INEL-94/00025

Conf-941207--28

INTEGRATED SPENT NUCLEAR FUEL DATABASE SYSTEM[a]

DOE Spent Nuclear Fuel Challenges and Initiatives

Steven P. Henline
Idaho National Engineering Laboratory
P.O. Box 1625
Idaho Falls, Idaho 83415-3879
(208) 526-2561
FAX (208) 526-2930

Kent G. Klingler
Idaho National Engineering Laboratory
P.O. Box 1625
Idaho Falls, Idaho 83415-3879
(208) 526-8376
FAX (208) 526-2930

Burrell H. Schierman
Idaho National Engineering Laboratory
P.O. Box 1625
Idaho Falls, Idaho 83415-3779
(208) 526-2879
FAX (208) 526-8363

Developed by

Distributed Information Systems Software

Subject

Quantity and Location of DOE Spent Nuclear Fuel

BACKGROUND

The Distributed Information Systems Software Unit at the Idaho National Engineering Laboratory has designed and developed an Integrated Spent Nuclear Fuel Database System (ISNFDS), which maintains a computerized inventory of all U.S. Department of Energy (DOE) spent nuclear fuel (SNF). Commercial SNF is not included in the ISNFDS unless it is owned or stored by DOE. The ISNFDS is an integrated, single data source containing accurate, traceable, and consistent data and provides extensive data for each fuel, extensive facility data for every facility, and numerous data reports and queries.

IEEE Standards

Development and maintenance of the ISNFDS follows the American National Standards Institute/Institute of Electrical and Electronic Engineers (ANSI/IEEE) Software Engineering Standards. Adhering to IEEE standards throughout the entire life cycle provides the rigor to ensure a quality product.

Software Quality

Software quality is managed by incorporating plans that govern the project, the development process, and the software. These plans work in concert to control procedures and methodologies. In the development, the product quality is a function of tracking the requirements through each phase to ensure traceability.

The system is managed according to the following published standards and procedures:

- The software quality assurance plan prepared for this project identifies and defines the standards and methodologies required to ensure conformance to accepted quality standards.
- The software verification and validation plan used for the project describes the criteria for verification and validation activities for the requirements, design, testing, and all necessary documentation.
- The project's software configuration management plan describes the methods used to identify software configuration items; control and implement changes; and record and report change implementation status.
- The software requirements specification (SRS) defines the requirements, based on the functional and operational requirements, essential to developing the product. All requirements must be internally consistent and verifiable through demonstration, analysis, or testing.
- The software design description for the project describes in detail how the product meets the requirements specified in the SRS. It defines the major features, including operating environment, databases, tables, internal and external interfaces, overall structure, sizing, modeling, and system throughput.

---

[a]Work performed for the Assistant Secretary for Environmental Management under DOE Contract DE-AC07-94ID13223.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

- The software test plan (STP) includes testing of the system and components, and acceptance and integration testing; it also details the procedures for describing the scope, approach, resources, and schedule of test activities. The STP identifies test items, features to be tested, testing tasks, person to perform each task, and risks that would require contingency planning. The plan also identifies the specific roles for preparing the other software test documentation and executing the tests.

- The software test cases are used to test the system; they also detail the specific requirements and design being satisfied and the step-by-step procedures for testing.

- The project's data management plan documents the data and database management activities. It describes history, problem resolutions, and general information contained in the system.

- The security plan details the information for handling security needs of the data, software, and hardware and describes the details of contingency planning and system protection.

- The software user documentation provides the user with information pertaining to installation, navigation, functions, and features of the system.

PROTOTYPE

Development of the Software Requirements Specification and Software Design Description incorporated a prototype developed and tested using data that were collected for a technical report. The prototype provided valuable experience, particularly in user interface and data entry and verification.

Prototype Methodology

The prototype, which facilitated communication with the end user, was developed from the functional and operational requirements, with input from the users. It was used to develop screens and user interface and was further used in the SDD to develop formats for the screens, reports, and queries.

The fact that the prototype was a milestone task placed an urgency on its development and did not allow a continual cycle of defining and redefining requirements. The prototype was frozen on the milestone date. Otherwise there would have been an endless cycle of iterations. The milestone forced us to make decisions and brought a focus to the project team.

Prototyping was an iterative process of discussing requirements with the customer, developing a working model, demonstrating the model and taking suggestions, and improving the model. Developing reports and queries worked in a similar fashion by sketching the report, developing a model, and letting the customer review and comment on format and content.

Requirements/Design

Prototyping was used in the requirements and design phases to demonstrate functionality. The prototype was also used extensively to show screen design, which helped build the interfacing and functionality of the screens. The screens were then used for the SRS and were also captured and placed in the SDD. The same was true for report design. The prototype helped determine the format and content of the reports. Again, the reports were used for the SRS and were captured and placed in the SDD.

The user interface, using function keys and a hierarchical menu structure was created and refined through the use of prototyping. The user helped determine the flow of screens and access through the hierarchy by using the prototype.

The prototype was also used to help establish database structure. It gave us the opportunity to populate our databases, determine possible problems with data integrity, and protect against those problems in the final system. We also used the prototype as an aid in modeling the relationships of the data.

Our customer was able to use the prototype as an aid in verifying data. Much of the first data was input into the system for purposes of demonstration and testing; this allowed us the opportunity to scrutinize the data. As the system was being developed, the users also were able to develop their operating procedures based on the flow of work in the prototype.

Lessons Learned

Use established programming standards while prototyping. We found that much of the prototype could be reused in developing the system. However, we had not always followed programming standards when making quick modifications. It took as much or more time to fix the problems as it would have to start over. It is difficult to start over when you have the programs in hand.

Build the prototype as though it were the production system. Because some of the code was put in place during the prototype, it was possible to reuse it. However, many shortcuts were taken in the haste of building or modifying the code. The time cut was amply re-spent in bringing it back into place.

Establish a clear definition of the prototype with the customer. When customers see something, they become anxious to use it. It is difficult to explain why it is not immediately available and to explain why it will take such a long time to make it available.

Acquire customer help and involvement. Prototyping is definitely a good communication tool. It eliminates trying to conceptualize how the product will function and look.

Prototyping helps test the unknowns of SRS and SDD. Prototyping can be used to test some of the unknowns of the software and hardware and telecommunications.

SYSTEM FUNCTIONS

The system is composed of two main functions. Data management includes the entry, verification, modification, and protection of the data, while data reporting allows the creation or retrieval of reports for printing, viewing, or downloading.

Data Management

Information sent as required from data calls and other requests is checked for format and content. Personnel following established data entry procedures enter information into the database using the data entry software. This information is isolated in a separate environment until it has been quality checked and electronically verified in preparation for electronic publication. These data may be modified as a result of change requests or data discrepancies discovered during data verification. Upon concurrence from the Data Change Control Board, the data administrator may modify these data. Data must be verified before they are published. Verification reports are available as an aid in the verification process. Data that are marked as verified are moved by a separate process initiated by the data administrator. As the data are processed, they are once again certified for data integrity.

Queries and Reports

The published information is made available to registered users in a variety of formats. The most common requests for information have been developed and are available for on-line queries, formatted for printing, and can be downloaded for importing into other software.

Reports are designed so that they can be reused. The user may use a single report format for different purposes, depending on the selection of data. This saves design time and provides the users commonality among the different reports generated. Reports may also be printed or used to create an ASCII file for downloading into other software packages.

System Design

Screens are also designed for reusability. A screen format may be used for different processing activities. Once again, this saves design time and provides the users commonality among the different screens. For example, the same screen is used for entering, modifying, verifying, querying, and reporting data. These capabilities are determined by the system according to the level of access granted to each user. The functions available for each user are listed at the bottom of each screen.

The system is designed in a hierarchical menu fashion. Many of the functions depend upon the data being present and valid before functionality exists.

Each record is stamped with an identifier of the person who last updated and last verified the correctness of the data. Also included is a date stamp indicating when the action occurred.

The data administrator is provided utility functions to optimize database performance. When records have been verified, the data administrator runs the process that makes them available for inquiry and reports.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.