

**ADVANCES IN DATA INTEGRATION AND QUALITY CONTROL IN SUPPORT OF THE NNSA
KNOWLEDGE BASE**

Richard Stead, Michael Begnaud, and Julio Aguilar-Chang

Los Alamos National Laboratory

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Development
Office of Defense Nuclear Nonproliferation

Contract No. W-7405-ENG-36

ABSTRACT

The goal of the NNSA Ground-based Nuclear Explosion Research and Engineering program (GNEM R&E) is to develop, demonstrate, and deliver advanced technologies and systems to operational monitoring agencies to support ground-based detection, location, and identification of nuclear explosions. One such system is a custom-designed data storage and access system known as the NNSA Knowledge Base (KB), primarily based on relational database schemas (sets of table structures). The GNEM R&E research conducted at the national laboratories to populate the KB requires collection and integration of a remarkably large and diverse collection of geophysical data to develop the types of products needed to improve monitoring capability. The size and diversity of these data present substantial technical challenges to achieve complete, correct, consistent, useful, and accessible information. These data are processed by the labs to produce the higher-level engineering products (e.g., travel time correction surfaces) that are needed for operational monitoring, but the basic data must also be included in the KB to fully test and verify the operational products. Without the supporting data and metadata capturing the processing details, the operational engineering products cannot be validated and thus will not be used for operations. Los Alamos National Laboratory (LANL) has developed and contributed to several versions of the Knowledge Base and in the process we have developed and refined a substantial foundation of software, structures, and procedures to assure high-quality integration of diverse data sets. Software advances include generalized database interfaces and generalized quality assurance/quality control (QA/QC) software. Structural advances include a metadata abstraction of supporting structures (themselves metadata) that we refer to as the schema schema. Procedural advances leverage the software and structures to create robust procedures for definition and transfer of data between groups.

The development and application of automated QC software is the primary topic of this paper. Attention to quality, particularly in the supporting data, has been a subject of growing importance and focus recently. Dealing with data quality in an established KB is difficult and time-consuming. A better approach is automated QC of supporting data before they are integrated into the KB. We have taken several steps in this area over the past year, including the development of automated quality-inspection software. The first critical step in automating QC was to make the information about the schema readily available to the QC software, which was done by developing a set of tables describing the schema itself, or a "schema schema." The schema schema captures information about the content of each table (what the columns are), the relationships between the tables, and information about each column (definition, acceptable range of values). With this in hand, we then developed a Perl-based QC tool to check the content of any set of tables against what is in the schema schema. The software performs three basic types of checks: 1) validity of column data within each table, 2) consistency of column data between related tables, and 3) more complicated consistency relationships between related tables. Because the software makes use of the schema schema, it was written in a generic manner and thus can be used for virtually any sort of check without modifying code. A user sets up a parameter file to designate the database tables that will be checked, writes the specific checks for #2 and #3 above, and then initiates the QC check. The output can then be used to direct the KB integrator to problems in the incoming data. The QC tool was used extensively for the production of the latest KB release with excellent results. Thousands of problems were found and fixed prior to integration thereby greatly improving the quality of the KB and dramatically reducing the amount of post-delivery editing.

OBJECTIVES

The GNEM R&E program has made recent progress in developing and automating QC of the NNSA KB and in the supporting metadata architecture. The goal is to improve the quality of data and derived calibration results in the KB, as well as improving the speed and accuracy of the procedures for incorporating new data and derived calibration results. A related goal is to improve human interfaces to increase the efficiency and effectiveness of KB integrators.

RESEARCH ACCOMPLISHED

Background: The Need for Quality Control

Quality control in this paper will refer to the quality assurance and quality control of both geophysical data and research products derived from those data. QC for the GNEM R&E KB is an issue that has increased in priority and significance recently. While QC has never been ignored, the increased maturity of various processes and products developed under GNEM R&E and advancing toward operations has brought necessary scrutiny. This scrutiny is different and far more intense in many ways than that typically experienced in the research community. At the same time, the volume and diversity of data and products is challenging by any standard (Stead, 2006).

A variety of typical methods of QC exist that are adopted in smaller research efforts. One is to include manual review of every field for every element of data and products by the researchers. This method works well, since the researchers are the experts and will immediately recognize errors, but it is only viable for very small and focused collections of data and products. A second method is sampling, where the expert examines a small sample of data and results to verify the correctness of these, and projecting the QC result to the whole. If properly applied, this can be a viable method for the QC of data, even large data sets. But it takes an intimate knowledge of the particular data source to know how best to properly sample the data and have confidence in the results. It is not a good approach for research products, since it implicitly assumes random errors and normal distributions. A third method is to develop a dedicated application to QC a particular type of data or research product. But this exercise then must be repeated for every new type of data or research product, and the dedicated applications require constant maintenance to account for changes in data or products. This is not meant to be an exhaustive list, but merely a representative sampling to indicate the disadvantages of such ad-hoc approaches when the QC problem deals with very large and diverse data and research products, many of which are inter-dependent, where there are frequently new kinds of information or important changes to existing information, and where much of the information is targeted for operational purposes (see Figure 1).

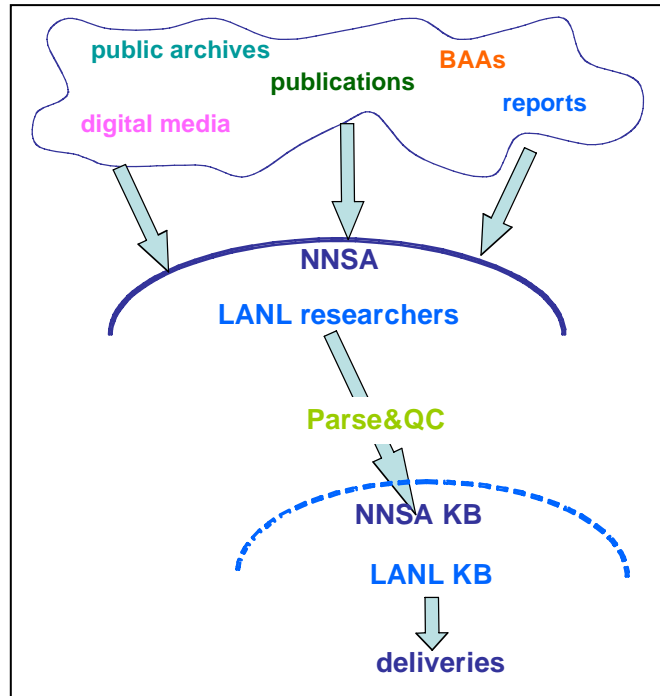


Figure 1. Generalized data acquisition and integration process. External resources of data are obtained or received by LANL researchers, then, after review and prioritization, are parsed into the standard KB structures and QC'd before being integrated into the LANL KB. The LANL KB is part of the larger NNSA KB. The formal NNSA KB, maintained by SNL, comes from the delivered portions, which undergo further review and QC at SNL before delivery to AFTAC. This view of the process is notional, and the actual procedures may involve multiple iterations and even loops among the kind of operations depicted.

Faced with this situation, LANL has been developing more comprehensive and flexible QC that can be automated. We have also worked closely with Sandia National Laboratories (SNL) on these issues, but this paper will primarily deal with the work done at LANL.

Infrastructure for QC

Any generalized, automated QC capability will require a certain amount of information infrastructure to support the process. At a fundamental level, such QC will require information that describes what qualified data and products are. A generalized QC process then merely has to match the data and products to the information and indicate which elements violate the qualification. The automated, generalized QC will only reach a level of achievement that the information infrastructure on which it is built will permit. Therefore, careful consideration must go into the construction and population of the infrastructure.

Some requirements for the infrastructure are that it be

- sufficiently general to describe all conceivable types of data and products;
- simple to understand and populate, directly or through user interfaces;
- simple to maintain, directly or through user interfaces;
- capture as much information about what qualifies data and products as possible; and
- multipurpose, so that any effort in populating and maintaining it can benefit other work.

To this end, LANL initiated the development of a database-based metadata model to support QC and other database development work beginning about two years ago. The basic concept behind this metadata model is that it is a metadata model to describe other metadata, and because of this design, it has come to be referred to as the “schema schema” (i.e., a schema to describe schemas). It is simple in that it has only four objects in the schema: an object to

describe tables, an object to describe columns, an object to describe the association between tables and columns, and an object to provide definitions of fixed values that columns may take. Additional objects can be added that relate to these basic four, to handle other special-purpose tasks (one is the **complexjoin** table which will be described below).

This concept was deliberately designed to have a close relationship to the Oracle data dictionary, to make the use and understanding of the schema simple and straightforward. However, it does go well beyond the Oracle data dictionary in various ways. In particular, because it exists apart from the tables it describes, it exists at all times regardless of which objects are currently defined or how they are defined. It also supports all of the concepts embedded in earlier documentation of the KB schema, including external flat-file formats and NA values.

The structure is quite amenable to web-based interfaces, making the population, maintenance, and visual interface simple to use and understand.

The structure is also multipurpose. By designing it to capture the information in previous database description documents, providing the capability to maintain multiple versions of such information in a single place and with a single interface, and providing simple tools to maintain the information, the schema schema has replaced the flat-file documentation as the medium of choice for direct maintenance of all schema information and documentation. LANL has developed web-based interfaces that allow users to quickly view database documentation in a familiar format, directly out of this schema, eliminating the piles of massive paper documents that had to be used in the past. SNL has also developed additional interfaces for viewing and maintenance. In addition, the schema provides a direct and simple means for software of all types to determine the objects and their structure at run time, as opposed to maintaining software-based descriptions of all of this information. SNL has made extensive use of this feature in the development of their Java-based **dbtoolkit**. LANL is using this feature in a variety of Perl-based software.

Figures 2–4 show the LANL-developed web interface for viewing the table descriptions and column associations, the column descriptions, and the glossary (the 4 core schema schema tables). See Begnaud, et al., 2005 for additional information on the schema schema and the web interfaces to the KB.

NNSA KB Core
[Tables](#) | [Columns](#)

Select a Table
 Click on table name at right to get details of all columns for that table

[affiliation](#)
[arrival](#)
[assoc](#)
[event](#)
[gregion](#)
[instrument](#)
[lastid](#)
[netmag](#)
[network](#)
[origerr](#)
[origin](#)
[remark](#)
[sensor](#)
[site](#)
[sitechan](#)
[region](#)
[stamag](#)
[wfdisc](#)
[wftag](#)

Schema Description for: NNSA KB Core

gregion
 NNSA KB Core

The **gregion** table contains geographic region numbers and their equivalent descriptions.

#	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	NA VALUE	DESCRIPTION
1	grn	number(8)	i8	1-8	-1	geographic region number
2	grname	varchar2(40)	a40	10-49	Not Allowed	geographic region name
3	lddate	date	a17:YY/MM/DD HH24.MI:SS	51-67	Not Allowed	load date

Flatfile Format Lines
 Perl, Matlab, C: %8d %-40s %-17s
 Fortran: i8, 1X, a40, 1X, a17

Keys:
 Data:

Primary	grn
Descriptive	grname
Administrative	lddate

Show create table script for a new table like: [gregion](#)
 Enter Name for New Table:

Figure 2. Table description web GUI. This web panel shows the complete description of a table from a particular schema, along with the column associations for the table and SQL statements and format lines that correspond to the table.

Column name Schema Description for: NNSA KB Core

Column Descriptions

Tables

Description

Formats

Constraints, etc.

List of columns

Name: **magnitude**
 Schema: NNSA KB Core
 Table: netmag, stamag
 Description: Magnitude. This column gives the magnitude value of the type indicated in magtype. The value is derived in a variety of ways, which are not necessarily linked directly to an arrival (see magtype, mb, ml, and ms).
 Format: Internal: float(24)
 External: f7.2
 Units: magnitude
 NA Value: -999
 Range Type: numeric
 Range: -9.99 < magnitude < 50
 Expert Opinion Range: -5 < magnitude < 10
 Regular Expression: -
 RefSchema: -
 RefTab: -
 RefCol: -
 Auth: NNSA

Figure 3. Column description web GUI. This web panel shows the complete description of a column from a particular schema, along with the table associations for the column.

Glossary Items for: NNSA KB Core Search Glossary:

Glossary Items for Column: dtype -- Schema: NNSA KB Core

Full search

Schema name

Definitions

Terms

Column name

List of columns

Name	ID	Table Name	Owner	Definition	Auth	Load Date
A	14768	-	-	assigned	NNSA	2004-10-22 17:37:01
D	14769	-	-	depth restrained > 2 pF phases	NNSA	2004-10-22 17:37:01
d	16578	-	-	from depth phases	NNSA	2004-10-22 17:37:01
E	14774	-	-	good depth estimate - < 8.5 km	NNSA	2004-10-22 17:37:01
f	16577	-	-	free, unconstrained	NNSA	2004-10-22 17:37:01
G	14771	-	-	from FINR, unknown meaning	NNSA	2004-10-22 17:37:01
g	16580	-	-	restrained by geophysicist	NNSA	2004-10-22 17:37:01
L	14772	-	-	less reliable - 8.5-16 km 90% conf	NNSA	2004-10-22 17:37:01
N	14770	-	-	restrained to normal depth - 33 km	NNSA	2004-10-22 17:37:01
P	14773	-	-	poor depth estimate - > 16 km error	NNSA	2004-10-22 17:37:01
Q	16624	-	-	from FINR, unknown meaning	NNSA	2004-10-22 17:37:01
r	16579	-	-	restrained by location program	NNSA	2004-10-22 17:37:01

Figure 4. Glossary web GUI. This web panel shows the complete description of selected glossary entries. The entries may be selected by schema name, table name, and column name, or by the search feature.

Automated QC

A first-generation QC tool has been developed at LANL. The tool leverages the schema schema described above to facilitate automated inspection of data and derived products that exist in database tables. The tool can operate on any collection of tables, but it can only do the most basic kinds of inspection unless the tool can match each table to a corresponding description in the schema schema. The tool is applicable in a variety of situations. A version has been adopted by the DOE labs, as a group, to handle QC of deliveries between the two science labs and SNL. LANL has found the tool to be of considerable use in dealing with raw or 'roughed-in' data sets and research products. Running the tool on such information allows a wide variety of issues to be identified quickly and corrections then planned and made separately. The results also then are used as feedback to improve the conversion of data sets from their raw form into the KB, and also to improve both the structure of research products, and even the processes that generate research products. The tool has significant advantages over ad-hoc QC, in that it adapts automatically to changes in the underlying schemas, no specialized software is required if new data or products become available, and there is no need for tedious manual inspection or for experts to remember or track all of the possible QC errors that may arise.

The automated tool requires very little information to run. The basic run-time parameters are the connection to the database and the list of tables to be examined. Currently, if cross-reference checks are desired, these must also be specified individually, since there is no generalized table in the schema schema to describe these. In practice, a user of the tool will keep a list of cross-reference parameters, and copy them as needed into a run-time parameter file for a particular set of tables to be examined. The tool learns everything else it needs from the database. The tool executes three stages of QC checks: single-table checks, cross-reference checks, and complex joins.

The single-table checks consist of four table-based checks (beyond simple existence of the table), followed by a large number of column-by-column checks (that depend on the type of column), for each table specified. First, the tool determines the standard names of the schema and table. This verifies that the table matches a documented table structure and provides the tool access to a complete description of that table and its columns, glossary entries and complex joins. It counts the number of rows (a primitive check – a lack of rows or too many may indicate a problem), and then validates the table's primary key, and its unique key (if any). That completes the table-based checks. The column-based checks may be basically divided into general checks and then checks performed only on string columns and only on numeric columns. The general column checks include counting the unique values, NA values and nulls, determining the minimum and maximum of the column, and finding duplicated values. None of these may be errors, but they can indicate errors when subsequently reviewed. The character column checks include checks for bad character strings, strings that do not fit a columns regular expression (if any), and comparisons to the glossary for columns that have defined or reference set values. The numeric column checks include various range checks, checks for NA values other than that defined for a column, and checks for negative values.

The multi-table or cross-reference checks are individually specified and executed. There is no need for the tables in the cross-reference checks to be specified in the list of tables to check. This allows checks against reference tables, without having the tool perform full checks (single table and complex join) of the reference table itself. There are three basic kinds of checks: single column cross-references between tables, two-column cross-references between tables, and indirect cross-references. A single-column cross reference requires that every value in the specified column of the table being checked is also found in the specified column of the table that the first table is to be compared against. There are different versions of this check, since the underlying queries may be faster or slower under different circumstances. The two-column check is the same as the single-column check, except that each unique pair of values from two specified columns in the table being checked must be found in the two column specified for the comparison table. The indirect cross-reference check handles cases where the reference for a column in a table being checked is specified by a second column in the same table. The archetypical example of this is the **wftag** table, where the reference for the **tagid** column is specified by the **tagname** column; that is, if **tagname** equals 'evid', then **tagid** contains an **evid** and should be compared to columns in other tables that are **evids** (like **event.evid**), but if **tagname** equals 'arid', then the contents of **tagid** must be compared to **arids** (like **arrival.arid**).

The third and final checks are the complex joins. These are specified in an auxiliary table to the schema schema. This table can specify up to three tables to be involved in the join from the list of tables being checked. The QC tool will automatically format and run a check if the kinds of tables the check requires are among the list of tables the user provided. Since the table contains a generalized SQL query, these checks can be as complex as SQL permits. Some simple examples include verifying that a date in a table corresponds to the time specified in the same table,

28th Seismic Research Review: Ground-Based Nuclear Explosion Monitoring Technologies

verifying that the contents of a field in one table correspond to those in another, when there is no direct link between the tables (such as **wfdisc.instype** versus **instrument.instype**), and verifying that a count listed in a field of one table equals the count of those objects in another table (such as **origin.ndef** versus the count of time-defining arrivals in **assoc**).

Figures 5 and 6 provide examples of the QC tool output.

```
...
TABLE AMPLITUDE (658054 rows):
* 'AMPLITUDE' found as 'amplitude' in 'USNDC P2B2'
  no PK violations
                                [keychk: 6 s]
...
COLUMN UNITS:
  NA value is '-', and allowed is n
    Unique values =           2
    NA values ('-') = not allowed
    Min =                nm
    Max =                nm/s
Most common repeated values, top 10:
-----
      VALUE      NUMBER
-----
          nm      413060
         nm/s     244994
[mult_chk: 1 s]
244994 (1 distinct) entries not in glossary [glosschk: 2 s]
Unreferenced values, top 10:
-----
      VALUE      NUMBER
-----
         nm/s     244994
[glosschk: 1 s]

No regexp
No bad string values.
                                [bvchk: 0 s]
...
```

Automated recognition

Checks keys

Checks column contents

Figure 5. QC tool: single-table check stage: example output.

```

...
487 (479 distinct) entries in stamag.(arid,delta)
not in assoc.(arid,delta) [ref2chk 130: 1 s]
Most common failed references, top 10:
-----
                ARID                DELTA        NUMBER
-----
                65206475            1.009         2
                65206479            1.194         2
...
'origin.nass = the count of rows in assoc for the orid'
Enforce: soft
Checking complex join 27 'ORIGIN.nass'
= (select count(*) from ASSOC s where s.orid=a.orid)
991 (69 distinct) values violate join, top 10:
-----
                VALUE        NUMBER
-----
                6            101
                8             87
                5             84
                7             68
                9             66
               10             62
                4             61
               11             45
               12             36
               13             30
               14             27
                [complexjoin_chk 27: 7 s]
...

```

Cross-references

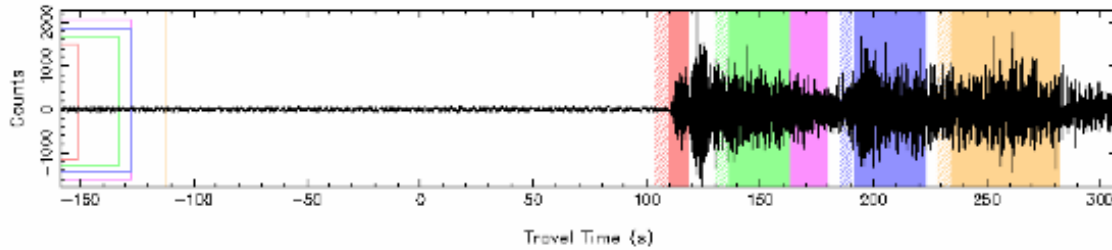
Complex joins

Figure 6. QC tool: cross-references and complex joins: example output.

Limitations

This is a first-generation QC tool and there are a variety of limitations. Two limitations are prominent. The first is that this is a QC inspection tool, not a QC inspection and repair tool. The second is that this QC tool is limited to inspection of data that are in the database. The NNSA KB schema (Carr, 2005) contains objects, principally **instrument** and **wfdisc**, which point to and describe external data. Other tools are required to incorporate these data into overall QC review.

Some tools exist to handle these limitations. In particular, we have recently extended the LANL web interface to the KB to include waveform review. Waveforms may be viewed in a browser, using basic selection criteria, and other information, such as amplitude measurement windows may be overlain on these images to provide a simple visual inspection (see Figure 7).



NNSA Amp_descript Holdings													
ORID: 572381, WFID: 1371													
Station: MAKZ, Channel: BHZ10, Delta: 7.603, SEAZ: 128.741													
(click WINDOWID to view Amplitudes for that window) (click on CLR to Zoom To that phase window)													
CLR	WINDOWID	PHASE	GVLO	GVHI	TOFF	START_TIME	DURATION	ALOGID	AUTH	LDDATE #	ARID	ARRIVAL_CHAN	ARRIVAL_LDDATE
521779	P	7.6	8.2	0	110.06	8.13	-1	LANL	2003-01-13 18:37:54	-1			
521788	S	4	4.7	0	191.8	31.47	-1	LANL	2003-01-13 18:37:54	-1			
521791	Lg	3	3.6	0	234.74	46.95	-1	LANL	2003-01-13 18:37:54	-1			
521785	Pc	4.7	5.7	0	148.26	31.54	-1	LANL	2003-01-13 18:37:54	-1			
521782	Pg	5.2	6.2	0	136.3	26.21	-1	LANL	2003-01-13 18:37:54	-1			
521792	PrEvNLg	-1	-1	-999	-159.22	46.95	-1	LANL	2003-01-13 18:37:54	-1			
521780	PrEvNP	-1	-1	-999	-159.22	8.14	-1	LANL	2003-01-13 18:37:54	-1			
521786	PrEvNPc	-1	-1	-999	-159.22	31.55	-1	LANL	2003-01-13 18:37:54	-1			
521783	PrEvNPg	-1	-1	-999	-159.22	26.22	-1	LANL	2003-01-13 18:37:54	-1			
521789	PrEvNS	-1	-1	-999	-159.22	31.47	-1	LANL	2003-01-13 18:37:54	-1			
521793	PrPhNLg	-1	-1	-999	228.74	5	-1	LANL	2003-01-13 18:37:54	-1			
521781	PrPhNP	-1	-1	-999	104.06	5	-1	LANL	2003-01-13 18:37:54	-1			
521787	PrPhNPc	-1	-1	-999	142.26	5	-1	LANL	2003-01-13 18:37:54	-1			
521784	PrPhNPg	-1	-1	-999	130.3	5	-1	LANL	2003-01-13 18:37:54	-1			
521790	PrPhNS	-1	-1	-999	185.8	5	-1	LANL	2003-01-13 18:37:54	-1			

Figure 7. Web-based frame for visual inspection of amplitude measurements relative to the waveform.

CONCLUSIONS AND RECOMMENDATIONS

We have developed a metadata schema to describe metadata, which has matured to the point now that it undergirds a variety of software within the GNEM program. It is now a component that is helping to meet a variety of tasks and requirements in a manner that improves efficiency and allows work done in documenting and maintaining metadata information to have immediate benefits elsewhere.

We have developed a first-generation QC inspection tool that relies in the schema schema. QC, which is always important, has become prominent in GNEM recently. Metadata and the proper handling of metadata are the keys to getting a handle on QC, particularly for large and diverse collections of data and research products like the GNEM KB. The QC tool has proven to be very useful in a variety of situations, and has now been deployed throughout the GNEM labs. We expect that the tool will undergo further development, based on experience, to increase both ease of use and automation.

The QC efforts have not yet extended to automated or semi-automated correction of errors. This is an obvious next step. We plan to investigate this subject as well, and we expect to find that a variety of QC problems can be addressed at least in a semi-automated fashion. By semi-automated, we mean that general rules for the repair of problems can be created that can then be adapted to particular problems in software, and applied to those problems

28th Seismic Research Review: Ground-Based Nuclear Explosion Monitoring Technologies

given simple approval by an expert. This differs from manual repair in that the repair does not need to be formulated from scratch each time.

ACKNOWLEDGEMENTS

The authors wish to acknowledge LANL personnel who are making or have made important contributions to the development and maintenance of the LANL KB, and have provided helpful discussions on these and related topics: Diane F. Baker, Hans E. Hartse, W. Scott Phillips, George E. Randall, and Marian D. Romero-Yeske. We also wish to acknowledge helpful discussions with SNL personnel, which improved the work reported here and its more general application: Sandy Ballard, Dorthe Carr, Marcus Chang, Jennifer Lewis, and Chris Young.

REFERENCES

- Begnaud, M. L., R. J. Stead, J. Aguilar-Chang, and H. Hartse (2005). Optimizing data Access and availability for seismic calibration research, in *Proceedings of the 27th Annual Seismic Research Review: Ground-Based Nuclear Explosion Monitoring Technologies*, Vol II, p.888-897.
- Carr, D. (2005). National Nuclear Security Administration Knowledge Base Core Table Schema Document, Sandia National Laboratories report SAND2002-3055.
- Stead, R. J. (2006). Data Collection and Integration in Support of the NNSA Knowledge Base, *Eos Trans. Am. Geophys. Union* 87: 36, Joint Assembly Supplement, Abstract S31B-03.

