ORG.

Aaron Brick, 2004
School of Information Management and Systems; University of California, Berkeley

0 Foreward

Natural language is divergent across time and users, but is our primary tool for description and precision. Traditional means to combat divergence in data sets, notably via vocabulary control and human oversight, can be both expensive and frustrating. I argue that a simpler, more expandable data structure such as Org will be able to both represent real information accurately and provide infrastructure for its own operations.

Uniform physical media can only be put in any one order; assets in a hierarchical directory structure embed much more information in their layout.

Org's vocabulary is uncontrolled and actively provides suggestions to reduce divergence. Given knowledge about the language in which the annotations are being made[1], users can be aided in encountering the most consistent annotation terms. These hand-holding mechanisms do not provide absolute convergence, but they should prevent divergence from growing at its natural rate. A central technique in this pursuit is the automatic recognition and deployment of new terms, in order to avoid the slower approval process of controlled vocabularies as described by Taylor[2].

The practice of rich, or typed, linking has been proposed on a larger scale by Berners-Lee. His Semantic Web[3] is composed of mutually referring declarations, which when formalized support high-level automatic "reasoning."

Org is a framework for storing axiomatic assertions, which as a whole constitute an ontology or a semantic network. Unlike the vastly more complicated MPEG7, all attributes are represented as relationships.

This project is a generalization of work[4] done at SIMS in 2004 by Brick, Kashnow, and Lawrence, advised by Marc Davis, which was also called "Org" but treated specifically the archival and distribution of recorded live music. Since I believed that our innovation was centered on data structures rather than interfaces, I refocused on information theory and away from media-specific applications.

1 Logical derivation

In this document I use set, graph, and information theory to describe the Org data structure and its functionality. I hope to reveal that liberal yet careful abstraction, with judicious leveraging of existing records, can enable highly precise description of real things. The project is predicated on the following axioms.

I. A body of annotations should be stored in a single, portable, open-standard file.
II. Records, relationships, and attributes all have attributes of their own.
III. Harvesting existing databases is easier than building them from scratch.
IV. Natural language provides the most accessible vocabulary for annotation.
V. All attributes can be expressed as relationships to concepts.
VI. Attributes should be expressed in parent records where possible.
VII. Predicates with embedded reference to components should force the instantiation of the subconcept as a record with its own attributes.
VIII. Quantify all attributes (default=0.9?) to increase querying precision with natural ranking terms

1 The contents of WordNet® are one possible source of this information.
2 Taylor[99], pp. 151
3 Berners-Lee[98]
4 M.I.M.S. Final Project "Org" was submitted by Jeremy Kashnow and Maria Lawrence in May 2004.

(e.g., “very”, “somewhat”).

And I hypothesize that:

Cross-reference techniques can reduce divergence.

1.1 Basic structure

Faceting establishes that an entity, e – any concept we want to describe by annotation – can be described with an unordered set of attributes a. Each of these entities is a graph vertex; its edges will correspond to its attributes.

$$e = \{a1, a2, ..., ai\} \mid i > 0$$

An attribute, or any relational information, can be well-described as a dyadic assertion containing relationship (verb) and referent (object) entities[5]. This datum is equivalent to an RDF tuple or a database join table with two foreign keys and one parameter, the relationship type. In the graph, relationships are directional cocolored edges between attribute holder and referent entities.

$$a = \{e\text{relationship}, e\text{referent}\}$$

Attributes themseves can have attributes, to enable metaännotation; this is useful for attribution and conditionality. For this point the strict graph theory analogy cannot apply, because one cannot join edges without adding vertices.

$$\exists\ a1 \subseteq \{\ a2\ \}$$

Reinforced by Hofstadter's exposition of strange loops in language[6], these mutually-referential concepts are fairly natural to implement in usage. This level of generality allows very diverse unforeseen cases to be represented; as in RDF, there are no top-level rules about assertions' validities.

1.2 Set size boundaries

Because every entity has at least one attribute, the set of attribute instances, A, can grow more quickly than the set of entities, E.

$$\mid \forall\ a = A \mid \geq \mid \forall\ e = E \mid$$

In practice we use a much greater variety of references than relationships (I estimate the ratio at 2 orders of magnitude).

$$\mid E\text{relationship} \mid << \mid E\text{referent} \mid$$

Possible attributes, the set α, are geometrically related to the size of the set of entities E; its cardinality is limited in practice because of the previous assertion.

$$\mid E \mid < \mid \alpha \mid < \mid E \mid 2$$

1.3 Entropy and precision

---

5 A more detailed attribute description could be appropriate in special cases such as the annotation of time-based events. See Davis[93] for a system with integrated temporal attribution.

6 Hofstader[79], pp. 22

Entities reflect natural language and therefore are subject to a Zipf distribution[7] with rank n and a language-dependent constant s. Assuming no correlation between The probability of the attribute a, composed of entities rank e_1 and e_2, is the product of Zipf's normal reciprocals.

$$P_a = (e_1 e_2)^{-s}$$

In contrast to attributes, we will consider entities equally-probable for the purpose of quantifying their informational content. They are not symbols to be employed in sequence but unique records to be found.

$$P_e = | E |^{-1}$$

Deploying Shannon's entropy measure[8], the attribute a contains H(a) bits of information.

$$H(a) = - \; P_a \log_2 P_a = \; (\log_2 (e_1 e_2)^{-1} / e_1 e_2)$$

The number of bits required to identify an entity follows.

$$H(e) = - \; P_e \log_2 P_e = \log_2 | E |^{-1}$$

We can thus calculate the number of average-probability attributes, N, required to uniquely describe an entity: it will be the ratio of the information in an attribute to that in an entity.

$$N = H(e) / H(a)$$

1.4 Shape of the graph

Leveraging existing large relational databases such as WordNet®, OpenCYC, a gazetteer, etc., in subjects potentially of interest for annotation, is a natural way to increase the size of E, and thus the information potential of the database. After this step the graph will be very large and complex.

$$| E0 | << | E1 = E0 \cup WN \cup OC \cup GZ \cup ... |$$

Sets of attributes, A, in common should be inherited from a more abstract entity. This adds a few vertices but removes more edges, exploiting its multidimensonal structure to reduce the graph's redundancy.

$$A = e_i \cap e_j \wedge | A | > 1 \rightarrow e_{parent} = A : e_i \cap e_j = \{ e_{relationship}, e_{parent} \}$$

Any restriction on relationships, such as hierarchality, via filter R(a) will fail to describe some natural attribute a. Natural entity sets are not hierarchies but improper graphs with uncontrolled shape. Hierarchies will appear in the subgraphs connected by just one relationship.

$$\forall R : \exists a : R (a) = \varnothing$$

All attribute relationships should have opposites, which improves the speed at which the graph can be traversed. (see section 2). All edges in the graph are accompanied by their converse.

$$\forall a \equiv \{ e_{relationship}, e_2 \} \in e_1 : \exists a_{opposite} \equiv \{ e_{opposite\text{-}relationship}, e_1 \} \in e_2$$

A query, A, is a set of attributes; the set of entities which share them are the results of the query operation Q (A). The cardinalities of the query and result sets are inversely related; this is the classic precision/recall tradeoff.

---

7 Zipf[48]

8 Shannon[48]

$$| A | \sim | Q (A) |^{-1}$$

2 Ramifications

Automatic matching on synonyms and hyponyms of given attributes A combats divergence and makes annotations accessible to more people's queries Q (A).

$$\forall A1 \supseteq A2 : Q (A1) \subseteq Q (A2)$$

A specified entity (subject) with an attribute duple (predicate and object) represents the same information as a simple sentence or an RDF entry. Since RDF is also an XML language, it is trivial to transform between Org and RDF data. Therefore, existing RDF databases such as OpenCYC can be leveraged where appropriate.

Descriptions in Org are as powerful as those made with SQL in conventional RDBMSs. Adding attributes to an entity is analogous to adding columns to a table; each related entity is a keyed row in another table. Because this allows a system to store its own configuration, it fulfills Codd's infamous rule "#0"[9]. Org meets most of his criteria by design, but no Org implementation yet features rollback and views as Codd requires. It can thus be called a data structure or knowledge base, intended to be complemented with small application interfaces (see section 3).

Calculating inherited attributes requires traversing the entire graph, producing a much larger, less entropic database – unfortunately this is the one which we really want to query. Because the graph is fully linked (see section 1.4), we can do this in linear time, but the input is huge. Perhaps the derived file, an *exdex*, might be calculated only occasionally, or, in exchange for storage space, continuously. See section 2.2 for the procedure.

2 Algorithms

2.1 Search: O(N3)

```
// preparation: O(N log N)

disambiguate typed input          // O(N)
OR synonyms & holonyms (linear)   // O(N) ?
sort query terms A by a_i.e_referent        // O(N log N)


// recruit entities with these attributes: O(N^3)

// hypothesis: since |A| & |e|<<|E|, this N^3 is smaller than
// the N^2 that would result from looking through all E.

for each a_1 ∈ A,                                   // O(N)
        a_1 → e_1,opposite, e_1,referent                    //  O(1)
        for each e 1,referent,                              //  O(N)
                for each a_2 ∈ e_1,referent         //   O(N)
                        a_2 → e_2,opposite, e_2,referent    //    O(1)
                        store a_2 → e_2,referentin  //    O(1)


// figure which of them have all the attributes: O(N^2)

for each e ∈ ,                          // O(N)
        for each a_1 ∈ A,               //  O(N)
                a_1 ∉ e ? next e        //   O(1)
        store ein                       //   O(1)

return P                                // O(N)
```

9 Codd[86]

2.2 Inheritance traversal

. . .

3 Schema

```
<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://lithic.org/XML/org" xmlns:ev="http://lithic.org/XML/org">

<!-- schema for org data -->

<xs:simpleType name="annotation">

        <xs:attribute name="id" type="xs:string" use="required" />
        <xs:attribute name="relationship" type="xs:integer" use="required" />
        <xs:attribute name="referent" type="xs:string" use="required" />

        <xs:element name="annotation" type="org:annotation" minOccurs="0"
maxOccurs="unbounded" />

</xs:simpleType>

<xs:complexType name="record">

        <xs:attribute name="id" type="xs:integer" use="required" />
        <xs:attribute name="name" type="xs:string" use="required" />

        <xs:element name="annotation" type="org:annotation" minOccurs="0"
maxOccurs="unbounded" />

</xs:complexType>
```

3 Implementation architecture

. . .

4 Ontology Formation

One way to build a comprehensive ontology is by collaboration, perhaps even voluntary. “Folksonomies” suffer from inconsistency and need significant overhead to keep on track, but can grow quickly and stay under iterative maintenance. The series of Categories in Wikipedia[10] comprise a large loose set of is-a assertions. Projects like Wordnet and CYC run a tighter ship, but grow much more slowly as a result.

For popular music in particular, several sites are engaged in developing such ontologies. MusicBrainz and del.icio.us - ??

ADD FURNAS VOCAB PROB

http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html

5. Related Work

Stanford's Protégé
MPEG7 “collections”
Frank Nack

10 http://en.wikipedia.org/wiki/Wikipedia:Categories,_lists,_and_series_boxes

Jane Hunter
MIT Simile
FramerD
Ted Nelson: hypertext bidirectionality (matthew rothenberg)
Hypertext '87, '89: “what's (in) a link?”
Babelvision
Oval
    Abstract collaboration engine
Caliph & Emir
    “MPEG-7 based Java prototypes for digital photo and image annotation and retrieval”
Project Xanadu
    http://xanadu.com/index.html
Semantic Web!!
    swoogle search
Behrang Mohit: Semantic Extraction

Pustejovsky(1995), from Aristotle, “qualia” in Clippinger's “biology of business”, pp. 74

n Bibliography

n.1 Publications

Berners-Lee, Tim. *Semantic Web Road Map*. Self-published, 1998.

Codd, E. F. *The Twelve Rules for Determining How Relational a DBMS Product*. The Relational Institute, San José, California, 1986.

Davis, Marc. *Media Streams: An Iconic Visual Language for Video Annotation*. Proceedings of 1993 IEEE symposium on visual languages, IEEE Computer Society Press, 1993.

Furnas, G. W., Landauer, T. K., Gomez, L. M., Dumais, S. T. ***The Vocabulary Problem in Human-System Communication*. Communications of the ACM, 1987.**

Hofstadter, Douglas R. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books, New York, 1979.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J. *Wordnet: An On-Line Lexical Database*. International Journal of Lexicography, 1990.

Shannon, Claude E. *A Mathematical Theory of Communication*. Bell System Technical Journal, 1948.

Taylor, Arlene G. *The Organization of Information*. Libraries Unlimited, Inc., Glenwood, Colorado, 1999.

Zipf, George K. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley, Cambridge, Massachusetts, 1949.