NISTIR 6401

COMPUTER-INTEGRATED KNOWLEDGE SYSTEM (CIKS) NETWORK: REPORT OF THE 2ND WORKSHOP

Lawrence J. Kaetzel, Editor
K-Systems
Brownsville, MD

Building and Fire Research Laboratory
Gaithersburg, Maryland 20899

NIST

United States Department of Commerce
Technology Administration
National Institute of Standards and Technology

NISTIR 6401

COMPUTER-INTEGRATED KNOWLEDGE SYSTEM (CIKS) NETWORK: REPORT OF THE 2ND WORKSHOP

Lawrence J. Kaetzel, Editor
K-Systems
Brownsville, MD

December 1999

Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

National Institute of Standards and Technology
William M. Daley, *Secretary*
Technology Administration
Cheryl L. Shavers, *Under Secretary for Technology*
National Institute of Standards and Technology
Ray Kammer, *Director*

2. KEYNOTE PRESENTATIONS

The Link between Terminology and Data Element Dictionaries

Sue Ellen Wright
Kent State University Institute for Applied Linguistics

1. Introduction

The following discussion of data element specifications for the purpose of data interchange is based on the author(s) experience in creating a data element dictionary in the context of ISO/TC 37, *Terminology (Principles and coordination)*, as well as long-term involvement with terminology standardization in association with the ASTM Committee on Terminology and its successor, Technical Committee E02 for Terminology.

Collecting data and maintaining data collections cost money. Given the availability of substantial data collections in both the public and private sectors, exchange of data among systems opens up the possibility for considerable savings, both by eliminating the need for duplicated research and data entry and by supporting increased efficiency in cooperative environments.

Although these apparent advantages can be exploited at least theoretically in materials databases and logistics, in every aspect of computer aided design, in medical diagnosis and insurance documentation, in military administration and countless other fields (not to mention terminology documentation itself), data incompatibility has in the past posed a serious obstacle to the realization of potential benefits. Nevertheless, development of such standard formats as the universal patient record and the MARC record are moving information management in the direction of harmonization. Such formats, however, rely on the clear specification and definition of the categories of data elements that are used in local databases so that data can be interpreted and utilized across system and organizational boundaries.

2. Terminological Aspects of Data Elements

2.1 Polysemy

Data incompatibility can be described with respect to terminological considerations. Starting at the lowest level of abstraction, even very similar data elements are frequently defined from different viewpoints, are assigned different scopes, and are subject to data modeling variance in the context of data architecture. From a terminological perspective, they can be said to be subject to *polysemy*.[i]

Polysemous terms are represented by the same term but have different meanings, i.e., the concepts they represent have different characteristics. In like manner, ambiguous data elements share the same data element name, but they have slightly, in some cases dramatically, different content. These differences are reflected in the form of differing data element definitions or different sets of data element attributes.

Of course, sound database management practice dictates the avoidance of multiple meanings for data element names, so one can expect that individual databases will avoid this kind of ambiguity internally. However, as soon as different systems (even different independent systems within the same organizational entity) are linked, variations become evident. Furthermore, ambiguous data element definitions or faulty implementation at the data-entry level may well impair data integrity and prevent effective reuse or sharing of resources.

2.2 Synonymy

Inadequate tracking of objects within a database can also result in the assignment of more than one data element name to the same object. This kind of duplication is akin to *synonymy*. The existence of such *doublettes* in, for instance, materials management modules is a major source of unnecessary inventory management cost and even materials acquisition error and expense. Not as obvious, but perhaps even more critical, is the absence of systematic structure in many object management systems. A well-conceived item master listing the objects managed by a system can reflect a semantic network that represents the intrinsic relations that exist among objects in the system. Put more simply, naming and numbering systems can be structured so that they reveal the answers to questions like: Is A a *kind of* B (generic relation)? Is A a *part of* B (part-whole relation)? Are A and B *subsequent steps in a process* C (sequential or possibly cause and effect relation)?

Incorporating this kind of conceptual organization into data structures lays the groundwork for data management systems to evolve into information and knowledge management systems. Unfortunately, many existing systems were based on illogical or arcane criteria, such as vestigial reference to obsolete project numbers, instead of building on intrinsic characteristics such as species identity (e.g., related bolt designs), partitive relations (sub-assemblies in a major component), or procedural sequence (operations in a process). Systems that do not reflect network structures not only increase current data-management costs; they also shut the system off from the introduction of more intelligent, inferential systems in the future.

2.3 Terminological Principles for Naming Data Elements

The logical result of these observations is that master data files should be subject to the same criteria that are dictated for standardized terminology.

- One and only one data element name is assigned per data element concept.
- One and only one concept is associated with a given data element name.
- (Terminologists will recognize these demands immediately: data element names, like terms, should be mononymous and monosemous.)
- Data element definitions must be concisely and yet adequately formulated.
- Data element dictionaries must be structured in logical ways that reflect meaningful, i.e. semantic, relations among data element concepts, such as parent-child, sibling, or part-whole relations.
- Data element dictionaries should clearly distinguish related or easily confused data elements from one another.

Within the context of terminological principles, it is also valuable to observe the fundamental differences between terminological entries and lexicographical ones. The chart appended to this article as Annex A illustrates the distinctions between these two methods for documenting words and meaning. Within the framework of this article, it is especially important to emphasize the concept-oriented basis for terminological resources, which is reflected in the kinds of definitions that must be associated with data dictionaries as well.

3. The Dangers of Implicit Information

Ambiguous data element definition and inaccurate data entry practices may not be critically problematic as long as a database only serves as an information repository for human users because humans infer additional information or supplemental organizational orientation based on unstated, i.e., implicit, contextual assumptions. Furthermore, information entered in individual databases may well be quite clear in its own immediate environment, but the potential for ambiguity arises when an effort is made to merge data with other databases designed to meet different needs.

These potential problems become more critical when data have to interact efficiently within automated systems or when they must be automatically exchanged among sub-systems within the same network or among different autonomous systems. Traditionally, automated routines have not functioned on an inferential basis, which means that to be successfully used in data interchange environments, data structures need to conform to one of possibly three options for achieving an environment where all participants (systems) have the same understanding of the meaning for each and every piece of data:

- All systems use the same data architecture.
- All systems have access to each other's data architecture and implement the appropriate translation routines to interpret data appropriately.
- Unified data element specifications provide the information needed to allow for inferential processing by intelligent systems.

Evaluation of the costs and effort involved in implementing any one of the three mechanisms leads inevitably to the conclusion that the cost of implementing the first option would be patently prohibitive, not to mention the fact that no one uniform architecture is likely to meet diverse needs and expectations. Furthermore, with respect to the second option, it is usually a challenge to any work group simply to develop, implement, and maintain *one* data architecture, let alone track other architectures as well. This line of reasoning leaves us with the final option of developing unambiguous data element specifications that offer the greatest benefits for future-oriented system design.

4. Data Element Attributes

Data element specifications are analogous to terminological definitions. A well-crafted definition clearly states the *characteristics* that it shares with other concepts in its class (shared parent-child characteristics) and that distinguish it from other closely related concepts (differentiating sibling characteristics). In like manner, data element specifications disambiguate data elements by

specifying the *attributes* associated with each data element. These attributes comprise the minimum information required for unequivocally identifying data elements.

The minimum information required for data element specifications includes the designation of the *data element name*, a statement of its content, and the formulation of an adequate *definition*. In addition to these features, specification of any given data element can include information on the *data type* of the data element, a listing of *permissible instances* (content), and determination of the level of *granularity* represented by a given element or set of elements. The function of data elements is further affected by such practical aspects as the avoidance of *redundancy* through the creation of *shared resources*, *data element autonomy*, *data modeling variance*, *combinability, repeatability*, and interaction with existing *standards*. All these considerations affect the naming and the definition of data elements. The examples cited in the following discussion are taken from the list of data categories (data elements) compiled by ISO/TC 37, *Terminology (principles and coordination)* (ISO 12620:1997; Schmitz 1997).

4.1 Data Element Definition

Data element *specifications* contain *definitions* as one of their principal attributes, although some terminologists prefer to talk about *data element* or *data category descriptions* instead of *definitions* in order to distinguish them from definitions in terminology collections. A data element specification may stipulate that a particular element contains xyz..., but it is also important to *define* xyz in order to ensure that users clearly understand what the content of the category really is, (e.g., contains xyz, which is ...). Definitions can, however, reside in a parallel terminology standard where they can be referenced, provided that they are all in the same place and readily accessible.

Definitions are important in order to avoid ambiguity. For instance, in terminology databases the data elements ***example*** and ***context*** are sometimes confused with one another. Other elements, such as the terminological data element ***transfer comment*** are widely used in some work groups but unknown to other terminologists. A dictionary of data elements is indispensable for mapping local data elements to standardized data elements that can be used for interchange purposes. Such definitions should ideally meet the requirements for terminological definitions by spelling out the relation of the concept represented by the data element within its semantic network and distinguishing each data element from closely related data elements.

4.2 Data Type

Data element dictionaries can also specify the *data type* of the element, e.g., free text, dates, numbers, etc. Data type designations can also be associated with references to standardized formats for representing information, such as standards for dates, country and language symbols, or international identification numbers used as bibliographical references. One of the most common forms of representation is to require the use of specific SI units. In this regard, some databases may also specify field length for such standardized elements, although more and more

knowledge-oriented systems are moving away from fixed field lengths, especially in terminological and bibliographical resources.

4.3 Permissible Instances

Restriction of element content to a specified data type and standard is one way to state the *permissible instances* that can occur in that data element. Another way is to create a *pick list* from which users select the appropriate content for any given case. The advantage of using defined pick lists where appropriate is to prevent generating different forms for representing the same meaning or introducing undocumented content. For instance, a standard could require that for interchange purposes, *part of speech* should be represented as *n, v, adj, adv*., thus ruling out the option for using *noun, verb, adjective, and adverb*, or opting for *substantive* instead of using *noun*. The advantage of this degree of specification is that merged databases will exhibit uniform content. Validation programs can be used to check to ensure that candidates for interchange conform to the standard.

4.4 Granularity

Examining data element definitions often reveals that different databases treat similar data elements in different ways. For instance, Figure 1 illustrates an example of the data element ***address***. Figure 2 divides the same information into finer units. In order to ensure that the correct data are input into any of the fields in Figures 1 and 2, or that users can interpret these data appropriately, the data elements must be clearly defined and data entry must conform to these definitions. This kind of difference between approaches to subdividing information is commonly referred to as *granularity*.

Data element	**Content**
Name	John Doe
Address	267 Prospect St. Kent Ohio 44240 USA

Figure 1: Minimum granularity

Data element	Content
Last Name	Doe
First Name	John
Street Address	267 Prospect St.
City	Kent
State	Ohio
Zip Code	44240
Country	USA

Figure 2: Increased granularity

The data modeling scheme followed in Figure 2 can be described as more powerful because data that are categorized in this way can be more easily retrieved or manipulated. For instance, the entry shown in Figure 2 can be sorted or retrieved by the last name, town, state, zip code, or country. (Theoretically, it could also be sorted by the given name, but this application is probably rare.) It might, however, be desirable to be able to sort by street name and number, in which case the data element ***street address*** could be subdivided in order to achieve higher granularity. A typical example of variation in granularity in a multilingual terminological data element involves the decision whether to include all grammatical information in a ***grammar*** data element or to specify ***part of speech, gender*** and ***number*** instead.

Individual database systems define the granularity of data elements in keeping with perceived data management needs, although it is very important that system designers think through these needs carefully when modeling data in order to ensure that the structures that are implemented at the outset will indeed meet future requirements for data manipulation and maintenance. It is important to bear in mind that in interchange environments, differences in granularity tend to limit data reusability. Allowing for broader categories facilitates interchange and may be less expensive (or look less expensive at the beginning of a project), but this practice reduces power and freedom to manipulate data. It is very simple, for instance, to write a conversion routine that would combine elements in Figure 2 to fit into elements in the database represented by Figure 1, but it may be very difficult, if not impossible, to devise an automatic way to split the elements shown in Figure 1 so that they can be assigned to the more granular model shown in Figure 2. Thus, when planning an interchange formalism or creating a data element dictionary, it is desirable to indicate the degree of granularity desired for automatic interchange when defining data elements.

Some instances of granularity, such as the *address* example cited above, reflect the needs and philosophy of the system designers. Other choices, however, are not so optional. For instance, the inclusion of source information in the same data element together with a text field or even the

inclusion of page numbers together with the source identifier that references a shared resource violates the essential *integrity* (*Sauberkeit*) of the database and prevents efficient manipulation of data. Special attention should be given to avoiding shopping basket style data elements such as undifferentiated ***notes*** or ***comments*** that can result in many different kinds of information being stored in the same data element.

4.5 Term Autonomy and Data Modeling Variance

The principle of *term autonomy* is particularly relevant to termbases, but similar situations may also exist in materials databases. Even when termbases adhere to the one-concept-per-entry dictum, *data modeling variance* from one system to another often results in non-uniform presentation of data. For instance, in a terminology entry, one possible presentation is to designate a so-called main entry term in a ***term*** data field, while other terms appear in other data elements such as ***synonym***, ***abbreviation***, and the like (Figure 3a). The weakness of this particular approach is that it frequently fails to provide the option to include complete documentation for each additional term (Schmitz 1997).

Term Entry, Data Element	**Content**
Term:	personal computer
Part of speech:	Noun
Definition:	A computer designed for personal use.
SourceID:	Oxford1990
Responsibility:	SEW
Abbreviation:	PC
Synonym:	Microcomputer
Term:	personal computer
Part of speech:	Noun
Term type:	preferred term
Definition:	A computer designed for personal use.
SourceID:	Oxford1990
Term:	PC
Part of speech:	noun
Term type:	Abbreviation

Term Entry, Data Element	Content
Context:	PCs are now common in all office environments.
SourceID:	Kluge1997
Term:	microcomputer
Part of speech:	noun
Term type:	synonym
Definition:	A computer containing a microprocessor as the CPU.
SourceID:	Oxford1990
Responsibility:	SEW

Figures 3a and 3b: Data modeling variance

Figure 3b illustrates a more egalitarian approach that reflects the view that *all terms are created equal.* Here each term associated with a concept is reported in an independent ***term*** element that can then be associated with its own data segment containing such subordinate elements as *part of speech*, *gender*, *number*, and *term type.* It is here where information such as *synonym, abbreviation,* etc. can be included as attributes of the *term.* Term autonomy is closely related to the principle of *repeatability* and *combinability*, which provides for the reuse of data elements and the establishment of internal data element relations within individual data entries.

4.6 Redundancy and Shared Resources

A cardinal rule of database management is that redundancy should be avoided. For instance, it is possible to include complete bibliographical references after each text element included in a term entry (definitions, contexts, etc.), but if an error occurs in a reference or information needs to be updated, it becomes necessary to find each citation to the same source and make the desired change. Consequently, it is more economical from the standpoint of database management to enter the bibliographical reference once in a single bibliographical entry or table and to link all the individual entries that cite that resource to a shared bibliographical entry. This approach also contributes to efficiency with regard to increased granularity because it is much more feasible to enter articulated data in a single bibliographical entry that will be used as a shared resource than it is to do so in each individual term entry. The treatment of these kinds of shared resources in databases impacts the content of function of the data elements used to form links and requires the specification of data elements used to document the shared resources themselves.

Bibliographical entries are not the only kind of shared resources included in terminology files. Termbases, for instance, feature links from term entries to responsibility entries, graphics, video and audio files, tables and charts, diagrams of concept systems, lists, materials management systems, thesauri, and other types of related resources. Other kinds of databases may feature these or similar shared resources as well. Shared resources may reside as bundled data inside actual databases or be accessed via hypertext links to other file types residing in the same system or accessible via network connections.

5. Domain-specific Subject Field Specifications

The unambiguous interchange of almost any kind of terminological or knowledge-based information is predicated on the presence of uniform subject classification codes. The selection of an existing coding system or the creation of a new one must be undertaken by or in close collaboration with experts in the field in question. Numerous disciplines have developed widely recognized thesauri for data retrieval purposes that can serve this purpose. In some areas, like medicine, for instance, great effort has been invested in standardizing a uniform classification scheme in order to resolve problems involving conflicting systems. Without the acceptance of universal context-oriented domain references it will be impossible to implement context-oriented inferential knowledge systems using shared data. Terminologists, information scientists, translators, and technical writers frequently serve as non-expert manipulators of information-related data. They need to be able to depend on the expert knowledge that subject-field specialists can provide by creating or standardizing a subject-specific classification system.

6. Coordinating Data Element Standardization

The Joint Technical Committee of the International Standards Organization and the International Elect-Technical Commission (ISO/IEC JTC 1/SC 14, ISO 11179) has undertaken to develop standards for coordinating data element standardization designed to ensure accurate, reliable, controllable, and verifiable data recorded in databases. They are currently developing a suite of standards dealing with:

- Standardization of data elements
- Classification of concepts for the identification of domains
- Specification of data element attributes
- Formulation of data definitions
- Naming principles for data elements
- Registration of data elements

These topics read like an outline for terminology management: identification of data element concepts (concept selection), subject-field classification, the identification of characteristics (attributes), composing adequate definitions, term formation, and providing repositories of standardized terms.

7. Application Environments

The developers of a data element dictionary need to determine for themselves whether this activity is sufficient to meet their needs or whether they need also to specify a uniform interchange format or a standard data structure. The developers of ISO FDIS 12620, *Computer Applications in Terminology Data Categories* have also been involved in the development of an SGML interchange format, ISO FDIS 12200, *Computer Applications in Terminology Machine-Readable Terminology Interchange Format (MARTIF) Part 1: Negotiated Interchange*. The concept of *negotiated interchange* implies that database managers will have to look carefully at the data they are importing and will subject that data to various validation and conversion routines before integrating it into their own systems. Indeed, the data elements in the source material do not have to conform to the standard in their native mode, but are brought into conformance during the conversion process.

A debate continues in TC 37 whether it is not more desirable to standardize a much more stringent so-called generic exchange architecture that will facilitate a uniform representation of data outside their original database environment. The purpose of such a structure would be to consolidate terminological data into a global terminology resource in the Internet environment. No final decisions have been made on the exact appearance of such a format, and it may well be that the desirability of such a uniform expression of information is typical only to terminology or perhaps bibliographical records, such as in the case of the MARC record. Similar discussions continue in the lexicography field as well, although there appears to be less commonality of viewpoint in that area than among the terminologists.

Regardless of the environment in which the data element dictionary will be applied, the need to identify and define the data-element-related terms used in a discipline is an essential component of a process designed to create a data element dictionary for the purpose of ensuring effective data management during the interchange of data among diverse systems.

References

Gesellschaft für Terminologie und Wissenstransfer (GTW), e. V. 1996. *GTW-Report: Guidelines for the Design and Implementation of Terminology Data Banks*. Saarbrücken: GTW/Universität des Saarlandes.

GTW. 1996. *Criteria for the Evaluation of Terminology Management Software*. Felix Mayer, ed. Saarbrücken: GTW.

ISO 690:1987, *Documentation — Bibliographic references —Content, form and structure*.

ISO 639:1988, *Code for the representation of names of languages.*

ISO 1087:1990, *Terminology — Vocabulary.*

ISO 1087-2, *— To be published. Terminology Work — Vocabulary — Part 2: Computational applications.*

ISO 3166:1993, *Code for the representation of names of countries*.

ISO 8601:1988, *Data elements and interchange formats — Information interchange — Representation of dates and times.*

ISO/IEC 11179-3:1993, *Information Technology — Coordination of data element standardization - Part 3: Basic attributes of data elements (types)*. ISO/IEC/JTC 1/SC 14.

ISO 12 200:1999, *Computer Applications in Terminology — Machine-readable Terminology Interchange Format (MARTIF) — Negotiated Interchange*. Geneva: ISO.

ISO 12620:1999, *Computer Applications for Terminology — Data Categories*. Geneva: ISO.

Schmitz, Klaus-Dirk. 1997. "Über wichtige Aspekte bei der Einrichtung einer rechnergestützten Terminologiedatenverwaltung" ["Important Aspects of Setting up Computer-Supported Terminology Management Systems"], *Proceedings of the 11th European LSP Symposium on Language for Special Purposes*. Copenhagen: Copenhagen Business School (in press).

Strehlow, R. A.; Kenworthey, W. H., Jr.; Schuldt, R. E. 1993. "Terminological Aspects of Data Elements in Databases", Sue Ellen Wright and Richard A. Strehlow, eds. *Standardizing Terminology for Better Communication. Practice, Applied Theory, and Results*. ASTM STP 1166. Philadelphia: American Society for Testing and Materials, 129-139

Wright, Sue Ellen, and Budin, Gerhard. 1997. *The Handbook of Terminology Management.* Amsterdam and Philadelphia: John Benjamins Publishing Company.

Annex A: Lexicographical vs. Terminological Entries

LEXICOGRAPHICAL ENTRY	TERMINOLOGICAL ENTRY
❑ Treats a word (frequently called a head-word)	❑ Treats a concept and is sometimes identified by a code rather than a word
❑ Treats multiple polysemic senses of the word based on one etymological derivation	❑ Treats one concept in one entry, and documents all terms assigned to that concept
❑ Treats homographic words with different derivations in separate entries	❑ Treats polysemic meanings of the same word in separate entries
❑ Provides all grammatical information pertaining to the word	❑ Cites only those grammatical differences that may be related to term-concept assignment
❑ Is arranged in strict alphabetical order for easy access	❑ Often is arranged according to a systematic concept structure, with alphabetical cross-listing
❑ Describes, or at most, recommends usage	❑ Frequently documents preferred or recommended usage
❑ Usually treats words as a universal set taken from general language	❑ Treats terms belonging to a domain-specific special language

(Wright and Budin 1997:328)

Unfortunately, the terminology used with data element dictionaries themselves involves some degree of polysemy. The term *data element* can be construed as either a single field (a *data element instance* or *data element item* or it can be viewed as a *class of data elements*, i.e., all instances of a data element occurring in a database. In the terminology community, this potential for confusion has led to the evolution of the term *data category*, which is defined in ISO/TC 37 standards as *an instruction for interpreting a given **data field*** (ISO DIS 1087-2.2: 1997). Unfortunately, this term evolved parallel to the development of the term *data element* in English parlance and can cause confusion. For purposes of this audience and this paper, I will use the term *data element* to refer to a category of data elements and, if necessary, *data element instance* to designate an individual instantiation of a data element, which can be construed as *a unit of **data** that in a certain context is considered indivisible* (ISO 2382-4:1987).