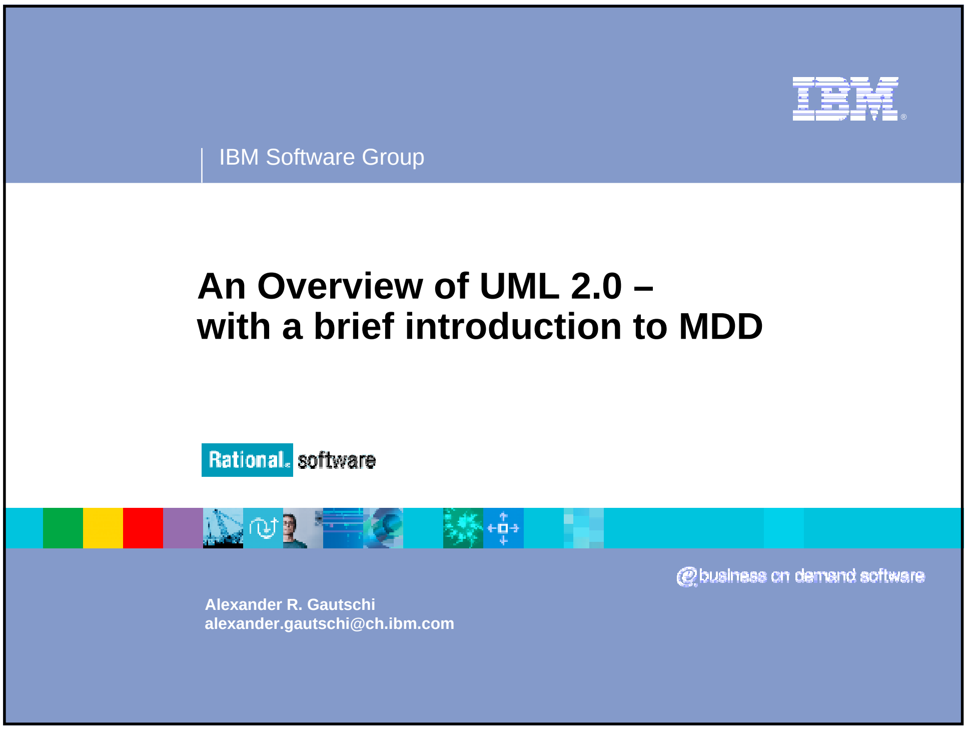


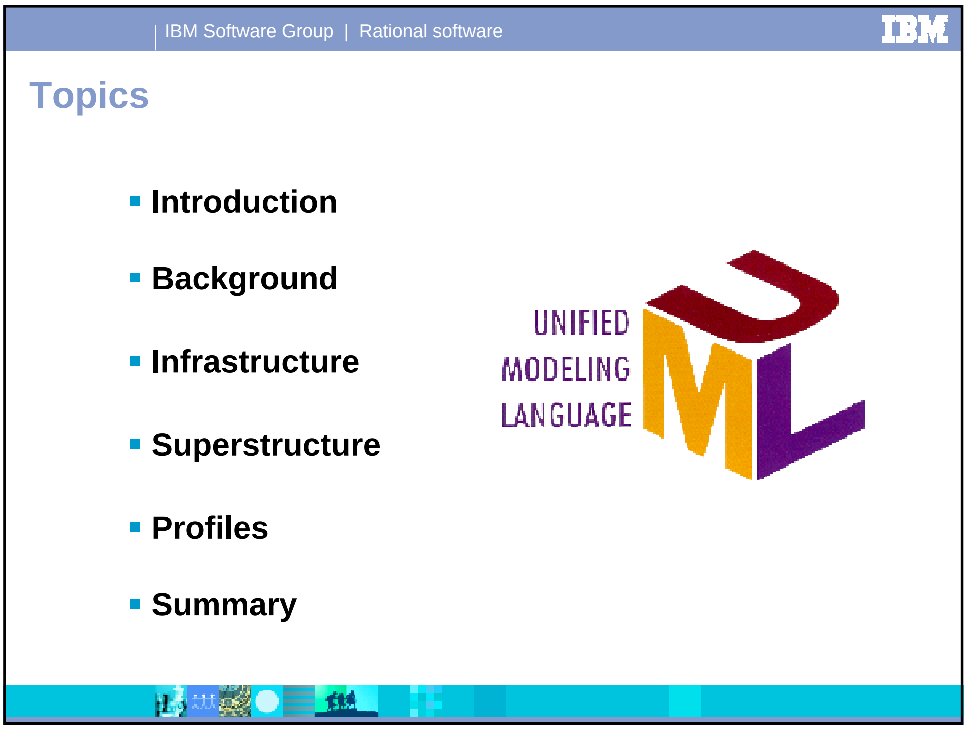
IBM Software Group

An Overview of UML 2.0 – with a brief introduction to MDD

Rational software

Alexander R. Gautschi
alexander.gautschi@ch.ibm.com

@business on demand software

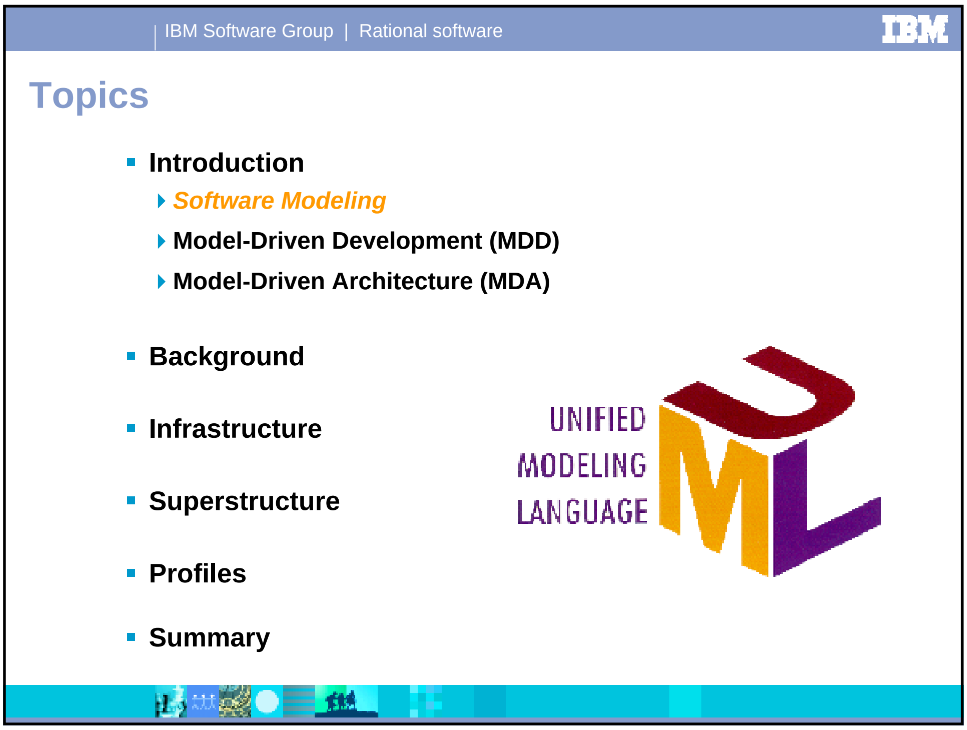


IBM Software Group | Rational software

Topics

- Introduction
- Background
- Infrastructure
- Superstructure
- Profiles
- Summary

UNIFIED
MODELING
LANGUAGE

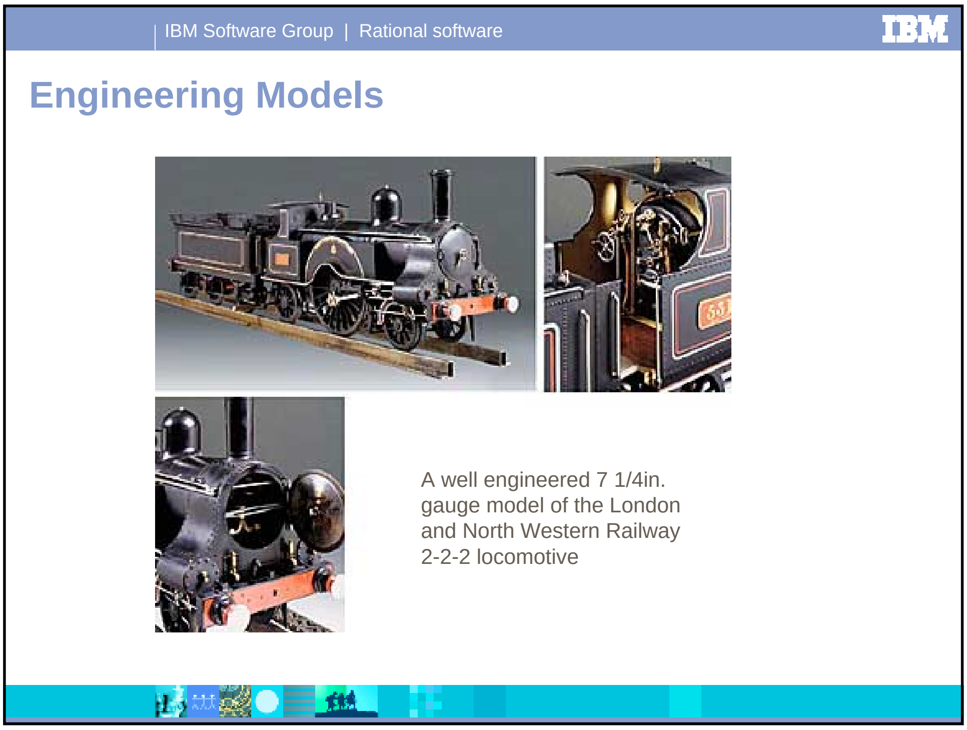


IBM Software Group | Rational software IBM

Topics

- Introduction
  - *Software Modeling*
  - Model-Driven Development (MDD)
  - Model-Driven Architecture (MDA)
- Background
- Infrastructure
- Superstructure
- Profiles
- Summary

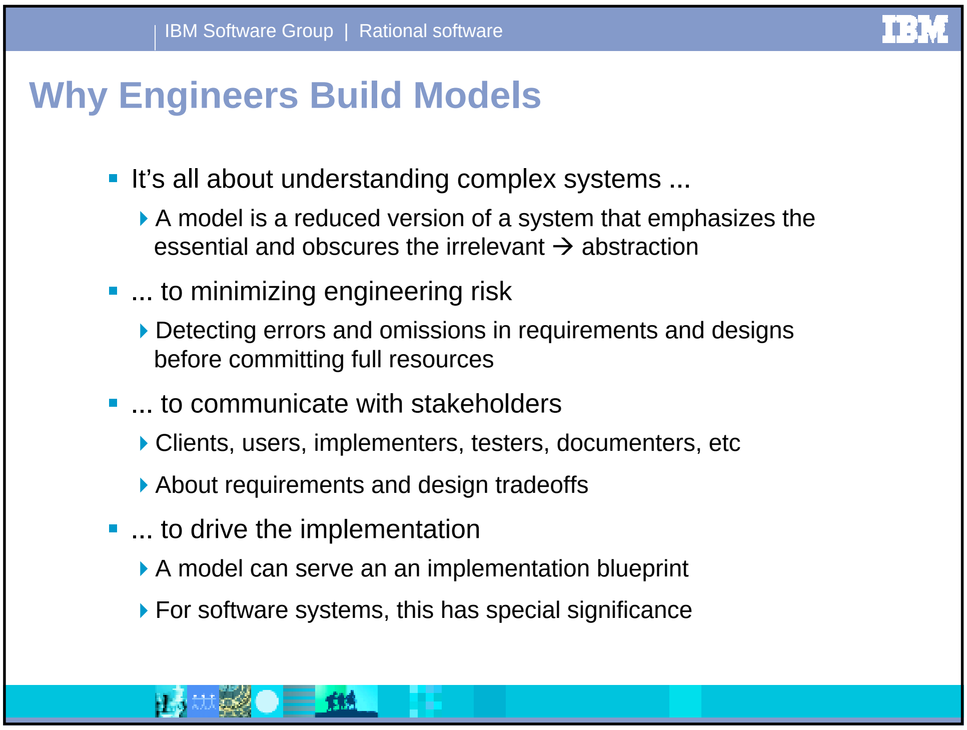
UNIFIED
MODELING
LANGUAGE



IBM Software Group | Rational software IBM

Engineering Models

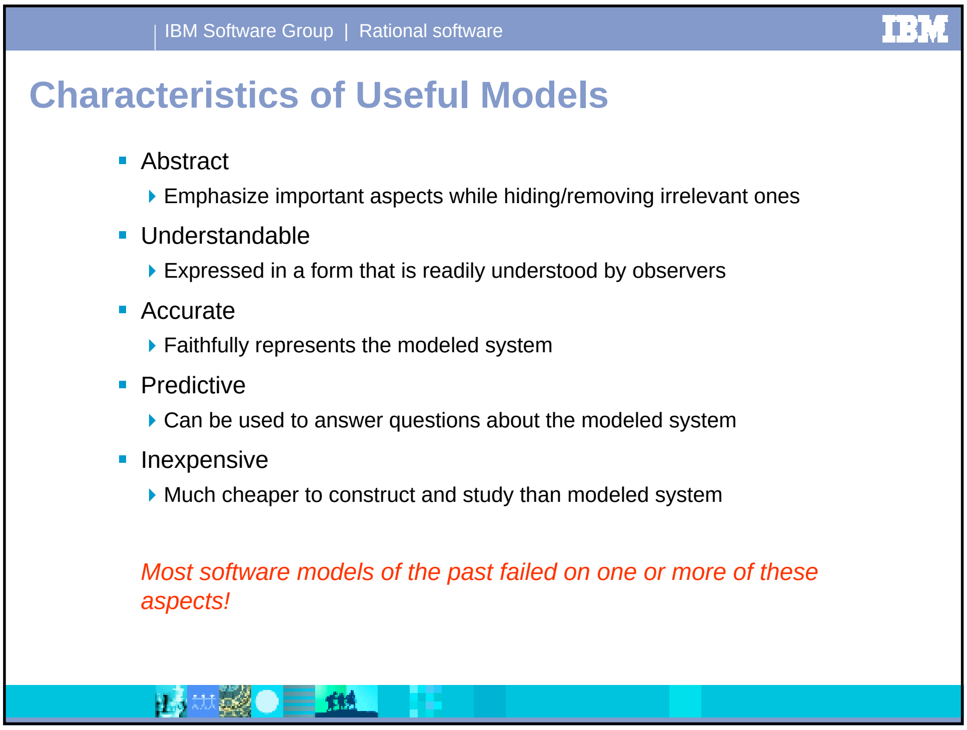
A well engineered 7 1/4in. gauge model of the London and North Western Railway 2-2-2 locomotive



IBM Software Group | Rational software

Why Engineers Build Models

- It’s all about understanding complex systems ...
  - A model is a reduced version of a system that emphasizes the essential and obscures the irrelevant → abstraction
- ... to minimizing engineering risk
  - Detecting errors and omissions in requirements and designs before committing full resources
- ... to communicate with stakeholders
  - Clients, users, implementers, testers, documenters, etc
  - About requirements and design tradeoffs
- ... to drive the implementation
  - A model can serve an an implementation blueprint
  - For software systems, this has special significance

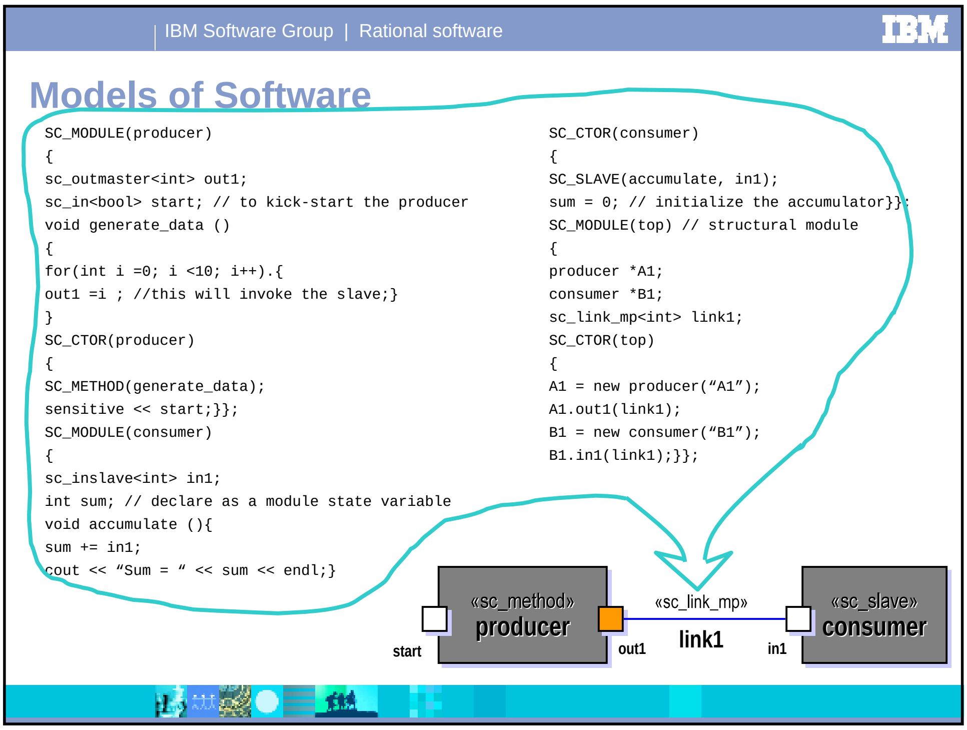


IBM Software Group | Rational software

Characteristics of Useful Models

- Abstract
  - Emphasize important aspects while hiding/removing irrelevant ones
- Understandable
  - Expressed in a form that is readily understood by observers
- Accurate
  - Faithfully represents the modeled system
- Predictive
  - Can be used to answer questions about the modeled system
- Inexpensive
  - Much cheaper to construct and study than modeled system

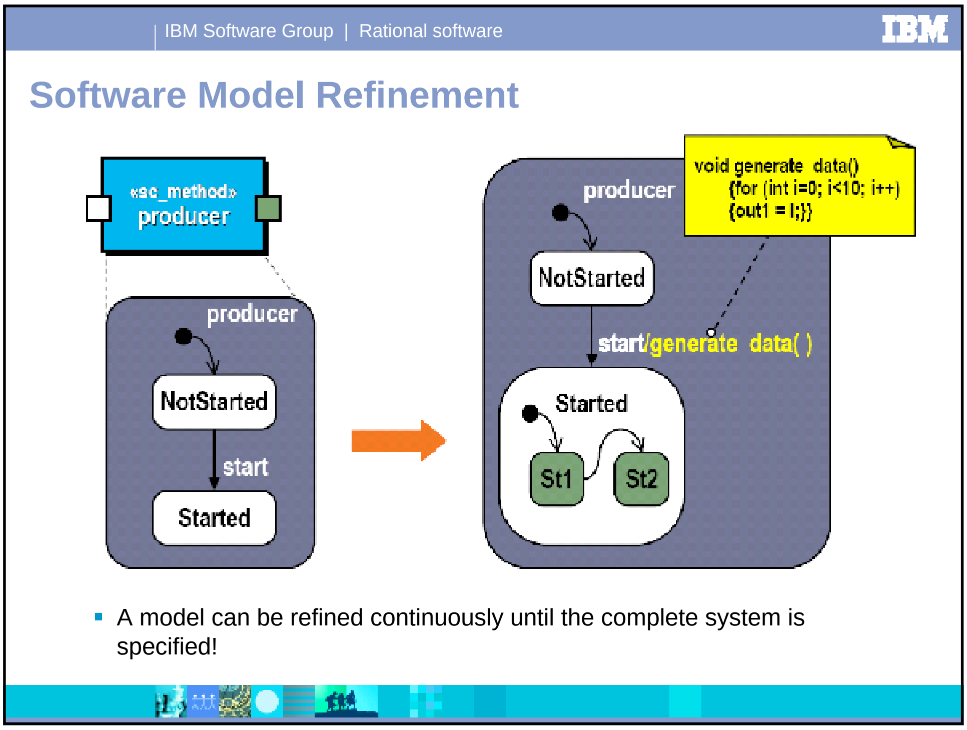
Most software models of the past failed on one or more of these aspects!



IBM Software Group | Rational software

Models of Software

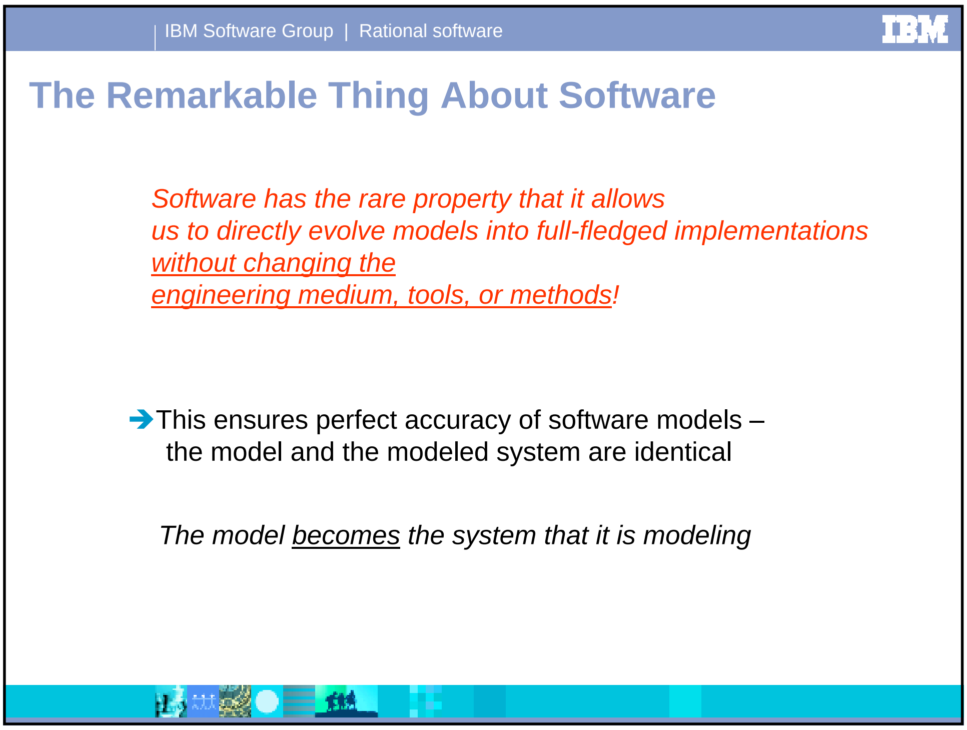
```
SC_MODULE(producer)
{
sc_outmaster<int> out1;
sc_in<bool> start; // to kick-start the producer
void generate_data ()
{
for(int i =0; i <10; i++).{
out1 =i ; //this will invoke the slave;}
}
SC_CTOR(producer)
{
SC_METHOD(generate_data);
sensitive << start;}};
SC_MODULE(consumer)
{
sc_inslave<int> in1;
int sum; // declare as a module state variable
void accumulate (){
sum += in1;
cout << “Sum = “ << sum << endl;}
SC_CTOR(consumer)
{
SC_SLAVE(accumulate, in1);
sum = 0; // initialize the accumulator}};
SC_MODULE(top) // structural module
{
producer *A1;
consumer *B1;
sc_link_mp<int> link1;
SC_CTOR(top)
{
A1 = new producer(“A1”);
A1.out1(link1);
B1 = new consumer(“B1”);
B1.in1(link1);}};
```



IBM Software Group | Rational software

Software Model Refinement

- A model can be refined continuously until the complete system is specified!



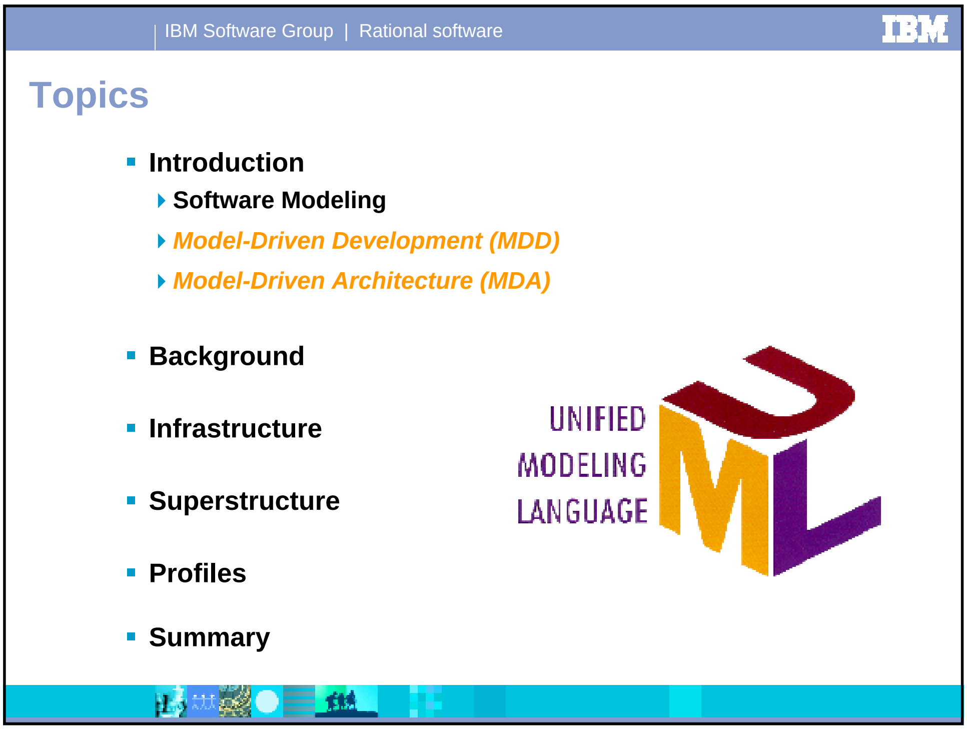
IBM Software Group | Rational software

The Remarkable Thing About Software

Software has the rare property that it allows
us to directly evolve models into full-fledged implementations
without changing the
engineering medium, tools, or methods!

➔This ensures perfect accuracy of software models – the model and the modeled system are identical

The model becomes the system that it is modeling

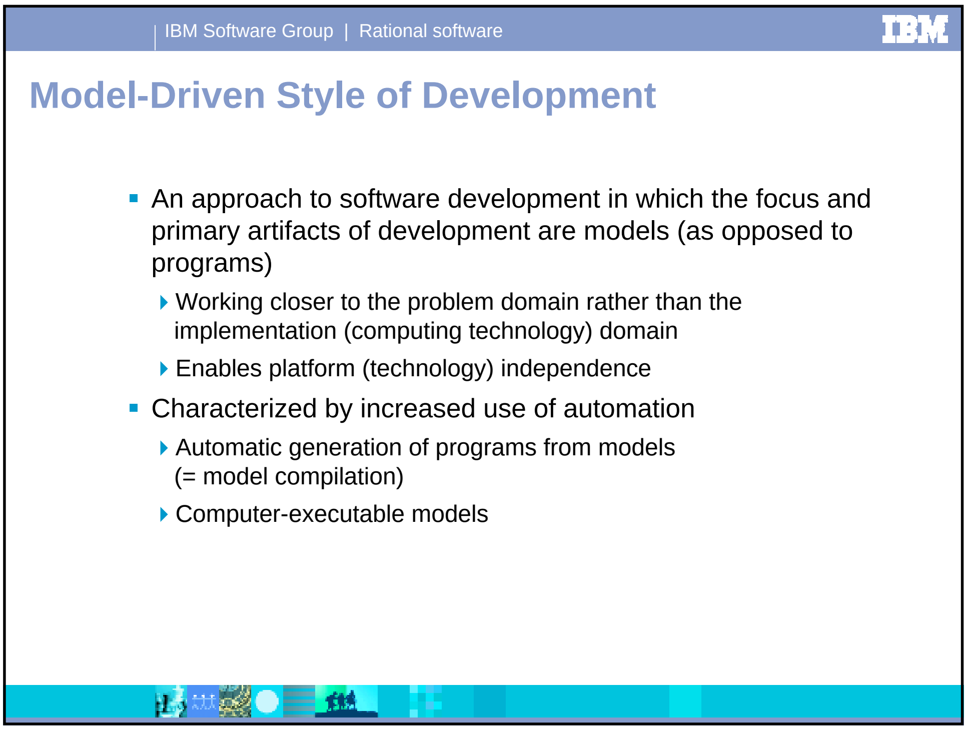


IBM Software Group | Rational software

Topics

- **Introduction**
  - **Software Modeling**
  - ***Model-Driven Development (MDD)***
  - ***Model-Driven Architecture (MDA)***
- **Background**
- **Infrastructure**
- **Superstructure**
- **Profiles**
- **Summary**

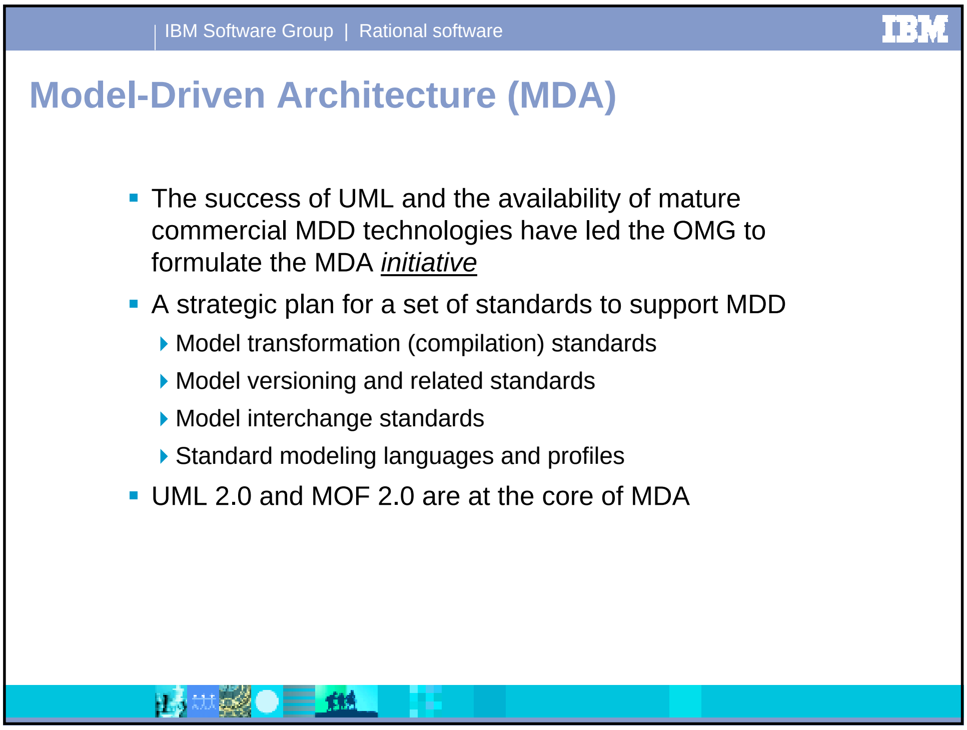
UNIFIED
MODELING
LANGUAGE
UML



IBM Software Group | Rational software

Model-Driven Style of Development

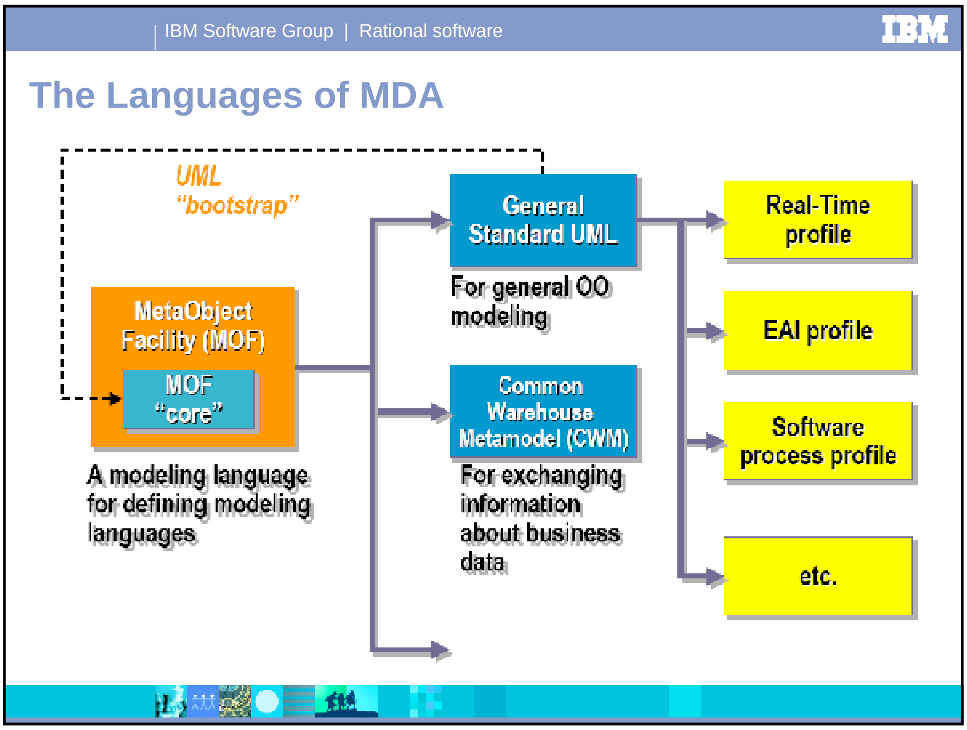
- An approach to software development in which the focus and primary artifacts of development are models (as opposed to programs)
  - Working closer to the problem domain rather than the implementation (computing technology) domain
  - Enables platform (technology) independence
- Characterized by increased use of automation
  - Automatic generation of programs from models (= model compilation)
  - Computer-executable models



IBM Software Group | Rational software

Model-Driven Architecture (MDA)

- The success of UML and the availability of mature commercial MDD technologies have led the OMG to formulate the MDA *initiative*
- A strategic plan for a set of standards to support MDD
  - Model transformation (compilation) standards
  - Model versioning and related standards
  - Model interchange standards
  - Standard modeling languages and profiles
- UML 2.0 and MOF 2.0 are at the core of MDA



IBM Software Group | Rational software

The Languages of MDA

UML "bootstrap"

MetaObject Facility (MOF)

MOF "core"

A modeling language for defining modeling languages

General Standard UML

For general OO modeling

Common Warehouse Metamodel (CWM)

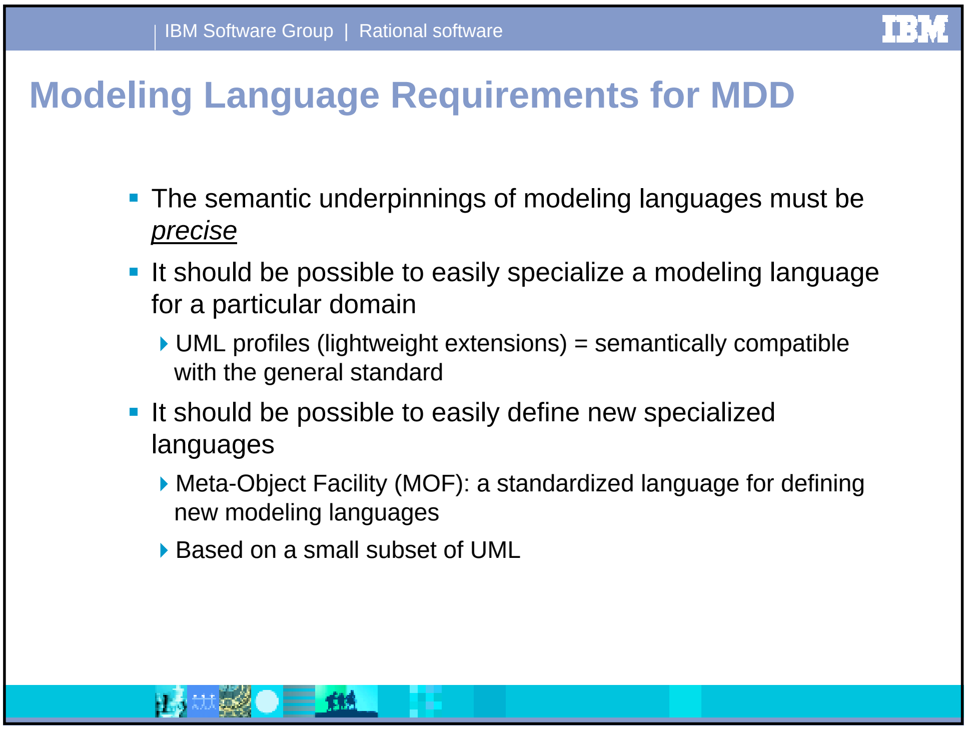
For exchanging information about business data

Real-Time profile

EAI profile

Software process profile

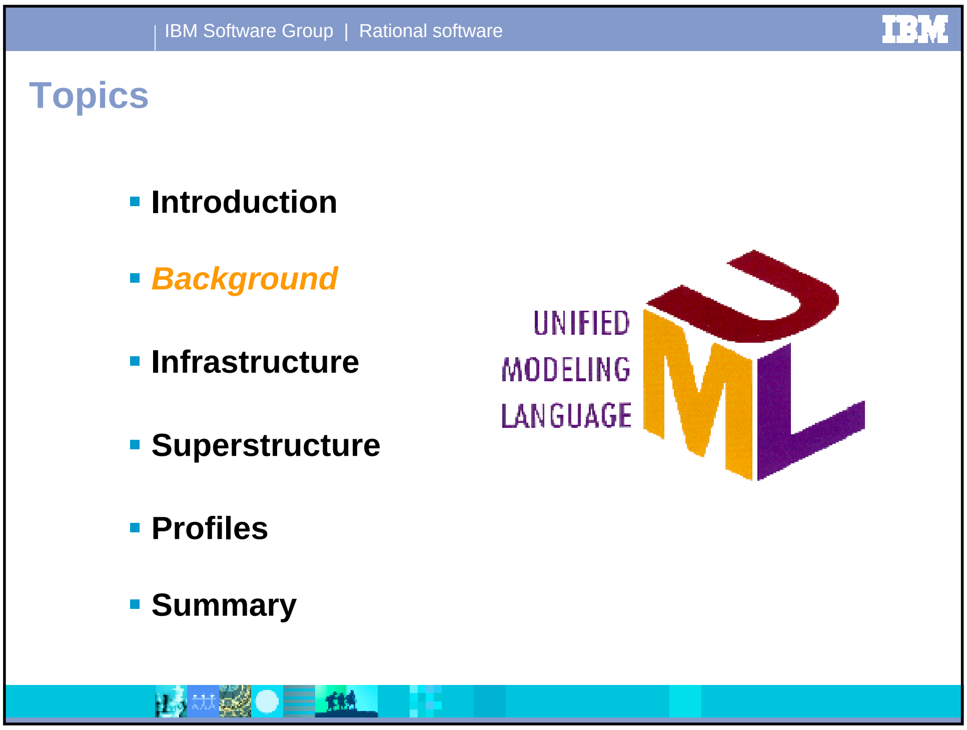
etc.



IBM Software Group | Rational software

Modeling Language Requirements for MDD

- The semantic underpinnings of modeling languages must be *precise*
- It should be possible to easily specialize a modeling language for a particular domain
  - UML profiles (lightweight extensions) = semantically compatible with the general standard
- It should be possible to easily define new specialized languages
  - Meta-Object Facility (MOF): a standardized language for defining new modeling languages
  - Based on a small subset of UML

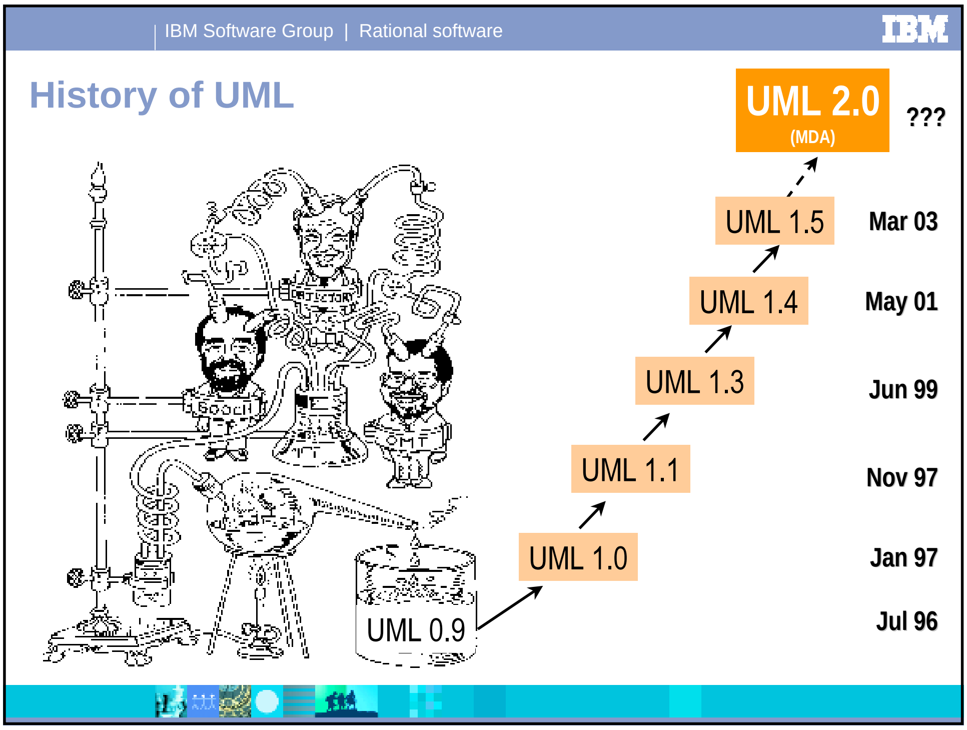


IBM Software Group | Rational software

Topics

- Introduction
- *Background*
- Infrastructure
- Superstructure
- Profiles
- Summary

UNIFIED
MODELING
LANGUAGE



IBM Software Group | Rational software

History of UML

UML 2.0 (MDA) ???

UML 1.5 Mar 03

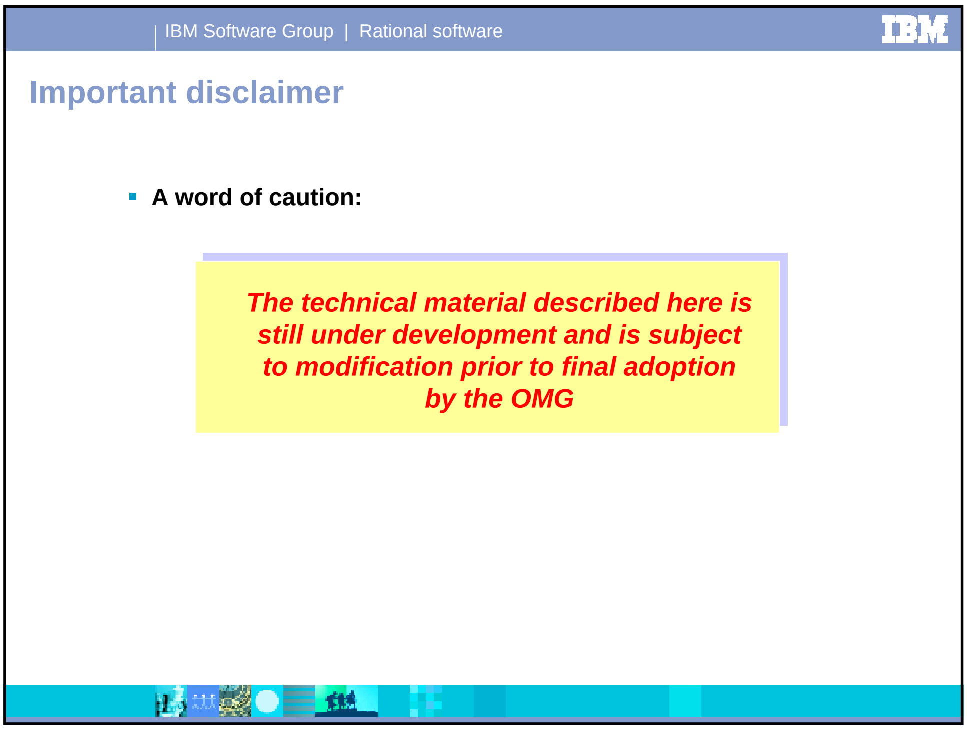
UML 1.4 May 01

UML 1.3 Jun 99

UML 1.1 Nov 97

UML 1.0 Jan 97

UML 0.9 Jul 96

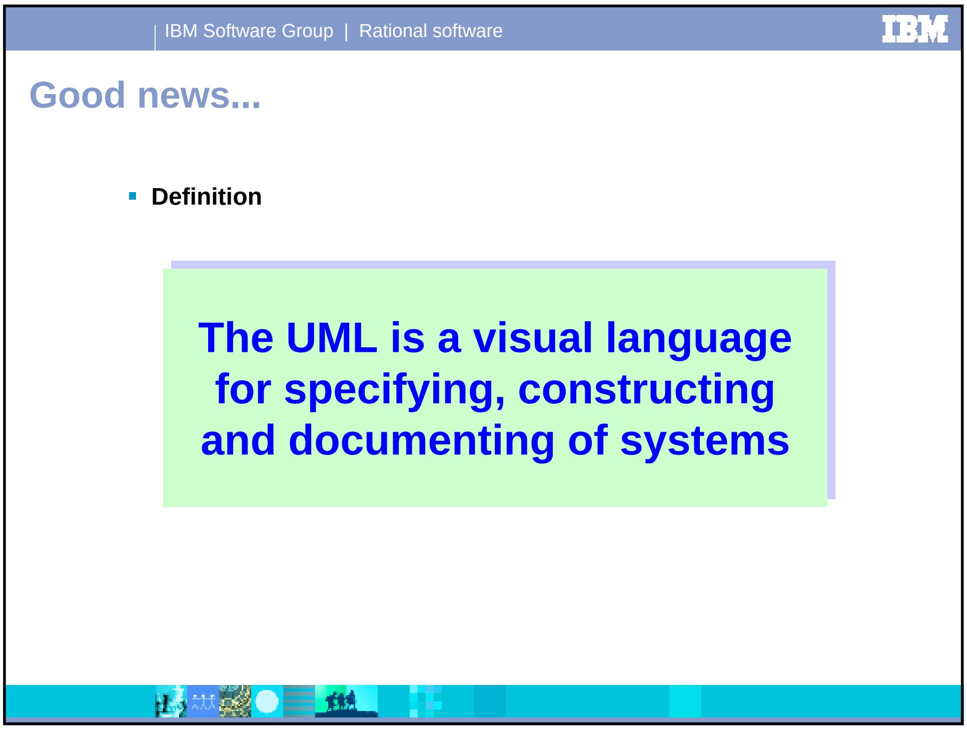


IBM Software Group | Rational software

Important disclaimer

- **A word of caution:**

The technical material described here is still under development and is subject to modification prior to final adoption by the OMG

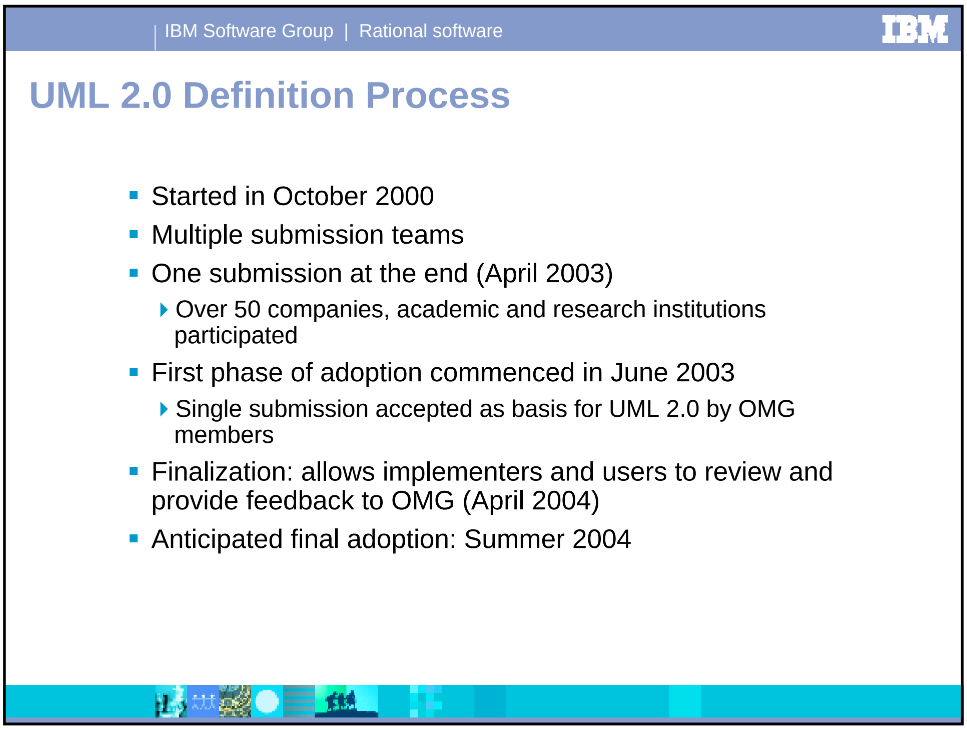


IBM Software Group | Rational software

Good news...

- **Definition**

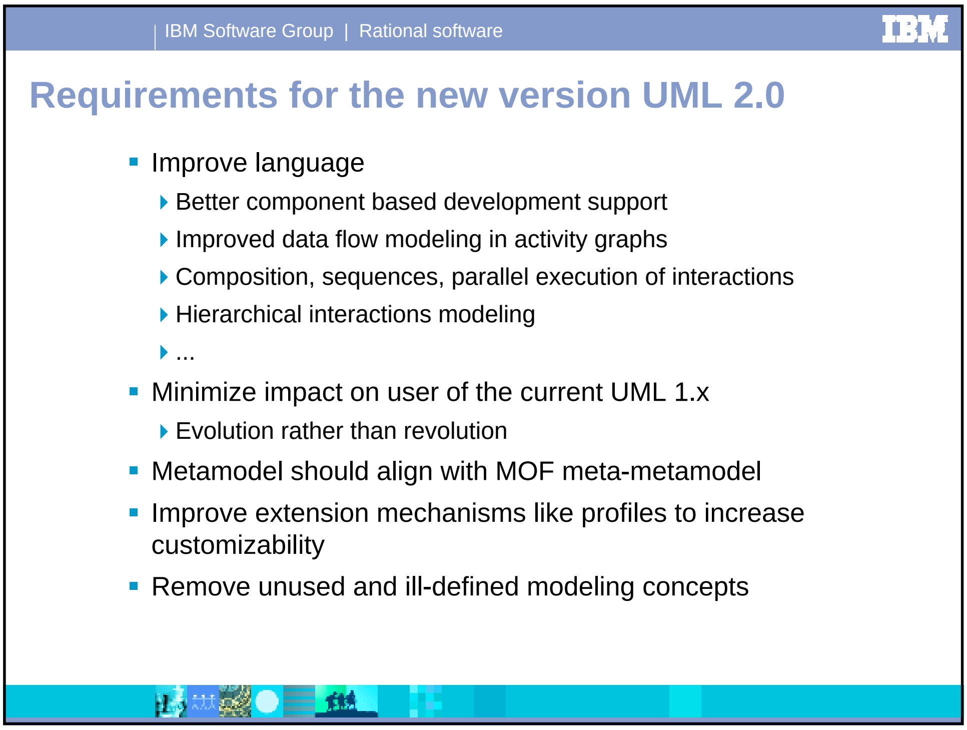
The UML is a visual language for specifying, constructing and documenting of systems



IBM Software Group | Rational software

UML 2.0 Definition Process

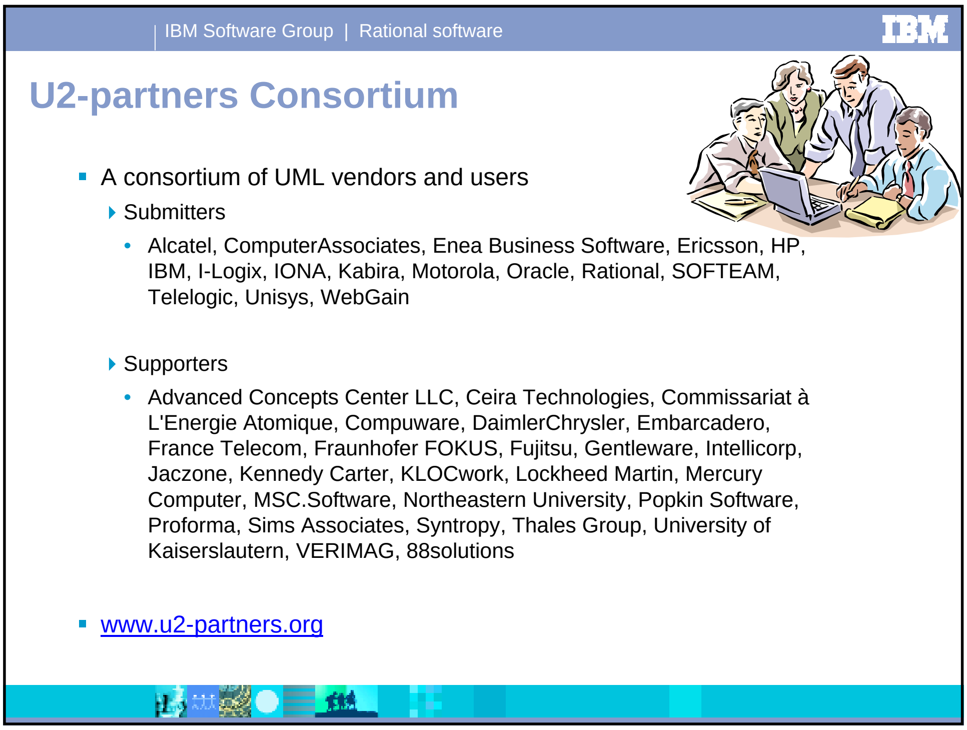
- Started in October 2000
- Multiple submission teams
- One submission at the end (April 2003)
  - Over 50 companies, academic and research institutions participated
- First phase of adoption commenced in June 2003
  - Single submission accepted as basis for UML 2.0 by OMG members
- Finalization: allows implementers and users to review and provide feedback to OMG (April 2004)
- Anticipated final adoption: Summer 2004



IBM Software Group | Rational software

Requirements for the new version UML 2.0

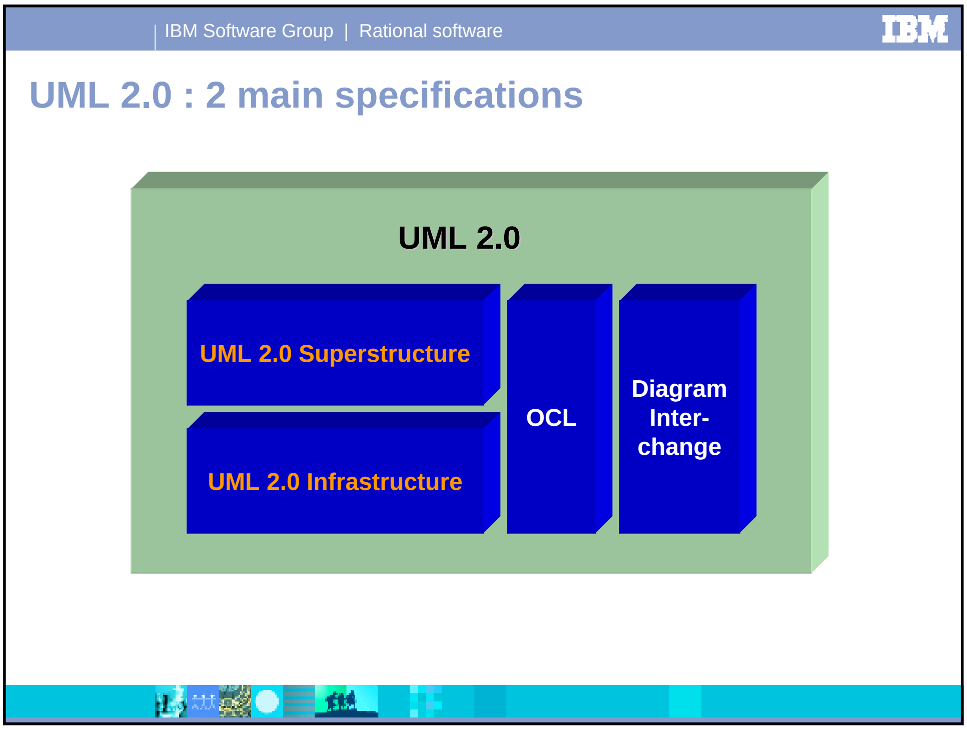
- Improve language
  - Better component based development support
  - Improved data flow modeling in activity graphs
  - Composition, sequences, parallel execution of interactions
  - Hierarchical interactions modeling
  - ...
- Minimize impact on user of the current UML 1.x
  - Evolution rather than revolution
- Metamodel should align with MOF meta-metamodel
- Improve extension mechanisms like profiles to increase customizability
- Remove unused and ill-defined modeling concepts



IBM Software Group | Rational software

U2-partners Consortium

- A consortium of UML vendors and users
  - Submitters
    - Alcatel, ComputerAssociates, Enea Business Software, Ericsson, HP, IBM, I-Logix, IONA, Kabira, Motorola, Oracle, Rational, SOFTEAM, Telelogic, Unisys, WebGain
  - Supporters
    - Advanced Concepts Center LLC, Ceira Technologies, Commissariat à L'Energie Atomique, Compuware, DaimlerChrysler, Embarcadero, France Telecom, Fraunhofer FOKUS, Fujitsu, Gentleware, Intellicorp, Jaczone, Kennedy Carter, KLOCwork, Lockheed Martin, Mercury Computer, MSC.Software, Northeastern University, Popkin Software, Proforma, Sims Associates, Syntropy, Thales Group, University of Kaiserslautern, VERIMAG, 88solutions
- www.u2-partners.org



IBM Software Group | Rational software

UML 2.0 : 2 main specifications

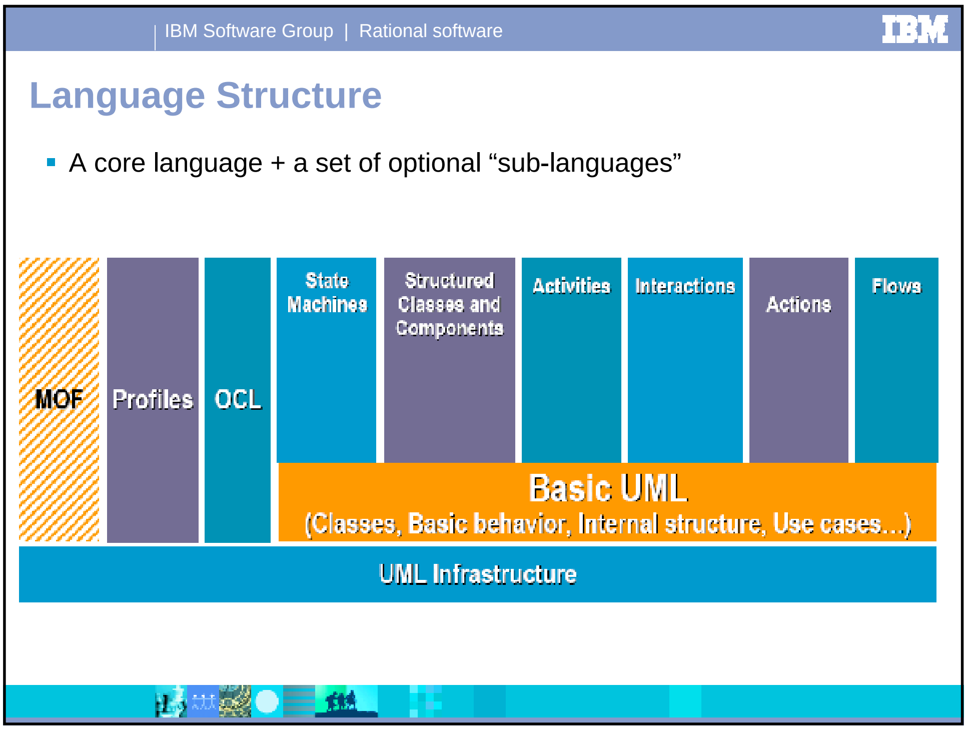
UML 2.0

UML 2.0 Superstructure

UML 2.0 Infrastructure

OCL

Diagram Inter-change



IBM Software Group | Rational software

Language Structure

- A core language + a set of optional “sub-languages”

MOF

Profiles

OCL

State Machines

Structured Classes and Components

Activities

Interactions

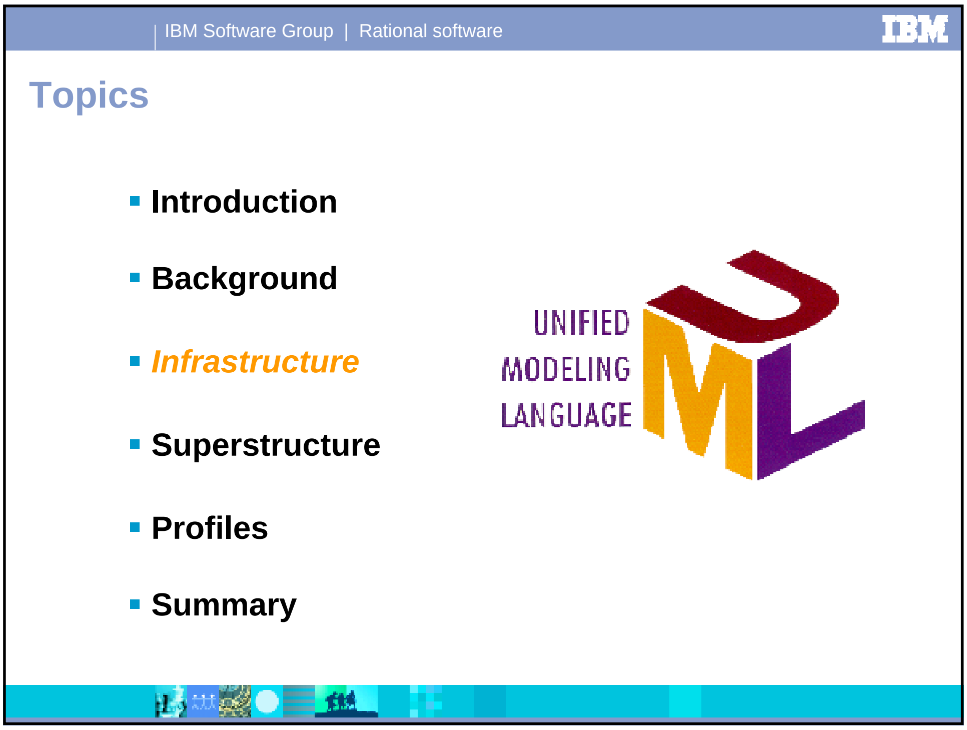
Actions

Flows

Basic UML

(Classes, Basic behavior, Internal structure, Use cases…)

UML Infrastructure

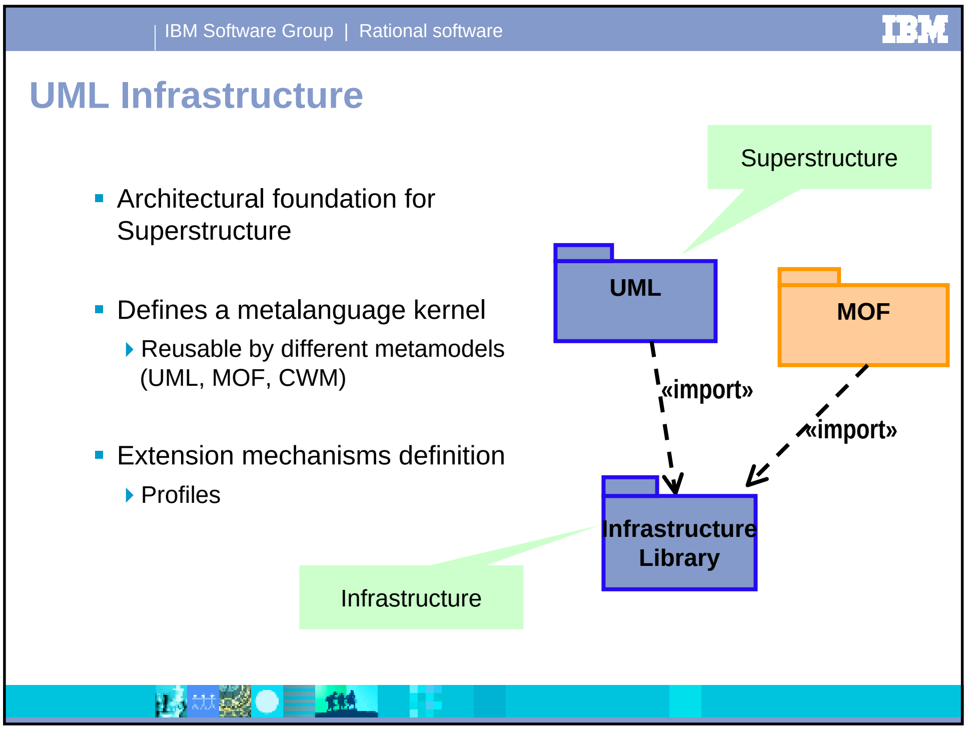


IBM Software Group | Rational software

Topics

- Introduction
- Background
- *Infrastructure*
- Superstructure
- Profiles
- Summary

UNIFIED
MODELING
LANGUAGE



IBM Software Group | Rational software

UML Infrastructure

- Architectural foundation for Superstructure
- Defines a metalanguage kernel
  - Reusable by different metamodels (UML, MOF, CWM)
- Extension mechanisms definition
  - Profiles

Superstructure

UML

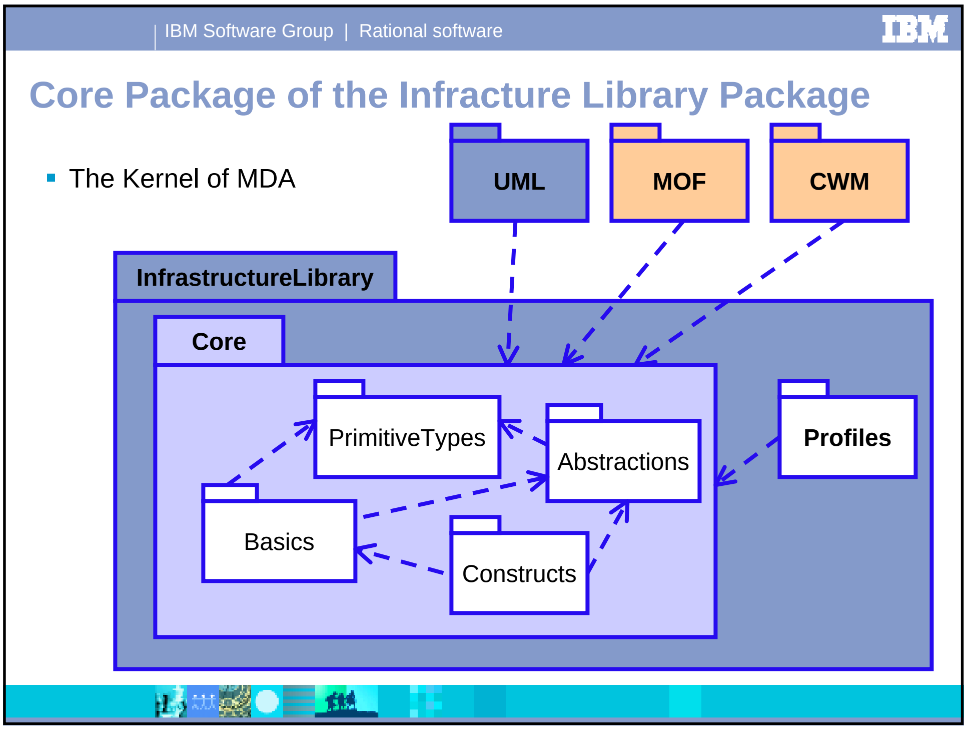
MOF

«import»

«import»

Infrastructure Library

Infrastructure



IBM Software Group | Rational software

Core Package of the Infracture Library Package

- The Kernel of MDA

UML

MOF

CWM

InfrastructureLibrary

Core

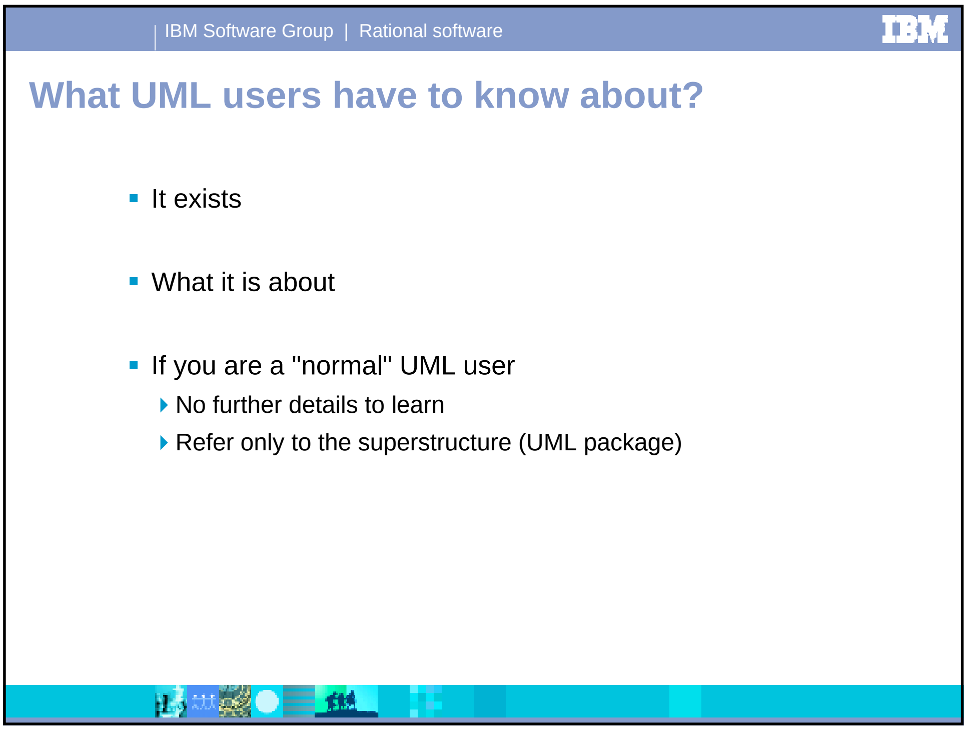
PrimitiveTypes

Abstractions

Profiles

Basics

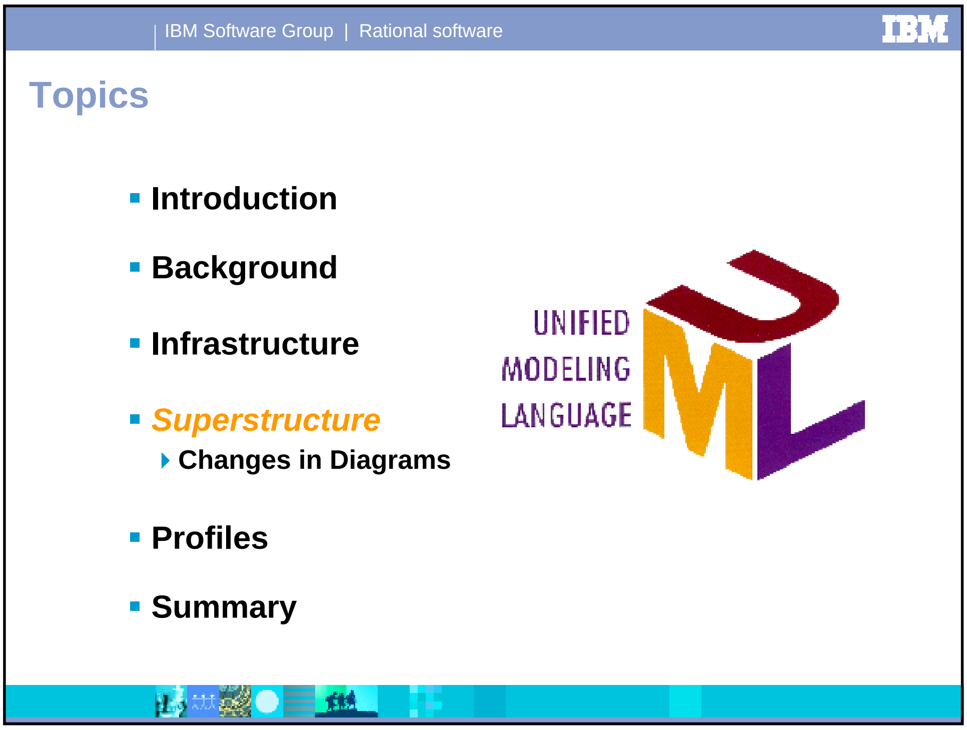
Constructs



IBM Software Group | Rational software

What UML users have to know about?

- It exists
- What it is about
- If you are a "normal" UML user
  - No further details to learn
  - Refer only to the superstructure (UML package)

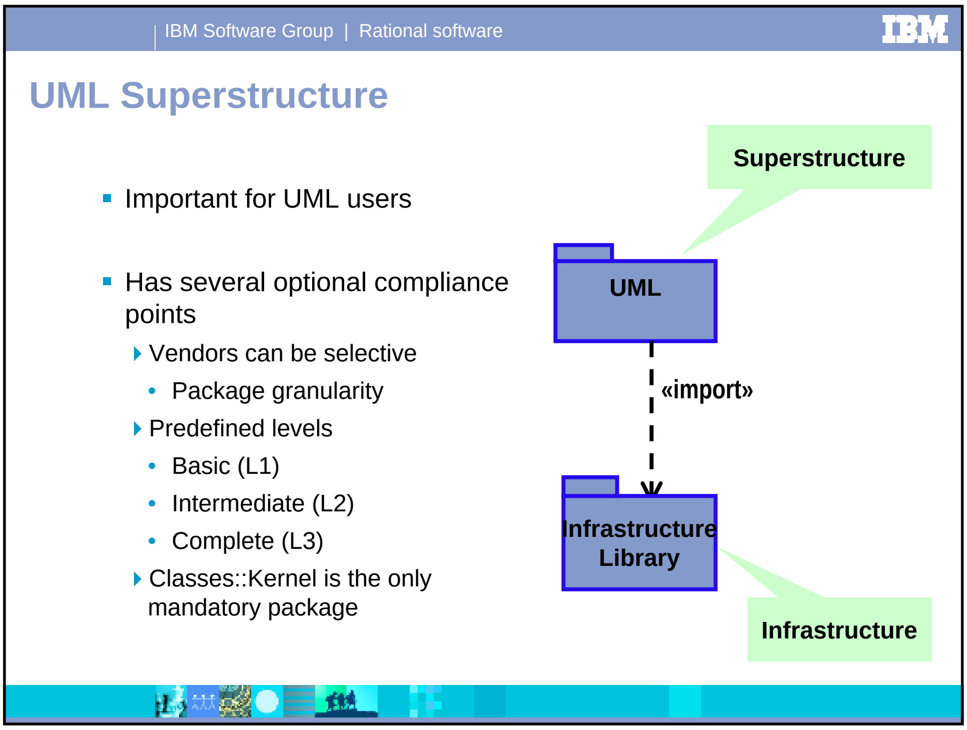


IBM Software Group | Rational software

Topics

- **Introduction**
- **Background**
- **Infrastructure**
- ***Superstructure***
  - **Changes in Diagrams**
- **Profiles**
- **Summary**

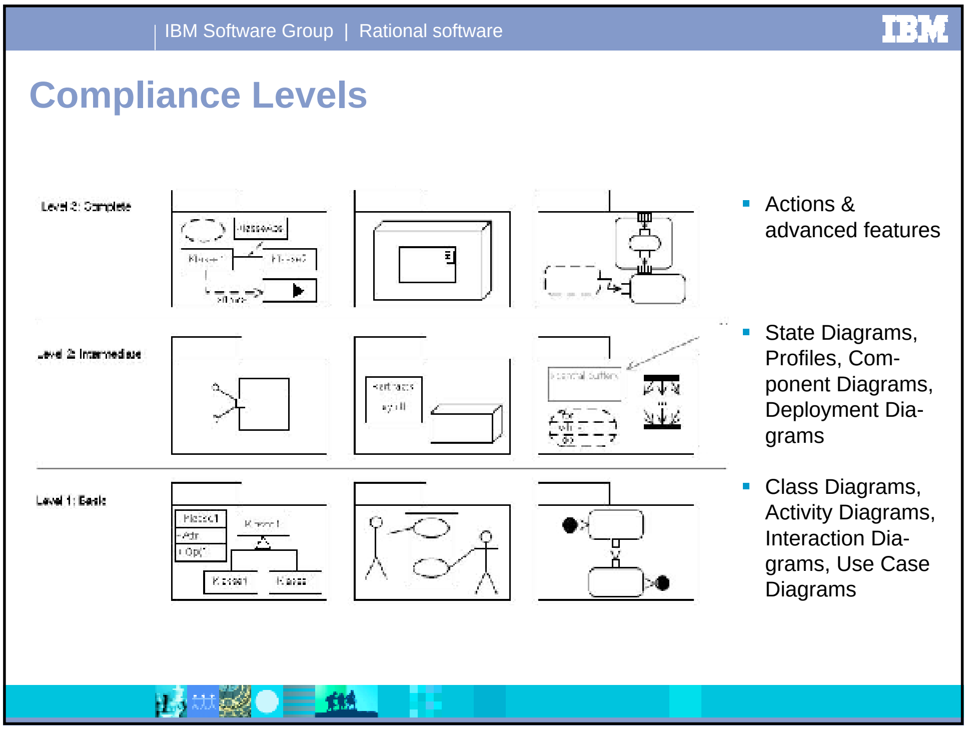
UNIFIED
MODELING
LANGUAGE
UML



IBM Software Group | Rational software

UML Superstructure

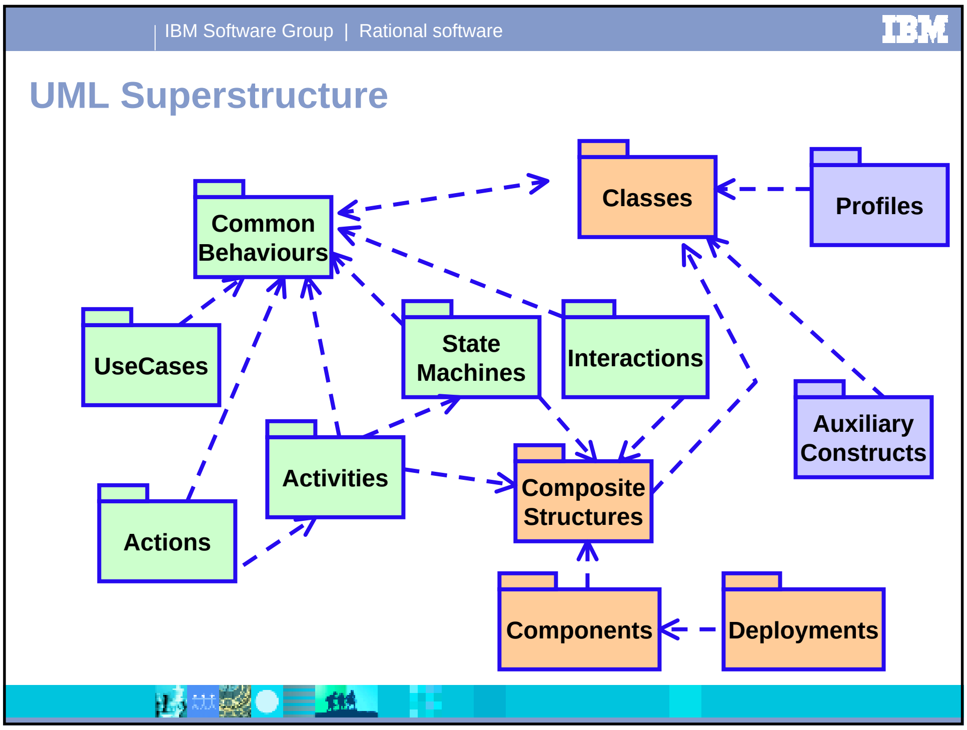
- Important for UML users
- Has several optional compliance points
  - Vendors can be selective
    - Package granularity
  - Predefined levels
    - Basic (L1)
    - Intermediate (L2)
    - Complete (L3)
  - Classes::Kernel is the only mandatory package



IBM Software Group | Rational software

Compliance Levels

- Actions & advanced features
- State Diagrams, Profiles, Component Diagrams, Deployment Diagrams
- Class Diagrams, Activity Diagrams, Interaction Diagrams, Use Case Diagrams



IBM Software Group | Rational software

UML Superstructure

Common Behaviours

Classes

Profiles

UseCases

State Machines

Interactions

Auxiliary Constructs

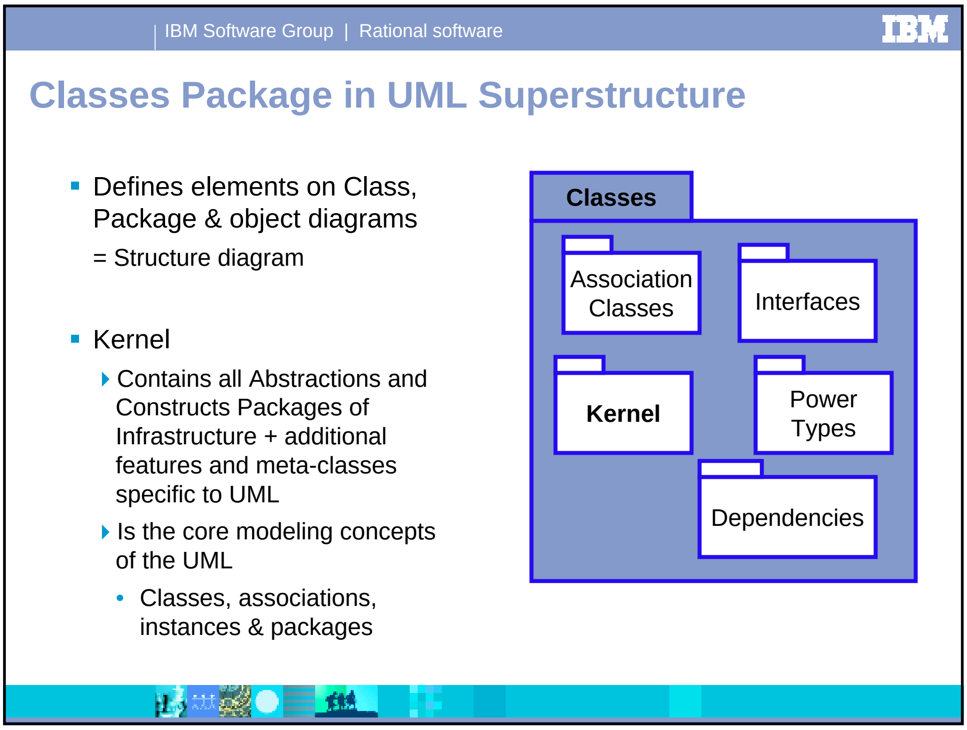
Activities

Composite Structures

Actions

Components

Deployments



IBM Software Group | Rational software

Classes Package in UML Superstructure

- Defines elements on Class, Package & object diagrams

  = Structure diagram

- Kernel
  - Contains all Abstractions and Constructs Packages of Infrastructure + additional features and meta-classes specific to UML
  - Is the core modeling concepts of the UML
    - Classes, associations, instances & packages

Classes

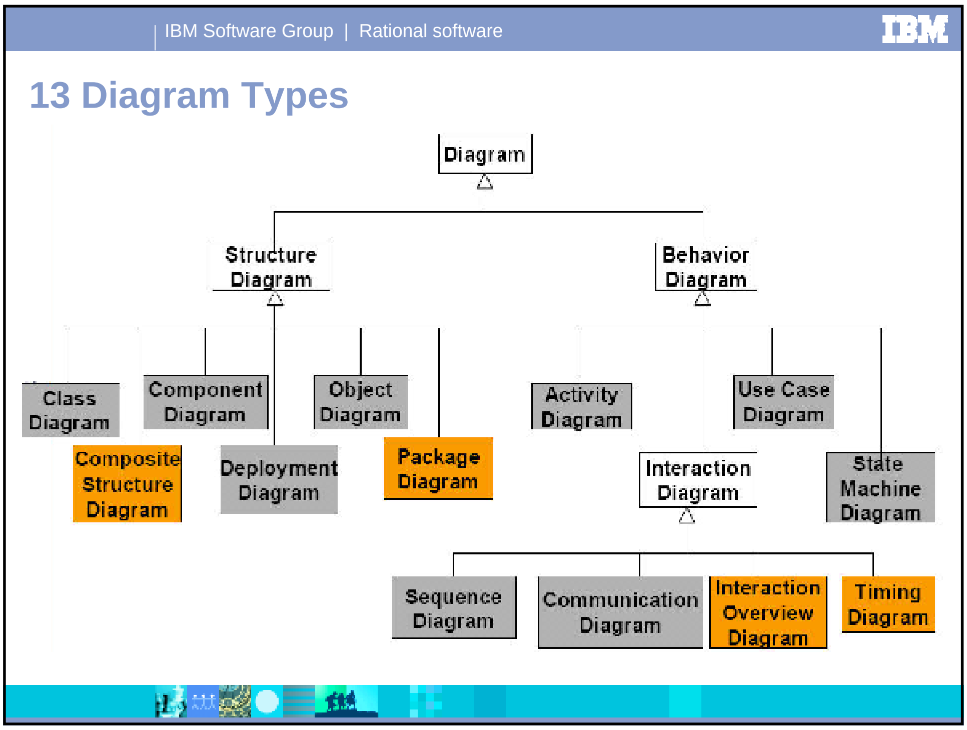
Association Classes

Interfaces

Kernel

Power Types

Dependencies



IBM Software Group | Rational software IBM

13 Diagram Types

Diagram

Structure Diagram

Behavior Diagram

Class Diagram

Component Diagram

Object Diagram

Composite Structure Diagram

Deployment Diagram

Package Diagram

Activity Diagram

Use Case Diagram

Interaction Diagram

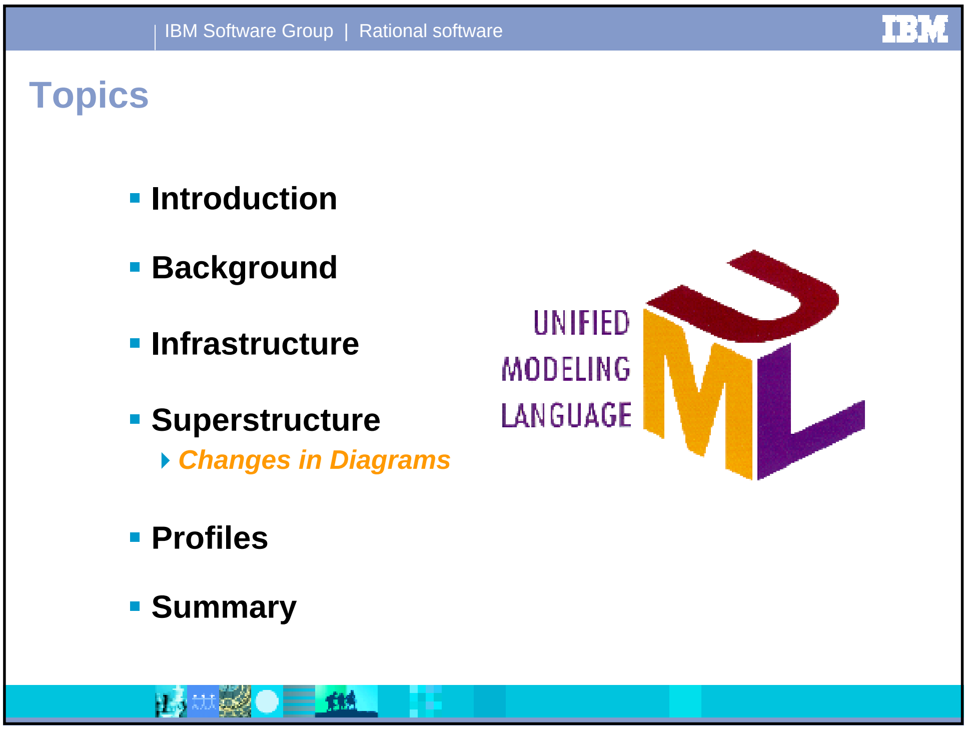
State Machine Diagram

Sequence Diagram

Communication Diagram

Interaction Overview Diagram

Timing Diagram

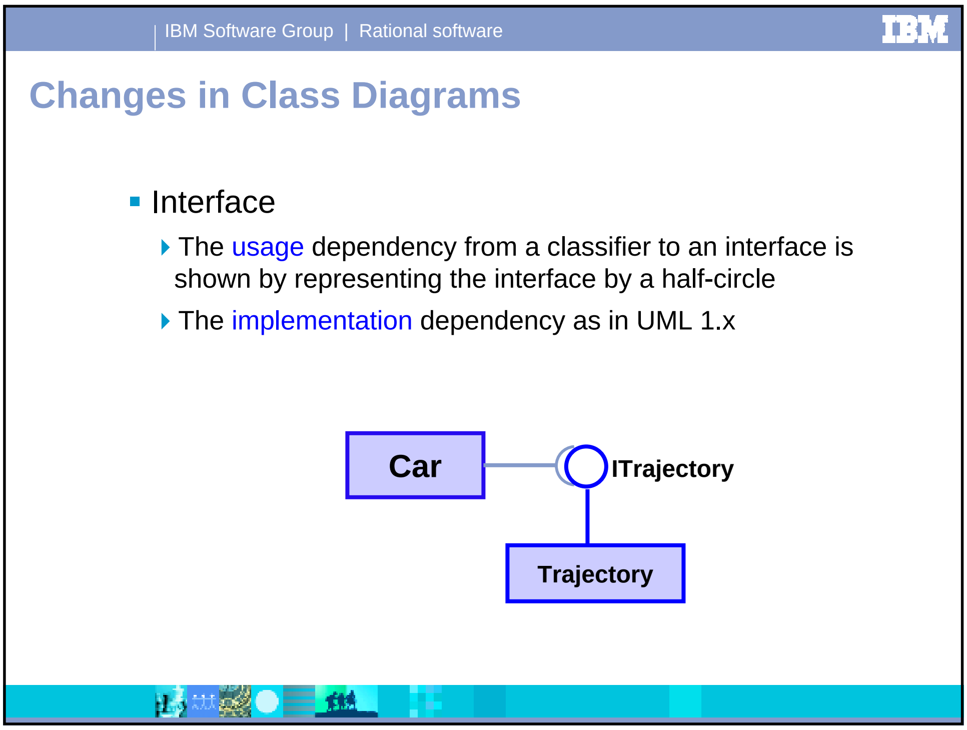


IBM Software Group | Rational software IBM

Topics

- Introduction
- Background
- Infrastructure
- Superstructure
  - *Changes in Diagrams*
- Profiles
- Summary

UNIFIED MODELING LANGUAGE



IBM Software Group | Rational software

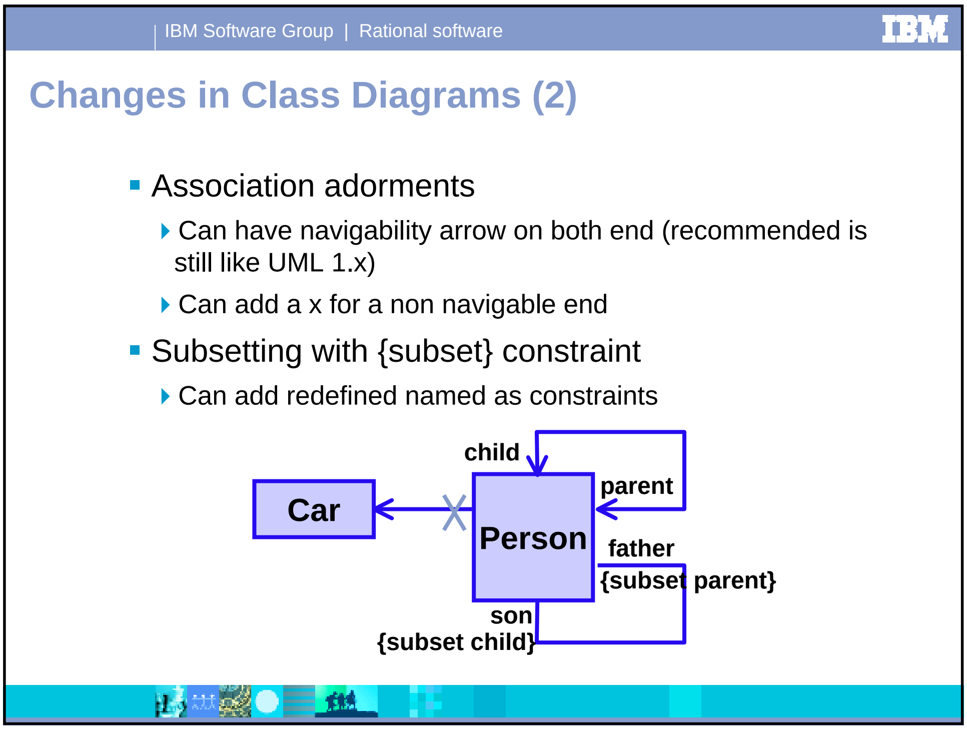
Changes in Class Diagrams

- Interface
  - The usage dependency from a classifier to an interface is shown by representing the interface by a half-circle
  - The implementation dependency as in UML 1.x

Car

ITrajectory

Trajectory



IBM Software Group | Rational software

Changes in Class Diagrams (2)

- Association adorments
  - Can have navigability arrow on both end (recommended is still like UML 1.x)
  - Can add a x for a non navigable end
- Subsetting with {subset} constraint
  - Can add redefined named as constraints

child

parent

Car

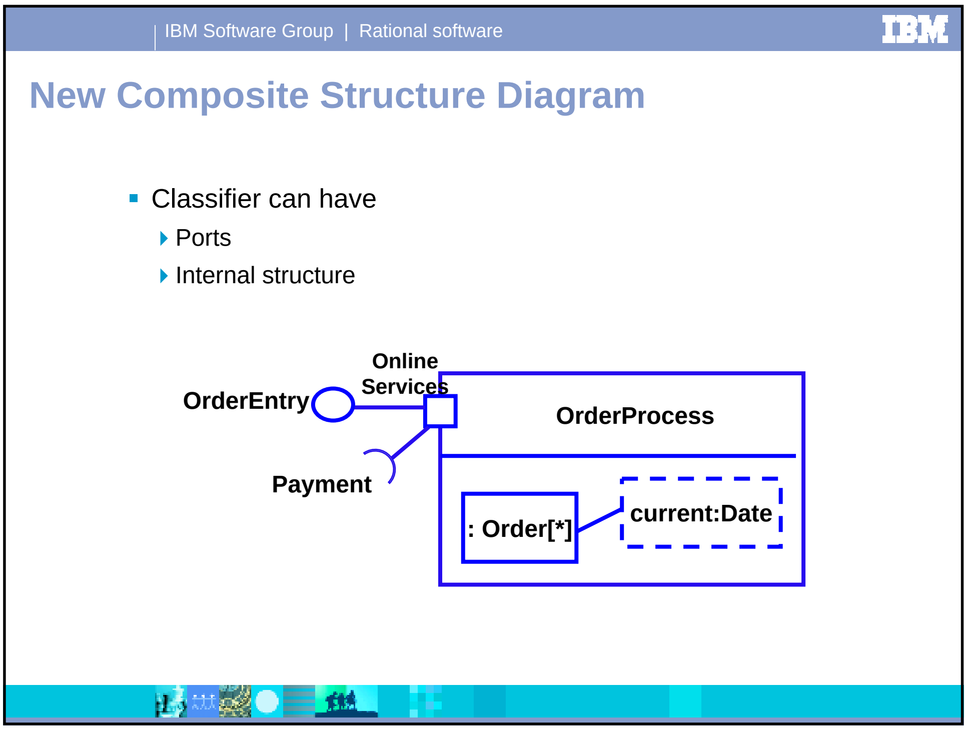
Person

father

{subset parent}

son

{subset child}



IBM Software Group | Rational software

New Composite Structure Diagram

- Classifier can have
  - Ports
  - Internal structure

Online Services

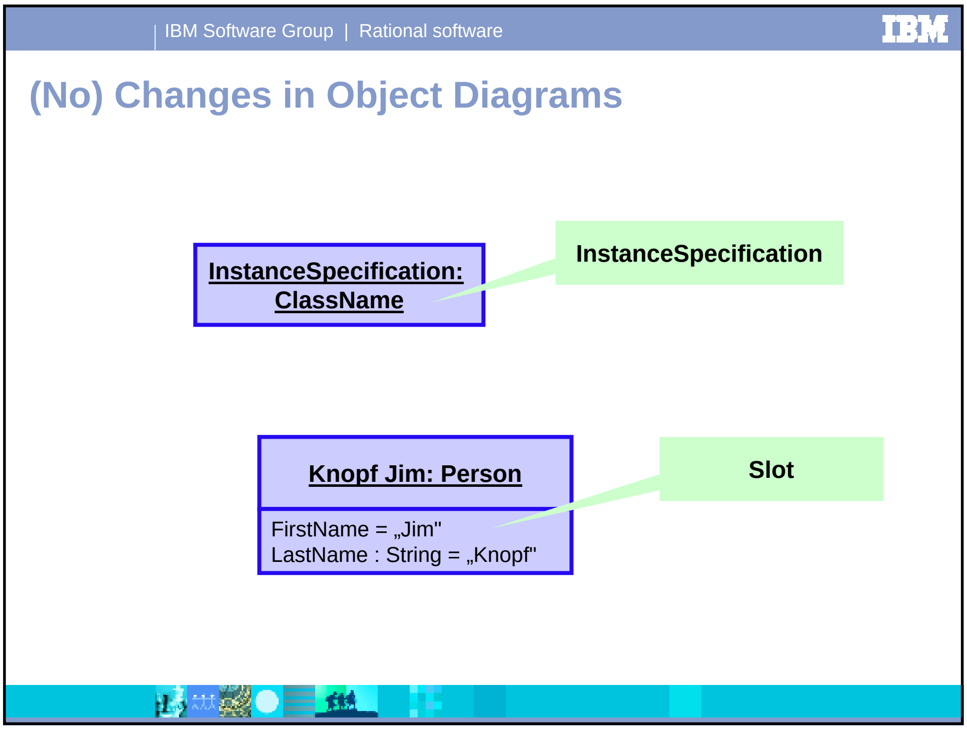
OrderEntry

Payment

OrderProcess

: Order[*]

current:Date



IBM Software Group | Rational software

(No) Changes in Object Diagrams

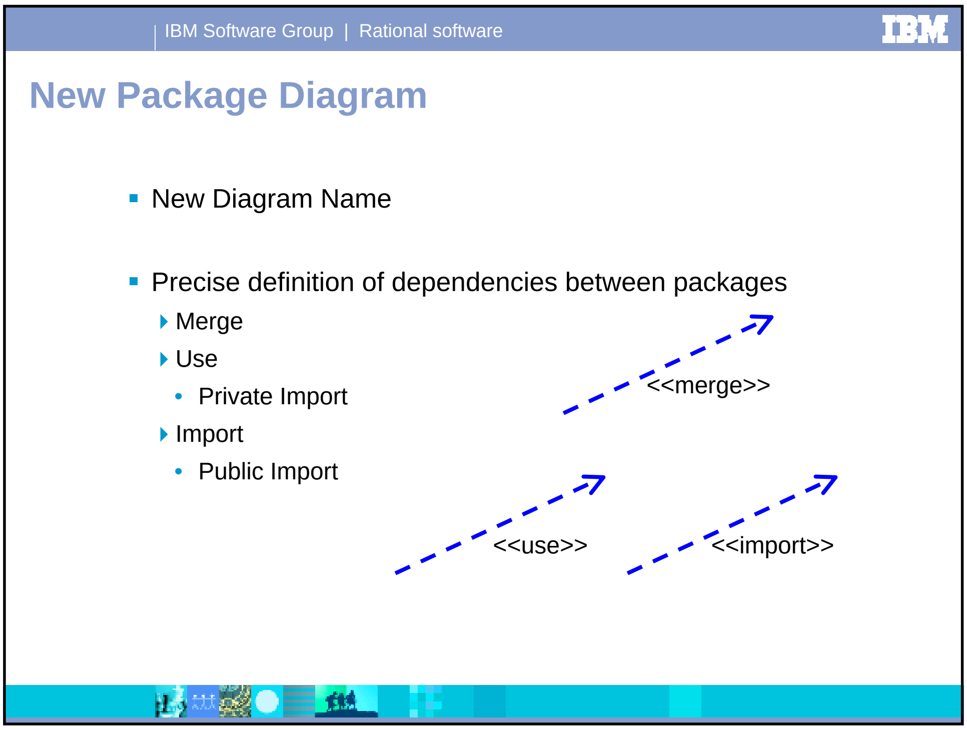
InstanceSpecification: ClassName

InstanceSpecification

Knopf Jim: Person

FirstName = „Jim"
LastName : String = „Knopf"

Slot



IBM Software Group | Rational software

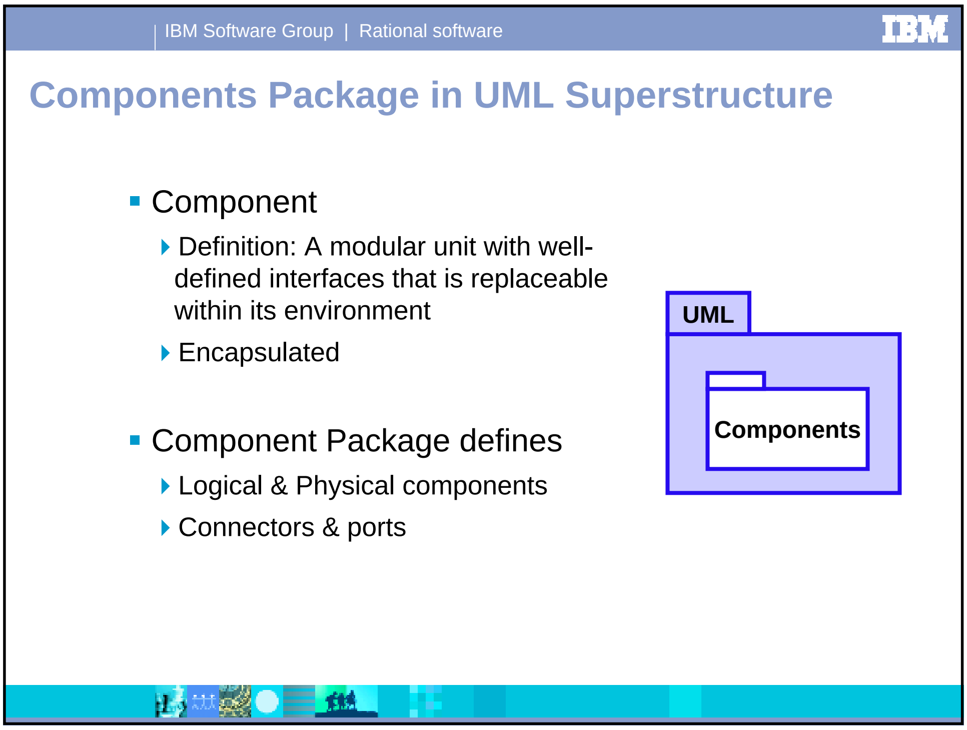
New Package Diagram

- New Diagram Name
- Precise definition of dependencies between packages
  - Merge
  - Use
    - Private Import
  - Import
    - Public Import

<<merge>>

<<use>>

<<import>>



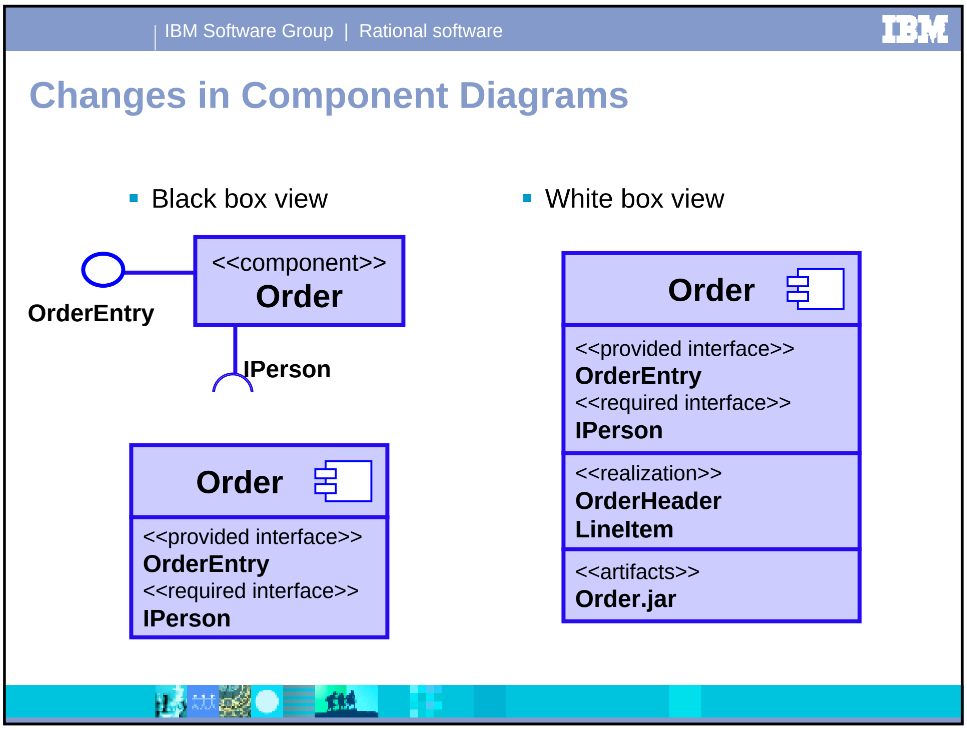
IBM Software Group | Rational software

Components Package in UML Superstructure

- Component
  - Definition: A modular unit with well-defined interfaces that is replaceable within its environment
  - Encapsulated
- Component Package defines
  - Logical & Physical components
  - Connectors & ports

UML

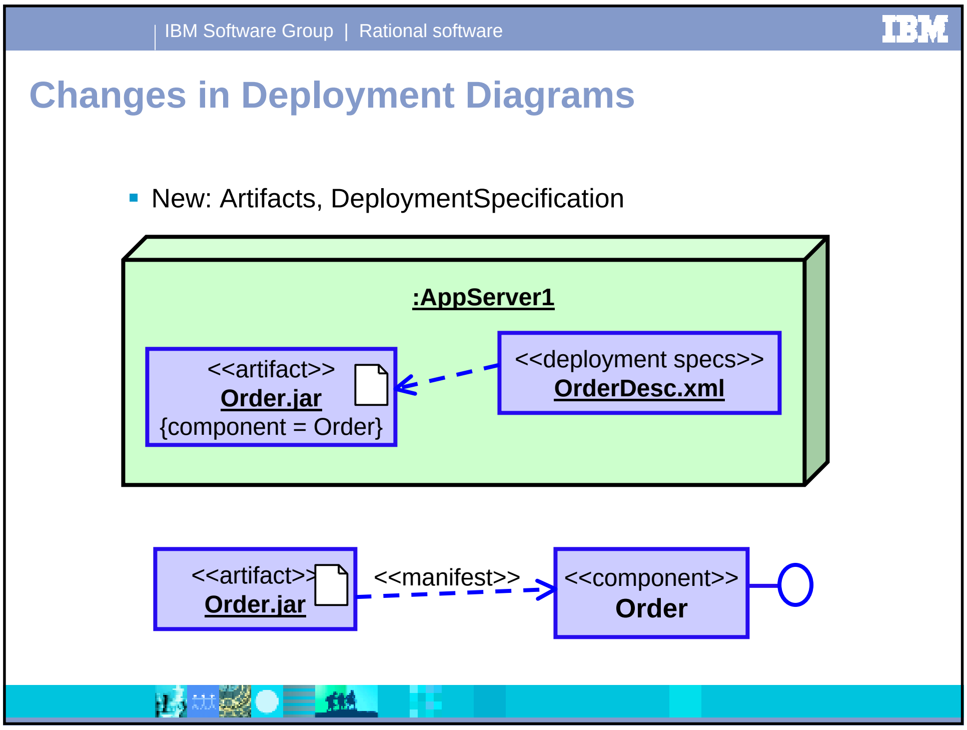
Components



IBM Software Group | Rational software

Changes in Component Diagrams

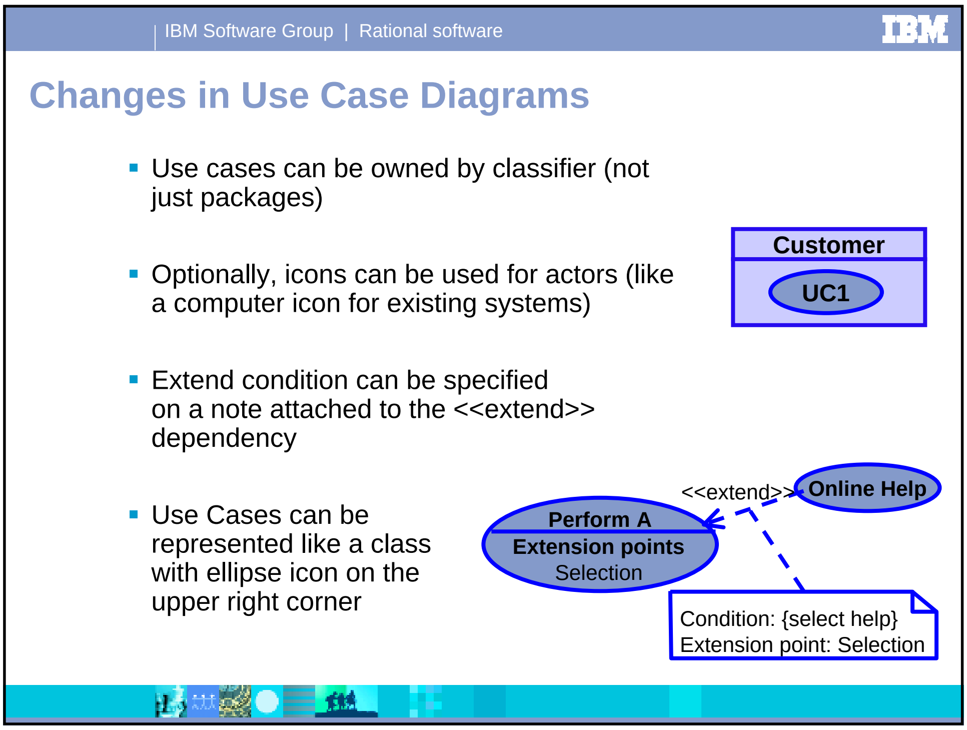
- Black box view
- White box view



IBM Software Group | Rational software

Changes in Deployment Diagrams

- New: Artifacts, DeploymentSpecification



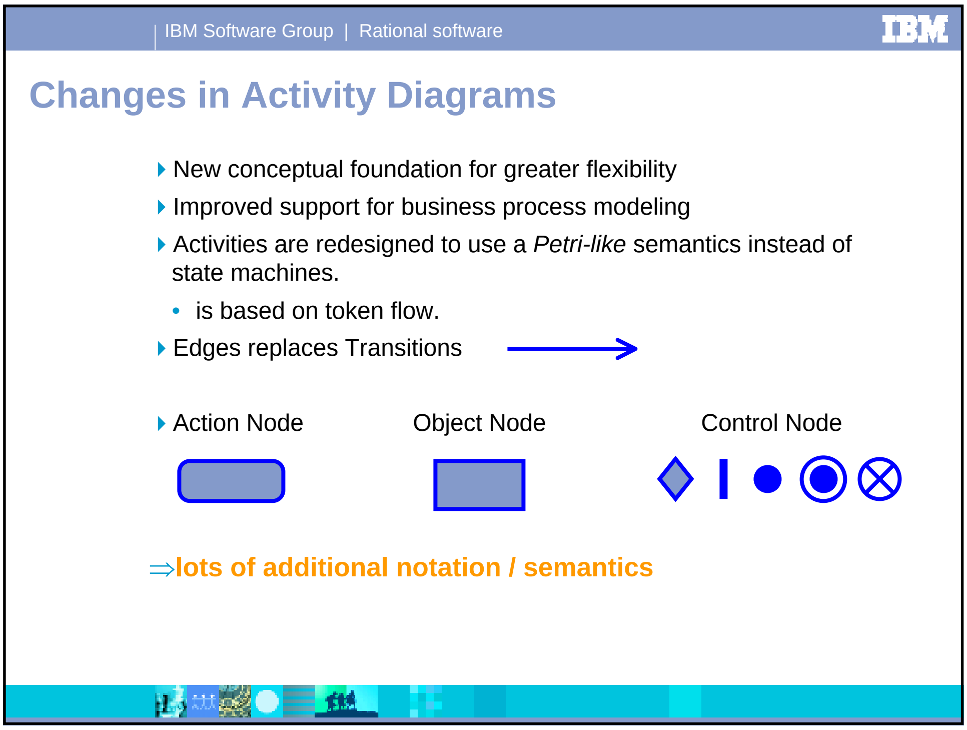
IBM Software Group | Rational software

Changes in Use Case Diagrams

- Use cases can be owned by classifier (not just packages)
- Optionally, icons can be used for actors (like a computer icon for existing systems)
- Extend condition can be specified on a note attached to the <<extend>> dependency
- Use Cases can be represented like a class with ellipse icon on the upper right corner

Customer
UC1
Perform A
Extension points
Selection
<<extend>>
Online Help

Condition: {select help}
Extension point: Selection



IBM Software Group | Rational software

Changes in Activity Diagrams

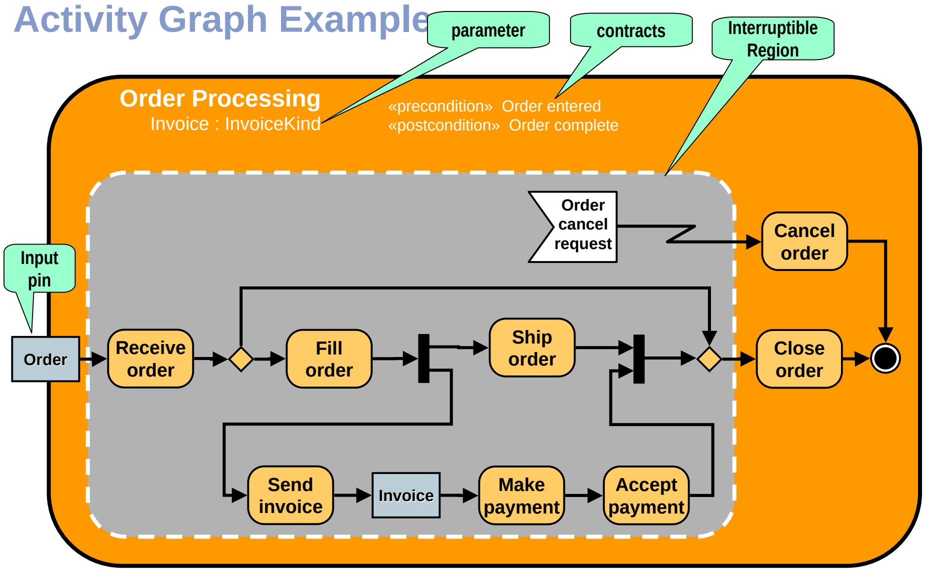
- New conceptual foundation for greater flexibility
- Improved support for business process modeling
- Activities are redesigned to use a *Petri-like* semantics instead of state machines.
  - is based on token flow.
- Edges replaces Transitions →
- Action Node

Object Node
Control Node
◆ | ● ◉ ⊗

⇒lots of additional notation / semantics

IBM Software Group | Rational software

Activity Graph Example



parameter

contracts

Interruptible Region

Order Processing

Invoice : InvoiceKind

«precondition» Order entered

«postcondition» Order complete

Input pin

Order

Receive order

Fill order

Ship order

Order cancel request

Cancel order

Close order

Send invoice

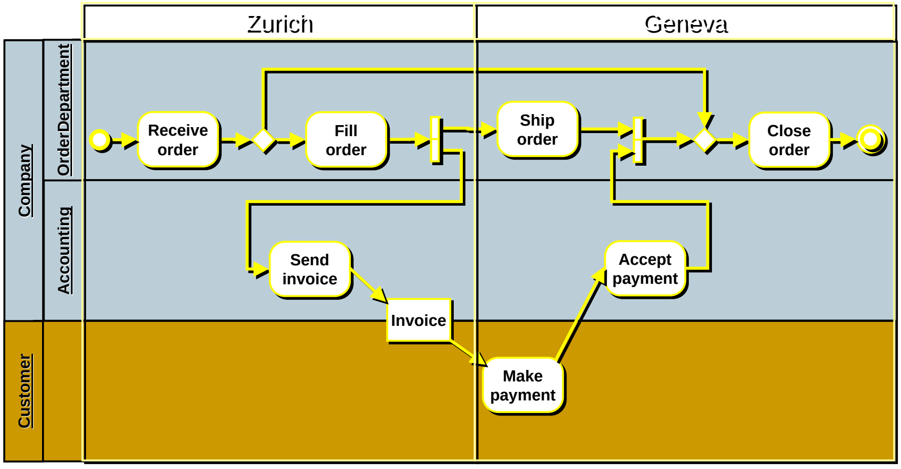
Invoice

Make payment

Accept payment

IBM Software Group | Rational software

Partitioning capabilities



Zurich

Geneva

Company

OrderDepartment

Accounting

Customer

Receive order

Fill order

Ship order

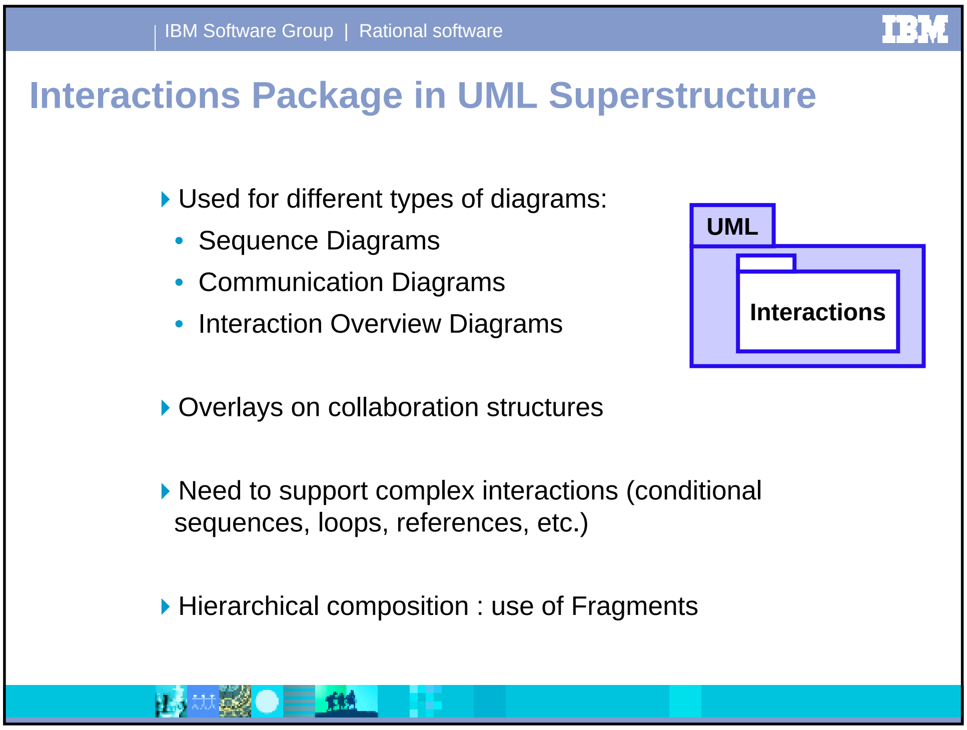
Close order

Send invoice

Invoice

Accept payment

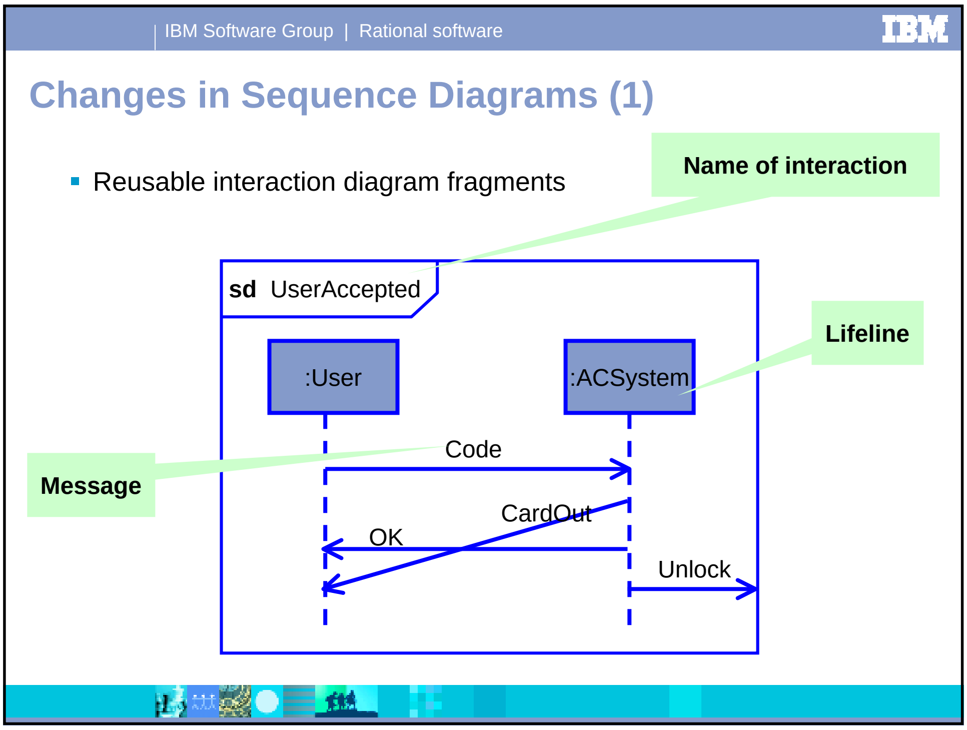
Make payment



IBM Software Group | Rational software

Interactions Package in UML Superstructure

- Used for different types of diagrams:
  - Sequence Diagrams
  - Communication Diagrams
  - Interaction Overview Diagrams
- Overlays on collaboration structures
- Need to support complex interactions (conditional sequences, loops, references, etc.)
- Hierarchical composition : use of Fragments



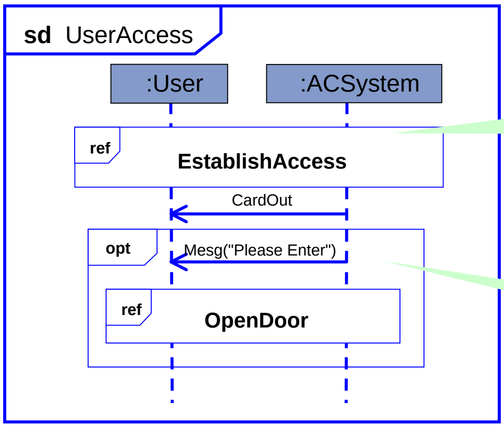
IBM Software Group | Rational software

Changes in Sequence Diagrams (1)

- Reusable interaction diagram fragments

Changes in Sequence Diagrams (2)

- Frame "Composition"

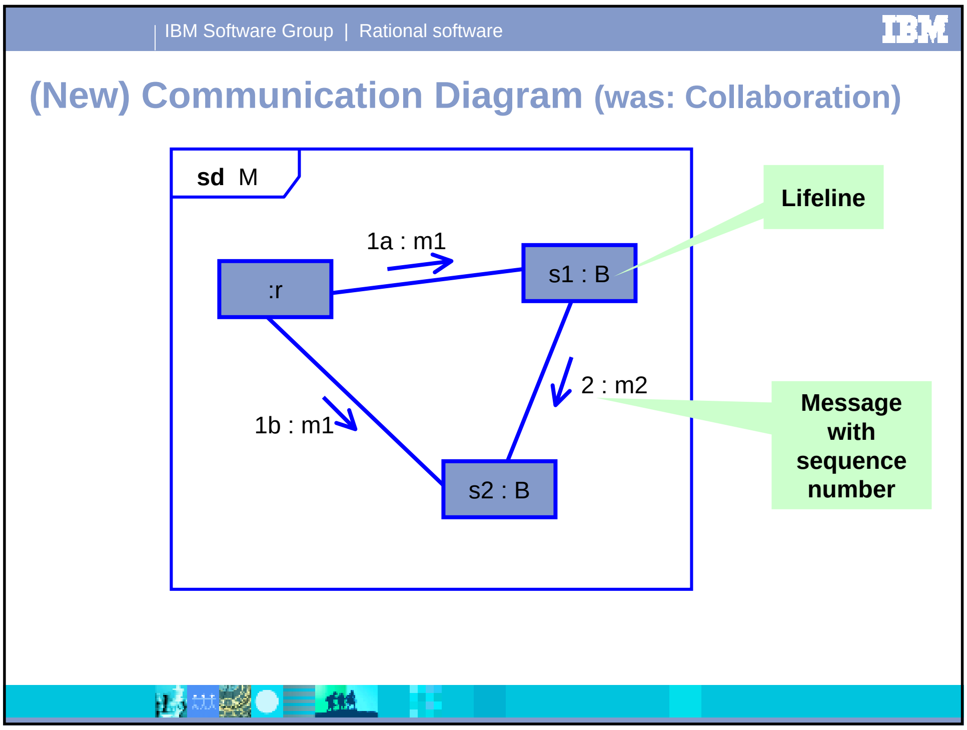


Reference to another interaction

Optional Interaction

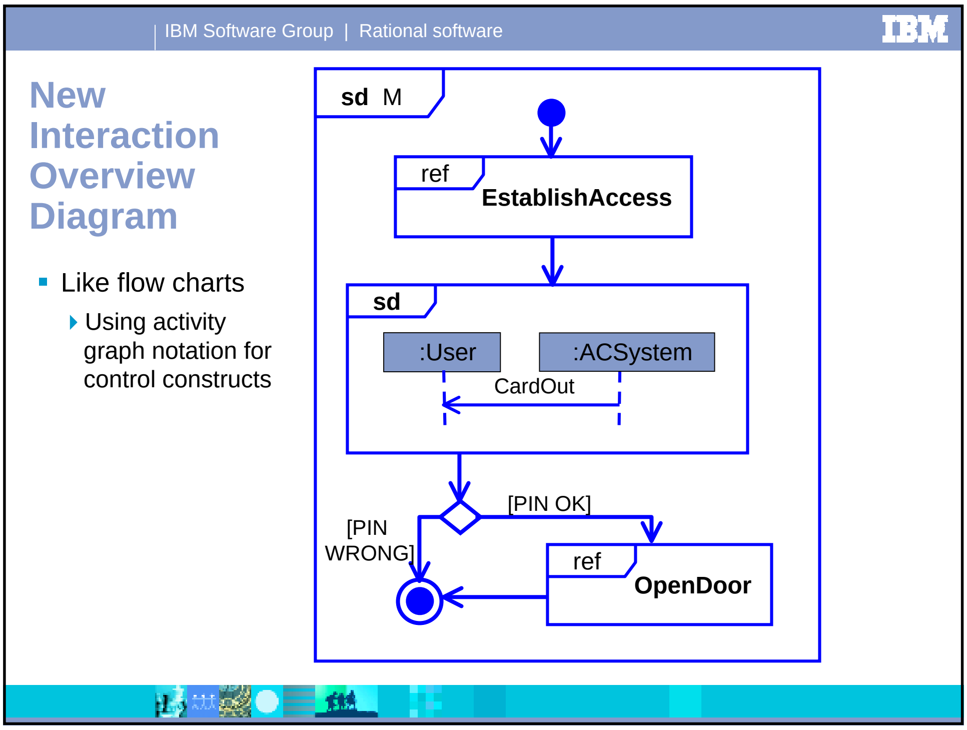
Combined Fragment Types

- Alternatives (alt)
  - choice of behaviors – at most one will execute
  - depends on the value of the guard (“else” guard supported)
- Option (opt)
  - Special case of alternative
- Break (break)
  - Represents an alternative that is executed instead of the remainder of the fragment (like a break in a loop)
- Parallel (par)
  - Concurrent (interleaved) sub-scenarios
- Negative (neg)
  - Identifies sequences that must not occur
- Critical Region (region)
  - Traces cannot be interleaved with events on any of the participating lifelines
- Assertion (assert)
  - Only valid continuation
- Loop (loop)
  - Optional guard: [<min>, <max>, <Boolean-expression>]
  - No guard means no specified limit



IBM Software Group | Rational software

(New) Communication Diagram (was: Collaboration)



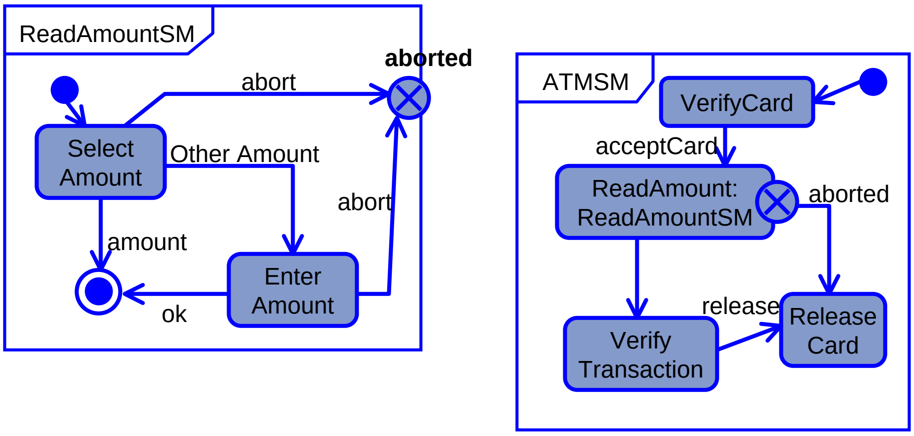
IBM Software Group | Rational software

New Interaction Overview Diagram

- Like flow charts
  - Using activity graph notation for control constructs

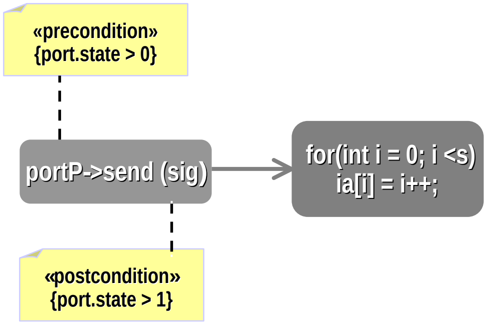
Changes in State Machine Diagrams

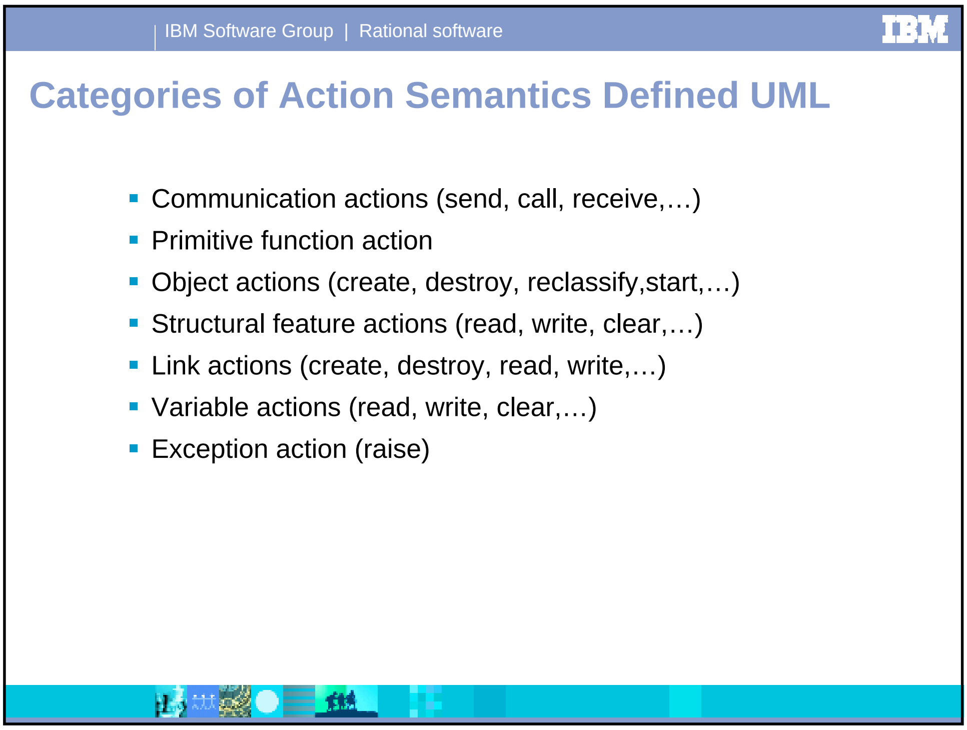
- Entry / Exit points



Action *Semantics*

- Actions: finest-grained form of behavior in UML
- Can be realized by most standard programming languages

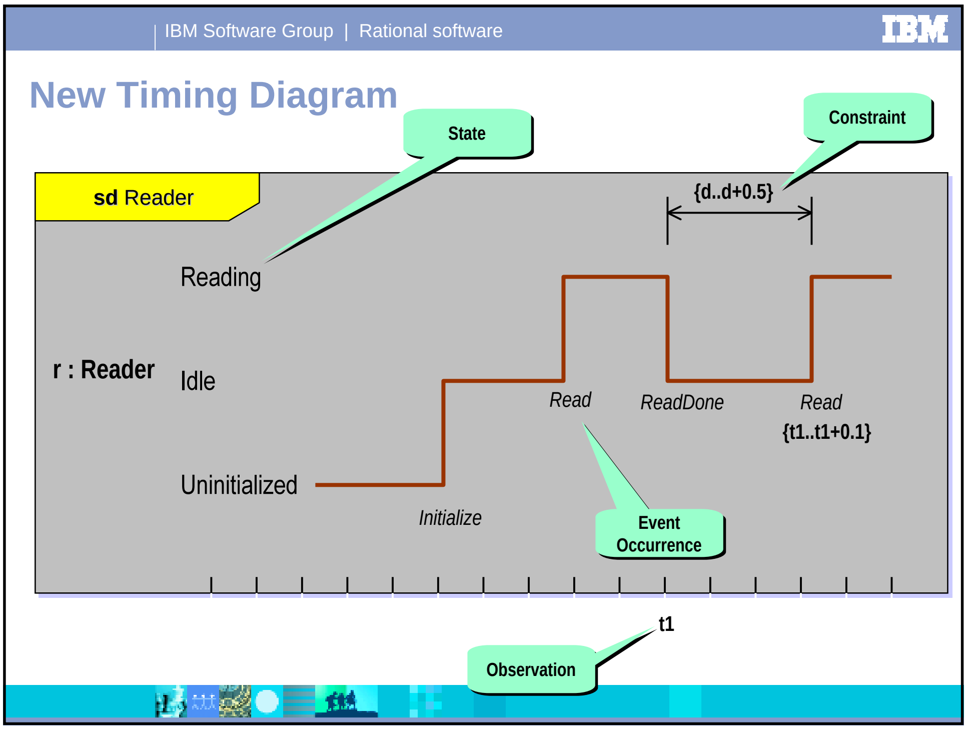

«precondition» {port.state > 0}
portP->send (sig)
for(int i = 0; i <s) ia[i] = i++;
«postcondition» {port.state > 1}



IBM Software Group | Rational software

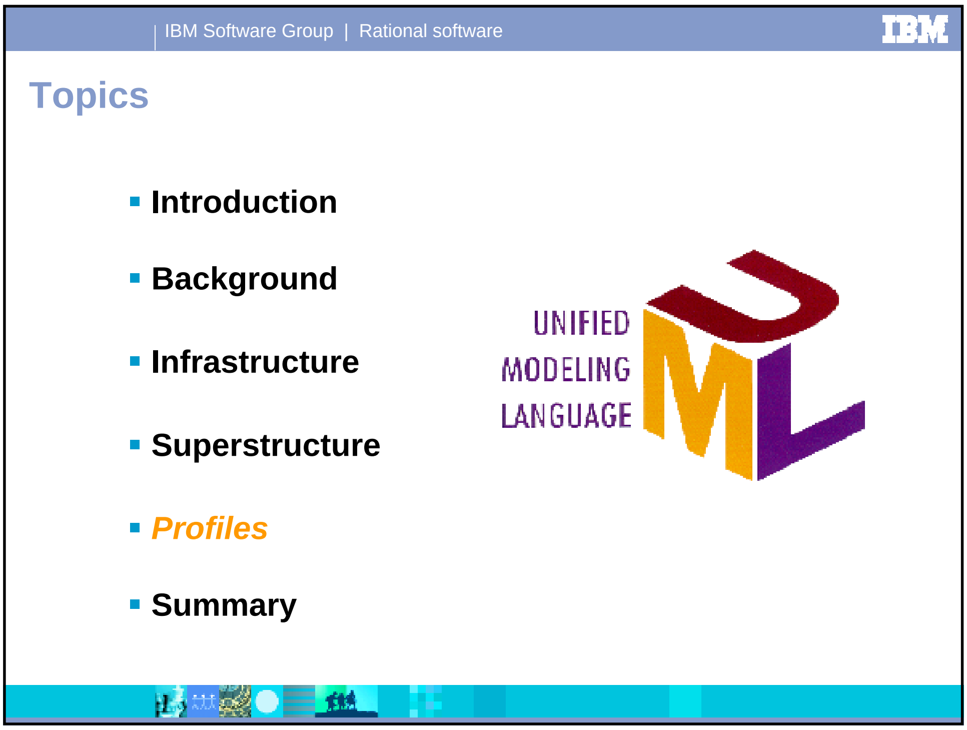
Categories of Action Semantics Defined UML

- Communication actions (send, call, receive,…)
- Primitive function action
- Object actions (create, destroy, reclassify,start,…)
- Structural feature actions (read, write, clear,…)
- Link actions (create, destroy, read, write,…)
- Variable actions (read, write, clear,…)
- Exception action (raise)



IBM Software Group | Rational software

New Timing Diagram



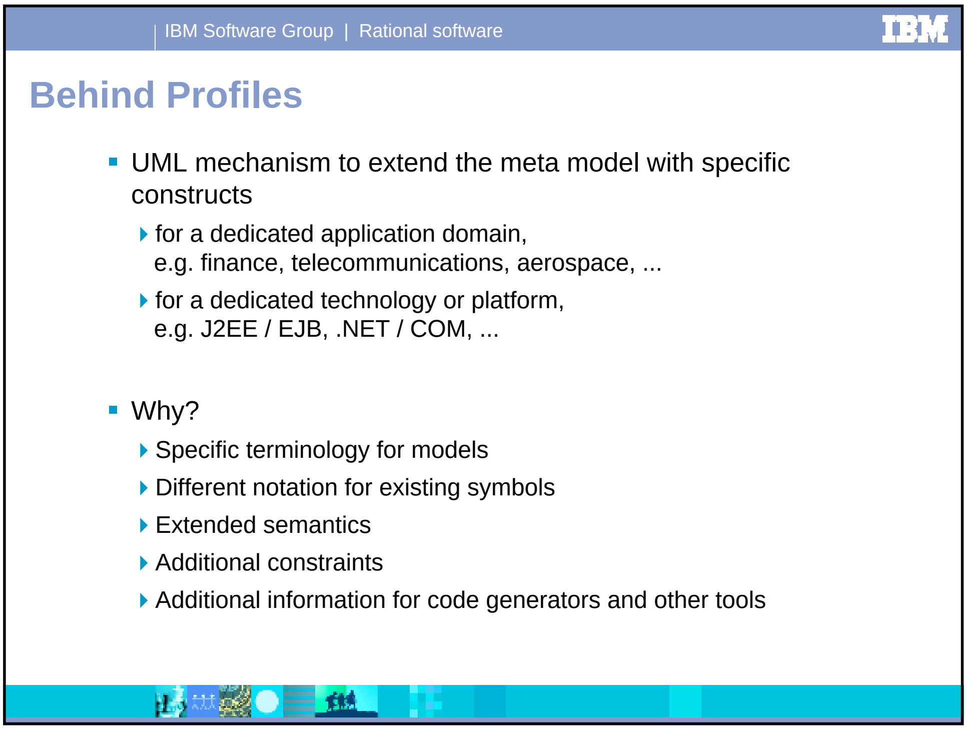
IBM Software Group | Rational software

IBM

Topics

- Introduction
- Background
- Infrastructure
- Superstructure
- *Profiles*
- Summary

UNIFIED
MODELING
LANGUAGE
UML

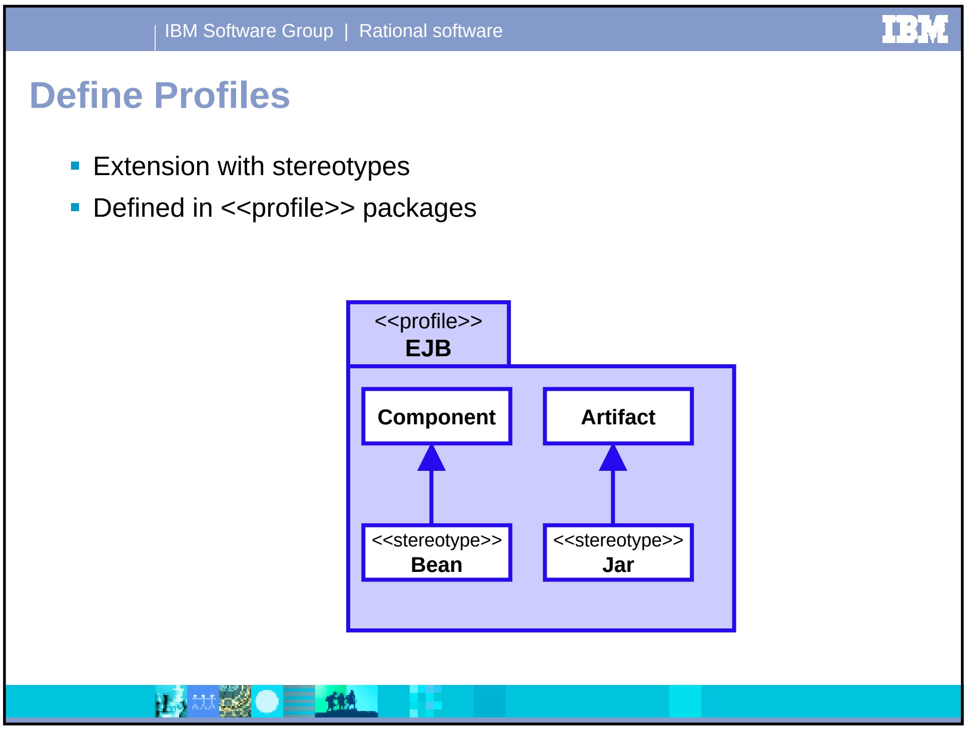


IBM Software Group | Rational software

IBM

Behind Profiles

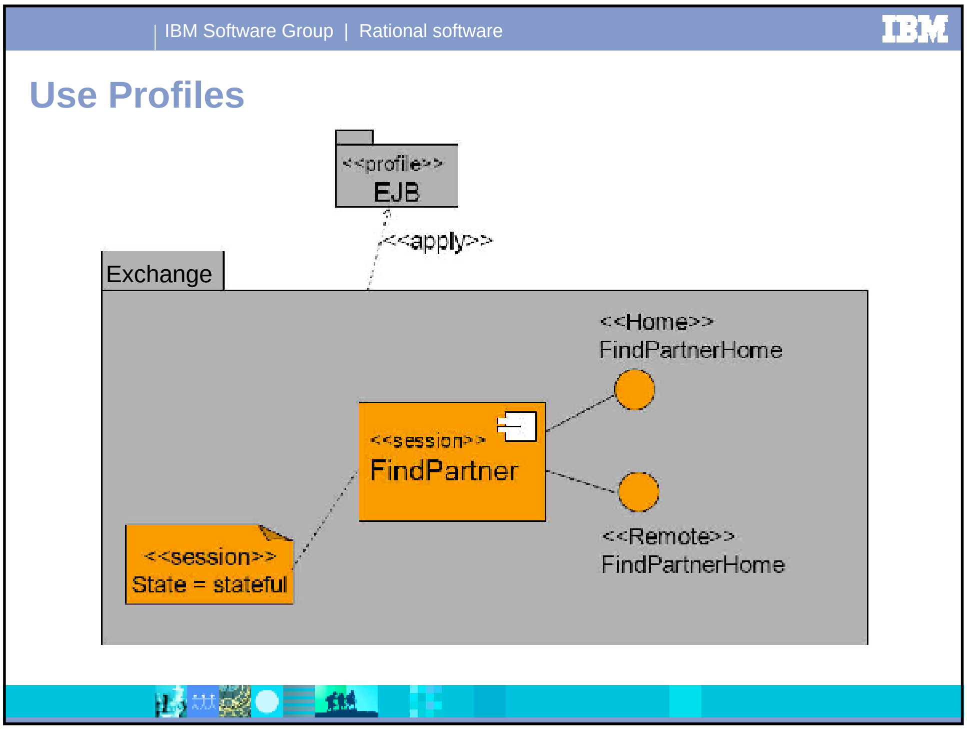
- UML mechanism to extend the meta model with specific constructs
  - for a dedicated application domain,
e.g. finance, telecommunications, aerospace, ...
  - for a dedicated technology or platform,
e.g. J2EE / EJB, .NET / COM, ...
- Why?
  - Specific terminology for models
  - Different notation for existing symbols
  - Extended semantics
  - Additional constraints
  - Additional information for code generators and other tools



IBM Software Group | Rational software

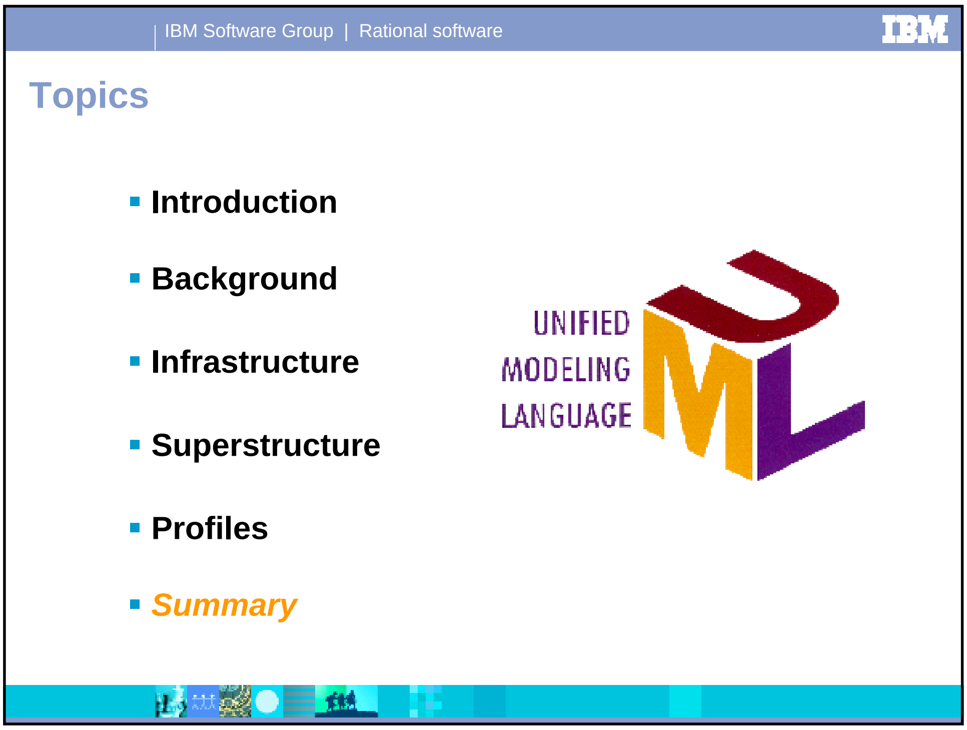
Define Profiles

- Extension with stereotypes
- Defined in <<profile>> packages



IBM Software Group | Rational software

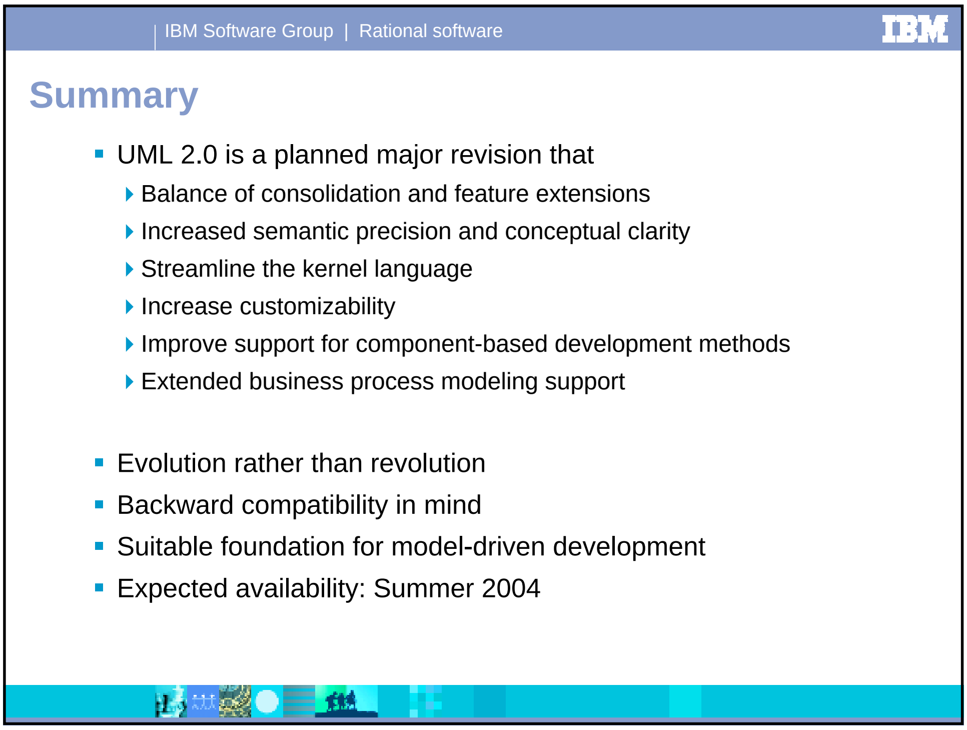
Use Profiles



IBM Software Group | Rational software

Topics

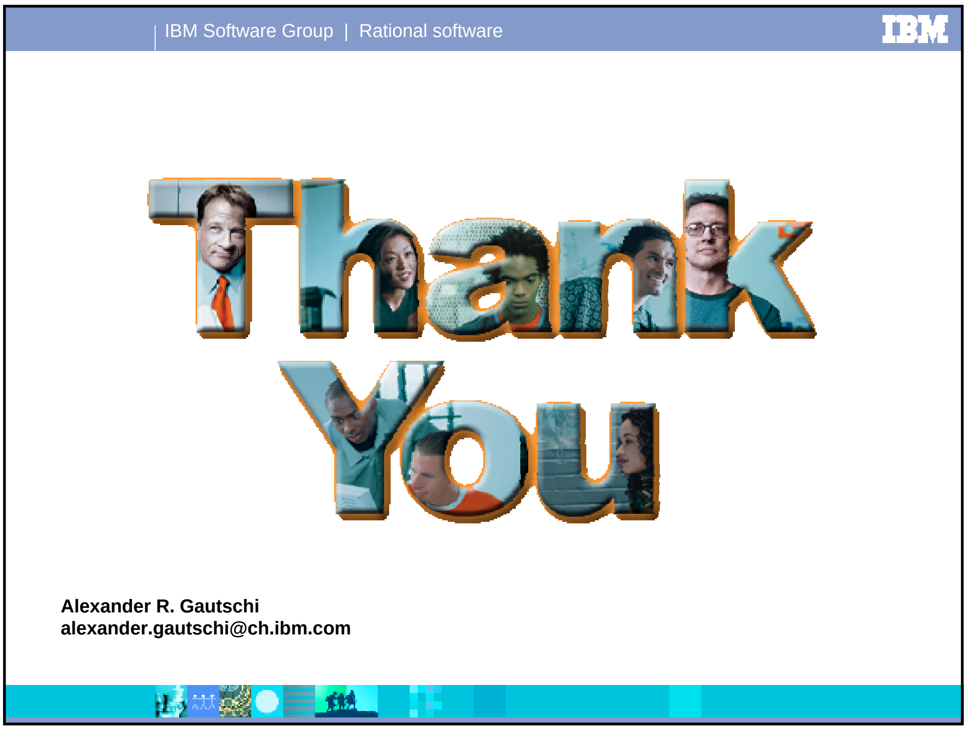
- Introduction
- Background
- Infrastructure
- Superstructure
- Profiles
- *Summary*



IBM Software Group | Rational software

Summary

- UML 2.0 is a planned major revision that
  - Balance of consolidation and feature extensions
  - Increased semantic precision and conceptual clarity
  - Streamline the kernel language
  - Increase customizability
  - Improve support for component-based development methods
  - Extended business process modeling support
- Evolution rather than revolution
- Backward compatibility in mind
- Suitable foundation for model-driven development
- Expected availability: Summer 2004



IBM Software Group | Rational software

Alexander R. Gautschi
alexander.gautschi@ch.ibm.com