

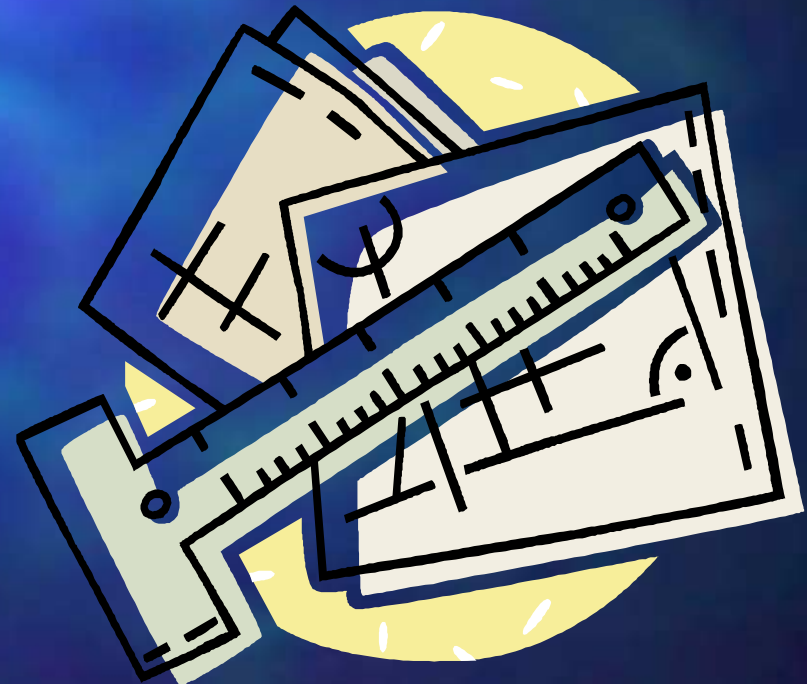
Modeling All the Way Up...

Modeling All the Way Down

*Richard Mark Soley, Ph.D.
Chairman and CEO*

People Share Design with Models

- Models—abstractions—are ancient in
 - Engineering
 - Architecture
 - Ship-building
 - Traffic control
 - Workflow
 - Maintenance
 - ...





People Talk to Computers with Symbols

```
while (x < 10) {  
    printf (stdout, array[x]);  
    X++;  
}
```

What's wrong with this (non)-picture?

Let's look at Enterprise IT as an example.



Enterprise IT Must Deal With

■ Business Factors:

- Defining & meeting changing business requirements
- Complex & changing business processes
- Shifting enterprise/application boundaries
- Semantic integration with customers, supplies & partners

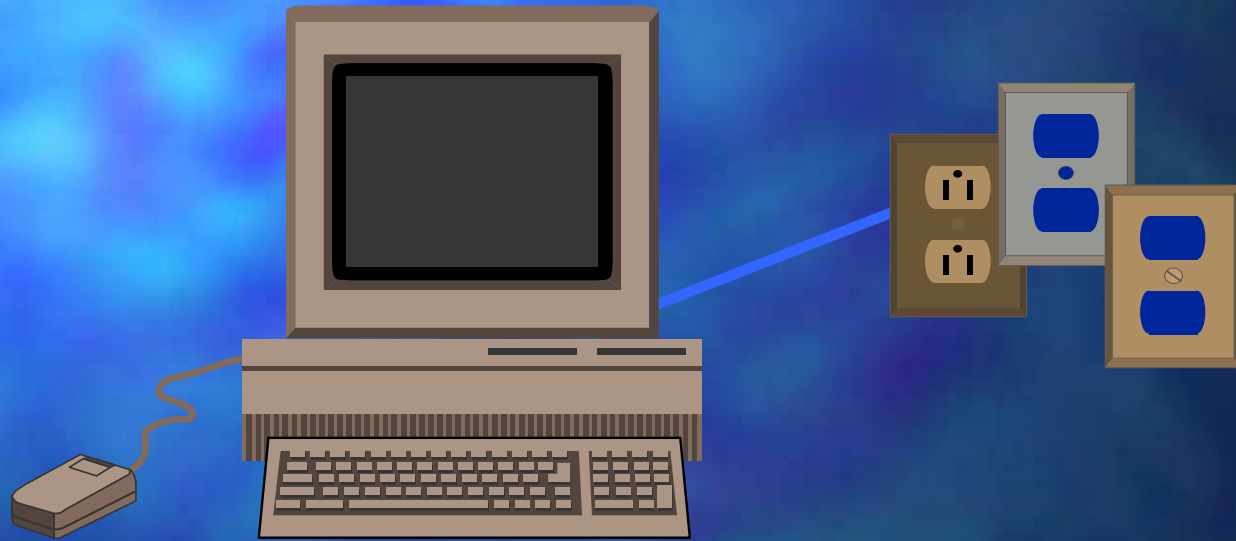
■ Technological Factors:

- Barriers to interoperability & integration
- Development & maintenance obstacles
- Evolving & unstable technology suites





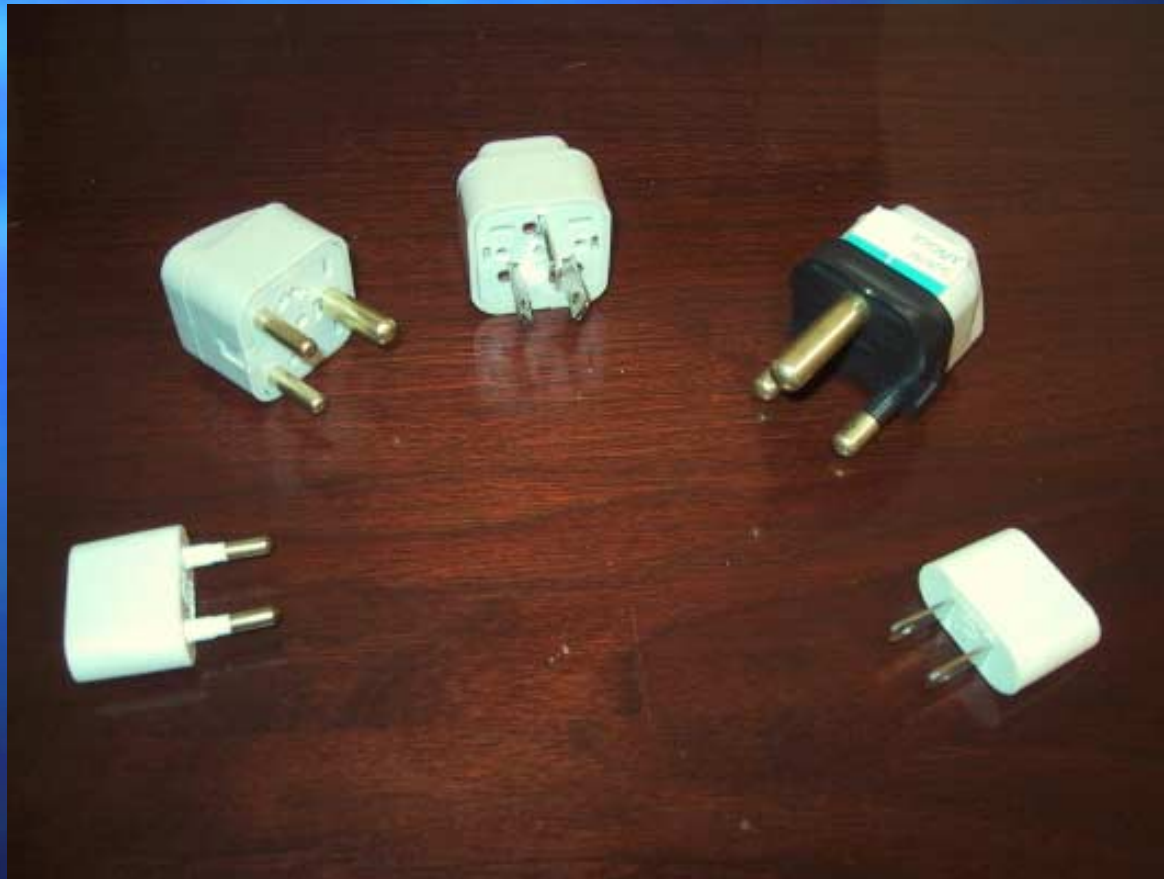
What is the *real* integration issue?



The Global Information Appliance



Not too bad for electrical power





...but a mess for telephony!

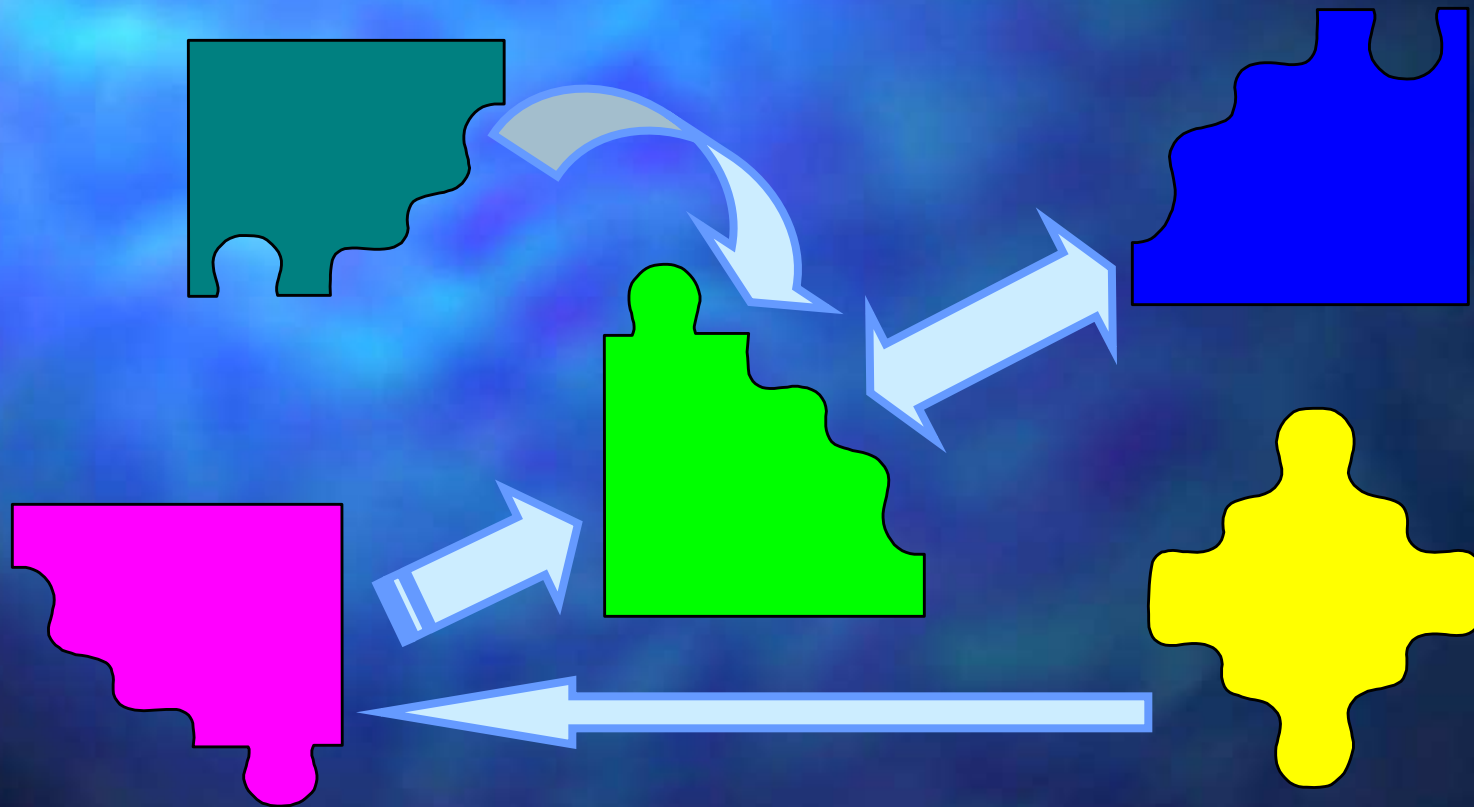




Heterogeneity is Permanent

- Programming languages
 - ~3 million COBOL programmers
 - ~1.6 million VB programmers
 - ~1.1 million C/C++ programmers
- Operating systems
 - Unix, MVS, VMS, MacOS, Windows (all 8!), PalmOS...
 - Windows 3.1: it's still out there!
 - Embedded devices (mobile, set-top, etc.)
- Networks
 - Ethernet, ATM, IP, SS7, Firewire, USB
 - Bluetooth, 802.11b, HomeRF

The integration picture is always changing



Executive decisions, mergers & acquisitions have a way of surprising us...



Roadkill on the Info Highway

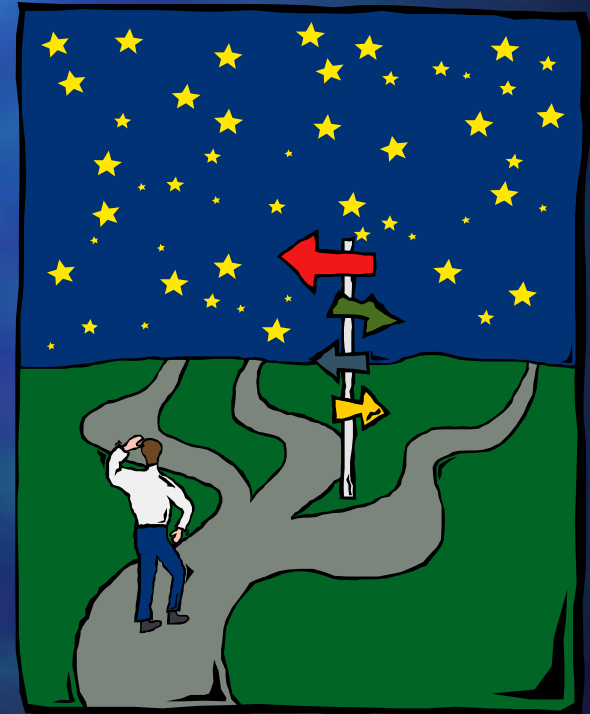


...and doing things “the way we always do them” isn’t the answer.



How Can We Deal with This?

*Make adaptability the design center
for your architecture.*





Modeling is the Focus

- Modeling, especially graphical modeling is
 - A natural human approach to design
 - Thousands of years old
 - Allows expression of design separate from implementation, as implementations change
 - Allows for long-term maintenance & integration
 - Is an *accelerator* of implementation
 - Is technology-independent

18th century B.C. multiplication table





OMG's Mission Since 1989

- Develop an architecture, using appropriate technology, for modeling & distributed application integration, guaranteeing:
 - reusability of components
 - interoperability & portability
 - basis in commercially available software
- Specifications *freely available*
- Implementations exist
- Member-controlled not-for-profit



Who Are OMG?

Adaptive	Fair, Isaac	Kaiser Permanente	PRISM
BEA	Fujitsu	MEGA International	SAP
Borland	General Electric	MITRE	SAS Institute
Boeing	Hewlett Packard	NASA	Select
Business Rules Group	Hitachi	NEC	Siemens
Business Semantics	IBM	NIST	Softeam
CA	ILOG	NTT DoCoMo	Sun
Citigroup	Inferware	Northrop Grumman	Unisys
DaimlerChrysler	IONA	OASIS	Visa
EDS	Kennedy Carter	Oracle	W3C





OMG's Best-Known Successes

- Common Object Request Broker Architecture
 - CORBA® remains the only language- and platform-neutral interoperability standard
- Unified Modeling Language
 - UML™ remains the world's only standardized modeling language
- Common Warehouse Metamodel
 - CWM™, the integration of the last two data warehousing initiatives
- Meta-Object Facility
 - MOF™, the repository standard
- XML Metadata Interchange
 - XMI™, the XML-UML standard





Protecting Software Investment

- The problem remains
 - Tracking the *next best thing*, retaining staff;
 - Protecting your investment in existing software base;
- Integrating what you've built,
 - With what you're building,
 - *With what you will build!*
- Architectures ought to be
 - Stable descriptions lasting decades
 - Capable of communicating the designers' *vision*
 - Testable, simulatable, *executable*



The Model Driven Architecture

- OMG's *Model Driven Architecture* (MDA™) initiative is aimed precisely at this problem
- You have an opportunity to increase your bottom line by *integrating your assets*
- Industry standards support that goal by future-proofing your application design
- The MDA will help you integrate the mix you have today, and give you an architecture to support the unexpected
- Focus on integrating legacy applications
- Ensure smooth integration of COTS applications
- Models are *testable* and *simulatable*
- The aim: *a 20-year software architecture*



What is Model Driven Architecture?

- A Better Way to Specify and Design & Develop
 - *Based on modeling standards like UML, MOF*
 - *Is extensible to all modeling problems*
 - Supports full lifecycle: analysis, design, implementation, deployment, maintenance, evolution & integration with later systems
 - Builds in Interoperability and Portability
 - Lowers initial cost and maximizes ROI



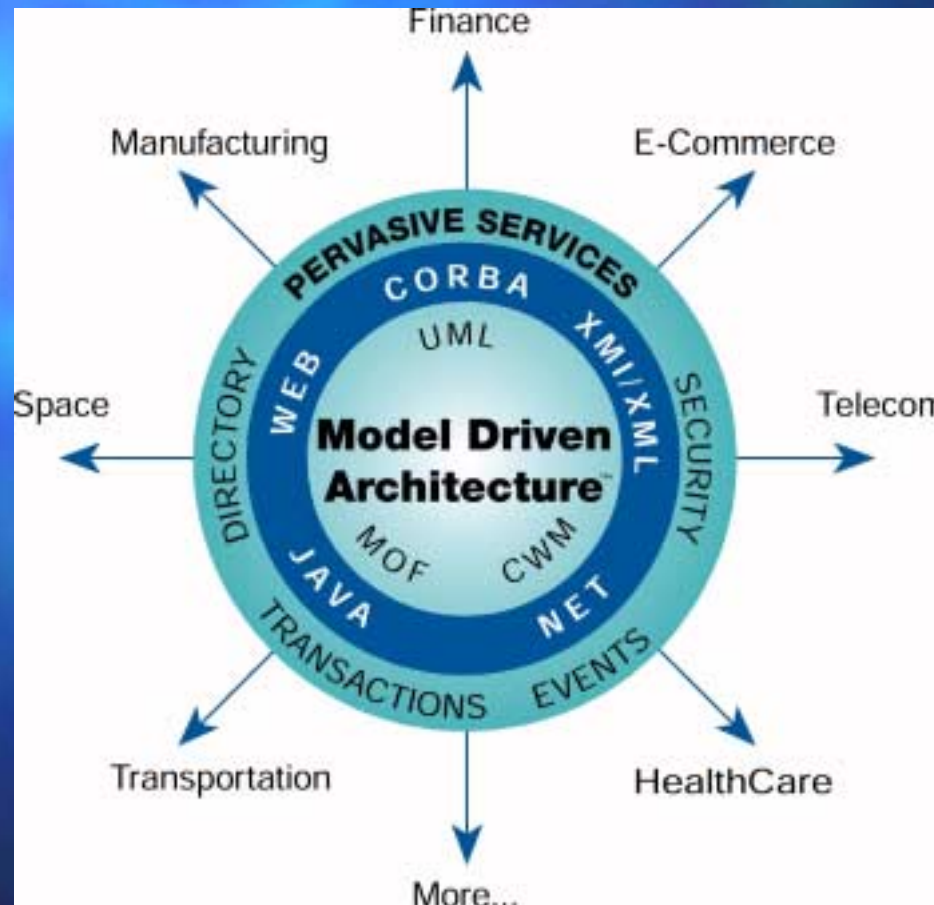


Modeling: Key Concepts

- Emphasis on *transformation techniques*
 - Based on a standard metamodeling framework; there will be many metamodels, and plenty of modeling languages (including UML)
 - Clear semantics, expressed consistently
 - Potentially many levels of abstraction
- *Enduring architectures* are the focus
 - Maintenance and integration aren't pretty, but they are the main job of IT
- Graphical languages as well as textual ones
 - Some generic, some domain-specific, just like the textual language world



Model Driven Architecture





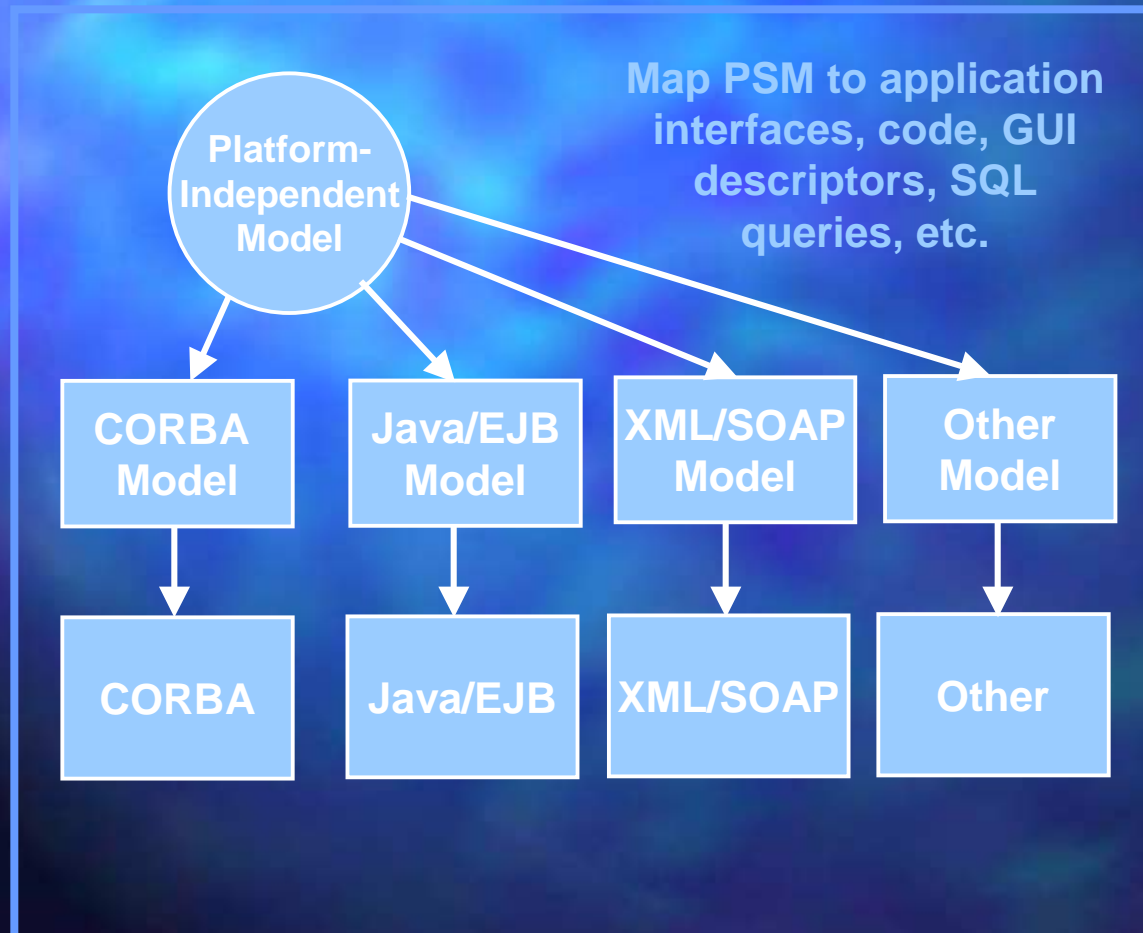
Leveraging UML is Critical

- The Unified Modeling Language is the successor to the dozens of OO A&D notations of the early '90s
- Result of an OMG standardization completed in '97
- Complemented with metadata (MOF) and XML interoperability specifications (XMI)
- Venture-neutral worldwide certification easily available
- Standardization primed the market
 - Hundreds of books
 - Dozens of commercial tools
 - Widely available training
- Supported by an open process
 - UML 2.0 updates came from 54 companies





Generating Implementations



MDA Tool generates all or most of the implementation code for deployment technology selected by the developer.

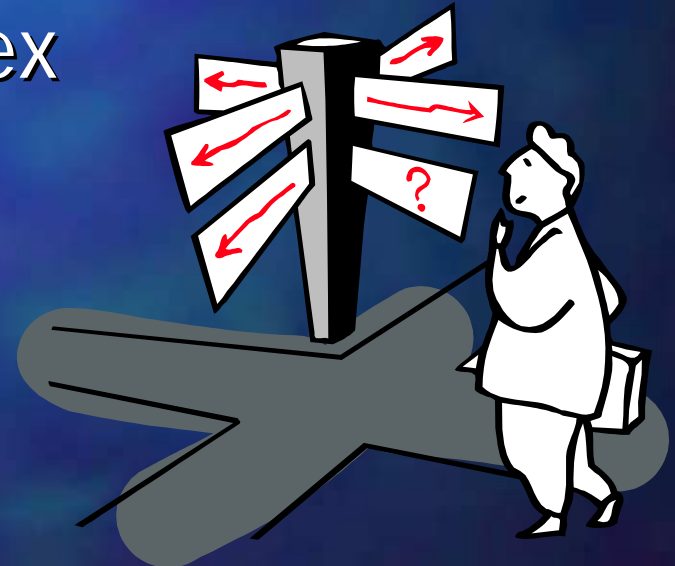


Data Integration Works Too

- MOF is the key here, behind the UML scene
- With one modeling language
 - Anything can be modeled
 - User must map (code) from domain to modeling language constructs
- MOF enables definition of multiple modeling languages
 - Specific to various domains
 - Semantics captured in metamodels
 - User no longer needs to “map”, tools do

UML Myths

- MDA is just about code generation
- MDA is just programming with UML
- MDA and DSL are different things
- UML is too big & complex





Code Generation is just one feature

- Sometimes we'll be able to generate all the
 - Code
 - Schemas
 - Deployment descriptors
- Sometimes we won't; but we'll still have the modeling values of
 - Clear, sharable graphical expression
 - Flexible transformation for agile retargeting
 - *An enduring description of the system*
- *Architecture matters (that's why MDA)*



MDA is more than UML!

- MDA is a *meta-design pattern*:
 - Discover multiple syntaxes for one semantic
 - Capture that semantic in a machine-readable model
 - Standardize the transformations to multiple syntaxes
- UML is a good general design and implementation language, but MOF provides the integration across languages



MDA and DSL are the Same!

- Domain-Specific Languages (DSL's) are a way to capture design semantics in languages closely fitted to specific problem areas (application domains)
 - Do we really want every programmer using a different language
- It pays to have a central way to query, view and transform languages (MOF)
- MDA actually offers two routes to standardized DSL's: MOF-defined languages and UML profiles

UML is a Toolbox!

- If all you've got is a hammer, every problem looks like a nail
- But if you have a whole toolbox of tools, you can choose the tool that best fits the problem
- A big toolbox is an *asset*





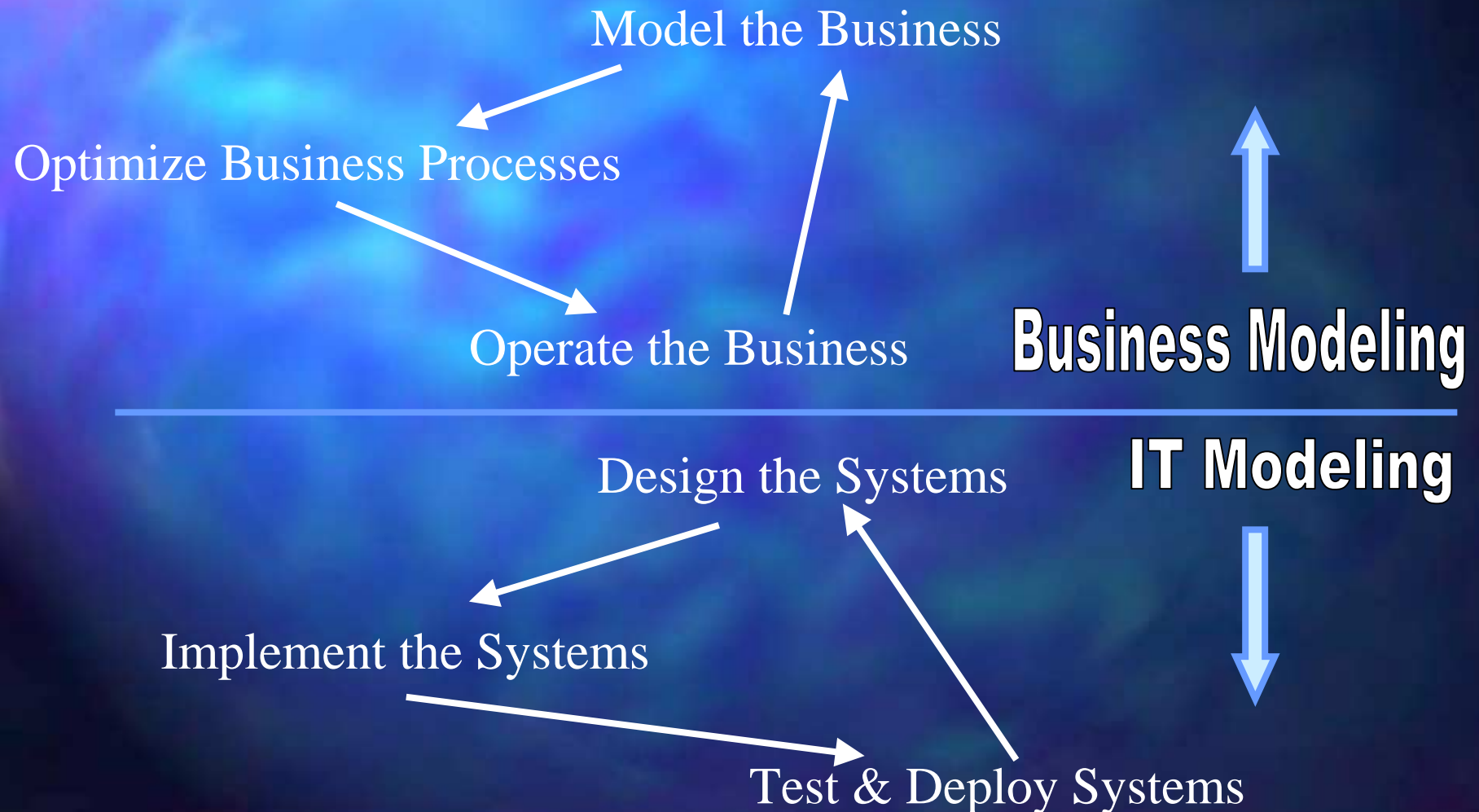
MDA is *Fractal*

- There is a general pattern to MDA:
 - *Discover* multiple syntaxes for a single semantic
 - *Derive & design* a model which underlies that semantic
 - *Develop* transformations between those models.



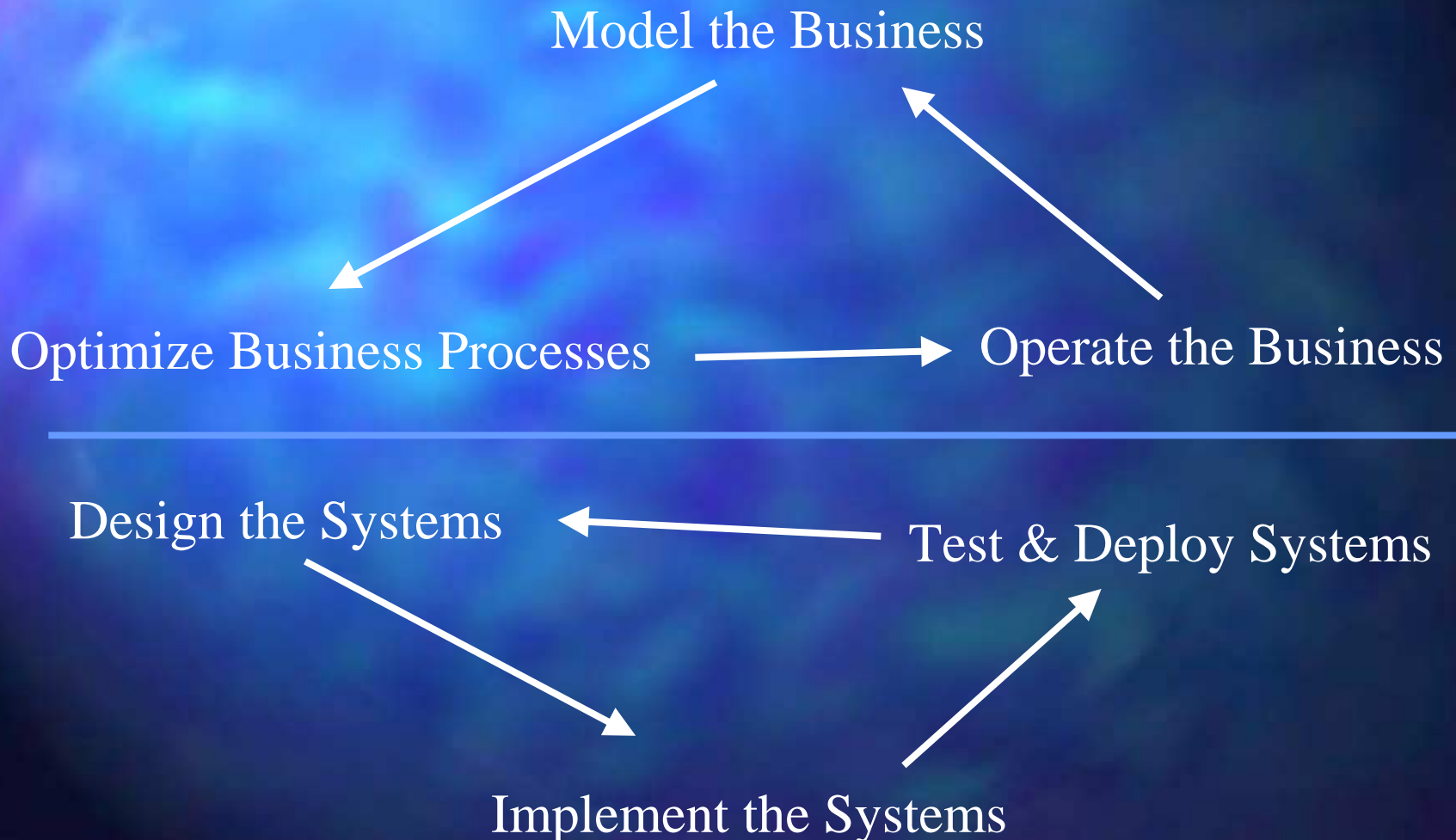


Shifting Gears: Business Modeling

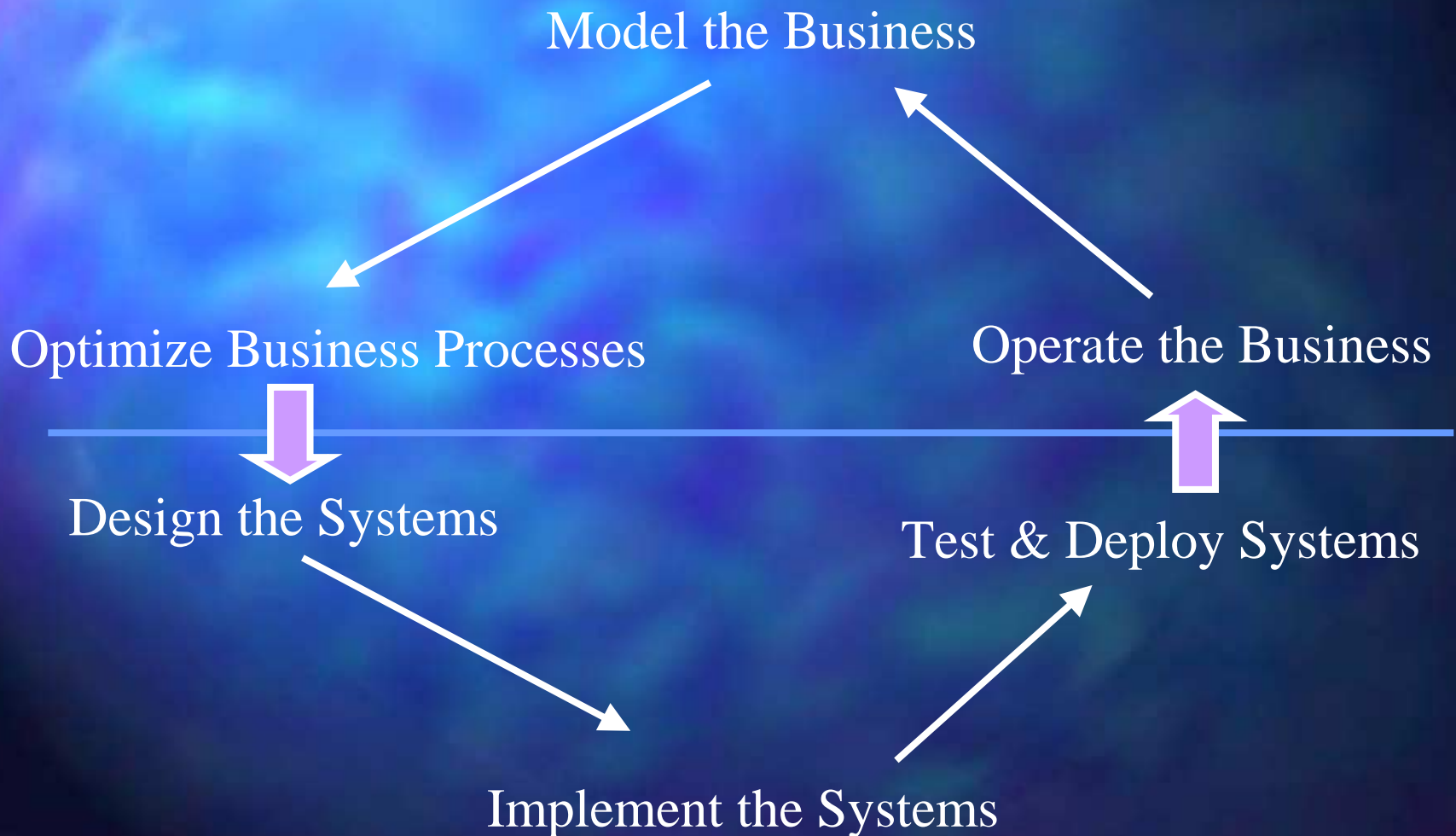




Can We Integrate Modeling?

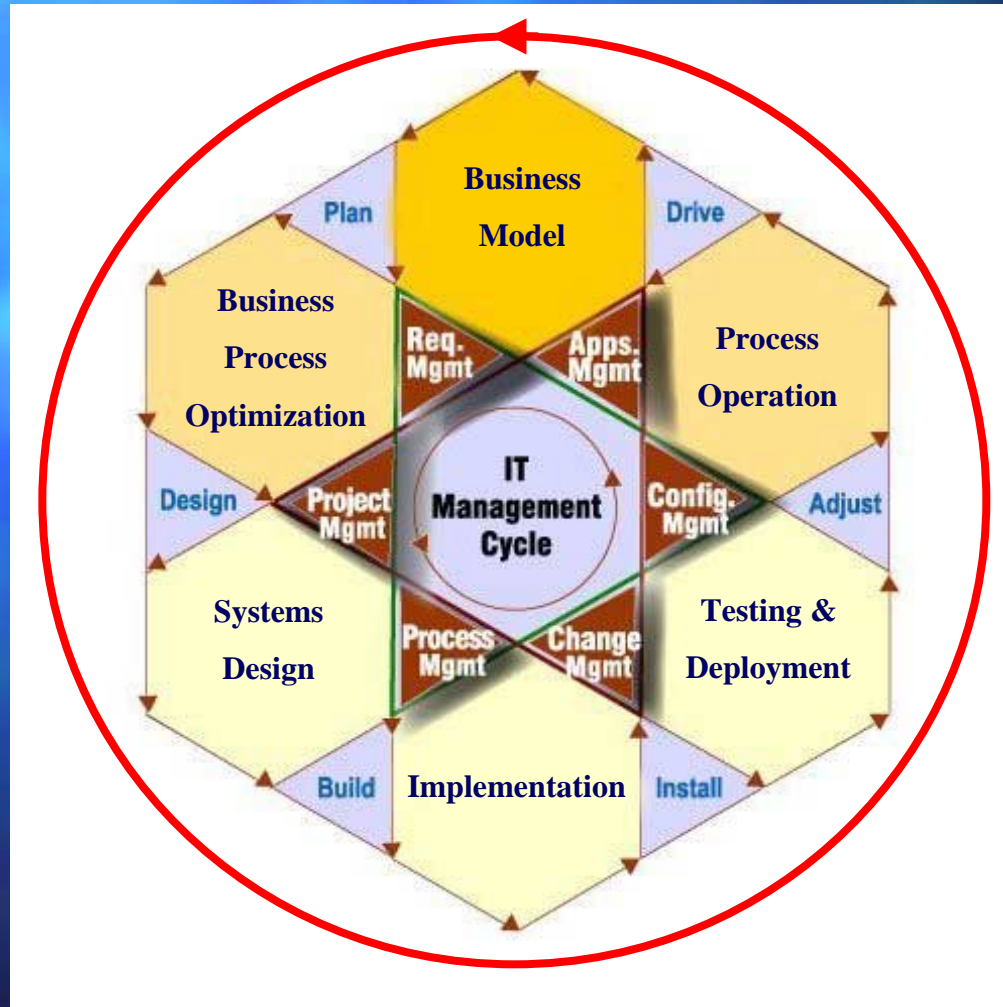


Integrating the Modeling Jobs



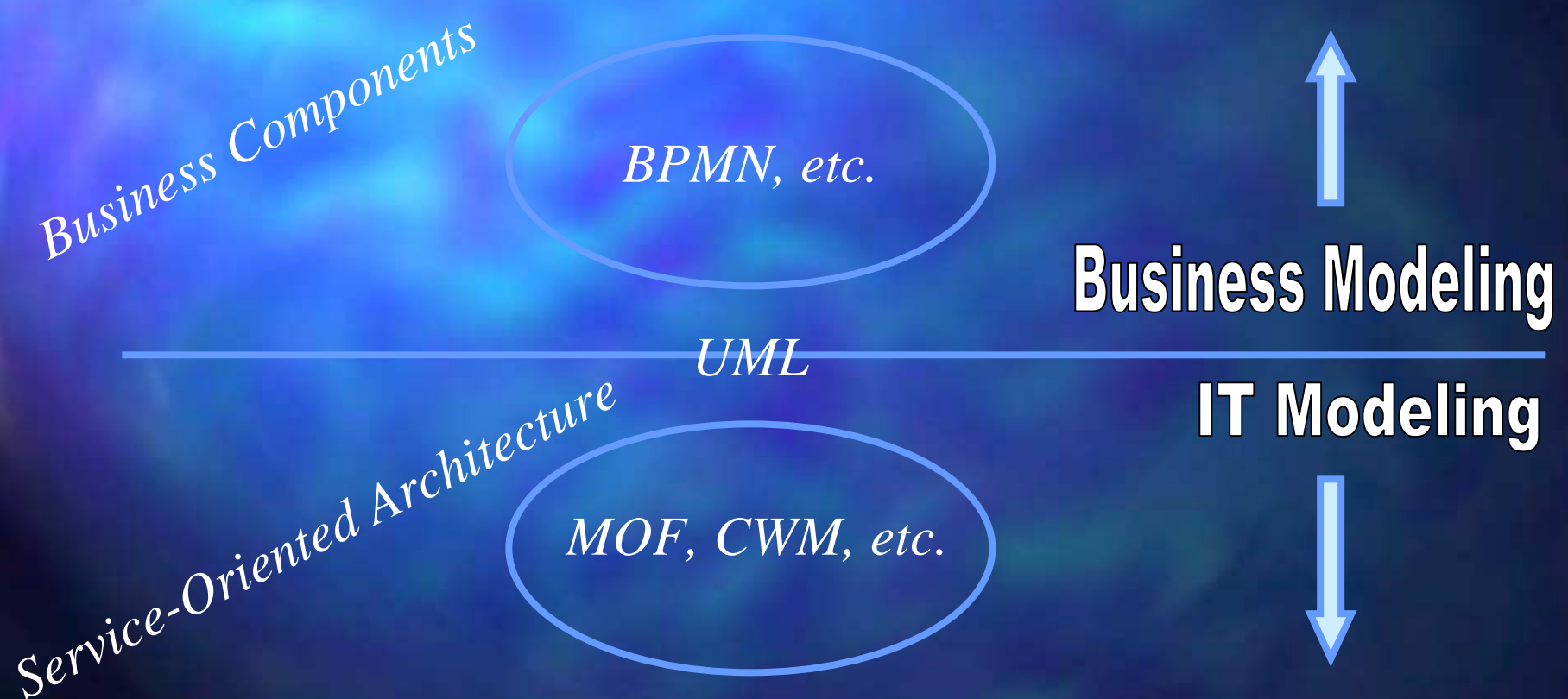


Business Operation Model





Modeling is Pervasive



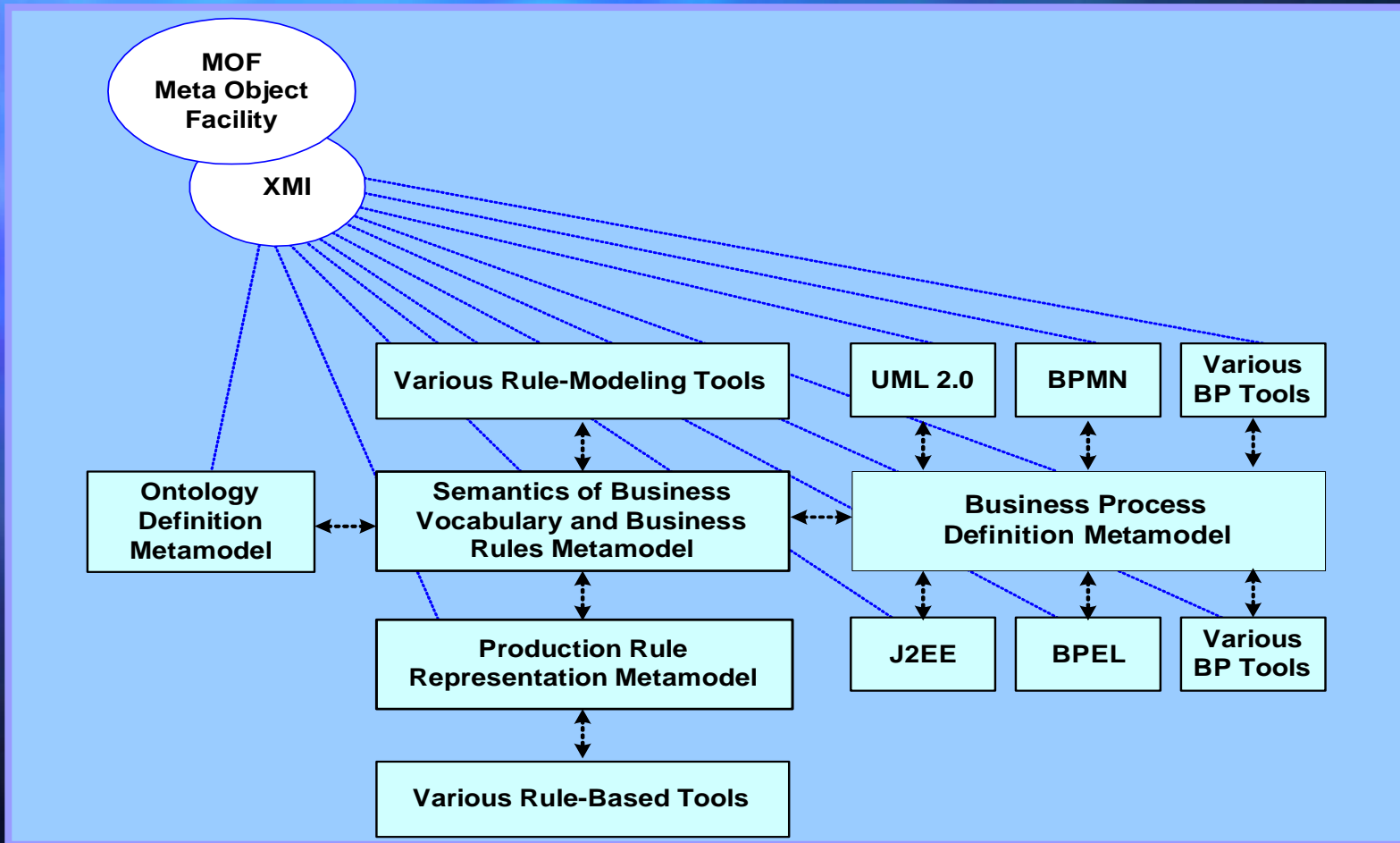


Business Modeling Work Under Way

- Business Motivation Metamodel
 - Completed by BRG; fast-tracked at OMG
- Business Process Modeling Notation
 - Complete by BPMI; fast-tracked at OMG
- Organization Structure Metamodel
 - In process; expected completion early '06
- Business Process Definition Metamodel
 - In process; expected completion mid '06
- Semantics of Business Vocabulary and Rules
 - Evaluation complete; adoption vote underway
- Production Rule Representation
 - In process; expected completion mid '06
- Business Process Maturity Model?



An Underlying MOF Model





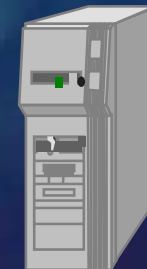
Enterprise IT Must Deal With

■ Business Factors:

- Defining & meeting changing business requirements
- Complex & changing business processes
- Shifting enterprise/application boundaries
- Semantic integration with customers, supplies & partners

■ Technological Factors:

- Barriers to interoperability & integration
- Development & maintenance obstacles
- Evolving & unstable technology suites





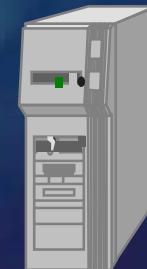
Enterprise IT: Adding MDA

■ Business Factors:

- Architectural viewpoint (service orientation—SOA) brings out how your applications work with each other, and with those on the outside
- Capture changing business requirements and shifting enterprise boundaries in editable models
- Define the business functionality and Behavior of each application as a technology-independent model
- Focus your IT investment in your core business

■ Technological Factors:

- Concentrate on the *business process*, speeding development
- Interoperability and portability are built in
- Move easily to the “next best thing”





MDA Benefits

- Full support for your “20 year architecture” across the application lifecycle
- Smooth integration across intra- and inter-business boundaries (across deployment technologies)
- Reduced costs from beginning to end
- Reuse of applications, code, training and people
- Technology-independent representation of the *business*
- Scalability, robustness & security via generated code
- Stable model-based approach maximizes ROI
- Rapid inclusion of the *next best thing*

The CIO Problem Solver



To Get More Information

- MDA Information Page
 - <http://www.omg.org/mda/>
- OMG General Information
 - <http://www.omg.org/>
- Contact the Author
 - soley@omg.org
- This presentation
 - <http://www.omg.org/~soley/mdaupdown.ppt>