The Data Services Layer

Building a Solid Foundation for SOA

Jason Bloomberg & Ron Schmelzer
ZapThink LLC

SOA09
SOASUMMIT2009
MAY 2009 | SCOTTSDALE, AZ

software AG

Business Driver: Visibility

- The Challenge of Business Intelligence:
  - Visibility into data
  - Visibility into business processes
  - Visibility into levels of compliance

Does your executive management have the visibility they need today?

Visibility & Heterogeneity

- SOA useful in environments of heterogeneity
- Traditional Business Intelligence (BI) leverages homogeneous data and pre-packaged heterogeneous data
- SOA fills in the gaps: real-time, *ad hoc* data needs

software AG

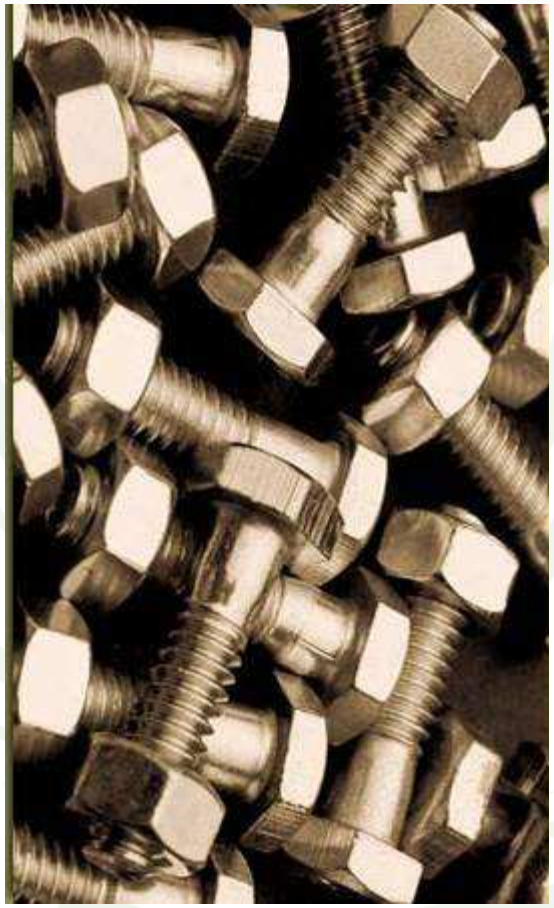
SOA, Integration & Legacy

- Key goal of SOA: Integration as a byproduct of Service composition
- Goal of legacy integration: building Services to support this goal, NOT connecting systems to address a particular business need

Move away from "connecting systems" and toward "composing Services"

Exposing Existing Capabilities

- Service interfaces exposed from existing systems
- Often pre-defined by existing software
- Low-level representations of internal application functions or interfaces often exposed as Web Services



Many Perspectives on SOA

Technology

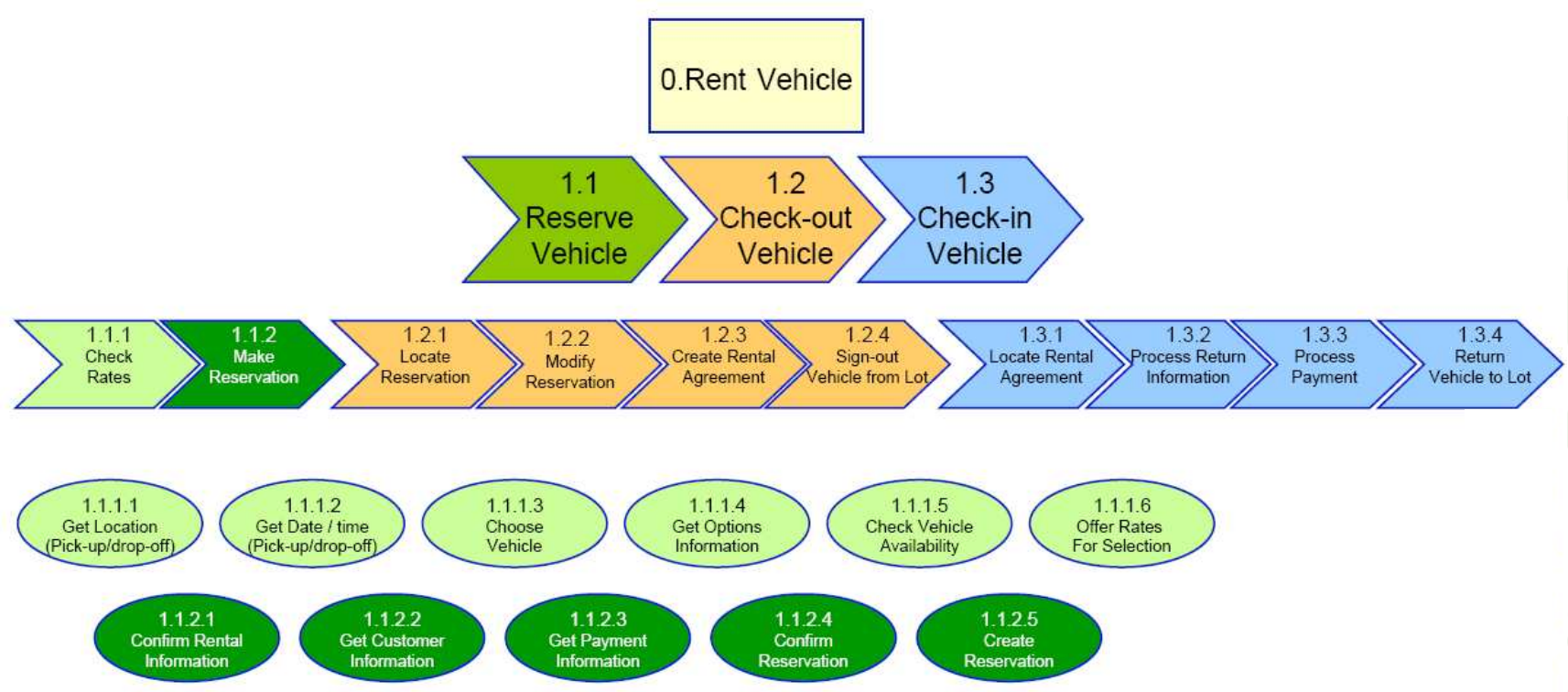
Business Processes

Data

Services

- All views relevant & important
- Service-Oriented Architects must have all perspectives

SOA: A Business Service View



Source: IBM

Defining Services

- Define Services via their behaviors
- How underlying information models are bound to those Services
- Typically, create data Services and/or transactional/application Services
- Services can composed into processes

Validating Service Assumptions

- Validate assumptions about Services, including:
  - Where their implementations are located
  - Their purpose
  - Information bound to the Service
  - Dependencies (e.g., if it's a composite Service)
  - Access mechanisms
  - Security issues

Key Service Abstraction Enabler: Proper Granularity

- Business-oriented requests and responses
- Blocks of information exchanged
- Driven by the business/data context of the Service

Granularity Example

- Fine-grained interaction:

» I'd like to order a shirt.

» What size would you like?

» Size XL.

» What style?

» The one on page 30 of your catalog.

» Which catalog is that?

» …

- Coarse-grained interaction:

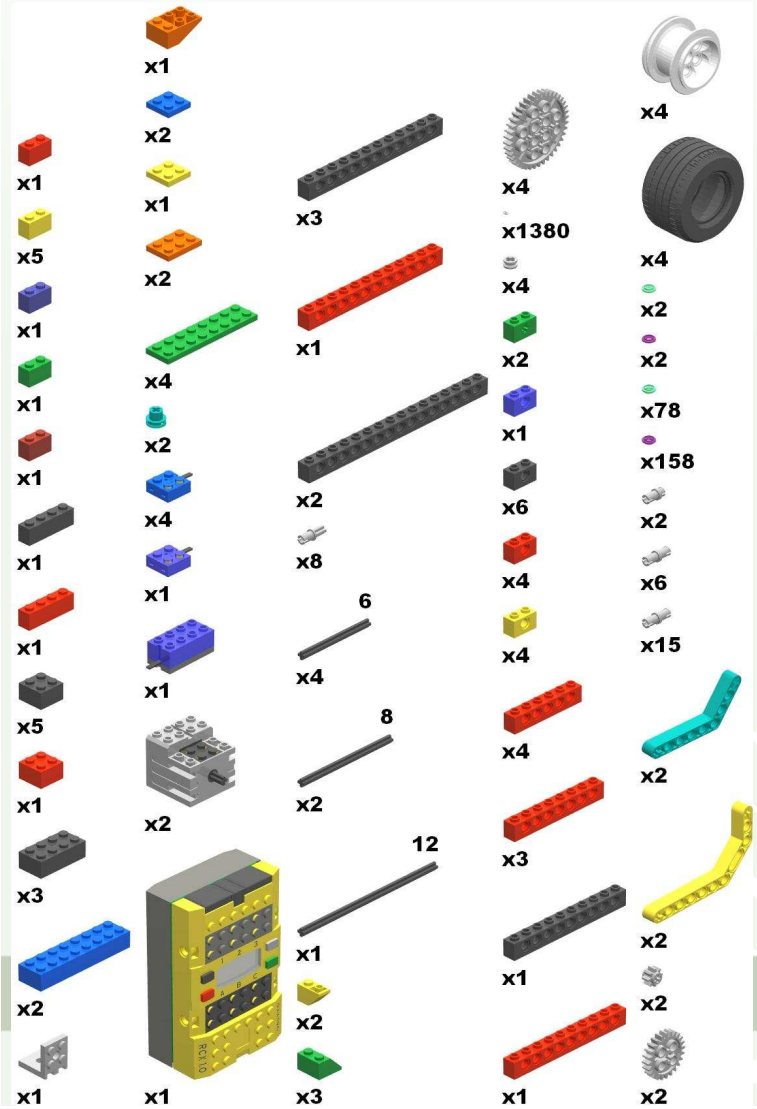
» Fill in the order form online with all required information

» Click submit

» Receive "Thank you, your order is complete" page

» …

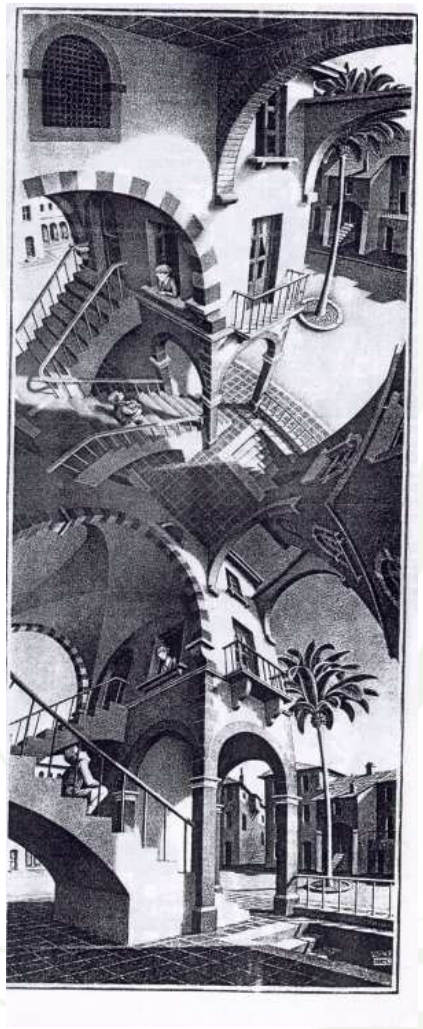
Achieving Proper Granularity



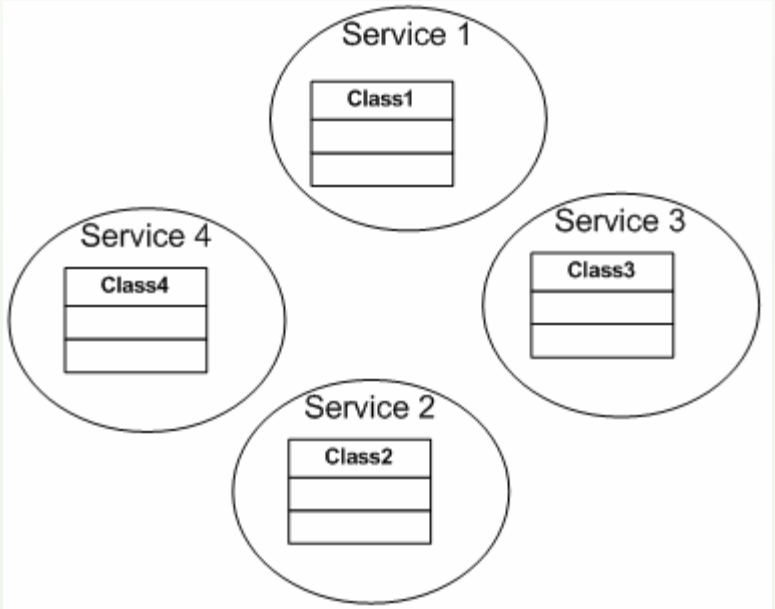
- Too small?
  - Difficult to build anything
- To big?
  - Too inflexible, not reusable
- Just right?
  - Depends on the project at hand
  - Most situations require a mix of sizes

Zeroing in on Proper Granularity

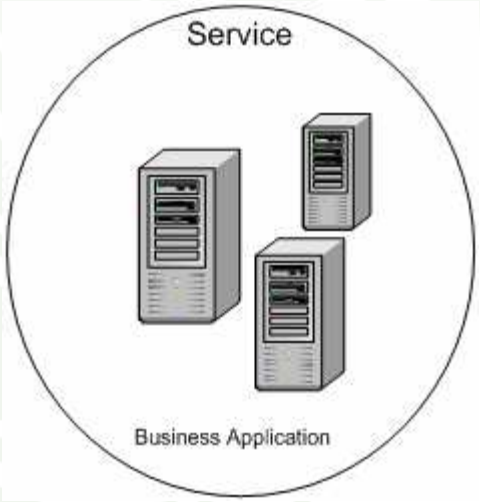
- Top down: process decomposition leading to suggested Business Services
- Bottom up: exposing existing capabilities to build Service interfaces
- Middle out: business requirements for Service responses drive granularity



Example: Too Fine Grained vs. Too Coarse Grained



Each object exposed as a Service

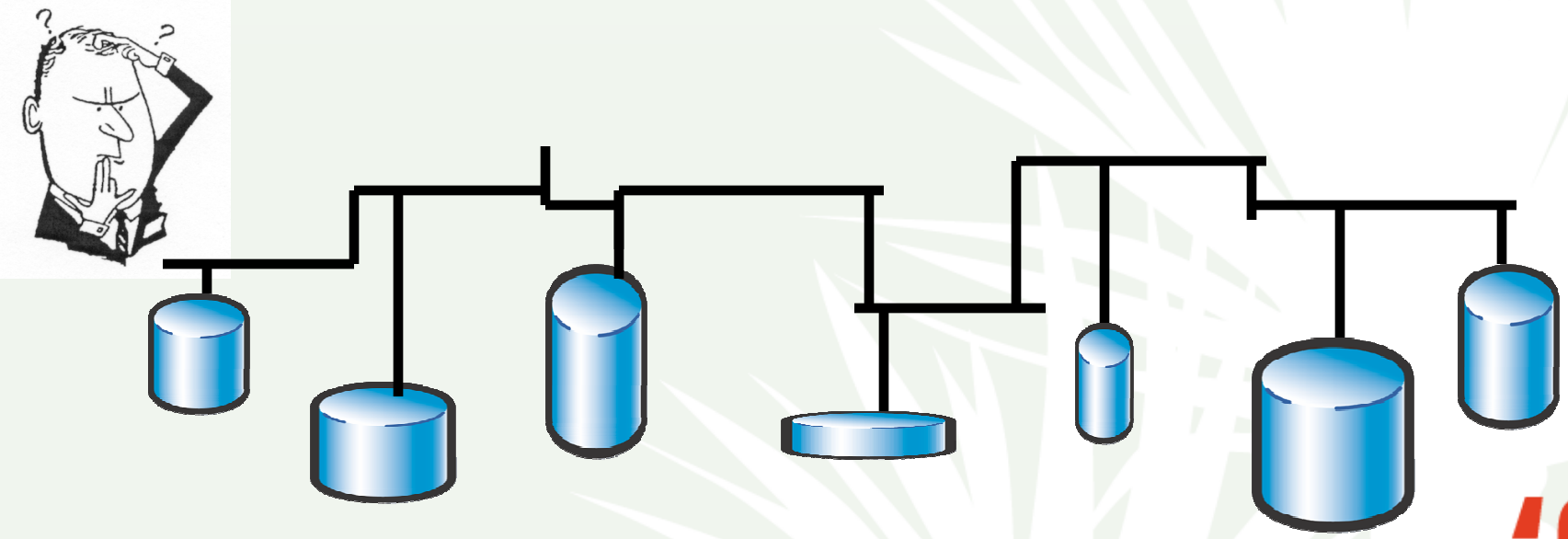


Entire application exposed as a Service

Source: Wipro

SOA & “Legacy” Heterogeneous Data & Data Stores

- Heterogeneity hides data and their meaning
- Data cleanliness, consistency, availability issues
- Data-related issues cascade thru the entire distributed computing environment

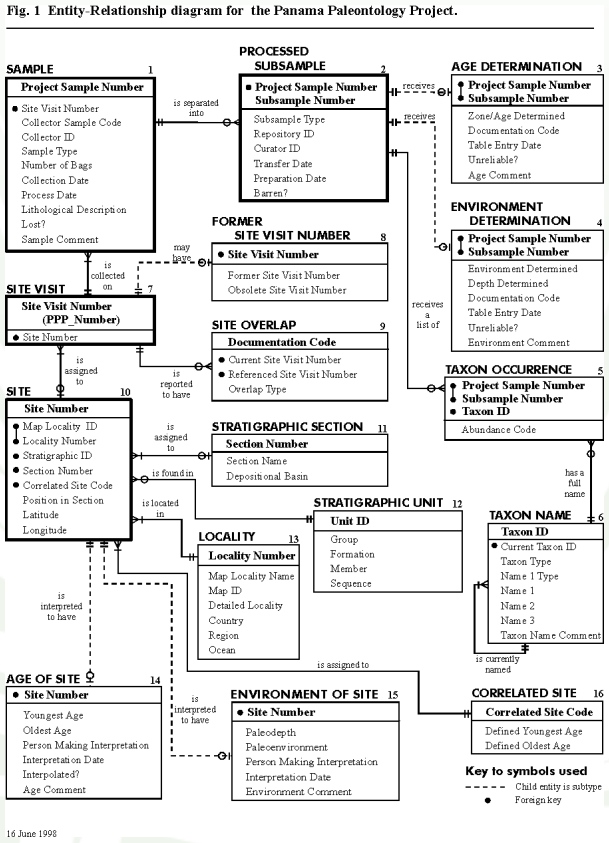


The Data Services Layer

- Provides a separation of concerns between the persistence tier and the Business Services/infrastructure Services
- Requires contracted Service interfaces
- Services can abstract single queries, joins, etc., depending on need

Designing Data Services

- Examine existing physical databases at the technology and schema levels
- Determine the meaning of the data & application semantics
- Identify how the data are bound into the existing Services and processes



Understanding Data Services

- Data Through Services
  - Service-enabling Data Integration
  - Providing a means to access structured and unstructured data in a heterogeneous fashion
  - Data integration *beneath* the Service abstraction
- Data-as-a-Service
  - Aggregating and Composing Data
  - Treating data as Services through composition
  - Data integration *above* the service abstraction
- Data Services
  - Services that perform transformations and other data operations

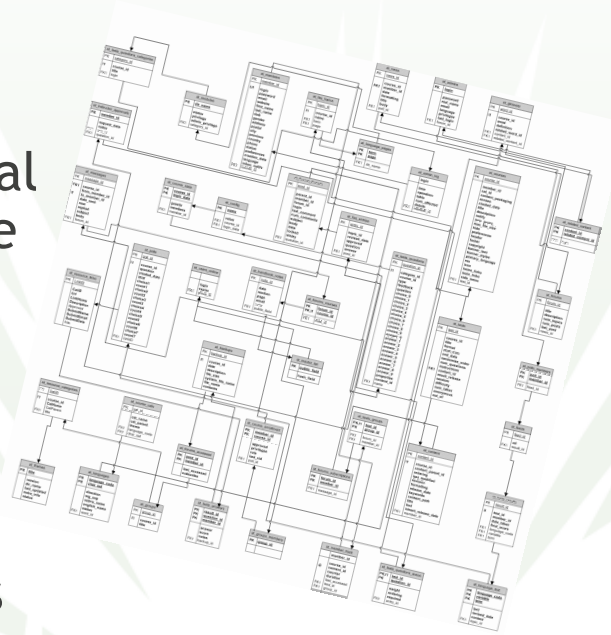
Data Integration & the Data Services Layer

- Data Integration becomes part of underlying IT infrastructure that supports Services
- Decisions about data caching or data virtualization (EII) part of the SOA implementation puzzle
- SOA can lead to less data replication

Supporting Data Services with Data Integration

- Data Integration Through Extract-Transform-Load (ETL)
  - Brittle and inflexible
  - Typically point-to-point and tightly coupled
  - As a result, difficult to abstract
- Data Integration through Enterprise Information Integration (EII)
  - Some "EII" is adapter-based, point-to-point – little SOA advantage
  - Better EII provides aggregated, read-only views of heterogeneous data – some SOA advantage
  - Best EII is truly bidirectional, offering full data virtualization – key to effective Data Services

Leveraging Existing Data



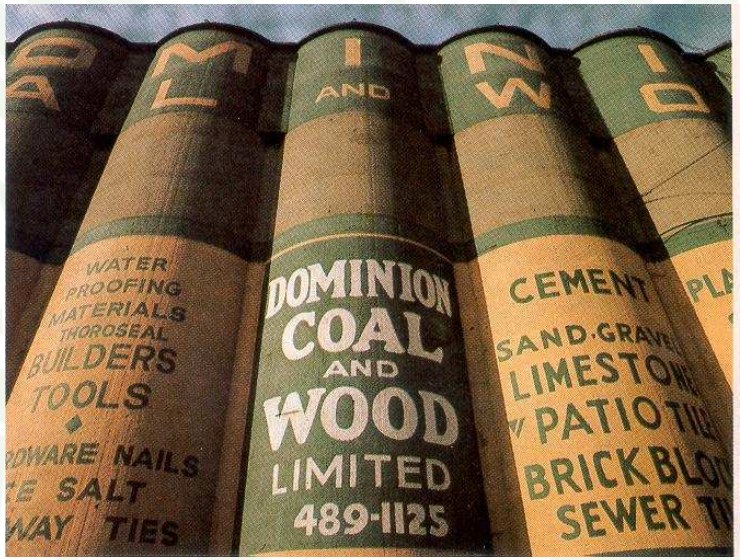
- Reverse-engineering existing physical and logical database schemas can help identify appropriate data
- The schema and database model may give insight into the structure of databases
- But – cannot determine how that information is used within the context of the application or Service

SOA & MDM

- *Master Data Management* (MDM) is a set of processes & tools for consistently defining and managing an organization's information
- MDM without SOA
  - Tight coupling leads to inflexibility, stalling MDM initiatives
- SOA without MDM
  - Semantic and data consistency issues may derail SOA initiative as it moves cross-departmental

Leveraging Data Services Layer

- Data Services Layer provides loose coupling between the Services & underlying databases & information stores
- Provides a point of configuration for:
  - Differences between the data as physically represented
  - Preferred SOA logical representation



The Data Services Layer Abstraction

- Data Services Layer can easily be reconfigured
  - Account for changes to data and Service requirements
  - Requires minimal work
- Example: physical databases in many instances need not be changed
- Simply change the abstraction via reconfiguration

software AG

What's Missing?

- SOA addresses *application* integration issues
- What about semantic integration?

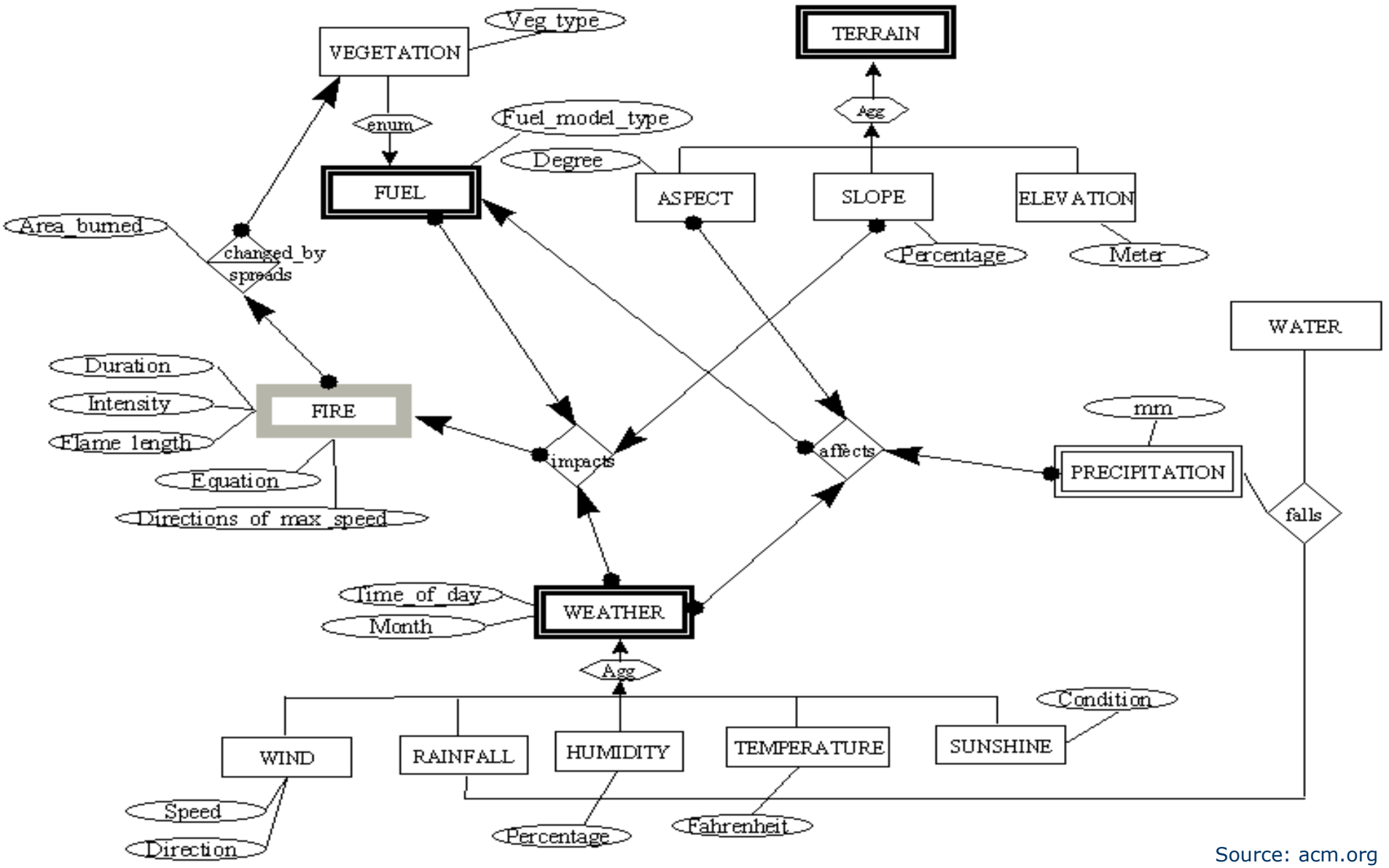
Semantic Level Understanding

- You can't deal with information you don't understand, including information bound to Services
- Important to identify all application semantics that exist in your domain
- Allows you to properly deal with those semantics as metadata

Role of Application Semantics

- Establishes the form of properties of the business process the particular application refers to
  - Example: one customer number for one application may have a different value and meaning in another application
- Eliminates contradictory information when the application is integrated/composed with other applications
- Achieving consistent application semantics requires an information integration “Rosetta Stone”
  - One of the major challenges to implementing SOA

Sample Semantic Model



Source: acm.org

software AG

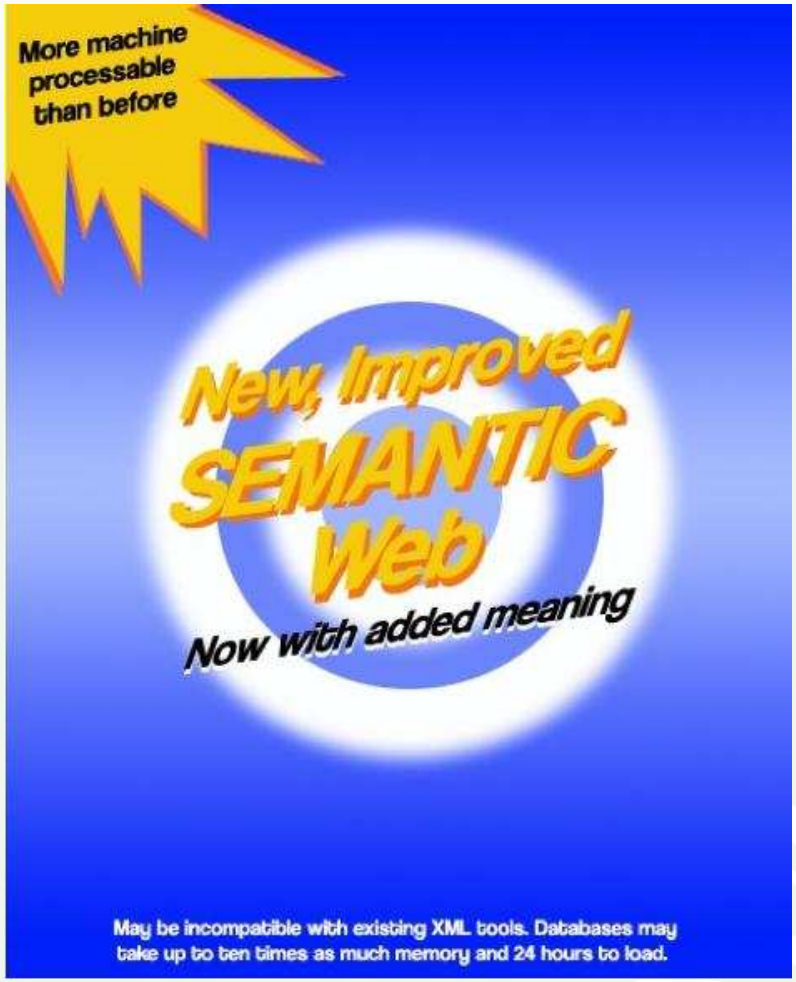
Semantics Challenges

- Existing systems are older, proprietary, or both
- First, create a list of candidate systems
- Determine which data repositories exist in support of those candidate systems

Semantics: The Greatest Integration Challenge of SOA

- Just because you can *call* someone, doesn't mean you can speak their language
- Adds layers of complexity to the transformation problem

Resolving Semantic Issues



- Active area of study and development: W3C Semantic Web, Web Ontology Language (OWL), etc.
- Some products on the market support semantic integration
- Largely a manual process: parties must agree on the meaning of terms in a human-to-human conversation

ZapThink is an industry advisory & analysis firm focused exclusively on SOA, EA, and Enterprise 2.0.

Register for an upcoming *Licensed ZapThink Architect* course and obtain your LZA Credential!

Thank You!

Jason Bloomberg

jbloomberg@zapthink.com

Ronald Schmelzer

rschmelzer@zapthink.com