

*Philip Sargent,<sup>1</sup> Eswaran Subrahmanian,<sup>2</sup> Mary Downs,<sup>3</sup> Reid Greene,<sup>3</sup>  
and Diane Rishel<sup>3</sup>*

## **Materials' Information and Conceptual Data Modelling**

---

**REFERENCE:** Sargent, P.M., Subrahmanian, E., Downs, M., Greene, R. and Rishel, D. “**Materials’ Information and Conceptual Data Modelling**” *Computerization and Networking of Materials Databases: Third Volume, ASTM STP 1140*, Thomas I. Barry and Keith W. Reynard, editors, American Society for Testing and Materials, Philadelphia, 1992.

**ABSTRACT:** We have been developing a novel method for systematizing the concepts behind the information stored in many disparate databases containing materials test data. The conceptual structure produced will be used as a basis for the development of software systems which integrate engineers’ access to many different databases. The major problems and the reason why the meanings of the concepts require explicit structuring are that (a) the everyday vocabulary used is ambiguous, (b) the same data item will have many valid names, (c) similar information is obtained by different laboratories using different methods and different scientific conceptual structures. There are also many organizational problems concerning responsibility, authority and maintenance. The complexity of the relationships between concepts requires a new technique of unusual sophistication in its handling of defaults and inheritance.

**KEY WORDS:** concepts, data dictionaries, frames, inheritance, knowledge, materials databases, materials expert systems, metadata, numeric databases, ontologies, relationships, terminology, thesaurus.

Alcoa Technical Center (ATC) is the premier light metals research facility in the USA. It provides technical support for research programs and materials testing facilities for the Aluminium Company of America (Alcoa). The testing facilities are distributed amongst of the order of a dozen different divisions and there are several hundred data repositories for test data results only some of which are computerized.

We have been developing a novel method for systematizing the concepts behind the information stored in these many databases as a basis for the development of software

<sup>1</sup> Research fellow at the Engineering Department, Cambridge University, CB2 1PZ, UK. (pms1@eng.cam.ac.uk)

<sup>2</sup> Research faculty at the Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.15213, USA. (sub@edrc.cmu.edu)

<sup>3</sup> In the Advanced Mathematics and Computer Division, Alcoa Technical Center, PA.15069, USA. (downs@aldncf.alcoa.com)

systems intended to help engineers find the information they need, irrespective of where it is physically stored or who originally arranged for the measurements to be made. These terminology problems have already been comprehensively described by Westbrook and co-workers [Wes91][Waw89].

The difficulty of finding data is such that there is the possibility that many more experimental tests are done than is necessary particularly where very similar materials' property data can be measured by very different techniques using equipment in different divisions. For example, certain chemical compositions can be determined by x-ray energy absorption spectroscopy and also by wet-chemical analysis. The system under development is intended to help engineers find the information they need, irrespective of any mismatch between the engineer's query and the actual naming system used in each database.

Our analysis of the concepts has used a new software tool "CODE" produced by the Artificial Intelligence laboratory of the University of Ottawa for the capture and structuring of complete sets of interrelated concept ontologies in particular areas of knowledge [Sku90a]. This tool has already been used to systematize the concepts used in specific software engineering and communications engineering projects (the latter in English and French simultaneously [Sku90b]), but this is the first attempt to use its uniquely sophisticated inheritance relationships to describe a wider engineering domain; that of materials' testing and materials' properties. The research also makes some use of the concept structures developed for ALADIN, an ATC knowledge-based systems project which encapsulates much of the knowledge of an expert aluminium alloy designer in the form of frames [Hul87].

Much of the structuring of concepts is a consequence of the physical fundamentals underlying materials properties and universally true facts relating properties to manufacturing processes. The results of this research will therefore be generally useful in the development of data dictionaries, data interchange procedures and next-generation, more broadly-based materials knowledge-based systems [Sar89][Sar90] [Sar91].

## **Problem Specification**

At Alcoa Technical Center over 1100 engineers, researchers, and technical support personnel are involved in materials research which includes testing, product design, alloy development, and advanced materials research. The autonomy of the various organizational groups within ATC has led to the generation of many diverse data management systems using a multitude of formats. These include paper systems, ASCII files, and proprietary formats for spreadsheet, database and statistical software systems which reside on either desktop or super-mini platforms.

Current paper systems spanning 60 years coupled with the deployment of diverse computerized data systems for storage and retrieval of automated material tests and analyses have resulted in large quantities of data. The growing volume of data has led to additional costs and time to retrieve historical information. These costs and a general lack of understanding and accessibility of previous results has led to the regeneration of expensive tests, or the use of approximations in cases where exact tests cannot be duplicated. These additional efforts reduce ATC's ability to reduce the research-to-production cycle time.

### *Independent Groups*

The nature of ATC 's research requires the sharing of data across disciplines and organizational boundaries. Each group is free to create and maintain its own data systems using procedures which satisfy its own requirements, for the most part ignoring the needs of other groups to obtain or share the data. This has resulted in islands of data with only localized knowledge about the existence, availability, structure and meaning of the information contained within the data systems. The current environment provides no simple way to determine what types of data exist or how to obtain information in a timely manner.

The situation at ATC is thus not dissimilar to that of the world community of materials database users. Users cannot be forced to coordinate and integrate their systems and integrated systems will only be developed if the benefit outweighs the effort *as seen by each group*. The major difference is that the cost of *developing* coordinating systems can be borne by an existing central body and paid for from consideration of the benefit to ATC as a whole over a period of many years. In the external community such central bodies only exist by coordination (usually voluntary and part-time) and do not have the assurance of long term funding to develop software systems.

### *The Project*

To address these issues, a project was initiated in 1990 to begin to understand ATC 's technical data environment. Only recently have researchers and management become aware of the importance of treating data as a primary 'product' of a research facility, and that the management of that resource is critical to the Corporation. The goals of this project are to "identify ATC 's critical requirements for data exchange" and "provide solutions which facilitate data access".

Before these goals can be met, a fundamental understanding of the complexity of ATC 's current data environment is required [1][Dow91]. This understanding can be gained by determining what types of data exist, where the data resides, who maintains it, and how to access the data. This knowledge will be the foundation upon which a model will be built which accurately describes ATC's data environment . The model will serve as a framework for the implementation of a data *directory* and is more detailed in the classification of materials and properties than other compilations of data sources [Waw89]. The information model and data directory is the first step in the development of a data *dictionary* and, eventually, a data delivery system. This paper describes the methodology used in creating the information model.

## **Representing Fundamental Concepts**

An important early step in the design of databases or knowledge-based systems is the listing and documenting of the basic concepts underlying the area of interest. This has to be done because these concepts represent the categories into which data will be arranged, and the classification of these concepts will influence the eventual structure of the database or knowledge-base [Nir88 Len89 Sar90 Sku90a] The structure of the relationships between these concepts must be well defined if the information content, the knowledge, embedded in the data is to be represented without distortion. This is particularly important where a number of differently trained people are involved with the construction of the

system, as has been the case for the many databases at ATC, if the whole collection of information is to make sense.

### *General Techniques*

When faced with any information structuring problem the analyst has a variety of representation schemes that can be applied. Which representation is most appropriate depends on the degree of formality required and the complexity of the task [Sar91]. If the information structure is going to become input for for some other software then it must be formally structured whereas human users can be presumed to be able to read text in a variety of forms. Recently developed software tools make it possible to use complex formal representations easily [AIL90][Par89].

The first task is to acquire a list of the important objects that must be represented, secondly these objects must be grouped in a variety of ways and relationships between these groups sought.

Hierarchy	The simplest technique for arranging sets of concepts is in a hierarchy which can be elaborated by permitting several <i>kinds</i> of inheritance (“is-a”, “is-similar-to”, “has-a”, “is-result-of-measurement-by” etc.).
Network	Networks arise when hierarchies are extended to represent <i>multiple</i> inheritance where a concept can have more than one parent.
Thesaurus	A list of terms and their equivalents or synonyms are always necessary when integrating a number of systems developed independently. <i>Structured</i> thesauri also record broader and narrower terms and thus form a multiple inheritance network (though using only two kinds of inheritance relationship, those of specialization/generalization and “relatedness”).
Dictionary	Dictionaries give definitions of concepts in terms of some base set of mutually agreed concepts. They are usually only usable by human readers. A <i>data</i> dictionary is different, it is a catalogue of variables and fieldnames used in a database.
Encyclopedia	Encyclopedias give more than a bare definition of a term, they include examples of use and illustrative material. Bare dictionaries use useful to systems developers but end-users usually require encyclopedias as well.
Indexes	Terms may be arranged in several ways: alphabetically, by keyword in the context of a phrase, as a sub-term alphabetically from a major term (as in most book indexes), or in a conceptual classification (as in the contents page of a book). Human users find a variety useful.

The software tool CODE (described in detail later) addresses most of these different needs for systems developers. McCarthy’s Data Thesaurus [McC87, McC88] combines a structured thesaurus with a formal, extensible data dictionary which defines allowable values for each type of variable.

### *Ontology and Taxonomy*

Ontology is the study of knowledge: why we know what we know. In formal systems such as databases and knowledge-bases this becomes more a technical than a philosophical study [Len89]. An ontology is a best guess at the organizing principles underlying some body of knowledge and it is typically expressed as a classification of concepts. Such classifications are usually termed *taxonomies* (by analogy with the biological taxonomy of living things) because they attempt to classify on the basis of some fundamental and important principle rather than on some convenient but superficial characteristic.

This is what we have attempted to do for ATC's data environment: produce an ontology of all the objects, variables and values that appear explicitly or implicitly in the constituent databases and document this ontology as a taxonomy of the basic concepts of materials, tests, laboratories and equipment.

### *Knowledge Acquisition and Representation*

Traditionally, data models such as the entity-relationship method [Tsi82, Rum90, Dat90] and, more recently, semantic data models [Hull87, Sar90] have been used to describe conceptual models. The tasks of building data models have usually preceded the task of designing the actual database used to store the data in a given enterprise. Our task is exactly the reverse in that data on materials is already stored in a variety of databases across the divisions of an organization and the need is for a conceptual model that serves as the basis for *integrating* these databases for the purposes locating the information.

Any effort to integrate the schemas of the databases containing the materials data leads to the problems referred to earlier: divergent and contradictory naming, differently represented metadata and the scatter of property data for particular materials across databases in different divisions [Wes91]. Nevertheless an integrated schema is what we need. It will have to have a more complex structure than most database schemas in order to take account of the diversity of its multiple constituent databases. This means that we will have to be very precise in our terminology. The size of the task also means that (a) non-materials experts will have to perform much of the work, and that (b) many different people will have to be involved in building different parts of the schema. Therefore a solid and well documented set of concepts and terms is a necessity. The need for systematic methods for developing this conceptual model becomes apparent.

### *Methods*

In searching for methods for developing the underlying "conceptual model" for materials information it became clear that using the traditional methods used for database design would not be sufficient for our purpose. Advanced representational methods in artificial intelligence such as frame based systems and object oriented systems seem appropriate for this task, but on closer evaluation no systematic methodology for developing conceptual models, especially in a scientific and technical areas, can be found.

Most methods used in the development of these systems have been one-shot or *ad hoc* in the sense that they do not make explicit the assumptions made on the characterization of the concepts that populate the system (e.g. [Hul87]). The KADS methodology does provide some structure to the knowledge acquisition process (administrative review procedures etc.) but does not include specific methods to aid the characterization and interrelating of the concepts after capture but before being programmed into the knowledge base [Bre85][Hic89]. More detailed aid is available from the knowledge acquisition portion of the KEATS system which provides tools for segmenting interview transcripts as hypertext and for drawing network relationships between knowledge frames but its built-in inheritance capabilities are restricted [Mot88][Gui88]. The SPLINTER [Zuc89b, Zuc89c] and ONTOS [Nir88] systems represent sophisticated concept structures but are university research tools not suited to use at ATC. Very recent work in the development of ontological structures for the purpose of knowledge acquisition for building text-intensive knowledge bases, machine translation and thesaurus building have started addressing this shortcoming [Nir88, Par89].

In our effort to build conceptual models we have used the a software tool CODE developed at the University of Ottawa. CODE's sophisticated inheritance features permitted us to develop multiply-inherited concepts with the ability to identify and easily

resolve inheritance conflicts. We were able to modify concept definitions and attributes while maintaining accurate graphical representations which were most useful for discussions with ATC experts whilst discussing details of particular databases.

## **CODE: a Tool for Conceptual Analysis**

The basic underlying structure of CODE is based on frame based representations from artificial intelligence and from object oriented technology. The system itself is implemented in Smalltalk and is commercially available on a variety of machines including Sun workstations and Apple Macintoshes. CODE makes explicit the representational assumptions made in the development of concept hierarchies. It also has a variety of facilities for editing, browsing and modifying the conceptual hierarchies. We will provide a brief summary of the CODE system by defining the primitive structures of the system [AIL90].

### *CODE Primitives*

Conceptual descriptors: A conceptual descriptor (Cd) is a frame or an object that is used for characterizing a concept. The conceptual descriptor is *set* of related information units known as *properties*. Conceptual descriptors usually denote classes of concepts.

Property: Property is a unit of information that can be added, removed and modified from a conceptual description. Properties can also be inherited to sub-concepts (sub-classes or instances). All properties have the following subproperties: 1) *name*, 2) a *body*, the text of the property itself, 3) *flags*, indicating the inheritance conditions, 4) *source* of inheritance and additional information on date and the name of the author who modified the property last

Concept Hierarchy: Concept hierarchy in the CODE system is an organization of concept descriptors with the specification of inheritance of properties from a general class of concepts to concepts of a more specific class *and* from a specific class to an instance of the class. Multiple inheritance of properties can be encoded.

Property Categories: In order build, maintain and modify concept hierarchies in a systematic manner, the properties themselves are categorized into *system* properties and *user* properties. *System properties* are properties shared by all concept descriptors, they include name of the concept, super-concepts from which properties are normally inherited, sub-concepts that are either divided into disjoint sets (kinds) or not and whether the concept descriptor itself is an instance of another concept descriptor or has instances of its own. *User Properties* are those that are encoded by the developers to characterize a concept further than can be accomplished by using the system properties alone.

There are three categories of *user properties*. They are: *attributes* (adjectives), *related entities* and *constraints*. An example of an *attribute* for the concept "material" would be its composition. (The attribute "composition" would have a *value* consisting of a list of elements and percentage ranges.) The category *related entities* is a relationship from concepts to other concepts unrelated by inheritance. For example, the relationship between "indentation test" and "hardness". *Constraints* allow for the specification of bounds or conditions on the property category or attributes.

An important system property category is *definitions*. These properties are used to store a definition of the concept descriptor in any format. An equational definition or a textual definition would be an example of the definition property.

There are other property categories such as *operations*, *functions*, *procedures*, and *states*. CODE is flexible enough that the users of the system are allowed to create new

categories for user properties. For example, if in defining a concept there are unresolved issues, one could add a property category called “*unresolved issues*” that can be referred to for future clarification and which would be inherited by dependent concepts.

### *Inheritance*

The flags which control inheritance are crucial to the unique flexibility of the CODE system. They control on a concept by concept basis how individual properties are inherited by sub-concepts. Inheritance is always useful for materials concepts [Dem88] but many different kinds of relationship and value must be inherited (or delegated) in different ways [Zuc89abc, Dem91]. There is fundamental difference between inherited value which *must* be identical in sub-concepts and those which are merely *defaults*.

TABLE 1 ∞ Inherited explanations for Yield Stress

#### EXPLANATIONS

**TDM Concept** is a concept used by Technical Data Management Project of Alcoa.  
**Material and Surface Property** is a fundamental attribute of materials.

**Mechanical Property** is a property of a material that is associated with elastic and inelastic interaction when stress is applied, or that involves the relationship between stress and strain; the imposed stress may be either static or dynamic.  
ASTM definition. For example, modulus of elasticity, tensile strength, endurance limit.

(Replaced force with stress in the ASTM definition. This is due to the distinction between material and specimen.)

**Strength Property** is a strength, where strength is the stress at which a material "fails". A material can have many failure modes and therefore many strengths.

**Uniaxial Strength** is strength property that is measured when force is applied along a single axis. It is a stress at which failure occurs where failure requires further definition. For example, plastic collapse, fracture, etc.

**Yield Stress** is where the failure is defined to be where the strain is no longer elastic and where the specimen achieves permanent deformation.

TABLE 1 shows the inheritance sequence of the EXPLANATIONS property (one of the category of definitions) for the concept Yield Stress. This text is produced automatically by CODE. It can be seen that some care is required to ensure that broader definitions properly include all the narrower definitions; note also that the terms “force” and “stress” must be used carefully. The need to be consistent so that software can use the concept structure is enough to force a much more precise type of definition than might be expected.

The EXPLANATIONS property is inherited with flags <i r n> meaning that it applies to *instances* (class properties are only inherited by sub-classes), that it is *refinable* and *necessary*. *Refinable* properties can only be changed in descendants in a logically consistent manner by specialization. *Modifiable* properties, conversely, could be changed completely.<sup>4</sup> Other flags are *fixed*, *universal*, *private*, *sufficient*, *optional*, and *typical*.; some define *when* a property inherits and others define whether the *value* of a property also inherits: so every concept has an EXPLANATION but the text of the explanation is different (refined) in each case.

<sup>4</sup> Bamkin has identified the handling of exceptions as a problem for inheritance systems. CODE’s use of inheritance flags to precisely specify the type of inheritance on a property-by-property basis removes most of the difficulty [Bam91].

During editing, any changes to the properties are propagated, using the inheritance flags, to all sub-concepts and the system prompts the user in case of conflicts. The *Grapher* tool allows the user to display the concept hierarchy graphically. The user is allowed to display any number of graphical views to display subtrees starting from any concept in the concept base. The user is also given the facility to move parts of the subgraphs for better readability. The Grapher can also display individual concepts and their properties as a semantic network.

CODE has the usual facilities for storing concept hierarchies as compiled images or as text files (for transfer between machines) and for pretty-printing text and graphics for better readability. There are several additional facilities that are available such as a first order deduction system (FOLDE) and ClearTalk, an english-like language for verb and noun phrases to provide for some semantic interpretation. As we do not use these subsystems for our analysis they are not discussed further.

## Concept Structuring for Materials Information

Some of the same basic elements of materials information and the relationships between them are required in computerized systems designed to support many different kinds of users: materials designers, failure analysts, materials selectors and laboratory and database administrators [Wes86]. The hope is that much of the structuring performed in the course of this project will be common to many other sets of materials databases and therefore might never have to be repeated.

However there are problems: inheritance representations are easy at first, but become difficult and arbitrary at detailed levels, i.e. different people produce different concept trees. If only a restricted set of allowable relationships is permitted (e.g. only “is-a” and “related-to”), then intrinsically complex relationships will be encoded differently by different people. If a rich set of descriptions is allowed, then some relationships could arguably be represented in any one of several ways. This is an area where standardization is expected to be required as data dictionaries become more sophisticated; as they will have to be if meaningful data interchange is to be achieved [Sar90,91].

### *Relationships and Adequacy*

The task at ATC, after the construction of a concept structure, is the generation of a complex database schema which encompasses all the constituent schemas. However before such a schema can be produced some kind of idea must be reached as to the scope of its implicit *data model*.

Codd's formal definition of a *data model* is the description of the capabilities of a database structure in terms of what can be stored, what kind of relationships are enforced and what operations can be done on the data when retrieving or updating it [Dat90]. It is distinguished from a database *schema* which is the description for a particular database structure which usually has only one set of data, one database, associated with it. When deciding what kind of data model should be *ideally* used to represent materials for any engineering purpose [Wes86], there are three fundamental issues that have to be addressed:

- (1) Is the representation *adequate*? Is it able to represent the data and relationships in a way which is useable without a distortion of the truth ?
- (2) Can the data model cope with the many different and independently invented *designations* for materials, properties, tests and processes ?

- (3) Is the data model able to make a useful distinction between structural matters (*syntax*) and the meanings (*semantics*) so that both can be handled appropriately? An example: a tensile test on a metallic specimen can give at least four types of *strength* property measurement: yield point, 0.2% proof stress, ultimate tensile stress and fracture stress. These must be distinguished correctly.

A database used by engineers must supply facilities so that (a) all these data can be stored, and (b) when data are retrieved, the user is informed when “similar” information is available in addition to that which was requested (not just identical data under a different name). For this particular problem a number of techniques are possible:

- (1) use a network or object-oriented database with some kind of default inheritance for both the terms and the data
- (2) use a relational database to store the data, coupled with an separate index to handle cross-references and conflicts, plus dictionaries and encyclopedias for user-oriented help [Bam91]
- (3) as (2) above, but instead of using an index derived from traditional exhaustive keyword manipulation, using an 'active thesaurus' derived from an investigation of the important concepts from an ontological point of view.

In the current problem the data itself is stored in a variety of existing databases and we just have to handle the description of the content and cross references intelligently. Our approach here is to use just the active thesaurus from the third technique.

#### *Hierarchical Structures: Limits to Applicability*

Trees are used for nearly all conceptual structure of materials information, but, at some level of detail, they fail to capture some important relationship. The most common example is that of *orthogonal* concepts: deformation stresses can be either *compressive* or *tensile*, and the onset of plasticity can be defined as either the *yield point* or the *0.2% strain proof stress*. These two classifications are independent and neither could really be considered a child of the other (unless an arbitrary decision is taken for the purpose of standardization). Another example is that grades of polymer can be characterized by branching frequency and by chain length, again these are orthogonal properties, best represented in tabular form.

The essay “The Architecture of Complexity” by Simon [Sim81] discusses the criteria by which one can decide whether hierarchical representations are appropriate and useful. The criteria that show when a set of objects form good hierarchies are arrived at by considering the interactions between members of the same set, and interactions between sets. If these interactions (defined in the broadest possible manner) form some kind of *sequence* from general to particular (even if only locally to those immediate sets involved), *and* if the sequence has *discontinuities* from level to level, then a satisfactory hierarchical description is possible which, without further elaboration, represents the important relationships in the system. There are many types of relationships between materials concepts where such discontinuities do not exist and thus where hierarchical descriptions can only be coarse approximations [Sar90].

CODE handles orthogonal concepts through the use of “kinds” of sub-concepts, e.g. *tensile* and *compressive* kinds of uniaxial loading, and *proof* and *yield* kinds of stress level. Bamkin has independently shown that this same principle can be represented in a relational (tabular) database [Bam91]. FIG.1 shows how the Grapher represents normal sub-concepts with an ‘s’ annotation, and the two kinds of sub-concept with a ‘k1’ and ‘k2’ notation.

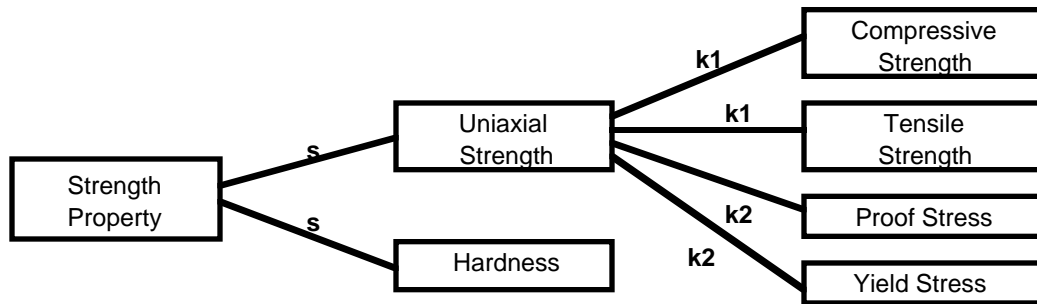


FIG.1 æ Kinds of Sub-concept

### *Tests and Properties*

The concepts and fields present in a particular database, such as strength, hardness, Rockwell C number, yield strength etc., have relationships with each other which are not usually made explicit - but they have to be described and made explicit if either (a) *standard interface* to data visualization software services is hoped for, or (b) if *data interchange* with other materials databases is planned.

The most popular and apparently natural representation of relationships between materials *concepts* is the hierarchical, tree-like structure, possibly with multiple cross-links, as used here with CODE. Thus *information* could be subdivided into *property* information or *test* information, a test could be an *indentation* test or a *tensile* test, and a property could be a *hardness* or a *yield point*. A cross-link of one type would be that *hardness* properties are measured by *indentation* tests, another type would be that yield point could be related to both *elastic* and *plastic* property classes.

The data from the same set of tests exists in a number of different databases in a variety of “states” or “bases” depending on whether it is raw data, validated data or fully evaluated information. The information which describes which state pertains to a particular piece of data, and the audit trail of the process, is information which the system under development will eventually have to maintain [Bam91]. Thus the concepts relevant to this process are classified even at this early stage.

### *Summary of Abstract Materials Issues*

Examination of the conceptual structures of many materials databases indicates that there is no one, simple correct data model structure, but that thinking about interactions and sets of concepts does give some guidance as to when to use which type of representation: *orthogonal* concepts imply tables, *sequence* and *discontinuity* in interaction imply trees, both require conceptual ontologies. The limitations mean that there is always a need for taxonomic support tools: *dictionaries* which *define* terms with respect to some basic vocabulary, *thesauri* which *relate* terms to other terms, and *encyclopedias* that *inform* on the meaning of a term with extra information and examples.

### **Data Modelling at Alcoa Technical Center**

At Alcoa, our initial modeling activity was to determine and define a set of common terms which could refer to information contained within the range of databases maintained

at ATC . This involved a compilation from a variety of resources including an ATC glossary of common terms, documentation on individual ATC databases, lists developed by ASTM committee E-49, the Common Reference Vocabulary [CRV89] and reference books on material properties. The first task was to define terms and place them into a conceptual structure.

A limited number of “high-level” concepts were identified as the major ways in which engineers or designers might want to locate data. We knew that modeling only one view of the data (e.g., based on the type of test run and not on the property measured) would not give the user interface to a data directory enough flexibility and thereby rule it useless before coding even began. These high-level concepts could be expanded to almost any level of detail. If any one concept was expanded to its fullest, information which might be pertinent to Alcoa, or others interested in material properties and testing, could be documented.

We agreed to limit the analysis to include only those concepts necessary to implement a data *directory*. In particular, the model had to include the types of tests performed at ATC since some queries might be formulated based on a particular type of test. Sufficient detail was also required to distinguish tests from each other. This included the types of results which are produced since it is these results which are most likely be stored in databases. For example, to implement ATC 's data directory, it is important that a *Tensile Test* yield values for *Tensile Strength* and *Yield Stress* and optionally for *Elongation* and *Poisson's Ratio*. However, it is not necessary to record detailed information about the machine on which the test was performed, or an exact description of either the testing method or the shape of the specimen used. If more information is required, our model could be expanded to include details omitted in the first pass.

### *Conceptual Analysis*

Early in the project, it became apparent that a method to capture all the detailed information and the inter-relationships between concepts was needed. As we were developing our methodology of classifying and relating concepts, the volume of information grew rapidly. Traditional tools such as paper, word processors, spreadsheets or relational databases were inadequate for capturing the complex inter-relationships that were evolving among our concepts. We needed a tool which would allow us to maintain and visualize our model. CODE was used for capturing and structuring our concept ontologies [Sku90a].

### *Major Concepts*

We have identified 5 major concepts in our domain: *Applications* (of materials or parts), *Data Sources* (Tests), *Entities* (Organizational Groups at ATC, Manufacturing Divisions, Databases, Materials, etc.), *Material and Surface Properties*, and *Process Parameters* (see FIG. 2). Each of these has been divided into sub-concepts extending several levels, with most of the effort to date concentrated in identifying Data Sources and Material and Surface Properties. Currently, nearly 200 concepts have been identified and this number will grow substantially as we expand the other major concepts and refine these two. A free-text definition of each concept was recorded to document agreements made as we were modelling, and to provide a means for users of a data directory to determine if terminology used at ATC has the type of information they require.

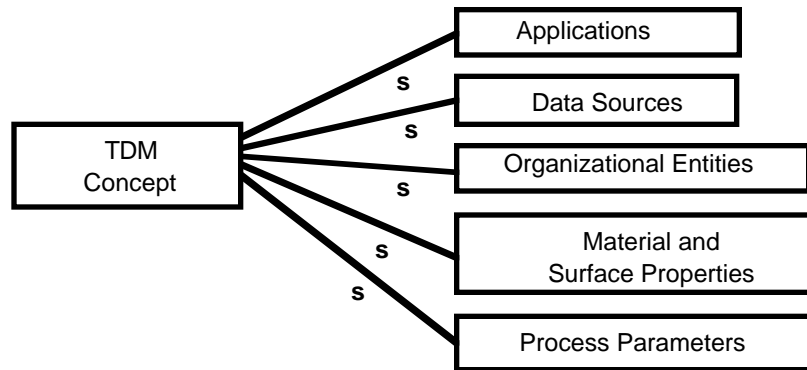


FIG.2 æ ATC Top-Level Concepts

*Detailed Example*

TABLE 2 shows the remainder of the pretty-printed listing (excluding the EXPLANATION properties shown in TABLE 1) of the description of Yield Stress at an early stage in the construction of the concept base.

TABLE 2 æ Yield Stress Concept Description

```

Cd Name Yield Stress
SuperConcepts Uniaxial Strength

ATTRIBUTES
  Name A Character String.
  Alias Yield, Y.S., Yield Strength, Tensile Yield, Fty.
  Potential Subconcept Elastic Limit, Proportional Limit, Reduction of
    Area, Strain-Hardening Exponent, Strain Rate Exponent, True Fracture
    Stress, Yield Point, Yield Strength.
  Unit Unit ; the unit of measurement.
  Value one of numeric value, symbolic value ; a numeric value can be
    either a single number or a matrix.

RELATED ENTITIES
  Stored In a set of Information Store.
  Tested By a set of Test.
  Valid For a set of Material.

UNRESOLVED ISSUES
  Strength Stress Strain Issue Need to resolve stress strain
    relationship with strength as related entities. For example, Proof
    stress and inelastic property..
  
```

TABLE 3 shows an excerpt from the full listing of the entire concept hierarchy in the form in which CODE models are transferred between machines or archived. Note the inheritance flags and the labelling of every value as to where precisely it has been inherited from, both where the property was first created and where the value of the property comes from. There is also some housekeeping information to help with version control.

TABLE 3 æ Excerpt from Full Text Concept Description

**Cd Name:** Yield Stress  
**SuperConcepts:** Uniaxial Strength

**done by** (n h v) <TDM Concept> & <Yield Stress> : Sub.  
**status** (n h v) <TDM Concept> & <Yield Stress> : nil.  
**last update date** (n h v) <TDM Concept> & <Yield Stress> : 26 July 1990.  
**creation date** (n h v) <TDM Concept> & <Yield Stress> : 15 June 1990.  
**level** (r n v) <TDM Concept> & <Yield Stress> : 5.

**ATTRIBUTES**

**Potential Subconcept** (i r n) <TDM Concept> & <Uniaxial Strength> :  
 Elastic Limit, Proportional Limit, Reduction of Area,  
 Strain-Hardening Exponent, Strain Rate Exponent, True Fracture  
 Stress, Yield Point, Yield Strength.

**Unit** (i r o) <Material and Surface Property> & <Material and Surface  
 Property> : Unit ; the unit of measurement.

### *Benefits of Having a Conceptual Model*

The major benefit is that it provides a single place to document and record information about *all* the data stored and maintained within the organisation. This is only feasible because the conceptual model records the information in a way which breaks through the diverse and imprecise naming schemes already in use. Success of linking diverse vocabularies depends on our ability to treat terms as semantic objects and to relate them based on their conceptual content [Eva91b]. The model can also be used as:

- ∑ a framework from which to elicit feedback from experts in narrow fields
- ∑ a means to obtain agreement between divisions to establish standardization of terminology
- ∑ a basis for systematic description of all stored data types
- ∑ a reference which can be used in the specification of a wide variety of new systems; perhaps most new systems

### **Further Application of the Concept Model**

Once a comprehensive data model is produced which accurately reflects the data environment, a progression of follow-on work can be performed; planned in three phases. These phases are such that if the project is terminated at the completion of any one of the phases, significant gains in the understanding and utilization of technical data would still have been achieved.

The first phase is the design and development of an on-line data *directory* accessible by every researcher and engineer. We envision this system to be one in which users can issue queries to find information about the existence of databases, classes of data, and organizations which maintain particular databases. It would point the user to a Division that could provide more information. A prototype of this system is currently being designed which will most likely be deployed in a distributed X-Windows environment accessing an object-oriented database with hypertext capabilities. CODE itself is too complex for casual users; its strength is in providing facilities for constructing and editing hierarchies, not consulting them.

Our plan for phase 2 consists of the expansion of the data directory into a passive *data dictionary*. The dictionary would provide details about the actual database tables and fields. It would relate fields to standard terms, and specify units, retrieval mechanisms, and key fields required for retrieving data.

The actual delivery of data will be the primary objective of phase 3. Engineers could retrieve data from a variety of established electronic databases via the uniform interface established in phases 1 and 2. The user would be shielded from having to know database schemas, locations, or retrieval methods. The critical issues will involve creating a communications environment to enable interoperability among diverse systems and the mobilization of resources to computerize currently paper-based data repositories.

#### *Potential extensions*

Once the dictionary has been established and covers a broad spectrum of ATC 's technical databases (at the end of Phase 2) it should be possible for *expert systems* to be built to facilitate navigation and dissemination of the various databases. For example, an expert system receiving a query requesting data pertaining to a certain chemical composition derived from x-ray energy absorption spectroscopy could also suggest data for that chemical composition derived by wet-chemical analysis. It is when expert systems begin to be developed that the true benefit of having a solid concept structure in place will be felt.

#### *Future issues*

Many organizational and technical issues remain to be addressed:

- Σ How to maintain the concept model and data directory to include new/changing research activities?
- Σ How to change the mind-set of divisions from “data is divisional property” to “data is Alcoa's property”?
- Σ How to computerize the vast amount of paper 'legacy systems'?
- Σ How to implement data management standards and ensure they are well understood and commonly used?

### **Research Issues for Materials Conceptual Structures**

Apart from its usefulness to Alcoa, formal conceptual structures for materials information have a wide application and generating them in the wider community involves additional research issues.

#### *Automatic Generation*

ATC was able to begin to structure concept relationships using the results of a survey of ATC databases which had been produced for another purpose. In general, finding appropriate concepts from a large technical vocabulary is very time consuming even before the inheritance semantics are added to the system. Developing semantic networks is expensive and, when more than one organisation is concerned, possibly controversial. They are certainly epistemologically and computationally problematic [Eva91b]. Systems for producing initial semantic networks from computerized terminology databases have

been developed for some medical fields [Eva91a] and their use for materials information should be considered.

### *Concepts and Data*

McCarthy recommended that a data thesaurus be developed which combined both data about individual materials (instances) and their classification relationships [McC88]. Although the current project has attempted to deal with just concepts, the essence of materials information makes a complete split impossible and undesirable [Sar91]. Sophisticated databases find that they have to deal with conceptual information as data, not just as part of the database schema design [Bam91]. Inheritance in concepts is different to inheritance in data [Bam91], but dealing with exceptions and classification into classes is still problematic and controversial. Nevertheless the payoffs for using inheritance are large [Bam91] since it levers knowledge and reduces bulk and repetition.

The view that classification into classes is fundamentally flawed and that inheritance based on prototypical materials is more soundly based has much to recommend it [Zuc89b, Dem91]. There is the real point that concepts intermediate in the class hierarchy have data associated with them that is not strictly deducible from the data attached to individual instances. Such intermediate level reasoning is characteristic of the early stages of engineering design and materials selection [Sar91].

There is often no real disagreement as to the substance of meanings of terms, just disagreements as to how the meaning should be represented in some particular restricted formalism (e.g. CODE, or DB2). Therefore we need to use *general* descriptive tools that do not force us to make unnecessary and difficult decisions. CODE is probably rich enough - we have even used a restricted set of facilities because it takes too much time to learn a very rich set of options. This is likely to be a problem with any sufficiently rich tool.

### **Summary**

More than half a century of organizational autonomy has resulted in a multitude of diverse technical data systems; some are automated but many are not. Numerous naming conventions and definitions have emerged which are not well-understood or accepted outside the group that defined them. Hence the development of a conceptual data model to accurately reflect a diverse technical organisation is a complicated and complex undertaking. The description must be both *formal* (explicit and accurate) and *adequate* (to sufficient detail and with enough discrimination). It must be robust enough to:

- ∑ display the network of relationships between concepts
- ∑ account for multiple types of inheritance
- ∑ provide for many alternate indexing schemes
- ∑ provide mechanisms for multiple naming schemes
- ∑ define terms completely
- ∑ deal with ambiguous everyday language

We feel that the model we have developed is an adequate representation of ATC's technical environment, and although we are still faced with the issue of keeping the model current we believe that it is a solid foundation on which to build data directory and data dictionary systems.

## Acknowledgements

We appreciate the consultations of Dr. Douglas Skuce of the University of Ottawa, the designer of the CODE software. Special thanks go also to Dr. Arthur Westerberg of Engineering Design Research Center of CMU who sat in on many of our meetings and provided us with insightful and critical feedback.

This work has been supported by the Engineering Design research Center, an NSF Engineering Research Center.

## References

- 1 Downs M., Greene, R., Rishel D., Sargent, P.M. and Subrahmanian, E. "Conceptual Data Modelling in a Materials R&D Organization", *Advanced Information Interfaces: Making Data Accessible*, Proceedings of a Symposium, National Institute of Science and Technology, Gaithersburg, June 1991.
- AIL90 Artificial Intelligence Laboratory (1990) *CODE User Manual*, Version 20 May 1990, University of Ottawa, Canada.
- Bam91 Bamkin R.J. and MacRae S.C.F. "**Experience of Designing a Relational Materials Database**" *Computerization and Networking of Materials Databases: Third Volume, ASTM STP 1140*, Thomas I. Barry and Keith W. Reynard, editors, American Society for Testing and Materials, Philadelphia, 1992.
- Bre85 Breuker J. and Wielinga B. *KADS: Structured Knowledge Acquisition for Expert Systems*, in 5th Intl. Workshop on Expert Systems and their Application, Avignon, 1985.
- CRV89 *Common Reference Vocabulary*, European Materials Demonstrator Programme (1989) copies available from Mr. G. Heine, DG XIII/B/2, Batiment Jean Monnet, Plateau du Kirchberg L-2920, Luxembourg.
- Dat90 Date C.J. *An Introduction to Database Systems*, Fifth Edition, Vols I (1990) and II (1988), Addison-Wesley Publ. Co., Reading, Mass., USA, ISBN 0-201-51381-1.
- Dem88 Demaid A. and Zucker J. "A Conceptual Model for Materials Selection", *Metals and Materials*, May 291-297.
- Dem91 Demaid A. and Zucker J. "Evolutionary inheritance and delegation as mechanisms in knowledge programming for engineering product design" in *Artificial Intelligence in Engineering*, Conference Proceedings, Oxford, June 1991. Publ. Computational Mechanics Institute.
- Eva91a Evans D.A., Rothwell D.J., Monarch I.A., Lefferts R.G. and Côte R.A. (1991) *Towards Representations of Medical Concepts*, presented at AIMA 91 (American Medical Informatics Association), San Francisco, June 8, 1991.
- Eva91b Evans D.A., Handerson S.K., Monarch I.A., Pereiro J., Delon L. and Hersh W.R. *Mapping Vocabularies Using 'Latent Semantics'*, Carnegie Mellon University Laboratory for Computational Linguistics technical report CMU-LCL-91-1, July 1991
- Gui88 Guida G. and Tasso C., Editors, (1988) *Topics in Expert System Design*, Elsevier, Amsterdam.

- Hic89 Hickman F.R. et al (1989) *Analysis for Knowledge Based Systems: A Practical Guide to the KADS Methods*, Ed. R.M. Taylor, Ellis Horward Publ.
- Hul87 Hulthage I. Przystupa M. Farinacci M. Rychener M.D. (1987) "The Metallurgical Database of ALADIN - An Alloy Design System", in Harrison R.J. and Roth L.D., Editors (1987) *Artificial Intelligence Applications in Materials Science*, Proc. Symp. held in Orlando, Fla., Oct.8 1986. Publ. The Metallurgical Soc. Inc. (AIME) ISBN 0-87339-067-9.
- Hull87 Hull R. and King R., (1987) "Semantic Database Modelling Survey, Applications and Research Issues" *ACM Computing Surveys* **19** (3) 201-260.
- Len89 Lenat D.B. (1989) *Ontological versus Knowledge Engineering*, IEEE Trans. on Knowledge and Data Engineering **1** (1) March 1989 84-88.
- McC87 McCarthy J.L., "**Information Systems Design for Materials Property Data**" in *Computerization and Networking of Materials Data Bases*, ASTM STP 1017, J.S. Glazman and J.R. Rumble Jr., editors, American Society for Testing and Materials, Philadelphia, 1989, pp135-150.
- McC88 McCarthy J.L., "The Automated Data Thesaurus: A New Tool for Scientific Information", 11th CODATA Conference, Karlsruhe, Sept.26-29th 1988.
- Mot88 Motta E. Eisenstadt M., Pitman K. and West M. *Support for Knowledge Acquisition in the Knowledge Engineer's Assistant (KEATS)*, Expert Systems **5** (1) Feb. 1988.
- Nir88 Nirenberg S., Monarch I., Kaufmann T., Nirenberg I. and Carbonell J. *Acquisition of Very Large Databases: Methodology, Tools and Applications*, Carnegie Mellon University Center for Machine Translation technical report CMU-CMT-88-108 (1988).
- Par89 Parsaye K, Chignell M, Khosafian S. and Wong H. (1989) *Intelligent Databases: Object-oriented, Deductive and Hypermedia Technologies*, John Wiley and Sons, ISBN 0-471-50345-2.
- Rum90 Rumble J.R. and Smith F.J. *Database Systems in Science and Engineering*, Adam Hilger, 1990 ISBN 0-7503-0048-5.
- Sar89 Sargent P.M., "**Use of Abstraction in Creating Data Dictionaries for Materials Databanks**", *Computerization and Networking of Materials Databases: Second Volume*, ASTM STP 1106, J.G. Kaufman and J.S. Glazman, editors, ASTM, Philadelphia, 1991, 114-131.
- Sar90 Sargent P.M. "**Data Models and Data Dictionaries for Materials Information**", Proceedings of CAMSE'90, 1st International Conference *Computer Applications to Materials Science and Engineering: Computer Aided Innovation of New Materials*, Aug.28-31, 1990 , Ikebukuro, Tokyo, Japan.
- Sar91 Sargent P.M., *Materials Information for CAD/CAM*, Butterworth-Heinemann Publ., Oxford, UK.
- Sim81 Simon H.A., *The Sciences of the Artificial* (MIT Press, Massachussets, 2nd Edition, 1981).
- Sku90a Skuce D. and I. Monarch, *Ontological Issues in Knowledge Base Design: Some Problems and Suggestions*, Banff Workshop on Knowledge Acquisition, 1990.
- Sku90b Skuce D. and I. Meyer (1990) *Concept Analysis and Terminology: A Knowledge Based Approach to Documentation*, 13th Intl. Conf. on Computational Linguistics, COLING 90, 20-25 August, Helsinki, Finland.
- Tsi82 D.C.Tsichritzis and F.Lochovesky, *Data Models*, Prentice Hall, New York, 1982.
- Waw89 Wawrousek H., Westbrook J.H. and Grattidge W. (1991) "**Data Sources of Mechanical and Physical Properties of Engineering Materials**", *Computerization and Networking of Materials Databases: Second Volume*,

- ASTM STP 1106*, J.G. Kaufman and J.S. Glazman, editors, ASTM, Philadelphia, 1991, 142-157.
- Wes86 J.H. Westbrook, H. Behrens, G. Dathe and S. Iwata, Editors, (1986) "Material Data Systems for Engineering", Proc. CODATA Workshop, Schluchsee, FRG, 1985. ISBN 3-88127-100-7, 78-80, 110.
- Wes89 Westbrook J.H. and Grattidge (1989) "**The Role of Metadata in the Design and Operation of a Materials Database**", *Computerization and Networking of Materials Databases: Second Volume, ASTM STP 1106*, J.G. Kaufman and J.S. Glazman, editors, ASTM, Philadelphia, 1991, 84-102.
- Zuc89a J.Zucker and A.Demaid, "Selection of Engineering Materials", *Proceedings 5th Scand. Symposium on Materials Science*, Ingeniorhuset, Copenhagen, Danish Soc. for Materials testing and Research, 1989.
- Zuc89b J.Zucker, *Engineering Design Computed by Prototypes and Descriptions*, PhD Thesis Sept. 1989, Faculty of Technology, Open University, Milton Keynes, UK.
- Zuc89c Zucker J. and Demaid A. (1989) "A Software Machine Designed for Selection", *Knowledge-Based Systems Journal*, Butterworths, **2** (3) 178-184.