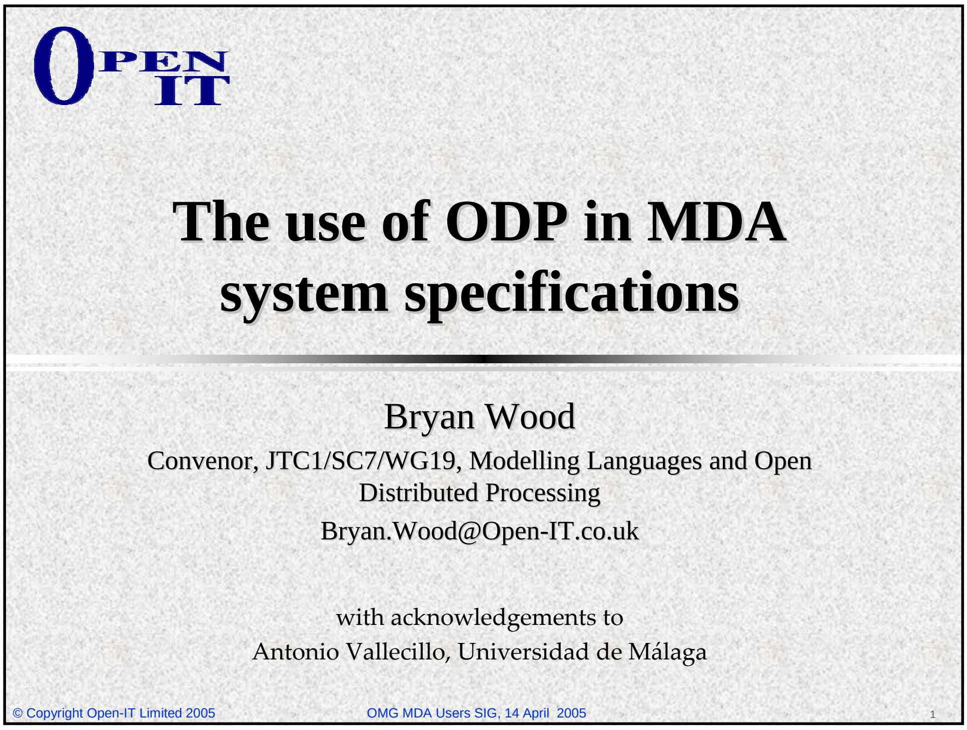


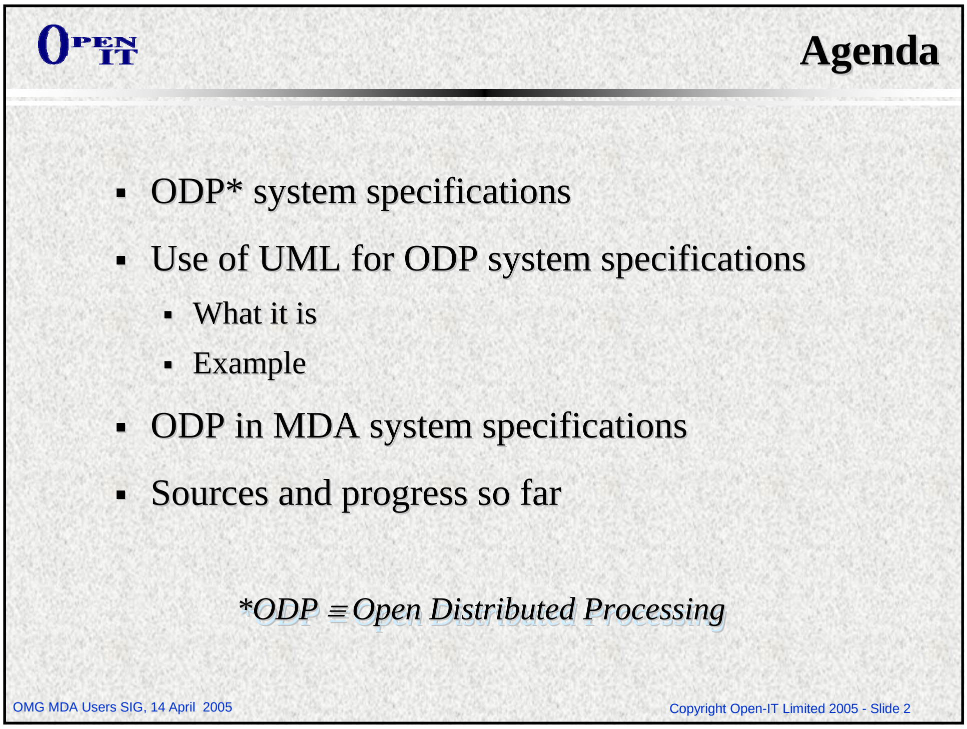
The use of ODP in MDA system specifications

Bryan Wood

Convenor, JTC1/SC7/WG19, Modelling Languages and Open Distributed Processing

Bryan.Wood@Open-IT.co.uk

with acknowledgements to
Antonio Vallecillo, Universidad de Málaga



Agenda

- ODP* system specifications
- Use of UML for ODP system specifications
  - What it is
  - Example
- ODP in MDA system specifications
- Sources and progress so far

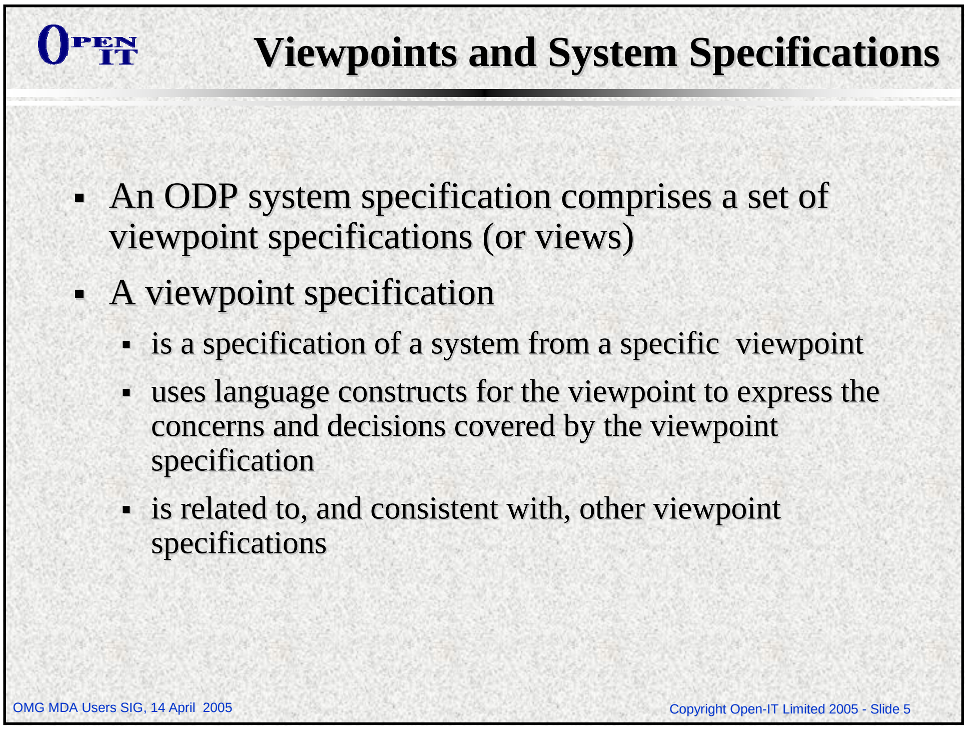
**ODP ≡ Open Distributed Processing*

ODP system specifications

- The *Reference Model of ODP (ITU-T Rec X.901-904 | ISO/IEC 10746)* defines a framework for system specification, covering all aspects of open distributed systems:
  - “enterprise” context, functionality, distribution, infrastructure, technology
- It comprises
  - a structure for system specifications in terms of viewpoints
  - a language (concepts and rules) for expressing each viewpoint specification
  - a set of object-oriented foundation modeling concepts common to all viewpoint languages

ODP viewpoints

- Different abstractions of the same system
  - each abstraction focuses on different concerns
  - each abstraction achieved using a set of viewpoint concepts and rules
- A mechanism for dealing with the complexity of distributed systems

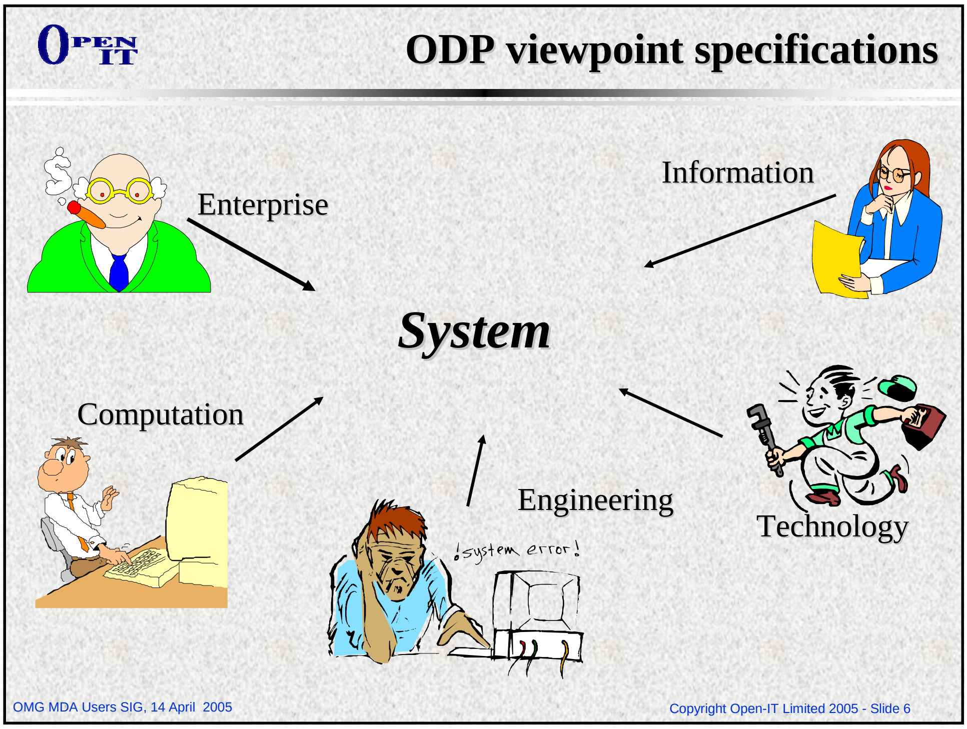


Viewpoints and System Specifications

- An ODP system specification comprises a set of viewpoint specifications (or views)
- A viewpoint specification
  - is a specification of a system from a specific viewpoint
  - uses language constructs for the viewpoint to express the concerns and decisions covered by the viewpoint specification
  - is related to, and consistent with, other viewpoint specifications

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 5



ODP viewpoint specifications

Enterprise

Information

System

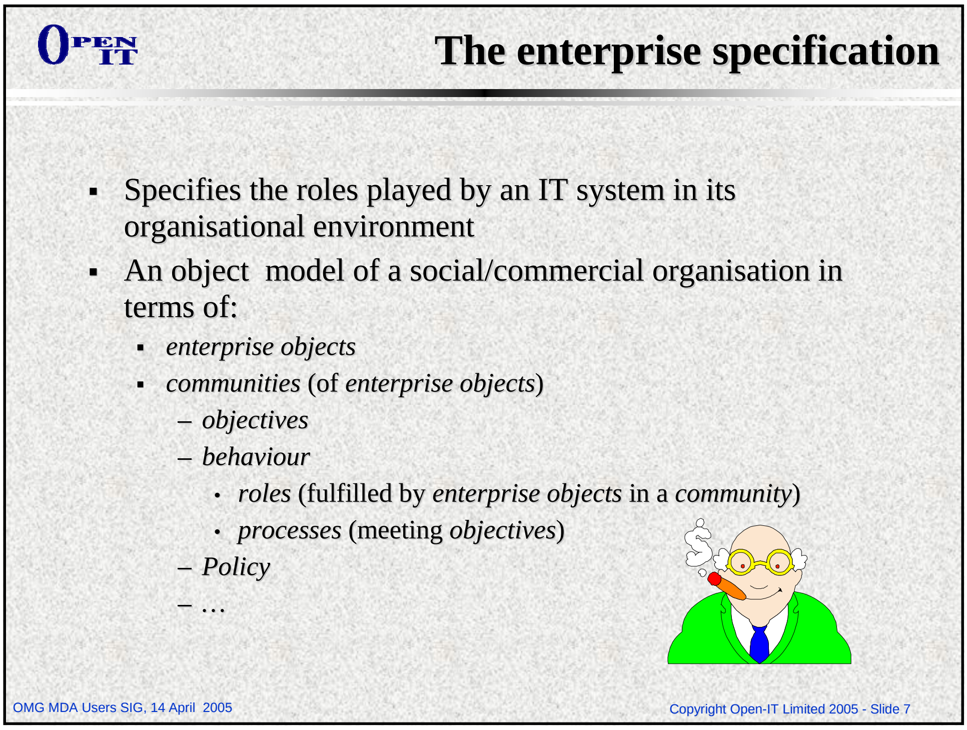
Computation

Engineering

Technology

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 6

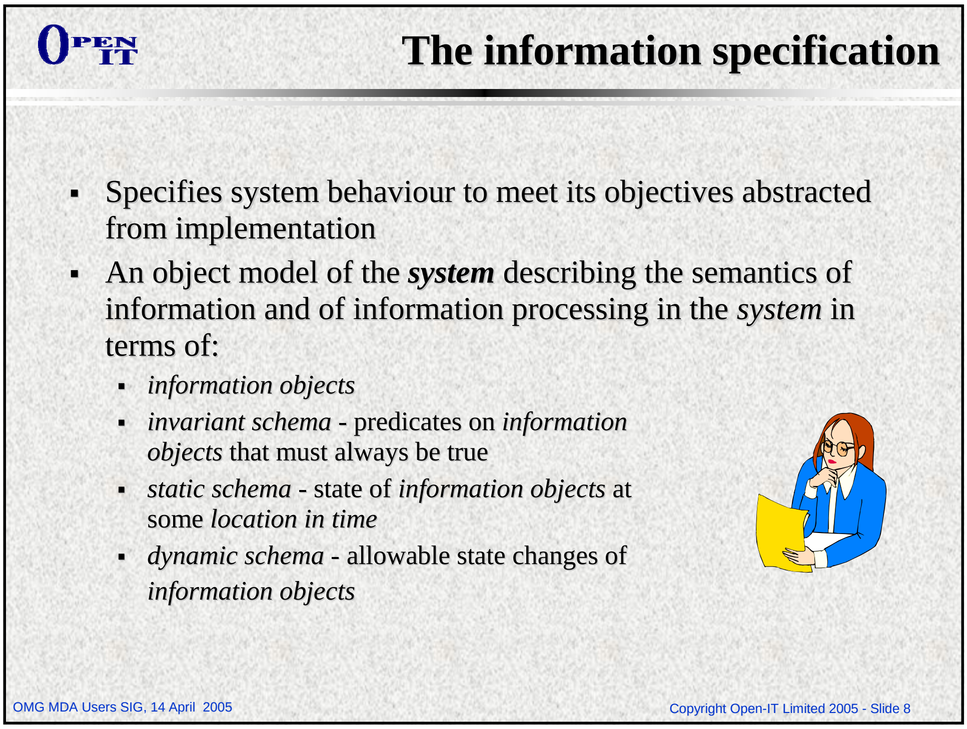


The enterprise specification

- Specifies the roles played by an IT system in its organisational environment
- An object model of a social/commercial organisation in terms of:
  - *enterprise objects*
  - *communities* (of *enterprise objects*)
    - – *objectives*
    - – *behaviour*
      - *roles* (fulfilled by *enterprise objects* in a *community*)
      - *processes* (meeting *objectives*)
    - – *Policy*
    - – …

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 7

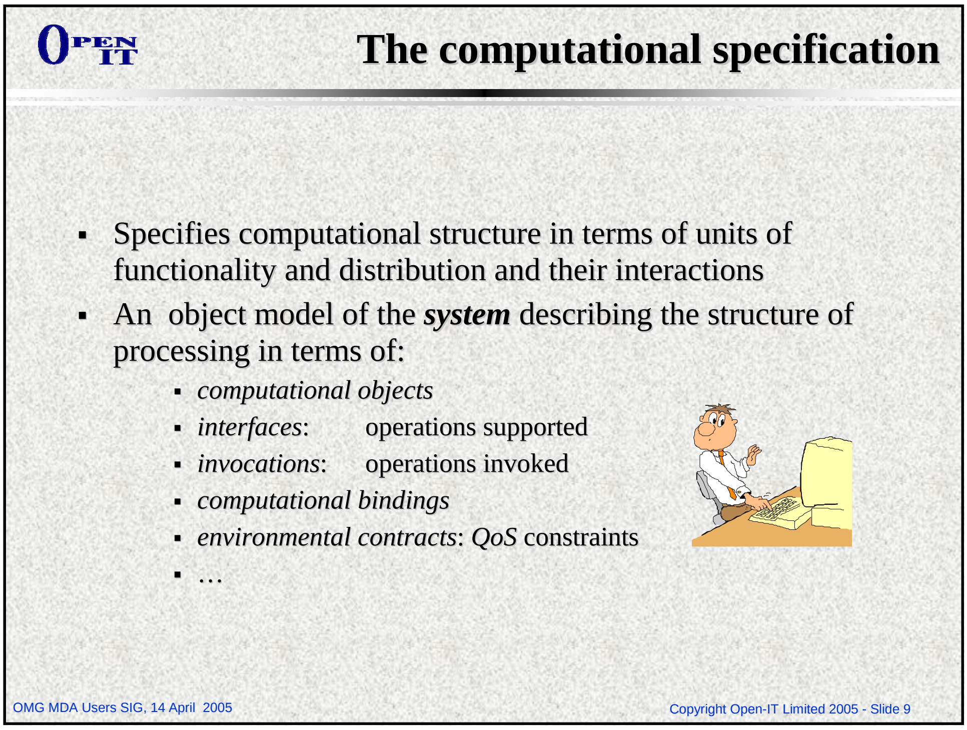


The information specification

- Specifies system behaviour to meet its objectives abstracted from implementation
- An object model of the ***system*** describing the semantics of information and of information processing in the *system* in terms of:
  - *information objects*
  - *invariant schema* - predicates on *information objects* that must always be true
  - *static schema* - state of *information objects* at some *location in time*
  - *dynamic schema* - allowable state changes of *information objects*

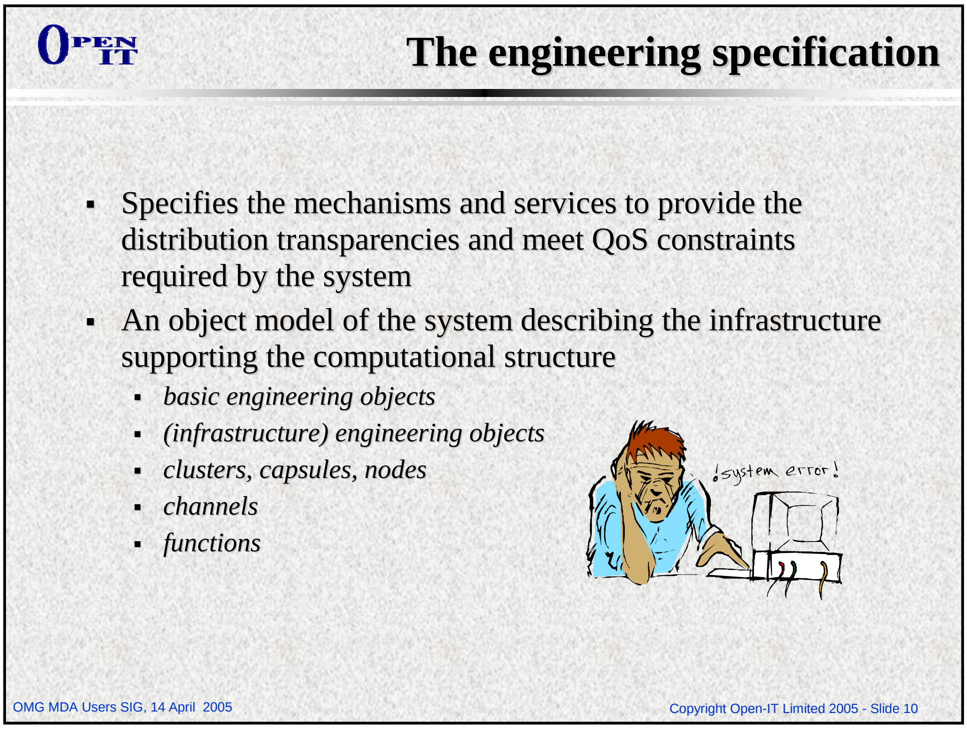
OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 8



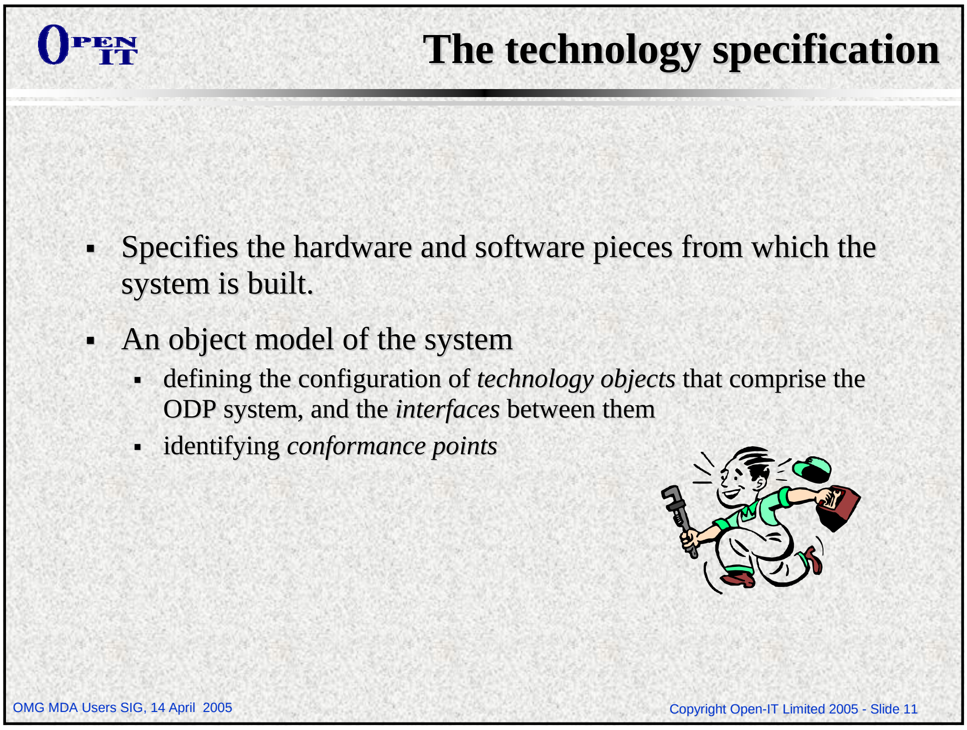
The computational specification

- Specifies computational structure in terms of units of functionality and distribution and their interactions
- An object model of the ***system*** describing the structure of processing in terms of:
  - *computational objects*
  - *interfaces*: operations supported
  - *invocations*: operations invoked
  - *computational bindings*
  - *environmental contracts*: *QoS* constraints
  - …



The engineering specification

- Specifies the mechanisms and services to provide the distribution transparencies and meet QoS constraints required by the system
- An object model of the system describing the infrastructure supporting the computational structure
  - *basic engineering objects*
  - *(infrastructure) engineering objects*
  - *clusters, capsules, nodes*
  - *channels*
  - *functions*

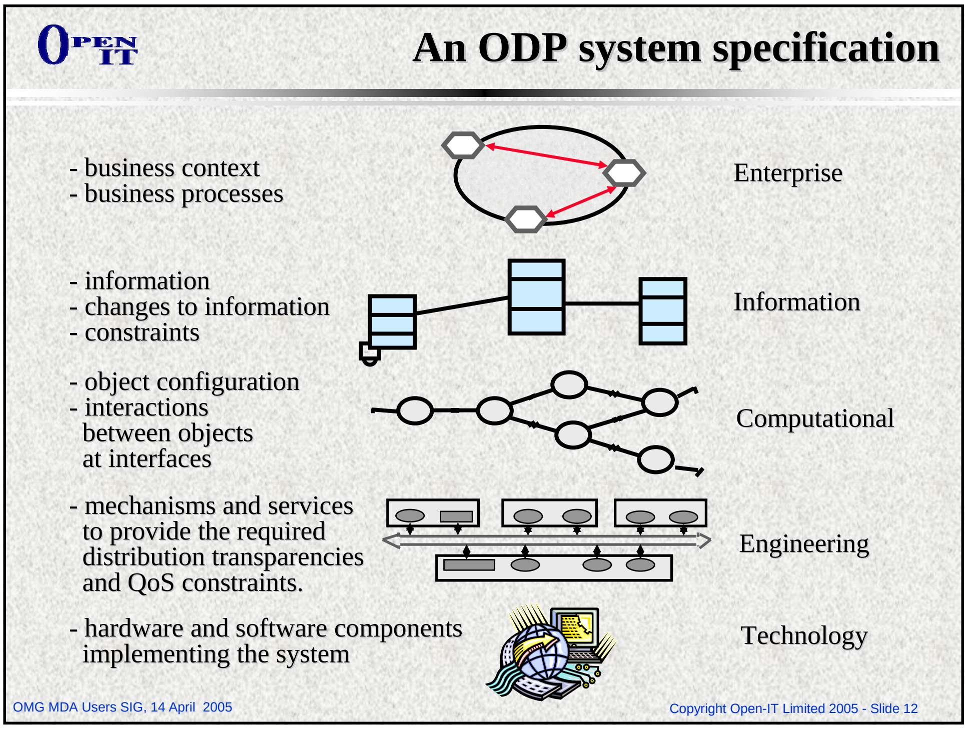


The technology specification

- Specifies the hardware and software pieces from which the system is built.
- An object model of the system
  - defining the configuration of *technology objects* that comprise the ODP system, and the *interfaces* between them
  - identifying *conformance points*

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 11



An ODP system specification

- business context
- business processes

Enterprise

- information
- changes to information
- constraints

Information

- object configuration
- interactions between objects at interfaces

Computational

- mechanisms and services to provide the required distribution transparencies and QoS constraints.

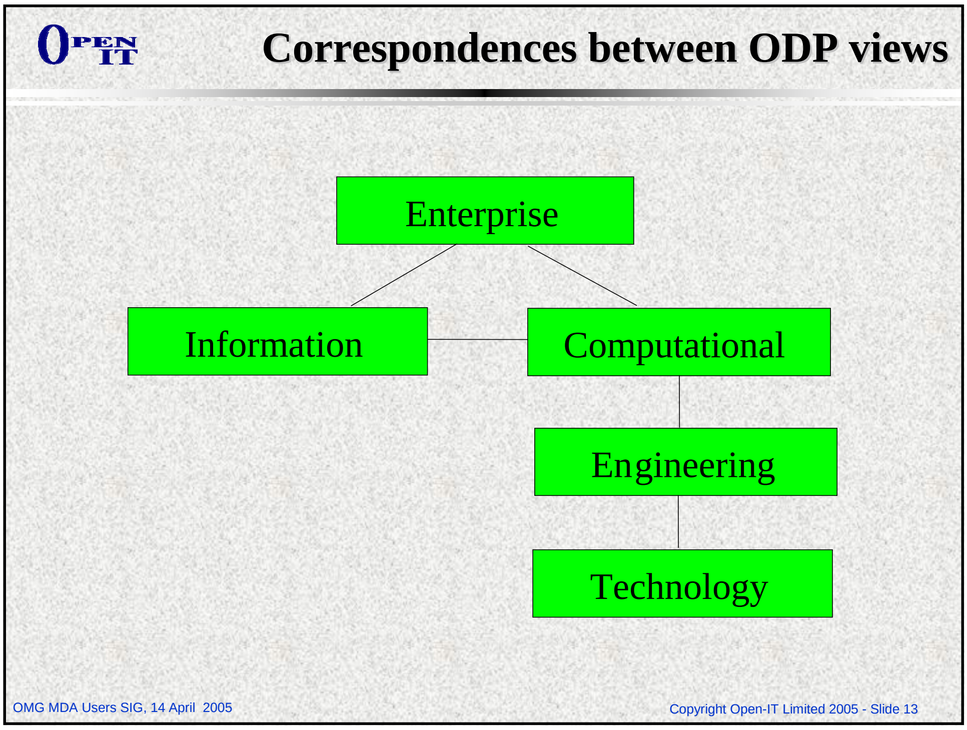
Engineering

- hardware and software components implementing the system

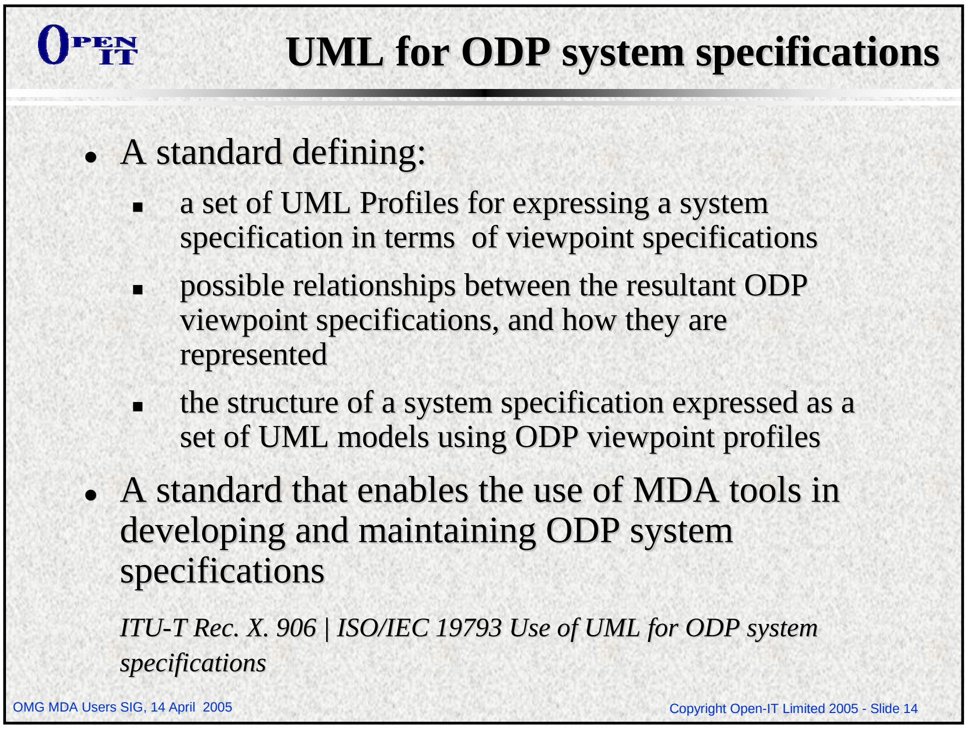
Technology

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 12



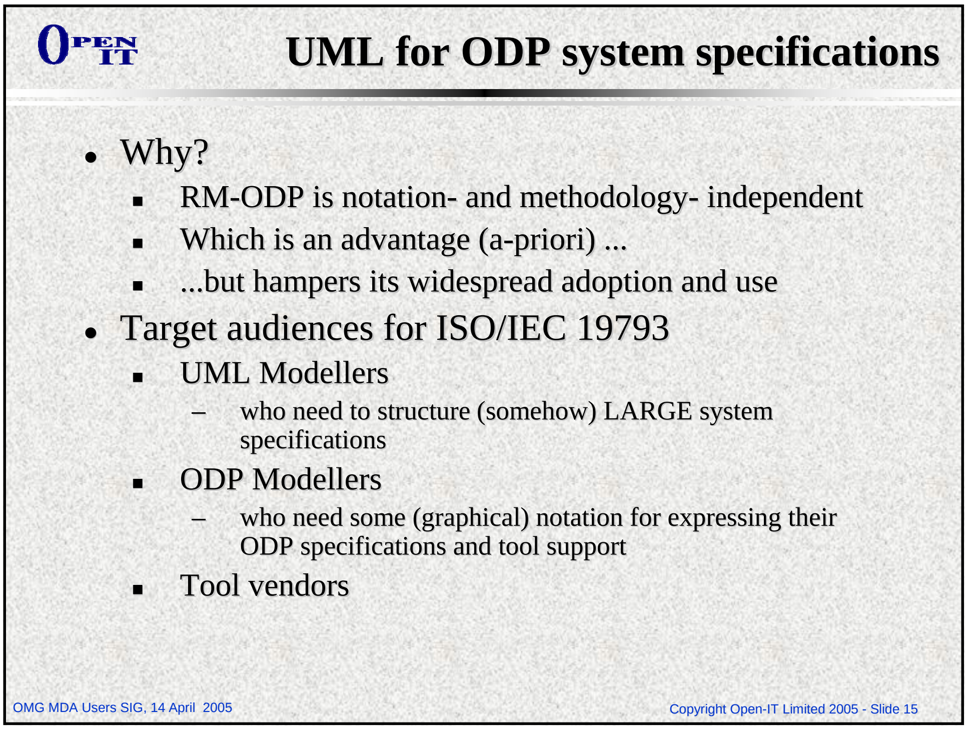
Correspondences between ODP views



UML for ODP system specifications

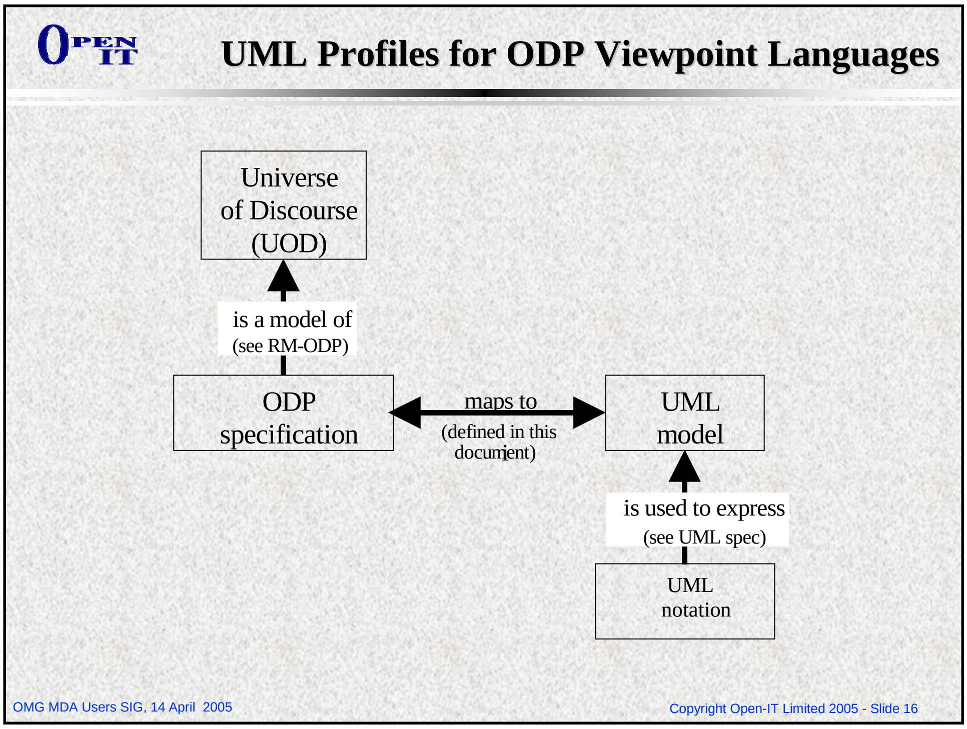
- A standard defining:
  - a set of UML Profiles for expressing a system specification in terms of viewpoint specifications
  - possible relationships between the resultant ODP viewpoint specifications, and how they are represented
  - the structure of a system specification expressed as a set of UML models using ODP viewpoint profiles
- A standard that enables the use of MDA tools in developing and maintaining ODP system specifications

ITU-T Rec. X. 906 | ISO/IEC 19793 Use of UML for ODP system specifications



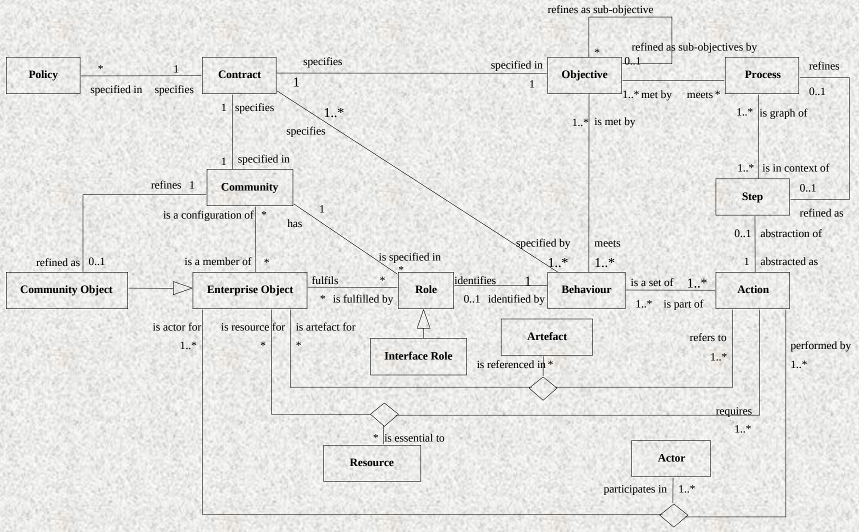
UML for ODP system specifications

- Why?
  - RM-ODP is notation- and methodology- independent
  - Which is an advantage (a-priori) ...
  - ...but hampers its widespread adoption and use
- Target audiences for ISO/IEC 19793
  - UML Modellers
    - who need to structure (somehow) LARGE system specifications
  - ODP Modellers
    - who need some (graphical) notation for expressing their ODP specifications and tool support
  - Tool vendors



UML Profiles for ODP Viewpoint Languages

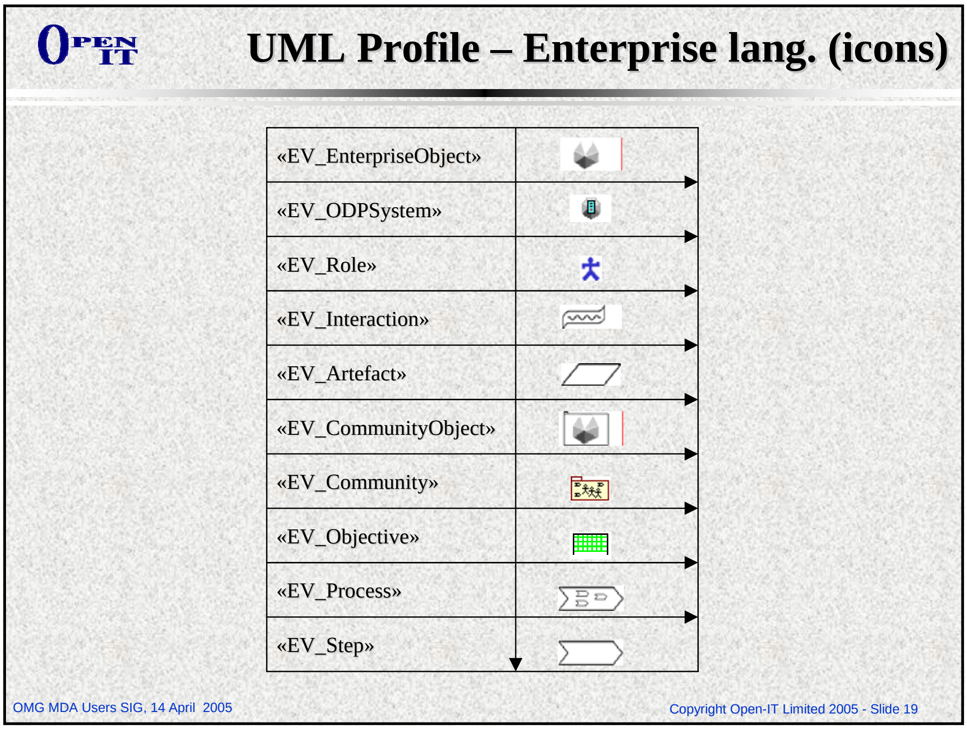
Enterprise metamodel (excerpt)



Policy
Contract
Objective
Process
Community
Step
Community Object
Enterprise Object
Role
Behaviour
Action
Interface Role
Artefact
Resource
Actor
refines as sub-objective
refined as sub-objectives by
specified in
specifies
specifies
specified in
specifies
specifies
specified in
refines
refined as
is a configuration of
is a member of
has
is specified in
fulfils
is fulfilled by
identifies
identified by
specified by
meets
met by
meets
is met by
is graph of
is in context of
refines
refined as
abstraction of
abstracted as
is a set of
is part of
is actor for
is resource for
is artefact for
is referenced in
refers to
performed by
requires
is essential to
participates in

UML Profile – Enterprise lang. (excerpt)

Concept	Stereotype	UML Metaclass
ODP System	«EV_ODPSystem»	Class
Field of Application	«EV_FieldOfApplication»	Comment attached to Package
Community	«EV_Community»	Subsystem
Enterprise Object	«EV_EnterpriseObject»	Class
Enterprise Object fulfilling Role	«EV_FulfilsRole»	Association
Community Object	«EV_CommunityObject»	Class
Objective	«EV_Objective»	Class
Role	«EV_Role»	Class, StateMachine (in role model), Partition (in process model)
Action	«EV_Action»	State
Interaction	«EV_Interaction»	Class, StateMachine, ActivityGraph
Process	«EV_Process»	Class, ActivityGraph
Step	«EV_Step»	ActionState
Artefact	«EV_Artefact»	ObjectFlowState (in process model), Signal (in role model)



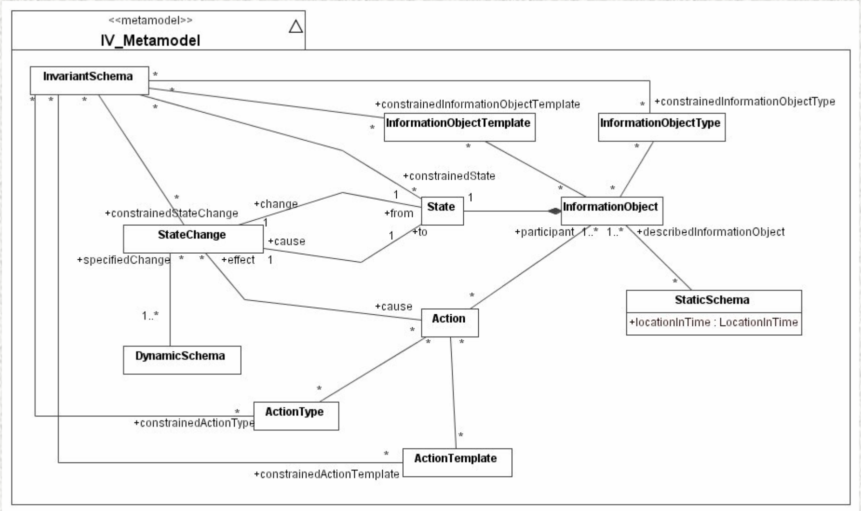
UML Profile – Enterprise lang. (icons)

«EV_EnterpriseObject»	
«EV_ODPSystem»	
«EV_Role»	
«EV_Interaction»	
«EV_Artefact»	▱
«EV_CommunityObject»	
«EV_Community»	
«EV_Objective»	🟩
«EV_Process»	
«EV_Step»	

OMG MDA Users SIG, 14 April 2005

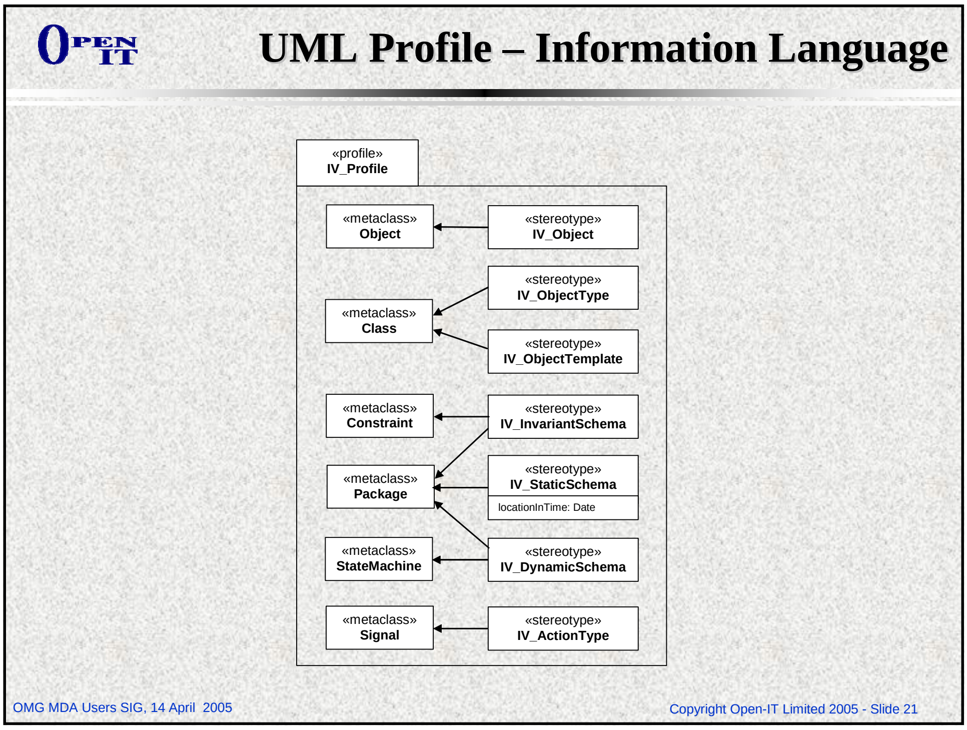
Copyright Open-IT Limited 2005 - Slide 19

Information Language metamodel



OMG MDA Users SIG, 14 April 2005

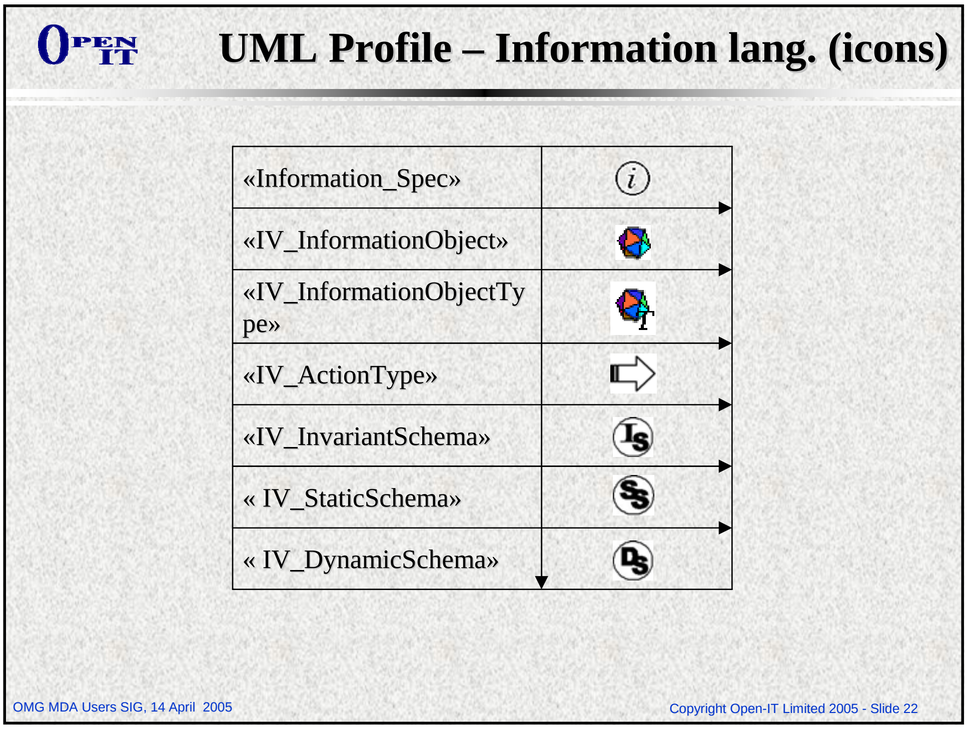
Copyright Open-IT Limited 2005 - Slide 20



UML Profile – Information Language

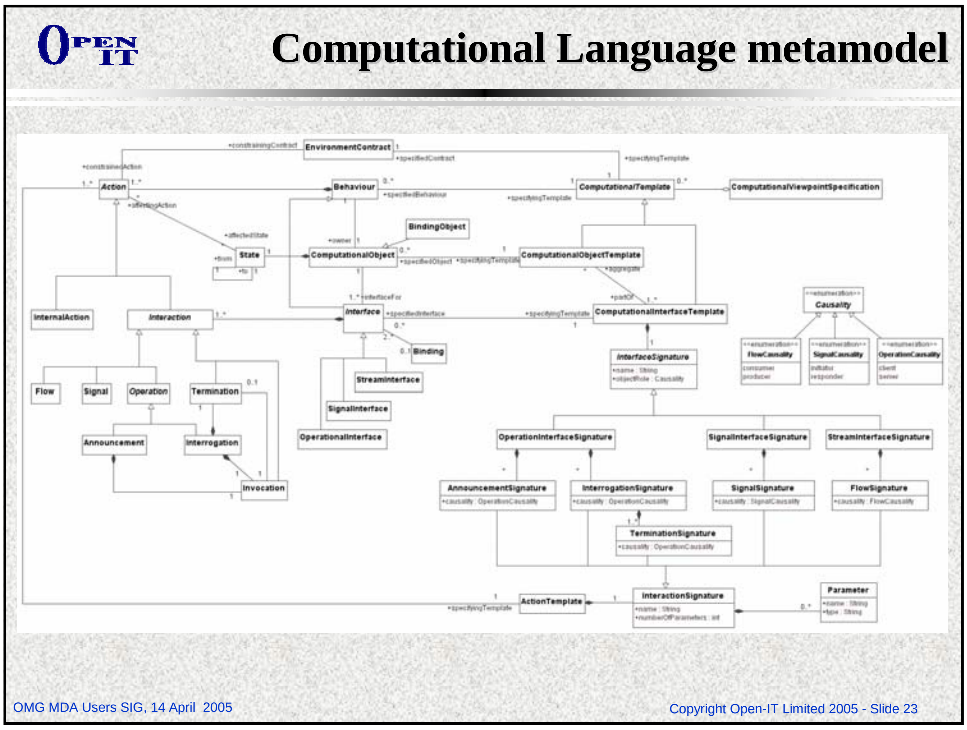
«profile»
IV_Profile

«metaclass»	«stereotype»
Object	IV_Object
Class	IV_ObjectType
Class	IV_ObjectTemplate
Constraint	IV_InvariantSchema
Package	IV_InvariantSchema
Package	IV_StaticSchema locationInTime: Date
Package	IV_DynamicSchema
StateMachine	IV_DynamicSchema
Signal	IV_ActionType



UML Profile – Information lang. (icons)

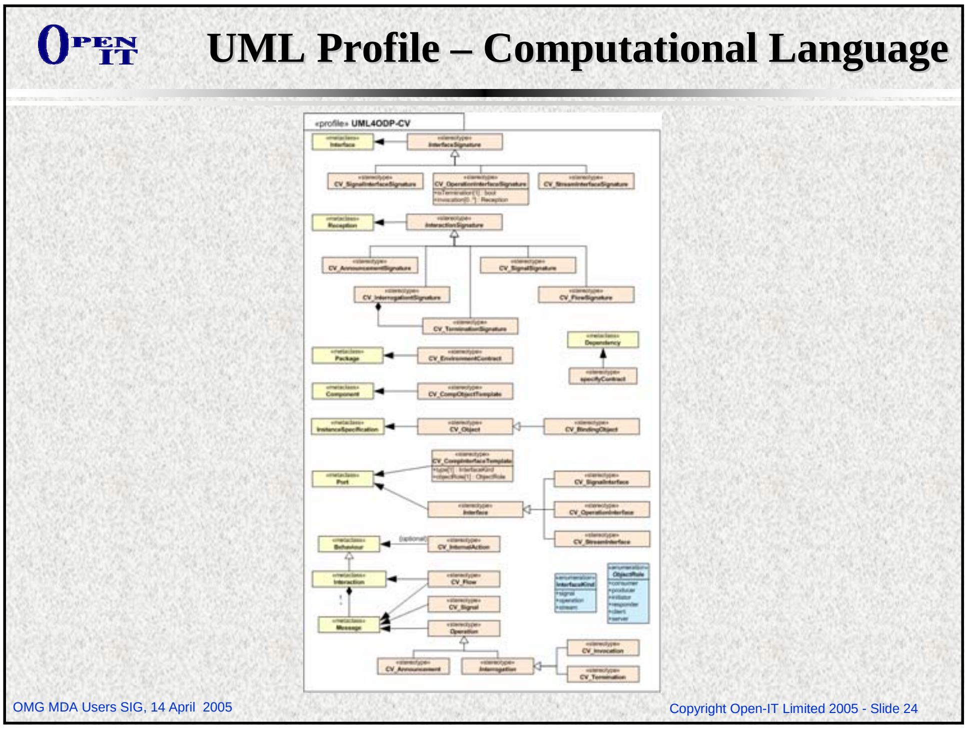
«Information_Spec»	ⓘ
«IV_InformationObject»	●
«IV_InformationObjectType»	●
«IV_ActionType»	⇨
«IV_InvariantSchema»	IS
« IV_StaticSchema»	SS
« IV_DynamicSchema»	DS



Computational Language metamodel

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 23

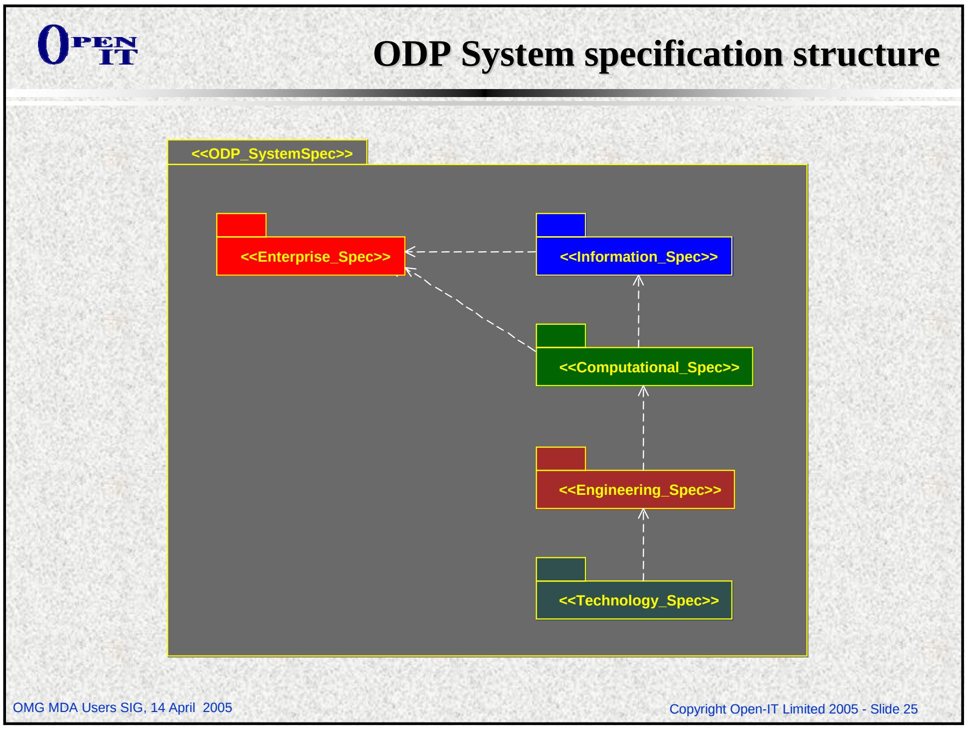


UML Profile – Computational Language

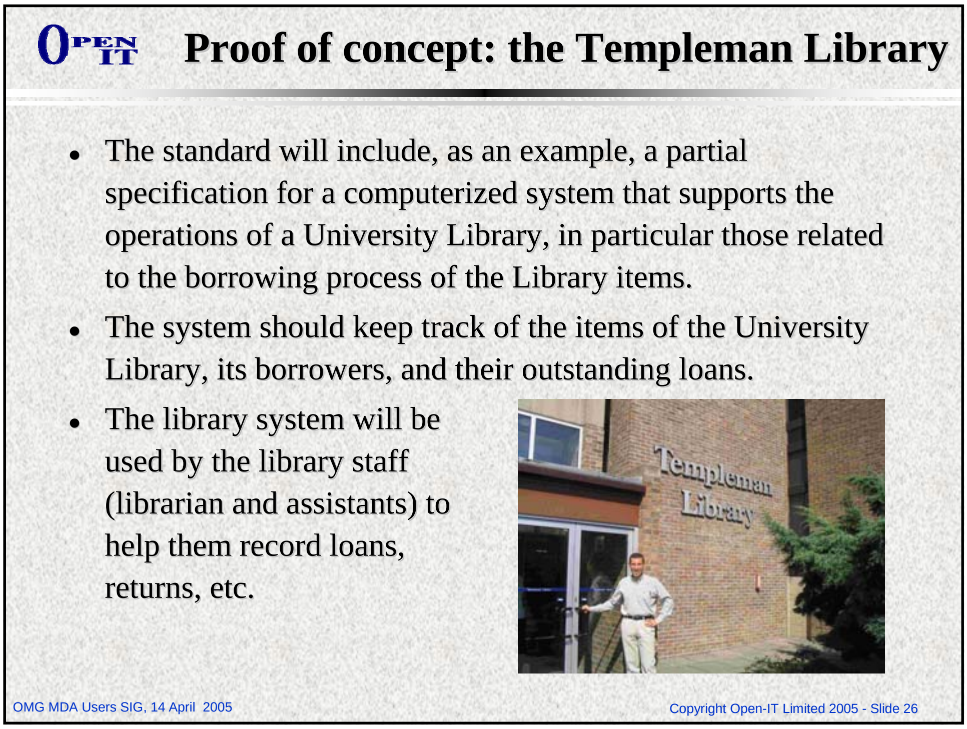
«profile» UML4ODP-CV
«metaclass» Interface
«stereotype» InterfaceSignature
«stereotype» CV_SignalInterfaceSignature
«stereotype» CV_OperationInterfaceSignature
+isTermination[1] : bool
+invocation[0..*] : Reception
«stereotype» CV_StreamInterfaceSignature
«metaclass» Reception
«stereotype» InteractionSignature
«stereotype» CV_AnnouncementSignature
«stereotype» CV_SignalSignature
«stereotype» CV_InterrogationtSignature
«stereotype» CV_FlowSignature
«stereotype» CV_TerminationSignature
«metaclass» Dependency
«metaclass» Package
«stereotype» CV_EnvironmentContract
«stereotype» specifyContract
«metaclass» Component
«stereotype» CV_CompObjectTemplate
«metaclass» InstanceSpecification
«stereotype» CV_Object
«stereotype» CV_BindingObject
«stereotype» CV_CompInterfaceTemplate
+type[1] : InterfaceKind
+objectRole[1] : ObjectRole
«metaclass» Port
«stereotype» CV_SignalInterface
«stereotype» Interface
«stereotype» CV_OperationInterface
«stereotype» CV_StreamInterface
«metaclass» Behaviour
(optional)
«stereotype» CV_InternalAction
«enumeration» InterfaceKind
+signal
+operation
+stream
«enumeration» ObjectRole
+consumer
+producer
+initiator
+responder
+client
+server
«metaclass» Interaction
«stereotype» CV_Flow
«stereotype» CV_Signal
«metaclass» Message
«stereotype» Operation
«stereotype» CV_Invocation
«stereotype» CV_Announcement
«stereotype» Interrogation
«stereotype» CV_Termination

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 24



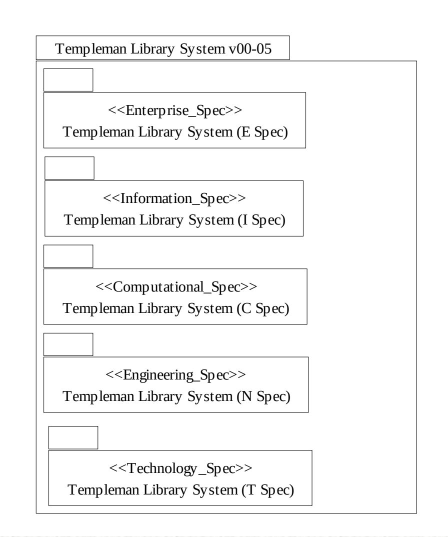
ODP System specification structure



Proof of concept: the Templeman Library

- The standard will include, as an example, a partial specification for a computerized system that supports the operations of a University Library, in particular those related to the borrowing process of the Library items.
- The system should keep track of the items of the University Library, its borrowers, and their outstanding loans.
- The library system will be used by the library staff (librarian and assistants) to help them record loans, returns, etc.

UML specification of the ODP system



Templeman Library System v00-05

<<Enterprise_Spec>>
Templeman Library System (E Spec)

<<Information_Spec>>
Templeman Library System (I Spec)

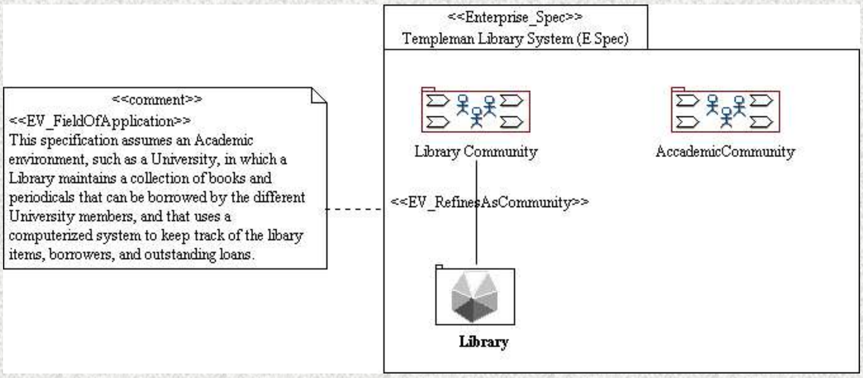
<<Computational_Spec>>
Templeman Library System (C Spec)

<<Engineering_Spec>>
Templeman Library System (N Spec)

<<Technology_Spec>>
Templeman Library System (T Spec)

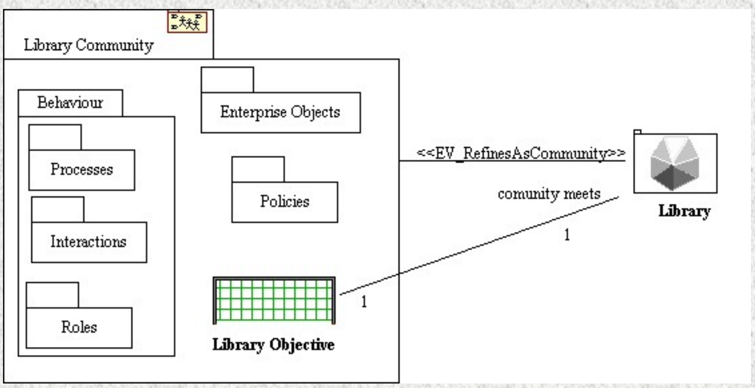
Enterprise Specification

Enterprise spec – Global

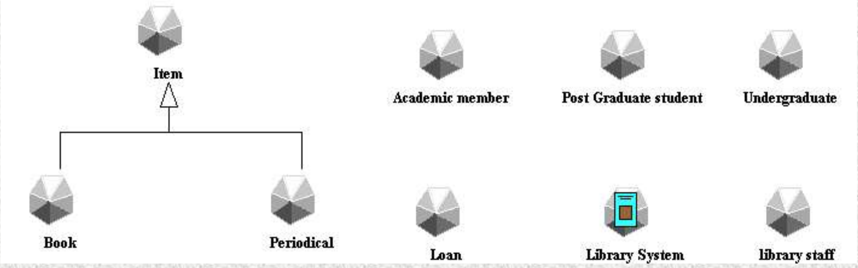


<<comment>>
<<EV_FieldOfApplication>>
This specification assumes an Academic environment, such as a University, in which a Library maintains a collection of books and periodicals that can be borrowed by the different University members, and that uses a computerized system to keep track of the libary items, borrowers, and outstanding loans.

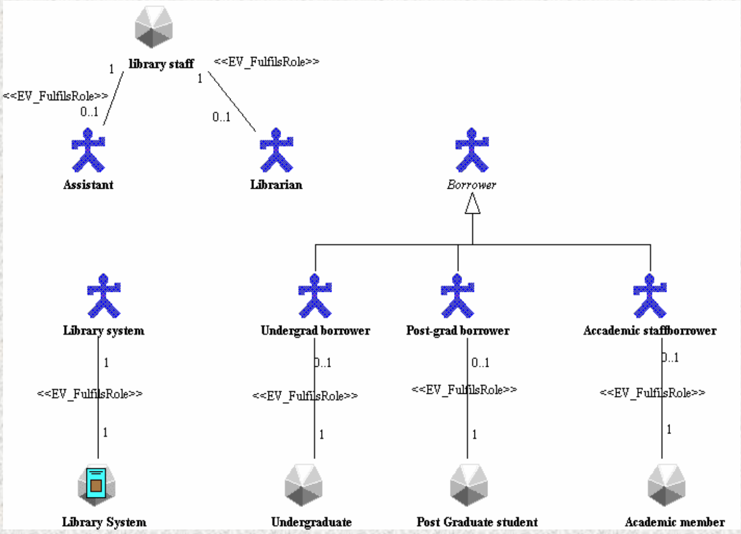
Enterprise spec – Library Community



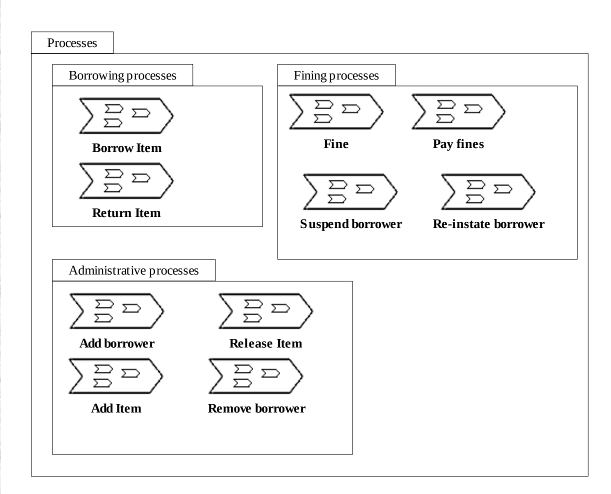
Enterprise spec – enterprise objects



Enterprise spec – Role fulfilment



Enterprise spec – Processes



Processes

Borrowing processes

Borrow Item

Return Item

Fining processes

Fine

Pay fines

Suspend borrower

Re-instate borrower

Administrative processes

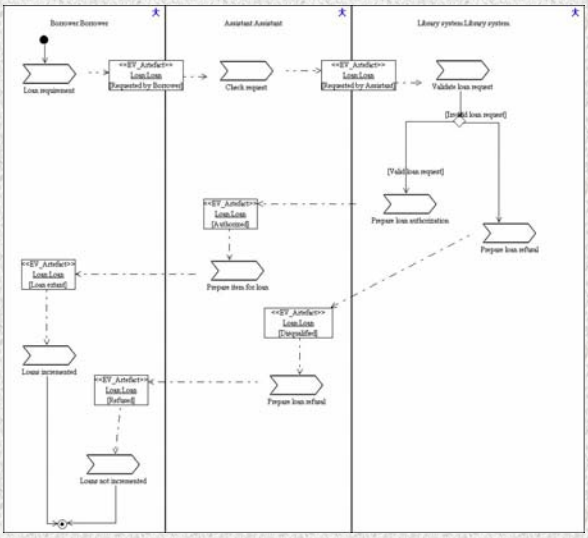
Add borrower

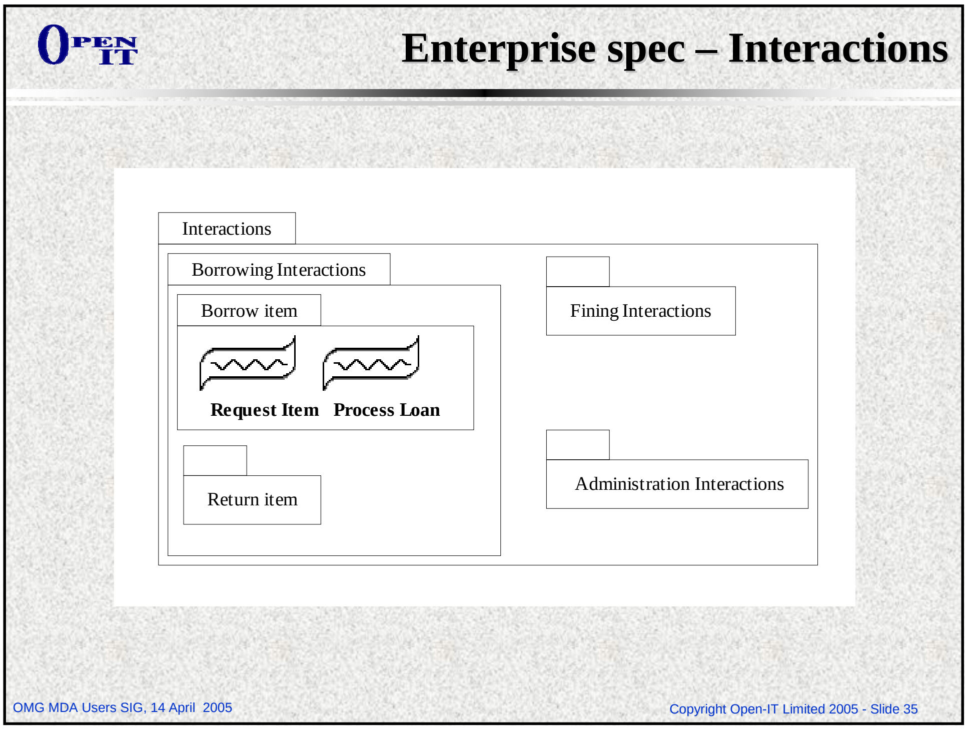
Release Item

Add Item

Remove borrower

Enterprise spec – Activity graph - Borrow Item

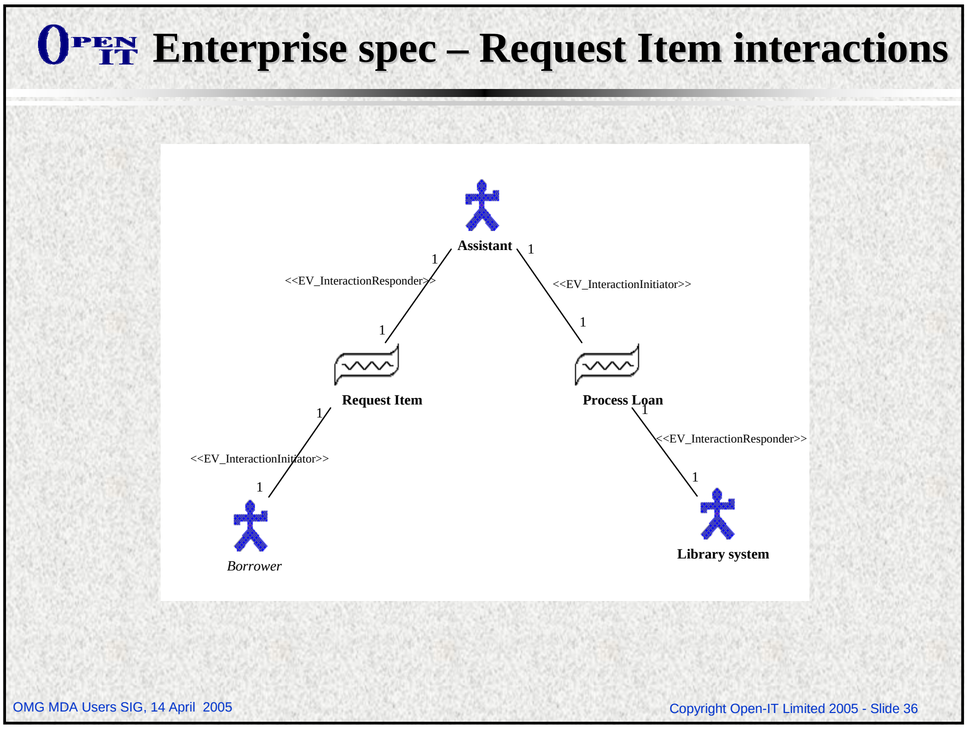




Enterprise spec – Interactions

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 35



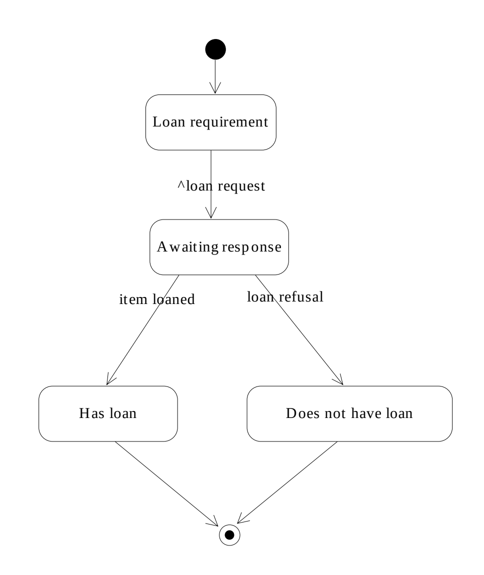
Enterprise spec – Request Item interactions

Assistant
1
1
<<EV_InteractionResponder>>
<<EV_InteractionInitiator>>
1
1
Request Item
Process Loan
1
1
<<EV_InteractionInitiator>>
<<EV_InteractionResponder>>
1
1
Borrower
Library system

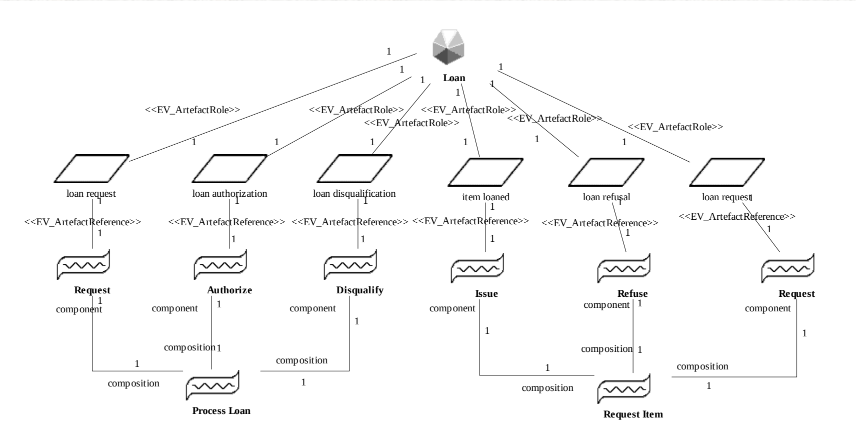
OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 36

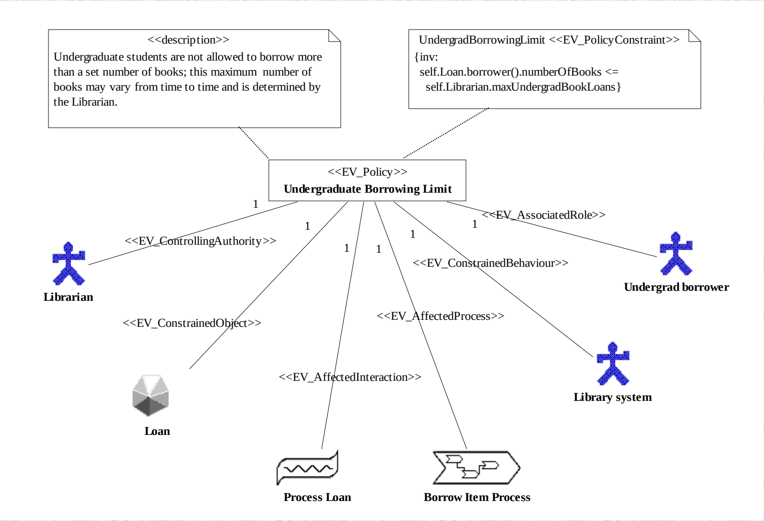
Enterprise spec – state diagram for Borrower role



Enterprise spec – Artefact roles of Loan



Enterprise spec – Policies

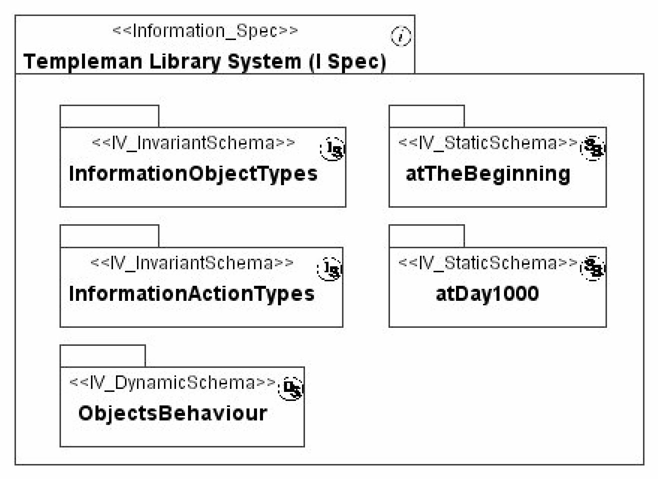


<<description>>
Undergraduate students are not allowed to borrow more than a set number of books; this maximum number of books may vary from time to time and is determined by the Librarian.

UndergradBorrowingLimit <<EV_PolicyConstraint>>
{inv:
self.Loan.borrower().numberOfBooks <=
self.Librarian.maxUndergradBookLoans}

Information Specification

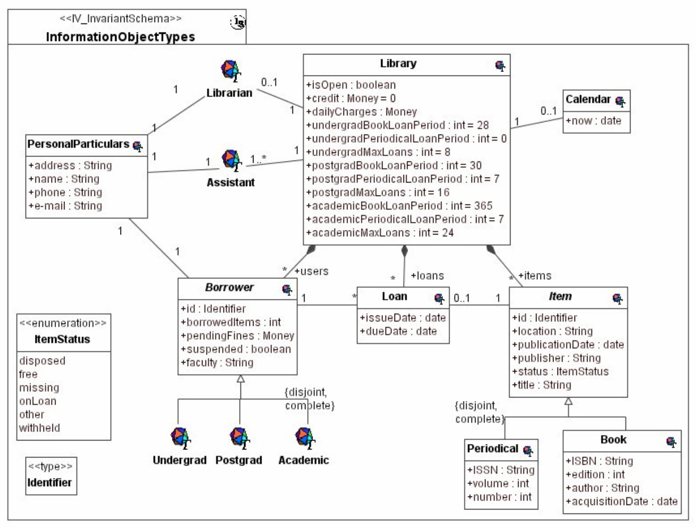
Information specification



OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 41

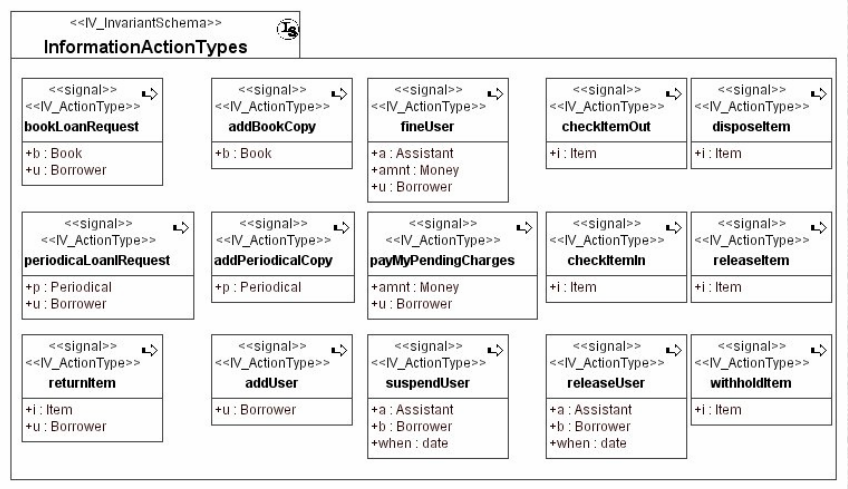
Information spec – Object types



OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 42

Information spec – Action types


<<IV_InvariantSchema>>
InformationActionTypes

<<signal>> <<IV_ActionType>> **bookLoanRequest**
+b : Book
+u : Borrower

<<signal>> <<IV_ActionType>> **addBookCopy**
+b : Book

<<signal>> <<IV_ActionType>> **fineUser**
+a : Assistant
+amnt : Money
+u : Borrower

<<signal>> <<IV_ActionType>> **checkItemOut**
+i : Item

<<signal>> <<IV_ActionType>> **disposeItem**
+i : Item

<<signal>> <<IV_ActionType>> **periodicaLoanlRequest**
+p : Periodical
+u : Borrower

<<signal>> <<IV_ActionType>> **addPeriodicalCopy**
+p : Periodical

<<signal>> <<IV_ActionType>> **payMyPendingCharges**
+amnt : Money
+u : Borrower

<<signal>> <<IV_ActionType>> **checkItemIn**
+i : Item

<<signal>> <<IV_ActionType>> **releaseItem**
+i : Item

<<signal>> <<IV_ActionType>> **returnItem**
+i : Item
+u : Borrower

<<signal>> <<IV_ActionType>> **addUser**
+u : Borrower

<<signal>> <<IV_ActionType>> **suspendUser**
+a : Assistant
+b : Borrower
+when : date

<<signal>> <<IV_ActionType>> **releaseUser**
+a : Assistant
+b : Borrower
+when : date

<<signal>> <<IV_ActionType>> **withholdItem**
+i : Item

Information spec – invariant schemata

context Library **inv** undergradLimits:
(undergradMaxLoans = 8) **and**
(undergradBookLoanPeriod = 28) **and**
(undergradPeriodicalLoanPeriod = 0)

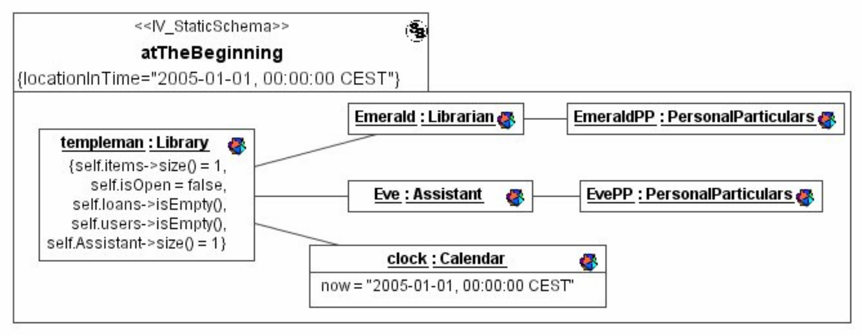
context Library **inv** uniqueIdentifiers:
self.items->**forAll**(itm1,itm2 | itm1.id <> itm2.id) **and**
self.users->**forAll**(usr1,usr2 | usr1.id <> usr2.id)

context Library **inv** oneLibrarianAndOneClockWhileOpen:
self.isOpen **implies**
(self.Librarian->**size**() = 1) **and** (self.Calendar->**size**() = 1)

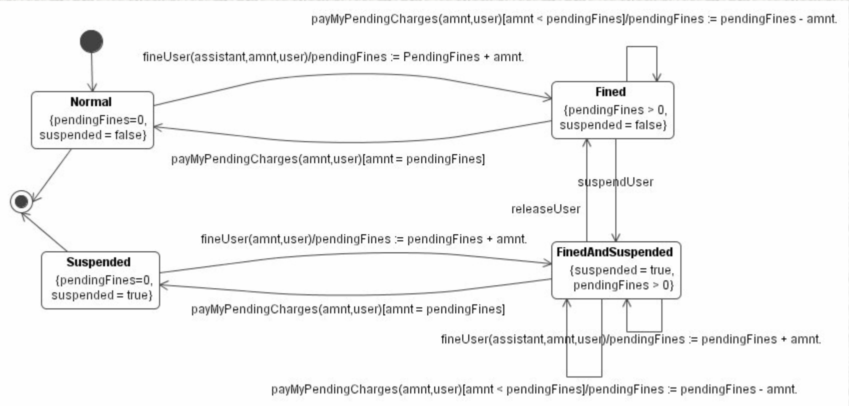
context Library **inv** consistentNumberOfLoans:
self.users.borrowedItems->**sum**() = self.loans->**size**()

context Loan **inv** wellFormedLoans: issueDate <= dueDate

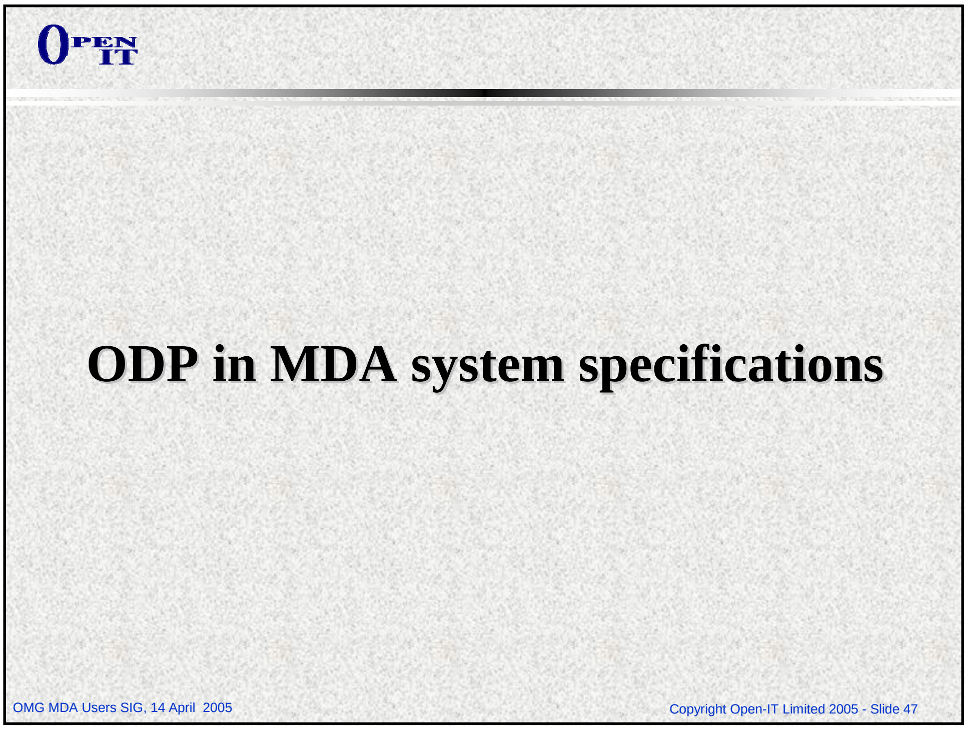
Information spec – Static schema



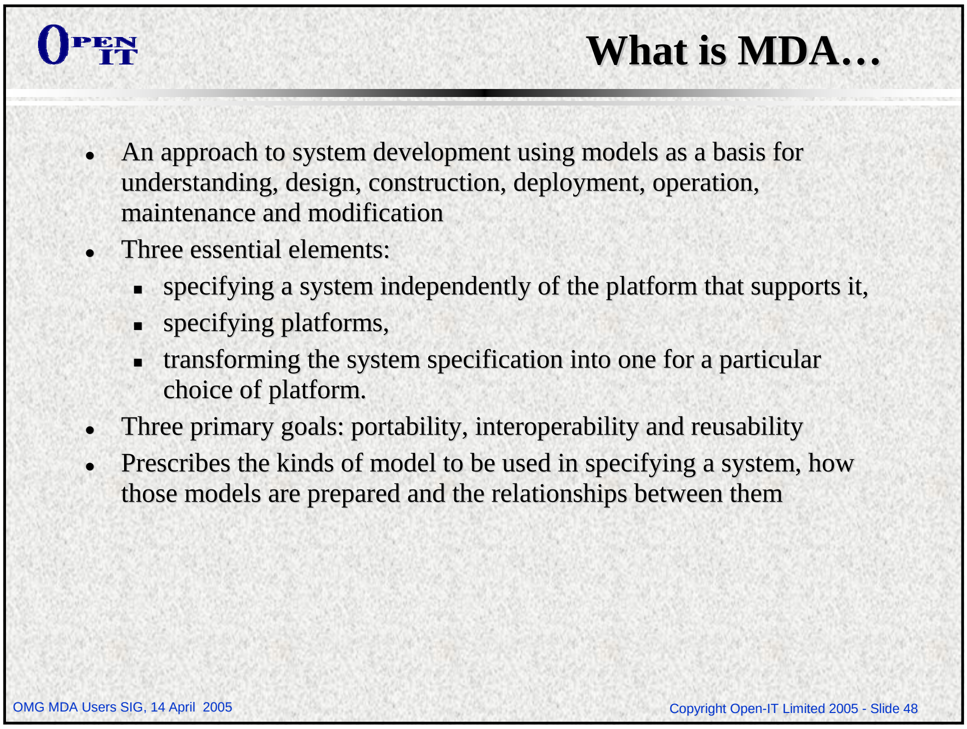
Information spec – Dynamic schema



payMyPendingCharges(amnt,user)[amnt < pendingFines]/pendingFines := pendingFines - amnt.
fineUser(assistant,amnt,user)/pendingFines := PendingFines + amnt.
Normal
{pendingFines=0,
suspended = false}
Fined
{pendingFines > 0,
suspended = false}
payMyPendingCharges(amnt,user)[amnt = pendingFines]
suspendUser
releaseUser
fineUser(amnt,user)/pendingFines := pendingFines + amnt.
Suspended
{pendingFines=0,
suspended = true}
FinedAndSuspended
{suspended = true,
pendingFines > 0}
payMyPendingCharges(amnt,user)[amnt = pendingFines]
fineUser(assistant,amnt,user)/pendingFines := pendingFines + amnt.
payMyPendingCharges(amnt,user)[amnt < pendingFines]/pendingFines := pendingFines - amnt.

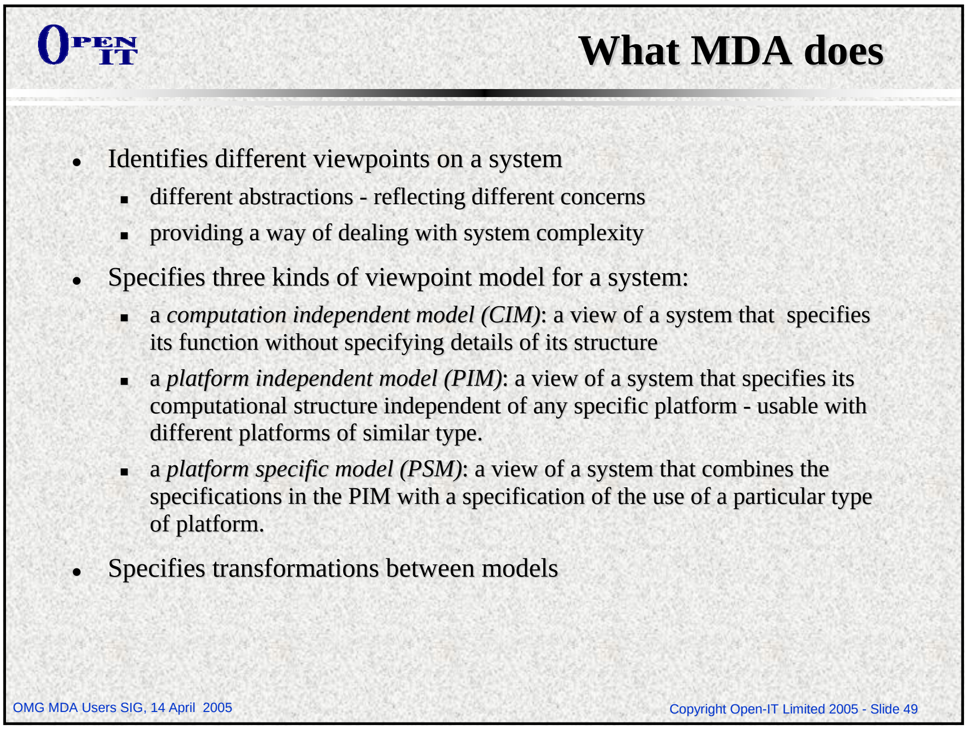


ODP in MDA system specifications



What is MDA…

- An approach to system development using models as a basis for understanding, design, construction, deployment, operation, maintenance and modification
- Three essential elements:
  - specifying a system independently of the platform that supports it,
  - specifying platforms,
  - transforming the system specification into one for a particular choice of platform.
- Three primary goals: portability, interoperability and reusability
- Prescribes the kinds of model to be used in specifying a system, how those models are prepared and the relationships between them

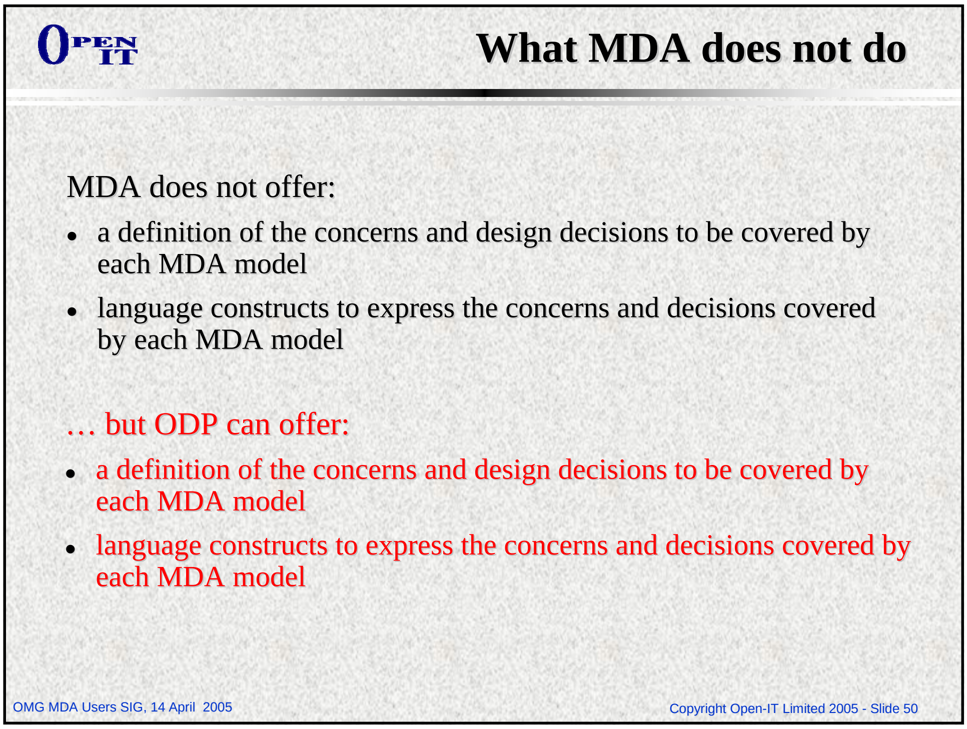


What MDA does

- Identifies different viewpoints on a system
  - different abstractions - reflecting different concerns
  - providing a way of dealing with system complexity
- Specifies three kinds of viewpoint model for a system:
  - a *computation independent model (CIM)*: a view of a system that specifies its function without specifying details of its structure
  - a *platform independent model (PIM)*: a view of a system that specifies its computational structure independent of any specific platform - usable with different platforms of similar type.
  - a *platform specific model (PSM)*: a view of a system that combines the specifications in the PIM with a specification of the use of a particular type of platform.
- Specifies transformations between models

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 49



What MDA does not do

MDA does not offer:

- a definition of the concerns and design decisions to be covered by each MDA model
- language constructs to express the concerns and decisions covered by each MDA model

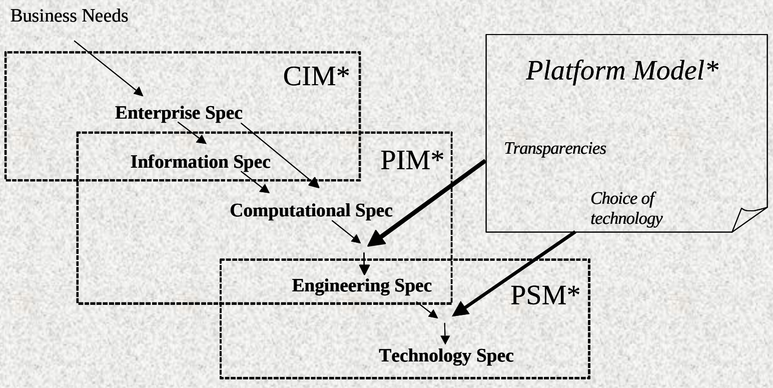
… but ODP can offer:

- a definition of the concerns and design decisions to be covered by each MDA model
- language constructs to express the concerns and decisions covered by each MDA model

OMG MDA Users SIG, 14 April 2005

Copyright Open-IT Limited 2005 - Slide 50

ODP system specifications and the MDA



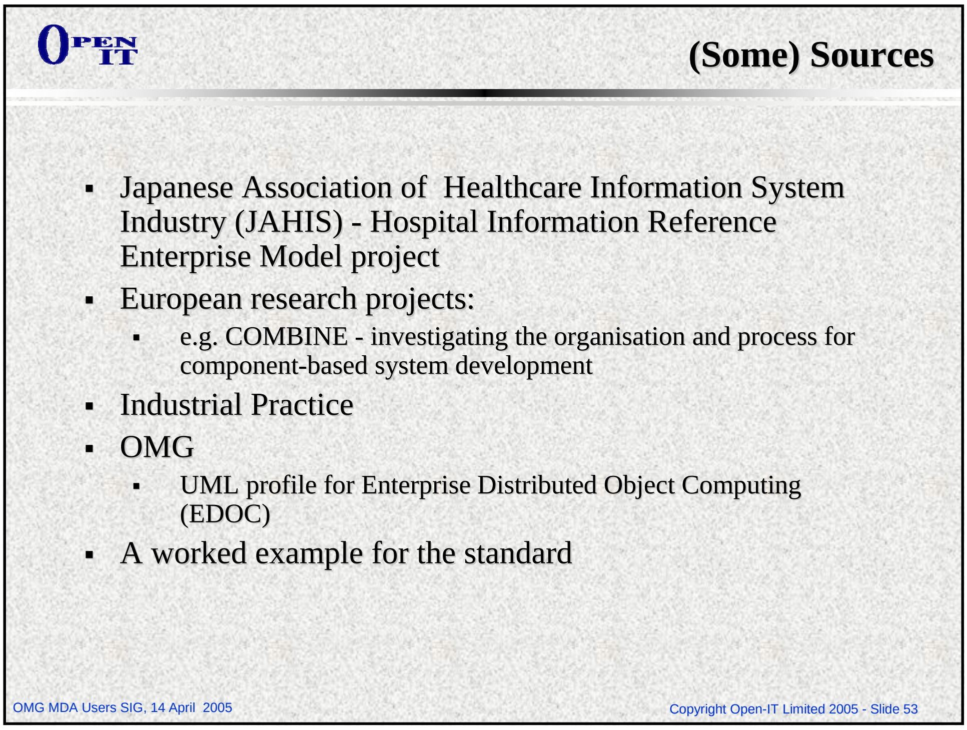
Note: Terms with "*" are from MDA Guide

(from draft for ISO/IEC 19793)

ODP and MDA together offer

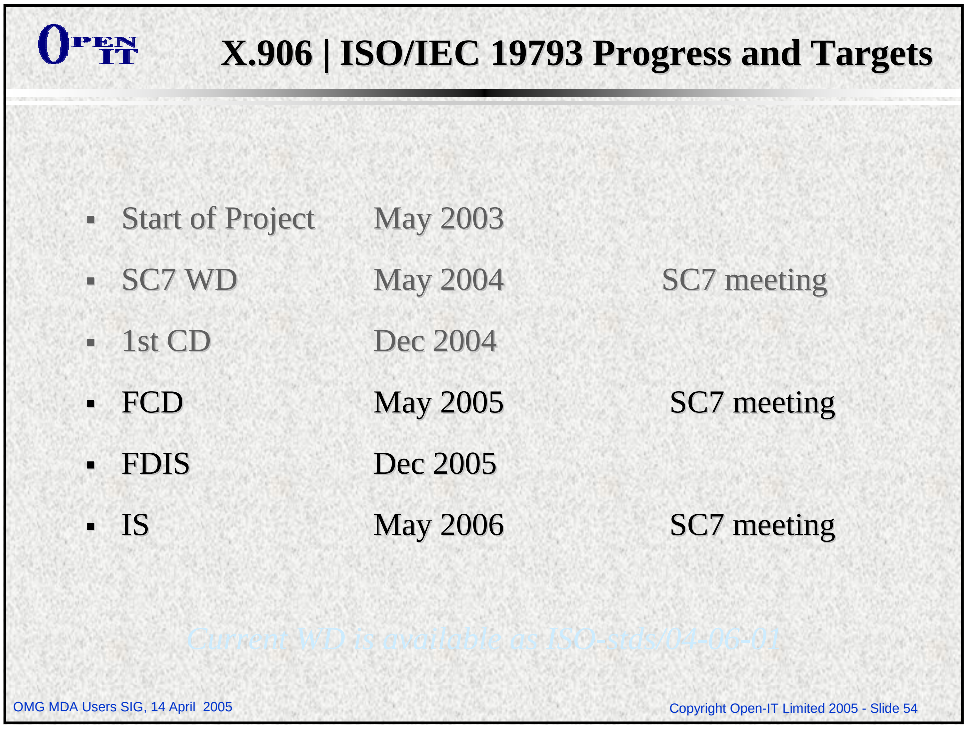
An IT based approach to system development that provides a framework for:

- separating and integrating different system concerns
- combining skills and experience
- assigning responsibilities
- automating development



(Some) Sources

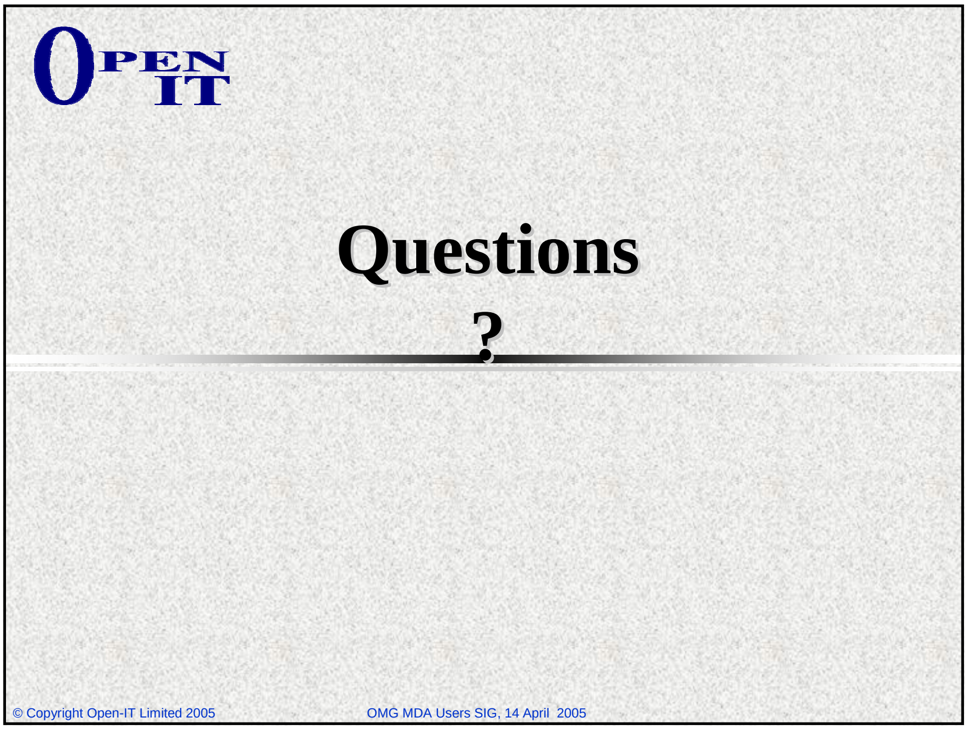
- Japanese Association of Healthcare Information System Industry (JAHIS) - Hospital Information Reference Enterprise Model project
- European research projects:
  - e.g. COMBINE - investigating the organisation and process for component-based system development
- Industrial Practice
- OMG
  - UML profile for Enterprise Distributed Object Computing (EDOC)
- A worked example for the standard



X.906 | ISO/IEC 19793 Progress and Targets

▪ Start of Project	May 2003	
▪ SC7 WD	May 2004	SC7 meeting
▪ 1st CD	Dec 2004	
▪ FCD	May 2005	SC7 meeting
▪ FDIS	Dec 2005	
▪ IS	May 2006	SC7 meeting

Current WD is available as ISO-stds/04-06-01



Questions ?