Information Architecture and Categorization
August 2008
Ian Koenig
What is Information Architecture?


• “Just as with infrastructure in the larger society, the benefits of an infrastructure for an organization, as well as its healthy evolution, generally depend on content, mechanisms for agreeing on standards, and a range of common services for enriching connectivity. [Lawrence Livermore Labs IA Program http://www.llnl.gov/projects/ia/library/ia-report/intro.html]

• “Enterprise information architectures provide a framework for reducing complexity and enabling enterprise information sharing…” [Cook, Melissa. Building Enterprise Information Architectures. 1996]

• “Enterprise Information Architecture extends beyond organizational boundaries to external sources and targets to enable rapid business decision making and information sharing. EIA also includes:
  1. A catalog of authentic information sources (e.g. company databases, commercial databases, …);
  2. (2) classes of relevant business information and their value to the enterprise;
  3. (3) Information governance processes
Information Architecture

**Connecting the Dots:** Information architecture is composed of four conceptual elements and technologies. At the conceptual level we have the Entity Model, Metadata Model, Interface (API) Standards and Categorization System (Taxonomy).
Information Architecture – Supporting Technologies

Technology supporting the Information architecture:
- Auto categorization + Entity Extraction – Algorithmically tag unstructured data|
- Entity & Dataset Authorities – Master Databases for Entities and other content sets
- Search & Navigation – To find content based on categories and keywords or related content
- Interface Bus(s) – To facilitate the distribution of content

Auto-cat + Entity Extraction Technology

Categorization

Search

Metadata Model (Data Elements / Facts)

The Metaverse

Entity Model

Metadata

SOA

XML

Interface Standards

Search & Navigation

Interface Bus (s)
Information Architecture – Why do we need one?

Because:

• Most companies accrete content silos organically, either by acquisition or by the individual practices of individual businesses. Even if they did not, the concept of a Service Oriented architecture implies that as you carve the overall functional architecture into independent services, you need an information architecture to put the content based services back together in a consistent fashion.

• A content silo is a content set that is tightly coupled between the content master, who collects and stores the content and the application processes that present the content for analysis / display.
The Content Distribution Pattern

- There is a common architecture pattern for creating / managing and ultimately presenting content.
- The Pattern asserts the precise roles of systems and boundary interfaces.
- The pattern for Content distribution spans the boundary between Services (in an SOA) and Applications (using those services – which begs the question of what we mean by an SOA and why we draw the boundary where we do -- but that is a topic for a different discussion.
- This all might seem fairly arcane, but...
... if you Squint just a little tiny bit ...

... if you squint just a little tiny bit, you might recognize it.
There are two types of Content Masters: Entity Authorities and Data-set authorities.

Entities are those Content sets (e.g. Companies, People, Geographies, Instruments, et al) that link the other Data-sets together (e.g. in the Financial world: Company Fundamentals, M&A, Company Ownership, et al)

Entities are joined via Relationships. An RDF-style triplet Resource Description Framework approach is used to do this (http://en.wikipedia.org/wiki/Resource_Description_Framework)

It is considered Best Common Practice to assign all Entities Permanent GUIDs to uniquely represent them and to use these permanent GUIDs as early in the API call tree as possible in deference to mnemonic aliases.

For example “C” is not the symbol for “Citigroup Inc.” It is the symbol for the NYSE quote for Citigroup Common Stock, but only since Chrysler Corp was acquired by Daimler and gave up the symbol “C”, which it had for its common stock. So not only does the symbol not actually represent the company consistently in time, many companies issue ADRs and list on multiple exchanges making the use of quote symbols as company identifiers highly problematic.
Content Services distribute Content across their Data Interface in their Canonical data Model.

The Services Canonical data model is the data Model optimized for distribution and specific to no one consumer.

When an application consumes content from a service, its first order of business is to transform that content into something quickly renderable / presentable through its Service Interface.

It is not allowed for the Pre-render step to create new content (otherwise the new content is not really mastered), except under rigorously controlled circumstances.

For example, it is acceptable for the pre-render step to calculate data “on-the-fly”, as long as all of the values needed for the calculation are mastered and the formula upon which the calculation is based is mastered.
1. **Human Interfaces:** Presentation servers access content from the pre-Render layer through a well defined Service Interface. They never access Content Masters and Data interfaces directly.

2. **Application Pre-render Databases:** access data from Content Masters through a well defined Data Interface (API). Pre-Render databases / caches are Copies. They do not create “new” data except they may perform ‘on the fly’ calculations as long as the result is not persisted (ex. Currency conversion)

3. **Content Masters:** contain the “single version of the truth” for all data, created, acquired or derived.

4. **Scalability:** Content Masters tend to scale proportional to the amount of content. Application pre-render databases / caches tend to scale proportional to both amount of content and usage. Content Masters should be architected to scale vertically and Application pre-render databases should be architected to scale horizontally

5. **Physicalization:** Content Masters and Application pre-render databases / caches belong in a core network in the Data center (i.e. not a DMZ). Presentation / Human Interface databases / caches belong in the DMZ. In a three network zoning model (i.e. where there is an application zone between the DMZ and the Core network zone, the application pre-render database / cache belongs in the application zone (duh!))

6. **D/R:** Content Masters should have synchronized copies in at least two Data Centers, preferably in the same geo-region. Masters should be backed up and restored from tape in a disaster. Application pre-render databases should be rebuilt from peers and synchronized to the master. They may not need to be backed up to tape at all and hopefully never need to be restored from tape. The same goes for presentation caches. Don’t confuse D/R and HA.

7. **Data Elements (Facts):** (i.e. columns) The Content Master is responsible for creating a unique Data Element Identifier for every piece of data it owns.
8. **Data Items:** (i.e. rows of data) Only the Content Master is permitted to create an item of that data class. The full list of Content Items (i.e. Entities and Datasets) is defined by the Information Architecture. The Content Master is responsible for allocating a permanent GUID for every data item it creates
   a. Once a permanent GUID is allocated it may be sunset (if the data item it is associated with has its “Effective To” date set). It may never be reused or take on a new meaning.
   b. Only the Permanent GUIDs of Entities should be used to link content between datasets

9. **Copying Data:** When content is copied between databases, the following contract applies:
   a. It is the responsibility of the copy to ensure it stays synchronized with the source. If the source does not support a “push” style interface, then it is desirable that the copy is periodically “dumped” and rebuilt.
   b. The copy is responsible for preserving the “name” of the Data Elements and the FactIds (for Traceability). Copies may not rename data
   c. When a master copies data from another master (e.g. for the purpose of deriving new data), the copy may not be on-passed from that DB; only the derived data can.
   d. The exception to the on-pass rule is for “symbology and classification data”

10. **Value Add Content:** is derived and is distinct from vs. “As Collected” content
    a. When new content is created (derived), this “value added” content must be mastered. This could either be the value data itself or the business logic which drives the calculations in the Application pre-render database.
    b. It is best practice to master value-add data with the data class that it is most closely aligned (rather than create a new class and a new master)
    c. The process of creating value added data must not “lock out” the process for adding new data or modifying data.
SEC will Allow Companies to use the Internet to Improve Investor-Management Communications

From CFO.com – August 16, 2007
According to SEC chairman Christopher Cox, the commission will allow companies to use the Internet to improve investor-management communications. As currently proposed by the commission, a company interested in offering this venue to shareholders would alert them via...
A Financial Entity Model – to link Content sets together

- Geographies
  - Regions, Countries, Physical Features

- Sectors + Industries
  - Sector Hierarchy (Multiple Schemes)

- Organization
  - Govs + Agencies
  - NGO Charity et al
  - Corps
  - Mutual

- Person
  - (Multiple Roles)

- Markets
  - Groupings of Market Participants

- Account Portfolio Holding

- Event
  - Regional Events
  - Economic Events

- Industry Events
- Market Events
- Gov't Events
- Corp Events
- Corp Actions
- Analyst Events
- Account Events

- Physical Asset
  - Currency, Commod, Energy, Real Estate

- Instrument
  - Equity, Bond, Future, Spot, Derivative, et al

- Index
  - Equity Index
  - Benchmark Bond

- Indicator
  - Economics, Market Stats

- Trade

- Quote
  - Level 1, Level 2, IOI, Advertisement,
Aboutness = Entities + Subjects

Defining what Financial content is About is distinct from Sentiment (Goodness / badness), Genre (Type), Source (Author / Publisher) and Targeting (who it is of interest to). Entities and Subjects define aboutness.
Categorization – Other than Aboutness

• When we categorize Content, there are domain other than aboutness, that are important and relevant.

• Genre: Literally means “Kind or sort“. Defined by IPTC NewsML as: “Nature… or characteristics of the item, not specifically its content”. For example, categories defining the type of document, or the primary focus, or its urgency, or frequency or how it was created or how it is formatted are all examples of Genre categorization

• Sentiment: In relation to financial content refers to positive/negative/neutral “feeling”. For example, sentiment categories could include Buy/Sell/Hold recommendations for Instruments, Company credit ratings, Government credit ratings, Market bullishness/ bearishness, consumer optimism/pessimism, etc.

• Source: There are multiple ways to identify source and actually multiple sources that are important. For example, the human Author, the Company owning the copyright, the feed/wire by which it was distributed, the product subset it was distributed on are all examples of Source identification.

• Targeting: There are likewise multiple ways content may be targeted to specific users/user groups. Content may be “Of Interest To” a User group based on the demographics of that group rather than the aboutness of the content. Content may be targeted by inclusion/organization into a product or distribution channel (e.g. an RSS feed).
Categorization – Canonical and Presentation Forms

Content is categorized to a high degree of specificity / granularity to allow for precise navigation and do adapt to the needs of multiple products and multiple user groups. This can easily lead to a canonical (or tagging) taxonomy of 1000’s or 10’s of 1000’s of terms.

<table>
<thead>
<tr>
<th>Entity Level</th>
<th>Canonical</th>
<th>Presentation 1</th>
<th>Presentation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate</td>
<td>Geography 3353 Categories</td>
<td>Geography 3 Regions + 5 Countries</td>
<td>Geography 5 Regions + 177 Countries</td>
</tr>
<tr>
<td></td>
<td>Industry 2482 Categories</td>
<td>ICB Industry 142 Categories</td>
<td>GICS Industry 177 Categories</td>
</tr>
<tr>
<td></td>
<td>Market 354 Categories</td>
<td>Market</td>
<td>Market</td>
</tr>
<tr>
<td>Body</td>
<td>Instrument Security, Future, Derivative, et al</td>
<td>Gov’t Agencies 296 Categories</td>
<td>Gov’t Agencies 100 Categories</td>
</tr>
<tr>
<td>Asset</td>
<td>Indicator Economics, Market Stats</td>
<td>Instrument</td>
<td>Instrument</td>
</tr>
<tr>
<td>Event</td>
<td>Event Corp Action, et al</td>
<td>Corp Actions &amp; Events</td>
<td>Event</td>
</tr>
</tbody>
</table>

But the average human can deal with between 30 and 300 codes at most. After that they resort to “keyword hunting”, letting the search engine do the work, obviating the need for a taxonomy.

To avoid this we define coarse grained Presentation taxonomies targeted to specific user types that are mapped to the fined grained canonical taxonomy. Good search engine statistic on what users search for can really help “tune” the presentation taxonomies and mappings.

Copyright © Ian Koenig
Identifiers and Relationships

**Search and Navigation:** Identifiers and Relationships are core concepts to “Lookup” and navigation. When you do a “Symbol lookup”, to get a quote you are using an Identifier. When you do a symbol lookup to get Company Fundamentals, you are using an Identifier and multiple relationships.

**Identifiers**
- Identifiers are mnemonic aliases for Entities.
- They may be defined within 3rd party schemes (e.g. ISIN) or Proprietary schemes (e.g. RIC)

**Identifiers**
- GUID
- Symbol
- Scheme
- Effective From
- Effective To

**Relationships**
- GUID From
- GUID To
- Relationship Type
- Effective From
- Effective To

**Organizations**
- Gov't, Agency, Company, NGO
- Market Participant

**Instruments**

**Persons**
- (Multiple Roles)

**Quotes**
- Level1, Level2

**Exchange Sym**
- RIC (mostly)

**Identifiers**
- ISIN
- SEDOL
- CUSIP
- VALOREN
Entity: Pharmaceuticals - An Industry Entity

Schering is the leading global seller of oral contraceptives, including Yasmin. It also sells multiple-sclerosis drug Betaferon. Merck's best-known drug is Erbitux, for colorectal cancer.

Schering announced a 20% dividend increase on Feb. 20 and has seen its share value rise by more than a third since mid-October.

Schering's fiscal 2005 sales topped 5 billion euro, with net profit rising 23%, helped by job cuts.

Related Research Notes
- SF Monday Rx Quarterback
- MS Pharmaceuticals, Major: Weekly Rx Commentary (Part 3 of 3)
- MS Pharmaceuticals, Major: Weekly Rx Commentary (Part 2 of 3)

Related Companies
- Company: Merck
- Price: 49.49
- Change: -0.33
- Company: Johnson & Johnson
- Price: 61.84
- Change: -0.24
- Company: Schering Plough Corp.
- Price: 29.01
- Change: +0.18
Staffing company shares mixed after jobs report

The Labor Department's report, he said, "is likely to fell one-tenth of a percentage point to 5.4 percent, the lowest rate since October 2001, primarily because 152,000 adults dropped out of the labor force." In the past two months, the headline payroll number meet expectations after two months of disappointments," said Tobey Sommer, an analyst.

The report, he said, "is likely to improve investor sentiment on employment-related stocks."
The Content Distribution Pattern – Data Interface

Content Source (s) (Content Creation + 3rd Party)

Ingest Interface (Publishing + Feeds)

Service Interface

Application

Data Interface (Content Distribution)

Content Pre-Render

Content Services Distribute Canonical Data Model (in XML)

The Content Master

Metadata

Presentation / Human Interface

Canonical Data Model (in XML)
The Data Interface

- In the Content Distribution pattern, the Data Interface is defined to provide the loosely-coupled interface contract across the boundary between the content master and the application pre-render database. It effectively governs the Service / Application boundary and the Producer / Consumer boundary (i.e. it's pretty important)

- A bus-style implementation of the data interface providing pub/sub capability is considered best common practice. The subscription protocol could either be topic based (like JMS) or content based (like provided by emerging content routers (e.g. http://www.solacesystems.com/)

- The Data Interface is:
  - One way
  - Encoded in XML in the Canonical Data Model of each Content Master
  - Keyed by non-Fragile Unique Permanent Entity GUIDs
  - Loosely Coupled: Sources should not ‘know’ targets. Sources publish. Targets subscribe. The Data Interface Bus mediates. Interface contracts are enforced.
### Initialization and Synchronization

In order for a Content master (service / Provider) and an Application / consumer to work across the interface boundary, the master must provide the ability to (1) Initialize the downstream database / cache and (2) Keep it synchronized as changes occur (adds, modifications, deletes).

#### 1. Initialize

<table>
<thead>
<tr>
<th>1a. Full Rebuild</th>
<th>An application pre-render database / cache upon initialization or after recovery needs a mechanism to fully rebuild its state from the beginning of time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b. Cumulative Delta</td>
<td>A cumulative delta includes all changes from the previous full rebuild, so to initialize, an application generally processes the most recent Full rebuild AND the most recent Cumulative Delta.</td>
</tr>
</tbody>
</table>

#### 2. Synchronize

<table>
<thead>
<tr>
<th>2a. Incremental Delta</th>
<th>An incremental delta includes all changes from the previous increment. Once an Application database is initialized, it hopefully never needs to initialize again.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2b. Event based (transactional) Messages</td>
<td>By far the most efficient way for an application to stay synchronized with a master is through transactional events. But outside of market data and news, this is (today) fairly uncommon in the content world. That being said, there are numerous ways to use the lessons of transactional messaging in the database world.</td>
</tr>
</tbody>
</table>
<?xml version="1.0" encoding="UTF-8"?>
<Content majorVersion = "1" minorVersion = "0.7"
  xmlns="" 
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:item="http://contentSet.schemas.yourCompany.com/2008-07-20/">
   <Header> urn:uuid:XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX </Header>
   <Body majorVersion = "1" minorVersion = "2.0">
      <ContentElement action="Insert">
         <NewsML xmlns="http://iptc.org/std/NewsML/2003-10-10/">
            <!-- Contents here -->
         </NewsML>
      </ContentElement>
   </Body>
</Content>
Geographies

- Created from Authority sources and then Extended as necessary based on de-facto Geographies already in-use (e.g. Deals Regions)
- Authorities: ISO 3166, Multiple Supranational organizations (e.g. the IMF, U.N, et al), U.S. census bureau
Industries

- Authority sources include: NAICS, SIC, GICS, ICB, RBSS,
- Relationships preserve the hierarchy. Identifiers map the scheme provided symbol to the Term. The scheme can be precisely recreated with no loss of fidelity.
- The “IsLike” relationship is used to map Terms in the 3rd party scheme to “equivalent “ canonical Industries.

Example Relationships:

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Relationship</th>
<th>Name From</th>
<th>Name To</th>
</tr>
</thead>
<tbody>
<tr>
<td>GICS10</td>
<td>ROOT</td>
<td>GICSChildOf</td>
<td>Energy</td>
<td>Industry</td>
</tr>
<tr>
<td>GICS1010</td>
<td>GICS10</td>
<td>GICSChildOf</td>
<td>Energy</td>
<td>Energy</td>
</tr>
<tr>
<td>GICS101020</td>
<td>GICS1010</td>
<td>GICSChildOf</td>
<td>Oil, Gas &amp; Consumable Fuels</td>
<td>Energy</td>
</tr>
<tr>
<td>GICS101020</td>
<td>GICS101020</td>
<td>GICSChildOf</td>
<td>Integrated Oil &amp; Gas</td>
<td>Oil, Gas &amp; Consumable Fuels</td>
</tr>
</tbody>
</table>

Example Identifiers:

<table>
<thead>
<tr>
<th>GUID</th>
<th>Name</th>
<th>Type</th>
<th>Identifier</th>
<th>Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>GICS10</td>
<td>Energy</td>
<td>Industry</td>
<td>10</td>
<td>GICS</td>
</tr>
<tr>
<td>GICS1010</td>
<td>Energy</td>
<td>Industry</td>
<td>1010</td>
<td>GICS</td>
</tr>
<tr>
<td>GICS101020</td>
<td>Oil, Gas &amp; Consumable Fuels</td>
<td>Industry</td>
<td>101020</td>
<td>GICS</td>
</tr>
<tr>
<td>GICS10102010</td>
<td>Integrated Oil &amp; Gas</td>
<td>Industry</td>
<td>10102010</td>
<td>GICS</td>
</tr>
</tbody>
</table>