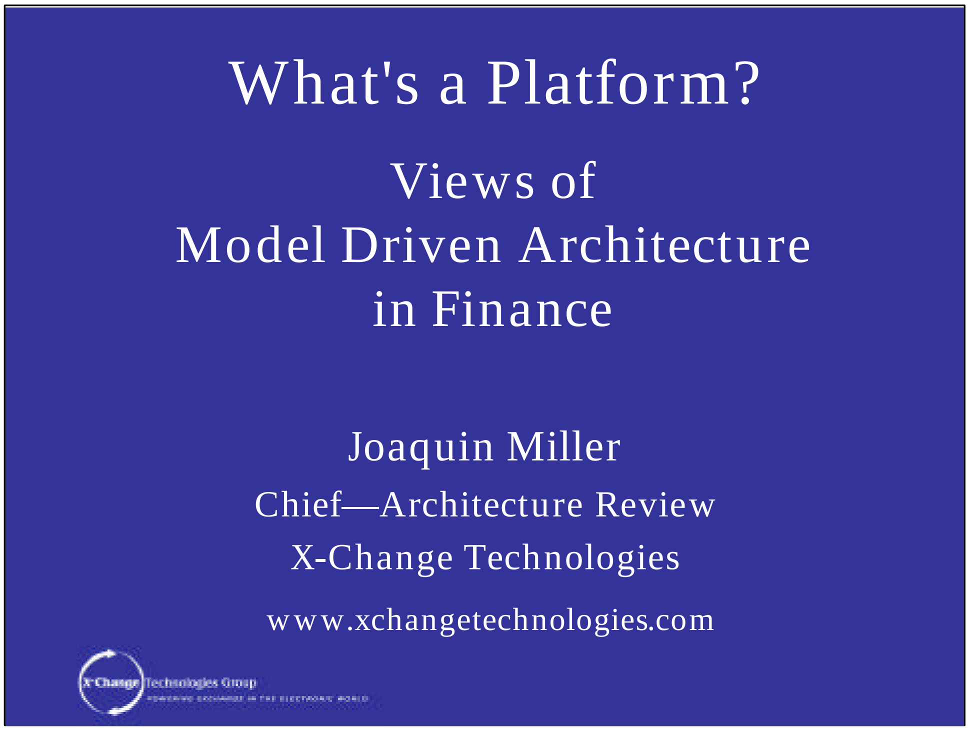
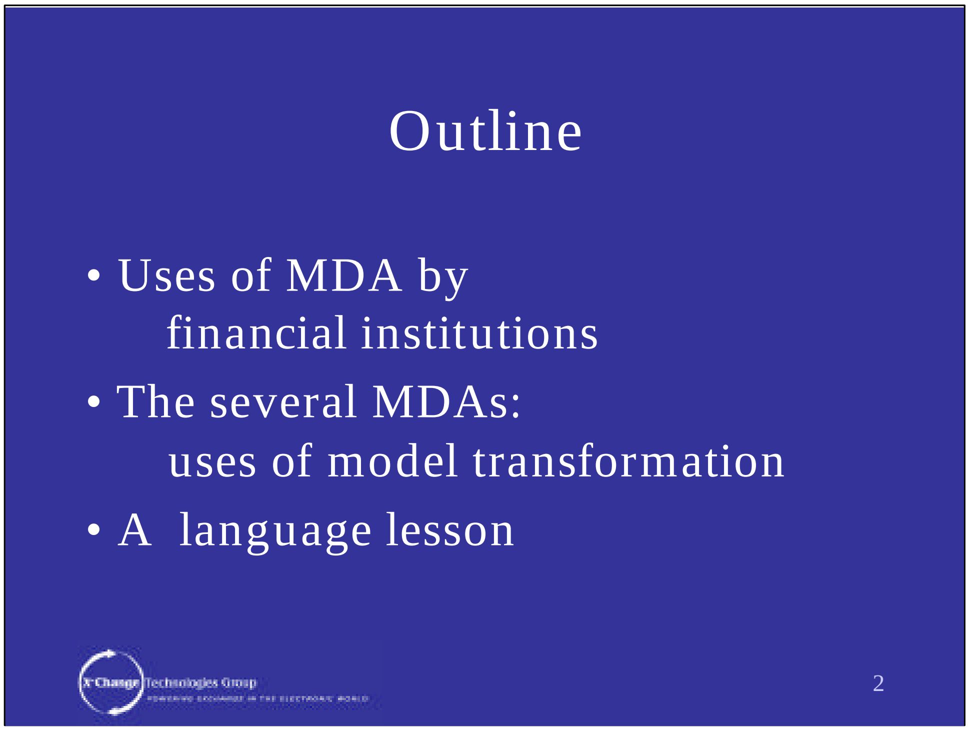


What's a Platform?

Views of Model Driven Architecture in Finance

Joaquin Miller
Chief—Architecture Review
X-Change Technologies
www.xchangetechnologies.com

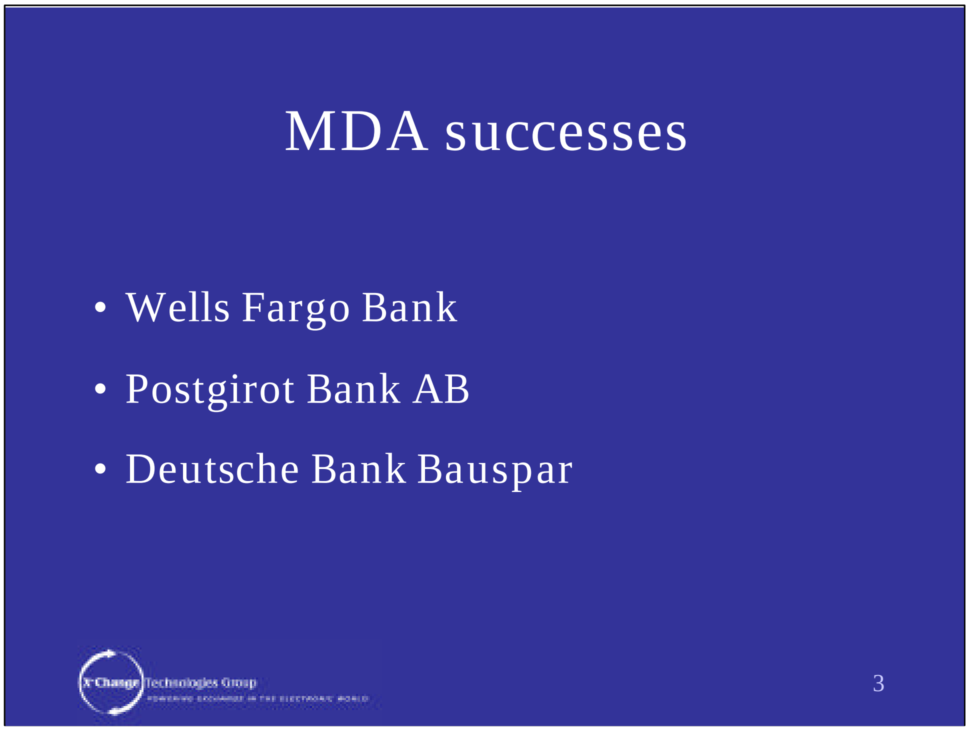


Outline

- Uses of MDA by
  financial institutions
- The several MDAs:
  uses of model transformation
- A language lesson

X-Change Technologies Group

POWERING EXCHANGE IN THE ELECTRONIC WORLD



MDA successes

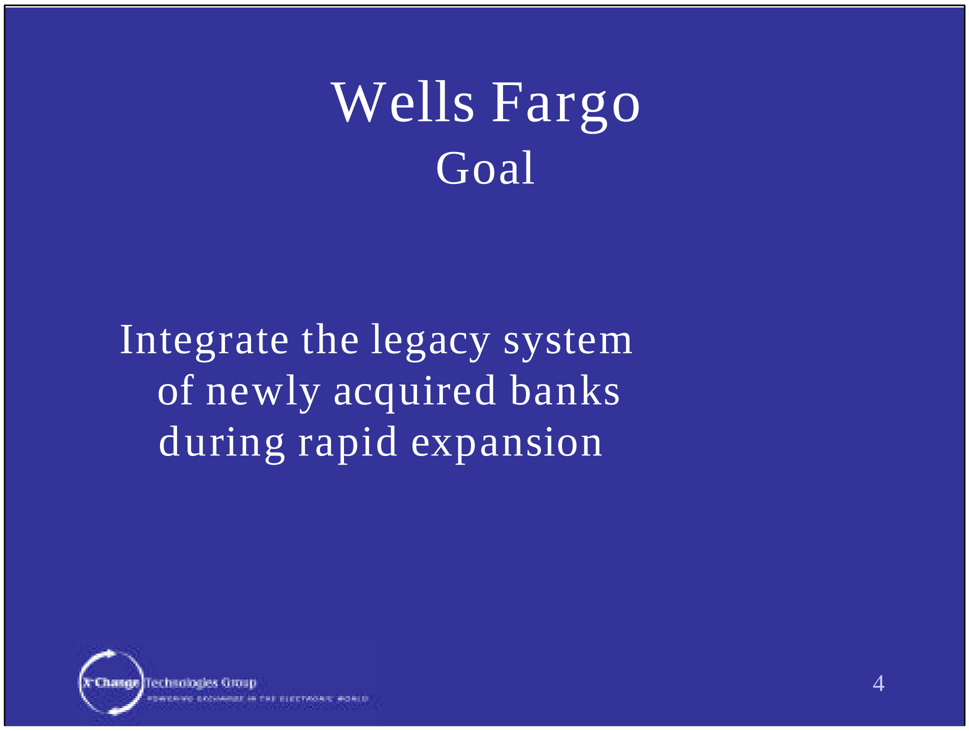
- Wells Fargo Bank
- Postgirot Bank AB
- Deutsche Bank Bauspar

X-Change Technologies Group

POWERING EXCHANGE IN THE ELECTRONIC WORLD

3

This talk is not about financial systems. I pick these examples for focus; since I have been working on Wall Street for the last four years; and because one important economic trend in East Asia is consolidation of financial institutions across national borders.



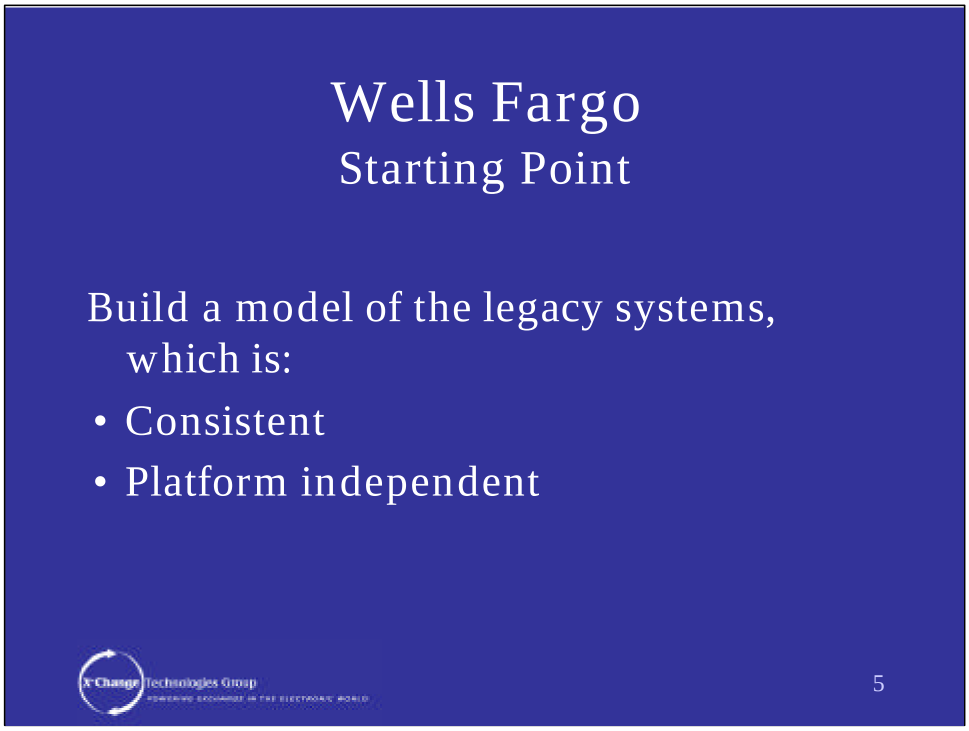
Wells Fargo
Goal

Integrate the legacy system of newly acquired banks during rapid expansion

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

4

The first example example is from my home, the San Francisco BayArea. This example illustrates one important point of this talk: MDA is not just about building new applications. It bring benefits in every area of information technology. This example shows the use of MDA in integrating existing legacy systems.



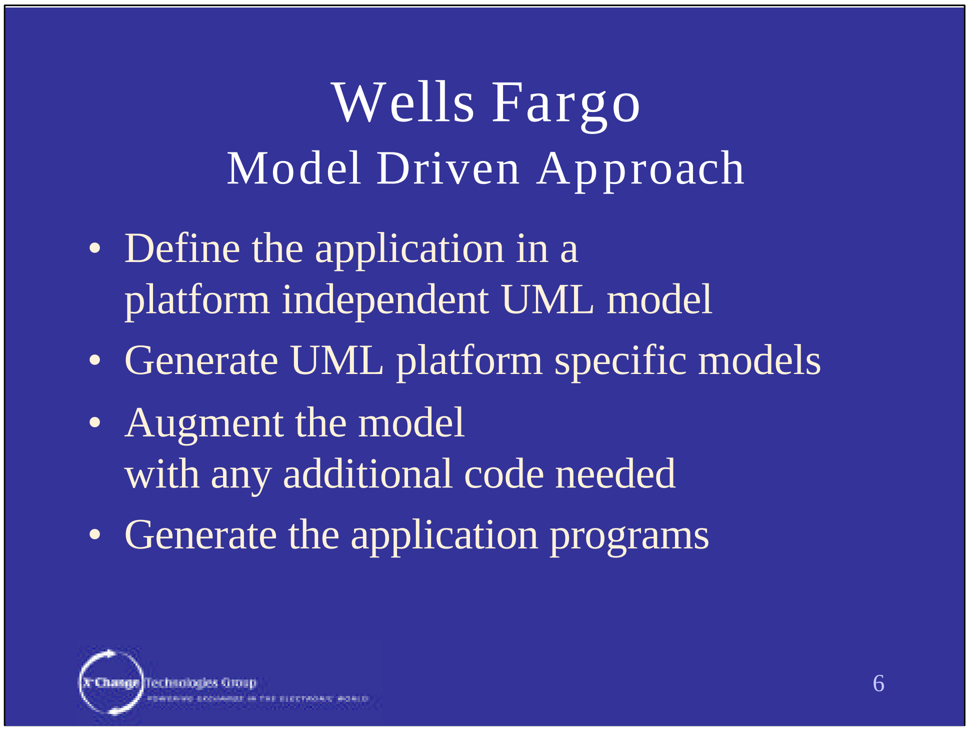
Wells Fargo
Starting Point

Build a model of the legacy systems, which is:

- Consistent
- Platform independent

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

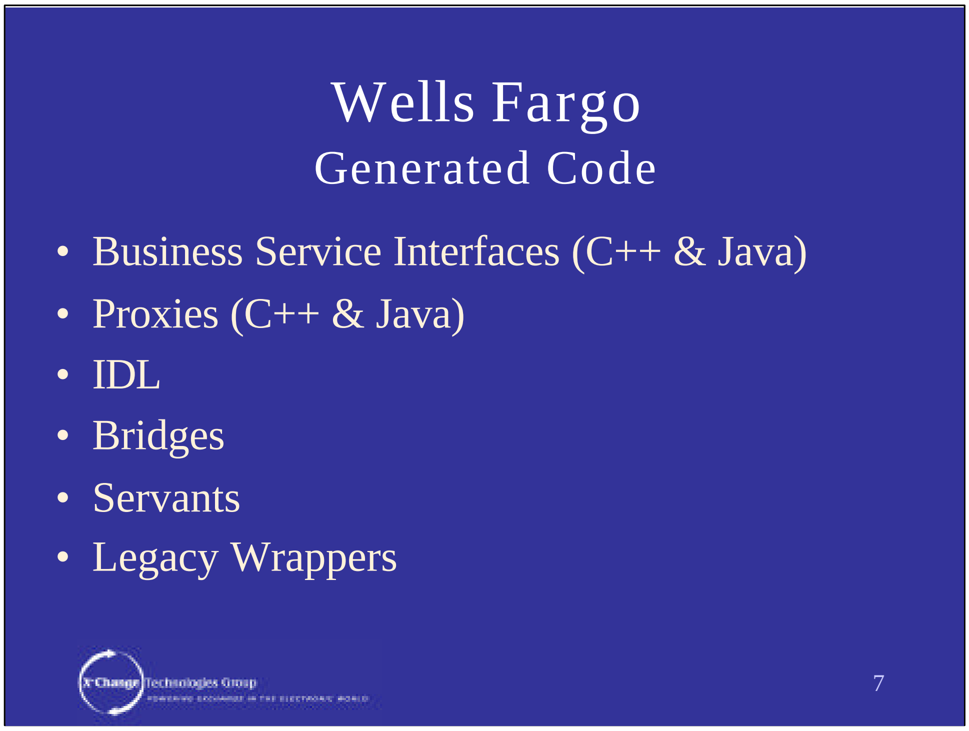
5



Wells Fargo
Model Driven Approach

- Define the application in a platform independent UML model
- Generate UML platform specific models
- Augment the model with any additional code needed
- Generate the application programs

X-Change Technologies Group



Wells Fargo
Generated Code

- Business Service Interfaces (C++ & Java)
- Proxies (C++ & Java)
- IDL
- Bridges
- Servants
- Legacy Wrappers

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

7

Wells Fargo
Underlying Platforms

CICS, COBOL, on Hitachi

Unix for new applications, on HP

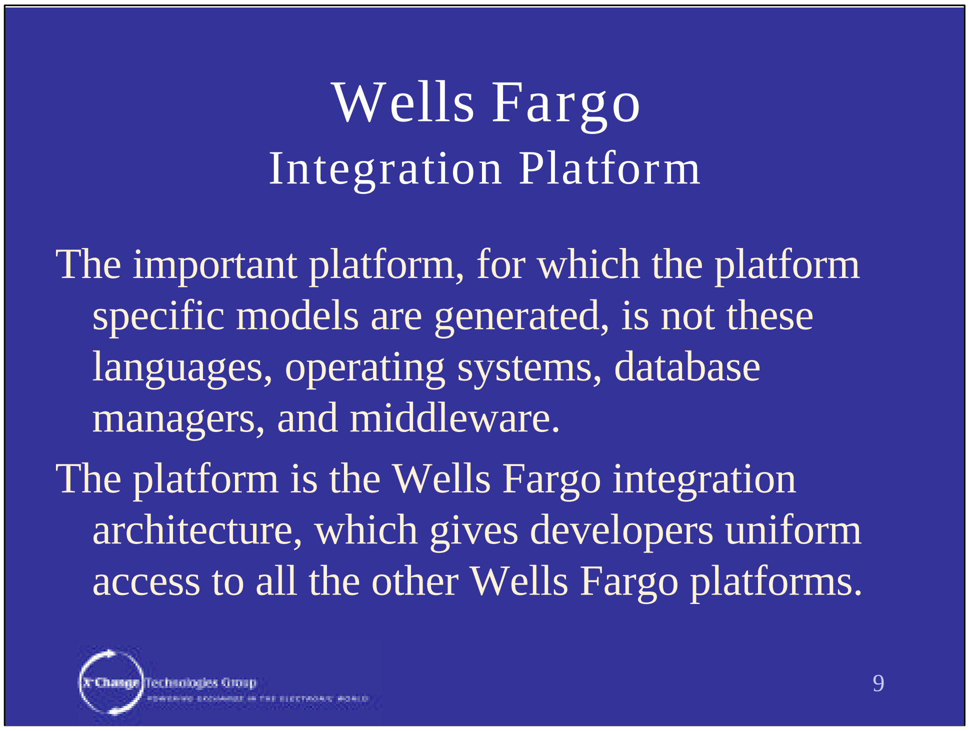
Microsoft NT for users, on PCs

IMS, DB/2, Informix, Oracle

CORBA

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

8



Wells Fargo
Integration Platform

The important platform, for which the platform specific models are generated, is not these languages, operating systems, database managers, and middleware.

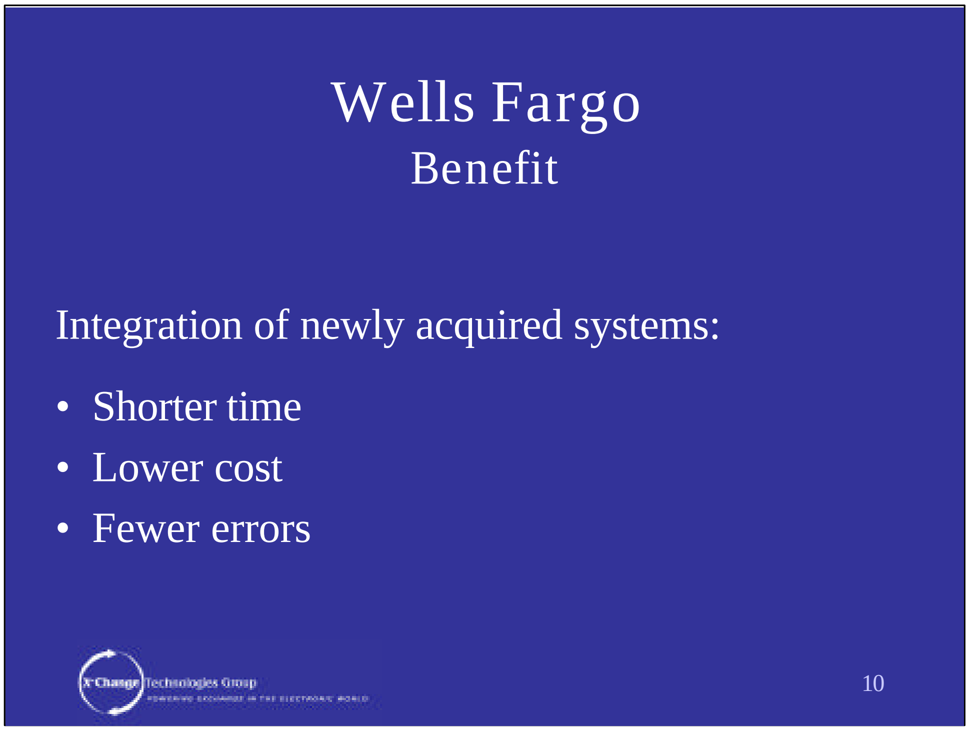
The platform is the Wells Fargo integration architecture, which gives developers uniform access to all the other Wells Fargo platforms.

X-Change Technologies Group

9

"In the middle is the system I am responsible for. It encapsulates these systems [of record] and provides a distributed environment that supports our client applications. We call this environment Wells Frame. It's an object-based framework that normalizes the disparate systems into a consistent enterprise model and provides a standard, consistent logical view to the client team. The enterprise model is described in UML for our clients. It is also expressed as CORBA interfaces for the application developers. Wells Frame is platform- and language-neutral as a result of being based on UML and CORBA."

— Eric H. Castain, a senior vice president of Wells Fargo



Wells Fargo
Benefit

Integration of newly acquired systems:

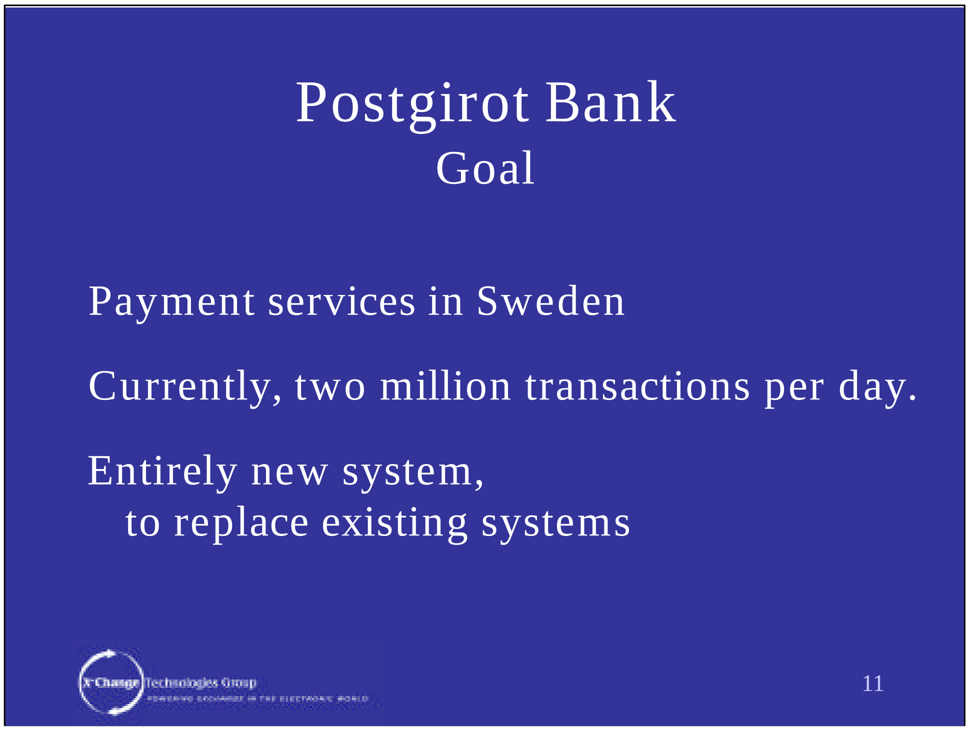
- Shorter time
- Lower cost
- Fewer errors

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

10

“In the middle is the system I am responsible for. It encapsulates these systems [of record] and provides a distributed environment that supports our client applications. We call this environment Wells Frame. It's an object-based framework that normalizes the disparate systems into a consistent enterprise model and provides a standard, consistent logical view to the client team. The enterprise model is described in UML for our clients. It is also expressed as CORBA interfaces for the application developers. Wells Frame is platform- and language-neutral as a result of being based on UML and CORBA.”

— Eric H. Castain, a senior vice president of Wells Fargo



Postgirot Bank
Goal

Payment services in Sweden

Currently, two million transactions per day.

Entirely new system,
to replace existing systems

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

11

Goals:

Capture new markets quickly by allowing fast implementation of new products and product features in order to decrease time to market of new product.

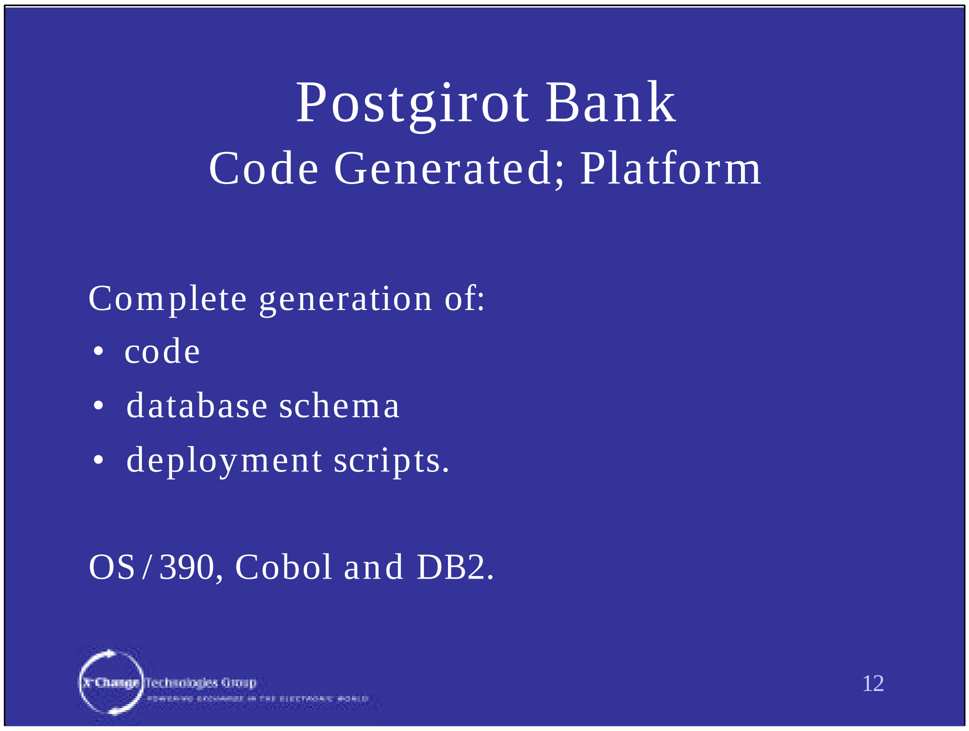
One flexible information system supporting all the payment transmission business. A system that is future proof.

Better insight and lower operational costs by using online, real time processing capabilities and straight through processing.

Integration with the other systems both internal and external to Postgirot.

Low deployment and maintenance costs.

Scalability to up to to much higher transaction rates.



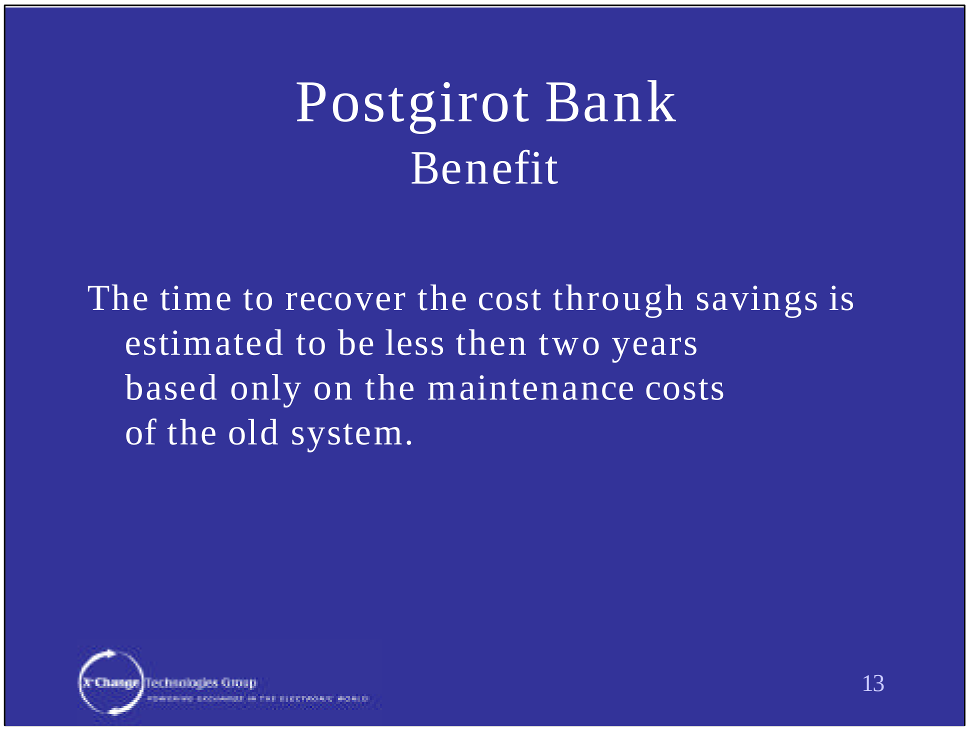
Postgirot Bank
Code Generated; Platform

Complete generation of:
- code
- database schema
- deployment scripts.

OS / 390, Cobol and DB2.

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

12



Postgirot Bank
Benefit

The time to recover the cost through savings is estimated to be less then two years based only on the maintenance costs of the old system.

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

13

Results:

Financial Gain - The operational cost are reported to have been reduced by 80% compared to the old situation.

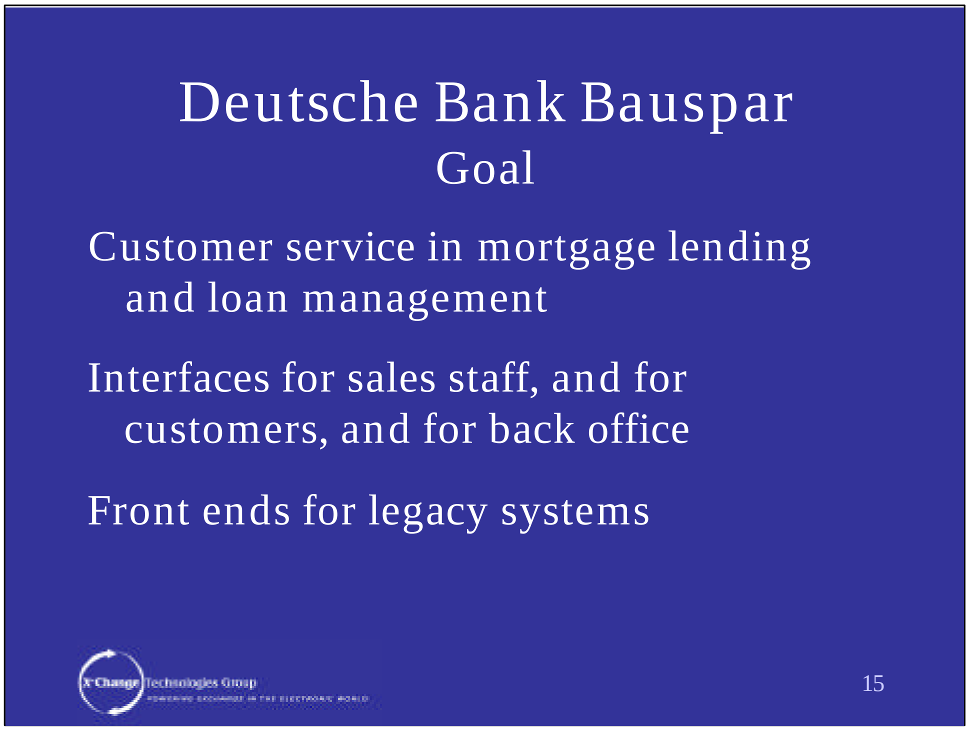
Flexibility - New products can be added quickly. New technology can be introduced and quickly integrated with legacy systems.

Scalablity - The solution can grow to 30 Million transactions per day.

Postgirot Bank

Benefit

Since code was generated from Platform Independent Models (PIMs) Postgirot can easily switch to J2EE or another technology.



Deutsche Bank Bauspar

Goal

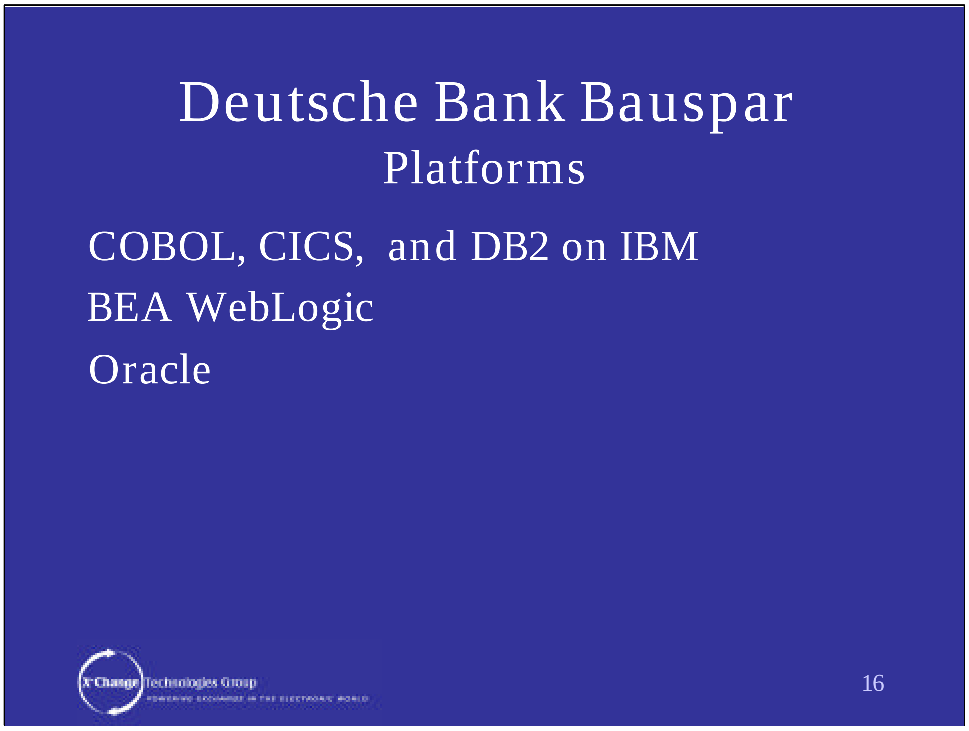
Customer service in mortgage lending and loan management

Interfaces for sales staff, and for customers, and for back office

Front ends for legacy systems

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

15



Deutsche Bank Bauspar

Platforms

COBOL, CICS, and DB2 on IBM

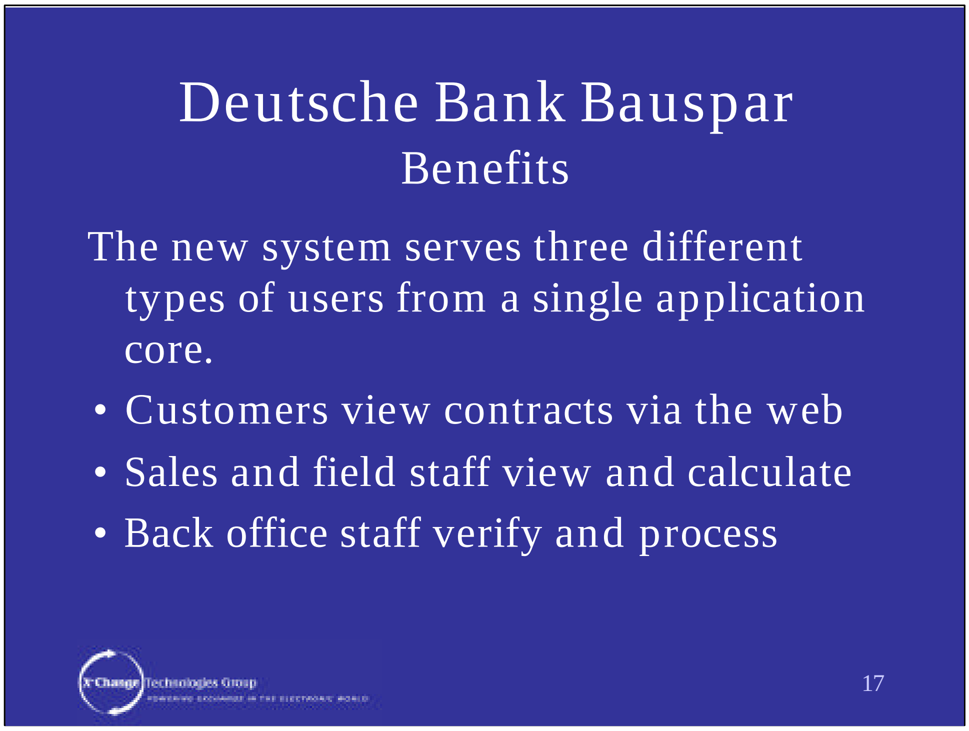
BEA WebLogic

Oracle

X-Change Technologies Group

POWERING EXCHANGE IN THE ELECTRONIC WORLD

16



Deutsche Bank Bauspar

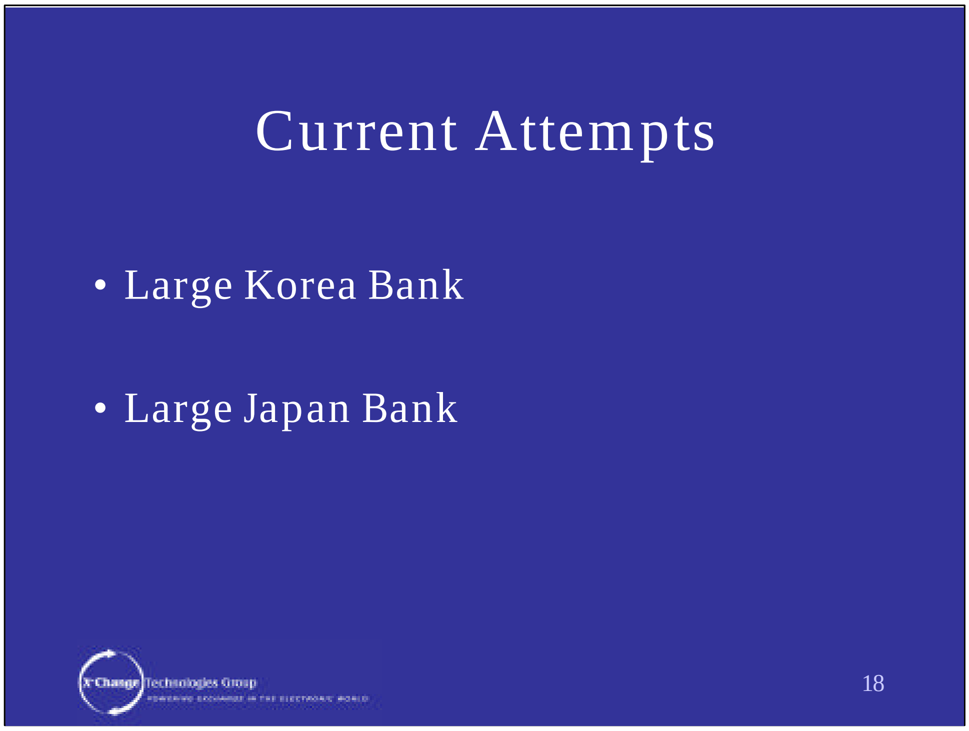
Benefits

The new system serves three different types of users from a single application core.

- Customers view contracts via the web
- Sales and field staff view and calculate
- Back office staff verify and process

X-Change Technologies Group

17



Current Attempts

- Large Korea Bank
- Large Japan Bank

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

18

Here is some information I have gathered recently, about the use of MDA in East Asia

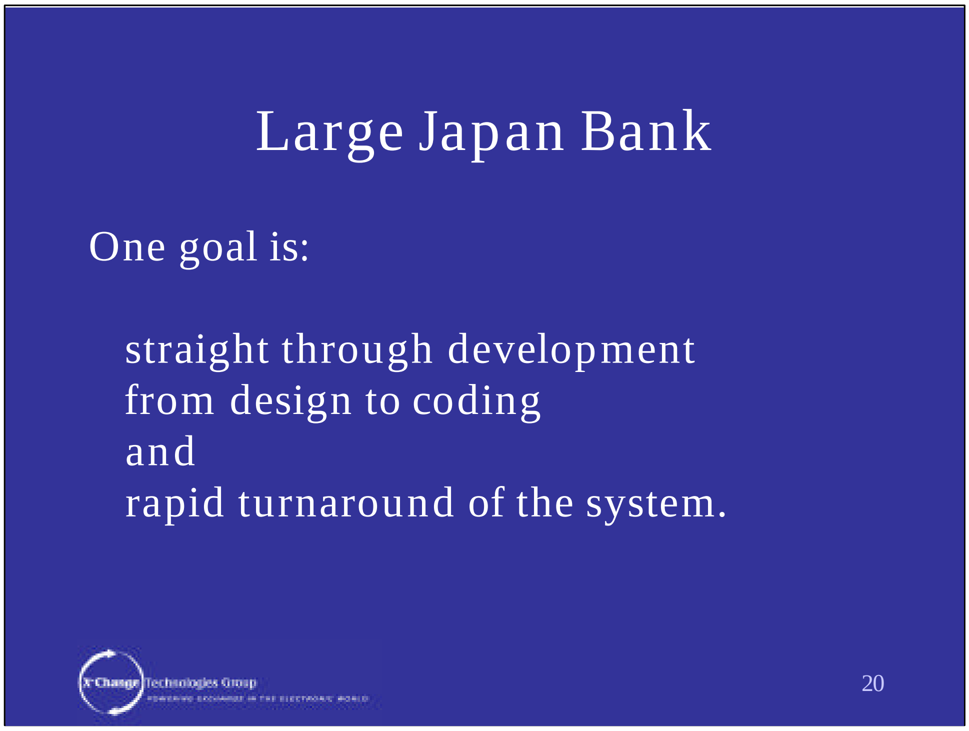
Large Korea Bank

Trying out MDA

for its next generation bank system

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

19



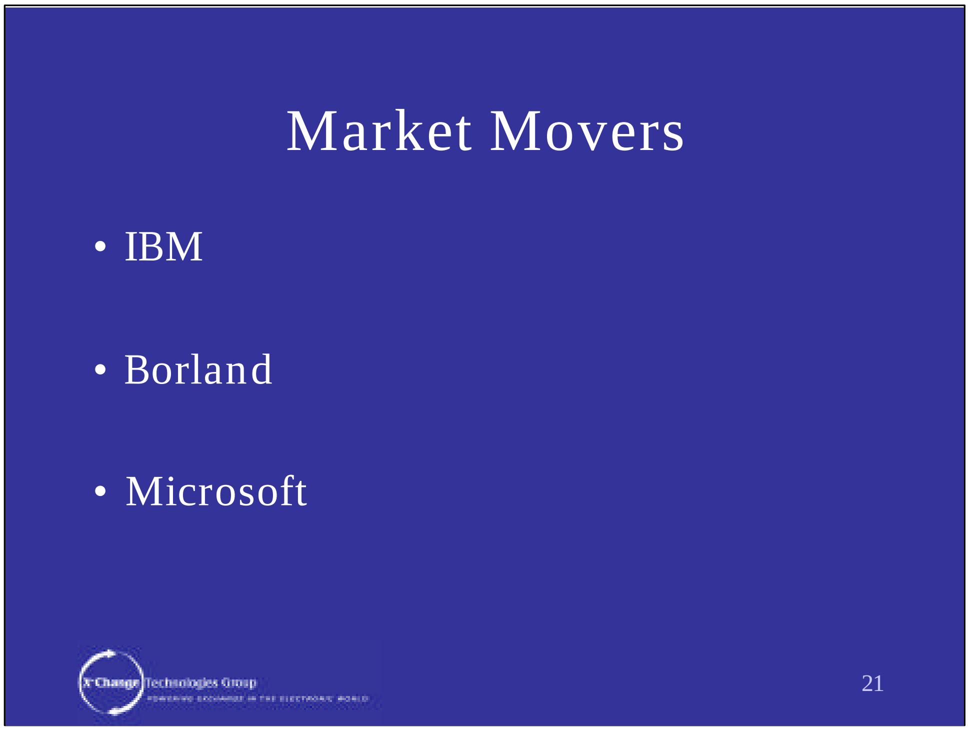
Large Japan Bank

One goal is:

straight through development
from design to coding
and
rapid turnaround of the system.

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

20

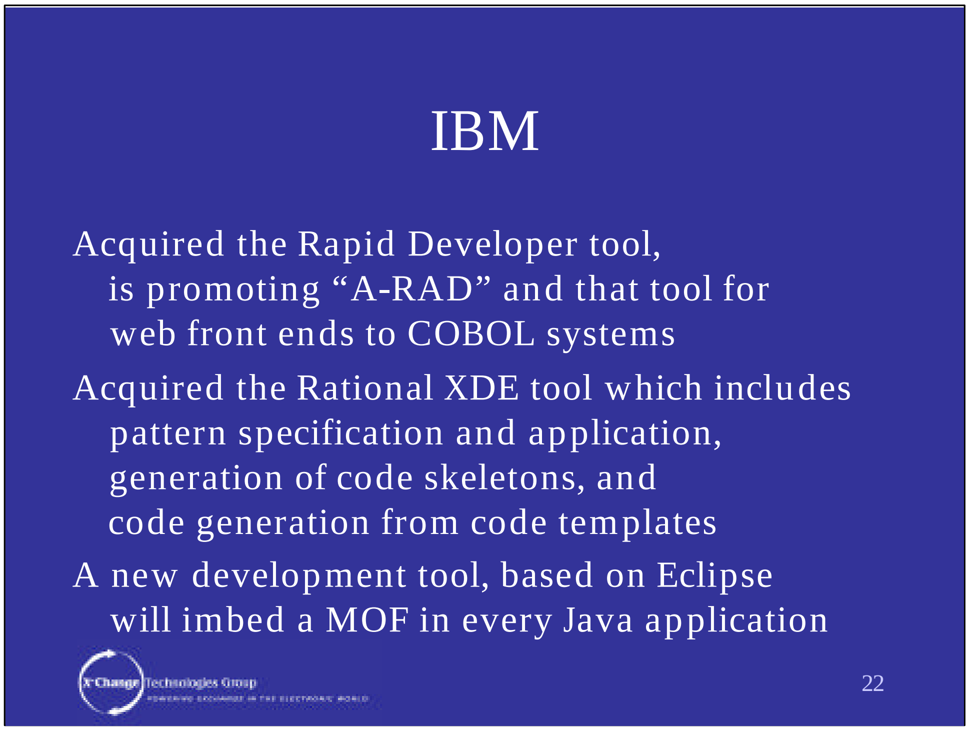

Market Movers

- IBM
- Borland
- Microsoft

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

21

Another way for you to evaluate MDA is to see what software vendors are doing.



IBM

Acquired the Rapid Developer tool,
is promoting "A-RAD" and that tool for
web front ends to COBOL systems

Acquired the Rational XDE tool which includes
pattern specification and application,
generation of code skeletons, and
code generation from code templates

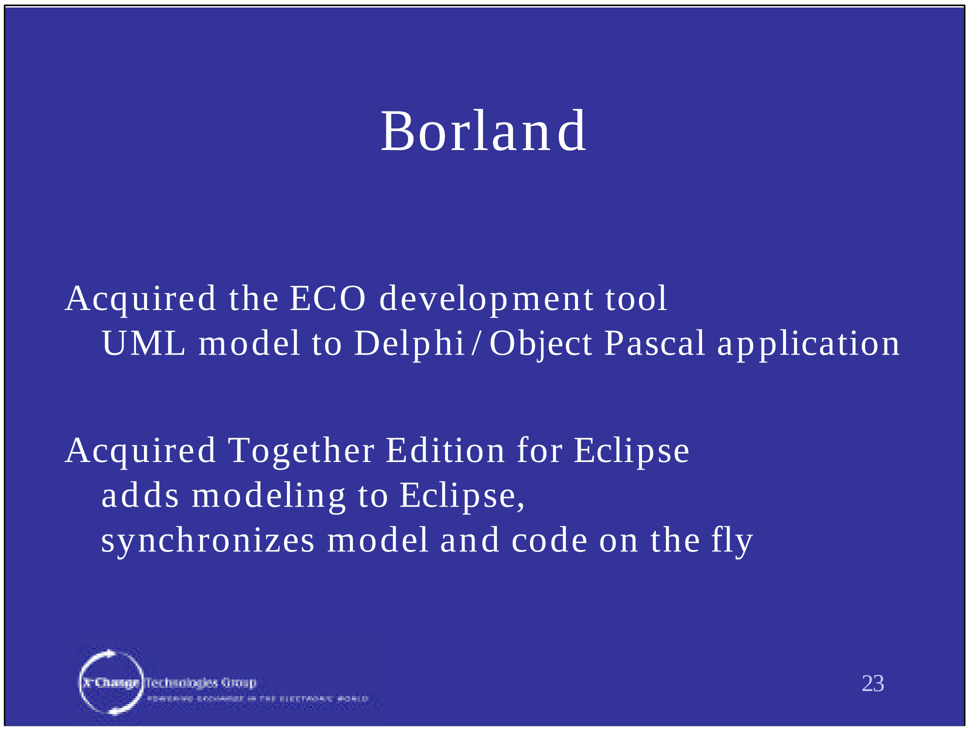
A new development tool, based on Eclipse
will imbed a MOF in every Java application

X-Change Technologies Group

22

When it bought Rational, IBM acquired the NeuVis software product, Rapid Developer, which it is now promoting for what it calls A-RAD, targeted at COBOL shops wishing to add web front ends.

IBM's new development tool, based on Eclipse, will imbed a MOF in every Java program, so that programs can refer to models during execution.



Borland

Acquired the ECO development tool
  UML model to Delphi / Object Pascal application

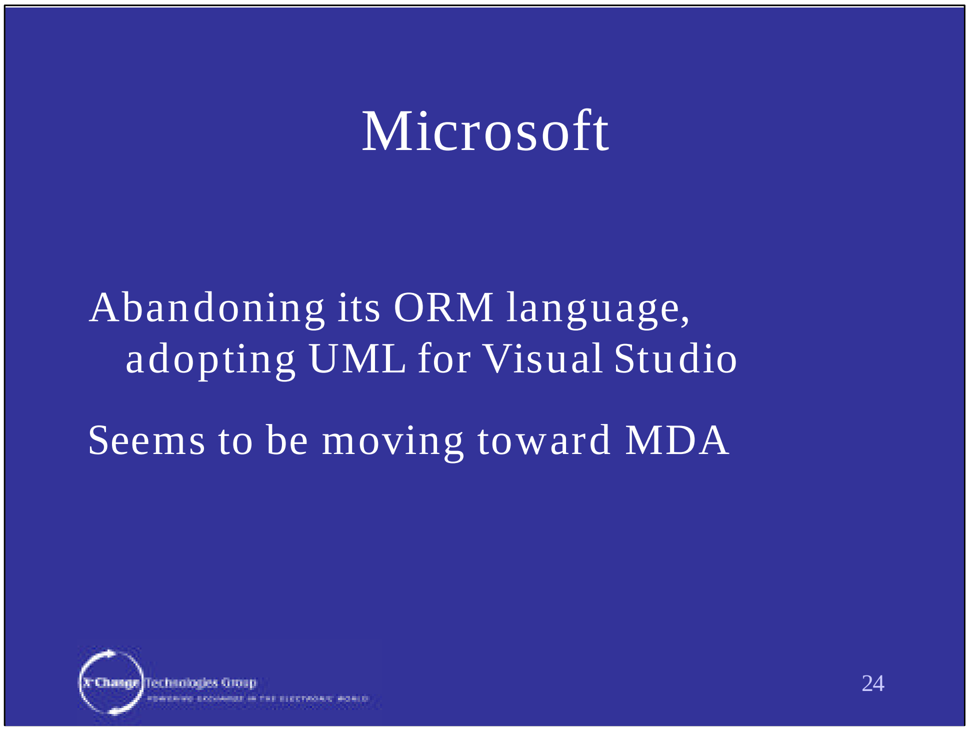
Acquired Together Edition for Eclipse
  adds modeling to Eclipse,
  synchronizes model and code on the fly

X-Change Technologies Group

23

Borland acquired BoldSoft, obtaining world class OCL expertise and a complete MDA product (now called ECO), which maps a problem domain UML diagram into a complete working application using Delphi/Object Pascal.
Prior to being acquired by Borland, TogetherSoft had developed Together Edition for Eclipse, which adds a modeling perspective to Eclipse, with on-the-fly synchronization between model and code, so models are reflected instantaneously in code skeletons, and changes to the code skeletons are reflected in the model.

These code skeletons do include much code. For example, if the programmer applies J2EE patterns to the model, the resulting code skeleton will compile and run if deployed, but there will be no business logic in the methods.



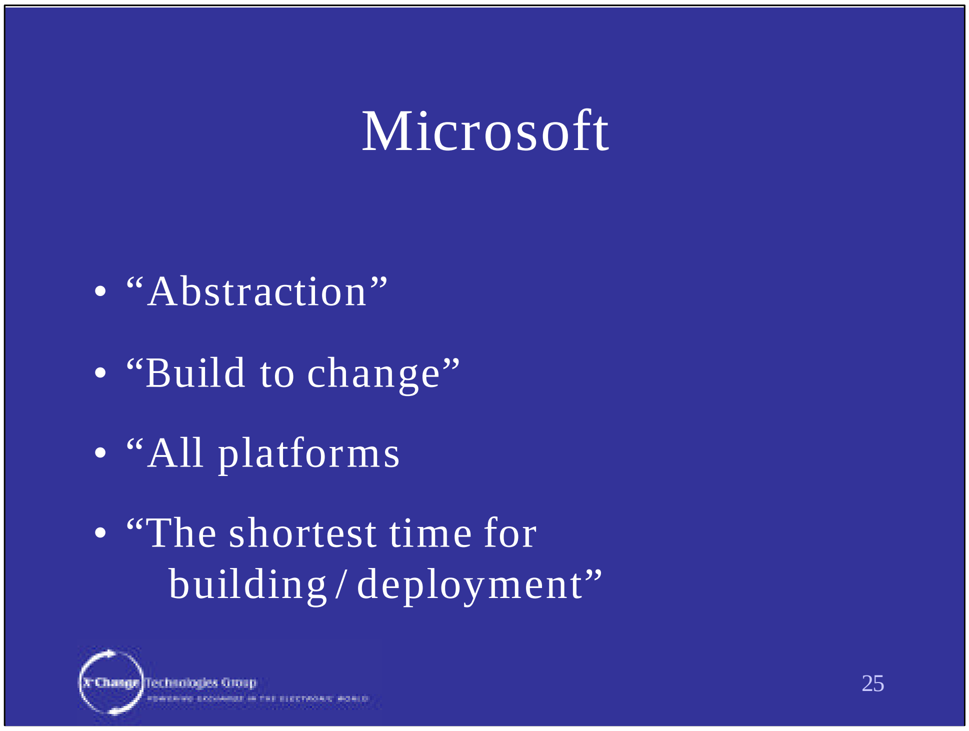
Microsoft

Abandoning its ORM language, adopting UML for Visual Studio

Seems to be moving toward MDA

X-Change Technologies Group

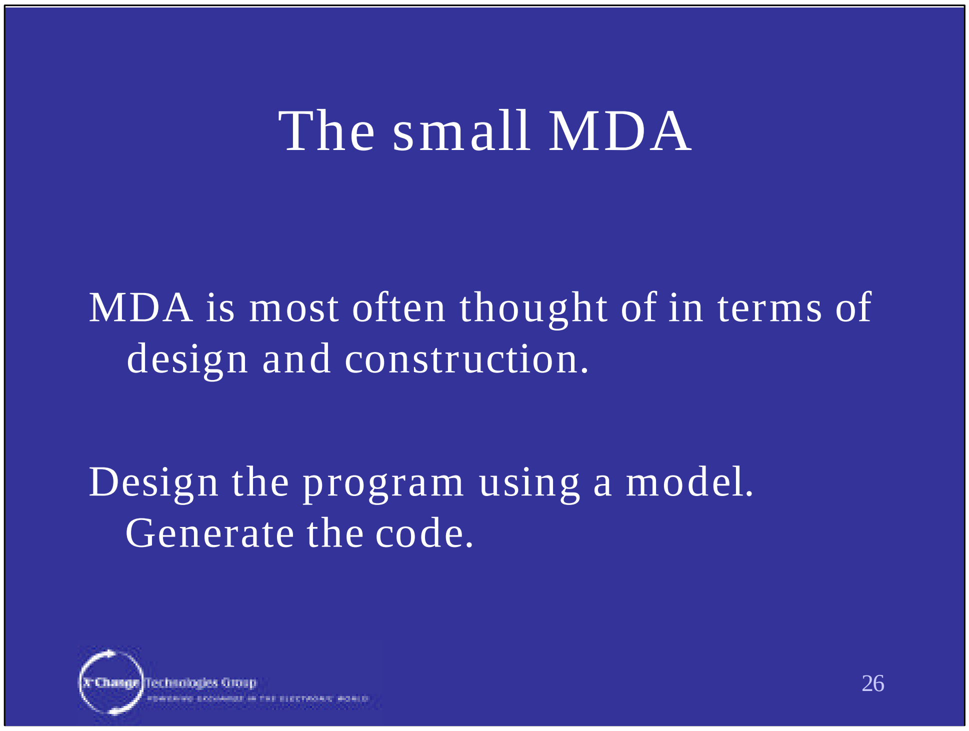
24


Microsoft

- “Abstraction”
- “Build to change”
- “All platforms
- “The shortest time for building / deployment”

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

25



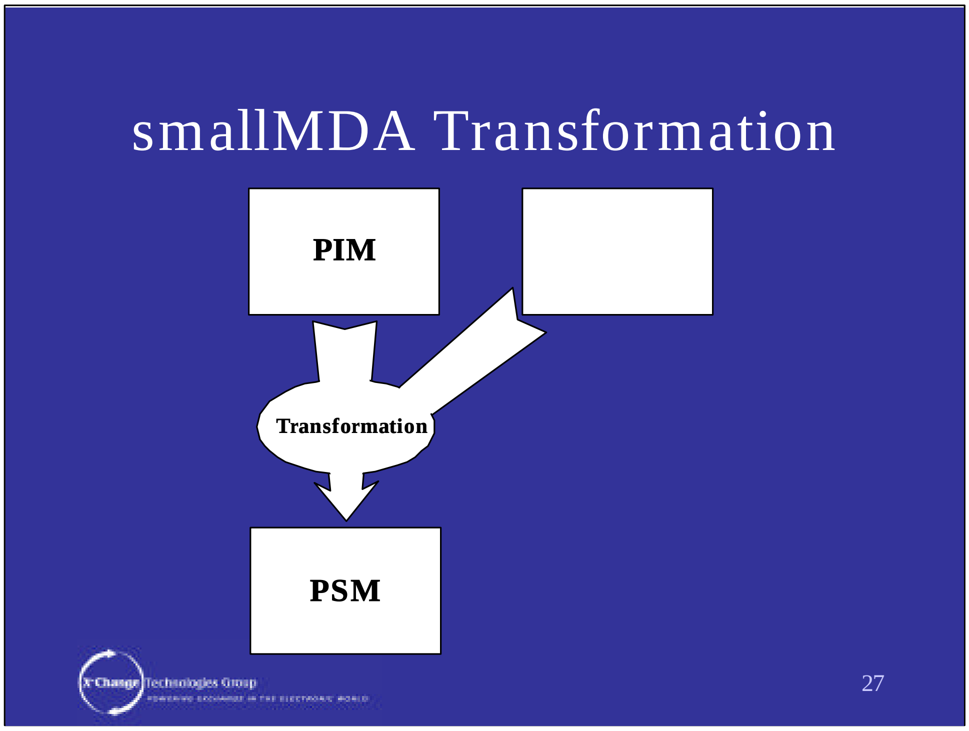
The small MDA

MDA is most often thought of in terms of design and construction.

Design the program using a model. Generate the code.

X-Change Technologies Group

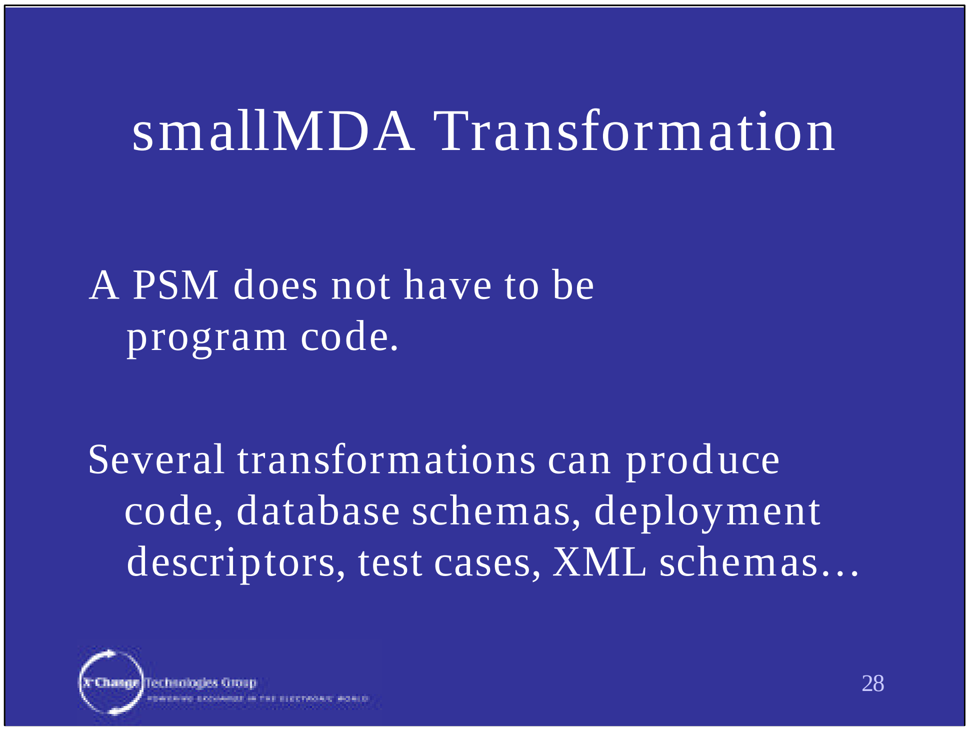
POWERING EXCHANGE IN THE ELECTRONIC WORLD



smallMDA Transformation

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

27

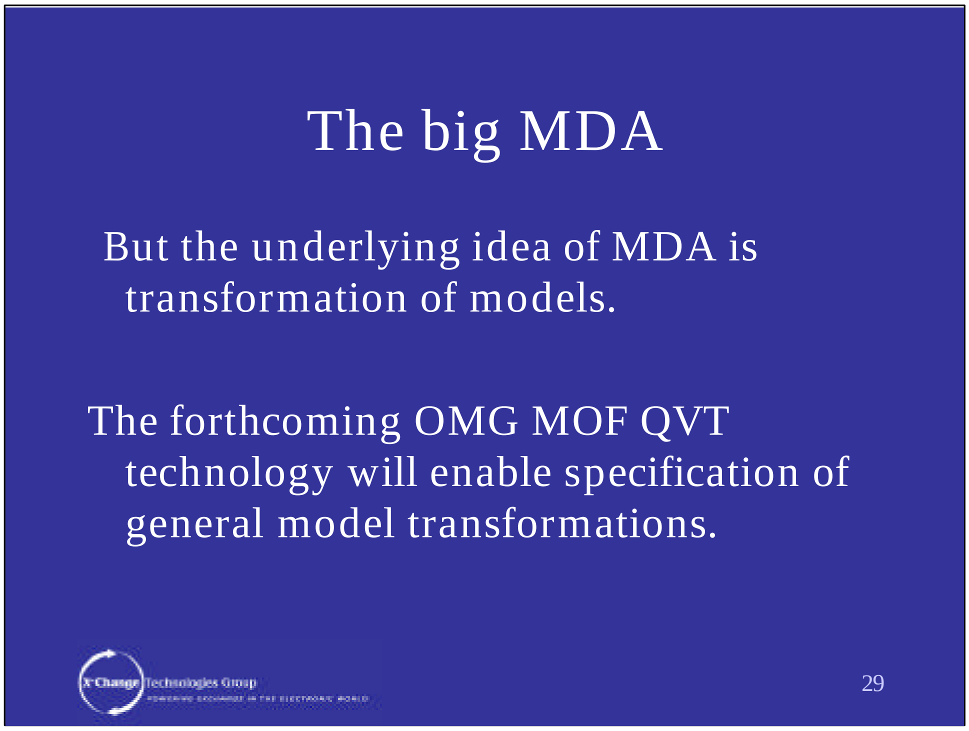


smallMDA Transformation

A PSM does not have to be program code.

Several transformations can produce code, database schemas, deployment descriptors, test cases, XML schemas…

28



The big MDA

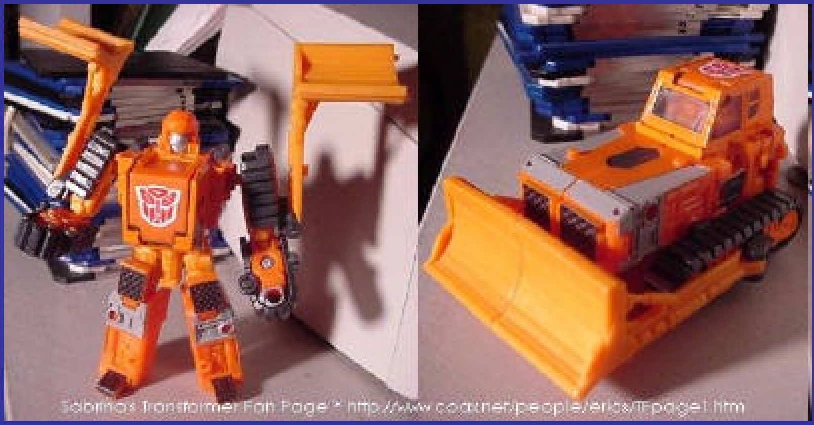
But the underlying idea of MDA is transformation of models.

The forthcoming OMG MOF QVT technology will enable specification of general model transformations.

X-Change Technologies Group

29

Transformation

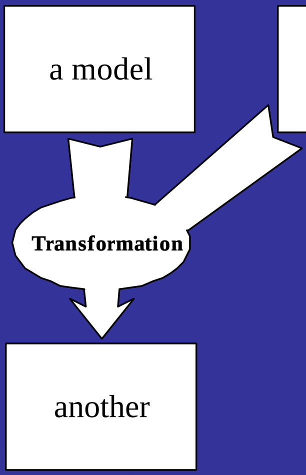

Sabrina's Transformer Fan Page * http://www.coaxnet/people/erids/TFpage1.htm

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

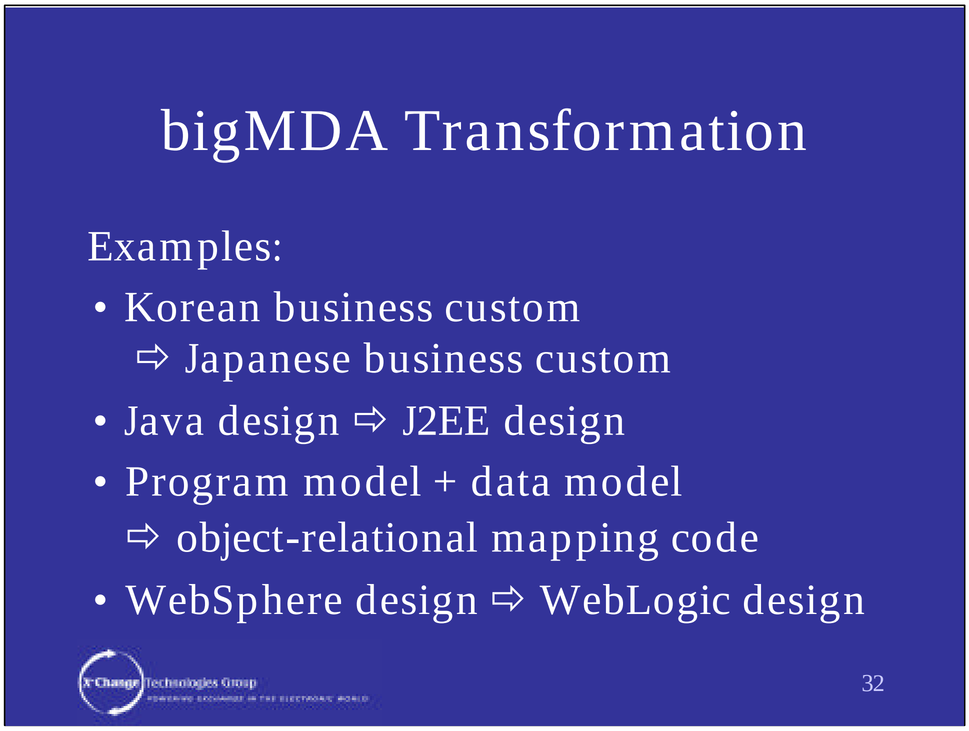
30

Notice the software in the background. It requires twenty-three diskettes of software to carry out this simple transformation from a robot to a bulldozer. That is quite a bit of software, but the results are extraordinary.

bigMDA Transformation



X-Change Technologies Group
POWERING EXCHANGES IN THE ELECTRONIC WORLD

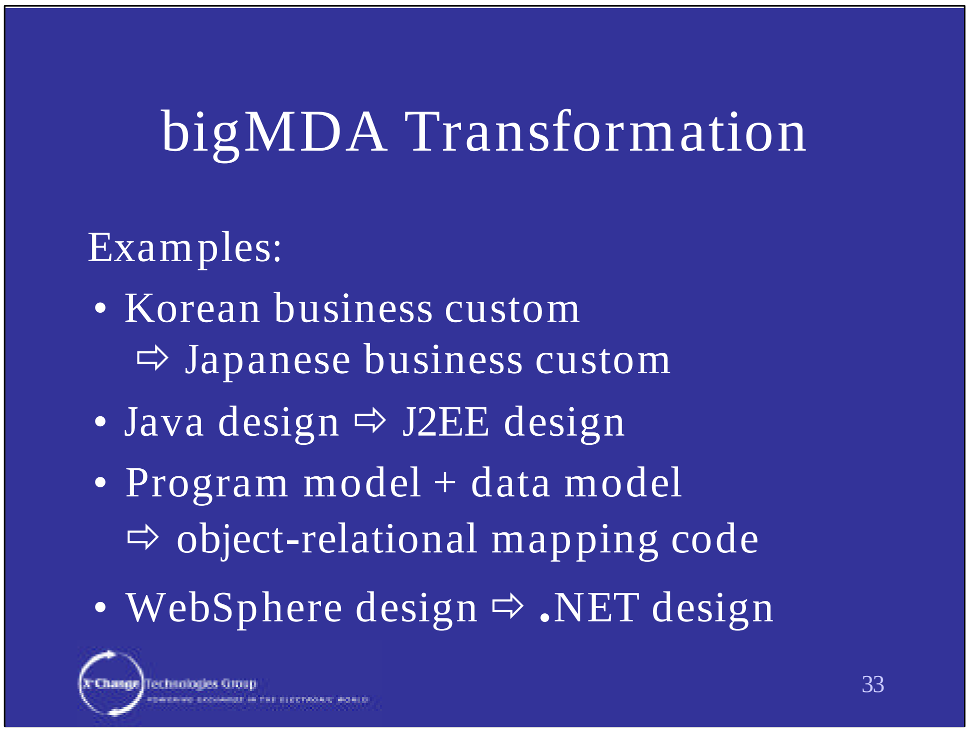


bigMDA Transformation

Examples:

- Korean business custom
  ⇨ Japanese business custom
- Java design ⇨ J2EE design
- Program model + data model
  ⇨ object-relational mapping code
- WebSphere design ⇨ WebLogic design

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

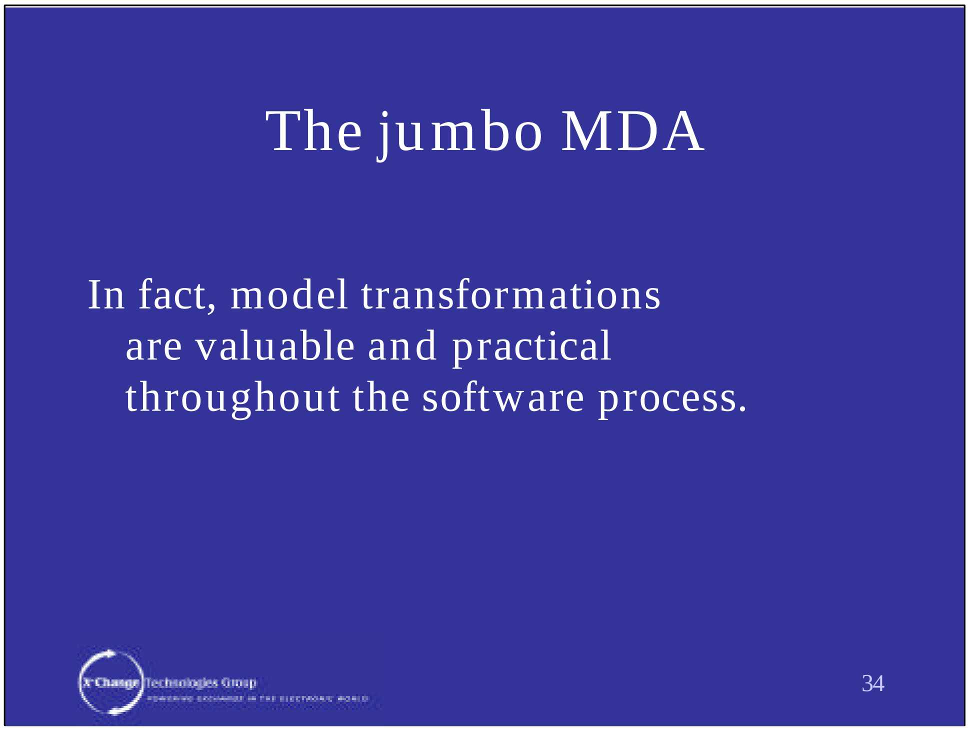


bigMDA Transformation

Examples:

- Korean business custom
  ⇨ Japanese business custom
- Java design ⇨ J2EE design
- Program model + data model
  ⇨ object-relational mapping code
- WebSphere design ⇨ .NET design

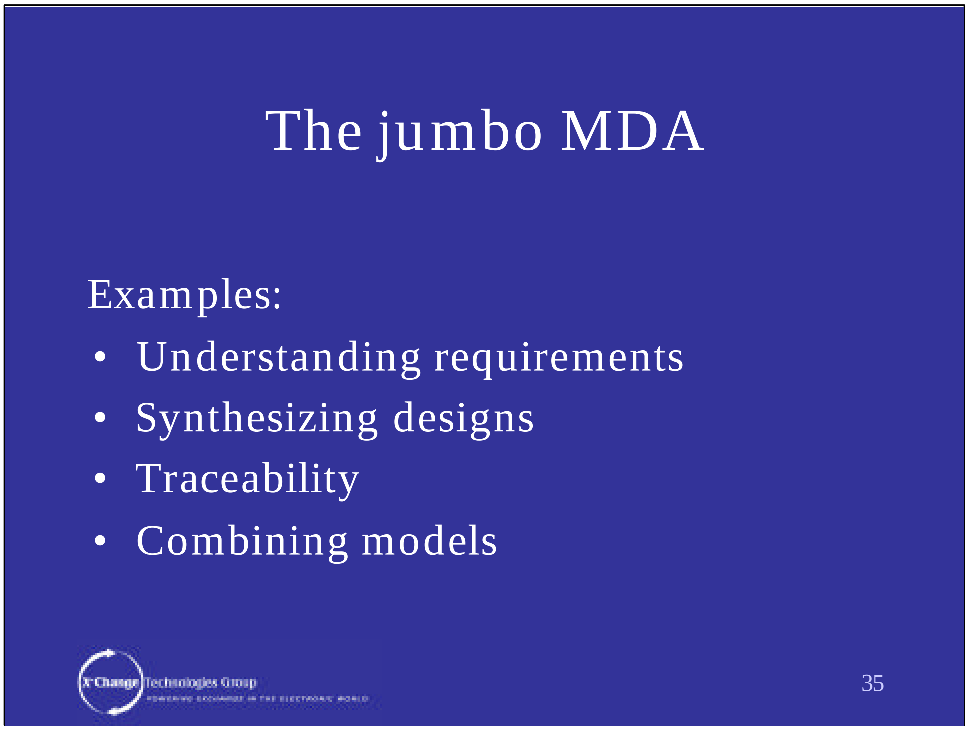
X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD



The jumbo MDA

In fact, model transformations are valuable and practical throughout the software process.

34



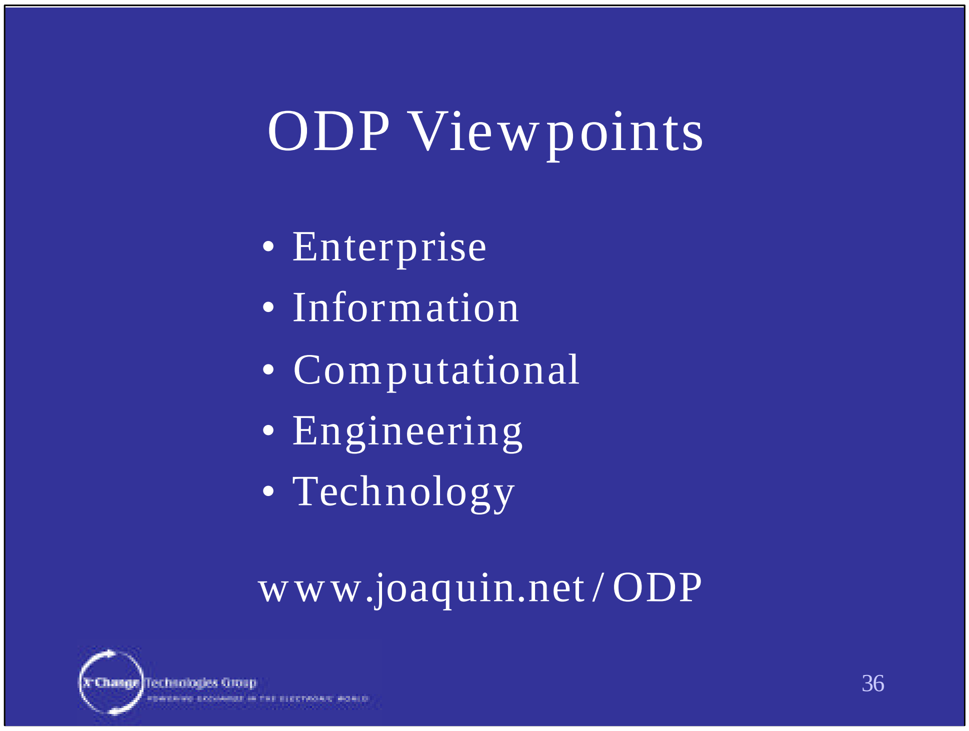
The jumbo MDA

Examples:

- Understanding requirements
- Synthesizing designs
- Traceability
- Combining models

X-Change Technologies Group

35



ODP Viewpoints

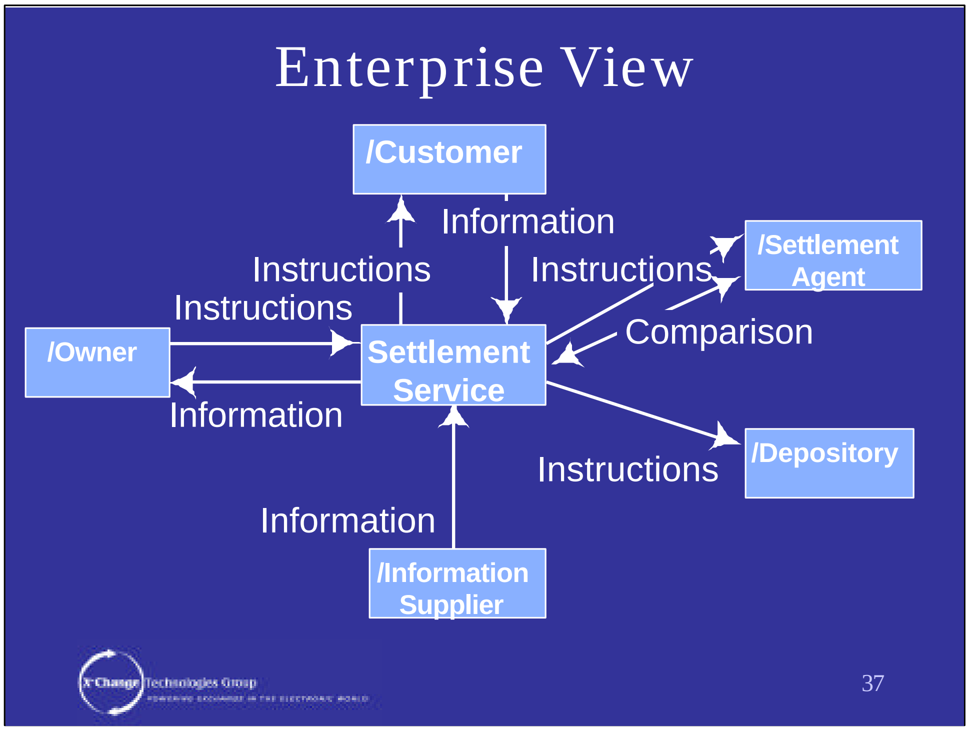
- Enterprise
- Information
- Computational
- Engineering
- Technology

www.joaquin.net/ODP

X-Change Technologies Group

36

Here is one example of the application of MDA technology in a way that might not be expected. This will be an extensive example. At the same time we set up the environment for the example, we will also see an example of the use of another important architecture, the Reference Model of Open Distributed Computing, RM-ODP.

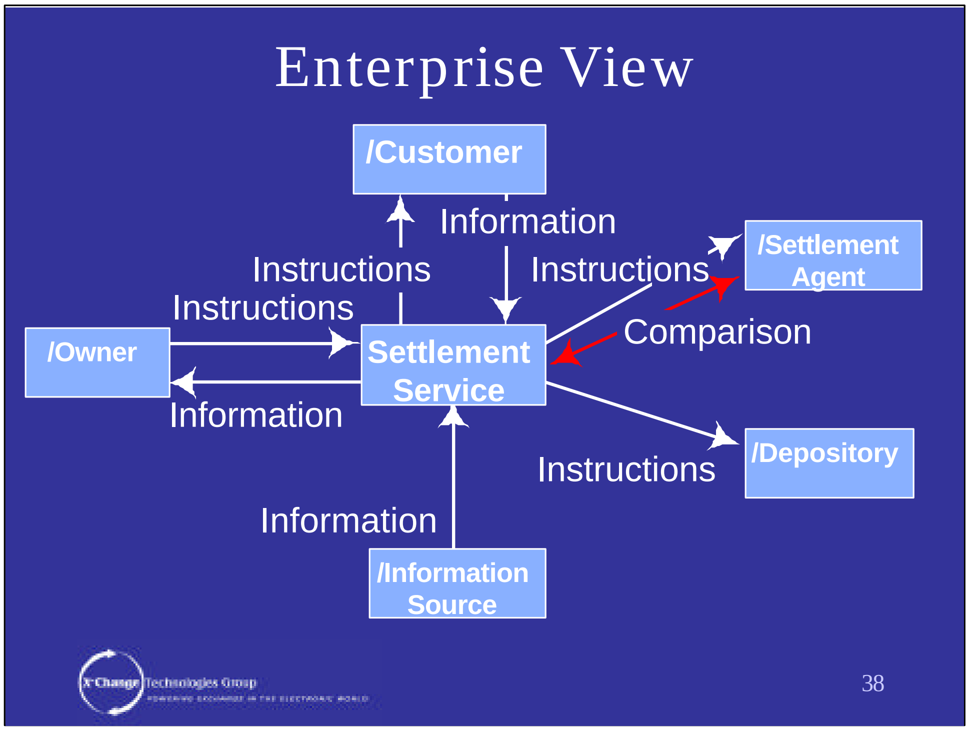


Enterprise View

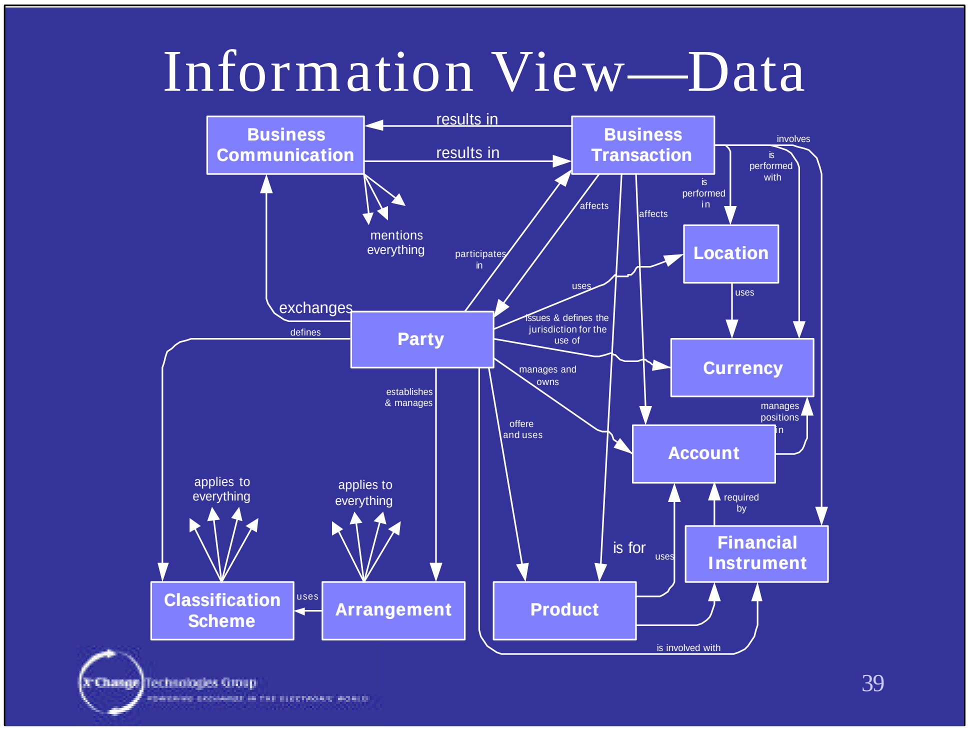
The model shows the system, Settlement Service, in its environment, specified from the ODP Enterprise viewpoint.

The other boxes represent roles of parties and other systems, and the lines show interactions with the system.

The system provides services to clear and settle trades of financial instruments.

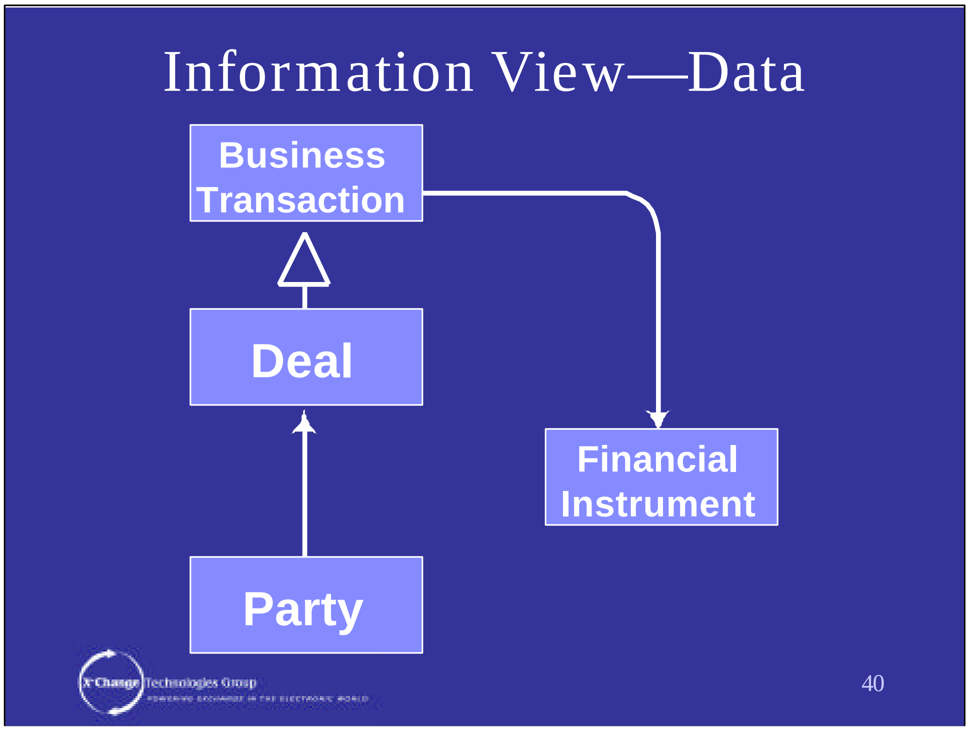

Enterprise View

One of the interactions is between the Settlement Service and other settlement agents, comparing instructions to ensure that both sides of the transaction agree on the terms. This particular interaction will come up again later in the example. It will be used to illustrate the use of MDA model transformation to detect failure of a detailed design to meet the original intent.



Information View—Data

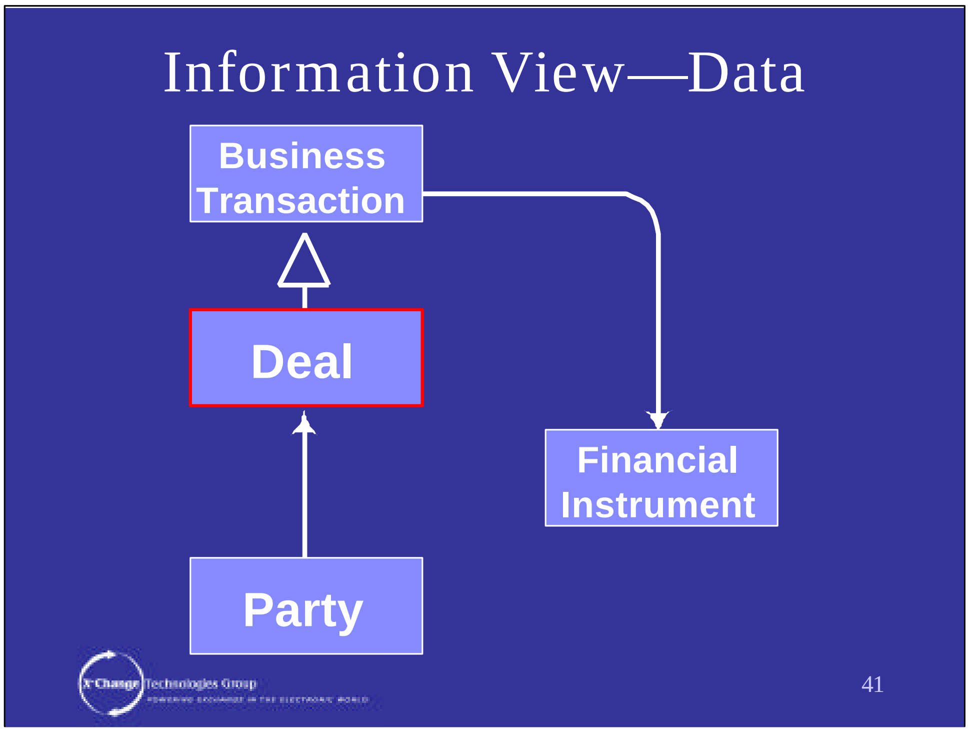
Here is a sketch of part of the specification of the system from the ODP Information viewpoint. The next slide shows a part of this model.



Information View—Data

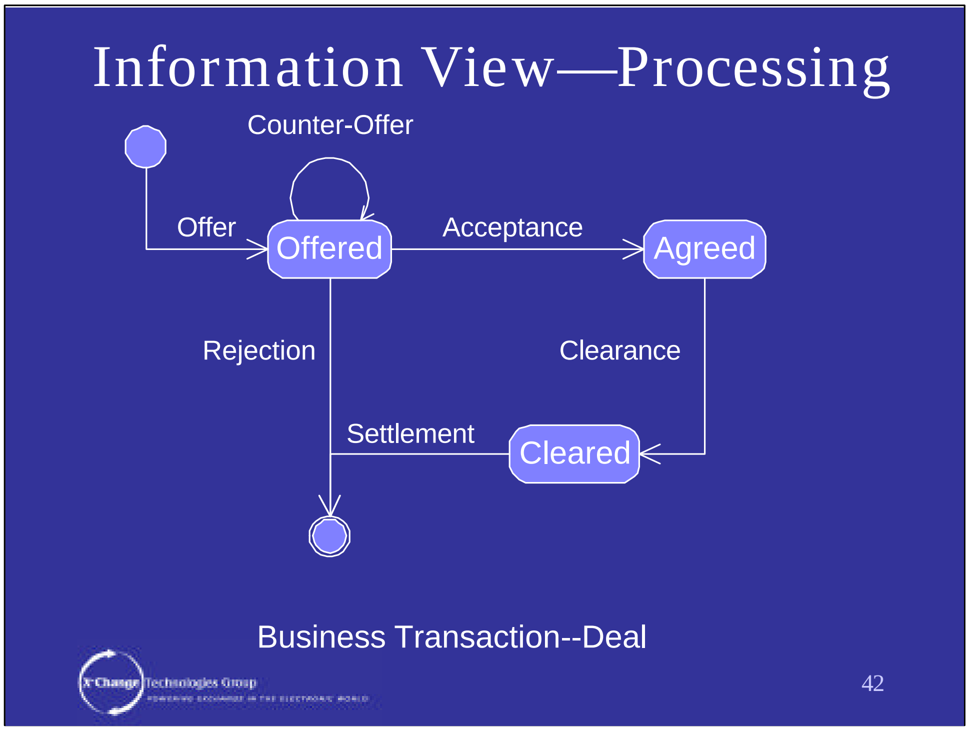
Business Transaction
Deal
Financial Instrument
Party
X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD
40

This focuses on part of the structure of the data. The ODP Information viewpoint specifies both the data, and the processing of the data. This model represents only the data, but not its processing. The Information viewpoint views the system as a monolith, not as a distributed system.



Information View—Data

If we focus on one class of information objects, the deal objects, we can add some of the specification of the processing of the data, as shown on the next slide.



Information View—Processing

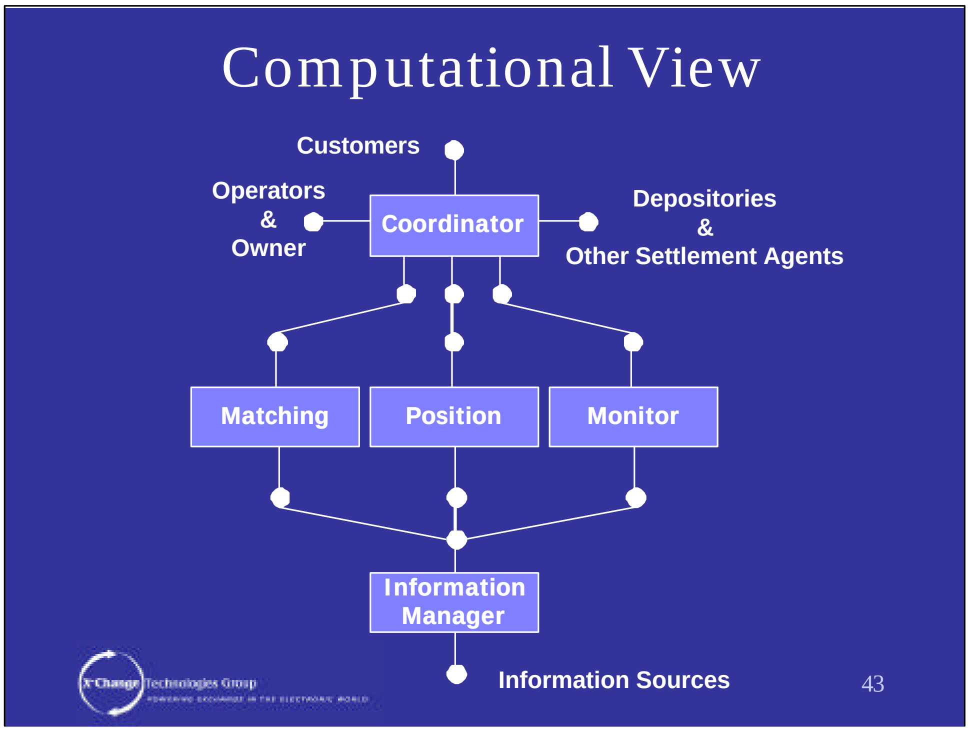
Business Transaction--Deal

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

42

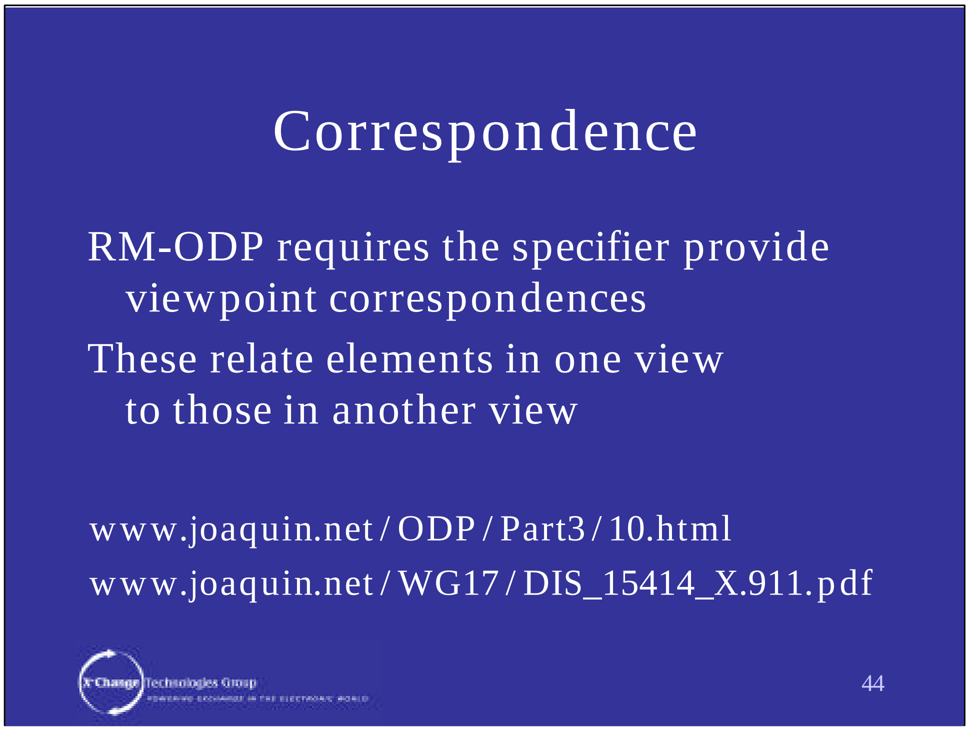
This is a state diagram of Deal information objects. It shows the lawful states of an object that represents a particular deal. The boxes represent those states. The lines represent transitions from one state to another. The names on the transitions identify the interactions that cause a state transition. For example, one possible interaction is when one party communicated a rejection to another party that made an offer. This terminates the attempted deal. On the other hand, if a party communicates acceptance, the deal becomes agreed.

Further detail can be provided using a state machine language, such as UML.



Computational View

The ODP computational viewpoint focuses on the specification of the system as objects able to interact at interfaces. Here the objects are subsystems of the Settlement Service system. These subsystems interact with each other and with systems and parties in the environment of the Settlement Service system.



Correspondence

RM-ODP requires the specifier provide viewpoint correspondences

These relate elements in one view to those in another view

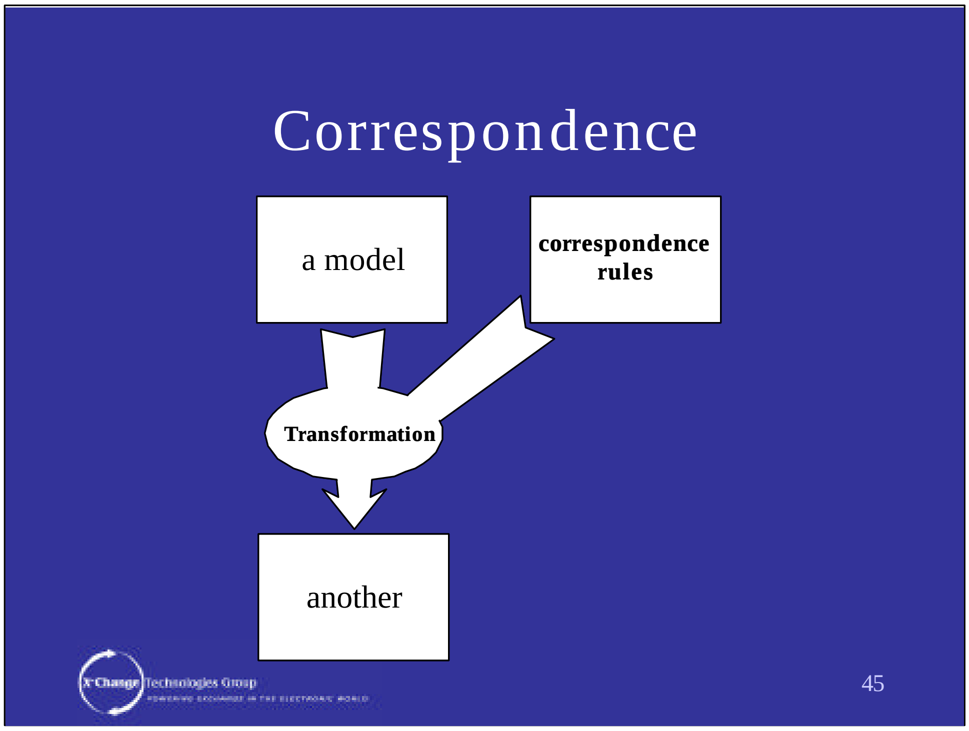
www.joaquin.net/ODP/Part3/10.html

www.joaquin.net/WG17/DIS_15414_X.911.pdf

X-Change Technologies Group

POWERING EXCHANGE IN THE ELECTRONIC WORLD

44



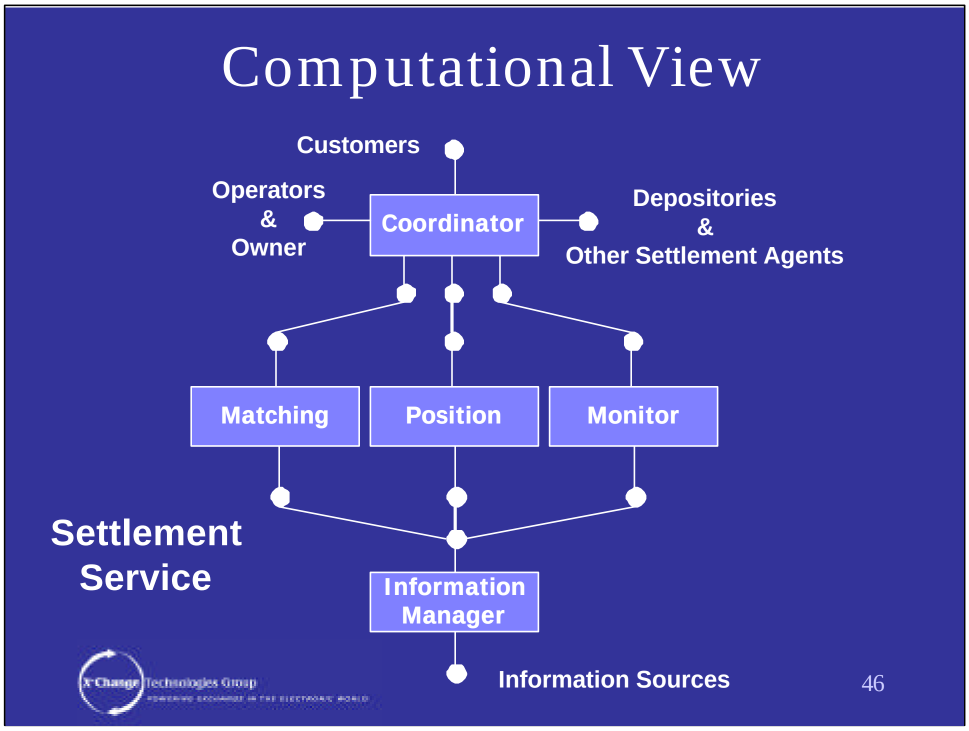
Correspondence

X-Change Technologies Group

45

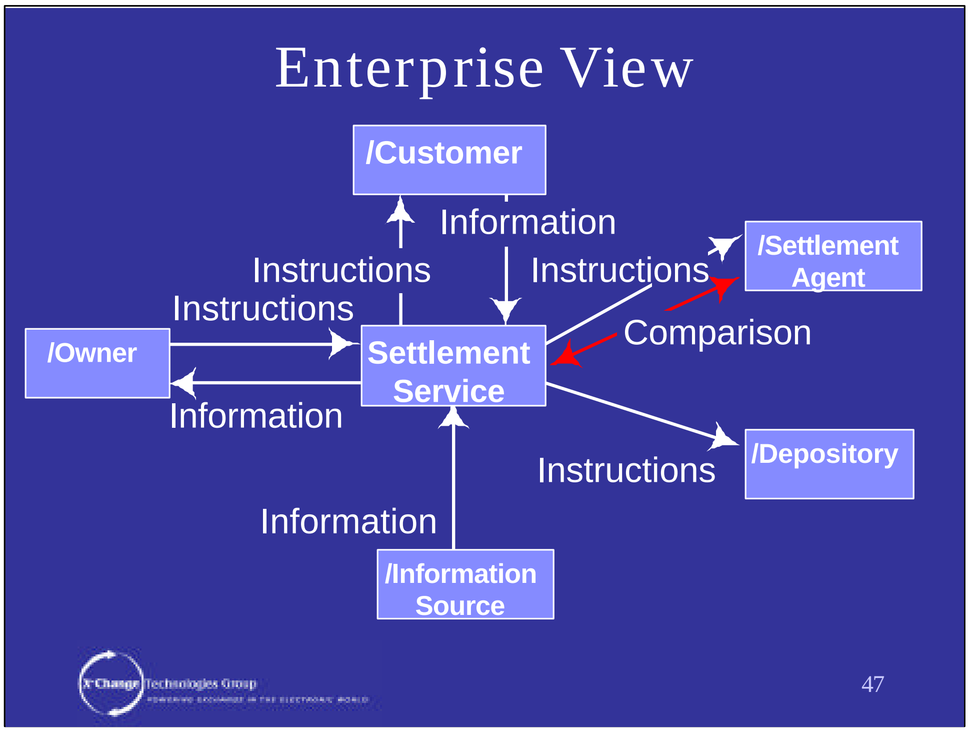
In the case with the least software support, the specifier will state the correspondences as elements of a transformation.

With more support, a tool will check that the correspondences conform to the correspondence rules.



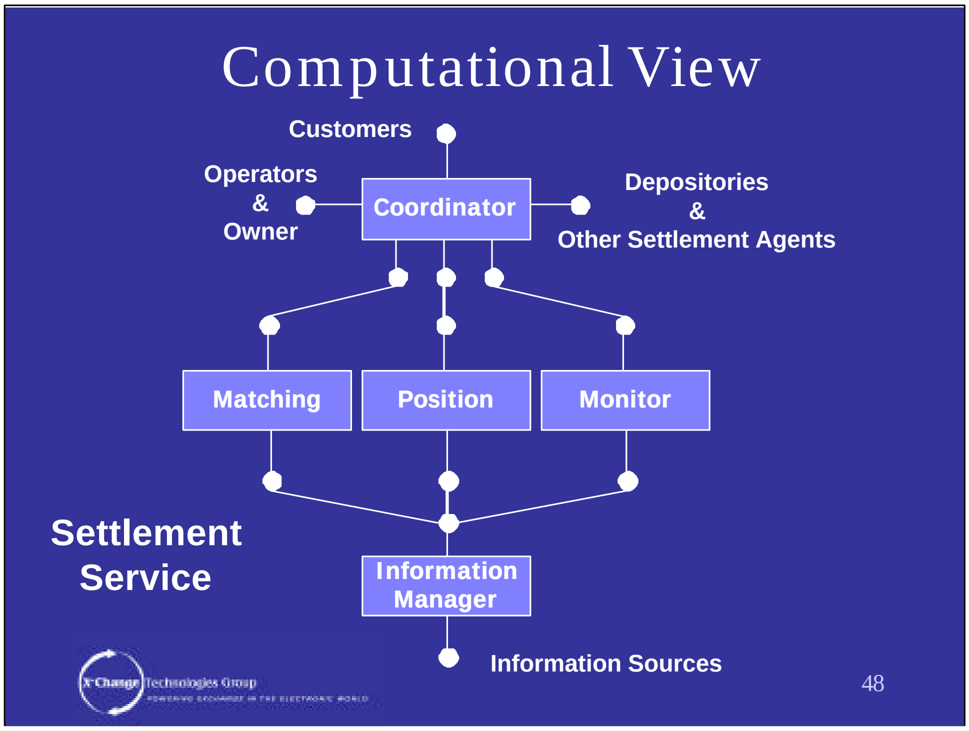
Computational View

Let’s look at the computational view of the system.



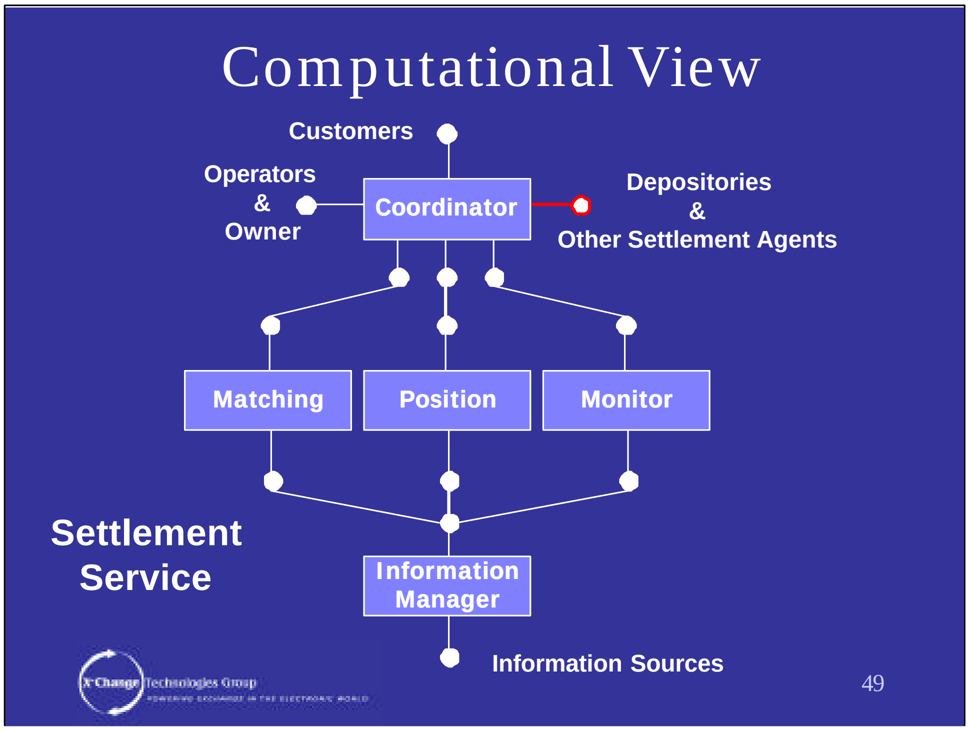
Enterprise View

And let's compare that with the Enterprise view. In this drawing the Settlement Service system appears as a single object; in the previous, Computational view, it appeared as several interacting subsystems.



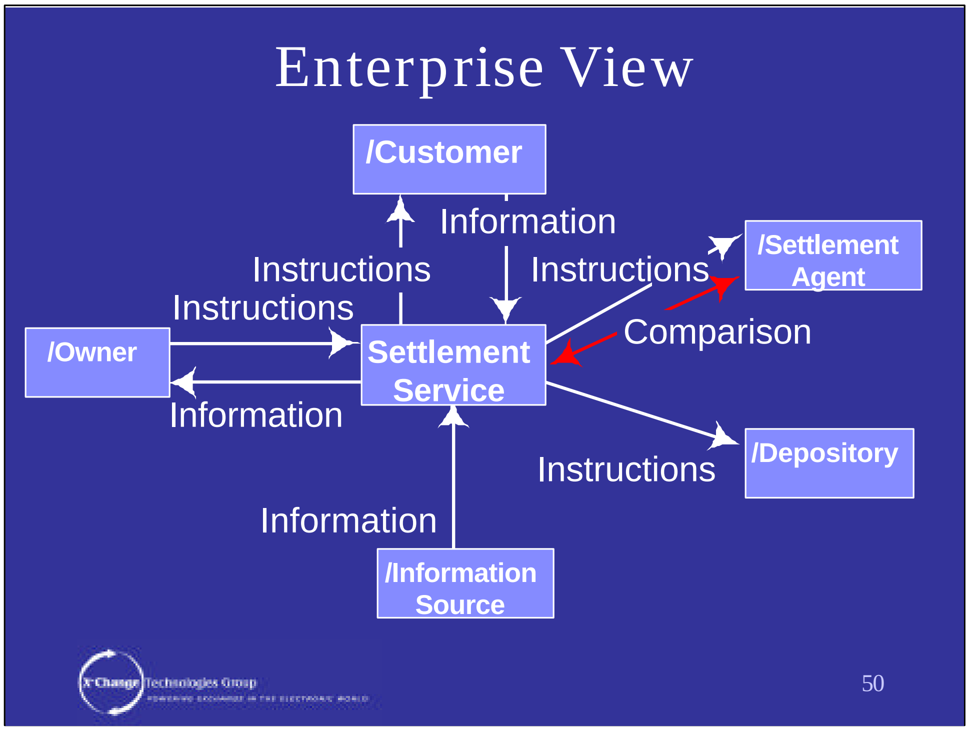
Computational View

Here is the computational view again. Do you see any discrepancy. Look back at the Enterprise view.



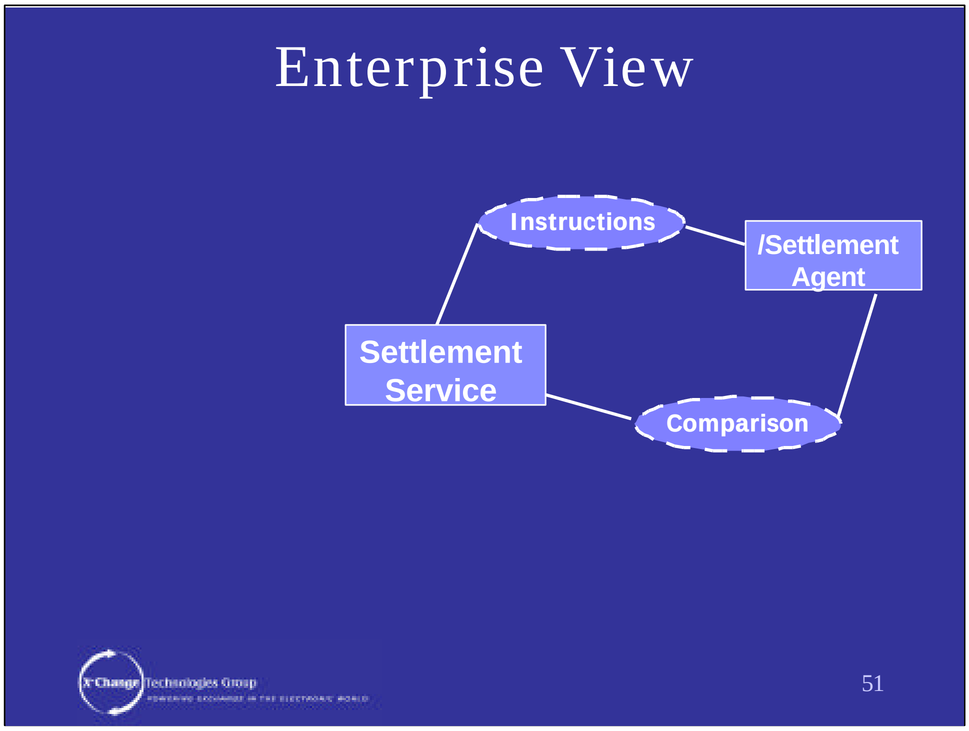
Computational View

Look at the interface on the right side of the Coordinator subsystem object. The interface that is used by other settlement agents is also used by Depositories.

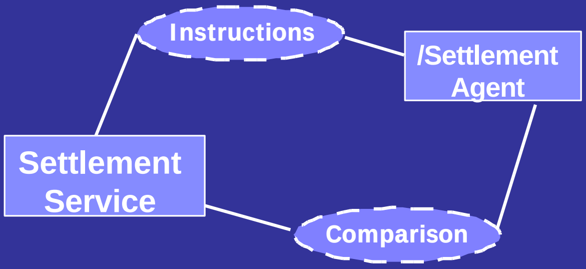


Enterprise View

But settlement agents not only receive instructions, they also collaborate in comparisons. An interface is missing from the Computational view.



Enterprise View



X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

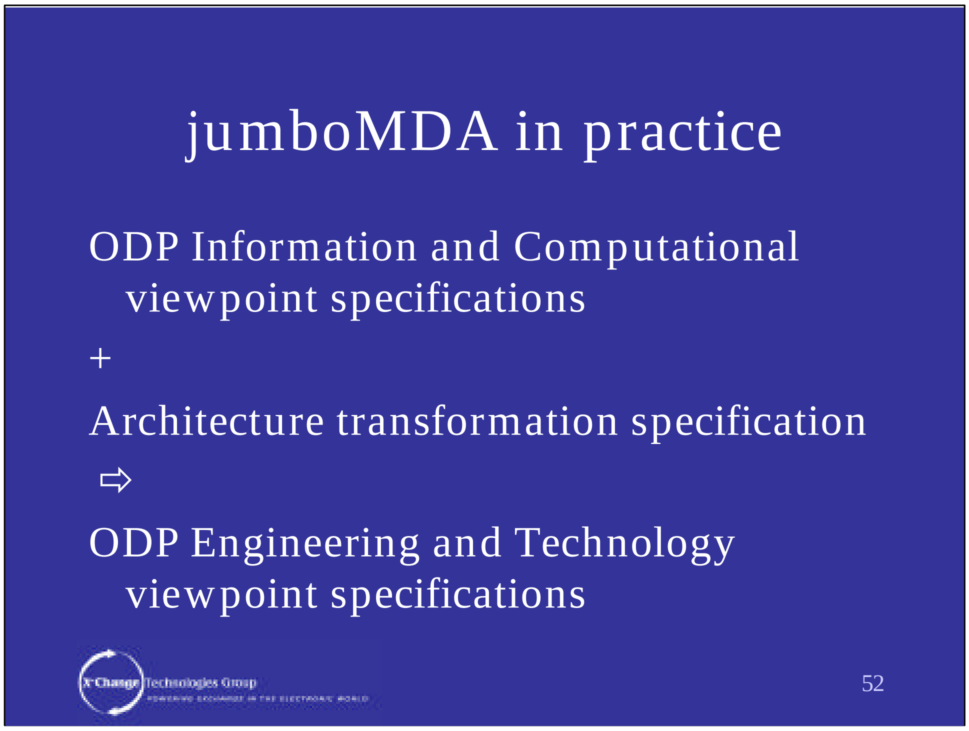
51

The notation of the previous drawings was not as expressive as it should be.

Rather than abuse the UML interaction notation, by using an arrow to represent the joint action of comparing settlement instructions, a joint action of the settlement service together with another system in the role of settlement agent , it will be much better to represent this graphically as a joint action. That is, an action of more than one object.

A deficiency of UML 2 is the absence of the concept of joint action. Here I press the UML notation for a collaboration into service. The UML 2 specification tells us “Collaborations are generally used to explain how a collection of cooperating instances achieve[s] a joint task or set of tasks.”

This use of a collaboration to represent a joint action could be explained in UML terms as using an abstract collaboration, which does not yet explain how the collection of cooperating objects achieves the joint task.



jumboMDA in practice

ODP Information and Computational viewpoint specifications

+

Architecture transformation specification

⇨

ODP Engineering and Technology viewpoint specifications

X-Change Technologies Group

52

A tool aware of viewpoints and correspondence rules might generate many of the correspondences. If the tool is following an architecture specification, that might be in the form of correspondence rules which the tool uses to both generate the correspondences and the corresponding model. This could be done by following the computation to engineering viewpoint correspondence rules of RM-ODP. Tools using the Shlaer-Mellor approach, such as the Project Technology tool, are, in effect, performing a computational and information viewpoint to engineering viewpoint transformation, using the correspondence rules in the project specific architecture specification.

jumboMDA in practice

Example:

- Component based development

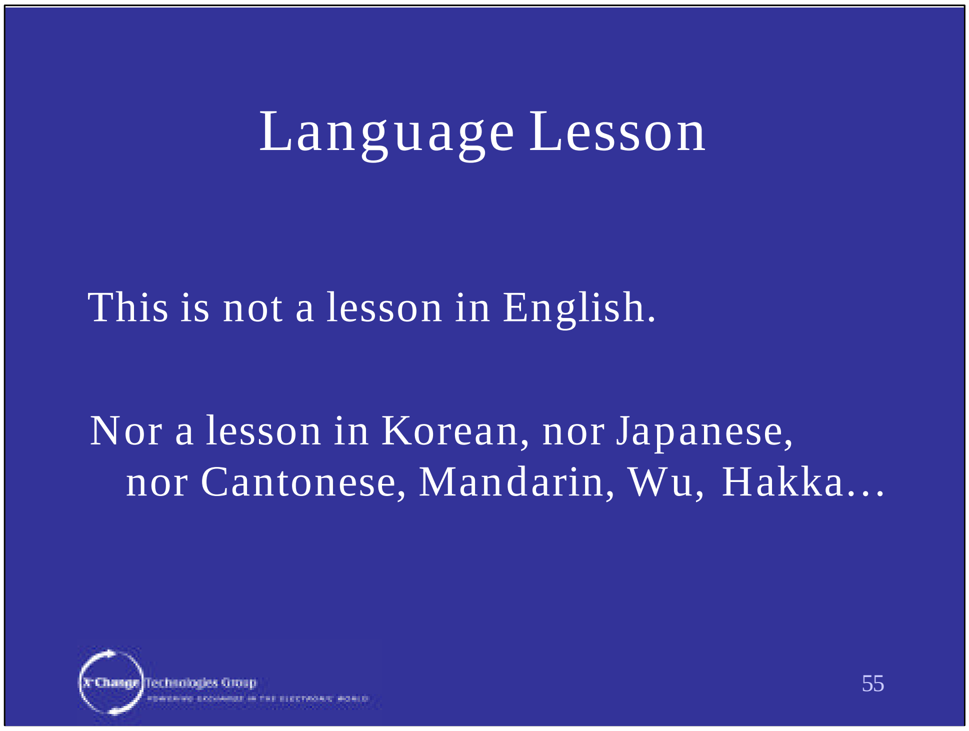
jumboMDA in practice

Example:

- Comparing component C1
  - ⇨ Component C2

to check for substitutability

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD



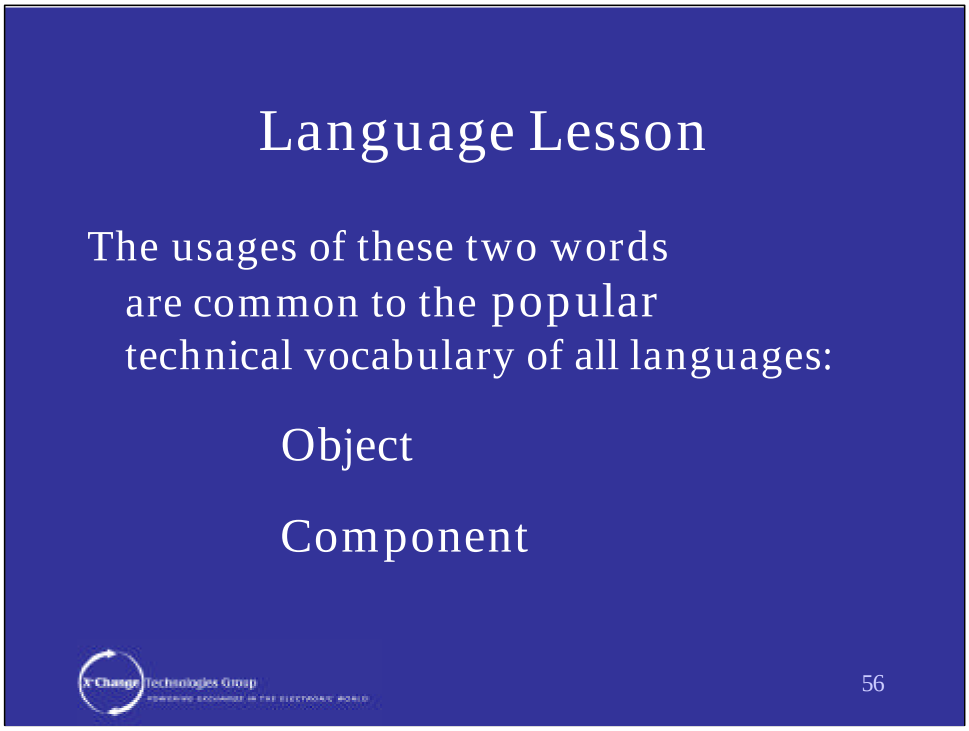
Language Lesson

This is not a lesson in English.

Nor a lesson in Korean, nor Japanese,
nor Cantonese, Mandarin, Wu, Hakka…

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

55



Language Lesson

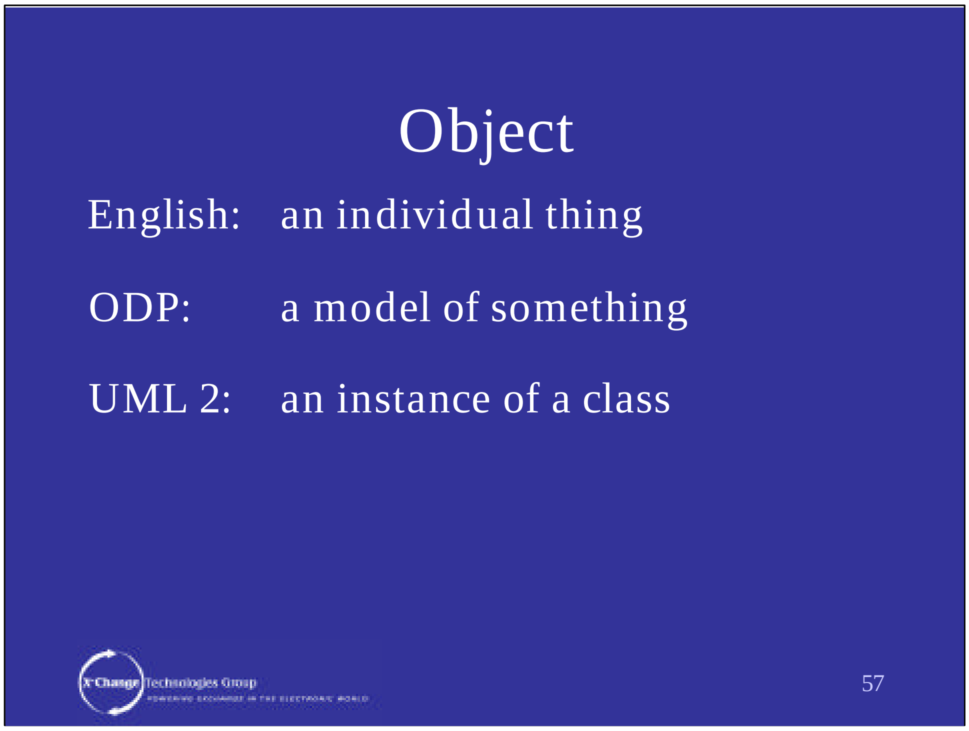
The usages of these two words are common to the popular technical vocabulary of all languages:

Object

Component

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

56



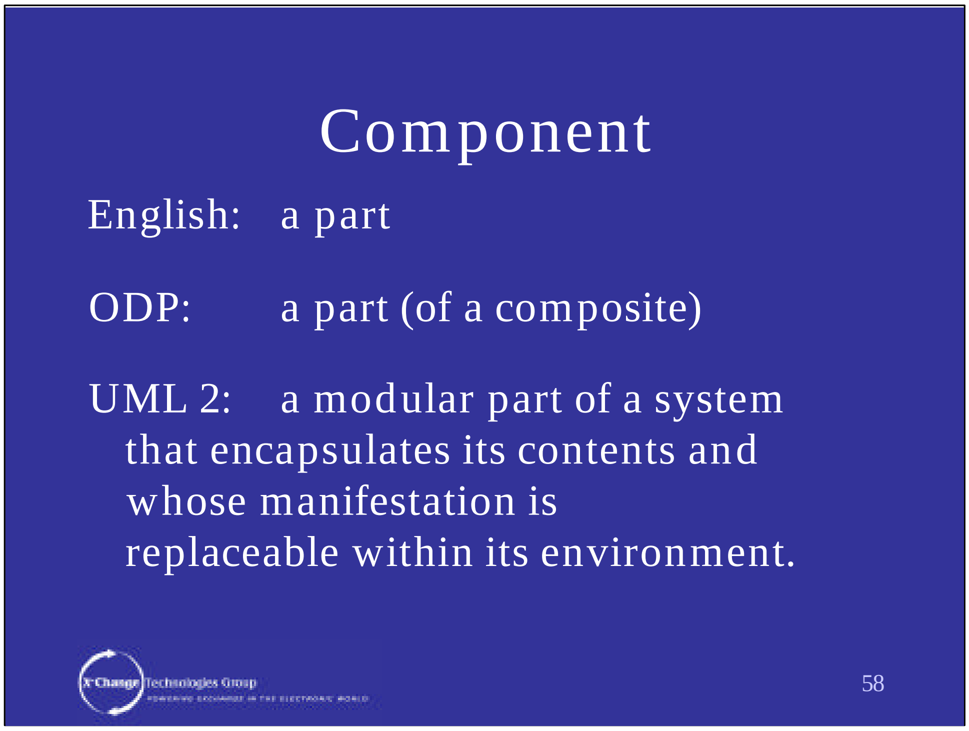
Object

English:	an individual thing
ODP:	a model of something
UML 2:	an instance of a class

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

57

UML 2 takes the meat out of the word, ‘object,’ leaving it with a much narrower and more specific meaning. Though a UML class can be used to represent anything, Class has been tailored to suit programming language classes.



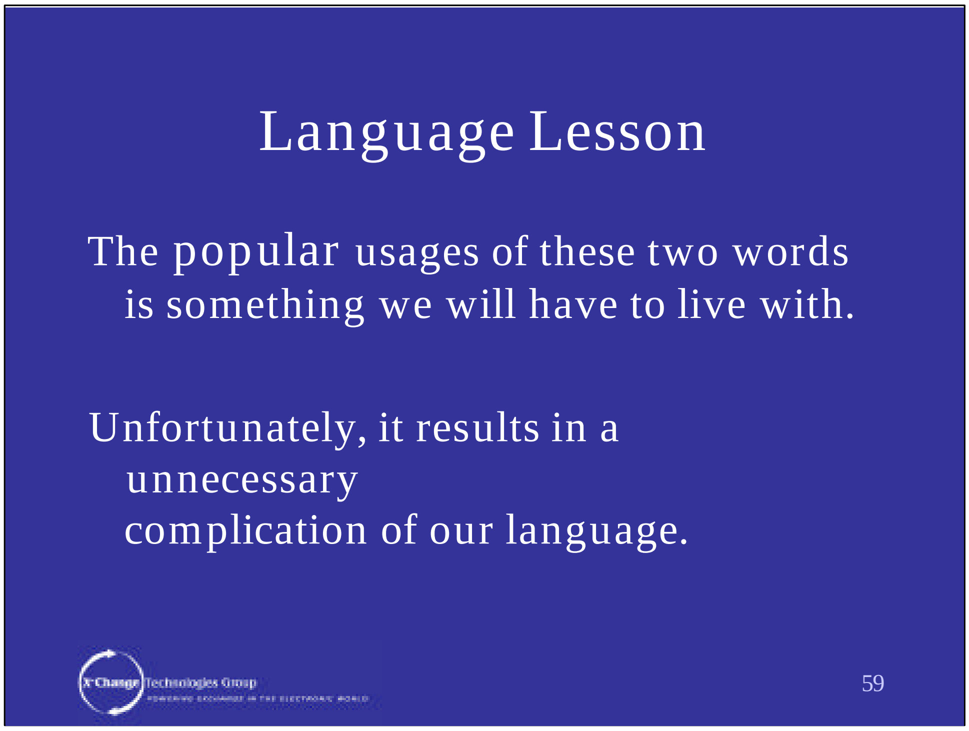
Component

English:	a part
ODP:	a part (of a composite)
UML 2:	a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment.

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

58

Component


Language Lesson

The popular usages of these two words is something we will have to live with.

Unfortunately, it results in a unnecessary complication of our language.

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD
59

This is one of the unfortunate results of UML being a class-oriented language, inspired by Ada and C++ and now driven by and focused on Java.

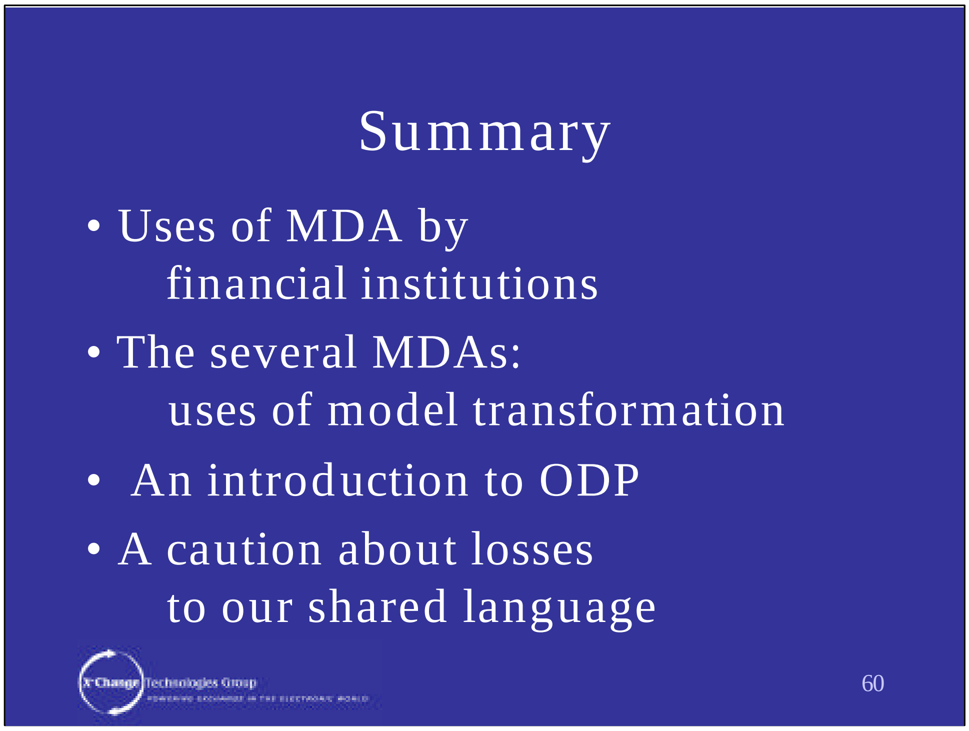
Why is this an unfortunate result? Because the architect has no general purpose model element that can be used, without choosing how a particular item in the system will be implemented. If the item will be implemented by sending new to a class, then the architect must use a class and an “instance specifications of a class.” If the item will be implemented using another technology, then the architect must use a component (or other structured classifier) and an “instance specifications of a component.”

If UML were “compatible with the architecture for system distribution defined in ISO/IEC 10746, Reference Model of Open Distributed Processing (ODP),” as required in clause 5.1.12 of the UML RFPs, then the architect could use objects (and types of objects) in the specification, and leave for later the choice of implementation technology.

A user of UML 2 can ignore restrictive statements in the text of the specification, but there is not one single UML 2 classifier that meets the requirements of a software architect for both abstraction and full general capabilities: all the classifiers given by UML 2 are specialized.

Of course, UML 2 is simply following the current popular use of ‘component’ to mean a program part in the Big Bill COM and .NET style, the Java Enterprise Edition style, and the CORBA component model, a generalization of J2EE. That’s why the meaning of the word, ‘component,’ has been changed, and that was not made by the authors of UML 2. It was made by all of us, following marketeers and the press.

The point of this lesson is to remind ourselves that it is a loss to our languages



Summary

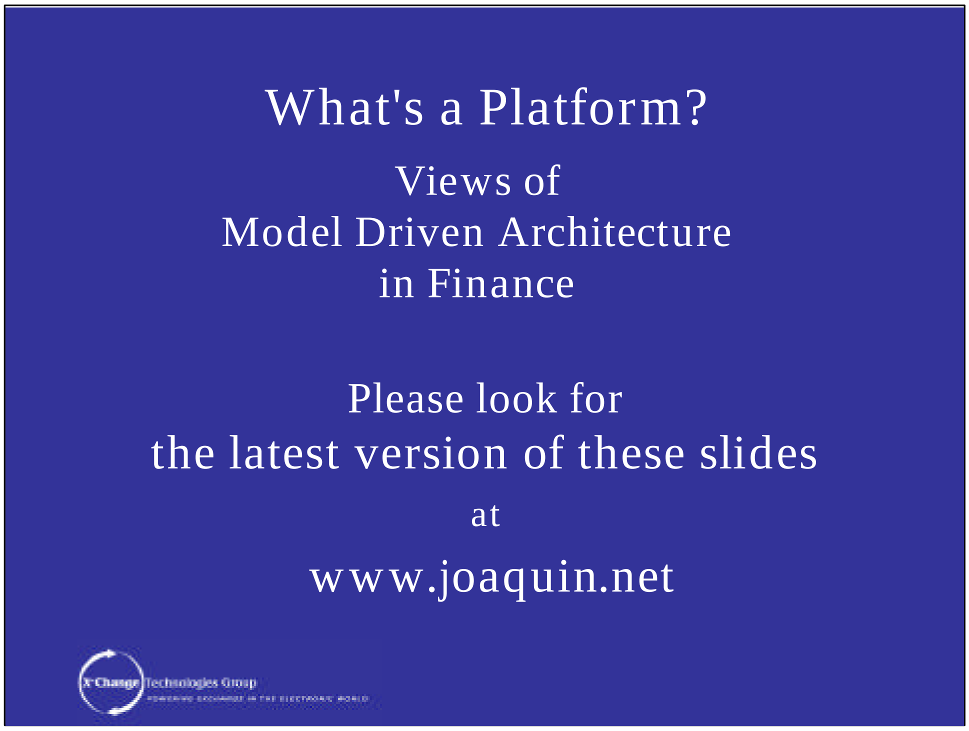
- Uses of MDA by
  financial institutions
- The several MDAs:
  uses of model transformation
- An introduction to ODP
- A caution about losses
  to our shared language

X-Change Technologies Group

POWERING EXCHANGE IN THE ELECTRONIC WORLD

60

We slipped into the outline a mini-tutorial on ODP, just a taste.



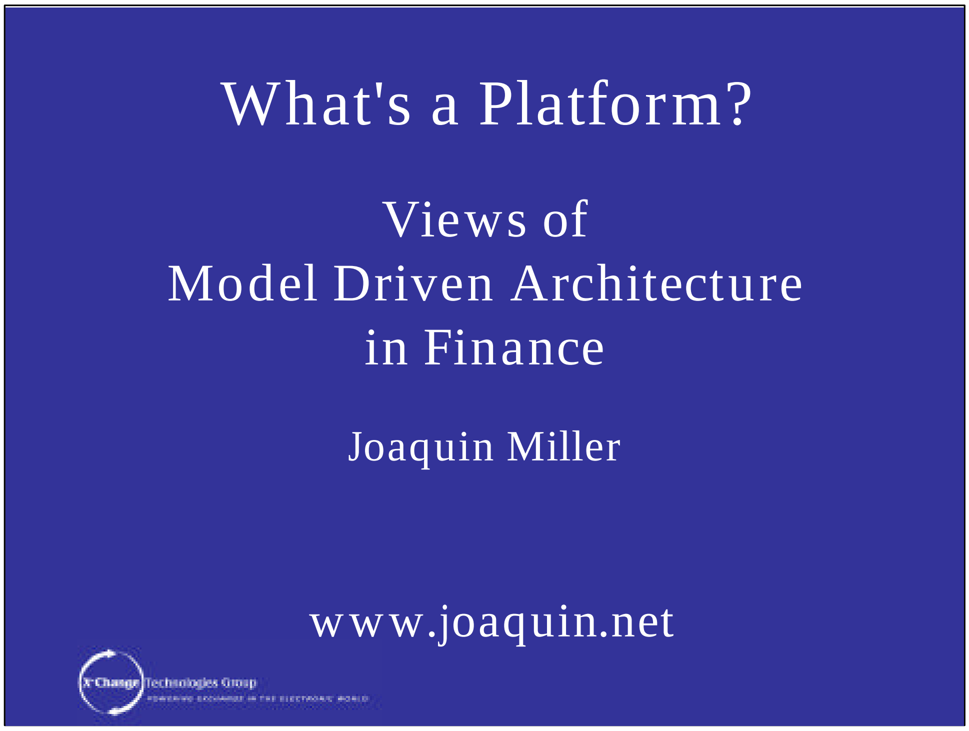
What's a Platform?

Views of
Model Driven Architecture
in Finance

Please look for
the latest version of these slides
at
www.joaquin.net

X-Change Technologies Group

After the BOSC 2003 Keynote in Seoul, and based on your questions and feedback, some changes may be made in these slides. The revised slides will be posted at www.joaquin.net/presentations



What's a Platform?

Views of
Model Driven Architecture
in Finance

Joaquin Miller

www.joaquin.net

X-Change Technologies Group
POWERING EXCHANGE IN THE ELECTRONIC WORLD

Thank you very much for your invitation and your attention today.

Cordially,

Joaquin