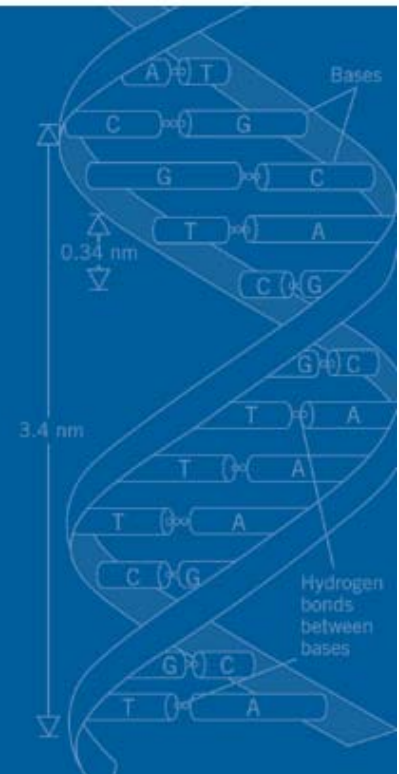


# Designing Canonicals for SOA: Bridging ER and XML Worlds

Mehmet Orun  
Jeff Pekrul

Prepared for DAMA, 2007

**Genentech**  
IN BUSINESS FOR LIFE



# Agenda

About the Presentation and Presenters

Services Oriented Architecture (SOA) Overview

Business Value, Terms, and Scenarios

Data Services – SOA relationship

Case Study: Enabling ESB through Canonical Design

Genentech environment and business need

# Purpose of this Presentation

- Sharing experiences to challenge and improve each other's knowledge
- Finding better ways of solving problems to make a difference in our organizations
- Educating data and development resources alike to understand each others' views

# About the Presenters

## Mehmet Orun

Mehmet Orun is the Principal Architect for Data Services in the Architecture and Engineering group of Genentech, responsible for data architecture, data services rollout, and associated technology roadmaps.

His interest and expertise includes data and meta data driven solutions to improve data quality, maintainability, and utility while bridging the gap across applications, processes, and structured and unstructured data sources.

Mehmet has an MBA in Management and Management Information Systems, and post graduate studies in Computer Science.

Mehmet is the Program Director for the San Francisco chapter of DAMA, and is a member of IDQ, MPO, and Taxonomy CoP.

## Jeff Pekrul

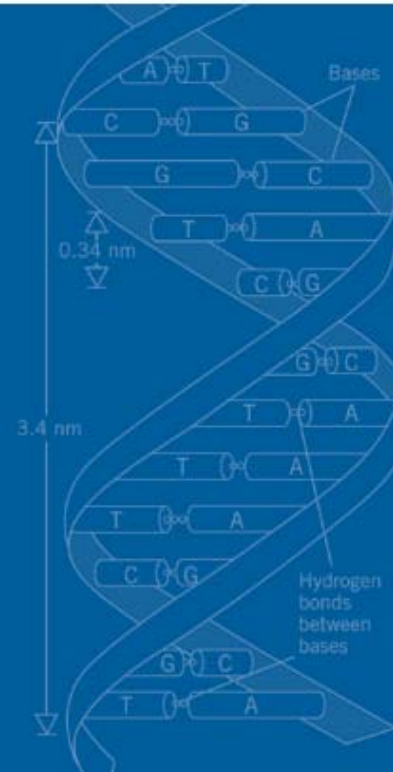
Jeff Pekrul is a data architect in the Business Solutions Engineering group at Genentech, responsible for canonical model development, and support of data integration projects that use EII technology.

Prior to Genentech, Jeff worked as a data modeler/data architect at Charles Schwab, Harmony Software and AT&T. Jeff has modeled databases in support of customer-facing applications in several industries, and has experience with data warehouse modeling as well.

Jeff has an MS in Information Systems Management, and is a member of the San Francisco chapter of DAMA.

# Services Oriented Architecture Overview

Business Value, Terms,  
and Scenarios



# Premise and Promise of SOA

Service-oriented architecture (SOA) expresses a perspective of software architecture that defines the use of loosely coupled software services to support the requirements of the *business processes* and *software users*. Resources on a network in an SOA environment are made available as independent services that can be accessed without knowledge of their underlying platform implementation.

(Channabasavaiah, Holley and Tuggle, Migrating to a service-oriented architecture, IBM DeveloperWorks, 16 Dec 2003)

Service Oriented Architectures allow

- **Development and delivery of *composite applications* that can leverage functions provided by other applications**
- ***Take advantage of proven implementations* without having to heavily customize those applications or create new custom applications with their own data store, business logic, and user interface**
- **Increase reuse and lower total cost of ownership**

# SOA Concepts

Concept	Definition	Analogy	Drivers
Software as a Service (SaaS)	A model of software delivery where a 3 <sup>rd</sup> party provides maintenance, technical operation, and support for the software provided to their client.	Application Service Providers (ASP). In addition to end-user access, data/web services interfaces to the backend data store.	Software vendors or hosting companies.
Composite Applications	Application with functionality drawn from different sources, e.g. individual web services, selected functions from other applications, or legacy systems whose outputs are packaged as web services.	Object oriented applications that reuse libraries and APIs	Internal IT initiatives.
Data Services	Enables [authorized] access to information sources from different sources.	Standards based APIs	EII, ESB
Web Services	A software system designed to support interoperable Machine-to-Machine interaction over a network.	Corba, EAI	ESB, EII

## Business Needs – IT Solutions

Sales Rep needs to track interaction with contacts.

Sales Force Automation (SFA) solution to capture interactions.

Sales Rep needs to track expenses. Expense submission system is part of the ERP system.

Enable SFA tool to capture and submit expense data to the ERP system, eliminating need for double data entry.

Sales Contact expenses must be tracked against state mandated compliance limits.

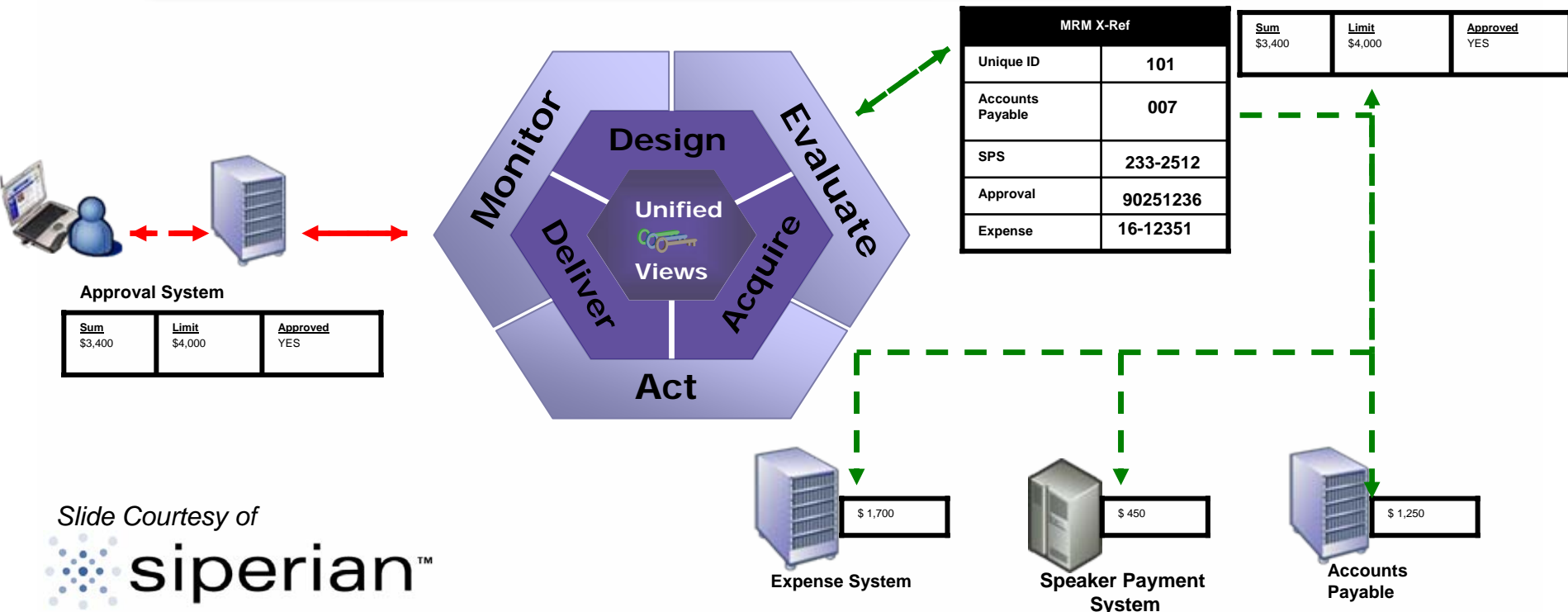
Traditional: As this is not standard capability, customize solution to capture and enforce spend rules within SFA.

SOA: Refer to the source with spend rules, compare past spend to limits, alert as needed.

# Scenario: Prescriber Spend Management

Prepared for DAMA International 2007

1. User makes a request for Prescriber spend approval for a specified State by passing the Customer ID and amount to AM
2. AM finds customer Xref in Hub
3. AM uses customer Xref to harvest existing spend data from other sources
4. AM forms and derives aggregate total across transactions, compares to state limit and calculates approval
5. Approval/Denial and other details such as current total and State cap info is passed back to the requesting system

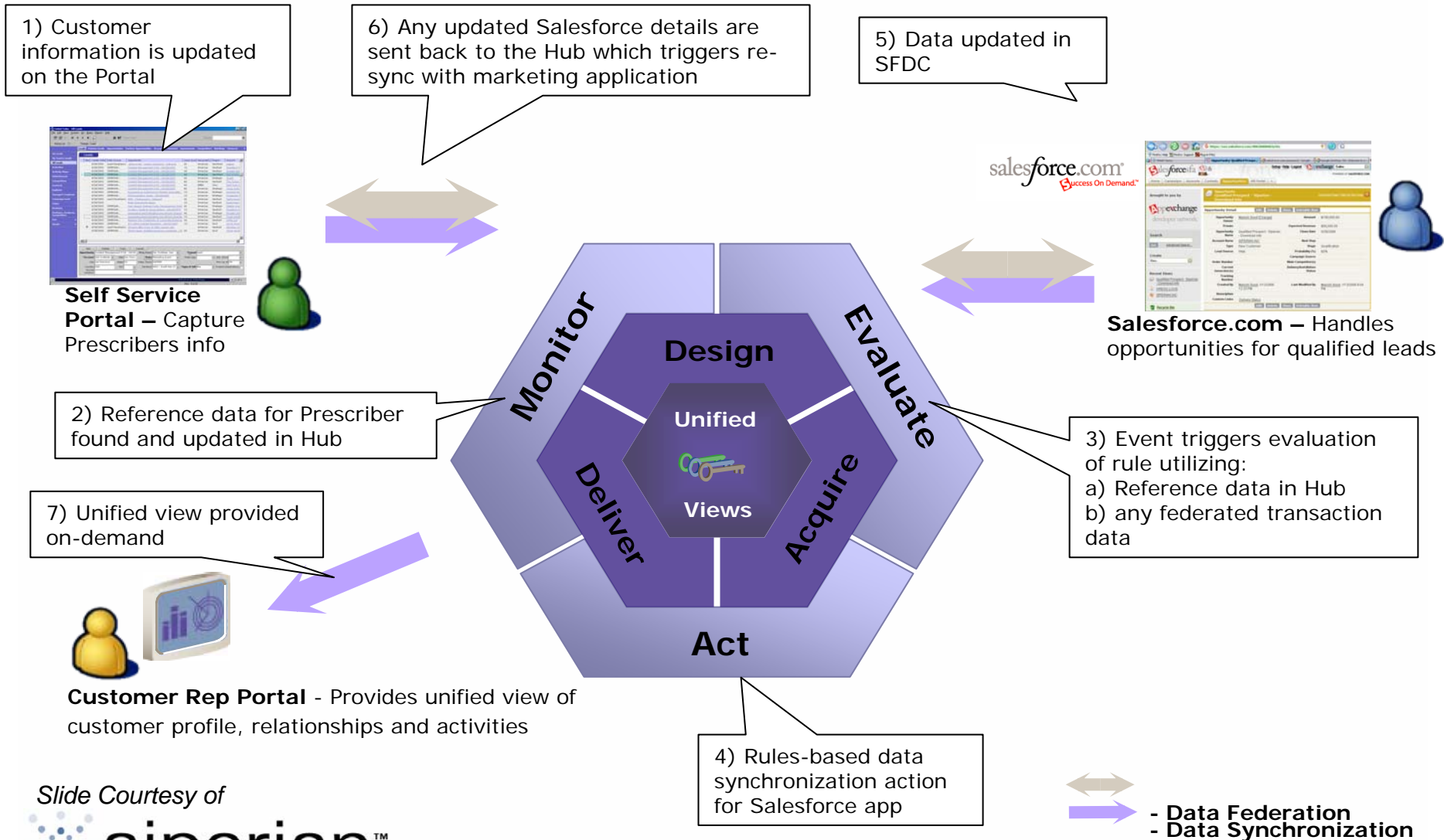


Slide Courtesy of



# Scenario: Data synchronization

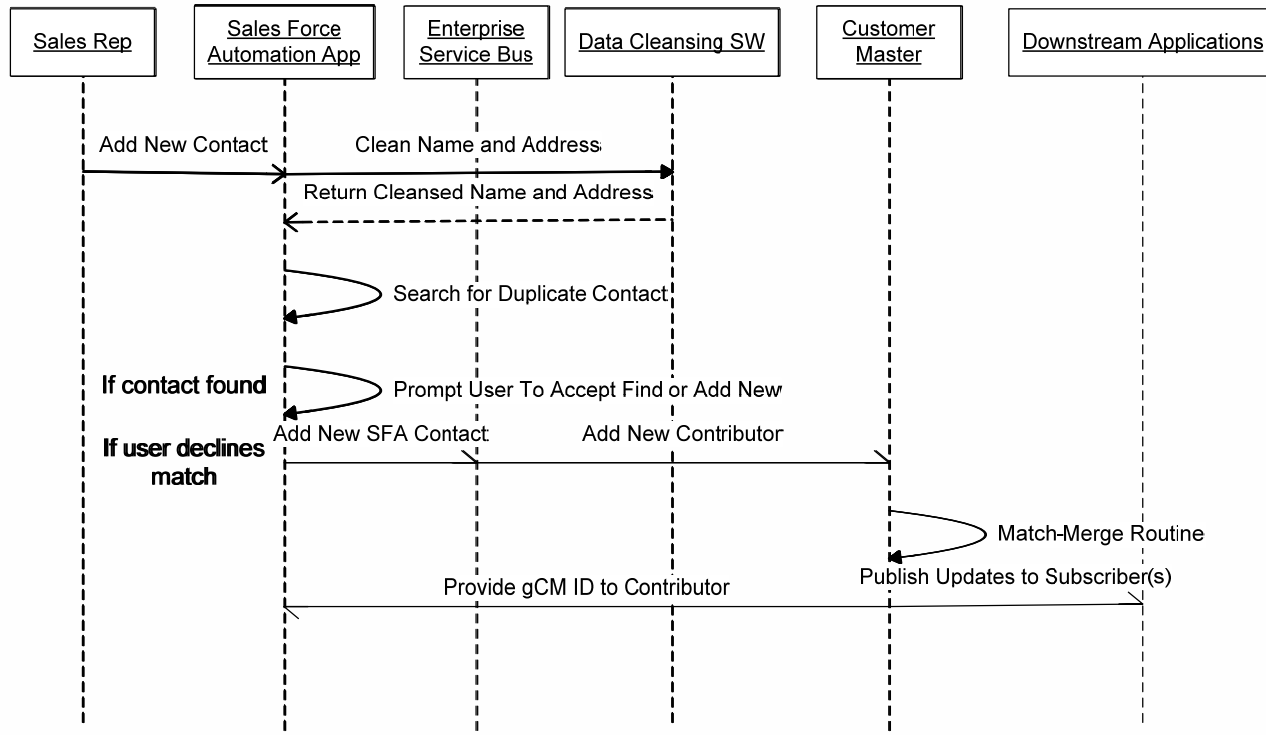
Prepared for DAMA International 2007



Slide Courtesy of



# In addition to lower TCO, Services can improve data communication and quality



## Services in this scenario

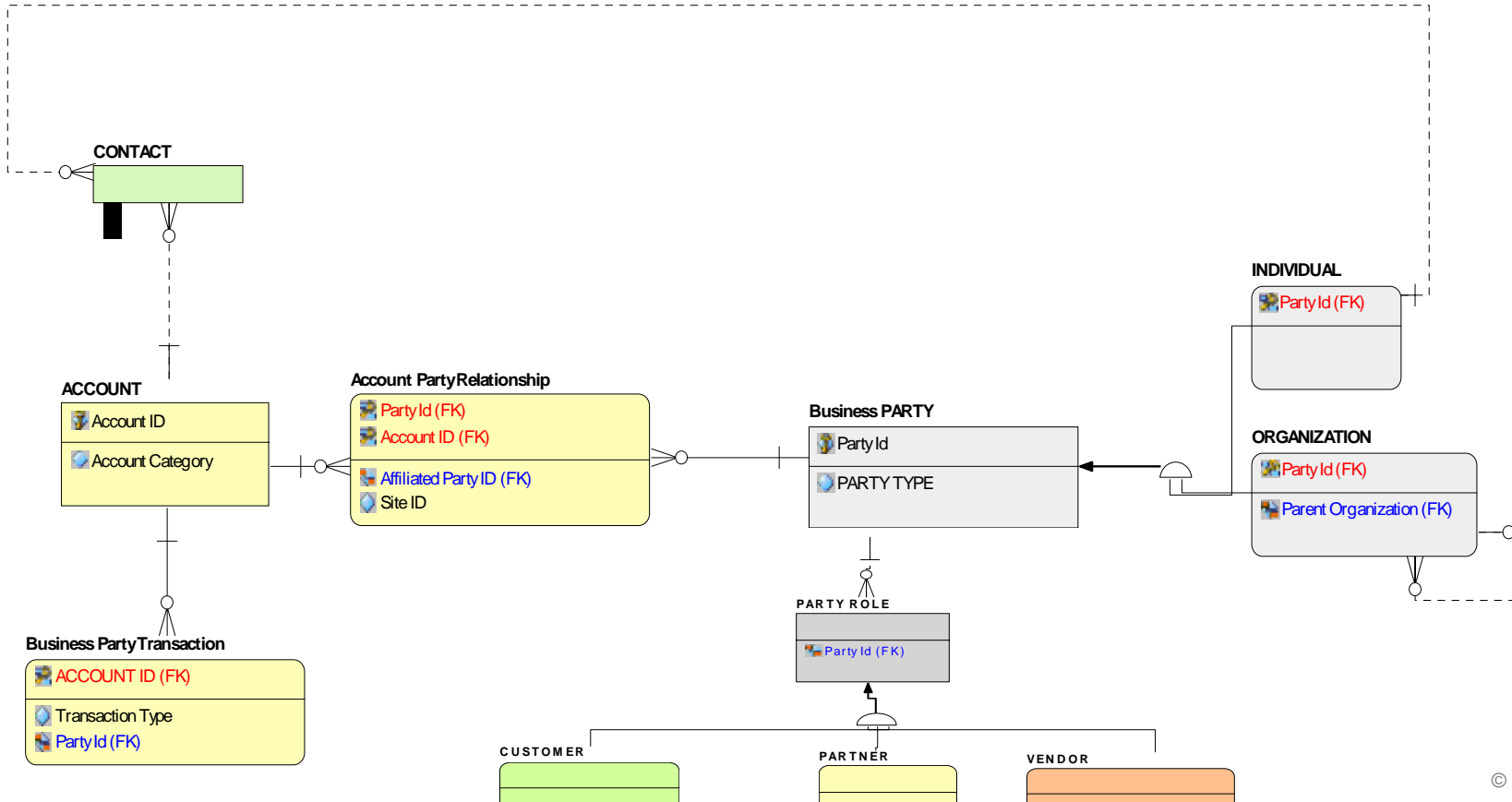
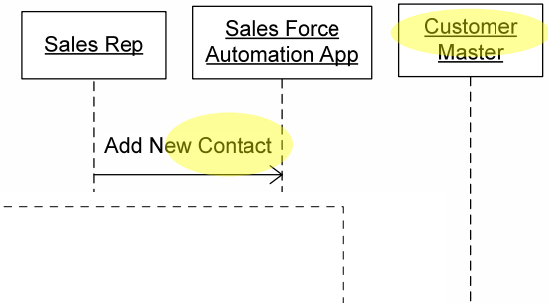
Cleanse Name  
 Cleanse Address  
 Lookup Name and/or Address  
 Add New Contact  
 Match Party Record  
 Update Party

## PS:

You also need to ensure downstream application updates do not trigger an infinite loop in the bus.

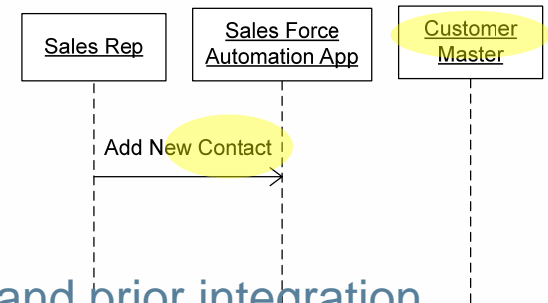
# Data Architecture Challenges in SOA

- [Web] Services cannot focus only on name or data element consistency
  - Is Contact and Customer the same thing?



# Data Architecture Challenges in SOA

- [Web] Services cannot focus only on name or data element consistency
  - **Is Contact and Customer the same thing?**



- This is not a new challenge. Data warehousing and prior integration technologies faced similar challenges.
- SOA's abstraction of "services" above applications require even better understanding of the data.
- Gartner advises that you should "develop an enterprise information management strategy as part of your SOA architecture. You will waste your investment in SOA unless you have enterprise information that SOA can exploit."
- Yet, most papers or books on SOA do not provide clear guidelines on how to specify and manage your services.

# Good news: Vendors are starting to offer frameworks. e.g. IBM's model for SOA

## Model:

- **Understand information assets and link to business context**
- Discover information metadata
- **Develop data & content models**
- Map information to business processes

## Assemble:

- **Compose information services**
- Extract, cleanse, transform & federate heterogeneous information

## Deploy:

- Service information requests
- Deliver unified data & content
- Deliver business context
- Discover relationships

## Manage:

- Monitor & manage Information
- Ensure performance, availability & security meet service levels

## Governance & Processes:

- Align business with IT information needs
- Monitor information usage over time
- Define & refine information management rules & policies

Source: <http://www.ibm.com/software/data/ondemandbusiness/soa.html>

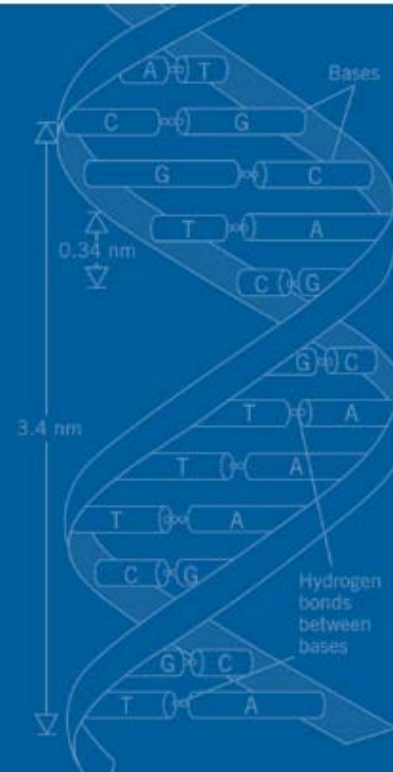


# Data Management skills support SOA

- Data Modeling [Focus of the next section]
  - Non-application specific representation of the business information
  - *Integration scenarios require additional considerations than storage centric data models*
- Semantics Management
  - Managing and mapping vocabularies, taxonomies, etc. and term relationships
- Data Analysis
  - Ability to understand information stored in sources rapidly to help with mapping design and testing
- Data Quality
  - Assessment, Cleansing, Monitoring, and Prevention of data quality within and across systems to ensure services provide reliable information

# A Real World Experience Enabling SOA Integration through Canonical Design

Bridging the ER-XML World

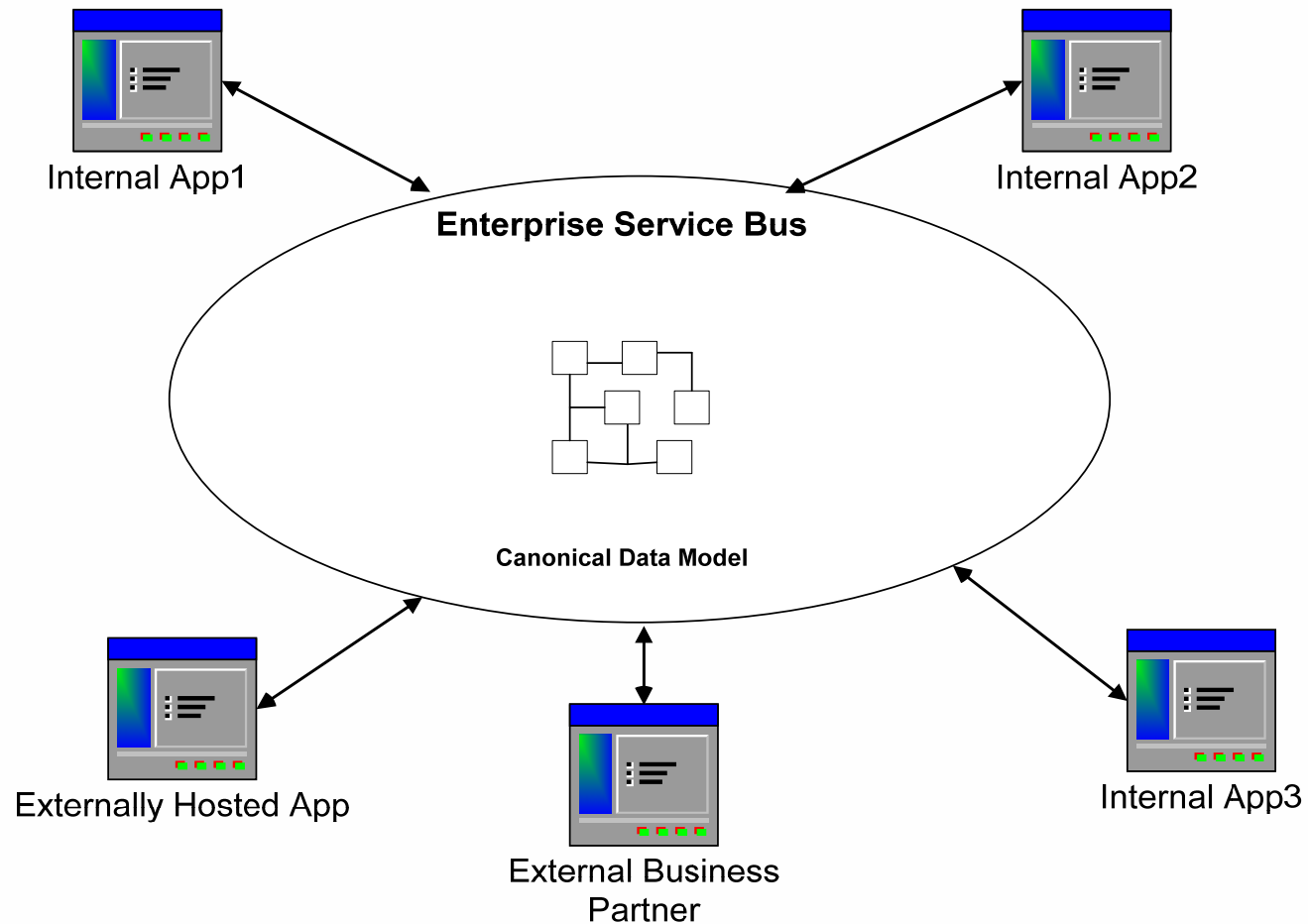


# What is an Enterprise Service Bus (ESB)?

*(Webopedia) ESB: (n.) Short for **E**nterprise **S**ervice **B**us, also referred to as a message broker. ESB is an open standards-based distributed synchronous or asynchronous messaging middleware that provides secure interoperability between enterprise applications via XML, Web services interfaces and standardized rules-based routing of documents.*

Along with a catalog of services and a registry, ESB is one of the foundational elements of an Service Oriented Architecture. ESB uses XML as the standard communication language.

# What is an Enterprise Service Bus (ESB)?



# What is a Canonical Data Model?

*(Webopedia): Canonical (**adj.**) Authoritative or standard; conforming to an accepted rule or procedure.*

*When referring to programming, canonical means conforming to well-established patterns or rules. The term is typically used to describe whether or not a programming interface follows the already established standard.*

The concept of a canonical message arises from the concept of a logical data model. A logical model provides a common construct for analysis, and a canonical message provides a common construct for interchange. (John Schmidt)

# XML Terms Relevant to Canonical Modeling

- XML schema – a way to describe and validate data in an XML document.
- XSD – (XML Schema Definition), a W3C standard for XML schemas.
- Instance document - an XML document passed via the ESB as a message, which can be validated using an XML schema.
- Schema document – a file containing XML schema definitions that conforms to a standard such as XSD. An individual schema document which is approved and authoritative is referred to as a ‘canonical’.
- Schema library - a collection of schema documents. The schema library may referred to as the ‘canonical library’.

# Industry Standard XML Schema Resources

- OAGi – Open Applications Group. This consortium provides an extensive library of XML standard schemas covering numerous subject areas
- B2MML – XML implementation of ANSI 95 standards, used to link ERP systems with manufacturing & supply chain management systems
- HR-XML – standards for human resources applications
- ebXML, cXML, xCBL – eCommerce standards

# SOA, ESB and Canonical Models – Summary

- ESB is a foundational technology necessary for implementing an SOA
- The value of SOA is predicated on the flexibility which results from reuse of business services
- Canonical models are required to enable data integration via the ESB, which in turn enables a Service Oriented Architecture
- The goals of canonical modeling include the definition of data structures which are reusable, durable and flexible

# Canonical Modeling at Genentech

- Primarily in support of data integration via the ESB.
- The ESB is implemented on the webMethods middleware platform, and has been in production for 2 years
- Both industry standard and custom canonicals are implemented on our webMethods ESB platform
- We use Entity-Relationship (E-R) modeling for our custom canonicals
- XSD schemas are generated from our E-R modeling tool. Some manual customization is required before these XSDs can be imported into the webMethods ESB tool

## Is E-R modeling relevant to ESB and SOA?

- Entity-Relationship and UML modeling are proven and effective ways of defining data structures that are durable, sharable, and application-independent.
- In the same way that a DDL script is a technical rendering of an E-R model for a relational database implementation, an XML Schema is a technical rendering for an ESB implementation.
- It is possible to generate both XML and DDL from the same E-R model. One E-R model can support both types of implementations.
- Business rules for both SOA services and ESB data integrations are sources of requirements for the canonical data model.

# XML Modeling Tools

- XML tools like XMLSpy allow graphical modeling of XML schemas directly.
  - **Advantages:**
    - Perfect match for XML constructs: simple and effective
    - Easy to represent XML sequence, choice, elements, attributes, etc. which are difficult or impossible to represent in E-R and UML models.
    - Easy to specify technical information in a natural way
  - **Disadvantages:**
    - May not scale to large models well. Difficult to work with models for large, complex projects because the visual language offers only one level of abstraction.
    - Doesn't work well on large projects that combine Java, SQL, and web services because others may be using UML or E-R.

## E-R and UML Modeling Tools

- We are using our standard E-R modeling tool (ER/Studio) for canonical modeling. This allows us to leverage our many existing E-R models, and our expertise with this type of modeling.
- ER/Studio can generate W3C-compliant XSDs. However, it lacks features needed for a smooth export to the ESB tool. We manually edit the XSD to remove certain tags, and to make the schema hierarchical.
- UML tools like Rational Rose may work as well as ER/Studio for generating XML schemas. UML models are also graphs, so the issue of relational-to-hierarchical translation would likely still be an issue.
- Since UML supports the definition of methods, it may have advantages when modeling both canonicals and services for SOA.

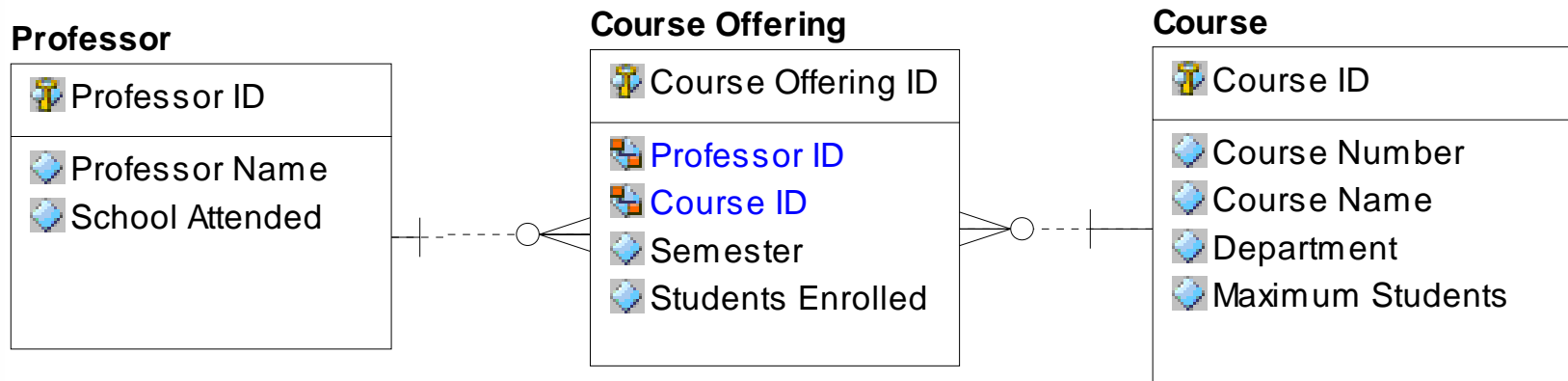
# Comparison of Tools for Canonical Modeling

	<b>Customization Required (e.g. graph to hierarchical)</b>	<b>Web Services design-time features</b>	<b>Re-use Existing UML and E-R Models</b>	<b>Scales Well to Large, Complex Projects</b>
<b>XML Modeling Tools</b>	No	Yes	No	No
<b>E-R Modeling Tools</b>	Yes	No	Yes	Yes
<b>UML Modeling Tools</b>	Yes	No	Yes	Yes
<b>ESB Tools</b>	No	Yes	No	No

# Hierarchical Implementation of a Relational Model

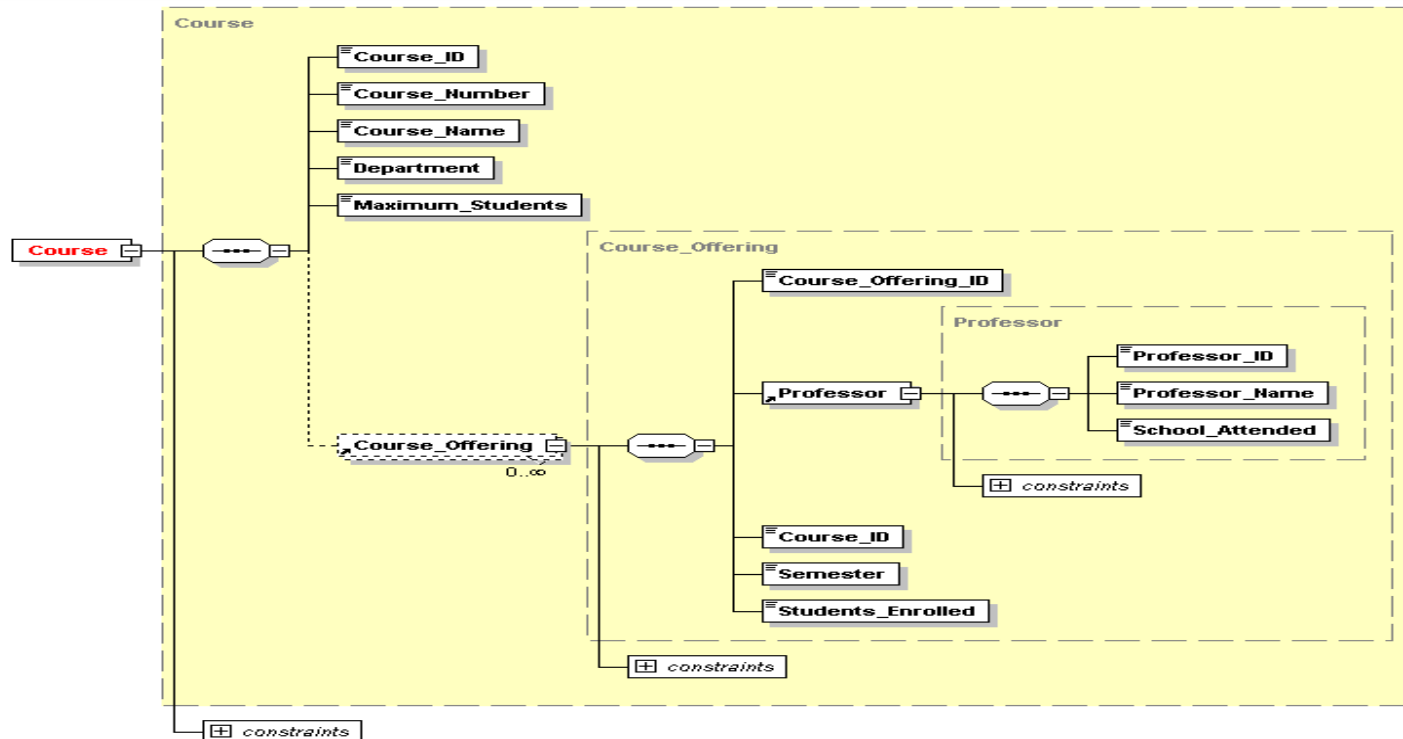
This simple relational model exhibits a common pattern that does not translate well into an XML schema.

In a hierarchical “tree” model, each node must have only one parent. An associative entity like Course Offering that resolves a many-to-many relationship has two parents.



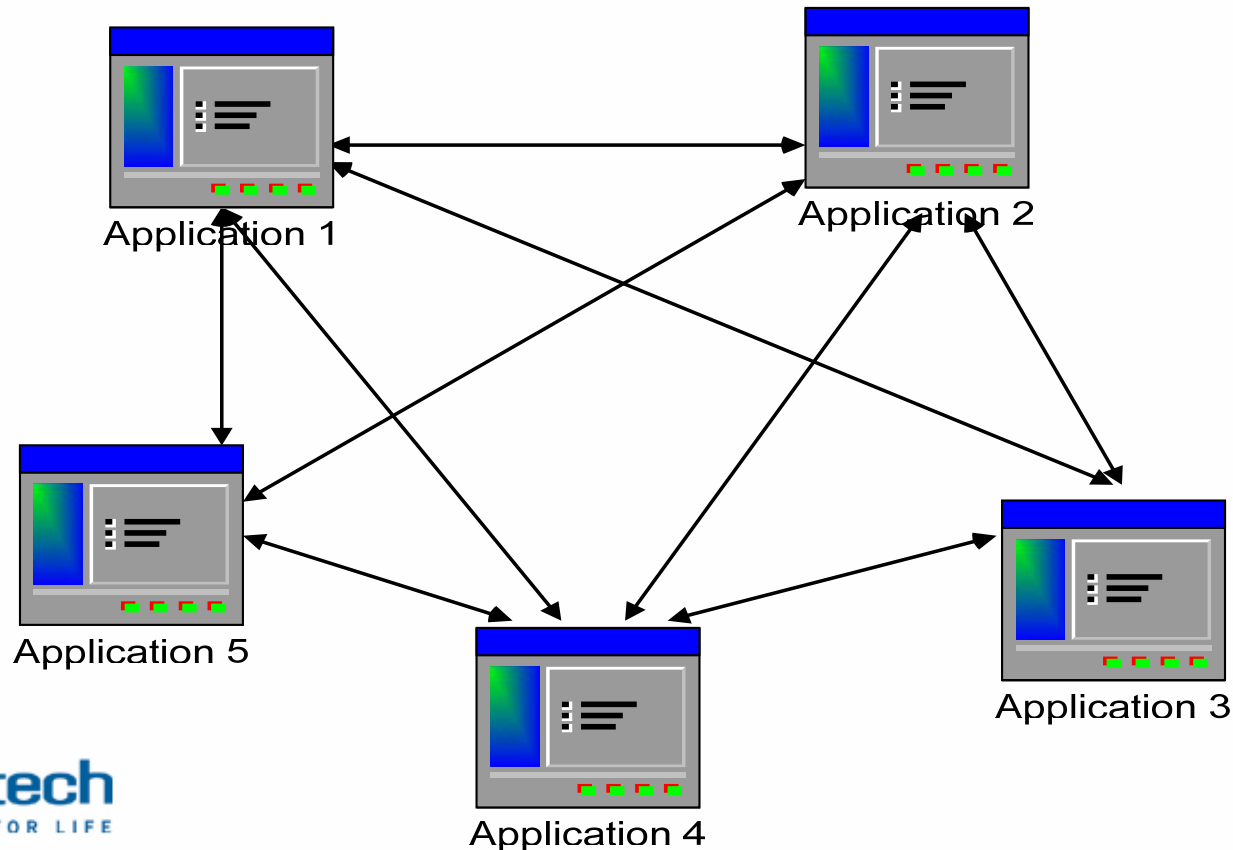
# Hierarchical Implementation of a Relational Model

An XML schema generated from this model where Course is the root. Professor must become a child of Course Offering.



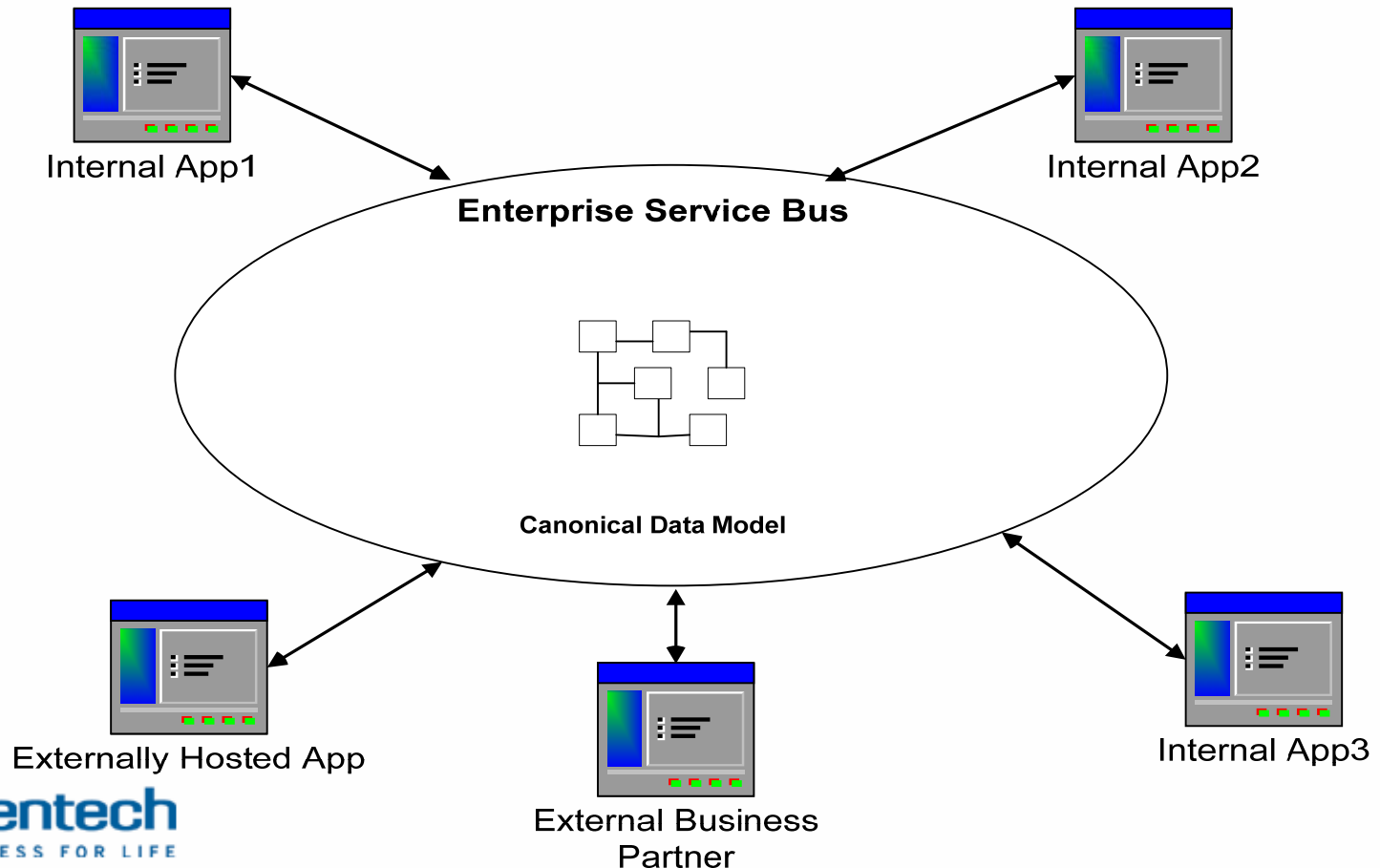
# An ESB Use Case for Canonical Models

Without an Enterprise Service Bus, point-to-point interfaces between applications become unmanageable and costly to maintain.



# An ESB Use Case for Canonical Models

Using the ESB, all apps can exchange data by mapping to a single data model – the canonical model.



# Issues Resulting from Differences in Data Models

## Application 1 Data Model

### Thought Leader

Thought Leader ID	NUMERIC(10,0)	NOT NULL
First Name	VARCHAR(100)	NULL
Last Name	VARCHAR(100)	NULL
Birth Date	CHAR(10)	NULL
Medical Specialty Code	VARCHAR(10)	NULL
Address ID	NUMERIC(10,0)	NULL
Alternate Address ID	NUMERIC(10,0)	NULL
Shipping Address ID	NUMERIC(10,0)	NULL

### Thought Leader Address

Address ID	NUMERIC(10,0)	NOT NULL
Address Line 1	VARCHAR(256)	NULL
Address Line 2	VARCHAR(256)	NULL
Address Line 3	VARCHAR(256)	NULL
Address Line 4	VARCHAR(256)	NULL
City Name	VARCHAR(256)	NULL
State Name	VARCHAR(256)	NULL
Country Name	VARCHAR(256)	NULL
Postal Code	VARCHAR(256)	NULL

## Application 2 Data Model

### Customer

Customer ID	INTEGER	NOT NULL
First Name	VARCHAR(50)	NOT NULL
Last Name	VARCHAR(50)	NOT NULL
Birth Date	DATE	NOT NULL
Medical Specialty Code	VARCHAR(10)	NULL

### Customer Address

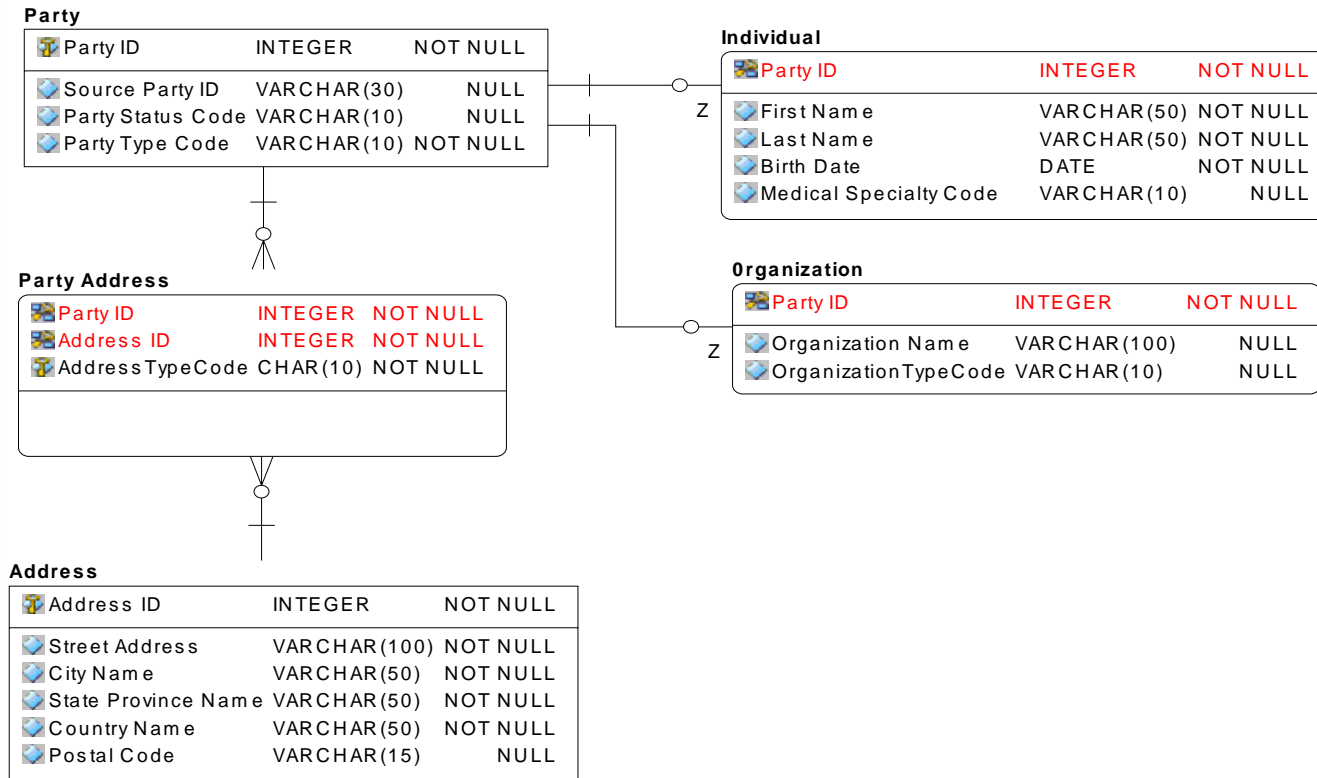
Customer ID	INTEGER	NOT NULL
Address ID	INTEGER	NOT NULL
Address Type Code	CHAR(10)	NOT NULL

### Address

Address ID	INTEGER	NOT NULL
Street Address	VARCHAR(50)	NOT NULL
City Name	VARCHAR(50)	NOT NULL
State Province Name	VARCHAR(50)	NOT NULL
Country Name	VARCHAR(50)	NOT NULL
Postal Code	VARCHAR(15)	NULL

# Issues Resulting from Differences in Data Models

## The Canonical data model



## Issues Resulting from Differences in Data Models

- Name and address fields are larger in App1 than App2. Data truncation may result when App1 passes data to App2. Should the canonical model be designed for maximum flexibility and allow this?
- Identify the system of record for all data represented in the canonical. In most cases, data representation in the canonical will closely resemble that used in the system of record.
- Whenever possible, identify data stewards who can review and approve canonical designs.

## Issues Resulting from Differences in Data Models

- App1 uses a CHAR field to store date data. Should the canonical use 'strong typing' which may result in App1 being unable to publish some documents to the ESB?
- App1 elements Address Line1,2,3 & 4 contain unfielded, low quality data. Does this data have to be cleansed before publication via the ESB?
- The canonical can play a role in data quality initiatives. By enforcing strong typing, and proper parsing of data before publication to subscribing systems, propagation of poor quality data can be prevented.

# Issues Resulting from Differences in Data Models

- App1 and App2 define different sets of values for the Medical Specialty Code. Which set of values is correct? If all ESB documents must conform to a standard set of values, how does the canonical model support this?
- In our canonicals, we may define enumeration values if a data steward has approved a standard set of values. Canonical modeling for the ESB has elevated the need for stewardship of reference data in our environment.
- Only code sets which are static, and small (< 20 values) are defined in our canonicals.
- Large and/or volatile reference codes are defined in a separate schema, or stored in database tables for lookup during ESB validation and mapping.

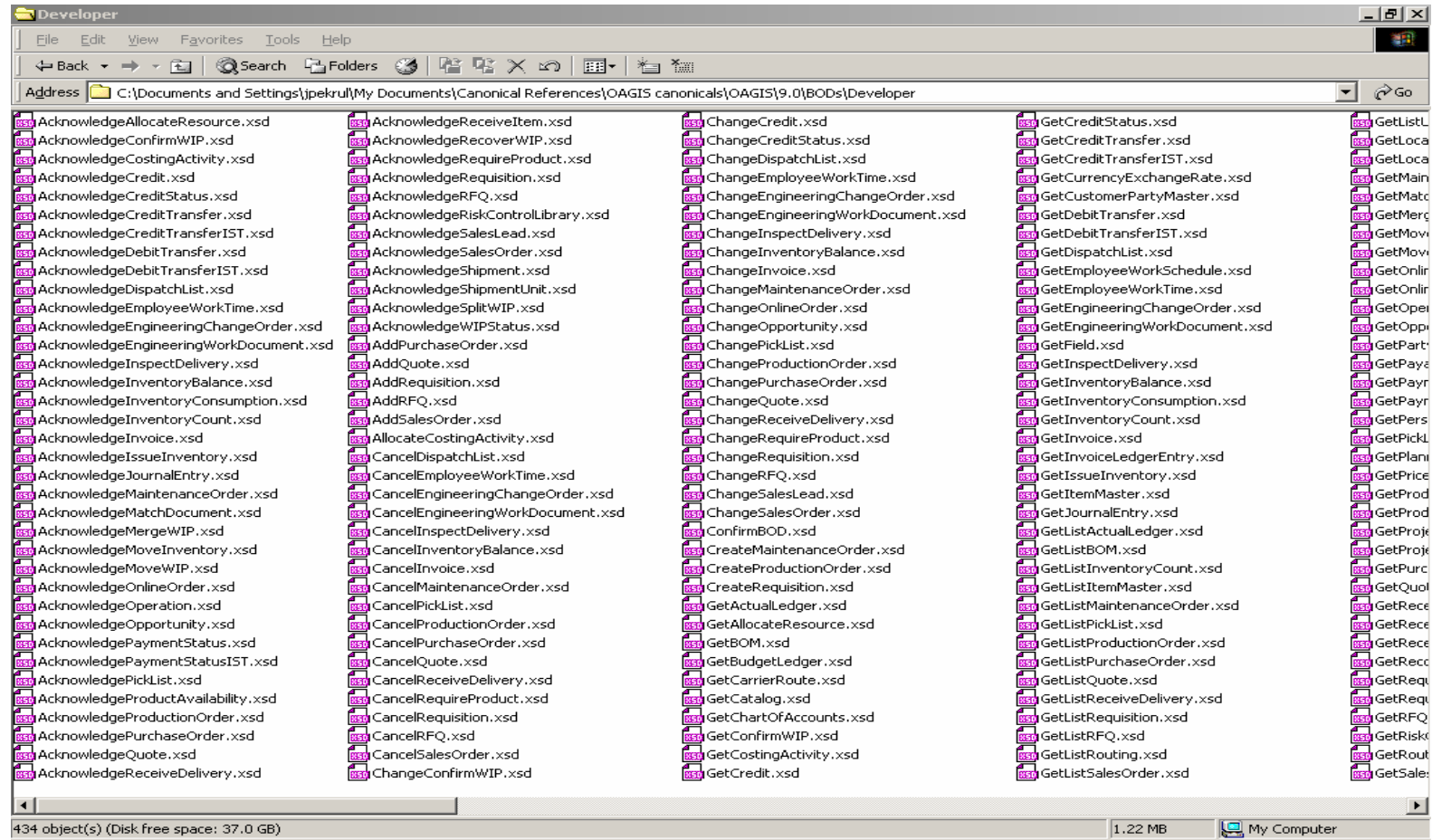
# Issues Resulting from Differences in Data Models

- Should local identifiers (Thought Leader ID, Contact ID) be published via the canonical? How should the canonical Party ID be populated?
- Our Master Data Management system cross-references instances of Customer/Contact, etc. across systems. Edge systems should exchange data with MDM and store the global ID for each instance of master data.
- When publishing master data via the ESB, edge systems must populate Party ID in the canonical with the MDM global ID if it is known. In some scenarios (e.g., new instance of Contact or Thought Leader) the global ID will not be known.
- The need to define rules for populating ID fields in various canonical entities has helped identify issues that are driving improvements in our overall data architecture.

## Choosing Standard Schemas vs. Custom Development

- Standard schemas (HR-XML, B2MML, etc.) are recommended for any message exchange outside the enterprise.
- Vendor-provided schemas may be required for interfaces with packages such as SAP and PeopleSoft.
- Try to leverage standard schemas such as OAGi in other situations whenever possible. These schemas are extremely detailed and may require paring-down to make them useful.
- We developed custom canonicals to support our Commercial business unit because our business definitions for 'Customer' and other concepts were too different from OAGi standards.

# A small sampling of OAGi XSD schemas



# Industry-Standard Canonicals: OAGi

**Free Downloads - Microsoft Internet Explorer provided by Genentech**

File Edit View Favorites Tools Help

Address <http://www.openapplications.org/downloads/oagidownloads.htm>

**OAGi**  
Open Applications Group

**Open Applications Group**  
Standards for Business Software Interoperability

Home How to Join Free Downloads OAGIS Support

**About OAGi**  
How to Join  
Free Downloads  
Projects  
Member Documents  
User Groups  
Members  
Collaborations  
Meetings  
Change Request  
Contact Us

**WS MEMBER**

**OASIS MEMBER**

*Our site is constantly being updated, check back with us for updates.*

**Welcome to the Open Applications Group Download Page**

From here you can download all of the OAGi white papers and specifications all for free.

**OAGIS® 9.0 Schemas Download**  
This includes the schemas, XML instances, and all base libraries.  
You may download either the zip version or the self-extracting version by [clicking here](#).

**Watch this space! OAGIS® UML data models coming soon.**

**OAGIS® 9.0 documentation Download**  
OAGIS® 9.0 has been built so that all of the schemas are annotated with the documentation inside.  
We have included downloads in two different formats for you. They include WinZip and also 7-Zip, an open source zip tool available at [www.7-zip.org/](http://www.7-zip.org/).  
Please [click here](#) to go to the download page.

**OAGIS® 9.0 Naming and Design Rules**  
The rules in the OAGIS Naming and Design Rules document are compatible with the UN/CEFACT Naming and Design Rules document. The members have extended the rules in some cases for OAGIS conventions but these rules do not conflict with the UN/CEFACT rules.  
Please [click here](#) to go to the download page.

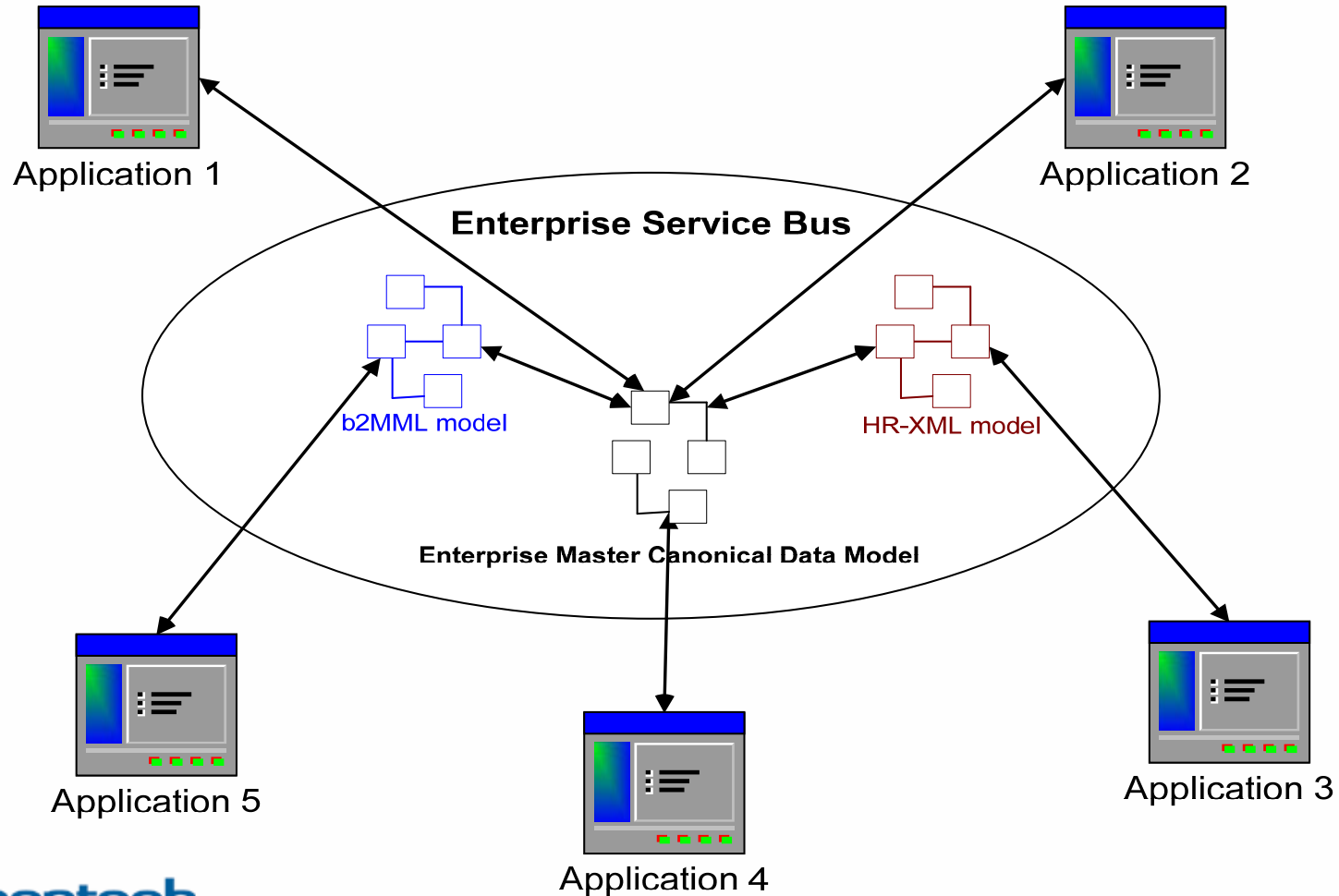
**Resources**  
[OAGIS Briefing Presentation](#)  
[Why Join OAGi?](#)  
[Who Uses OAGIS®?](#)  
[OAGIS Development Methodology](#)

**OAGIS 8.0 SP3**  
Documentation, schemas, and examples  
[WSDL files for OAGIS 8.0 SP3](#)

**OAGIS 7.2.1**  
[OAGIS Prior Releases](#)  
Includes 7.2.1 back to OAGIS 6  
[WSDL files for OAGIS 7.2.1](#)

Internet

# Using both Standard and Custom Canonicals



## Emphasize Data Quality over Flexibility in Canonicals

- Define the principles that guide your canonical development. Will you emphasize flexibility or data governance?
- Developers often prefer to maximize flexibility – loose typing, all fields defined as NULL, and no code set enumerations in the schema. This approach ensures quick turnaround on requirements to publish new data, and places no data quality constraints on publishers.
- Our canonicals use strong typing, field lengths approved by stewards, and standard code set enumerations. This places a greater burden on publishing systems. This may slow down some project delivery cycles.
- Use of data governance principles in the canonical is helping to surface data quality issues, the need for reference data management, and more comprehensive data stewardship.

## Combine Top-Down & Bottom-Up Approaches to Modeling

- A “top-down” approach to canonical modeling is similar to developing an Enterprise Logical Model.
- A “bottom-up” approach uses only project requirements for data integrations – the canonical will support only that which is required.
- There is literature on the internet supporting both approaches.
- We are using a hybrid of the two. A conceptual data model for our Commercial business unit was used as a starting point. Next, project requirements were incorporated. We believe this approach will make it simpler to incorporate future project requirements into the canonical.
- The Commercial Canonical model may evolve into a detailed logical model for a significant part of the enterprise. We could leverage this for multiple other purposes.

## Limitations of the ER-to-XML Approach

- ER/Studio's XSD generation feature is not currently robust. The manual edits we make on the XSD files may not scale well as the scope of this work grows.
- We developed scripts for certain edits such as stripping out enumeration values from the XSD, and for adding certain XML attributes that we should be defaulted when the XSD is generated.
- Manual edits to make the model a valid tree structure are less straightforward and have to be done manually. Project requirements determine which entity is the roots in a given context. Tool support similar to that which allows denormalization in the physical model only would be helpful.

## Additional Topics

- ESB processing is inefficient if XSD schemas are too large. What is the ideal level of granularity in these documents – individual entities or subject areas within the overall model?
- Multiple applications using a common canonical model also means that many applications are impacted if there is a new version. We are still working out the best solution to this.
- How best to integrate multiple canonicals or reconcile differences between them. For example, 'Employee' is modeled differently in the HR-XML standard schema, and our Commercial Canonical.

# Questions?

Thank you,

Mehmet Orun <mehmet@gene.com>

Jeff Pekrul <jpekrul@gene.com>