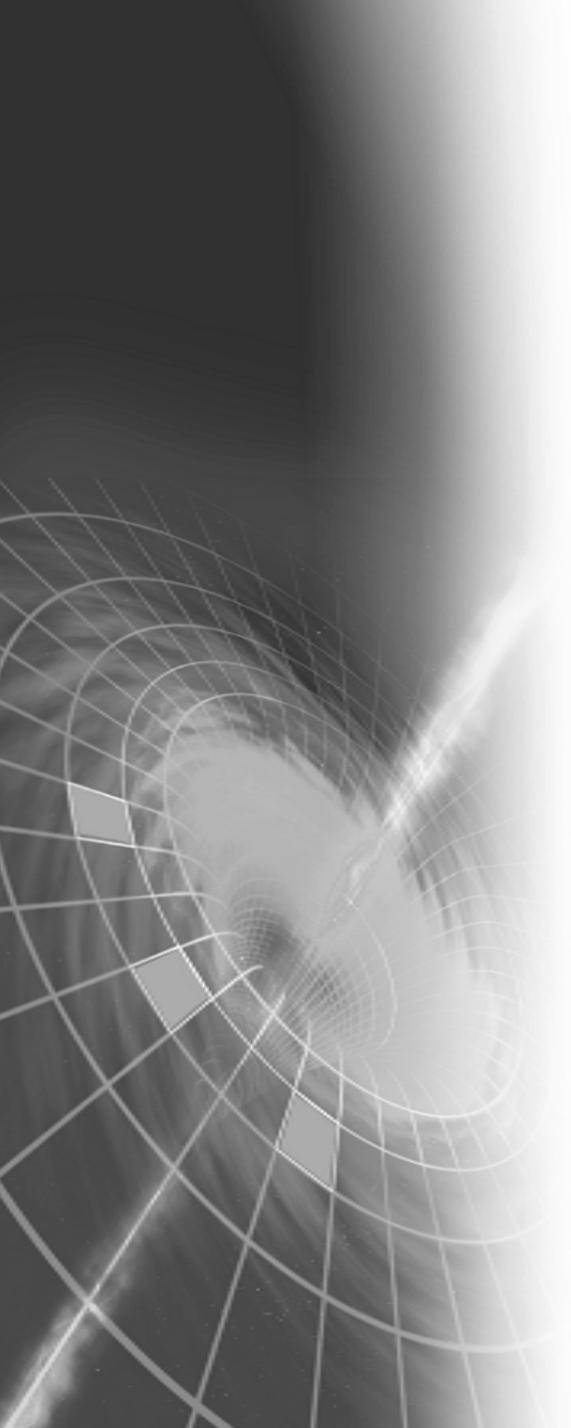
Architectural guidelines, a standards mess and technology pieces ...

Modeling Infrastructure

- All modeling instruments and principles are defined by models themselves, model databases are derived
 - We use the Meta Object Facility (MOF) for this purpose
- Based on these metamodels, domain specific visual modeling languages are defined
 - We use different techniques here, e.g. UML based languages, textual languages, new graphical languages
- Also based on the metamodels, further processes like code generation or model transformations are defined
 - We keep notation and these processes separate

Architecture Guidelines: Concepts, Notations and Mappings



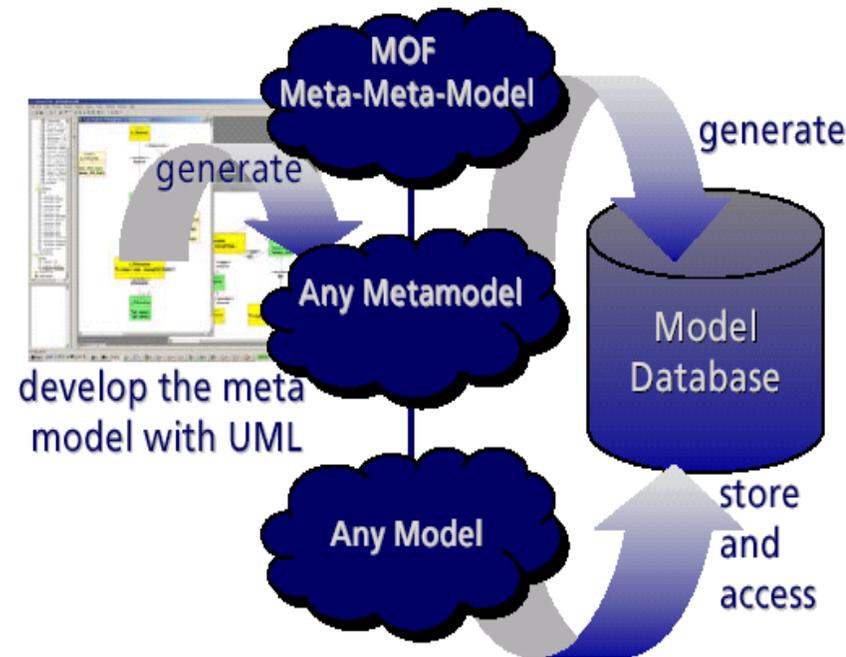


Project results:

- PIM Modeling Tool based on EDOC***
- enago OSP modeling tool***
- MDSC eclipse integration***
- modeling infrastructure for tool collaboration***
- model transformer generator***
- concrete model transformers***

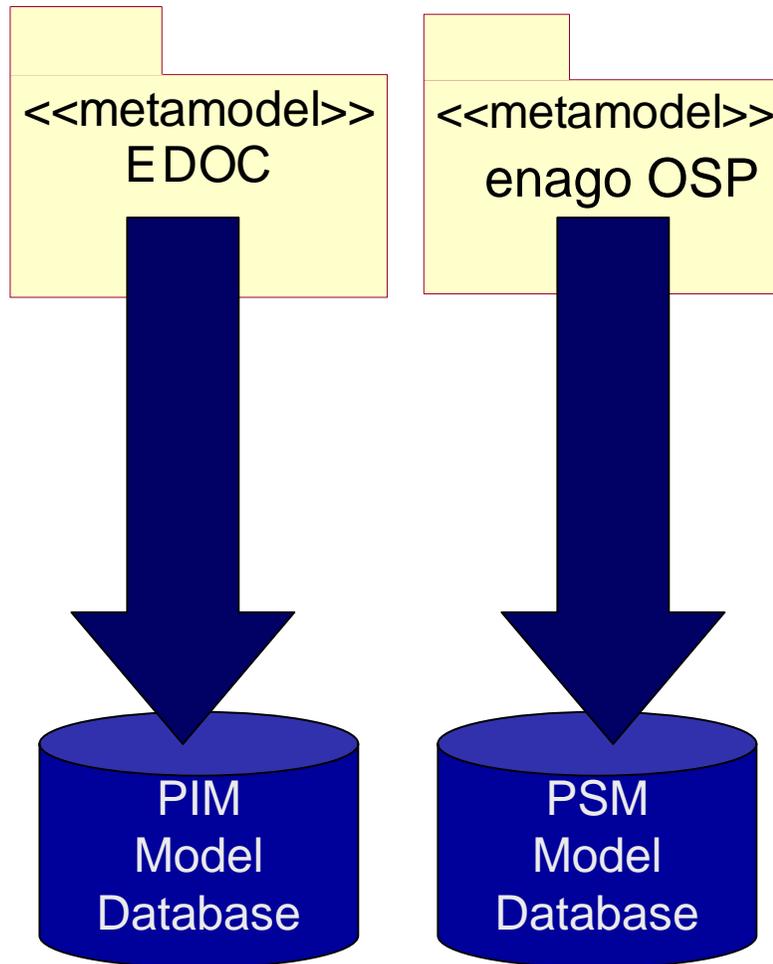
MDSC modeling infrastructure

- Meta Object Facility (MOF) provides a technology to share models based on their metamodels
- The sharing of models is to be fined between tools
- There exist a number of technology mappings of the MOF:
 - MOF to XML → XMI
 - MOF to Java → JMI
 - MOF to IDL → CORBA
- Out of a metamodel, a set of CORBA interfaces is being generated that can be used to exchange models



- ☞ *the medini tool of IKV realizes the MOF to IDL technology mapping*
- ☞ *medini is being applied to connect modeling tools to the modeling infrastructure and*
- ☞ *to establish connections within the infrastructure*

Application of medini



Automatically generate model databases out of the metamodels, using *medini*

These model databases support mechanisms for

- CORBA access
- transactions
- model change notifications
- XML import/export
- consistency check
- ...

EDOC based PIM Modeling

- EDOC is an OMG standard for modeling of enterprise distributed systems
 - *Component Collaboration Architecture* - Component based definition of business process modules, definition of component collaboration using choreographies
 - *Entity Profile* – definition of information models and their connection to business processes
 - *Events* – definition of messaging based interactions between business components
- In MDSC, EDOC is being realized as an extension to the Rational Rose modeling tool
 - Specific modeling wizards, dialogs, ...

EDOC based PIM Modeling

The screenshot displays the Rational Rose IDE interface. The main window shows a UML diagram with several components and their relationships. Two dialog boxes are open over the diagram:

- Create Choreography for Sales:** This dialog has a 'Transition' field set to 'anyData'. The 'Source code' field contains the text:


```
total <<CreateDataCode>>
```

 The 'Target code to create' field contains a list of code blocks:


```
success <<CreateDataCode>>
failure <<CreateDataCode>>
ok <<CreateDataCode>>
ShippingNoticeBT <<PortActivityCode>>
QuoteBT <<PortActivityCode>>
OrderBT <<PortActivityCode>>
PaymentNoticeBT <<PortActivityCode>>
```
- Create port for ShippingNoticeBT:** This dialog has a 'Name' field set to 'outPort'. The 'Type' section has 'Raw Port' selected. The 'Direction' section has 'Inlets' selected.

The background diagram shows components like 'ShippingNoticeBT', 'Quote', 'PaymentNoticeBT', and 'ShippingNotice'. Relationships include '<>', '<>', and '<

EDOC UML
 AddOn for Rational Rose®
 (prototype)

Menu items shown in the overlay:

- Model Properties
- Options...
- Open Script...
- New Script...
- EDOC UML**
 - Propagate model
 - Options...
- COM
- Class Wizard...

Below the menu is a small diagram showing a component with an inlet and an outlet, connected to another component.

enago OSP Modeling

- enago OSP is IKV's CORBA based platform for rapid definition, realization, deployment and operation of distributed services
 - introduces the service concept, *services* can be provided, used, sold, accounted on and so forth
 - Services are being realized by software components
- based on the CORBA metamodel, we developed a metamodel and a UML profile *for modeling of enago services*
 - realized again as Rational Rose extension

enago OSP Modeling

The image displays the Rational Rose software interface for modeling an OSP (Open Service Platform) in UML. The main window shows a Class Diagram in Logical View for a project named 'enago'. The diagram includes several components and their relationships:

- CO**: A green circle representing a component, labeled 'p_STM (non-Std)'.
- Su**: Two yellow circles representing components, labeled 'mgmt (non-Std)'.
- I**: Two orange squares representing components.

Relationships are shown as lines connecting these components. A 'Toolbox' window is open, showing various UML symbols. In the foreground, the 'Enago Component' dialog box is open, showing the configuration for a component named 'p_STM' in the 'Logical View-Sub' context. The dialog has the following fields and lists:

- Name:** p_STM
- Context:** Logical View-Sub
- Super:** (empty)
- Initial:** Logical View-Sub:p_STMinit
- Required:** (empty list)
- Supported:** mgmt

Buttons for 'Add', 'Modify', and 'Remove' are available for both the Required and Supported lists. The dialog also has 'OK' and 'Cancel' buttons at the bottom.

The bottom of the screen shows the Windows taskbar with the Start button, application icons, and system tray information including '96%' battery and the time '15:28'.

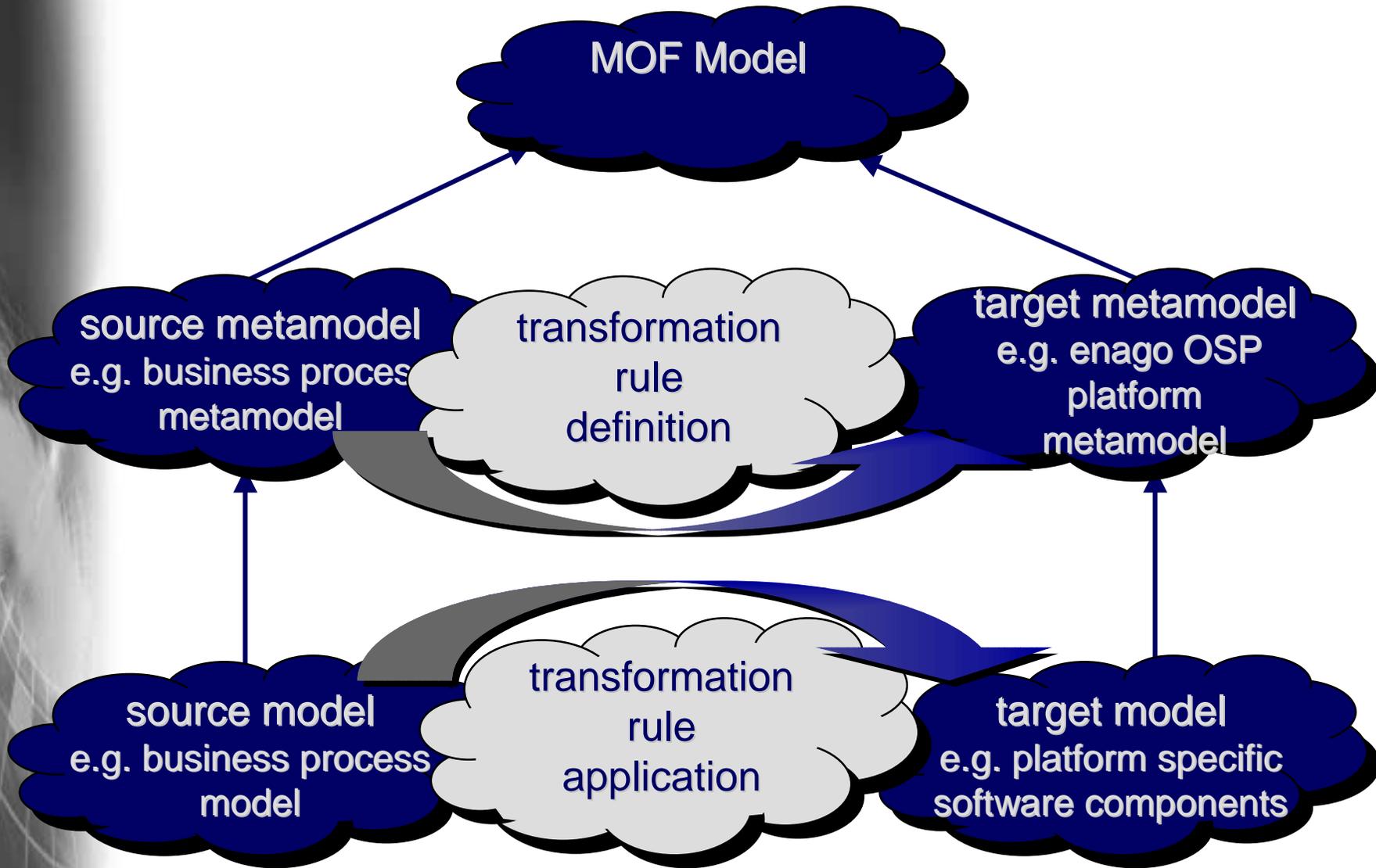
MDSC eclipse integration

- Purpose: generation of Java code out of enago OSP models
 - The modeler defines an enago OSP model of collaborating services
 - The services are being realized by software components
 - These components are implemented in the Java language
- eclipse provides an ideal Java development environment
- in MDSC, we use eclipse as target for code generation

Model Transformation

- Model transformation is the new software construction paradigm
 - Systems are being constructed by transforming a model of higher abstraction into a model of lower abstraction
 - Programming language code is also being treated as a (part of a) model of a system
- In MDSC, we find two categories of model transformations
 - Transformation of EDOC models into enago OSP models
 - Transformation of enago OSP models into Java programming language code

Model Transformation

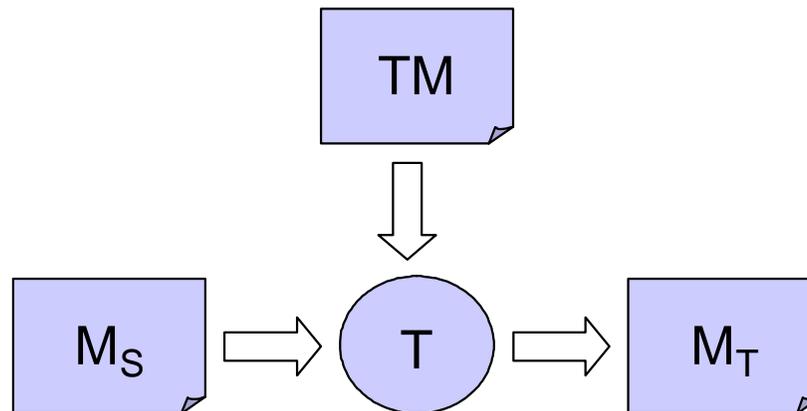


Model Transformations in MDA

- PIM -> PSM is most mentioned transformation
 - Model refinement (partially also PIM->PIM and PSM->PSM)
 - Different abstraction levels
- Also important: PIM->PIM, PSM->PSM on same abstraction level
 - Version upgrading
 - Models of the same domain with similar content but slightly different modeling concepts
 - ...

When talking about Model Transformation Techniques ...

- ... no need to stick to abstractions like PIM and PSM
- ... no need to consider abstraction levels
- ... no need to distinguish between model transformation and code generation
- ... just use terms source, target, transformation model, graph, etc.



We are on the right track!



Berlin.de
· POLITIK & VERWALTUNG · BÜRGERSERVICE · DIE HAUPTSTADT · TOURISMUS &

PROJEKT ZUKUNFT
Berlin in der Informationsgesellschaft

New Media/IT
Breitband/Mobil
Logistik

(Initiativen/Projekte (Veran

Veranstaltungen
Förderung
Kontakt
eNews
Presse
Dokumentation

Projekt des Monats Juli 2003
Modellbasierte Softwareentwicklung für E-Government
Das Berliner Faunhofer-Institut Fokus und Nit Data Corporation erhalten internationale Auszeichnung

Veränderte Links von Projekt Zukunft

