IBM Almaden Research Center

Laura Haas

IBM Distinguished Engineer
Director, Computer Science

Beauty and the Beast:

The Theory and Practice of Information Integration

Bertinoro

September 30, 2007

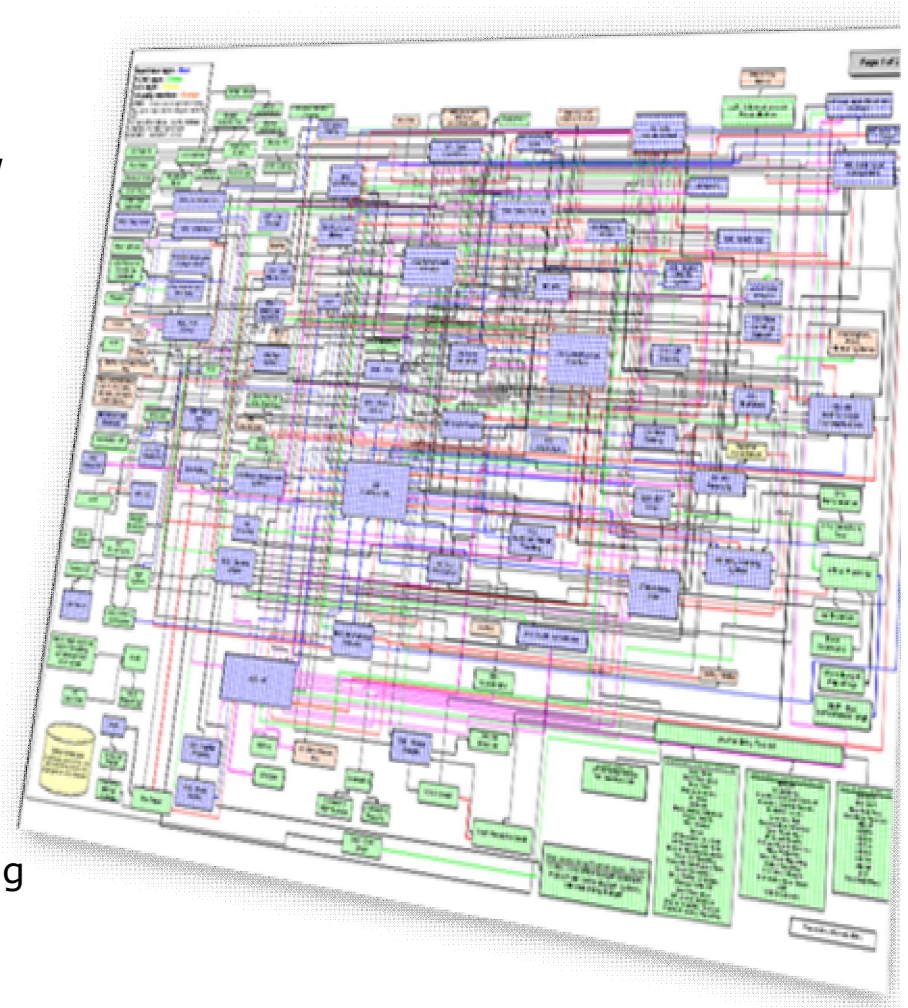
IBM

Agenda

- The Beast
  - What makes integration hard?
  - A customer success story
- Taming the Beast
  - The state of the art in commercial integration technology
  - A simple integration scenario
  - The beast remains
- Breaking the Spell?
  - Raising the level of abstraction
  - Increasing automation

What Makes Integration Hard?

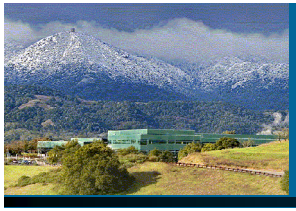
- Diverse sources and types of information
  - –Different models and operations
  - –Different interfaces
- Limited knowledge about the information
  - –Where is it from? How was it derived? How related to other information?
  - –Complex environments the norm
  - –Much of the knowledge is only in people's heads
- Overlapping and incomplete information
  - –Choice of source(s)
  - –Entity resolution
  - –Reconciliation of information
- Solution requirements may vary
  - –Is performance or availability (or both) important?
  - –How current must the data be? How accurate?
  - –Are there limits on storage space? Processing power?
  - –What policies must be respected?



Integration is a process

- Understand
  - What data is available
  - Important properties and values
  - Meaning or intent
- Standardize
  - Schema, field level, terminology and abbreviations
  - How to identify information about the same object
  - How to handle missing or inconsistent values
- Specify
  - Choose an integration engine or engines
  - Produce the executable(s) for integration
- Execute (Integrate)

- Iterate!!



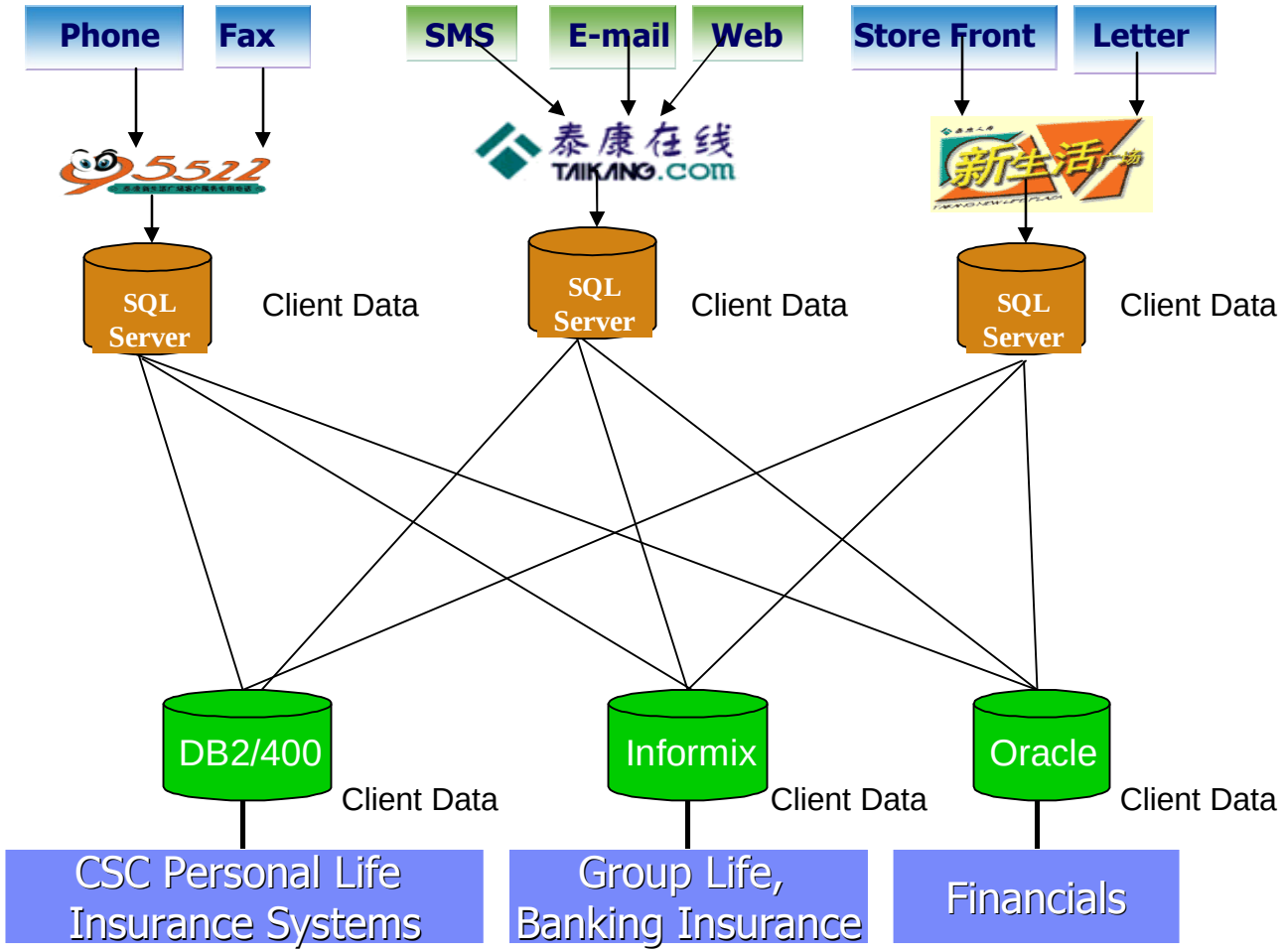
Taikang Life Insurance:
Getting a complete view of the customer

Challenge

- Provide a clear picture of customers and associated services
- Capitalize on new business and growth opportunities with an up-to-the minute view of the organization

Background

- 4th largest Chinese insurance company
- 8,000 employees, 150,000 agents
- 3.5 million customers
- Need information from DB2 UDB, Informix, Oracle, SQL Server, XML, e-mail, CRM and Portal



泰康人寿 TAIKANGLIFE

Taikang Integrated Information Platform Architecture

泰康人寿 TAIKANGLIFE

Channels

Phone | Fax | SMS | Email | Web | Store Front | Mail | Agents | Financial Planner

Application Platform

95522 | 泰康在线 TAIKANG.com | 新生活广场

XML | SQL | Web Services

Information Integration Platform

Data Service

Integrated Information

cache

SQL

ODS

Mapping (nicknames)

Core Systems

Informix — Group & Banking

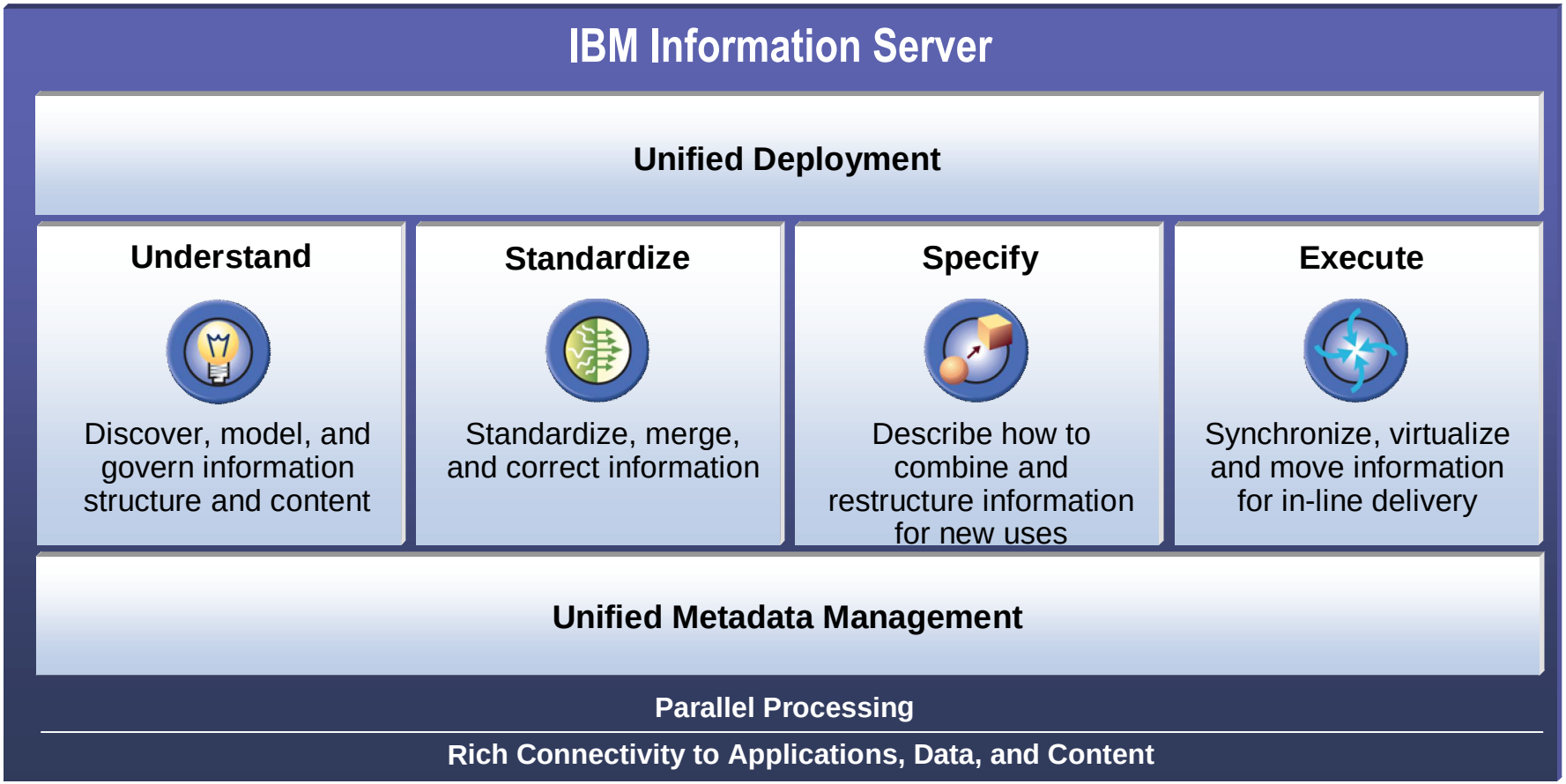
DB2/400 — CSC Personal Life

Oracle — Financials

IBM

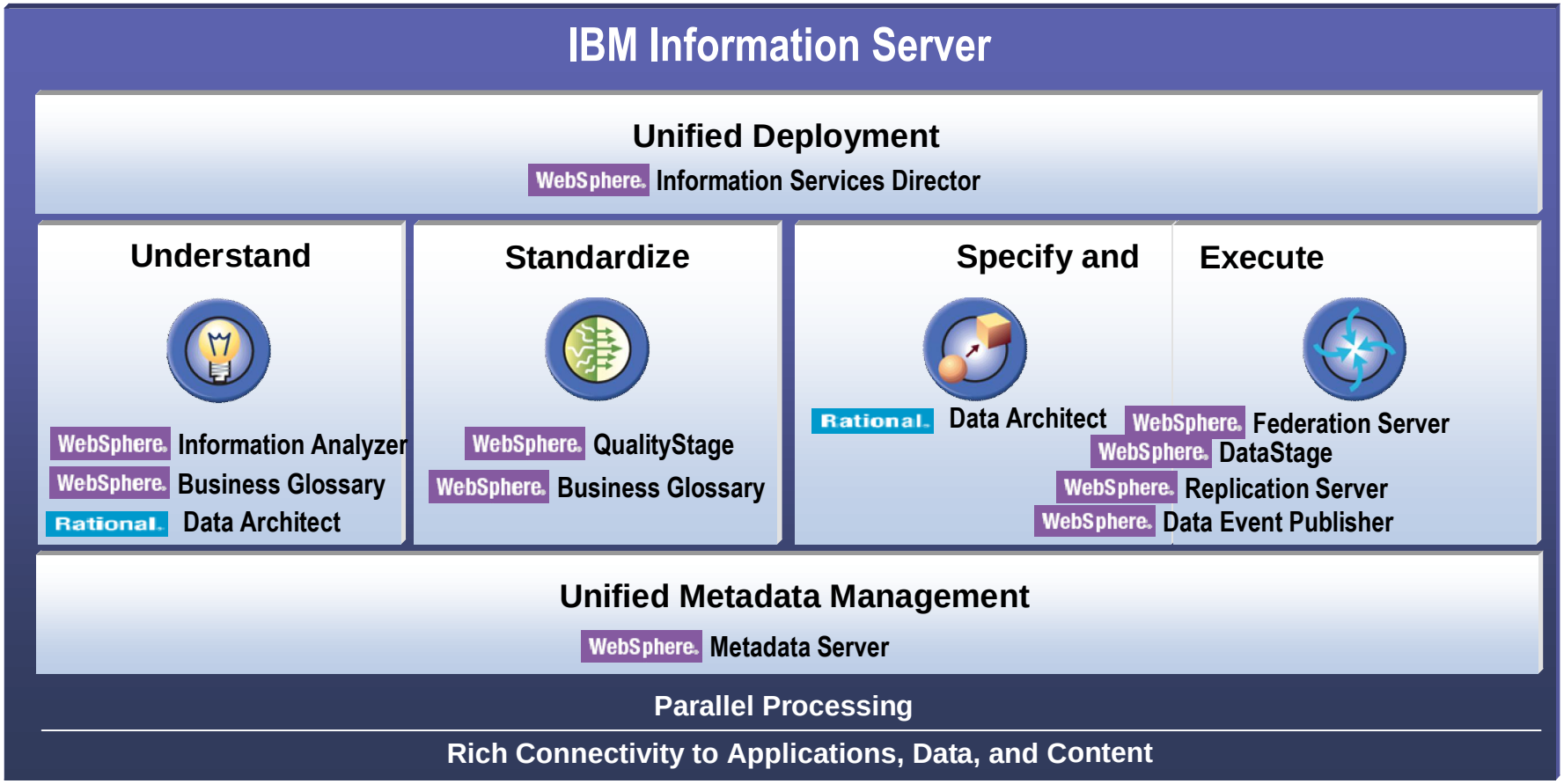
Marketing: IBM Information Server

Delivering information you can trust



IBM is the acknowledged industry leader for vision and execution in information integration

Reality:
Lots of products, lots of choices

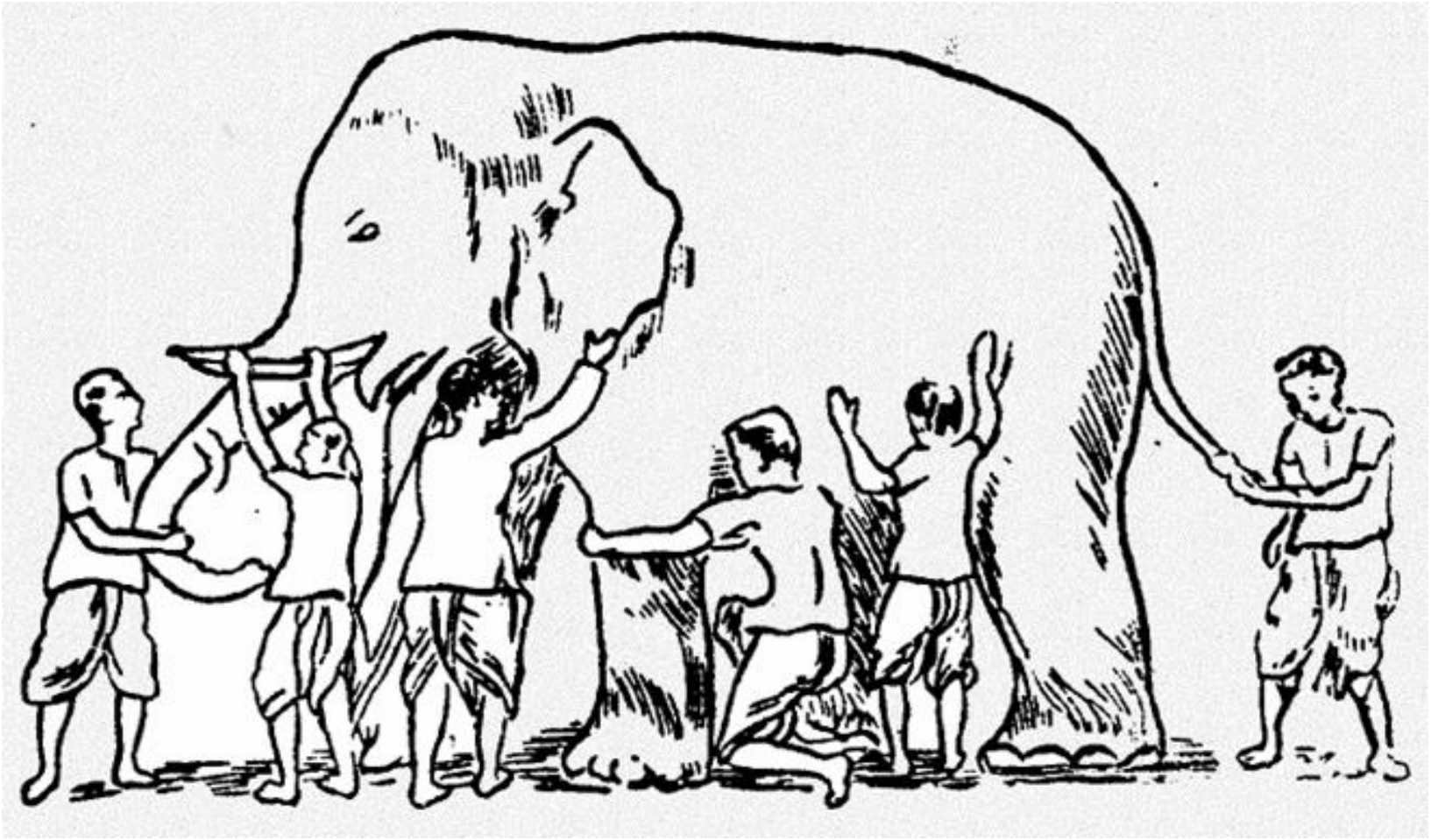


+ additional products for content federation, search, and special architectures or sources

IBM is the acknowledged industry leader for vision and execution in information integration

- We can provide an end-to-end solution
- Otherwise, the problem is much worse – many vendors, mismatched products

Information Integration is Just _____



Merging Chico and Grande

Grande Corporation

Customer DB

Cname	CID	Cmpy	Addr	City	ST	Zip	Slvl
Jane Doe	134	ABC	24, Grimm	Boston	MA	02138	Gold
Bob Dole	573	Banan	1832 15th St	San Jose	CA	95139	Silver

Chico

Orders

Mail Orders (CustId, ProdID, ServiceTyp)
Billing (CustID, Address)

Defect Tracking

Defects (ContactName, ProdID, Phone)

Warranties (Text)

Extended Warranty for Gyro Model 2

Super Service Guarantee T & Cs

Web Sales

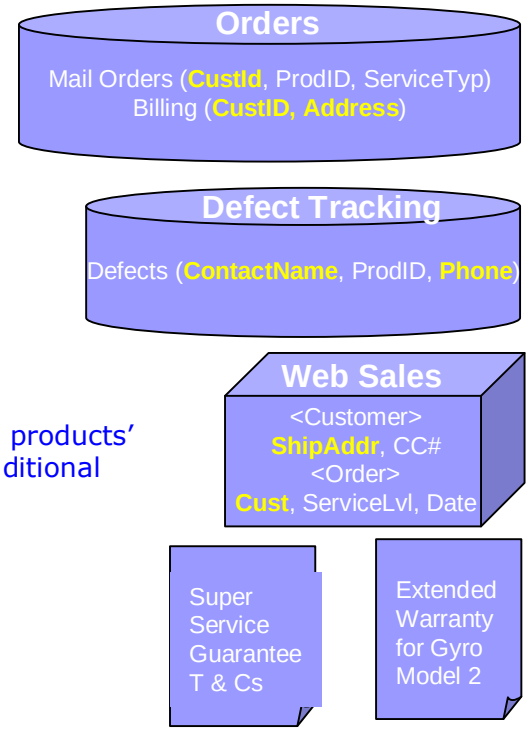
<Customer>
ShipAddr, CC#
<Order>
Cust, ServiceLvl, Date

Understanding What's There

- Where is the customer information?
- Assess the information quality
  - –Is it complete? Any conflicts? Duplication?
- Understand statistical properties
  - –Volumes? Important fields and distributions?
- Understand intent (semantics, meaning,…)
  - –How are fields related? Any constraints?
- And so on…
- Available tools
  - –Data explorers or profiling tools
  - –Discovery engines
  - –Enterprise catalogs
  - –Modeling tools
  - –Ontologies and business glossaries

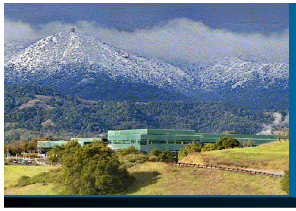
ABC is both a mail order and web customer of Chico – and of Grande

Chico service levels relate to products' warranty and purchase of additional warranties, if any

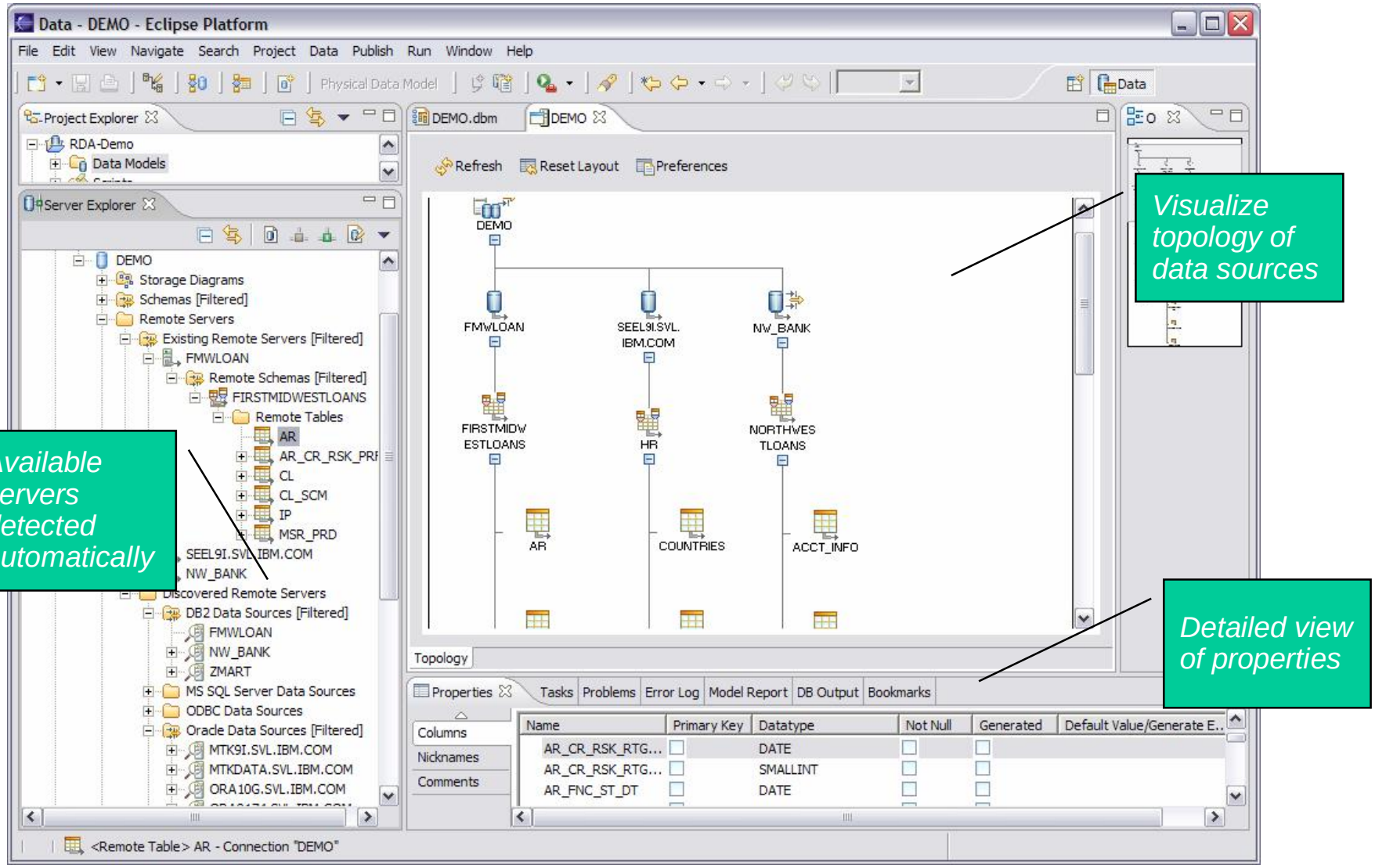


Customer DB

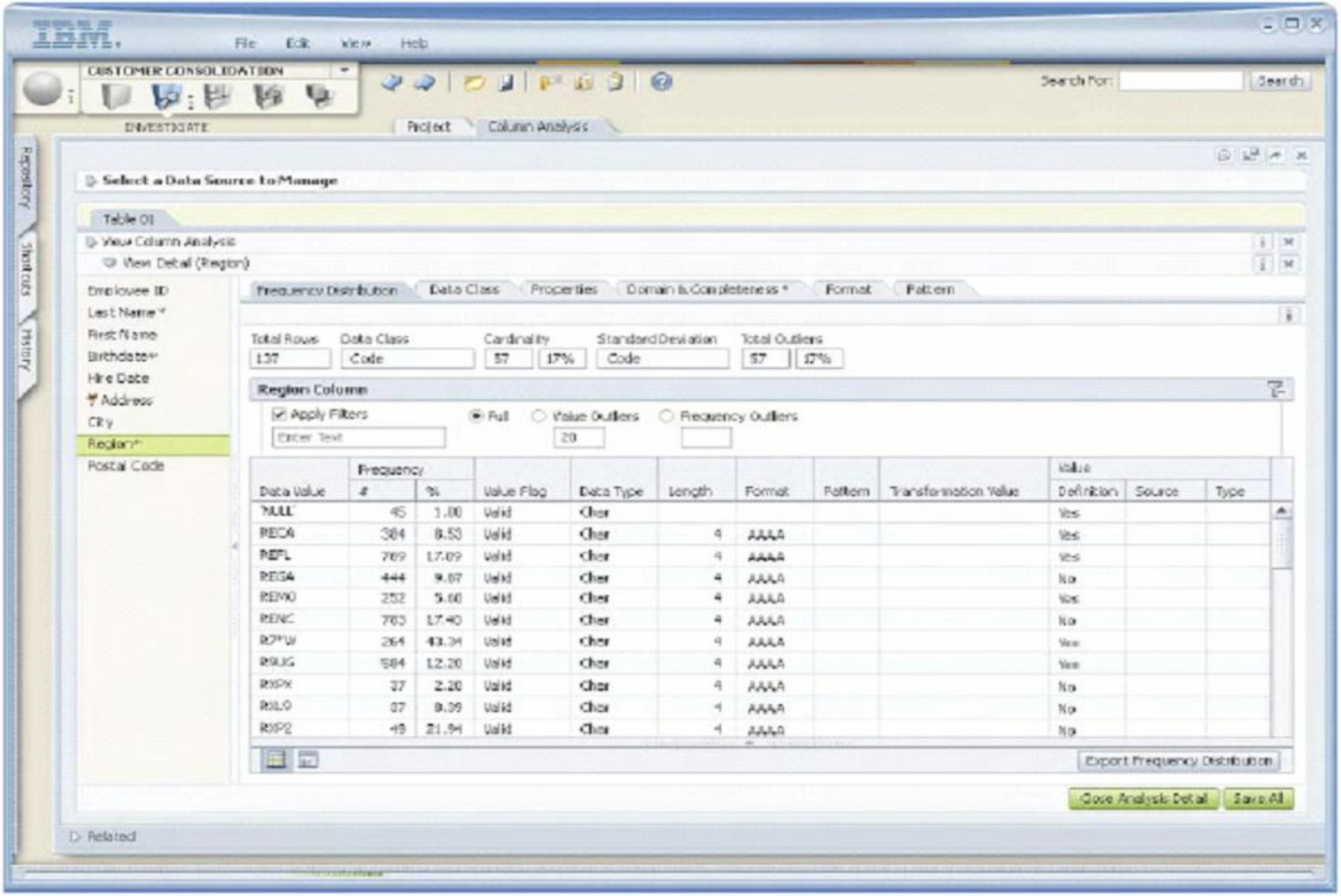
Cname	CID	Cmpy	Addr	City	ST	Zip	Slvl
Jane Doe	134	ABC	2, Grimm	Boston	MA	02138	Gold
Bob Dole	573	Banan	1832 15th St	San Jose	CA	95135	Silver



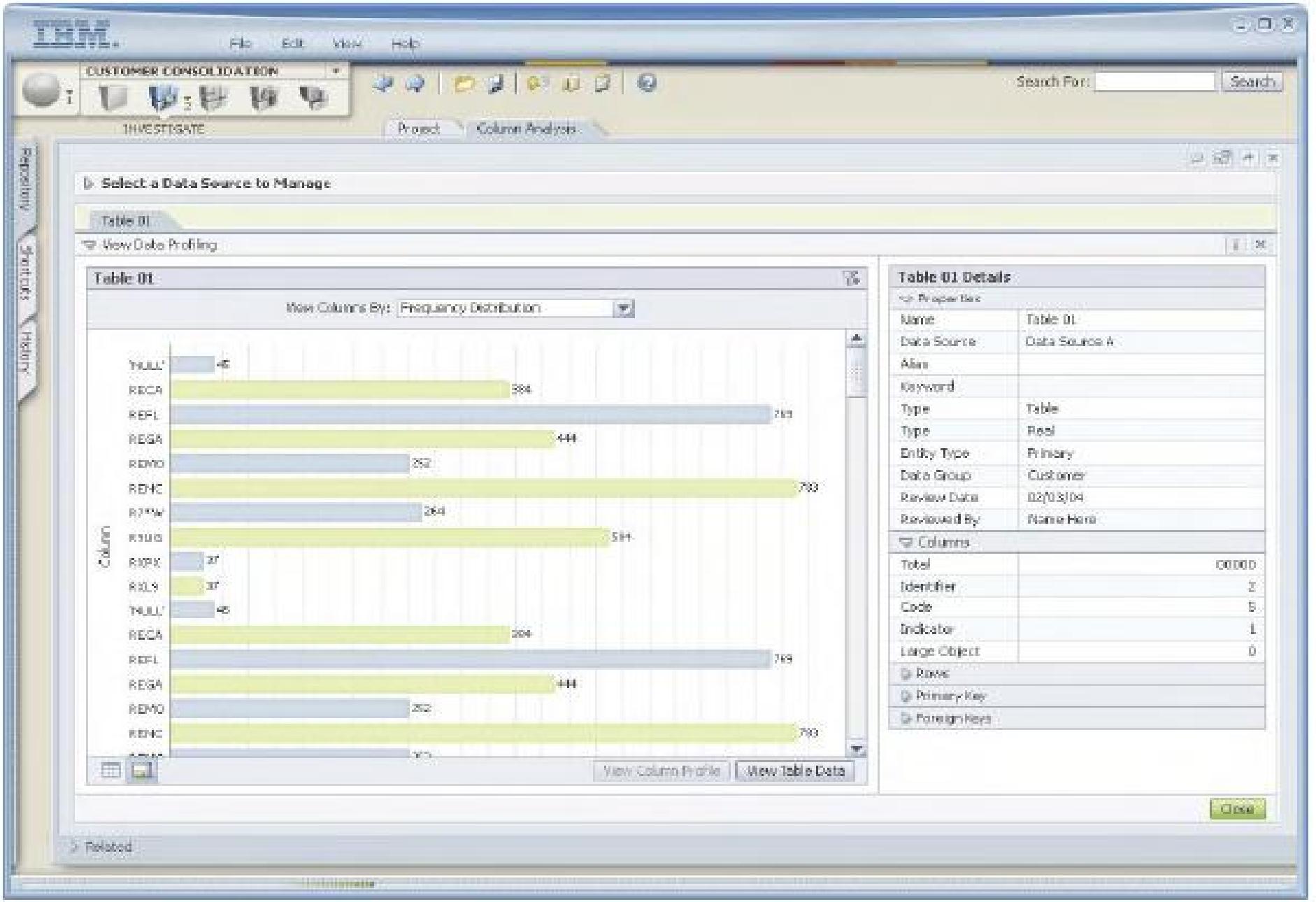
RDA: Discover and visualize data sources



Information Analyzer: Looking at Data



Information Analyzer: Looking at Data



Standardizing the Representation

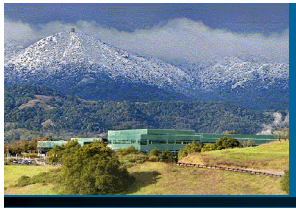
- Determine how the information should be represented
  - Target schema
  - Field level data formats
  - Abbreviations and terminology
- Rules for repairing information
  - Detecting overlap (object resolution)
  - Dealing with missing data
  - Handling inconsistencies
- Available tools
  - Schema modeling
  - Data cleansing
  - Business glossaries, ontologies
  - Triggers or other rule engines

Grande address: separate fields for street, city, state and zip. **Chico address**: a single string

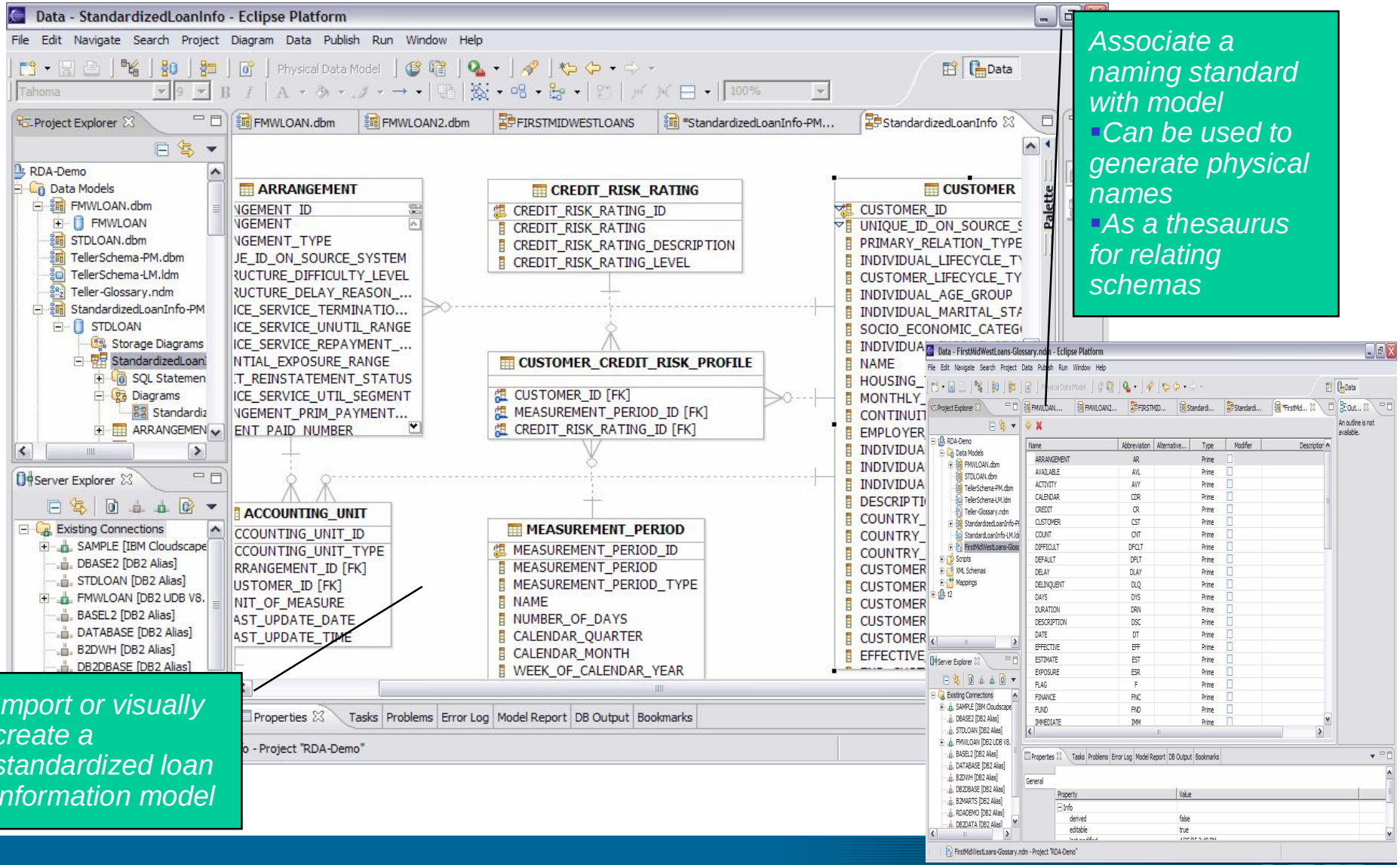
Last name, first name or first then last?

Street = Str or St?

ABC has different customer ids, and phone numbers and addresses in Grande and Chico dbs



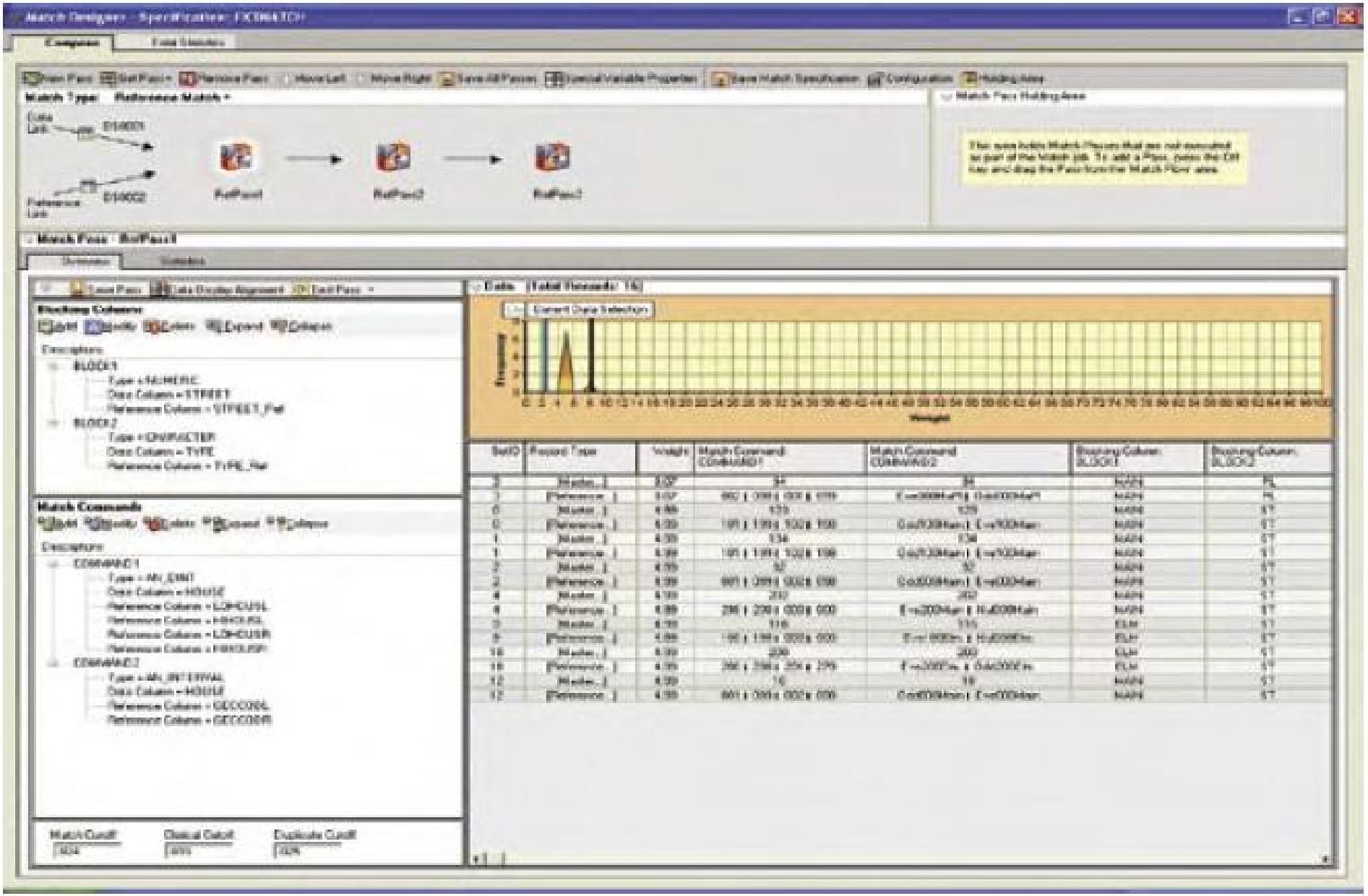
Create a standard model for loan information



Associate a naming standard with model
- *Can be used to generate physical names*
- *As a thesaurus for relating schemas*

Import or visually create a standardized loan information model

QualityStage Designer

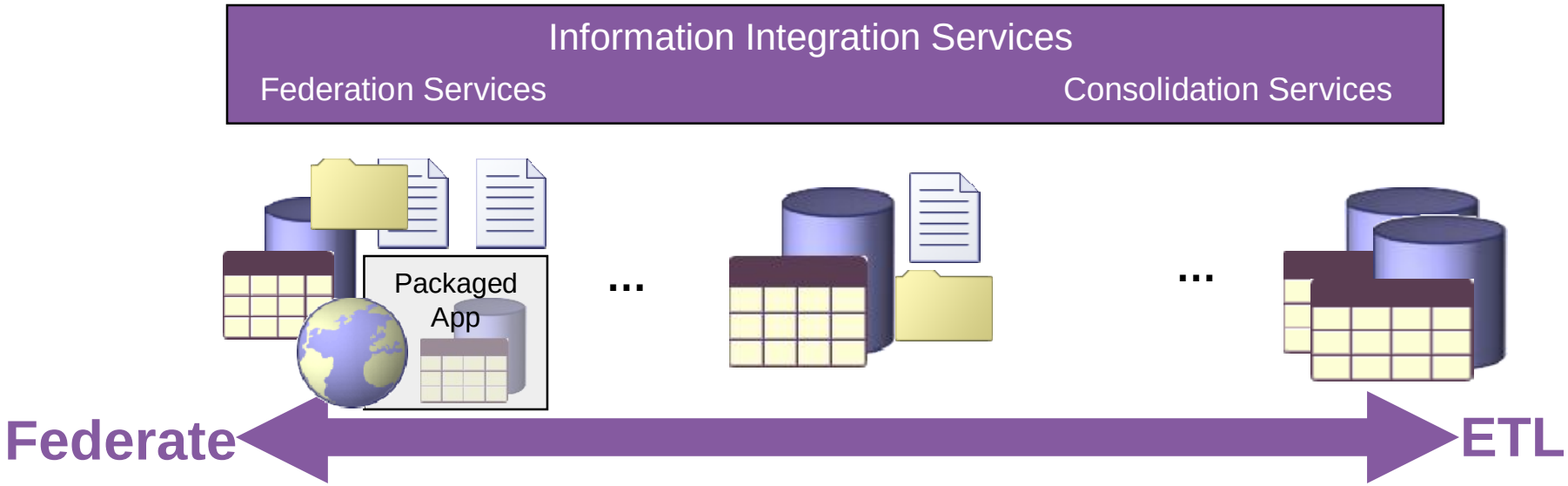


Specification and Execution

- What execution engine(s) should be used?
  - Materialize
    - Extract, Transform, Load (ETL)
    - Replication
  - Virtualize (Federate)
  - Search
- What input is needed to drive the execution engine of choice?
  - ETL: Dataflow graph or script
  - Replication: subscription definitions, e.g.
  - Federation: SQL queries
  - Search: Keywords
  - Plus, configuration of each engine (connectivity, tuning, …)
- Available tools
  - Each engine usually provides some configuration tool
  - Query builders for SQL
  - Many engines… all different, few standards

Choosing an Execution Engine: Example 1

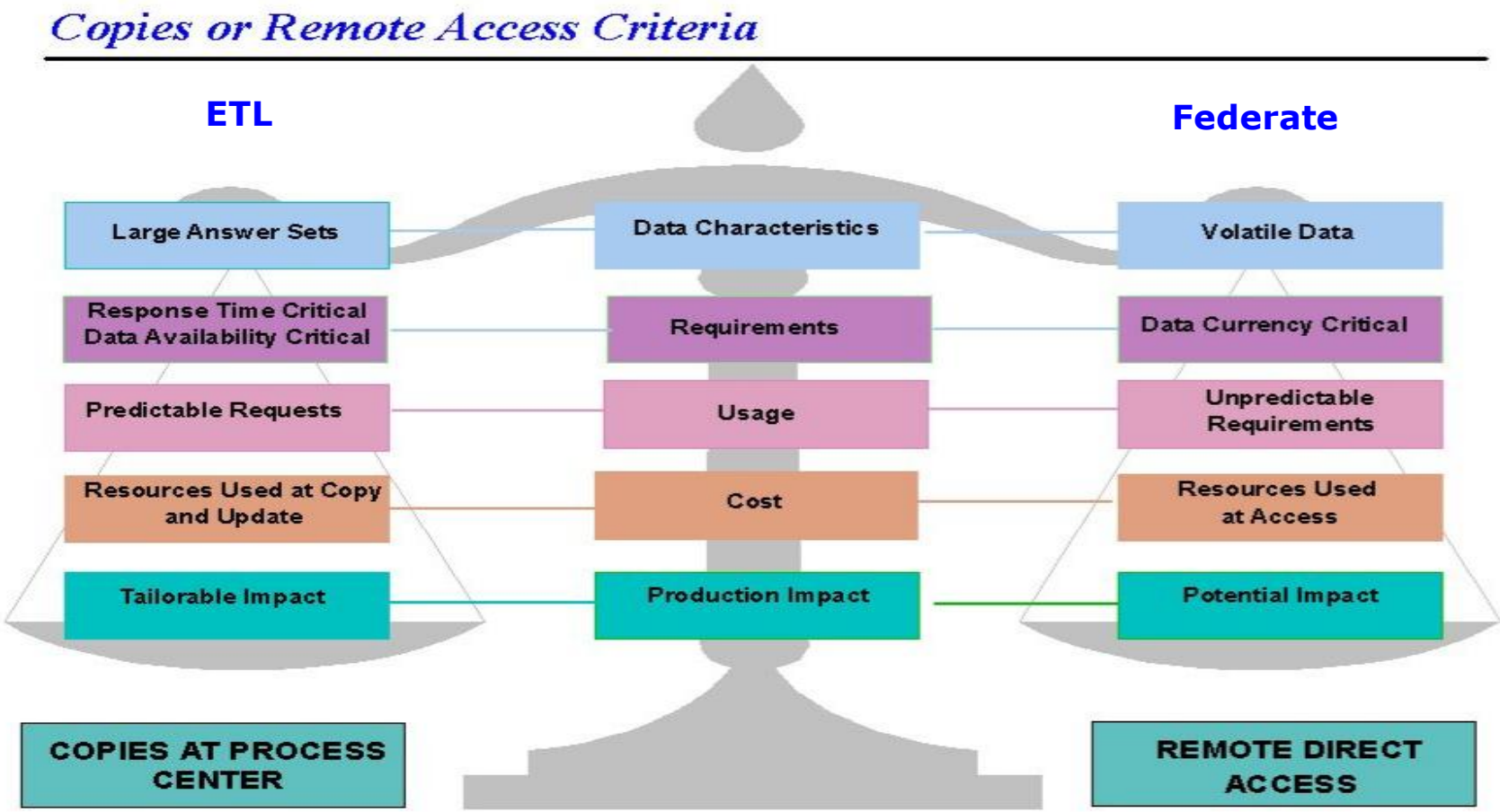
Materialize or Federate



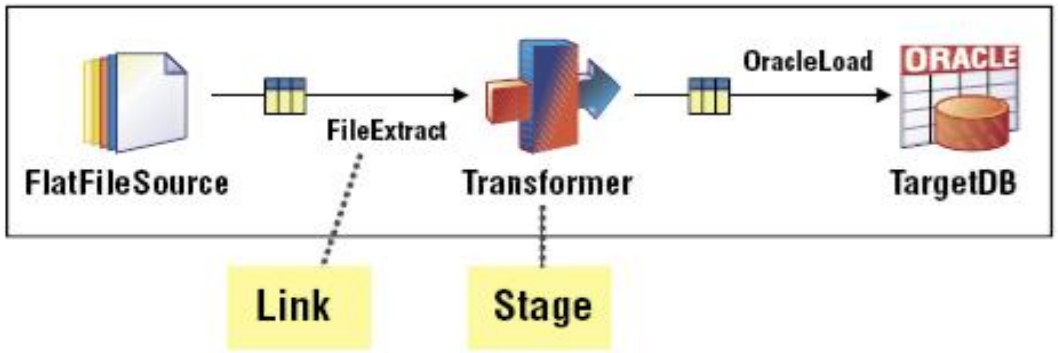
Key need:	very current data	high performance queries
Data volume:	moderate result sets	large result sets
Data types:	heterogeneous	largely relational
Availability need:	best effort	highly available
Copying:	independent of policy	must be allowed
Data processing:	SQL-based & optimized	data flow for special purpose engine

Choosing an Execution Engine: Example 2

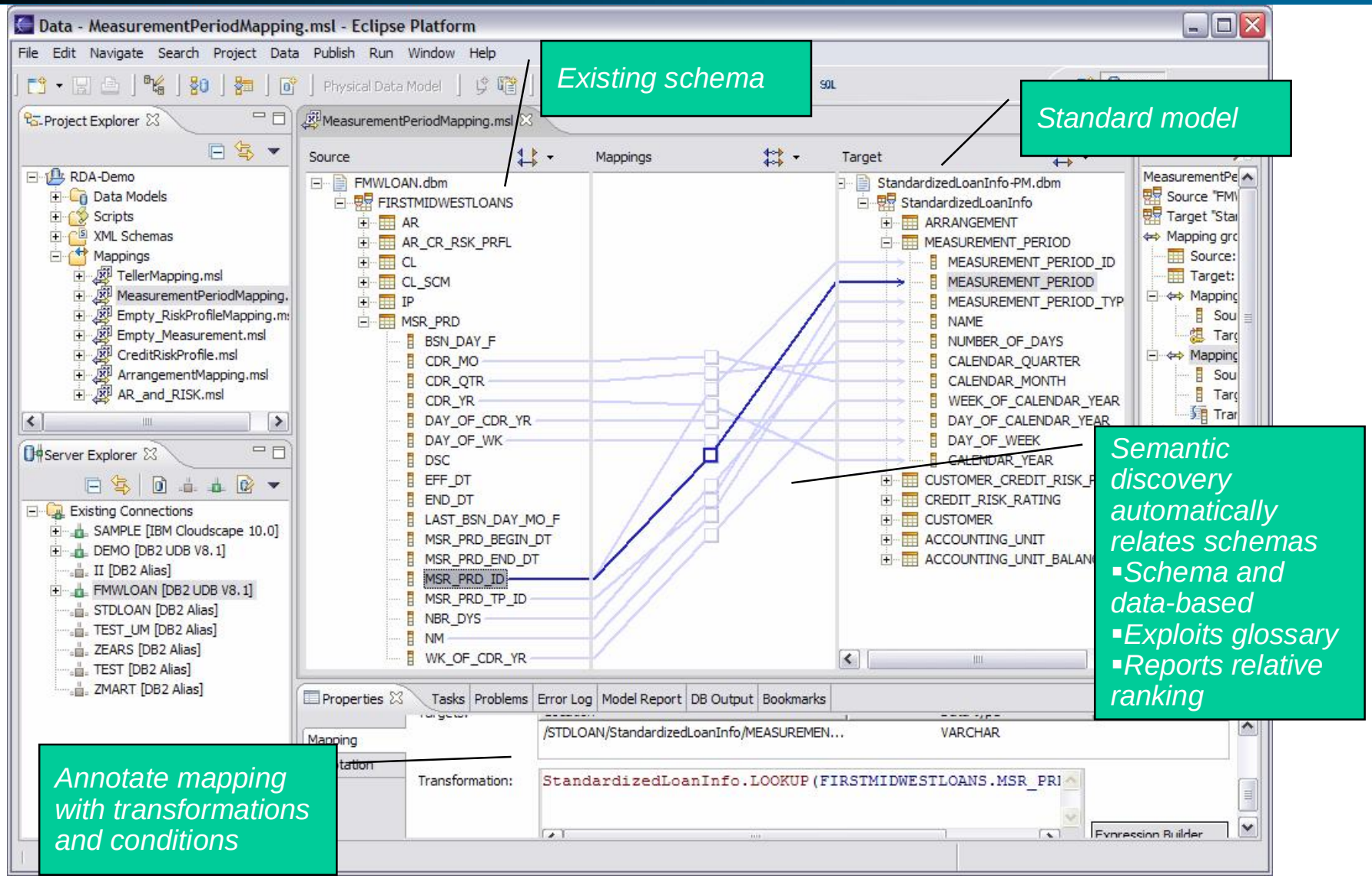
Materialize or Federate

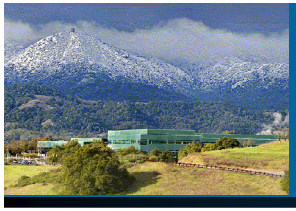


A Very Simple DataStage Job



Discover and relate existing schemas to standard





Relate and map existing schemas to standard

Data - CreditRiskProfile.msl - Eclipse Platform

File Edit Navigate Search Project Data Publish Run Window Help

Project Explorer

RDA-Demo
- Data Models
- Scripts
- XML Schemas
- Mappings
  - TellerMapping.msl
  - MeasurementPeriodMapping.
  - Empty_RiskProfileMapping.m
  - Empty_Measurement.msl
  - CreditRiskProfile.msl
  - ArrangementMapping.msl
  - AR_and_RISK.msl

Server Explorer

Existing Connections
- SAMPLE [IBM Cloudscape 10.0]
- DEMO [DB2 UDB V8.1]
- II [DB2 Alias]
- FMWLOAN [DB2 UDB V8.1]
- STDLOAN [DB2 Alias]
- TEST_UM [DB2 Alias]
- ZEARS [DB2 Alias]
- TEST [DB2 Alias]
- ZMART [DB2 Alias]

MeasurementPeriodMapping.msl | CreditRiskProfile.msl

Source	Mapping Groups	Target
FMWLOAN.dbm		StandardizedLoanInfo-PM.dbm
FIRSTMIDWESTLOANS		StandardizedLoanInfo
AR		ARRANGEMENT
AR_CR_RSK_PRFL		MEASUREMENT_PERIOD
CL		CUSTOMER_CREDIT_RISK_PRO
CL_SCM		CREDIT_RISK_RATING
IP		CUSTOMER
MSR_PRD		ACCOUNTING_UNIT
		ACCOUNTING_UNIT_BALANCE

Visualize relationships at table level

Maps can combined and composed

Properties | Tasks | Problems | Error Log | Model Report | DB Output | Bookmarks

Details
Filter
Join

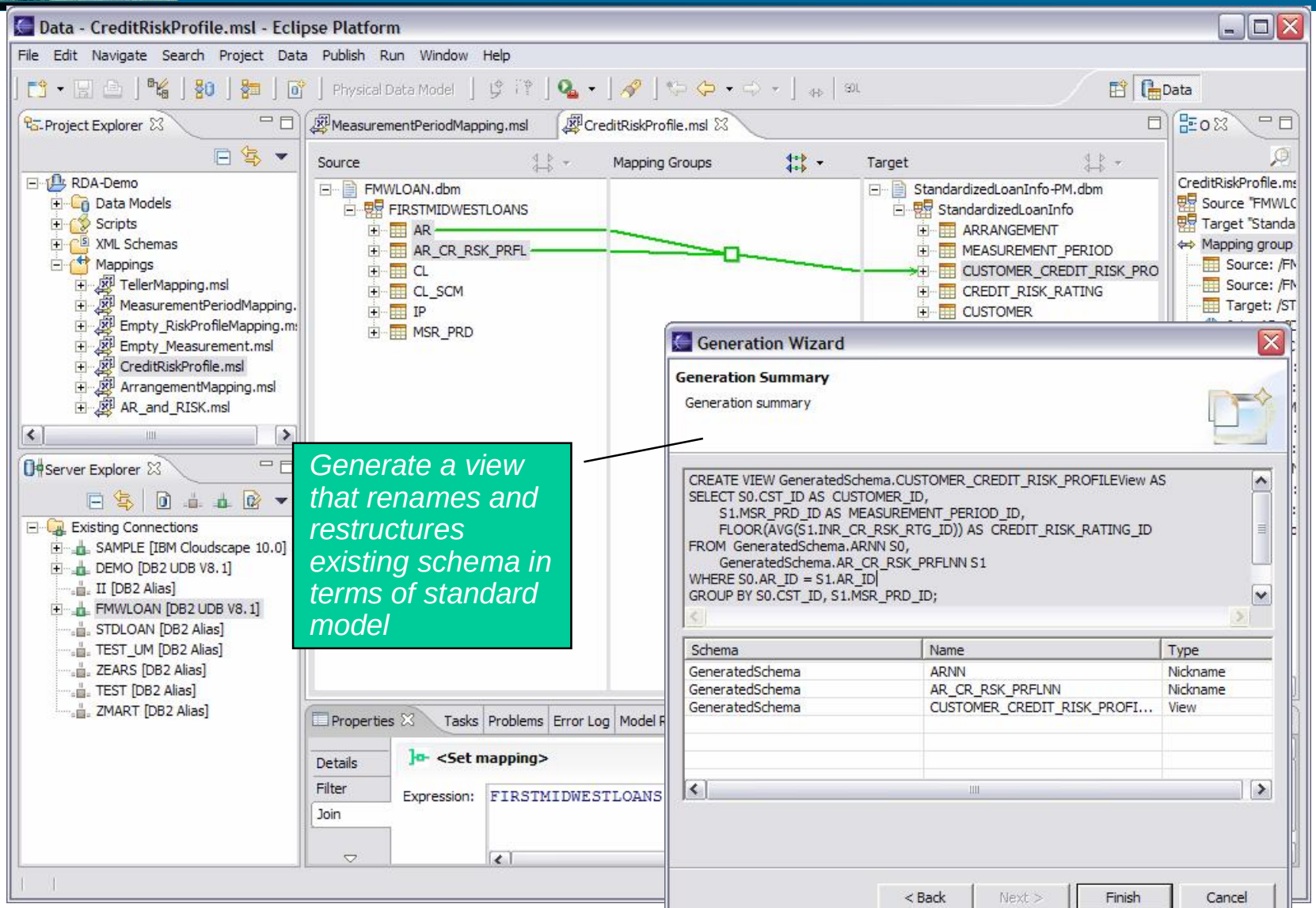
<Set mapping>

Expression: FIRSTMIDWESTLOANS.AR.AR_ID = FIRSTMIDWESTLOANS.AR_CR_RSK_PF

Expression Builder...

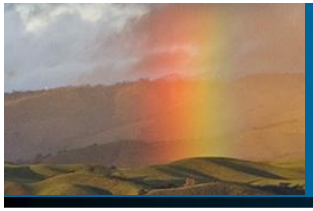
Discovers join conditions

Generate compliant view over existing schema



Customers Want Integration, Not Federate, Materialize, Search, …

- Hard and dangerous choices
  - Too many products with too much functional overlap
  - Too hard to choose, have to choose too early
  - Too hard to switch among products
- Too hard to use to build effective solutions
  - Too many knobs, too much training to use well
  - Too stove-piped (hard or expensive to use in combination)
  - Too little easy life-cycle flow among products
- Time to delivered value is too long
  - Need consulting-services engagements to survive
  - No way to kick-start the process
- Integration never ends
  - Integration usually starts with a project or two
  - Future projects should extend, reuse – avoid duplicate work



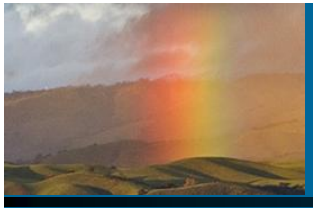
We Need to Simplify the Process

- Integration is the new data access
  - –Applications need information that comes from multiple sources
  - –Thus, writing applications that access data => integration
  - –But integration is too difficult

- We need to raise the level of abstraction
  - –Use "semantics" (more information)
  - –Generate configuration and execution artifacts automatically

- Goal: non-procedural integration
  1. Automated understanding and standardization
  2. Complete, high level specification
     - Logical: what information is wanted
     - Practical: what qualities must the solution have
  3. Tools to turn the high level description into actionable specifics
  4. Powerful engine (or engines?) to do the specified integration

- Research issues for both systems and theory communities



Some "formal" issues

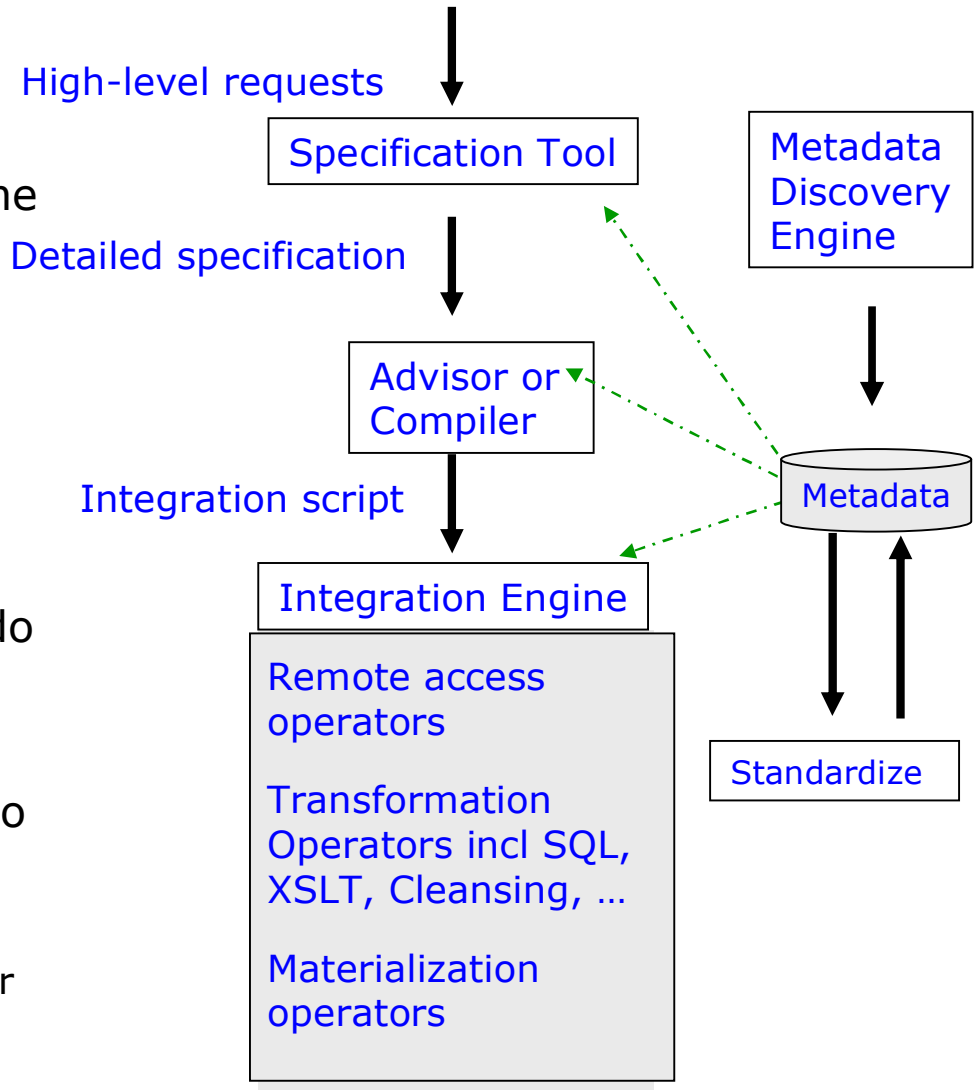
- Creating a complete picture of information integration
  - What are the fundamental operations for information integration?
  - Can we provide a theoretical basis for the information integration process?
  - Can data integration and data exchange even be viewed as different implementations of the same task?

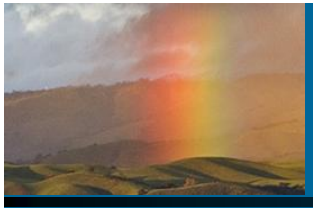
- **Hypothesis**: there is a set of requirements (the desiderata) that must be represented and reasoned about as part of any complete integration solution.
  - What are the critical solution desiderata?
  - How do they relate to the integration techniques needed?
  - Can we represent them formally? Which? How?
  - If they were formally represented, what could we learn? Do?

- Will a formal theory help create a single integration engine for all integration tasks?
  - Is this a worthy goal? (It sounds simpler…)
  - What is the analog of the relational calculus for integration?
  - How close can we come to fully automating integration?

Some systems issues

- What software do we need?
  - Automate understanding and standardization
  - Specify the solution desiderata
  - Integration Advisor to suggest the engine?
  - "Compiler" to generate the integration "script"?
  - How much can we automate?
- How should the integration engine be structured?
  - A "blade" approach: Specialized engines as needed to federate, do bulk transforms, search, …?
  - Or a "super-engine" that does it all?
  - How to make it simple yet able to meet a range of requirements?
    - Parallel infrastructure for performance, scalability?
    - Services-oriented architecture for flexibility?





Saying What Is Desired: Desiderata

- Logical: How to specify what information is needed?
  - –Naming?
    - "Information about customers", e.g. (not "cname from client")
  - –Form of the request?
    - From precise query to imprecise search
    - Language or a set of operators or ?
  - –Metadata?

- Practical: What other properties of the solution are desired?
  - –Qualities of service
    - Performance, availability, scalability, etc
  - –Qualities of data
    - Currency, completeness, accuracy, etc
  - –Physical constraints that must be met
    - Available storage, processing power, floor space, …
  - –Policies that must be obeyed
    - Privacy, security, cost sharing, legal, …
  - –Which of these are critical for determining integration techniques?
  - –How can they be modeled?

Summary: Can Beauty Tame the Beast?

- Information integration is a beast
  - –Hard and pervasive
  - –A multi-stage process
  - –Many (incompatible) choices of technology for each phase
  - –Little guidance on how to choose
- Research has looked only at aspects of the big picture
  - –Progress on some fundamental issues
  - –Integration technologies for pieces of the problem
  - –There are many open problems
- The *desiderata* are key
  - –Logical: target schema, constraints
  - –Practical: Qualities of service and data, physical constraints, policies
  - –No precise decision rules today
- Can we automate more of the integration process?
  - –Raise the level of abstraction
  - –Derive appropriate decision rules
  - –Use them to create nonprocedural integration

Thanks

- Phokion Kolaitis, Ron Fagin, Lucian Popa
- The Clio team
  - Renee Miller, Mauricio Hernandez, Lucian Popa, Howard Ho…
- IBM Information Integration Solutions development team
  - Lee Scheffler, Mike Beckerle…
  - Technical enablement and strategy teams
- The Garlic team
  - Peter Schwarz, Mary Roth…
- And many colleagues world-wide