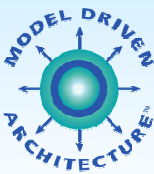




Introduction à l'Ingénierie des Modèles






Crédit :
Richard M. Soley, Ph.D., OMG Chairman
et Jean Bézivin, IRIN Université de Nantes
Daniel Petisme, Michelin
David Hill, ISIMA – Université Blaise Pascal

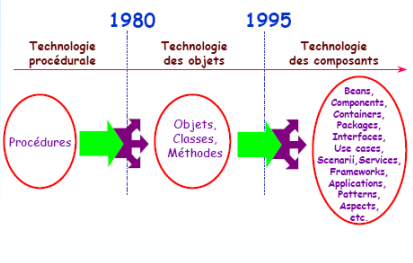






Constat d'Hétérogénéité permanente

- Langages de programmation
 - ~3 millions de programmeurs COBOL
 - ~1.6 million de programmeurs VB
 - ~1.1 million de programmeurs C/C++
- Systèmes d'Exploitation
 - Unix, MVS, VMS, MacOS, OS X, Windows (8 versions!),
 - Windows 3.1: est toujours utilisé !
 - Dispositifs embarqués (mobile, Palm)
- Réseaux
 - Ethernet, ATM, IP, SS7, Firewire, USB
 - Bluetooth, 802.11b, HomeRF








Des modèles de complexité croissante

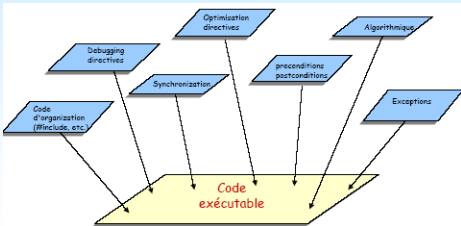






Introduction de la programmation par aspects

« We have found many programming problems for which **neither procedural nor object-oriented programming techniques are sufficient to clearly capture some of the important design decisions** the program must implement. This forces the implementation of those design decisions to be **scattered** through-out the code, resulting in **"tangled"** code that is excessively difficult to develop and maintain. We present an analysis of why certain design decisions have been so difficult to clearly capture in actual code. We call the properties these decisions address **aspects**, and show that the reason they have been hard to capture is that they **cross-cut** the system's basic functionality. We present the basis for a new programming technique, called **aspect-oriented programming**, that makes it possible to clearly express programs involving such aspects including appropriate isolation, composition and re-use of the aspect code. » from Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, John Irwin. (ECOOP June 1997).

AOP : Le code comme référentiel unique



Où pouvons nous être d'accord ?

- Il n'y aura pas de consensus sur les plates-formes matérielles
- Il n'y aura pas de consensus sur les systèmes d'exploitation
- Il n'y aura pas de consensus sur les protocoles réseau
- Il n'y aura pas de consensus sur les langages de programmation
- Il doit y avoir un consensus sur les interfaces et l'interopérabilité !**




La mission de l'OMG depuis 1989

- Développer une architecture, utilisant la technologie des objets pour l'intégration d'applications distribuées garantissant :
 - La réutilisabilité des composants
 - Interopérabilité & la portabilité
 - L'emploi des logiciels commercialement disponibles
- Les spécifications doivent être *gratuitement* disponibles
- Des implémentations doivent exister
- Contrôlé par des membres « not-for-profit »



Quelques entreprises de l'OMG

AT&T	Glaxo SmithKline	Microsoft	Rational
BEA	Hewlett Packard	MITRE	SAGA Software
Borland	Hitachi	MSC Software	SAP
Boeing	Hyperion	NASA	SAS Institute
CA	IBM	NEC	Secant
Citigroup	IONA	NetGenics	Siemens
Compaq	io Software	NTT	Sprint
Ericsson	Kabira	OASIS	Sun
Ford	Kennedy Carter	Oracle	Unisys
Fujitsu	John Deere	Pfizer	Vertel



Les succès principaux de l'OMG

- Common Object Request Broker Architecture
 - CORBA® reste le seul langage et la seule plate-forme neutre pour l'interopérabilité...
- Unified Modeling Language
 - UML™ reste le seul langage mondial de modélisation standardisé
- Common Warehouse Metamodel
 - CWM™, intégration des deux dernières initiatives mondiales d'entrepôt de données.
- Meta-Object Facility
 - MOF™, standard de méta-modélisation UML
- XML Metadata Interchange
 - XMI™, le standard XML-UML



Malgré tout la diversité continue...

- Le Middleware a proliféré :
 - CORBA®: Vendor, OS & language independent middleware
 - COM/DCOM/MTS (Microsoft Transaction Server)
 - Java/EJB
 - XML/SOAP
 - C#.Net
 - Quel sera le prochain intergiciel de pointe ???
- Constat : Il faut préserver son investissement logiciel malgré les changements d'infrastructure



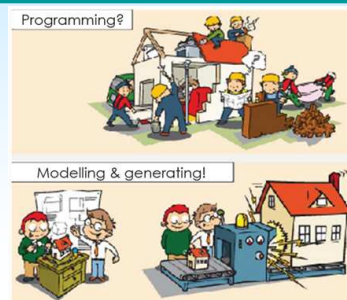
Comment protéger l'investissement logiciel ?


- De nombreux problèmes demeurent
 - Être à l'affût de la dernière technologie pour étudier son impact.
 - Protéger son investissement en s'appuyant sur la base logicielle existante
 - Garder le personnel qualifié
 - Maintenir le code existant
 - ...
- Intégrer ce que vous avez développé
 - Avec ce que vous développerez !



Une solution possible...


L'Ingénierie Générative





OMG


OBSERVATORY OF MANAGEMENT GENEVA

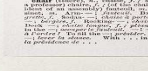


ISIMA

« Everything is a model ? » L'Ingénierie Dirigée par les Modèles


« One and three chairs »
Joseph Kossuth, 1965






« A model is a simplification of a system built with an intended goal
in mind. The model should be able to answer questions in place of the
actual system»

Bézibin & Gerbé, 2001



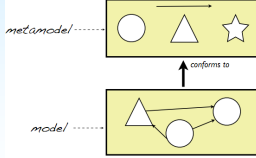
OMG
UNIVERSITY MANAGEMENT OF KNOWLEDGE



ISIMA

Métamodéliser ?

Layer	Description	Example
meta- metamodel	The infrastructure for a meta modeling architecture. Defines the language for specifying metamodels.	Metaclass, MetaAttribute, MetaOperation
Metamodel	An instance of a metamodel. Defines the language for specifying a model.	Class, Attribute, Operation, Component
Model	An instance of a metamodel. Defined a language to describe an information domain.	StockShare, askPrice, sellLimitOrder, StockQuoteServer
user objects (user data)	An instance of a model. Defines a specific information domain.	<pre> 654.56 sell_limit_order <Stock_Quote_srv_3 2123> </pre>





The diagram illustrates the Eclipse Modeling Framework (EMF) and Epsilon architecture. At the top left is the OMG (Object Management Group) logo, and at the top right is the ISIMA logo. The main title is "Eclipse Modeling Framework & Epsilon". Below the title, the EMF logo (a globe with a grid) is shown next to the text "emf" and "© ECLIPSE MODELING FRAMEWORK". To the right, the Epsilon logo (a globe with a grid) is shown next to the text "epsilon". A blue arrow points from the Epsilon logo to the EMF logo. Below the EMF logo is a box labeled "Ecore". Below the Epsilon logo is a box labeled "Epsilon Object Language". Below the "Epsilon Object Language" box are four boxes: "ETL" (orange), "EVL" (white), "EWL" (white), and "EGL" (white). Arrows point from the "Epsilon Object Language" box to each of these four boxes. Below the "ETL" box is the text "Epsilon Transformation L.". Below the "EVL" box is the text "Epsilon Generation L.". Below the "EWL" box is the text "Epsilon Validation L.". Below the "EGL" box is the text "Epsilon Wizard L.". Below the "Ecore" box is a quote: "The EMF project is a modeling framework and code generation facility for building tools and other applications based on a structured data model."

```

graph TD
    SM[Source model] -- Type --> SMM[Source metamodel]
    TM[Transformation model] --> TMM[Transformation metamodel]
    TModel[Target model] --> TMM
    TMM --> Ecore[Ecore]
    SMM --> Ecore
    Ecore --> SMM
    Ecore --> TMM
    Ecore --> TModel
    SM -- "Source-target execution" --> TModel
  
```

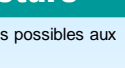
The diagram illustrates the M2M transformation process. It shows a hierarchy where Source, Transformation, and Target models are transformed into their respective metamodels. These metamodels then interact with an Ecore metamodel. A red arrow labeled "Source-target execution" points from the Source model to the Target model.

The slide content includes the OMG logo at the top left, the title "Introduction à MDA" in large blue font, the subtitle "Model Driven Architecture" in large blue font, and the credits "Crédit : Richard M. Soley, Ph.D., OMG Chairman et Jean Bézivin, IRIN Université de Nantes David Hill, ISIMA" in smaller black font.

L'architecture MDA – Model Driven Architecture

- L'architecture MDA de l'OMG est une des réponses possibles aux problèmes du Génie Logiciel.
- C'est une possibilité d'appréhender les développements nouveaux en intégrant les atouts et le savoir faire de l'entreprise
- Elle se base sur les standards de l'Industrie qui sont en mesure de concevoir les applications d'aujourd'hui et de demain
- Le MDA aide concrètement à intégrer l'hétérogénéité actuelle et donne une architecture capable d'intégrer l'inattendu.
- Permet une intégration en douceur des applications COTS (Commercial Of The Shelf)
- Les Modèles sont testables et simulables
- Le but inavoué : *Etre l'architecture logicielle des 20 prochaines années*

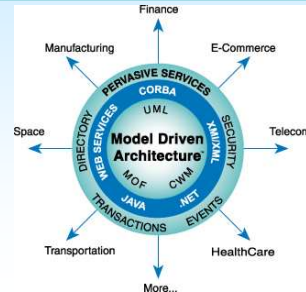


MDA : Qu'est-ce que c'est ?

- **Une nouvelle façon de spécifier et de construire les systèmes**
 - **MDA repose sur une modélisation UML**
 - L'ensemble du cycle de vie est pris en compte : analyse, conception, implementation, déploiement, maintenance, évolution & intégration avec les systèmes futurs
 - L'interopérabilité et la portabilité sont pris en compte
 - On prévoit une diminution des coûts de développement initiaux et une maximisation du ROI (Return On Investment)
- MDA s'applique directement à la « variété » :
 - Langage de programmation
 - Réseau
 - Systèmes d'exploitation
 - Middleware

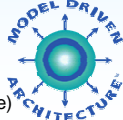


Model Driven Architecture

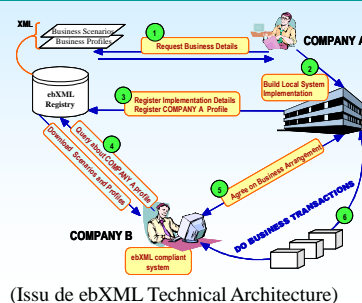


UML prend sa pleine puissance

- UML (Unified Modeling Language) est le successeur d'une douzaine de notations d'Analyse et de conception produites dans les années 90.
- Un consensus à débuté en 1996 et un première proposition été adoptée et complétée en 1997
- On a ensuite ajouté le (MOF) et mes spécifications Méta de XML (XMI)
- La standardisation s'est imposée par le marché
 - Plus de 100 ouvrages
 - Des dizaines d'outils commerciaux
 - Une formation largement diffusée
- Un processus ouvert (une méthodologie ouverte) est en cours de proposition (UML 2.0 process)



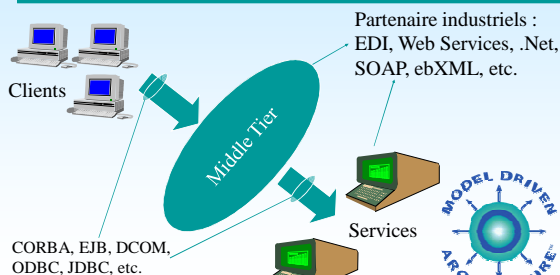
Le rêve : les Web Services



(Issu de ebXML Technical Architecture)



La réalité : l'Intégration



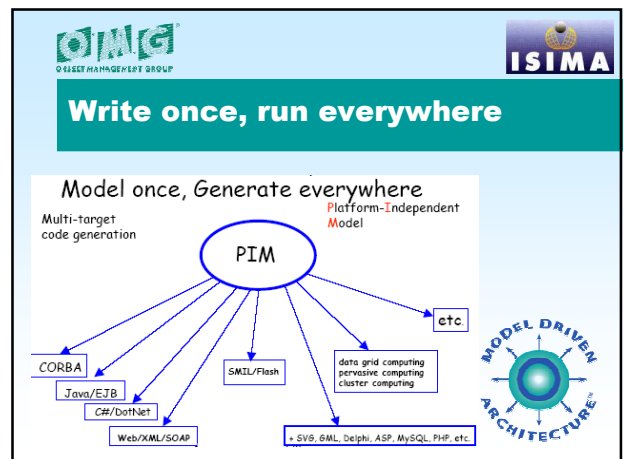
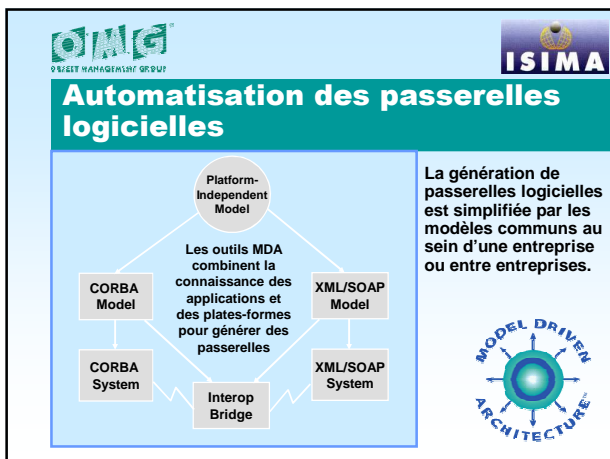
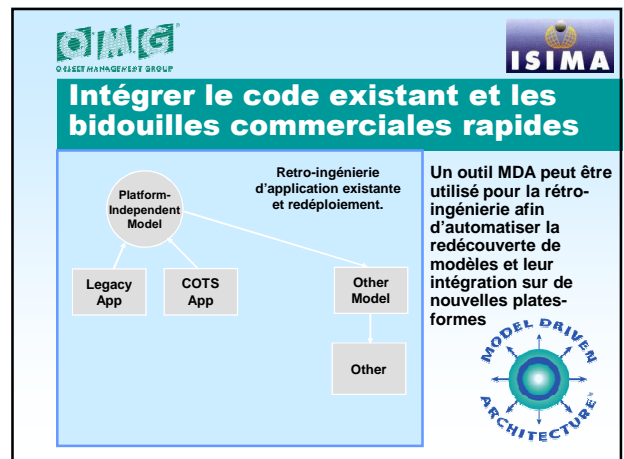
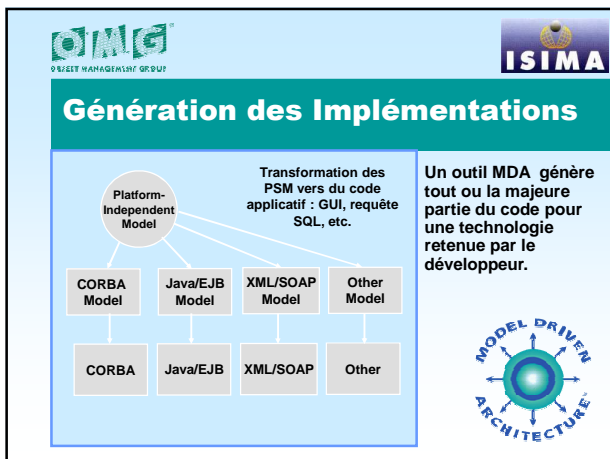
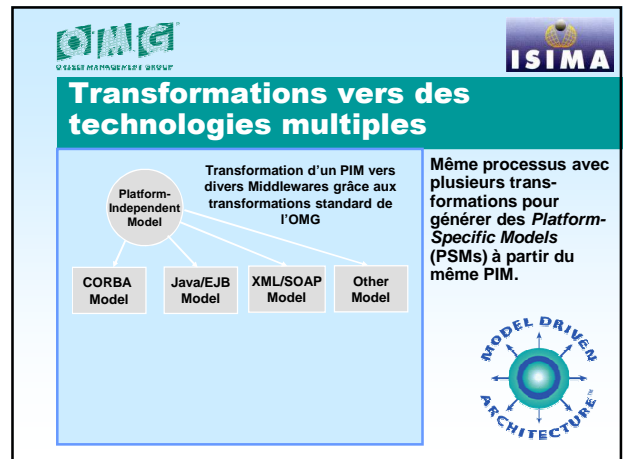
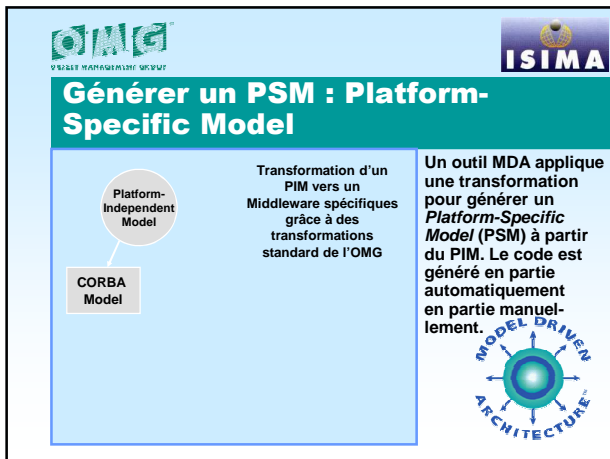
La construction d'une application MDA

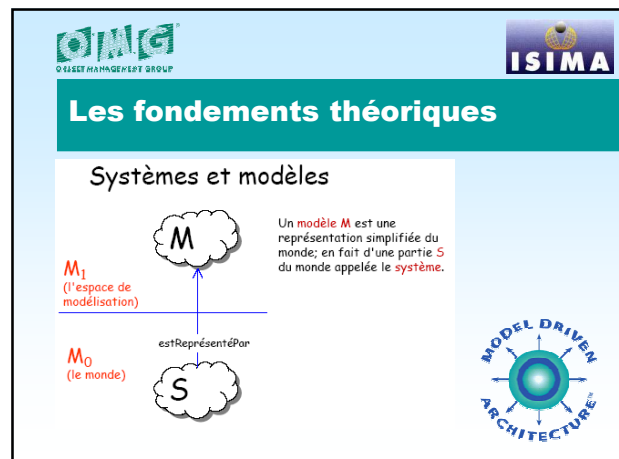
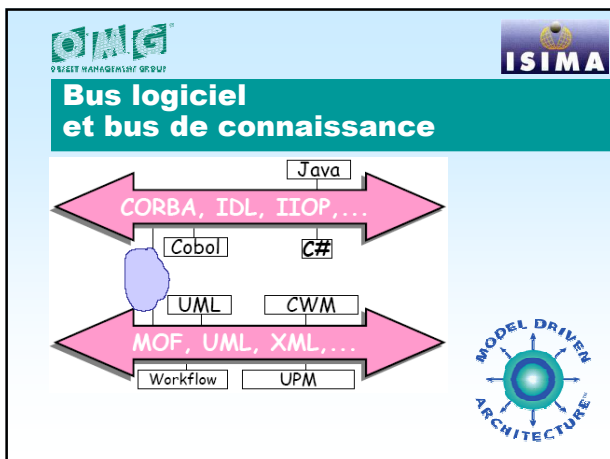
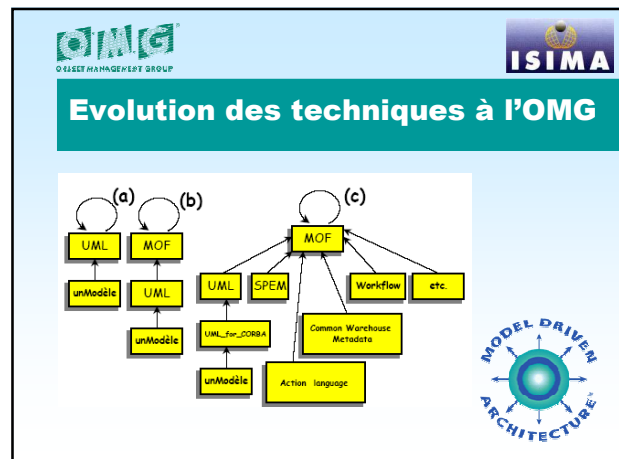
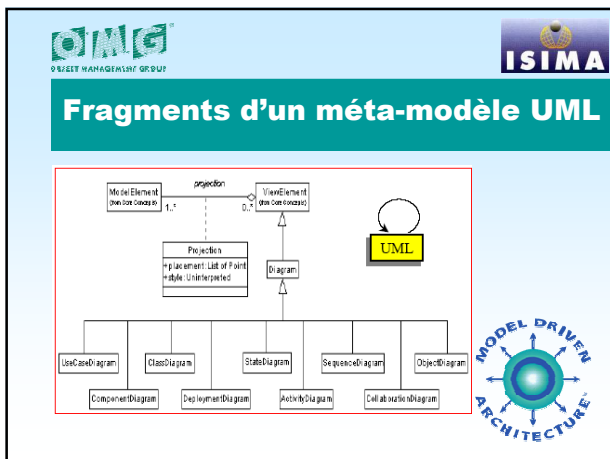
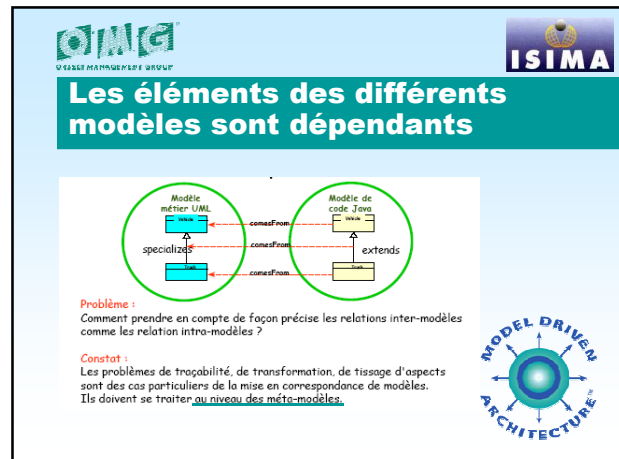
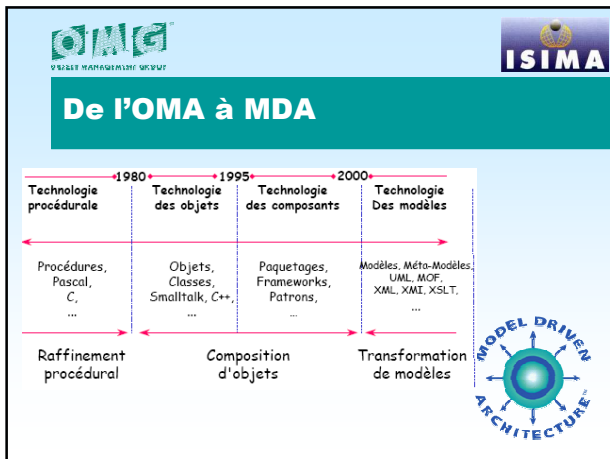
Platform-Independent Model

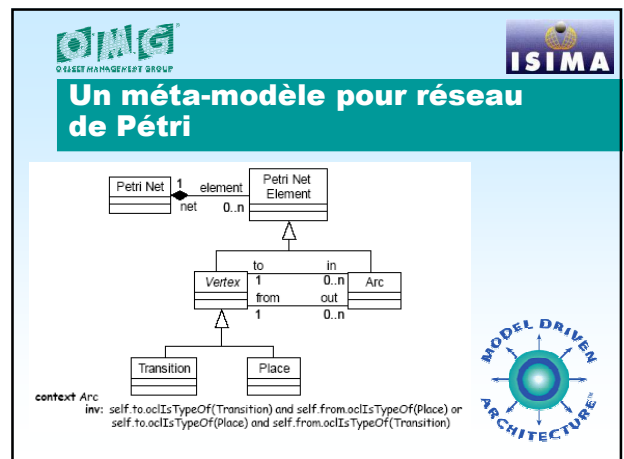
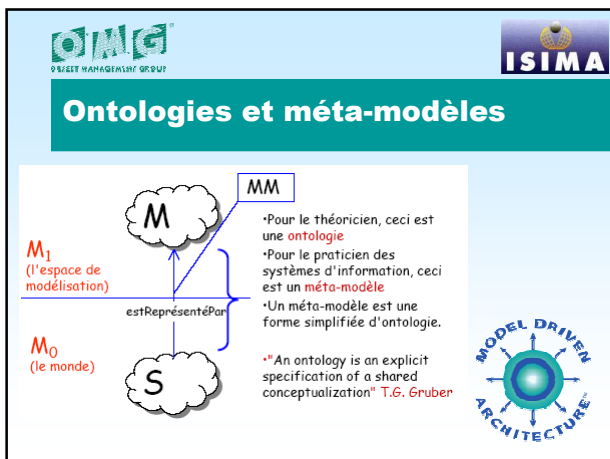
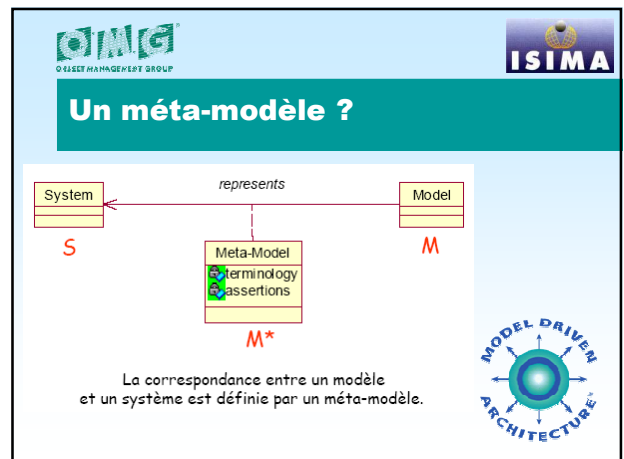
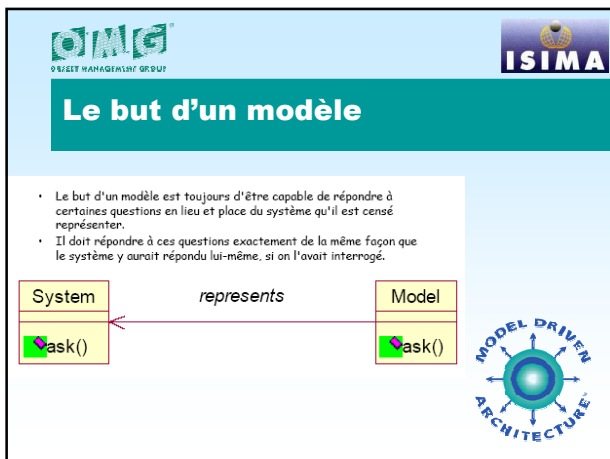
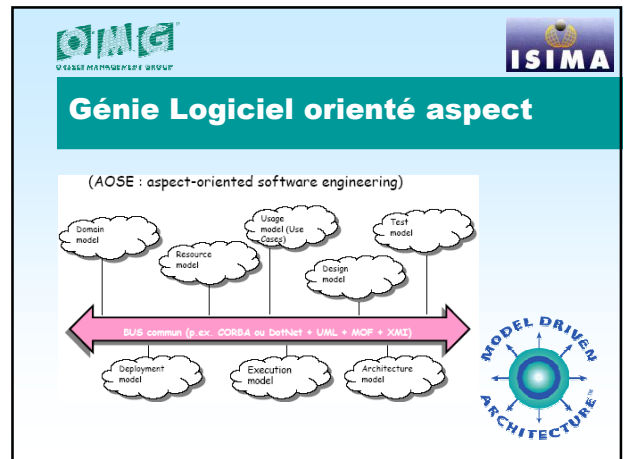
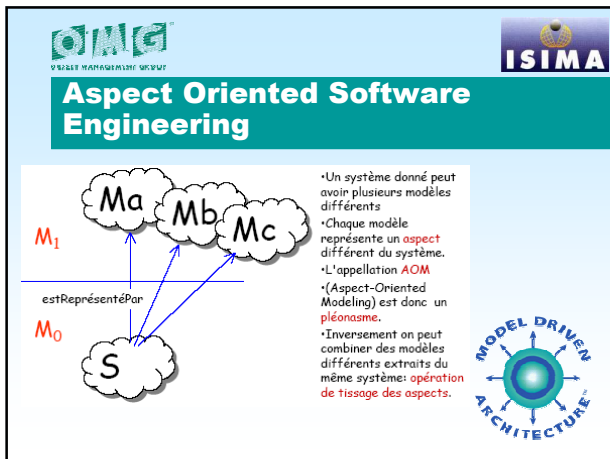
Un modèle détaillé, précisant les pré et post conditions en OCL (Object Constraints Language), et la sémantique dans un langage exécutable

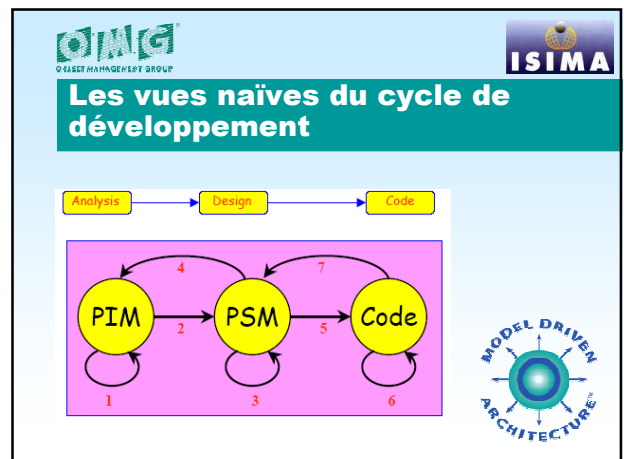
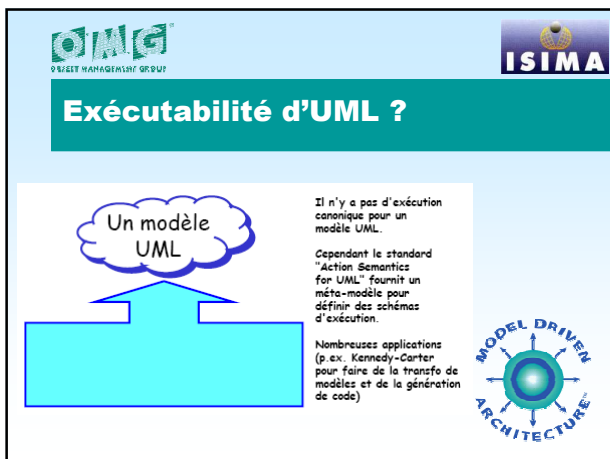
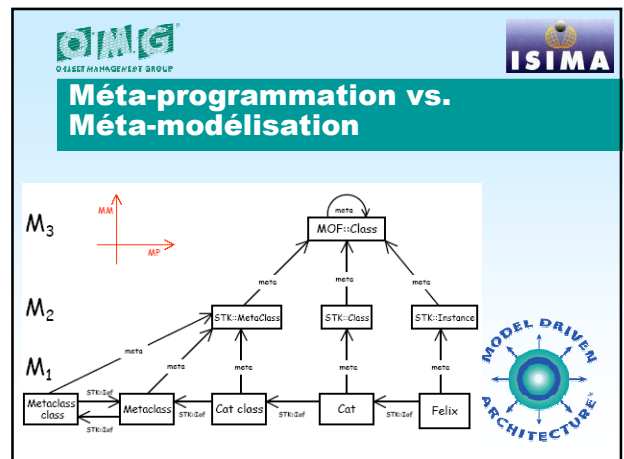
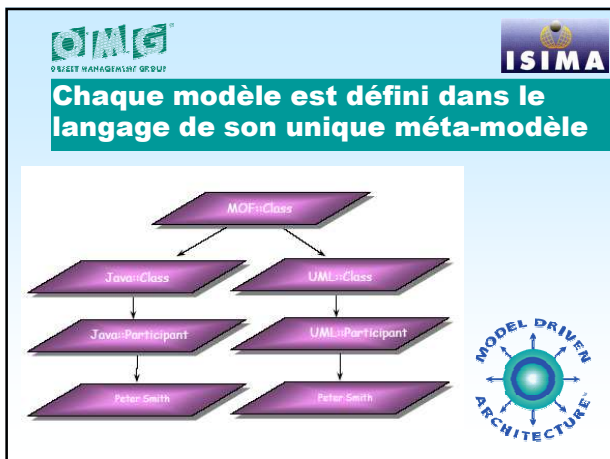
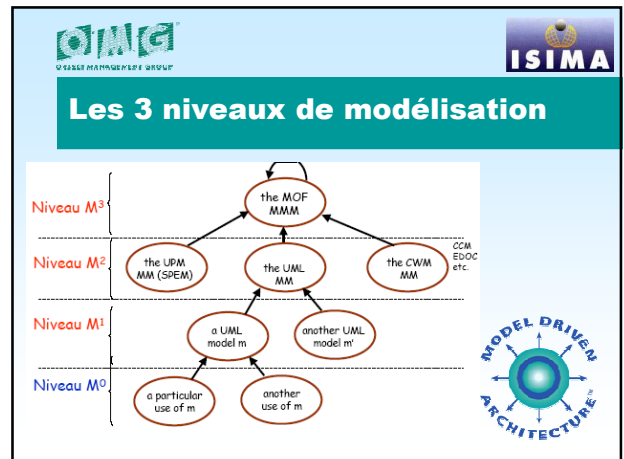
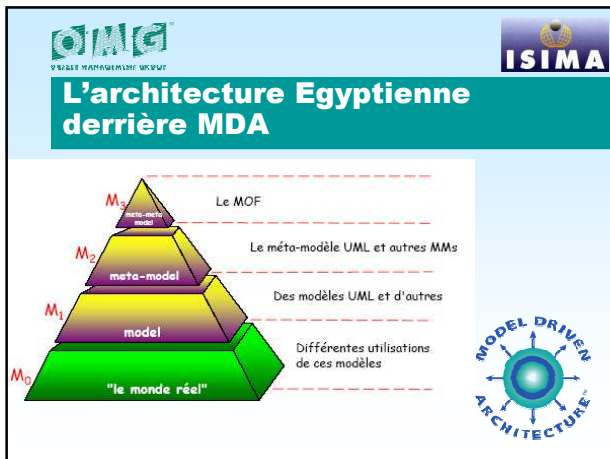
On commence avec un **Platform-Independent Model (PIM)** représentant les fonctionnalités « métier » sans les déformer par des détails techniques

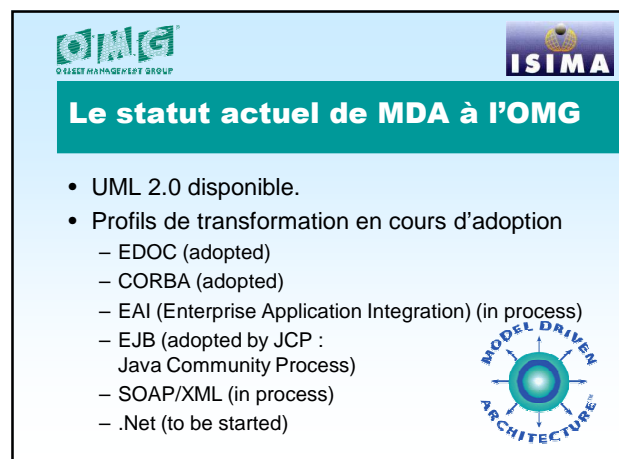
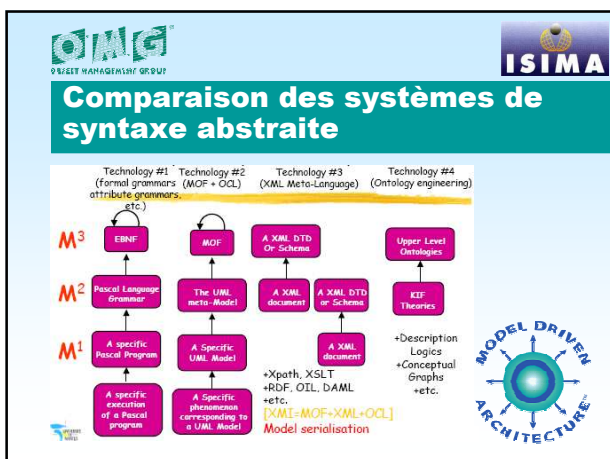
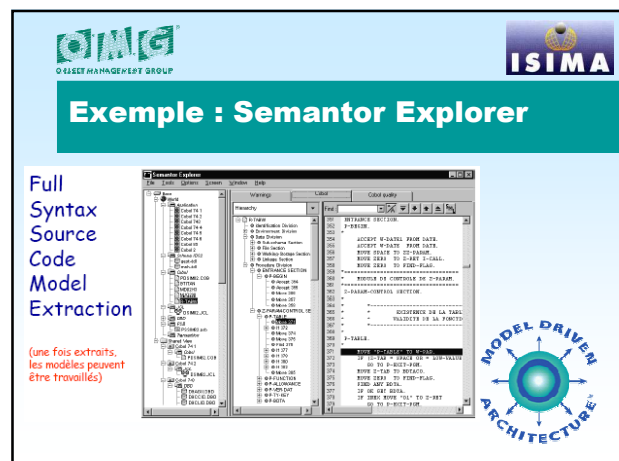
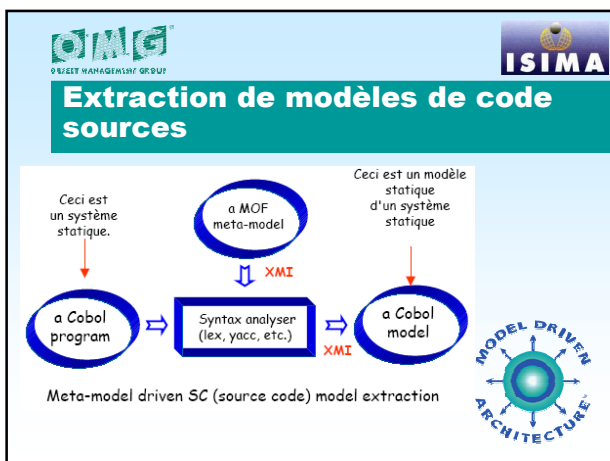
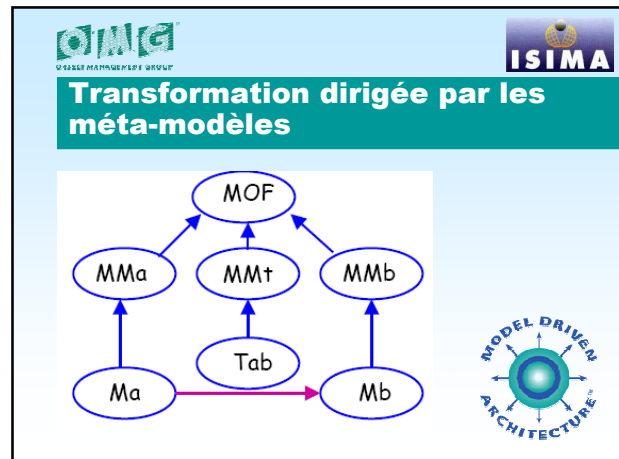
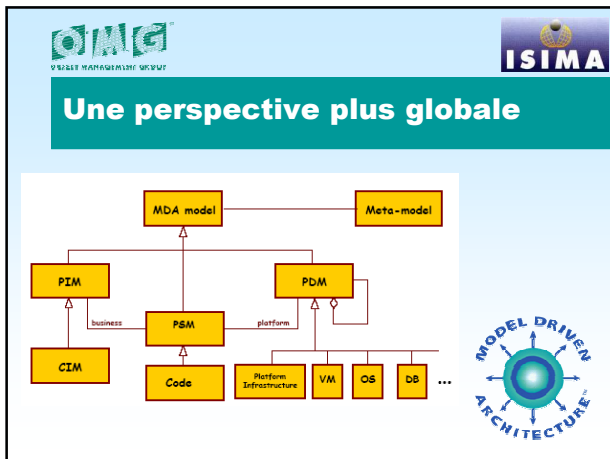












Les groupes de marché intéressés

- De nombreux groupes de marché ont commencé à travailler sur l'approche MDA
 - Le commerce électronique
 - Les services financiers
 - La domaine de la santé
 - La recherche pour les sciences de la vie
 - La productique
 - L'aéronautique
 - Télécommunications



MDA en Pratique

- Quelques preuves de faisabilité :
 - Wells Fargo (une architecture qui a résisté à une dizaine d'année de changements)
 - Lockheed Martin Aeronautics
 - Logiciels au sein du gouvernement US
- Ce sont des exemples « MDA-like »
 - Les standards apporteront la portabilité



Des implémentations !

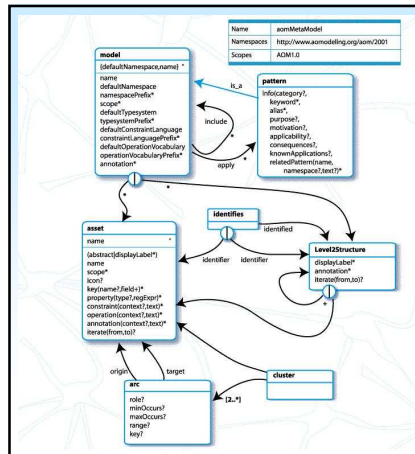
Interactive
Objects

KENNEDY CARTER

Kabira
TECHNOLOGIES, INC.

SECANT

- MDA Information Page
 - <http://www.omg.org/mda/>
- OMG General Information
 - <http://www.omg.org/>

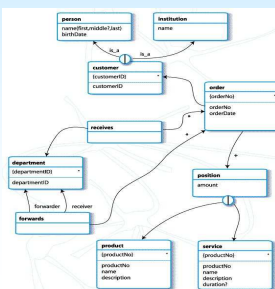


Asset
Oriented Modeling

www.aomodeling.org/



Exemple AOM : Niveau 1



<http://www.aomodeling.org/tools.htm>

KLEEN : CASE Tool

