12-21

SANDIA REPORT

SAND95–2760 • UC–905
Unlimited Release
Printed December 1995

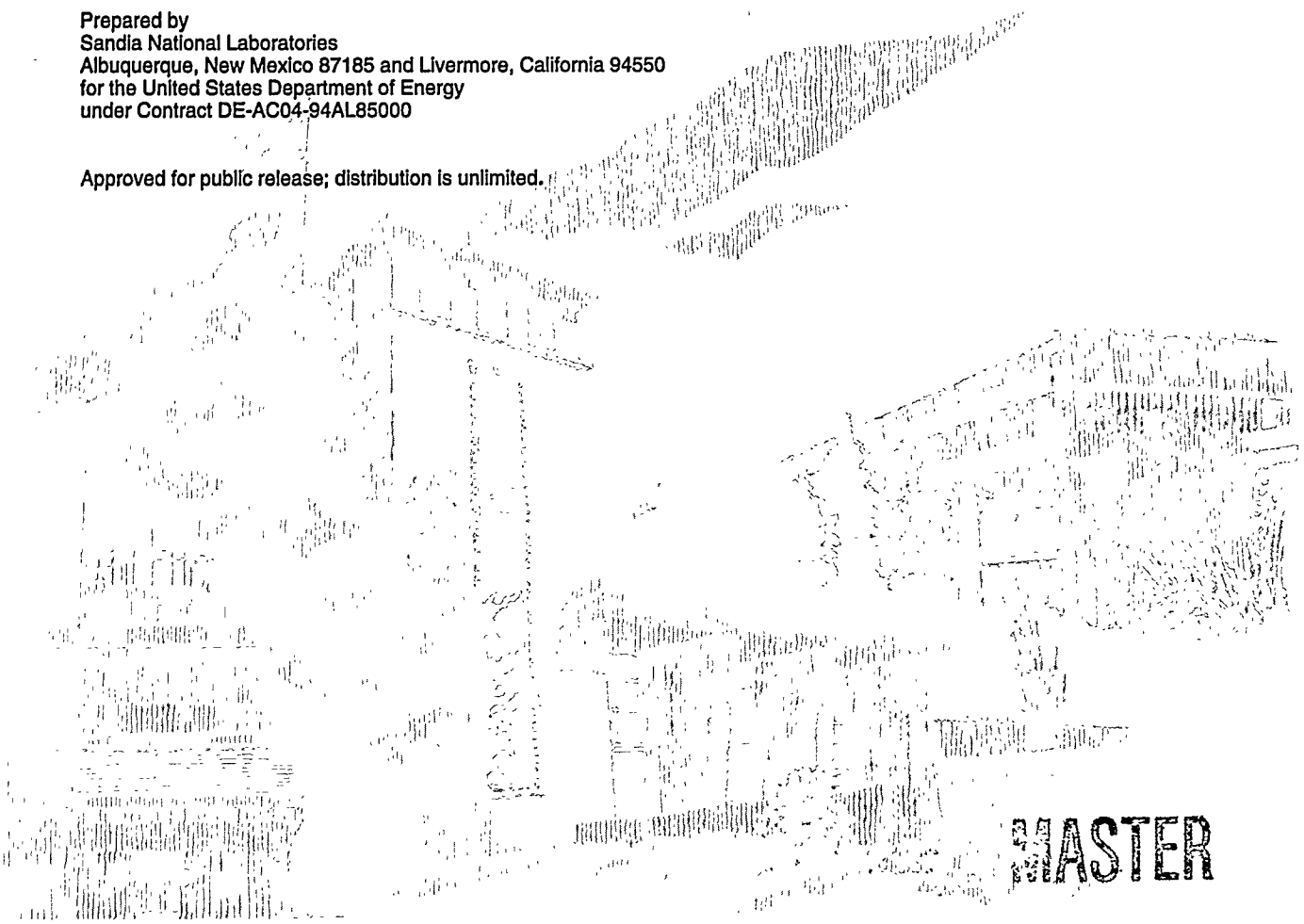
Software and Information Life Cycle (SILC) for the Integrated Information Services Organization

Adaptations of the Sandia Software Guidelines: Issue B October 16, 1995

Donna Eaton, Andrea Cassidy, David Cuyler, Shelley Eaton, Scott Joyce, Elisa Kephart, Irene Thurston, Joe Schofield, Dwayne Knirk

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A06
Microfiche copy: A01

SAND95-2760
Unlimited Release
Printed December 1995

Distribution
Category UC-905

Software and Information Life Cycle (SILC) for the Integrated Information Services Organization

Adaptations of the Sandia Software Guidelines: Issue B
October 16, 1995

Donna Eaton, Andrea Cassidy, David Cuyler, Shelley Eaton, Scott Joyce, Elisa Kephart, Irene Thurston, and Joe Schofield (Integrated Information Services)

Dwayne Knirk (Quality Engineering Department)

Sandia National Laboratories
Albuquerque, NM 87185

Abstract

This document describes the processes to be used for creating corporate information systems within the scope of the Integrated Information Services (IIS) Center. Issue B describes all phases of the life cycle, with strong emphasis on the interweaving of the Analysis and Design phases. This Issue B supersedes Issue A, which concentrated on the Analysis and Implementation phases within the context of the entire life cycle. Appendix A includes a full set of examples of the deliverables, excerpted from the Network Database. Subsequent issues will further develop these life cycle processes as we move toward enterprise-level management of information assets, including information meta-models and an integrated corporate information model. The phases described here, when combined with a specifications repository, will provide the basis for future reusable components and improve traceability of information system specifications to enterprise business rules.

This document is available for viewing on the Sandia Internal Web, under the URL

http://sass902d.itd.sandia.gov/SILC_B.pdf

Table of Contents

Introduction

This report is designed to serve three overall purposes:

- Short-term:
  - To unify the baseline deliverables produced by the analysis and design processes
  - To provide guidance for software developers who need to prepare for formal design reviews
- Long-term: To move corporate software and information life cycle processes toward a unified and consistent methodology, serving diverse software development projects at Sandia

Scope

Short-Term Scope—Deliverables

This report defines processes and checklists that will move the Integrated Information Services (IIS) organization toward an enterprise-level specification management process that will clarify and document:

- The rules for business operations
- The corresponding requirements for the automated information systems

This report prescribes a useful set of materials that describe the conceptual design of a system. A complete set of these deliverables will provide a development team with an understanding of the desired information system. This set of information will guide the customer, software developer, review team, and management to comprehend system goals, background, and solutions to business problems.

Long-Term Scope—Methodology

The life cycle of a software product has been defined in many classical software engineering documents (see "References" on page 20). This report adapts the Sandia Software Guidelines to the IIS organization class of information systems. It is written not to replace those documents, but to position Sandia to take advantage of new technologies and tools. It also supports current three-layer or full three-tiered architecture and future distributed computing environments.

This document, Issue B, supersedes Issue A (Reference 20 on page 20), the initial effort to describe the software process. Issue A concentrated on the analysis phase. The analysis and design phases described here are not necessarily two separate processes. They are strongly coupled and may be completed concurrently. The steps require constant evaluation and iteration until the required result is achieved, i.e., the final evolutionary prototype and/or the critical deliverable set.

This work is on-going; it will continue to evolve in parallel with the dynamic process of applying new technologies within Integrated Information Services (IIS). We publish our efforts here as Issue B, to describe conceptual design and analysis within the context of the entire software life cycle. In this effort to achieve enterprise-level management of information assets, we will move toward information metamodels, rapid prototyping, and an integrated corporate information model.

Desired Future State

Automatic generation of application systems, based on precisely defined specifications, is the ultimate goal for future information systems. Full configuration management of all specifications at the enterprise level will support dynamic systems and ensure information timeliness, as well as data reliability.

Principles

We have incorporated the principles stated in the Information Architecture document (Reference 18 on page 20). In addition, a series of "core beliefs" or "guiding principles" contributed to the interaction of this team. These principles capture how the team attempted to understand and advance the software engineering process within the organization. Their inclusion below may contribute to the understanding of this document.

- We take most seriously this opportunity to improve the software development process (hereafter referred to as process).
- The process is our foundation for productivity, improvement, and measurement.
- The process is defined with adequate rigor to earn the confidence of customers.
- A repeatable, documented process will enable us to reduce variation in the quality of our software products.
- The process includes embedded validation and verification techniques, eventually eliminating the need for management review.
- Management leadership in the strategic planning process establishes priorities and fosters integration.
- Responsibility and accountability for conformance to specification for software rests with the software team.
- Each task and deliverable is assessed by the SILC team for the value added to the process.

Impact

This document represents a plan for moving the development of integrated information systems from our current reality to a desired future state. The process changes by which we attain our desired future state are only possible if management accounts for full life cycle software costs and takes responsibility and ownership for actively balancing initial delivery schedules with total software quality.

The perceived goals of the IIS organization are to:

- Build systems that serve the needs of the enterprise
- Speed the deployment of new applications or enhancements
- Improve traceability of the design specifications at a sharable enterprise-level

- Reduce overall costs throughout the life cycle (thus allowing for workforce redeployment or acquisition of additional work)

To further these goals, we provide a process that will shift costs and time from coding and nonproductive rework toward the analysis and design phases of the life cycle, addressed in this report. In addition, start-up costs are evident to define and build or acquire and populate a corporate metadata repository (Reference 18 on page 20). Overall, we expect total development costs to decline as requirements are gathered correctly and completely the first time.

The success of applying these processes and methodologies and thus attaining management's goals will be possible only if:

- Management enforces this life cycle process, or
- This life cycle process is enforced by a tool requiring changes be made only to the specifications.

Flexibility

The methodology and deliverable set presented here are designed as a core set to accomplish future enterprise-sharing and requirements traceability goals. The methodology and deliverables can and should be scaled for use in projects of any size.

A document set selection process is described in Reference 3 on page 20. In general, waiving production of any deliverables requires a management decision. For a software development project, the determination metric is based on weighted scores for size, development cost, customer and life cycle characteristics, platform, team experience, political risk, and special requirements from the customer or from the project environment.

Before the desired future state is reached, real problems will surface with loosely interpreted application of this process. Issue B of this document extends Issue A to include the actual development activities; Issue C will extend Issue B to discuss the tailoring of this process to the area of enhancements to existing systems, including commercial systems.

Deployment

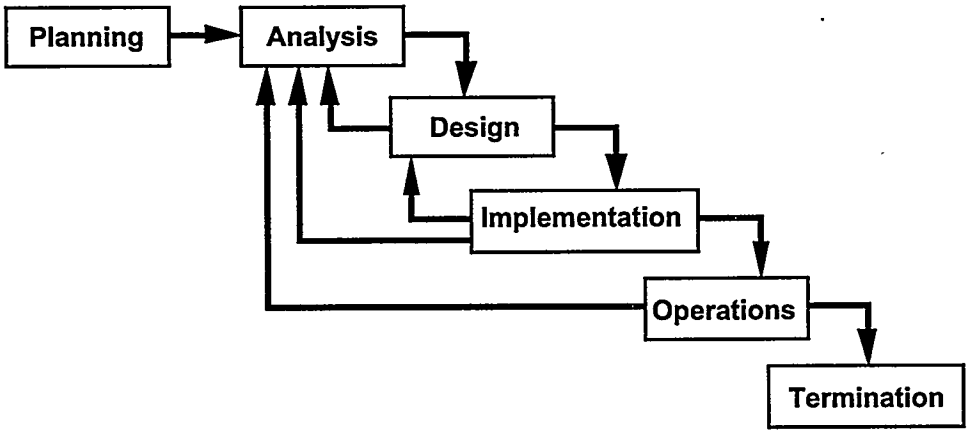
The software and information life cycle is normally described as a series of phases. Although we would prefer that these phases be simplified, we cannot imagine valid justification for skipping any activity or deliverable they prescribe. Overall, we feel that this methodology does not call for any more work than is already applied in current projects. It simply codifies and documents that work.

A compressed time-scale and improved quality result from a more consistent and defined methodology. The evolutionary prototype shortens development time and allows continuous improvement in understanding for both the customer and the software developer. Quality will improve through continuous iteration and refinement of requirements modeling and customer/subject expert evaluations.

The corporate repository for specifications allows the developer access to all models and future functions available for reuse and sharing at a level that is detailed and precise. Rapid application development should result from applying previously developed functions and framework.

A diagram of the software process using prototyping comes closest to describing the actual steps for development of new software:

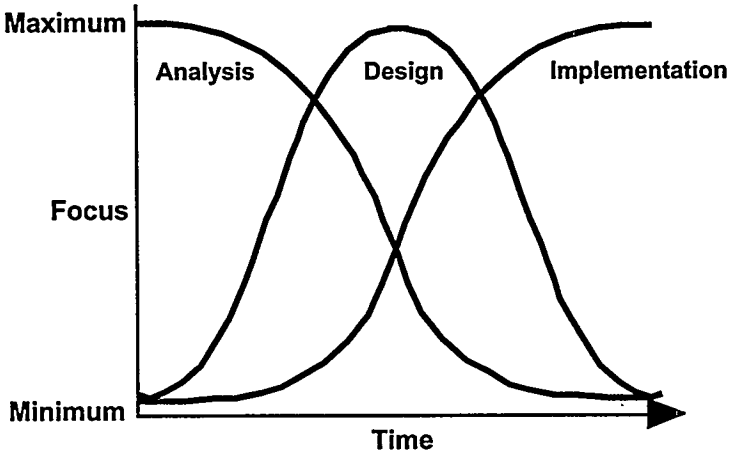
Figure 1 Software Process



The classic phases are described as steps to apply and deliverables to produce. During the production of these deliverables, evaluation is the key to the completion of that phase or to cycle through the steps again. Both designers and customers need constant feedback, which is the role of evaluation. A formal evaluation may be required to end each phase, depending on the level of risk associated with the software product and its development. The formal design review is recommended currently for all enterprise-wide software implemented by the IIS organization. Management of the IIS organization must choose to waive the formal design reviews.

Even though we discuss phase completion, phases overlap when considered as an iterative development architecture. Figure 2 shows how the phases of a complex software development interweave.

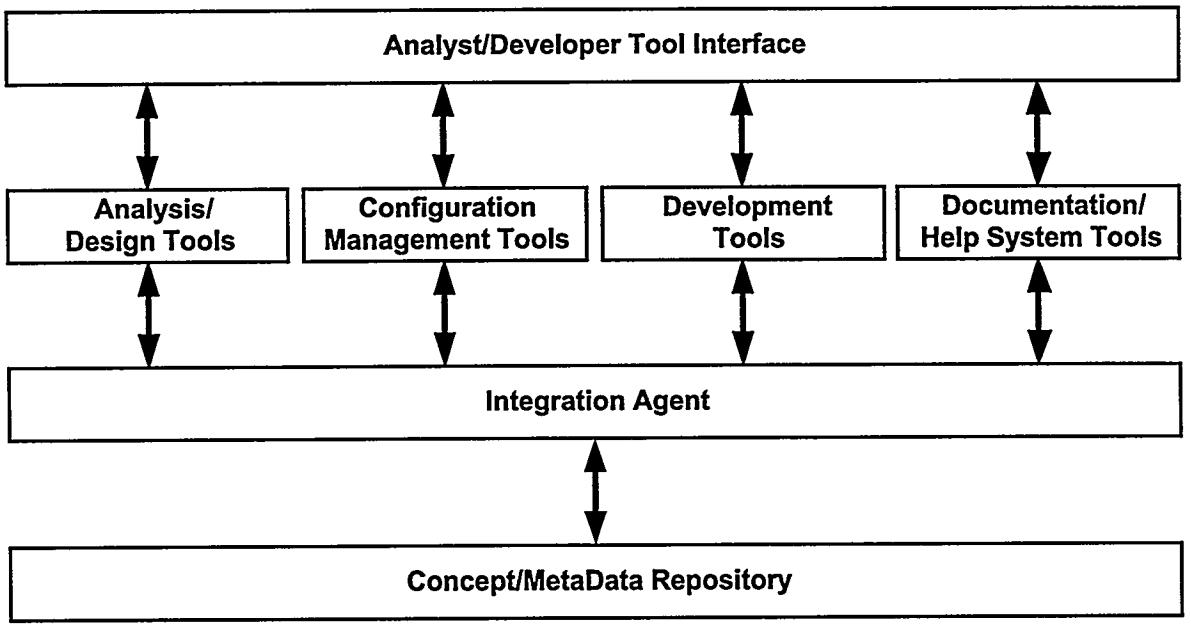
Figure 2 Software Phase Interweaving



(Adapted from Reference 22 on page 21)

Interfaces with the corporate repository are equally important as those with the customer, while requirements models are built and refined. The corporate repository is provided by the Data Administration Department as an integration framework for software engineering. This framework consists of the basic infrastructure (i.e., architectural and functional capabilities) to integrate resources, tools, and metadata about software. Figure 3 shows the conceptual architecture of an integration framework.

Figure 3 Conceptual Integration Framework



(Adapted from Reference 21 on page 21)

Software Phases

Deliverables

The activities described in the software phases lead to specific software development project deliverables. The following tables indicate the specification deliverable set (Table 1) and extended descriptions of these deliverables (Table 2). In Table 1, numbered items refer to their corresponding definitions in Table 2. The numbered items are covered in more detail in the example shown as Appendix A.

Table 1 Critical Specification Deliverable Set

Primary Beneficiary	Plan Set	Analysis Set	Physical Design Set	Implementation Set	Operational Engineering Set
Customer	• Information systems plan • Project proposal • Project plan	(1) Problem statement (2) Business rules (6) Presentation sets	(8) Extended presentation sets (including flows, controls, and screen templates)	• Data layer production database (16)User documentation	
Software engineering team	• Configuration management plan (including change control) • Corporate repository plan • Records management plan • Computer security plan • Corporate Style Guides	(3) Conceptual information model (4) Logical information model (5) Elementary processes	(7) CRUD matrix (9) DB schema (10)Extended dependencies & implementation constraints *(11)IDEF0 decomposition (12)Test plan, test database, and test data *(13)Detail process specifications and algorithms (may include design decisions) (14)External and internal interfaces (may include conversion map) (15)List of reused functions	• Implementation checklist completed • Presentation layer application • Functional layer components	
Operations team	• System-level requirements				• Operational specification
Review team	• Metric plan • Quality assurance plan	CDR "Conceptual Design Review" or "Preliminary Design Review"	DDR "Detailed Design Review" or "Critical Design Review"	(17)Test results FDR "Final Design Review" or "Pre-Production Review"	

* indicates optional items.

Table 2 Descriptions of Deliverables Produced by the Processes in Table 1

	Deliverable	Primary Communication Roles	Dependencies	Description
1	Problem statement	Customer : Analyst	(none)	Consists of: • Problem narrative—a textual deliverable that states the business problem in concise terms • Context diagram—may be included to show the scope of the system and which objects flow between the system and an external system interface. Large problems should be decomposed into subsystems that can be defined in two to three pages.
2	Business rules	Customer : Analyst	(1)	Formal natural language sentences with examples specify aspects of each object to be maintained in the information system. Initial storyboards or cartoons and flow diagrams are also used to document the business rules and verify them with the customer. General constraints and derivation rules are included in textual form.
3	Conceptual information model	Analyst : Designer	(2)	Constrained sentences in textual form and in a natural language model diagram communicate precisely the information required by the designer.
4	Logical information model	Analyst : Designer	(3)	The database structure (without vendor software or implementation considerations) is documented with a diagram of the form used in IDEF1X (a FIPS representation of an ER model).
5	Elementary processes	Analyst : Designer	(4, 2)	Elementary processes that represent the smallest units of work which can be executed to modify the content of the information base, and which will leave the information base in a state consistent with integrity rules.

Table 2 Descriptions of Deliverables Produced by the Processes in Table 1 (Continued)

	Deliverable	Primary Communication Roles	Dependencies	Description
6	Presentation sets	Customer : Analyst Analyst : Designer	(5, 2)	Consists of: • Process/Actor State Transition Matrix—adds rigorous definitions of the processes related to each object. In addition, from the customer's viewpoint, it helps integrate the actions performed and the state of the information system for each elementary process. • Completed storyboard diagrams with annotated sets denoting groups of data items to be treated as a single unit (as in elementary processes). Standard templates, when available, will be applied to provide a consistent "look and feel" with other systems. • Event and process lists define the system behaviors as related to the storyboards.
7	CRUD matrix	Designer : Implementer	(4, 5, 6)	The CRUD (Create, Read, Update, Delete) matrix identifies each elementary process used on each core object of the information system. This assists in verifying that each object is supported by a create, read, update, and delete process—it also helps identify redundancy of processes for an object.
8	Extended presentation sets	Customer: Designer Designer: Implementer	(6)	Storyboard diagrams that include vendor-specific and implementation-specific features, as well as application of standard template features for the enterprise systems.
9	DB schema	Designer : Implementer	(4)	The Physical Database, using the constructs of the vendor software and the additional information needed by the implementer of the system, is documented with the database schema diagram in the form of IDEF1X. Optionally, the SQL listings may be used.
10	Extended dependencies and implementation constraints	Designer : Implementer	(6, 2)	A formal definition of precedence required in the behavior steps.

Table 2 Descriptions of Deliverables Produced by the Processes in Table 1 (Continued)

Deliverable	Primary Communication Roles	Depen-dencies	Description
11 IDEF0 decomposition	Customer : Analyst Analyst : Designer	(2, 5, 7)	IDEF0 decomposition is a formal, hierarchical breakdown of the functions or processes that the software will provide. In addition to showing these processes in progressively more detail, these graphical charts also emphasize the inputs, which are transferred by the processes, and the outputs that result from the processes. They also show inputs that are not transformed, but are controls on the processes, and the "mechanisms" that are involved or used by the processes.
12 Test plan, test database, and test data	Designer : Implementer	(2, 9, 10)	Test data may be acquired in the modeling phase as valid and invalid examples of each fact. The design of the test plan will validate the modeling examples, in addition to other tests as needed to check the algorithms and the output or results of the processes and the system. The test database will be a significant data set to establish and carry out the test plan. It must be a reusable baseline database, and it should be established early in the prototyping process.
13 Detail process specifications and algorithms (may include design decisions)	Designer : Implementer	(8, 10)	Processes, other than elementary processes and functions that can be generated from the models, should be documented to the level of detail that would eventually assist in the generation of code. Several formats will suffice, until an automatic code generation tool is chosen or written. (See Appendix A for formats.)
14 External and internal interfaces (may include conversion map)	Designer : Implementer	(5, 9)	Most systems rely on data and controls from other systems, either by direct network links or file transfer. The interfaces must be documented to the level of detail required for future maintenance of the software. If the data must be converted for this application, a conversion map will show the input and transformation of each field, and the mapping to the new software.
15 List of reused functions	Designer : Implementer	(10, 13)	Sharable routines such as functions and methods will be documented. Those existing functions found in the repository will be listed, along with new sharable routines that are additions to the repository.

Table 2 Descriptions of Deliverables Produced by the Processes in Table 1 (Continued)

Deliverable	Primary Communication Roles	Dependencies	Description
16 User documentation	Implementer : Customer	(3, 6, 8, 10, 17)	At a minimum, the documentation set includes online help and a printed quick reference card. A writer will manage production of the materials, working with the application developer and the customer.
17 Test results	Implementer : Designer Designer : Customer	(12)	Following the test plan, a tester documents results in a standard template.This would include the results from testing complex functions and algorithms.

Planning Phase

Enterprise-Wide Business Planning

Enterprise-wide business planning and information modeling are the preferred front-end to all information system development.

Sometimes called Information Strategic Planning (ISP), this process identifies the state of and the needs for data, process, and organizational alignment. Strategic information planning for business systems within the corporate information model includes prioritizing, sequencing, and scoping of business areas. This planning provides the foundation for future decisions regarding the selection of business areas for development, redevelopment, and business process re-engineering. A stronger emphasis on strategic and integrated planning is advocated as the number of systems, platforms, and technologies increase.

Planning Phase Steps—Corporate Level

Members of top-level management participate in corporate information systems planning to develop critical success factors for the business:

1. The corporate-level planners create association matrices that correlate organizations, data, and processes.
2. The participants identify and resolve matrix inconsistencies.
3. The participants produce a prioritized list of information development efforts. As process owners they often propose a set of solutions, from which one is chosen to be studied and developed into a system proposal by software engineers.
4. The participants produce an organizational structure established on information flows and usage.

Planning Phase Steps—Project Level

This life cycle phase involves the conceptual development of a computer solution to a business need. The planning phase may be very short if the project is described in an overall business strategic plan. However, often the project is dynamically created as an evolution of business changes in reaction to changing circumstances. Many projects result

from a formal change control system, where changes and enhancements require extensive work.

1. The information system analysts develop a project proposal. They begin to scope the problem, assess the current systems, and describe a set of steps with estimates of costs and schedules for each. Once approved, the recommended solution is documented in procedures and policy documents as the business rules for the enterprise. The IIS is requested to build or acquire a system to support these rules.
2. The information system analysts write a project plan. A complete plan includes the goals, objectives, deliverables, milestones, estimates of resources, and schedules. They consider the following mechanisms:
   - Configuration Management Architecture (includes change control)
   - Quality Assurance Architecture
   - Records Management Architecture
   - Corporate Repository Architecture (includes reuse procedures)
3. The software engineers analyze and document system-level requirements and constraints, including:
   - Overview of product perspective and functions overview
   - Description of future state
   - Intended users and usage scenarios
   - System general constraints—limits on design decisions
   - External interfaces—hardware and software plans, communications plans
   - References to standards and guidelines that apply
   - Security constraints/data sensitivity/protections—an analysis to determine the highest level of data protection needed for the system, and the expected granularity of protections required for the data
   - References for risk management plan
4. The software engineers develop a quality assurance plan, which includes:
   - References for standards, tools, technologies, and integration with Software Management Plan requirements (SLP 1011)
   - Metrics plan
   - References for configuration management plan (specifications, documents, code, media, supplies, data)
   - References for design decision documents and audit/review results
   - References for records management plan (collection, maintenance, retention, archiving, deletion)

Analysis Phase

Analysts perform this phase by working directly with the customer, precisely defining the information facts managed by this information system (application). The preferred methodology for standard specifications is one of those presented in the Sandia Software Guidelines (Reference 6 on page 20). This methodology has been further developed to support modeling processes and events, in addition to information modeling. Because the

natural language of both customer and analyst is used, communication is greatly enhanced.

The quality of a rigorous analysis phase produces the desired specifications to be managed at the enterprise level. Analysts will use a version-controlled repository to manage all model constructs. This enterprise-level tool helps manage reusable information and functions for multiple models and systems. The repository will eventually support automatic generation of information systems and business rule descriptions.

Analysis Phase Steps

1. The analysts define a scope or "universe of discourse" to set boundaries on the problem. In addition, they develop a problem statement, including a problem narrative, context diagrams, and other descriptive materials.
2. The analysts query the repository for existing building blocks to support the software engineering reuse of model constructs. As new objects are created in the natural language models, they must be coordinated with the data administrator. The data administrator will add these objects to the repository if they are not already defined and used in other systems. Negotiations with the data administrator will be necessary for classifying new objects and functions, i.e., to leave them as local definitions (for this application) or to promote them to enterprise-shareable status.
3. The analysts create a natural language (English sentences) model from the building blocks selected from the enterprise model repository and from newly defined constructs resulting from analyst-customer reviews. This model rigorously defines the information, its meanings, identifiers, relationships, and constraints (especially mandatory and uniqueness requirements).
4. The analysts produce a natural language process model and graphical presentation model to define precisely the processes that will act on the information. (The model can include storyboards, known as presentation sets or prototypes.) Constraints and derivation rules are expressed textually and through concrete examples. Each analyst-customer review of the specifications should produce further refinements to the model as more areas of interest are discovered, and all business rules are brought out and explicitly documented, using a set of customer-provided examples.
5. The analysts repeats Steps 3 and 4 until the model completely specifies the problem scope and until the customer and process owner agree on its completeness. The customer or process owner and analysts accept this reviewed analysis set by formal sign-off.
6. The analysts schedule a conceptual review (see "Glossary of Terms" on page 23) to verify the specifications produced in this analysis of this application or subsystem. The review may also be attended by business experts and technology experts, including the developers. Other aspects of the system may be addressed in this review, such as training and special requirements.

Analysis Phase—Future Goals

An ongoing goal in the evolution of this methodology is to shift the traditional notion of testing earlier in the life cycle through the verification and validation of specifications. The examples collected during the analysis play a critical role in the validation process. At some future point, confidence in the rigor of the specification and development process

should reduce our reliance on classical unit, integration, system, and acceptance testing. Reasons for shifting this emphasis include:

- The growing sophistication and reliability of software analysis and design tools will assure that generated code performs in accordance to the specification from which the code was generated. We will only need to test that the code generated from a specification of ideal behavior actually does what is intended in the real computing system, interacting with hardware, the operating system components, and other applications.
- The traditional "acceptance test plan" can be derived with tool assistance and the validation completed from the English syntax used to capture requirements and examples.
- Presentation sets developed from prototyping tools serve as training and communication vehicles, establishing a foundation for eventual customer acceptance.
- By managing the design intent, traceability to business practices can be maintained for expected gains in business process improvement and business re-engineering.

Physical Design Phase

This step will use the specifications and extend them into a physical database schema and either an extended specification or an evolutionary prototype. The system designers should be knowledgeable developers with experience in building and using preferred technology information systems tools, as bounded by the standard architecture of the IIS systems. The designers should also have skills in human factors engineering as well as the classical programming training and skills.

The designers will specify the amount of flexibility of design format regarding the presentation sets transmitted to the development team. Aiming for consistency and user-centered design, human factors guidelines will affect the screen design and planned system behaviors.

The designers will also consider implementation perspectives. Future presentation, functional, and data layer guidelines, along with assistance from the data administrator, will direct the choice of the components of the three-tier model. Choice of certain vendors' tools may impose physical constraints on the implementation. The design will address those issues that were not specified in the analysis phase.

The risks associated with the software product and its development affect the choice of which of the following optional documentation elements the team will produce and review:

- Decomposition. IDEF0 is the demonstrated format in this example, as it is the Government standard for documenting process decomposition.
- Detail Description and Algorithms. Data derivations, known as derived facts

Physical Design Steps—Presentation Layer

1. The designers extend the presentation sets and screens, considering Human Computer Interaction (HCI) and following standard templates and HCI guidelines. In either hard copy form or software, this deliverable evolves into a prototype. Based on the information model that contains the information requirements, a prototype can

resemble the final product for data input as described by the elementary processes from the analysis phase. The designers produce the prototype using templates, reusable processes, and the latest technology tools.

2. The designers document flows and controls (which may also be in the functional layer) and include a list of edits. These edits control the occurrence of triggering events. As presentation elements, they indicate points of user input. Here the designers also include guidelines for choosing pick lists and checklists over user input, based on known factors such as data changeability and volume.

Physical Design Steps—Functional Layer

1. The designers and data administrator identify all reusable facts and functions.
2. The designers describe the physical design. External and internal critical interfaces include:
   - (External) control data from other sources, working data, and time-dependent triggers
   - (Internal) data and message passing in the system

Physical Design Steps—Data Layer

1. The designers plan for the production of the physical schema by proposing extensions to the logical schema. Here the designers:
   - Add the surrogate (generated) keys
   - Denormalize selected tables to enhance performance and ease of use
   - Apply estimates and statistics, which indicate where additional alternate keys should be placed and indexes used
   - Add protection and controls by examining data sensitivity and adding row level protections where necessary
   - Design access paths, which may include more indexes, key placement in tables, etc.
   - Add and document additional requirements to the schema, for such functions as tracing or archiving—these should be developed from functional processes rather than information requirements at this phase
2. The designers develop test data from examples the analysts stored in the repository and they map the examples to database tables. Future plans call for the development of a utility to automate this function.
3. The designers and customer formulate a conversion map for external interfaces. The map will be used to populate the database with existing legacy data or any continuing external controls and output formats to interface with other systems.

Physical Design Steps—Completion

1. The designers document and cross-reference all design decisions and the processes affected by them.
2. The designers add the physical model and general constraints (other rules and information, not contained in the model) to the repository.
3. The designers hold a Detailed Design Review to verify the physical design.

Implementation Phase

The implementation of a system requires skilled developers and the coordination of many interfaces to produce a new system and to put it into production. In the future, as system code is automatically generated, the coordination of interfacing workgroups will take on even more importance to:

- Complete the system
- Test the system
- Set up such components as networks, security plans, training, and production procedures

Implementation Phase Steps

1. The database administrator creates the physical database, distributed location and features, and access requirements.
2. The database administrator provides a test database to the customer and the implementers. Test data is created using the test data map provided by the designers.
3. The implementers build the presentation layer components according to the current software and hardware architecture and standards of the IIS systems. Where possible, the functional layer programs will be generated directly from the specifications.
4. The implementers build the functional layer components not already available in the corporate repository, according to the current software/hardware architecture and standards of the IIS systems. Where possible, the code will be generated directly from the specifications.
5. The implementers perform integrated testing against the test database.
6. The implementers perform system testing on the corporate test bed system.
7. The implementers hold an as-built design review for the pre-production application.
8. The implementers perform the steps of the deployment process to move the system into a fully operational production system.

Implementation Steps—Deployment

This process or sub-phase includes verification, training, and production definition to move the system into a production phase where the application will be available as a corporate asset. The deployment coordinator should possess the skills necessary to manage enormous amounts of data and place the required modules and components in a configuration management system. The deployment manager must also be able to perform multiple tasks and manage coordination of schedules of many customer and IIS groups.

1. The analysts and production specialist define the processing duplication, i.e., parallel runs for validation and verification requirements.
2. The analysts and customer populate the production database with the static data and legacy data.
3. The network administrator and production specialist connect the networks to access related and dependent data from other systems.

4. The analysts, customer, and training services develop the training classes and train users, as well as production support staff and help desk personnel. This team also develops user documentation.,
5. The production specialist and analysts develop operational procedures, including company asset protection processes and plans.
6. The analysts and production specialist perform a customer acceptance review of the Final Design Review, with an official sign-off for the system.

Information Systems Implementation Checklist

Table 3 on page 17 indicates the checklist as currently defined. The checklist is also available on the World Wide Web:
http://www_irn.sandia.gov/organization/div13000/isiproc/isistart.htm)

The checklist is a standard that incorporates required sign-offs necessary for moving new systems into production and enhancing existing ones within the Integrated Information Services Organization. This process ensures continued support without dependence on the original developer. It is used by developers and the production representative to identify stakeholders who need to be contacted as products are developed. The upper portion of the checklist (above the double line) contains early life cycle considerations. Items in the lower portion of the checklist (below the double line) are coordinated before the product is released.

The Production Center Implementation Coordinator is the single point-of-contact for questions about this process. In addition, this Coordinator will serve as or provide a production representative as a full member of the project team. This representative will:

- Coordinate with all production support stakeholders who are not project team members
- Ensure that production requirements are included in the project plan with reasonable milestone schedules and ensure that these milestones are met (including responsibility for writing some of the required documents)
- Write the service-level agreements documenting operational and performance requirements

Table 3 Information Systems Implementation Checklist

Consult with End Customer about ...	Consult with System Administration about...	Consult with Application Support about...	Consult with Technical Test Bed about...	Consult with Data and Database Administration about...	Consult with Network Communications, Security, and Production Services about...
❑ application requirements ❑ performance requirements ❑ access control	❑ server software requirements ❑ server hardware requirements	❑ transition plan	❑ workstation and PC installation and execution requirements ❑ architectural compliance including where applications reside and execute ❑ system design reviews	❑ Data Administration data modeling requirements ❑ Data requirements including backup and recovery, security, conversion, volumes and growth, source/target and archive/delete	❑ Production Services documentation requirements ❑ Security data security classifications and security plans ❑ Network Communications traffic tolerances and network needs
❑ early life cycle aspects above ❑ customer acceptance and verification testing	❑ early life cycle aspects above ❑ backup and recovery requirements ❑ scheduling and compatibility of system software installs and upgrades (software integration) ❑ stress test	❑ early life cycle aspect above ❑ change control procedures ❑ negotiated level of support ❑ hardware and software training for supporting staff	❑ early life cycle aspects above ❑ product executes on standard platforms ❑ software distribution needs	❑ early life cycle aspects above ❑ conformance to database standards ❑ database performance considerations ❑ database implementation	❑ early life cycle aspects above ❑ Production Services service level agreements ❑ Production Services Help Desk needs ❑ Production Services final sign-off of completed checklist

Operation Phase

The operation phase, traditionally called the production phase and maintenance phase, will be viewed as static, operational processes until the business rules change. This phase requires quality management techniques to assure the integrity and protection of valuable corporate assets and to maintain customer delight during day-to-day business operations. This phase requires extensive configuration management tools and an intensive commitment to the maintenance of the specifications and constructs in the corporate repository.

An application in the operation phase requires the entire software life cycle process to implement new requirements. To the extent required, all procedures will be performed as outlined in the life cycle phases: Planning, Analysis, Physical Design, and Implementation.

Operation Phase Steps

1. The analysts and designers determine changes to the system specifications.
2. The implementers generate changes in operational software from changes to the new system specifications. Initially coding will progress with current tools, eventually progressing to automatic generation of the software application code.
3. The analysts and designers are responsible for the documentation of new business rules resulting from the system changes. Initially these documents will be manually-generated, with text generation development eventually progressing to the point of automatic generation of the information system specifications and automatic generation of the business-rule documents.

Termination Phase

The system will be closed out after it is no longer needed.

Termination Phase Steps

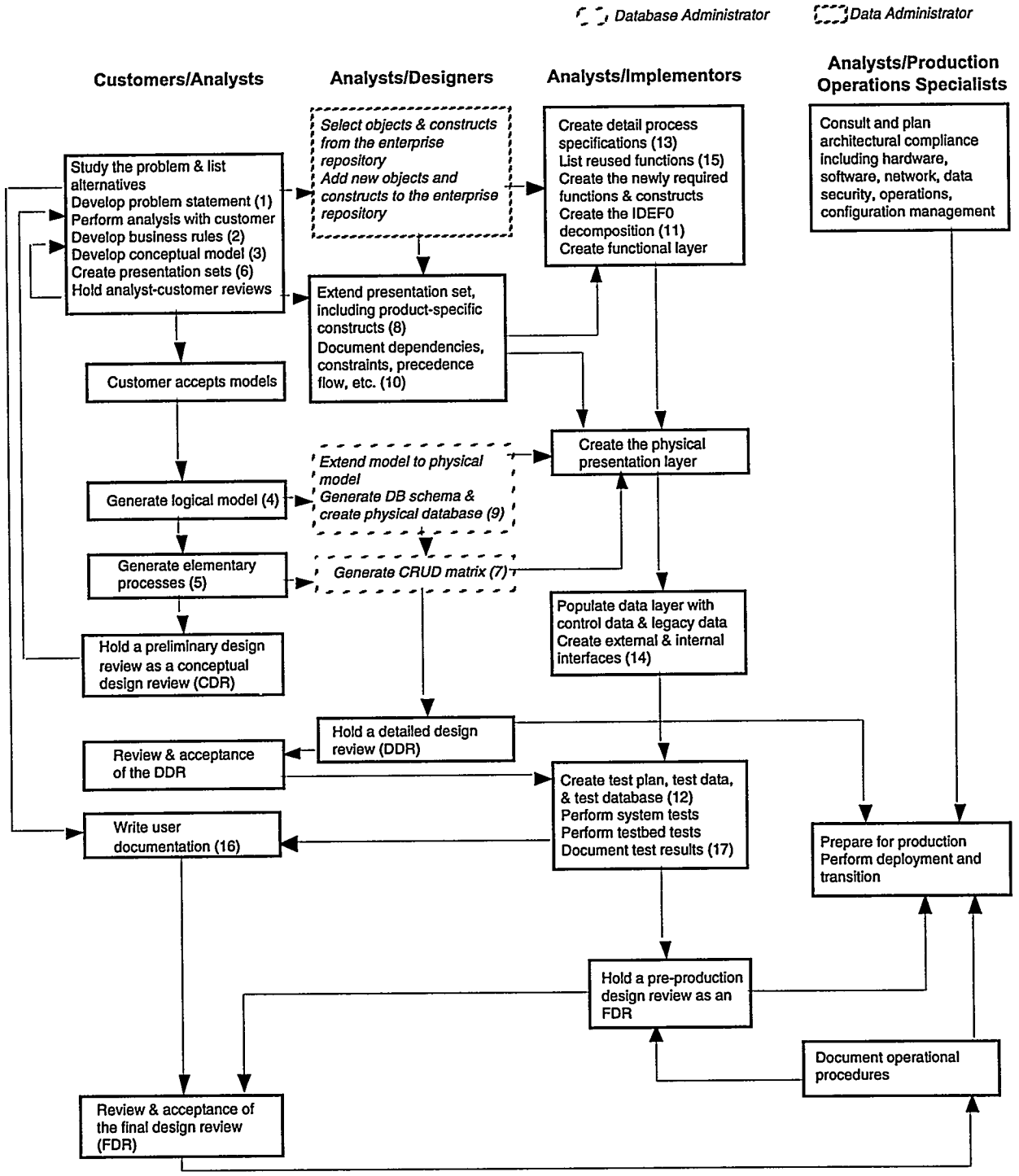
1. The production specialist moves any legacy data with continuing requirements to other systems.
2. Corporate Records Management evaluates the transfer of specifications and historical information to be retained for proper disposition and thereafter saved and managed by that organization.
3. The data administrator removes from the corporate repository constructs and process specifications that are no longer required for this or other systems.

Summary

The software process we have described will be used, evaluated, and modified in a continuing process improvement cycle. As technology changes and tools and architecture become more stable, this life cycle process will be modified to incorporate current best practices.

"...Good systems are not developed by only using the correct methods. A system is developed by people who are motivated, willing, and able to use the appropriate technology," wrote Fleischmann (Reference 23 on page 21). To manage the project, a strong leader is vital, as well as different team structures based on the right team mix of technical and social skills and the variation in work packages for each unique software project. "Program development is done by people; there they are the most important resource," noted DeMarco (Reference 24 on page 21). With the addition of the balanced mix of team members and methods and processes, plus supportive management and toolsets, the software and information life cycle may be summarized graphically as in Figure 4.

Figure 4 Roles and Responsibilities



References

1. IEEE Software Engineering Standards, Third Edition, IEEE, 1989.
2. Sandia Software Guidelines, Volume 1, Software Quality Planning, SAND85-2344, August 1987.
3. Sandia Software Guidelines, Volume 2, Documentation, SAND85-2345, in press.
4. Sandia Software Guidelines, Volume 3, Standards, Practices, and Conventions, SAND85-2346, July 1986.
5. Sandia Software Guidelines, Volume 4, Configuration Management, SAND85-2347, June 1992.
6. Sandia Software Guidelines, Volume 5, Tools, Techniques, and Methodologies, SAND85-2348, July 1989.
7. Software Development Process Methodology, Sandia, June 23, 1993.
8. SLP 1011, Software Management, April 1993.
9. SLP 2815, Preferred Processes, December 1992.
10. Computer Software Management, DOE Order 1330.1D, Office of Information Resources Management, U.S. Department of Energy, Washington, D.C., May 1992.
11. Becker, Steve D. Annotated Summary of the Information Model Design Procedure (IMDP), SAND94-0460, May 1994.
12. Nijssen, G.M. and Halpin, T.A. Conceptual Schema and Relational Database Design. Prentice Hall, 1989.
13. Nijssen, G.M. "The Essence of Universal Informatics." Class given in Albuquerque, New Mexico, July 1992.
14. Nijssen, G.M. and Twine, S.M. "A Summary of the Conceptual Schema Design Procedure (CSDP)." Nijssen Adviesbureau voor Informatica, 1990.
15. Nijssen, G.M. and Twine, S.M. "A Framework for Information System Development Methodologies." Nijssen Adviesbureau voor Informatica, 1990.
16. Organization 13316. "Department 13316 Home Page." Sandia Web, IRN. http://sass902d.itd.sandia.gov/Dept13316_Home.html
17. Organization 13316. "Natural-language Information Analysis Methodology (NIAM)." Sandia Web, IRN. http://sass902d.itd.sandia.gov/Niam/NiamIntro.html
18. Organization 13000. "Information Architecture." Sandia Web, IRN. http://www_irn.sandia.gov/organization/div13000/ctr 13200/dpt13202/ouia_94.htm
19. Myers, Glenford J. Art of Software Testing. John Wiley and Sons, 1979.
20. Software and Information Life Cycle (SILC) for the Integrated Information Services Organization; Analysis and Implementation Phase Adaptations of the Sandia Software Guidelines: Issue A. SAND95-1614, July 1995.

21. Wybolt, Nicholas. Frameworks for CASE Tool Integrations: IPSE Development. Proceedings of the Digital Equipment Computer Users Society, Spring 1990.

22. Booch, Grady. "Practical Software Engineering." Crosstalk, July 1995.

23. Fleischmann, Albert. Distributed Systems: Software Design and Implementation. Springer-Verlag, 1994.

24. DeMarco, T. and Lister, T. Peopleware: Productive Projects and Teams. Dorset House, 1979.

Acknowledgments

Contributors—Issue A and B Teams

Duane Garrison	Administrative Systems
Jim Stromberg	Administrative Systems
Andrea Cassidy	Administrative Systems
David Cuyler	Data Administration
John K. Sharp	Data Administration
Ronald Hall	Data Administration
Mary Lynn Clark	Electronic Commerce
Scott Joyce	Information Technologies
Irene Thurston	Information Technologies
Donna S. Eaton	Information Technologies
John Hatley	Information Technologies
Shelley Eaton	Information Technologies
Joe Schofield	Information Technologies
Gary E. Rivord	Information Technologies
Fay Tamashiro	Information Technologies
Greg Conrad	Information Technologies
Elisa Kephart	Information Technologies
Charlene Strickland	Information Technologies
Dwayne Knirk	Software Quality
Al Alvarado	Program Management
Carol Jones	Program Management
Mike Pendley	Test Bed System
Bill Mertens	Database Administration
T. C. Adams	Production Services
Nancy Marsh	The Production Process

Glossary of Terms

Table 4 Glossary

Term	Definition
4GL	Fourth Generation Language
CASE	Computer-Aided Software Engineering
Change management Configuration management	A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements
CIO	Corporate Information Officer
Conceptual Design Review	CDR; A review to verify the conceptual information model created through interactions with customers for completeness, logical integrity, and validity. Also discussed will be the traceable path from the conceptual model to targeted physical data stores, either existing or new. The conceptual model must be capable of being expressed with structured English sentences with concrete examples provided for validation. Fact by Fact mapping to screens, reports, and databases will be examined. Mappings of facts (data attributes) to middle layer functions will also be presented. Algorithms used to derive new facts will be reviewed, and targeted implementation strategies will be discussed. Proof of the customers' acceptance of the model must exist at this time.
Conceptual schema	A formal specification of the user's information requirements; a common world model that explicitly defines the set of meaningful sentences which can be communicated via machine, or stored within the machine
Constraint	Defines a restriction on the permitted population(s) of one or more populatable constructs in a conceptual schema
Context Diagram	The highest level data flow diagram, depicting the system's interfaces to other systems within the enterprise and its interfaces to the outside world; defines the scope and boundary for the system. It represents the system as a single circle, hiding all information about what occurs within the system. Each external entity is depicted as a single square. These squares, and the flows into and out of them, define the context within which the system must operate. Arrows drawn between the system and each external entity depict flows; they are labeled with the names of objects that flow into the system from the external entity and that flow out from the system to the external entity.

Table 4 Glossary (Continued)

Term	Definition
CRUD	Create, Read, Update, Delete
CSU	Customer Service Unit
Customer	The internal or external buyer or recipient of a product
DCE	Distributed Computing Environment
Detailed Design Review	DDR; To review the implementation information model. The implementation model may include implementation-specific facts (such as surrogate keys, internal objects, etc.) used to optimize physical implementation. Full compliance of the physical structures to the model specifications will be reviewed. Any constraints implemented in the schema will be tested. Specific use of data by the functional layer will be reviewed. Production support/operation documents should exist.
DOE	Department of Energy
ERD	Entity Relationship Diagram; A data model that describes attributes of entities and the relationships among them
Evolutionary prototype	A preliminary instance of a system, usually with limited interfaces and functionality, that serves as a starting point for continued development of a system until that system exists in its fully implemented form. In contrast to an exploratory prototype, an evolutionary prototype becomes an integral part of the delivered product.
Fact	A piece of information with specific meaning, communicated in a formal natural language sentence
Final Design Review	FDR; The final opportunity to review the deliverables prior to deployment in a production status. Assure that all test bed anomalies have been resolved. All issues of support (CSU and Production Operations) should be resolved.
FIPS	Federal Information Processing Standard
Functional diagrams	Graphic depictions of functional components of a system
ICAM	Integrated Computer-Aided Manufacturing; a definition language
IDEF	Integrated Computer-Aided Manufacturing **Def**inition; designated government standards, FIPS 183 and 184
IDEF0	Part of the IDEF set of standards; widely used method for graphically describing how a process works that uses functional decomposition to provide both an overall picture and a detailed view of the process

Table 4 Glossary (Continued)

Term	Definition
IDEF1X	Part of the IDEF set of standards; an entity-relationship diagramming standard
IEEE	Institute of Electrical and Electronics Engineers—an organization of engineering and electronics professionals; notable for developing standards relating to software engineering
IIS	Integrated Information Services
Implementation	The process of converting a software design into executable code
Methodology	A set of measurable core process definitions to be applied to software developed at Sandia
Metric	A quantitative measure to the degree to which a system, component, or process possesses a given attribute
NIAM	Natural-language Information Analysis Methodology
Project management	The processes by which projects are administered from conceptualization through maintenance and retirement
Pseudocode	A combination of programming language constructs and natural language used to express a computer program design
Quality assurance	• A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements • A set of activities designed to evaluate the process by which products are developed or manufactured
SDD	Software Design Description—A representation of software created to facilitate analysis, planning, implementation, and decision making; used for communication of software design information, and may be thought of as a blueprint or model of the system
SLP 1011	Sandia Laboratories Policy for Software Management
Software engineering	The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software
Software Management Plan	An auditable mechanism of each individual software group; describes the methods, tools, and techniques to acquire, develop and document, use, and support the software under the stewardship of that group. Development methodology documents and conduct of operations manuals are often the mechanisms used to document the Plan. The Software Quality Department distributes templates of the Plan.

Table 4 Glossary (Continued)

Term	Definition
Software Management Program	Provides guidance and assistance to Sandia groups in established policies, process analysis, and preparation of Software Management Plans; administered by the Sandia Software Quality Department in response to DOE Order 1330.1C requirements that Sandia properly manage its software assets.
SRS	Software Requirements Specification—Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces
Standards	Mandatory requirements employed and enforced to prescribe a disciplined and uniform approach to software development; Mandatory conventions and practices
Structure charts	Graphic depictions of the invocation control of modules which comprise a software system
Test database	An exact replica of a production database, created for testing purposes
Verification	The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase

Appendix A

Example of Deliverable Set to Complete the Analysis, Design, and Implementation Phases and to Use at the Design Reviews

This example from the Network Database Project includes Deliverables 1 through 17, to demonstrate the deliverable set (considered the minimum critical set) for the analysis, design, and implementation phases of a software development project.

We chose this example because it is a real system, currently in development. It will probably be implemented by the time the SILC document is published. As a teaching tool, it uses concepts familiar to all computer users at Sandia, namely User IDs and accounts on those computer systems for which access has been approved.

This network database contains extensive network and security management concepts, which have been left out of our example to simplify the content to a manageable size for both the reader and the authors. As shown in the Problem Statement and Context Diagram, the concepts demonstrated are all aspects of the customer and logins/ accounts, plus the report, "Reauthorization Forms."

Only customers who are Sandia employees are considered here. The entire Network Database specifications are available in the corporate repository. For information about access to the repository, contact the Data Administration Department or the Corporate Computer Help Desk at 845-2243. The browsable functions, currently in development, will be available to any Sandian through the internal Web.

Problem Statement

Deliverable 1

The problem statement contains a problem narrative, a textual deliverable that states the business problem in concise terms. A context diagram shows the scope of the system and which objects flow between the system and an external system interface. A high-level function diagram shows the decomposition of the system into major sub-processes. Large problems should be decomposed into subsystems that can be defined in two to three pages each.

Problem Statement—Purpose

The intent of a problem statement is to establish an initial and mutually acceptable understanding between the analyst and the customer(s) as to the nature and scope of the system. It establishes a boundary (fuzzy, to be sure) to the capabilities of the system. It gives the analyst a starting point for determining the major components of the system. It also launches the customer/analyst modeling sessions to clarify and verify all information and processes that the customer will require and agree to own.

The problem statement has value beyond the development period. It is a project resource that can and should be reused in socializing and training for application of the system; it is a good introduction for user manuals. Versions of this statement should be maintained to keep it current with the system "as built."

Content Guide

The problem statement should include:

- A high-level, simplified description of the major functions the system is to perform, its boundaries, and its interfaces
- Identification of users and operators, including their departmental (individual) business or operations
- Description of phasing if intended (large or complex systems may be developed in several phases).

Problem Narrative—Version 1

Deliverable 1

The Network Database is the framework for administering an enterprise-wide information system consisting of distributed computers and groups and multiple levels of networks. It covers these major functions:

- *Accounts*—Keeps track of "customers" authorized to access Sandia networks or any centrally managed services on those networks. In this sense a "customer" is anyone who has a legitimate need to access Sandia networks, whether they are Sandians or not, and whether they are U.S. citizens or not.
- *Networks*—Keeps track of Sandia's overall network configuration, including all of the logical LANs defined under the four major protocols "TCP/IP," "Appletalk," "DECNET," and "IPX," the placement of those logical LANs on physical LANs and networks, and all machine connections to those logical and physical LANs.
- *Security Plans*—Provides a means to create and manage machine security plans. Also, manages and create portions of the major security plans that cover physical LANs, networks, and multi-user machines.

These major functions, and the user groups that perform them, are described in more detail in Table 1. The functions describe the complete system, to be developed in two phases.

Note: The shaded functions represent the focus of this example, all pertaining to the accounts function.

Table 1 Network Database Functions by User Group

User Group	Function
Password Administration	Registration of customers
	Generation of Login Names and Unix user IDs
	Registration of corporate-level central services
	Setting up customers and entity accounts on central services
	Expiration and reauthorization of accounts
	Placement of customers in NWDB user groups
	Registration of entities
	Setting up customer membership in named Unix groups
	Registration of physical LANs and logical LANs
	Administration of control information concerning user groups, countries, sites, e-mail types, and classification ranks
NWDB Administrators (LAN Points of Contact)	Management of LAN connection information
	Management of basic machine information
	Registration of LAN-level central services
	Setting up customers and entity accounts on LAN central services

Table 1 Network Database Functions by User Group (Continued)

User Group	Function
	Verification of LAN points of contact
Customer Service Units	Maintenance of desktop configuration information
	Management of LAN connection information
	Management of basic machine information
	Generation of CSU charges
Central Service Administrators	Maintenance of central service accounts
	Administration of control information concerning Internet domains
	Feeding Sandia's Domain Name Servers
	Registration of Cryptographic Public Keys
Network Analysts	General network troubleshooting
Network Design Team	Management of LAN connection information
	Management of basic machine information
	Network troubleshooting
	ISN Control Files
Computer Security Administrators	Management of major security plans
	Verification of Security Officers
	Approval and release of machine security plans
	Registration of Computer Security Representatives
	Administration of control information concerning security plan approvers, security managers, protection methods, sensitivity categories, security officer titles, and major security plan categories
Computer Security Representatives	Preparation of machine security plans
	Maintenance of LAN connection information
	Maintenance of basic machine information
Physical Security Administrators	Registration of Vaults
E-Mail Support	Registration of E-Mail Post Offices
	Generation of Login Names for the LIS System
Network Database Administrators	General troubleshooting
	Administration of control information concerning networks, security environments, network interface types, tech control centers, protocols, TCP/IP network implementations, vendors, machine makes and types, operating systems, and network functions

The following functions have been chosen for detailed treatment to demonstrate the software life cycle:

Password Administration

Registration of Customers

People who are in the Human Resources file automatically have much of their data reflected into the Network Database, but certain data, such as country of citizenship, Sandia sponsor, and network access expiration date must be entered by hand even if the customer is in Human Resources. For people who are not in the Human Resources file, including "visitors" from throughout the world, Password Administration registers all data. There are currently 9,256 Sandians (8,527 regular and 729 non-regular, all of which are in Human Resources), 3,083 contractors (of which 2,214 are in Human Resources), and 487 visitors (none of which are in Human Resources), listed as active in the Network Database.

Generation of Login Names and Unix User IDs

The "Login Name" (such as "meaton" or "mjmurph") is now in use throughout Sandia to identify users on computerized systems, and is also a standard part of everyone's e-mail address. For accounts on Unix systems, Login Names are paired with numeric Unix User IDs, with different aspects of the Unix account controlled by one or the other. The Network Database contains an algorithm to generate unique Login Names and Unix User IDs (up to three pairs for each person or entity—one pair each for the unclassified, confidential, and secret classification levels). Generation of unclassified Login Names and Unix User IDs is done automatically for people in the Human Resources file when they become active employees. All other such generation is hand-initiated by Password Administration. There are currently 14,328 active pairs of Login Names and Unix User IDs in the Network Database.

Registration of corporate-level central services

"Corporate-level central services" are major computing resources and services to which access can be restricted by password, and for which it has been decided that Password Administration will formally control access.

When each service is registered, the highest classification rank allowed is recorded, whether or not it is available for access by named Unix groups, whether or not it requires file disposition for terminated accounts, whether or not it requires a case for account charges, and where it is located as far as what server machines, physical LANs, and/or networks on which it is available.

Setting up customer and entity accounts on central services

Password administration uses the Network Database to set up accounts on central services they manage before they give out passwords. (Passwords are generated outside the Network Database.)

The Network Database applies numerous rules to make sure that account data is complete and that a given customer's access to a service does not violate security considerations before an account may be activated.

If the service requires a case number for account charges, it is entered here and checked against the list of valid cases in HRIS.

There are currently 9,382 active corporate-level accounts.

Expiration and reauthorization of accounts

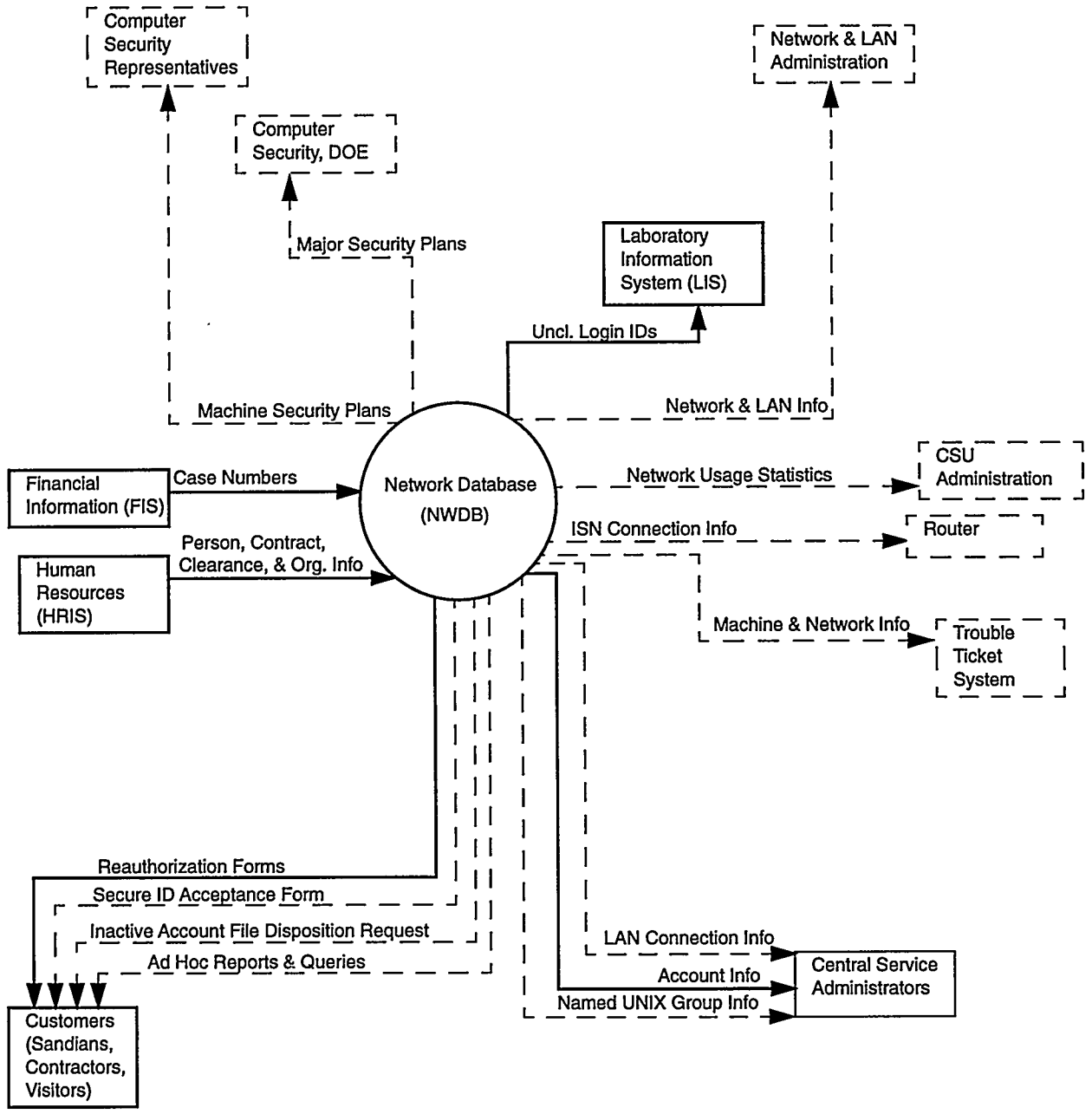
The Network Database determines the expiration date of accounts held by any customer or entity based on a number of factors such as the classification level of the accounts, a contractor's expiration date, a visitor's access expiration date, and so on. When it is time for a set of accounts belonging to a customer or entity to expire, the system will generate a Reauthorization Form, which is automatically delivered via e-mail to the account holder or entity owner if an entity. The customer can then print it out, get it signed, and return it to set the account expiration dates ahead.

Context Diagram

Deliverable 1

This shows the scope of the system and which objects flow between the system and an external system interface. Dashed lines indicate business actions not covered in this example set.

Figure 1 Context Diagram—Network Database Example

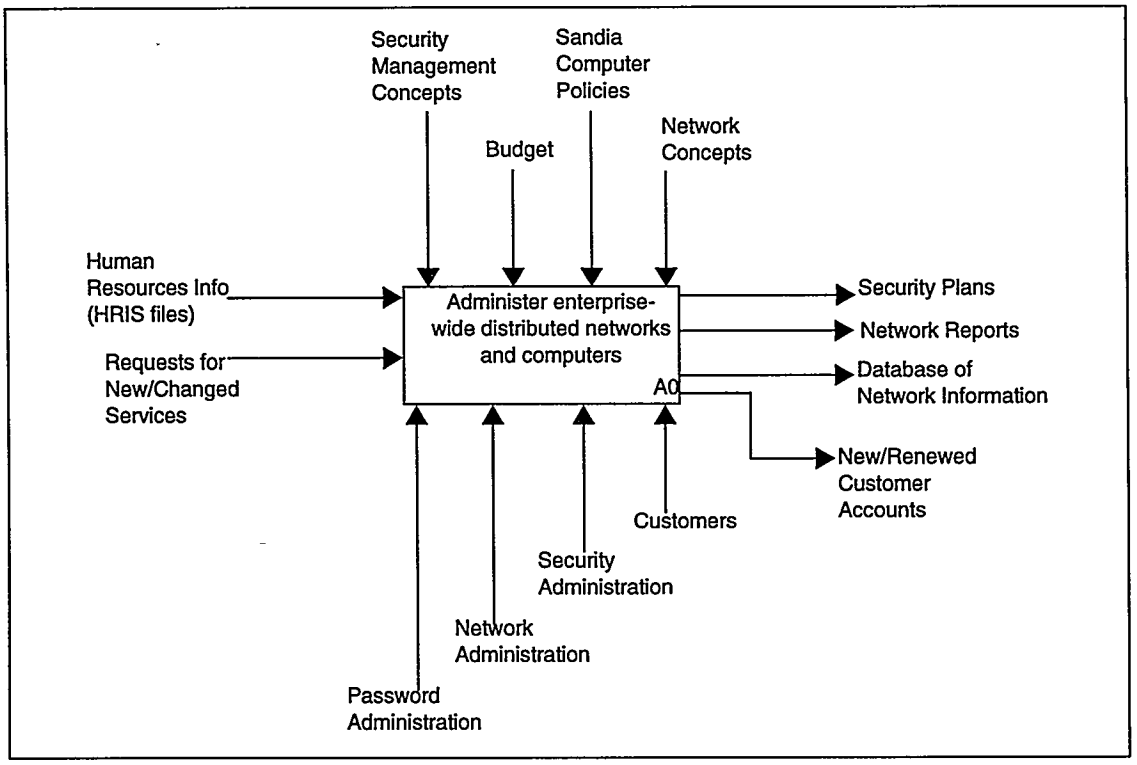


High-Level Function Diagram

Deliverable 1

This example shows the major functions of the NWDB. This depicts the central part of Figure 1 on page A-7.

Figure 2 Node A-0, Network Database Example



Reading from the top, clockwise:

- The arrows entering the top of the box containing the process description show "controls" on this process.
- The arrows exiting to the left show the outputs
- The bottom arrows show mechanisms, i.e., answers to who or what will act on this process
- Finally the arrows entering from the right show those inputs that will be affected by the processes and transformed to an output

Business Rules

Deliverable 2

Formal natural language sentences with examples specify aspects of each object to be maintained in the information system. These sentences, together with initial storyboards or cartoons and flow diagrams, are also used to document the business rules and verify them with the customer. The initial storyboards or cartoons and flow diagrams with process and event information (not shown here) are refined throughout the analysis and design processes into Deliverable 6, presentation sets with event and process lists (see "Presentation Sets" on page A-25). General constraints and derivation rules are included in textual form or as a reference to the process list.

The formal natural language sentences with examples require an object listing with definitive information about each object. The listing includes the object's label, along with examples.

Note: We prefer that a minimum of three examples be supplied. Where the Network Database team had only two examples, this generated list simply displays **<<a value>>**.

This example extracts specific information from the NWDB. To view the complete deliverable for this subset of the NWDB, see the World Wide Web:

http:// sass902d.itd.sandia.gov/SILCExam.pdf

Table 2 Object and Fact Type Listing, With Examples

Object Type	Example Sentence
Account	Central_Service <LIS> has access authorized for Login <<a value>>
	Central_Service <LIS> has access authorized for Login <jschavez>
	Central_Service <SOMNET> has access authorized for Login <dbblack>
Agent	agent_ID < [redacted] Customer> identifies a(n) Agent
	agent_ID <HR update> identifies a(n) Agent
	agent_ID <New IRN/NWDB Accts> identifies a(n) Agent
Case	Case_Nbr <33214000000> identifies a(n) Case
	Case_Nbr <89991110000> identifies a(n) Case
	Case_Nbr <<a value>> identifies a(n) Case
Central_Service	Short_Name <<a value>> identifies a(n) Central_Service
	Short_Name <LIS> identifies a(n) Central_Service
	Short_Name <SOMNET> identifies a(n) Central_Service
Clearance	Clearance_type <L> identifies a(n) Clearance
	Clearance_type <Q> identifies a(n) Clearance
	Clearance_type <none> identifies a(n) Clearance
Customer	Customer_ID < [redacted] identifies a(n) Customer
	Customer_ID < [redacted] identifies a(n) Customer
	Customer_ID < [redacted] identifies a(n) Customer

Table 2 Object and Fact Type Listing, With Examples

Object Type	Example Sentence
Entity	Entity_ID <000031000> identifies a(n) Entity
	Entity_ID <000032007> identifies a(n) Entity
	Entity_ID <<a value>> identifies a(n) Entity
FT225	Account <LIS,dbblack> had status last changed by Agent_Type <<a value>>
FT225	Account <LIS,mfjohnson> had status last changed by Agent_Type <process>
FT225	Account <SOMNET,jschavez> had status last changed by Agent_Type <<a value>>
FT226	Account <LIS,dbblack> charges by default to Case <<a value>>
FT226	Account <LIS,mfjohnson> charges by default to Case <33214000000>
FT226	Account <SOMNET,jschavez> charges by default to Case <<a value>>
FT227	Account <LIS,dbblack> had its status last changed by Agent <<a value>>
FT227	Account <LIS,mfjohnson> had its status last changed by Agent <New IRN/NWDB Accts>
FT227	Account <SOMNET,jschavez> had its status last changed by Agent <[redacted] Customer>
FT228	Account <LIS,dbblack> had its status last changed on Date_Time <<a value>>
FT228	Account <LIS,mfjohnson> had its status last changed on Date_Time <198812110545>
FT228	Account <SOMNET,jschavez> had its status last changed on Date_Time <<a value>>
FT229	Account <LIS,dbblack> is accepted on Date <<a value>>
FT229	Account <LIS,mfjohnson> is accepted on Date <19941521>
FT229	Account <SOMNET,jschavez> is accepted on Date <19931203>
FT230	Account <LIS,dbblack> is marked with Status <<a value>>
FT230	Account <LIS,mfjohnson> is marked with Status <inactive>
FT230	Account <SOMNET,jschavez> is marked with Status <<a value>>
FT231	Customer <286449007> had status last changed by Agent <<a value>>
FT231	Customer <399886113> had status last changed by Agent <New IRN/NWDB Accts>
FT231	Customer <576447650> had status last changed by Agent <[redacted] Customer>
FT232	Customer <286449007> had status last changed by Agent_Type <<a value>>
FT232	Customer <399886113> had status last changed by Agent_Type <process>
FT232	Customer <576447650> had status last changed by Agent_Type <<a value>>

Conceptual Information Model

Deliverable 3

The primary components of Deliverable 3 are the constrained sentences in textual form and in a natural language model diagram. These communicate analysis results concisely to the designer.

"Information systems are systems which support the effective communication of meaningful information between human beings by design. If these information systems are to be truly effective, then the principles governing their construction and use must be derived from the principles of human communication and not from the principles of computerized data processing. Our most fundamental axiom is:

> Information systems are systems which support the communication of *linguistic* sentences (called facts) between humans.

"... in order to restrict ourselves to the effective communication of meaningful information via these information systems we must define a formal representation of the common world model to be shared by the computer and the human. This common world model is called a *conceptual schema.* For the human user, the conceptual schema explicitly defines the set of meaningful sentences which can be communicated via the machine (or, for that matter, can be stored within the machine). For the computer, the conceptual schema explicitly defines the permitted states of the database contents and the permitted transitions between those states." (Reference 14; see "References" on page 20).

Sentences and their constraints in textual form (facts) follow.

Constrained Sentences

Deliverable 3

Table 3 Objects—NWDB Example

	Object Type	Description
1.	Account	Nested: Central_Service has access authorized for Login
		Examples:[LIS,mfjohnson;SOMNET,jschavez;LIS,dbblack]
2.	Agent	Object is identified by label agent_ID+
	agent_ID+	AgentID is made up of either process name or customer ID plus agent type.
		Examples:[HR update;New IRN/NWDB Acct[redacted]
3.	Agent_Type	values { c, p }
		Examples:[customer;process]
4.	Building_Number	Examples:[892;T-56;859]
5.	Case	Object is identified by label Case_Nbr-
	Case_Nbr-	Examples:[89991110000;33214000000]
6.	Case_Status	Examples:[active;inactive]
7.	Central_Service	Object is identified by label Short_Name-
	Short_Name-	Examples:[LIS;SOMNET]
8.	Classification_Level	values { u, c, s }
		Examples:[unclassified;classified;secret]
9.	Clearance	Object is identified by label Clearance_type-
	Clearance_type-	values { Q, L, none }
		Examples:[Q;L;none]
10.	Customer	Object is identified by label Customer_ID+
	Customer_ID+	[redacted]
11.	Customer_ID	[redacted]
12.	Customer_Lookup_Name	Examples:[Johnson, Marie F.;Chavez, Jose S.;Black, Dan B.]
13.	Customer_Mail_Name	Examples:[Marie F. Johnson;Jose S. Chavez;Daniel B. Black]
14.	Date	Examples:[19950213;19941521;19931203]
15.	Date_Time	Examples:[199302211730;198812110545]
16.	Email_Address	Examples:[mfjohns@snl2800.sandia.gov;jschave@E-mail_print.cs.sandia.gov]
17.	Email_Post_Office_Name	Examples:[E-mail_print;SOMNET]
18.	Entity	Object is identified by label Entity_ID+
	Entity_ID+	Examples:[000031000;000032007]
19.	Flag	values { Y, N }
		Examples:[yes,no]
20.	Login	Object is identified by label Login_Name-
	Login_Name-	Examples:[mfjohnson;jschavez;dbblack]

Table 3 Objects—NWDB Example (Continued)

Object Type	Description
21. Login_Holder	Object is identified by label Login_Holder_ID+
Login_Holder_ID+	Examples:[3[redacted]
22. Login_Holder_Type	values { customer, entity }
	Examples:[customer;entity]
23. Mail_Stop_Number	Examples:[0661;0330;0933]
24. NWDB_User_Group	Object is identified by label User_Group_Title-
User_Group_Title-	Examples:[router;NWDB Administrator;Password Administrator;Computer Security Administrator]
25. Organization	Object is identified by label Org_Nbr-
Org_Nbr-	Examples:[13211;02811;04000]
26. Phone_Number	Examples:[5058443333;5102945555;5058452222]
27. Reference_Record	Object is identified by label Ref_Rcd_Key-
Ref_Rcd_Key-	Examples:[30035;30036;30037]
28. Room_Number	Examples:[M100;268;L400]
29. SecurID_Card_Serial_Nbr	Examples:[[redacted]
30. Site	Object is identified by label Site_code-
Site_code-	Examples:[SA,SC]
31. Site_Desc	Examles:[Sandia Labs, NM; Sandia Labs, CA]
32. Status	Object is identified by label status_code-
status_code-	values { a, i }
	Examples:[active;inactive]
33. Unix_User_ID	Examples:[12001;18003;30035]

Table 4 Facts—NWDB Example

Fact Type	Mandatory Object	Unique Object
1. Account had status last changed by Agent_Type	Account	Account
2. Account charges by default to Case	--	Account
3. Account had its status last changed by Agent	Account	Account
4. Account had its status last changed on Date_Time	Account	Account
5. Account is accepted on Date	--	Account
6. Account is marked with Status	Account	Account
7. Customer had status last changed by Agent	Customer	Customer
8. Customer changed by Agent_Type	Customer	Customer
9. Case is statused by Case_Status	Case	Case
10. Central_Service has access authorized for Login	--	Central_Service + Login
11. Customer can be reached at Phone_Number	--	Customer

Table 4 Facts—NWDB Example (Continued)

	Fact Type	Mandatory Object	Unique Object
12.	Customer holds Clearance	Customer	Customer
13.	Customer is identified by Customer_Lookup_Name	Customer	Customer
14.	Customer is identified by Customer_Mail_Name	Customer	Customer
15.	Account is originally set up by Customer	--	Account
16.	Customer receives electronic mail through Email_Post_Office_Name	--	Customer
17.	Customer receives faxes at Phone_Number	--	Customer
18.	Customer receives hard-copy mail at Mail_Stop_Number	--	Customer
19.	Customer works at Site	--	Customer
20.	Customer works for Organization	--	Customer
21.	Customer works in Building_Number	--	Customer
22.	Customer works in Room_Number	--	Customer
23.	Reference_Record has as last used for a customer a Customer_ID	Reference_Record	Reference_Record
24.	Account has its card expire on Date	--	Account
25.	Customer had status last changed on Date_Time	Customer	Customer
26.	Account has file disposition made Flag	--	Account
27.	Customer holds a login as flagged by Flag	Customer	Customer
28.	Entity flagged as holding a login by Flag	Entity	Entity
29.	Customer has registered in hr system Flag	Customer	Customer
30.	Login changed by Agent_Type	Login	Login
31.	Login had its status last changed by Agent	Login	Login
32.	Login had its status last changed on Date_Time	Login	Login
33.	Login has its accounts expire on Date	--	Login
34.	Login holds Login_Holder	Login	Login
35.	Login is given Unix_User_ID	Login	Login
		--	Unix_User_ID
36.	Login is held in level Classification_Level	Login	Login
37.	Login is marked with Status	Login	Login
38.	Login is typed by Login_Holder_Type	Login	Login
39.	NWDB_User_Group can be contacted in ca through Mail_Stop_Number	--	NWDB_User_Group
40.	NWDB_User_Group can be contacted in nm through Mail_Stop_Number	--	NWDB_User_Group
41.	NWDB_User_Group contains Customer	--	NWDB_User_Group + Customer
42.	NWDB_User_Group is ca contact for Email_Address	--	NWDB_User_Group
43.	NWDB_User_Group is nm contact for Email_Address	--	NWDB_User_Group

Table 4 Facts—NWDB Example (Continued)

	Fact Type	Mandatory Object	Unique Object
44.	NWDB_User_Group can be contacted in ca through Organization	--	NWDB_User_Group
45.	NWDB_User_Group can be contacted in nm through Organization	--	NWDB_User_Group
46.	NWDB_User_Group can be contacted in ca through Phone_Number	--	NWDB_User_Group
47.	NWDB_User_Group can be faxed to in ca at Phone_Number	--	NWDB_User_Group
48.	NWDB_User_Group can be contacted in nm through Phone_Number	--	NWDB_User_Group
49.	NWDB_User_Group can be faxed to in nm at Phone_Number	--	NWDB_User_Group
50.	SecurID_Card_Serial_Nbr is authorized by Account	--	SecurID_Card_Serial_Nbr
		--	Account
51.	Site is described by Site_Desc	Site	Site
52.	Customer is marked with Status	Customer	Customer
53.	Reference_Record has as last used for a customer a Unix_User_ID	Reference_Record	Reference_Record
54.	Reference_Record has as last used for an entity a Unix_User_ID	Reference_Record	Reference_Record

Natural Language Model Diagram

Deliverable 3

We show two pages of Natural-language Information Analysis Methodology (NIAM) models, which is one form of a conceptual model in diagram form. This diagram provides the designer with another view that demonstrates the data relationships and data roles.

Notes: * denotes a "derived fact."

The **FTxx** numbers shown in the diagram reference those defined in the formal natural language sentences in Deliverable 2.

Customers

Figure 3 NIAM Model—Customers

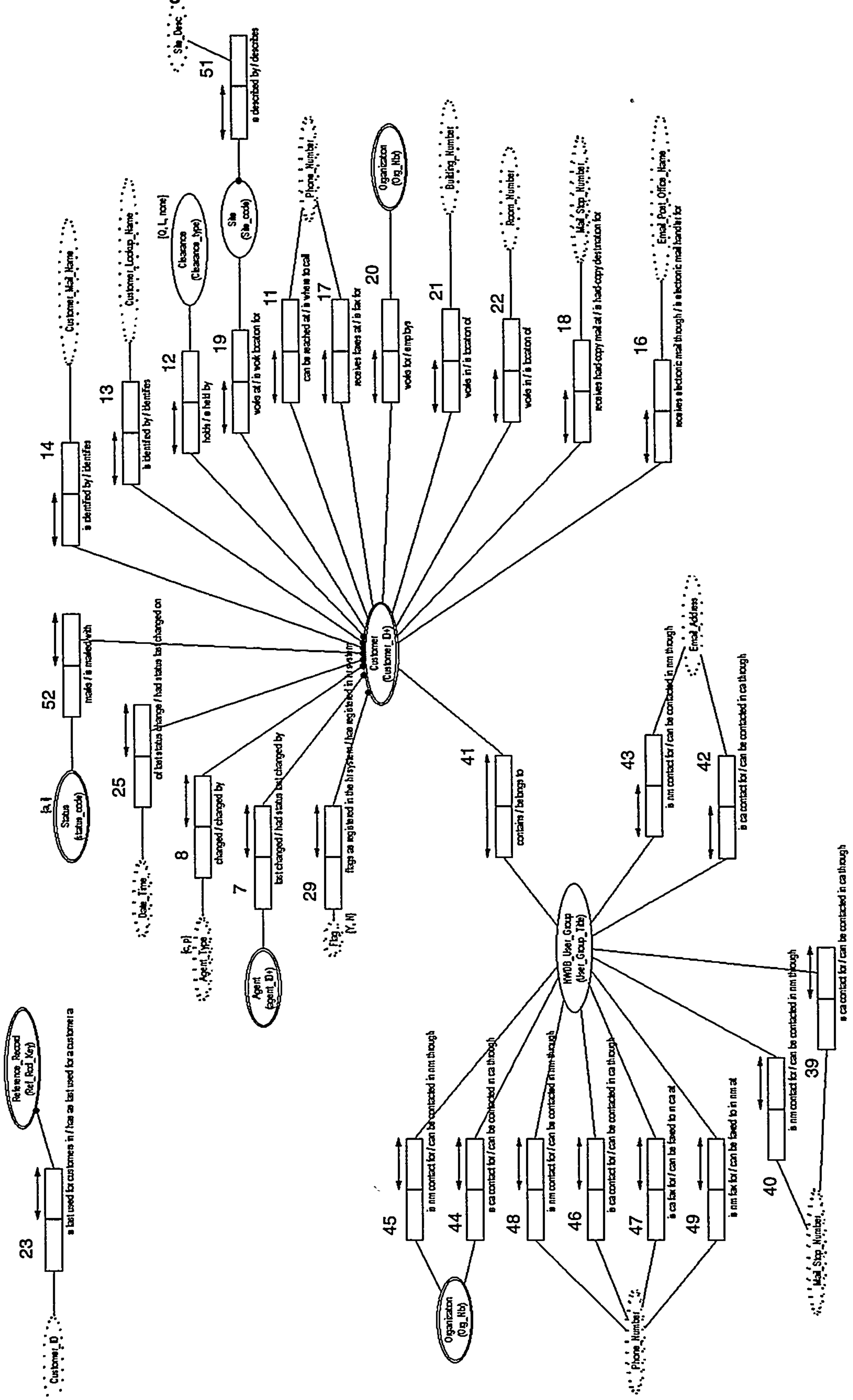
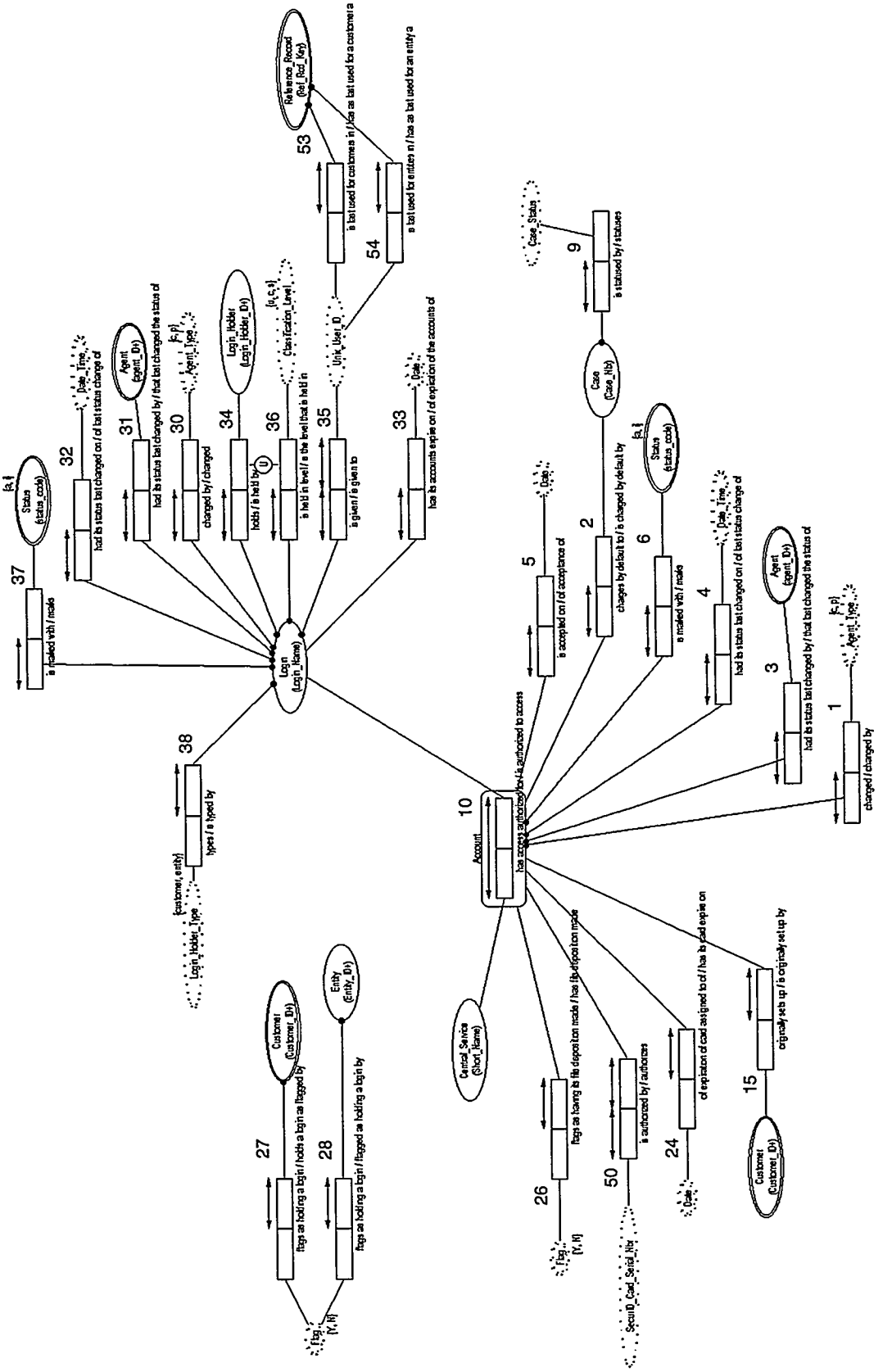


Figure 4 NIAM Model—Logins/Accounts

Logins/Accounts



Logical Information Model—IDEF1X Diagram

Deliverable 4

The logical database structure (without vendor software or implementation considerations) is documented with a diagram of the form used in IDEF1X (a FIPS representation of an ER model). This diagram shows the objects and attributes with foreign key relationships.

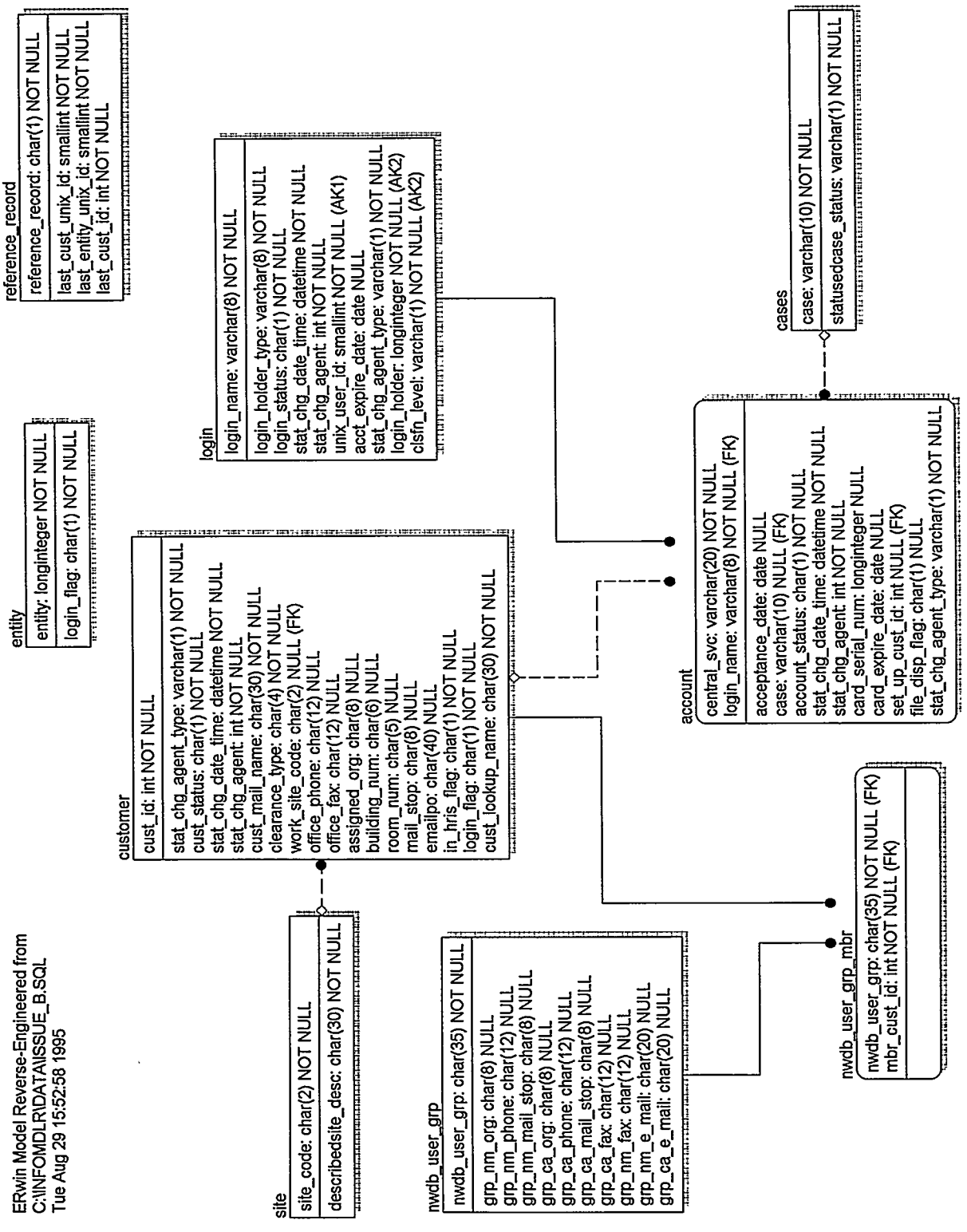
The IDEF1X format shown here is a standard chosen for government agencies.

Note: Keys are listed first in each table with a line following the last item in the key. Data items which uniquely identify an object are usually the primary key. The following describes the symbols used in an IDEF1X diagram.

PKxx	Primary key; key that uniquely identifies one object from another
AKxx	Alternate key; a secondary identifier
FKxx	Foreign key; key of another entity when it migrates into a given entity as a result of a relationship between the two entities
Parent	An entity on the 1-end of a 1-to-many relationship
Child	An entity on the many end of a 1-to-many relationship
———	Identifying relationships (foreign key is part of primary key)
- — -	Non-identifying relationships (foreign key is not part of primary key)

Cardinality	The quantifier in a relationship between two entities; a quantifier answers the question, "How many?"
——P● or - - P●	Existence of at least one matching child required if parent exists (1: 1 or more)
——Z● or - - Z●	Existence of maximum of one child matching parent (1: 0 or 1)
———● or - —— ●	1: 0 or more
- — ◇	Parent-child relationship where the parent entity is optional (i.e., optional non-identifying relationship)

Figure 5 NWDB IDEF1X Logical Model



Elementary Processes

Deliverable 5

Elementary processes represent the smallest units of work which can be executed to modify the content of the information base and which will leave the information base in a state consistent with integrity rules.

The elementary processes in the example define the Create/Update/Delete (CUD) processes used on each core object. They can be generated from the Information Model, and they form the basis for the CRUD matrix in Deliverable 7 (see "CRUD Matrix" on page A-35). They also form the basis for analyzing the processes by beginning and ending state and recording the changes made to the data by a person or automated system.

Notes: * items are processes which include the enforcement of the mandatory relationships among tables, shown in Deliverable 4 (see "Logical Information Model—IDEF1X Diagram" on page A-19). However, the following example demonstrates no mandatory relationships among tables.

This example is for the tables of the NWDB. They demonstrate the elementary processes by table.

This example extracts specific information from the NWDB. To view the complete deliverable for this subset of the NWDB, see the World Wide Web:

http:// sass902d.itd.sandia.gov/SILCExam.pdf

Notes: The Optional column notes Null Allowed. The following describes the Data Entry designation:

md	Mixed determination—both the user and other data supplied by the system are involved in changing the data
derived	Only the system may change the data, using a "derivation rule"
user	A person enters or modifies the data without using other data from the database (also called strict determination)

Elementary Processes

Table 5 CUD Actions by Table

CUD/Table Row	Optional	Data Entry	Comments
customer			
cust_id		derived	system-generated key
stat_chg_agent_type		derived	system-generated key
cust_status		md	user/pick list
stat_chg_date_time		derived	system clock
stat_chg_agent		derived	
cust_mail_name		user/HRIS	
clearance_type		user/HRIS	
work_site_code	✓	md/HRIS	user/pick list site.site_code
office_phone	✓	user/HRIS	
office_fax	✓	user/HRIS	
assigned_org	✓	user/HRIS	
building_num	✓	user/HRIS	
room_num	✓	user/HRIS	
mail_stop	✓	user/HRIS	
emailpo	✓	user/HRIS	
in_hris_flag		derived	
login_flag		derived	
cust_lookup_name		derived	
account			
central_svc		md	pick list
login_name		md	pick list from login table
acceptance_date	✓	user	
case	✓	HRIS/FIS	
account_status		md	pick list
stat_chg_date_time		derived	system clock
stat_chg_agent		derived	administrator login
card_serial_num	✓	user	
card_expire_date	✓	user	
set_up_cust_id	✓	user	
file_disp_flag	✓	md	pick list
stat_chg_agent_type		derived	administrator type
login			
login_name		md	system-generated/user-modified

Table 5 CUD Actions by Table (Continued)

CUD/Table Row	Optional	Data Entry	Comments
login_holder_type		md	pick list
login_status		md	pick list
stat_chg_date_time		derived	system clock
stat_chg_agent		derived	administrator login
unix_user_id		md	system-generated/user-modified
acct_expire_date	✓	user	
stat_chg_agent_type		derived	administrator type
login_holder		user	
clsfn_level		md	pick list
nwdb_user_grp			
nwdb_user_grp		user	
grp_nm_org	✓	user	
grp_nm_phone	✓	user	
grp_nm_mail_stop	✓	user	
grp_ca_org	✓	user	
grp_ca_phone	✓	user	
grp_ca_mail_stop	✓	user	
grp_ca_fax	✓	user	
grp_nm_fax	✓	user	
grp_nm_e_mail	✓	user	
grp_ca_e_mail	✓	user	
nwdb_user_grp_mbr			
nwdb_user_grp_mbr		md	assign *customer* as member of nwdb group pick list
mbr_cust_id		md	pick list/validation=customer table
cases			
case		HRIS/FIS	case must be valid case number from this list
statusedcase_status		md	pick list
site			
site_code		md	site code must be validated from this list
describedsite_desc		user	
entity			
entity		user	
login_flag		md	pick list

Table 5 CUD Actions by Table (Continued)

CUD/Table Row	Optional	Data Entry	Comments
reference_record			
reference_record		derived	system-generated keys
last_cust_unix_id		derived	system-generated keys
last_entity_unix_id		derived	system-generated keys
last_cust_id		derived	system-generated keys

Presentation Sets

Deliverable 6

Presentation sets consist of one Process/Actor State Transition Matrix per core entity (usually represented by a table), the completed storyboard diagrams, and the event and process lists:

- The Process/Actor State Transition Matrix (PAST) will add rigorous definitions of the processes related to each presentation set or object. In addition, from the customer's viewpoint, it helps integrate the actions performed and the state of the information system for each elementary process.
  - –The analyst first draws up a rough draft of the matrices and then confirms them with all the customers. It is this process of confirming the matrices that adds value of using PAST. Matrices that have not been confirmed with the customer will clarify few concepts.
- Completed storyboard diagrams with annotated sets denoting groups of data items to be treated as a single unit (as in elementary processes). Standard templates, when available, will be applied to provide a consistent "look and feel" with other systems. Diagrams may be hand-drawn initially, to help the customers view their data and how they will enter data into the computer. Other diagrams will enable the customers to think about their data and how they want to use it, i.e., reports, screen displays, graphs, etc. The application development software or other vendor software may be used, starting the prototyping processes early in the customer-analyst dialogues.

Process/Actor State Transition Matrix

Deliverable 6

Description

- One matrix for each major database entity, or each major object (see "Logical Information Model—IDEF1X Diagram" on page A-19)
- One column for each major actor (actors can be people or software modules/ systems)
- One row for each major customer or system task (each thing that can happen to the entity/object)
- Column for Beginning state (status) on the left; Ending state (status) on the right
- Not every row results in a state (status) change
- Cells show what each actor does

Uses

- Help customers and developers understand the system
- Help define the requirements
- Help think through business needs
- Provide systematic, structured framework for process analysis
- Get more detail in specifications
- Help all parties communicate about requirements and specifications
- Capture knowledge about the system for managers, customers, future maintainers

Inputs Needed

- Context diagram
- Core requirements (more may be found later)
- Core data entities (more may be found later)
- List of players (actors)
- List of tasks or events
- List of processes

Skills/Guidelines for Writing

- Listen
- Understand customers' needs
- Communicate clearly in writing, with precision and conciseness
  - –Use customer's terminology
  - –Avoid computer jargon
  - –Avoid methodology's jargon

Table 6 Customer Process/Actor State Transition Matrix (Valid Status List: Null, Active, Inactive)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Create customer from HRIS upload file	Null	n/a	Sends batch upload files (Changes file 5 nights a week; Big file every Friday)	n/a	n/a	n/a	n/a	If customer does not exist, creates customer, generates e-mail name, generates lookup name, generates UNIX User ID, sets in_HRIS flag to Yes	Reads customer Creates customer	*If edits pass* Active *If edits fail* Null
Create customer on NWDB online screen	Null	Submits User ID Request Form	n/a	Accesses screen; creates customer	Accesses screen; creates customer	n/a	n/a	If customer does not exist, creates customer, generates e-mail name, generates lookup name, generates UNIX User ID, sets in_HRIS flag to no	Reads customer Creates customer	Active
Display customer info online	Active or Inactive	Accesses NWDB screen	n/a	Accesses NWDB screen	Accesses NWDB screen	Access es NWDB screen	Computer Security accesses NWDB screen	Accesses all related tables to get customer info	Reads customer	Active or Inactive
Reactivate customer online	Inactive	May place phone call to NWDB Admin or Password Control	n/a	Reactivates status	Reactivates status	n/a	n/a	Enforces edits; Changes customer info	Reads customer Updates customer	Active

Table 6 Customer Process/Actor State Transition Matrix (Valid Status List: Null, Active, Inactive) (Continued)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Reactivate customer from HRIS upload file	Inactive	n/a	Sends batch upload files (Changes file 5 nights a week; Big file every Friday	n/a	n/a	n/a	n/a	Enforces edits; Changes customer info	Reads customer Updates customer	Active
Deactivate customer	Active	n/a	n/a	Reads report of inactives, greater than 1 year	Reads report of inactives, greater than 1 year	n/a	n/a	Changes status of customer	Reads customer Updates customer	Inactive
Deactivate customer from HRIS Update File	Active	n/a	Sends batch upload files (Changes file 5 nights a week; Big file every Friday	n/a	n/a	n/a	n/a	Changes status of customer	Reads customer Updates customer	Inactive
Update customer from HRIS Update File	Active	n/a	Sends batch upload files (Changes file 5 nights a week; Big file every Friday	n/a	n/a	n/a	n/a	Enforces edits; Changes customer info	Reads customer Updates customer	Active
Report customers with inactive status greater than 1 year		This task is reported on the login entity								
Report customers with expired logins		This task is reported on the login entity								

Table 6 Customer Process/Actor State Transition Matrix (Valid Status List: Null, Active, Inactive) (Continued)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Assign customer to NWDB User Groups		This task is reported on the NWDB User Group								
Assign customers to a login holder		This task is reported on the login entity								
Generate e-mail address & pass to HRIS	Active or Inactive	n/a	Receives e-mail address and updates HRIS	n/a	n/a	n/a	n/a	Reads all new/ updated customers; Creates e-mail addresses; Builds file for HRIS	Reads customer Update customer	Active or Inactive
Delete customer	Inactive	n/a	n/a	Reads report of deleted customers	Reads report of deleted customers	n/a	n/a	Deletes customer who has been inactive more than 1 year	Deletes customer	Null
Process SecurID acceptance forms		Outside the project's scope								
Create customer from online screen (only for contractors & visitors)		Outside the project's scope								
Update customer's cryptographic public key		Outside the project's scope								

Table 7 Logins Process/Actor State Transition Matrix (Valid Status List: Null, Active, Inactive)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Create login name from HRIS upload file	Null	n/a	Sends batch upload files (Changes file 5 nights a week; Big file every Friday)	n/a	n/a	n/a	n/a	If login does not exist, creates a login	Reads login Creates login	*If edits pass* Active *If edits fail* Null
Assign customer a login name for each classification level	Null	Submits request form for confidential and secret	n/a	Assigns login holder & login holder type	Assigns login holder & login holder type	n/a	n/a	Enforces edits (validates clearance against Physical Security clearance file)	Reads login Creates login	*If edits pass* Active *If edits fail* Null
Display login info online	Active or Inactive	Accesses NWDB screen	n/a	Accesses NWDB screen	Accesses NWDB screen	Access-es NWDB screen	Accesses NWDB screen	Accesses all related tables to get login info	Reads login	Active or Inactive
Update login name	Active	May place phone call to NWDB Admin or Password Control; Completes form	n/a	May receive phone call from Line Org Individual; Receives Discrepancy Report; Updates login	May receive phone call from Line Org Individual; Receives Discrepancy Report; Updates login	n/a	n/a	Enforces edits; Changes login info	Reads login Updates login	Active
Update account expiration date for each classification level	Active	n/a	n/a	n/a	Reads report; Uses account screen to reauthorize account expiration date	n/a	n/a	Enforces edits; Sets expiration date ahead either 6 months or 1 year, according to classification level	Reads login Updates login	Active

Table 7 Logins Process/Actor State Transition Matrix (Valid Status List: Null, Active, Inactive) (Continued)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Report customers with expired logins	Active	n/a	n/a	Reads report	Reads report	n/a	n/a	Generates report of account expiration dates less than today's date	Reads login	Active
Report customers with inactive login status greater than1 year	Inactive	n/a	n/a	Reads report	Reads report	n/a	n/a	Generates report of logins whose login status has been inactive greater than 1 year	Reads login	Inactive
Deactivate login	Active	n/a	n/a	Reads Discrepancy Report; Updates login	Reads Discrepancy Report; Updates login	n/a	n/a	Enforces edits; Records update in login info	Reads login Updates login	Inactive
Delete login	Inactive	n/a	n/a	Reads report of deleted logins	Reads report of deleted logins	n/a	n/a	Deletes login that has been inactive more than 1 year	Deletes login	Null
Create login from online screen (only for contractors & visitors		Outside the project's scope								
Reauthorize account expiration date from online screen (only for contractors & visitors)		Outside the project's scope								

Table 8 Accounts Process/Actor State Transition Matrix (Central service/LAN POC accounts) (Valid Status List: Null, Active, Inactive)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Assign customer an account for each desired central service	Null	Submits request form for confidential and secret	n/a	Assigns account	Assigns account	Assigns account (LAN POC accounts only)	n/a	Enforces edits (validates case number & clearance level)	Reads account Creates account	*If edits pass* Active *If edits fail* Null
Set up accounts for service group	Null	Submits request for service group under RSN or SSN	n/a	Reads request; Accesses account screen	Reads request; Accesses account screen	n/a	n/a	Enforces edits; Creates accounts for service groups	Create account	*If edits pass* Active *If edits fail* Null
Display account info online	Active or Inactive	Accesses NWDB screen	n/a	Accesses NWDB screen	Accesses NWDB screen	Accesses NWDB screen	Accesses NWDB screen	Accesses all related tables to get login info	Reads login	Active or Inactive
Reauthorize account	Active	This task is reported on the login entity							Updates account	Active
Report customers with account expiration dates less than today's date	Active	n/a	n/a	Reads report	Reads report	n/a	n/a	Generates report of account expiration dates less than today's date	Reads account	Active
Report customers with inactive account status greater than1 year	Inactive	n/a	n/a	Reads report	Reads report	n/a	n/a	Generates report of accounts whose account status has been inactive greater than 1 year	Reads account	Inactive
Deactivate account	Active	Submits request to deactivate account	n/a	n/a	Reads requests; Updates account	Reads requests; Updates account (LAN POC accounts only)	n/a	Enforces edits; Records update in account info	Reads account Updates account	Inactive

Table 8 Accounts Process/Actor State Transition Matrix (Central service/LAN POC accounts) (Valid Status List: Null, Active, Inactive) (Continued)

Object Life Cycle Event (Task)	Beginning State	Actions Taken—By Business Function							Elementary Processes	Ending State
	Incoming Status	Line Org Individual	HRIS	NWDB Admin	Password Admin	LAN POC	Security System	NWDB Application Program		Resulting Status
Deactivate account through HRIS	Active	n/a	Sends batch upload file (Changes file 5 nights a week; Big file every Friday)	n/a	n/a	n/a	n/a	Enforces edits; Records update in account info	Reads account Updates account	Inactive
Delete account	Inactive	n/a	n/a	Reads report of inactive accounts	Reads report of inactive accounts	n/a	n/a	Deletes account that has been inactive more than 1 year	Deletes account	Null
Create account from online screen (only for contractors & visitors		Outside the project's scope								
Reauthorize account expiration date from online screen (only for contractors & visitors)		Outside the project's scope & reported on login entity								

Storyboard Diagrams

(● = field must be filled in)

Customers

Mail Name: David P Duggan
Lookup Name: Duggan, David P.
First Name: David
Middle Name: Phillip (Please provide at least middle initial if at all possible)
Last Name: Duggan
Clearance: none (Q, L, or none)
Customer Status: a (a or i)
Status last changed on date: 4/20/88
Status last changed at time: 12:30 PM
Status changed by: Eddie L Foster
Works at site: SA
Phone: 505-885-8100
FAX: 505-885-2331
Works for organization: 01957
Works in building: 880
Works in room: C20
Hard-copy Mail Stop: 0661
E-mail Post Office: snl2800
E-mail type: UX
E-mail address: dduggan@snl2800.sandia.gov
[redacted]
Does this customer hold a cryptographic public key? Yes
Cryptographic public key(s): David Duggan's public key 1; David Duggan's public key 2 (if yes)

NWDB Customer ID: 555223333
Belongs to NWDB user groups: Router; Network analysts; Password administration
Customer in HR system? Yes
Customer Type: Contractor (sandian, contractor, or visitor)

if a Contractor or Visitor:
Mailing Address: 322 Northhampton Lane; Suite 727; Maryvale, RI 83442
Company: Fred's Hi-Tech Co.
Country: ME
Sandia Sponsor: Jane J Freeland

if a Visitor:
Access Expires: 3/21/94

if a Contractor:
Contract #: AF-88332

Logins

Name of Login holder: David P Duggan
Type of Login holder: Customer (Customer or Entity)

Logins by classification level

Level	Status	Status last changed on	Status last changed at	Status changed by	Login Name	User ID
u	a	9/30/94	12:30 AM	Mike J Smith	dduggan	1209
s	i	3/21/94	5:30 PM	HR update	dduggans	1390

CRUD Matrix

Deliverable 7

The CRUD (Create, Read, Update, Delete) matrix identifies each elementary process used on each core object of the information system. Accompanying analysis assists in verifying that each object is supported by a create, read, update, and delete process—it also helps identify redundancy of processes for an object.

This approach clusters elementary processes with objects, revealing a matrix in process sequence and identifying clusters from top left to bottom right.

In most instances, the C's for Create will fall on the diagonal. All other uses appear in the lower triangle of the matrix. The software engineer determines the cluster boundaries. The resulting rectangle will show the related process and objects that should be in a common module or grouping of objects and activities.

Some CASE tools will perform initial clustering on a matrix, but human interaction is usually required by the software engineer to finalize these groupings.

Note: In the following example, words in italics indicate the groupings that form logical screen processes. The **R** indicates mandatory read information, while **r** represents read to find optional information.

CRUD Matrix

Table 9 CRUD Matrix—*Create* Version Only

Object creation dependency matrix (The C in CRUD)	organization	site	case	agent	status	clearance	customer	login_holder	login	central_service	account	nwdb_user_group	nwdb_user_group_customer	secureid_serial_nbr
create organization	C													
create site		C												
create case			C											
create agent				C										
create status					C									
create clearance						C	*cust*							
create customer	r	r		R	R	R	C		*login*				*grp*	
create login_holder							R	C						
create login				R	R			R	C		*acct*			
create central_service										C				
create account			r	R	R		r		R	R	C			
create nwdb_user_group	r											C		
create nwdb_user_group_customer							R					R	C	
create securid_serial_nbr											r			C

Extended Presentation Sets

Deliverable 8

Extended Presentation Sets consist of storyboard diagrams that include vendor-specific and implementation-specific features, as well as application of standard template features for the enterprise systems. These are normally "extended" or evolved from initial diagrams in Deliverable 6 (see "Storyboard Diagrams" on page A-34). They are often produced and managed electronically. An evolutionary prototype might begin with diagrams and electronic screens produced by the presentation layer tool. As these designs evolve, test data and coding are added for rapid production of the first product.

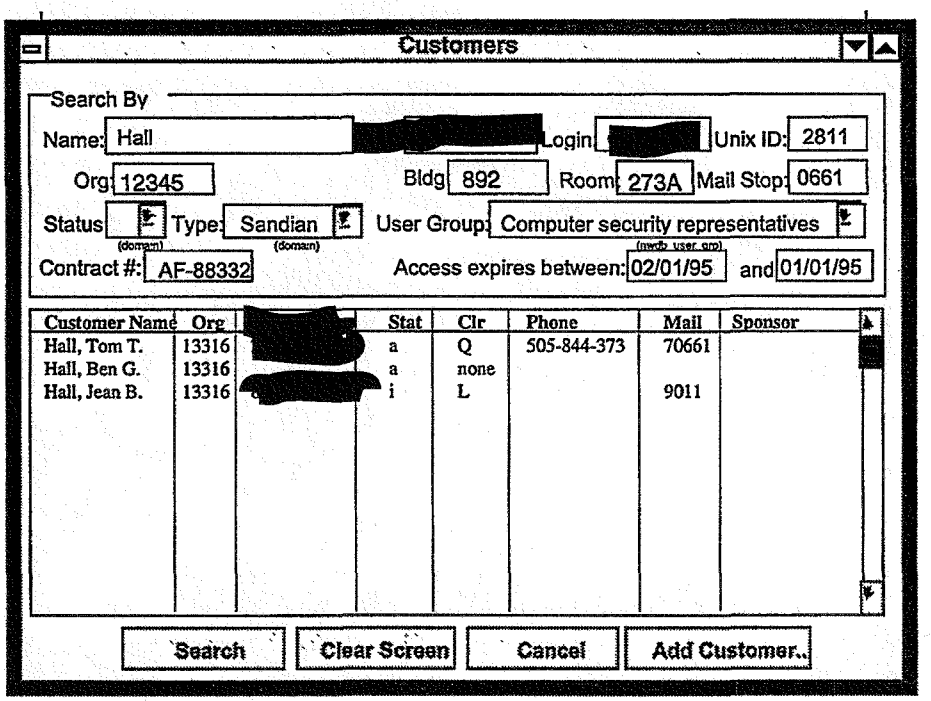
This example extracts specific information from the NWDB. To view the complete deliverable for this subset of the NWDB, see the World Wide Web:

http:// sass902d.itd.sandia.gov/SILCExam.pdf

Extended Presentation Sets

Customer List Screen

Figure 6 Customer List Screen



Records used

contract
customer
login
nwdb user_grp_mbr

Access controls

clist-10 A user must belong to one of the NWDB user groups "Password administration" or "NWDB administrators" to click the Add button on this screen.

Details

clist-20 [redacted] Login name, or Unix User ID is entered, all other data should be ignored and a match made directly on that field. Otherwise, search for all matches. Entering any one of [redacted] Login name, or Unix User ID will blank out the other three.

clist-30 Matches should be obtainable based on partial login name, lookup name or org. (Other matches should be by full field content.)

Say in Help what fields can be partially entered. Also remind them to place leading zeroes on org if want "013xx" rather than "13xxx."

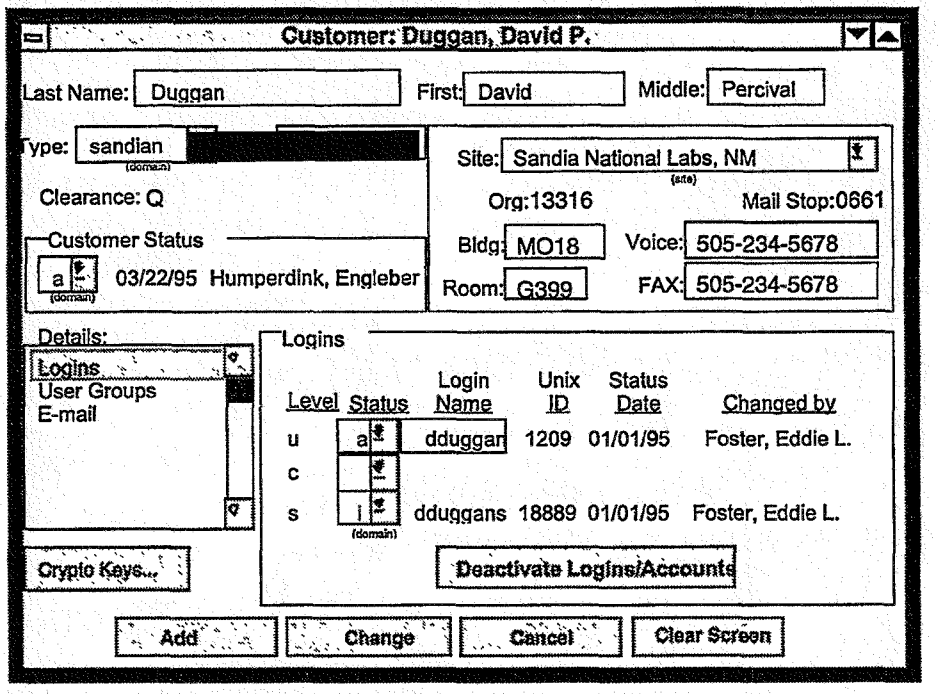
[redacted]

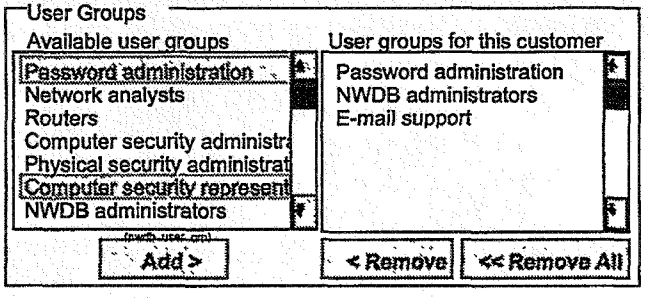
clist-50 When a building is input, the input field should be altered to strip any leading "B" and any embedded spaces, blanks, "-," or "/," and should show on the screen in stripped form. Only a-z, A-Z, and 0-9 may be input, and these will be forced to upper case.

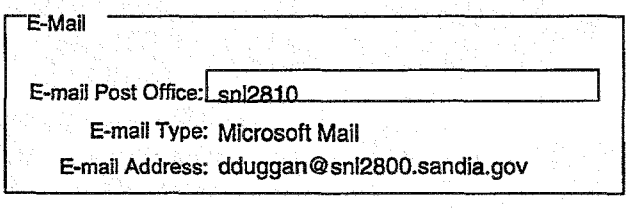
clist-60 When a room is input, the input field should be altered to strip any embedded spaces, blanks, "-", or "/", and should show on the screen in stripped form. Only a-z, A-Z, and 0-9 may be input, and these will be forced to upper case.

Customer Detail Screen

Figure 7 Customer Screens







Records used

accounts (update)
customer (update)
emailpo
login (update)
nwdb_user_grp
nwdb user_grp_mbr (update)

reference_record (update)
site

Access controls

cust-10 A user must belong to one of the NWDB user groups "Password administration" or "NWDB administrators" to update any data on this screen.

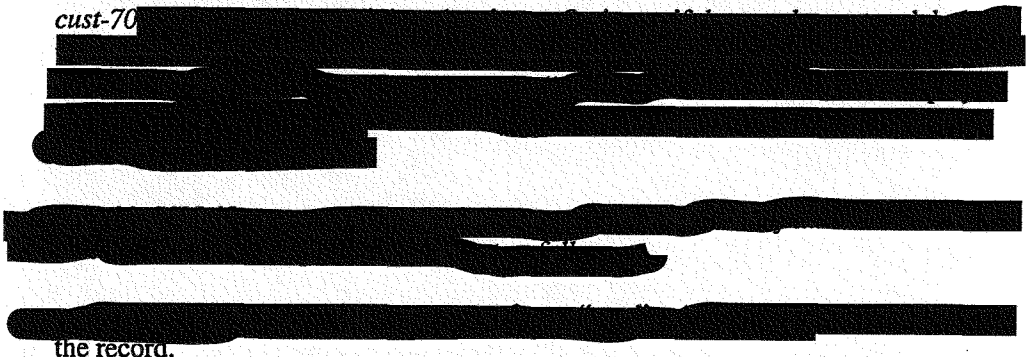
Details

cust-20 When a customer is added for the first time (when "Add" is clicked), a new customer ID must be generated (this is the primary key to customer). Do the generation as follows: get the last customer ID used from the reference record and add 1. Set the last customer ID used to that value, and set the customer ID to that value.

cust-40 Last, first, and middle name fields should be only alphabetic plus any embedded "-" or blank spaces, and the first character of the name should be forced to be upper case, and the first character after any "-" or blank should be forced to be upper case.

cust-50 If the customer first, middle, or last name is changed, the following should happen: The customer's first name, a space, the first character of the middle name and a space (if there is a middle name), and last name should be concatenated to form the customer mail name. The customer's last name, a comma, a space, first name, and if there is a middle name, another space, the first character of the middle name and a period should be concatenated to form the customer lookup name. Customer lookup name and mail name do not show on the screen.

cust-70 [redacted]



the record.

cust-100 Clearance code should be automatically defaulted to "none" if a customer is initially added on this screen. Clearance code cannot be entered on-line. Clearance code is updated only from the HR system.

If the customer status is changed to "a," or first entered as "a," a Login Name and User ID may need to be automatically generated (this is detailed below).

The field shown for "Site" should be the site description, even though that is not the primary key.

cust-120 Site is required for customers of type "sandian", even though it is nullable in the record.

Note that a customer's org. may not be entered here—it is updated only from the HR system.

Note that mail stop may not be entered here—it is updated only from the HR system.

cust-150 When a building is input, any leading "B" and any embedded spaces, blanks, "-", or "/" should be stripped out and the building number displayed in stripped form. Only a-z, A-Z, and 0-9 may be input, and these will be forced to upper case.

cust-151 ** The building number that a customer works in should exist in the liscopy "building" table, in attribute building.bldgnum.

cust-160 When a room is input, any embedded spaces, blanks, "-", or "/" should be stripped out and the room number displayed in stripped form. Only a-z, A-Z, and 0-9 may be input, and these will be forced to upper case.

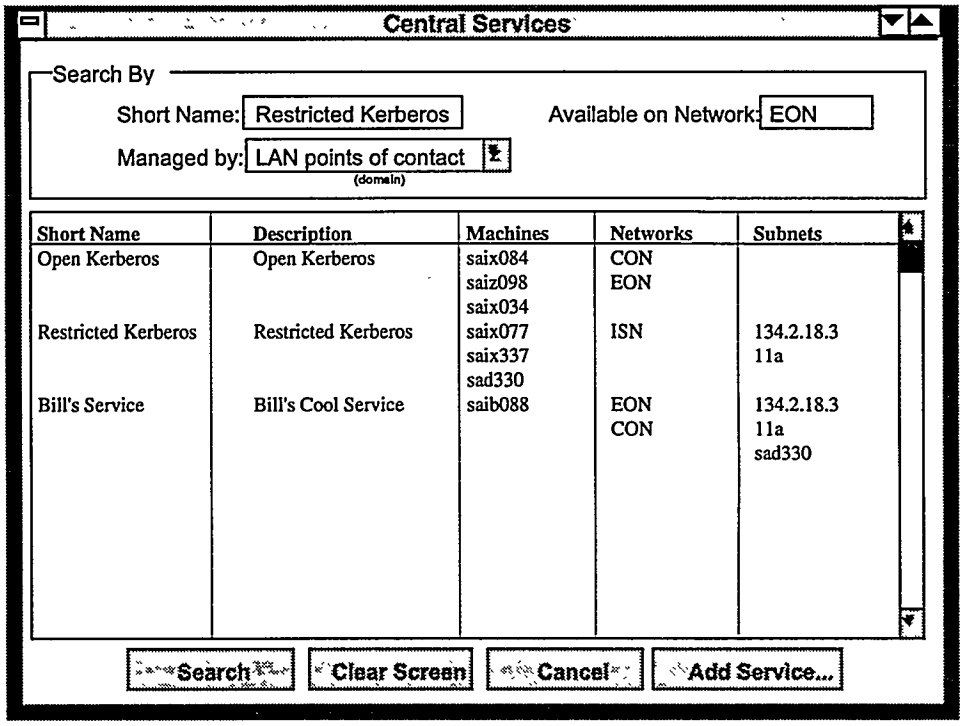
Phone numbers may be entered on-line for phones in other countries, so there is no set format for phone number. Phone number should be stored without any embedded dashes or spaces, so these should be stripped on input.

cust-170 For display, any phone numbers made of 10 characters should have parens and a dash inserted as they are typed and be shown in the format (999) 999-9999. Phone numbers with more or less than 10 characters should display just as a number. (Note that the HR update will keep most phone numbers current.)

(*Note:* The Logins detail is listed last in this section, even though it is the default first detail, since it is the most complicated and you might want to save it for last).

Central Services List Screen

Figure 8 Central Services Screen



Records used

central_svc
central_svc_machine
central_svc_network
central_svc_subnet

Access controls

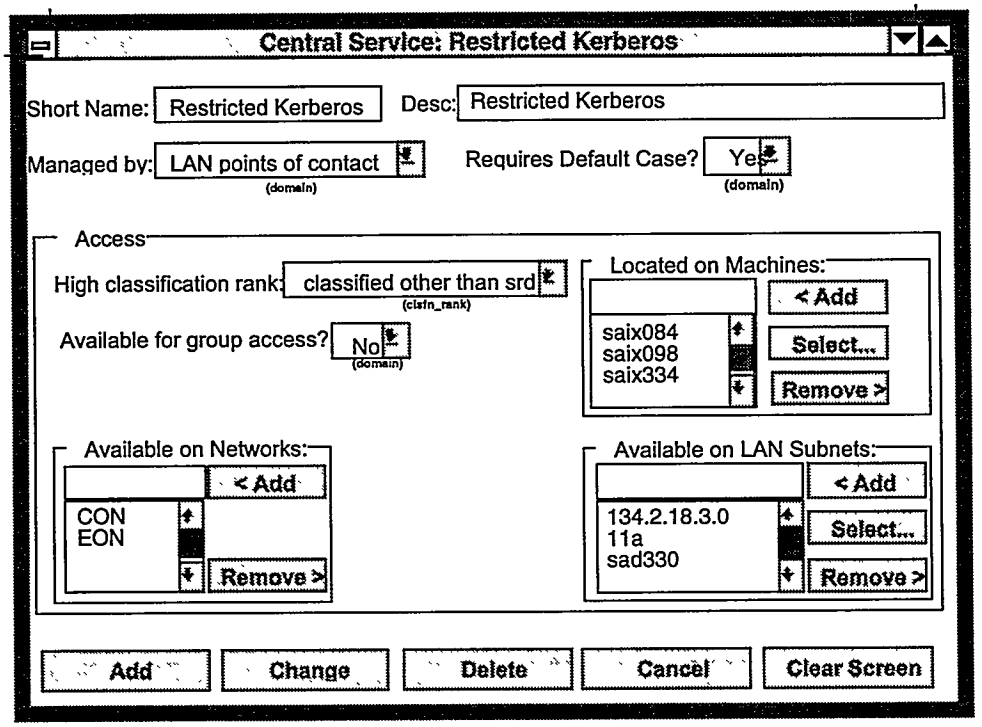
cslist-10 A user must belong to the user group "NWDB administrators" or "Password administration" to click the Add button on this screen.

Details

(none)

Central Service Detail Screen

Figure 9 Central Service Detail Screen



Records used

central_svc (updated)
central_svc_machine (updated)
machine
clsfn_rank
lan_subnet
central_svc_subnet (updated)
network
central_svc_network (updated)

Access controls

cs-10 A user must belong to the user group "NWDB administrators" or "Password administration" to update any data on this screen.

Details

Note: The pick-list for clsfn_rank is not the primary key field. For all display purposes (here and elsewhere), we are using the description of the classification rank, not its numeric code. Users other than NWDB administrators should never see the numeric code.

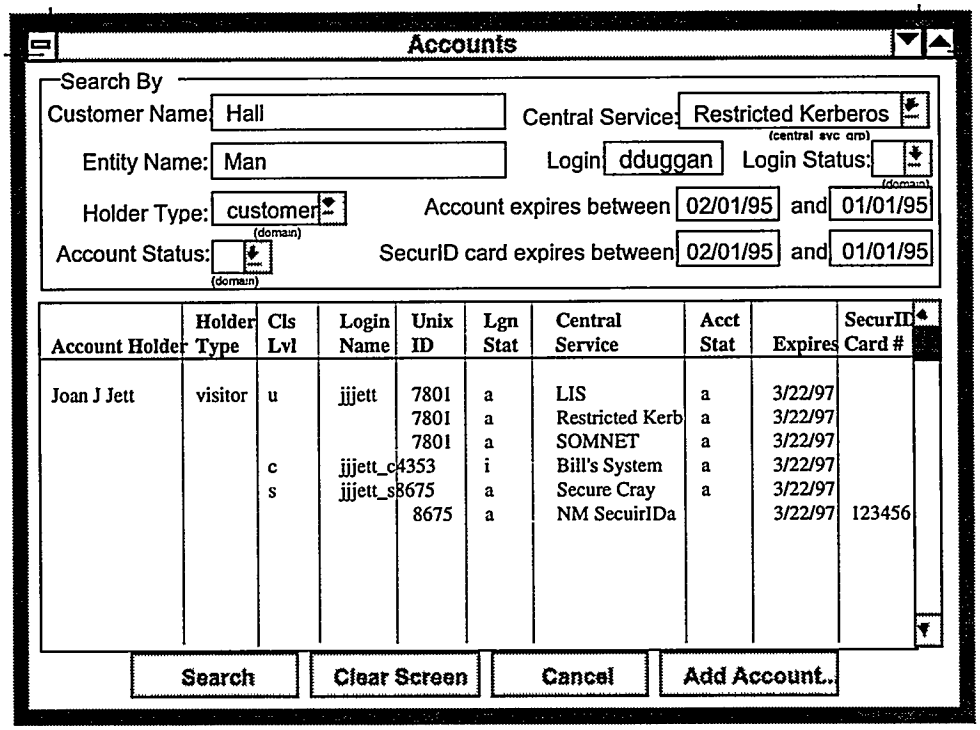
cs-20 A central service may only be managed by the NWDB User groups "LAN points of contact" and "Password administration," so rather than having a pick_list into the nwdb_user_grp table, we suggest a short custom picklist with these two values.

Note: The two flags "Requires default case?" and "Available for group access?" are shown as "Yes" and "No," but these values are stored in the database as "Y" and "N."

cs-30 A central service is a POC-managed central service when it is managed by the NWDB User group "LAN points of contact." Only a POC-managed central service may be available on a subnet—hide this box otherwise.

Accounts List Screen

Figure 10 Accounts Screen



Records used

customer
entity
login
account
central_svc

Access controls

alist-10 A user must belong to the user group "NWDB administrators," "Password administration," or "LAN points of contact" to click the Add button on this screen.

Details

alist-20 Login name when entered must be alpha and lower case.

alist-30 If the search criteria yields a customer or entity without a Login Name, or with a Login Name but without any accounts, go ahead and show them (i.e., show all

matches, even if you can't fill out the whole report line). If the search criteria yields both customers and entities, show both in the report.

alist-40 "Account holder type" is a combination of login holder type (customer, entity) and customer type. Basically, we want to convert login holder type "customer" into the three types of customers for display on this query.

alist-50 Each login name for a login holder should show only once, even if there are multiple accounts for that login name.

Note that the fields "Account Holder" and "Central Service" in the list are truncated to make room.

DB Schema

Deliverable 9

The Physical Database, using the constructs of the vendor software and the additional information needed by the implementer of the system, is documented with the database schema diagram in the form of IDEF1X. Optionally, the SQL listings may be used. The SQL to support all of the elementary processes for CRUD (Create, Read, Update, and Delete) may be automatically generated from the physical model by a tool.

To view the complete SQL listings for this subset of the NWDB, see the World Wide Web:

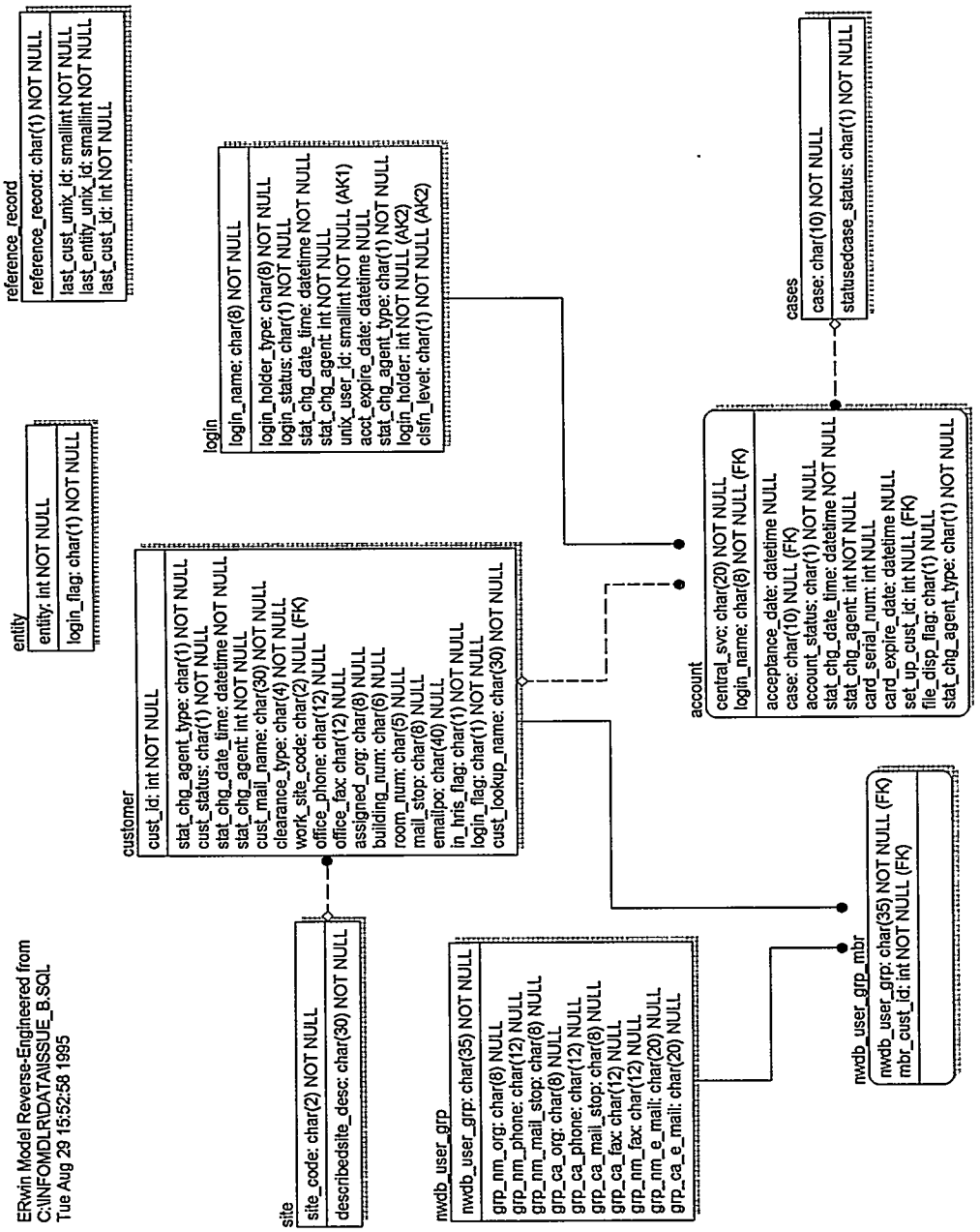
http:// sass902d.itd.sandia.gov/SILCExam.pdf

Note: Keys are listed first in each table with a line following the last item in the key. Data items which uniquely identify an object are usually the primary key. The following describes the symbols used in an IDEF1X diagram.

PKxx	Primary key; key that uniquely identifies one object from another
AKxx	Alternate key; a secondary identifier
FKxx	Foreign key; key of another entity when it migrates into a given entity as a result of a relationship between the two entities
Parent	An entity on the 1-end of a 1-to-many relationship
Child	An entity on the many end of a 1-to-many relationship
———	Identifying relationships (foreign key is part of primary key)
- — -	Non-identifying relationships (foreign key is not part of primary key)
Cardinality	The quantifier in a relationship between two entities; a quantifier answers the question, "How many?"
——P● or - — P●	Existence of at least one matching child required if parent exists (1: 1 or more)
——Z● or - — Z●	Existence of maximum of one child matching parent (1: 0 or 1)
——● or - — ●	1: 0 or more
- — ◇	Parent-child relationship where the parent entity is optional (i.e., optional non-identifying relationship)

DB Schema

Figure 11 DB Schema



Extended Dependencies and Implementation Constraints

Deliverable 10

This deliverable consists of a formal definition of precedence required in the flow steps.

This example shows an overall description that applies to all screens. The NWDB does not have complex algorithms or calculations. If a system does contain these, they would be included in this deliverable.

The format presented here was very useful for the testers and for generating the error messages, and it helped clarify material for the user documentation. This format is not necessarily recommended for every system.

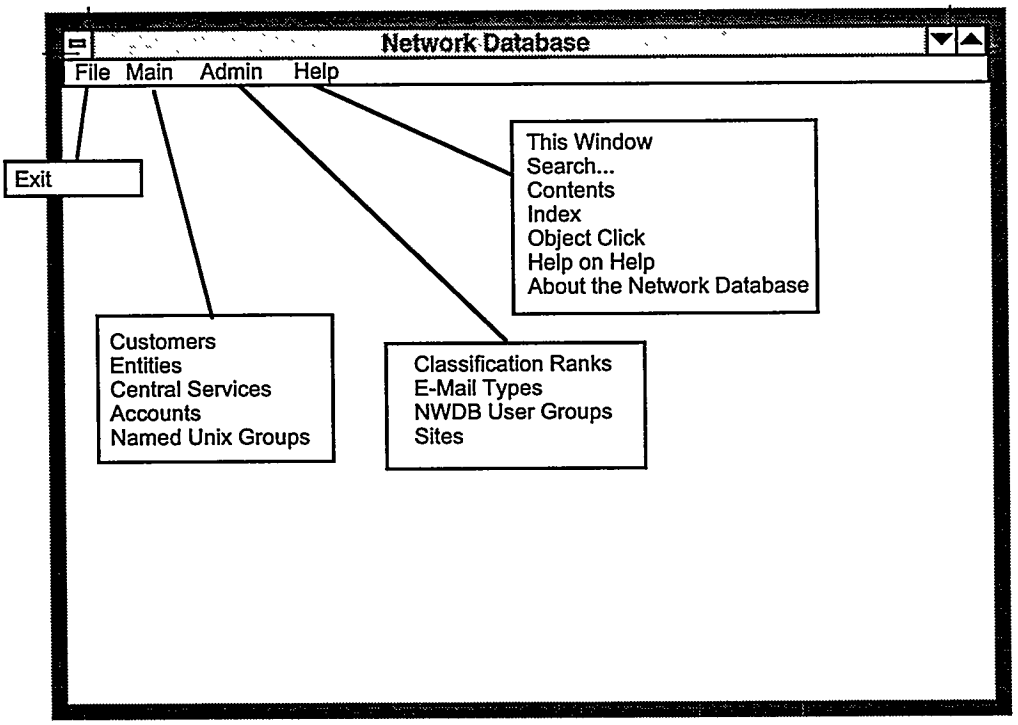
This example extracts specific information from the NWDB. To view the complete deliverable for this subset of the NWDB, see the World Wide Web:

http:// sass902d.itd.sandia.gov/SILCExam.pdf

Extended Dependencies and Implementation Constraints

Process Flow Dependencies (Menu)

Figure 12 Network Database Menus



General Screen Criteria (Reusable GUI Guidelines and Framework)

gen-10 The menus should match what is shown. Each menu pick should bring up the correct screen. The screen title should match the menu.

gen-20 For every input field: If the user tries to type data into a field that does not match the basic data type as shown in the IDEF1X diagram, (e.g., alpha data into a numeric field or numeric data into an alpha field) the system should not allow it, preferably by not allowing the data to be typed into the field in the first place.

gen-30 When a field in a record is listed as "Not Null" on the IDEF1X diagram, and that record is stored from a screen, the screen should either require the user to fill in that field (if the field shows on the screen) or loaded with an appropriate default (if the field does not show on the screen).

gen-40 Primary key fields may not be Changed once added. There is one exception, as explained in login-220, which is login names (in the case of login names we will have to go out and fix all the related foreign keys to match the new login key). Otherwise, if key fields have been changed and the user clicks Change, an error message should appear.

gen-50 If a user has entered or changed data on a Detail screen (i.e., not just search criteria on a List screen) and then tries to close the screen either by clicking the close box or by choosing Cancel, the system should pop up a warning that requires confirmation.

gen-60 An Add on any screen when an entity with that key already exists in the database should produce an error message.

gen-70 On all Deletes, the system should ask for confirmation before doing the delete.

gen-80 The Clear button should remove all data from a screen.

gen-90 Cancel (if it is not the situation where changes have been made and not saved) should simply close a screen.

gen-100 If a screen title includes the basic title plus the key of the primary entity appearing on that screen (this is true on most complex detail screens), the key shown in the title should match the key shown in the body of the screen. If the key is compound (like on the Accounts screen), its components should be separated with a "/."

gen-110 When there are fields dependent on a subtype, those fields should be automatically shown when an occurrence of that subtype is on the screen, and hidden otherwise. Each such field dependency is described in the text below.

gen-120 If a user does not have privileges as a result of his NWDB user group to update data on a screen, "Add," "Change," and "Delete" buttons should be grayed out. This includes the Add buttons on the List screens.

gen-150 The domain of status fields (other than doc_status, which is a different concept), should always be "a" and "i." Status field is always in a little pick-list. It may be shown initially with blanks to force input.

The domain of status_chg_agent_type and rcd_chg_agent_type is "c" and "p" (i.e., customer and process). This does not show on screens.

gen-180 Flags should be stored as "N" or "Y," and are shown on screens as "Yes" and "No," so we want to convert them back and forth, showing "Yes" and "No" in the pick list, and storing "N" and "Y."

gen-190 Dates must be entered as 6 digits in the sequence MMDDYY. The system should insert slashes in between as necessary, and display the date in the format MM/DD/YY.

Input Process Dependencies

In the NWDB, most input process constraints and dependencies were described in the Extended Presentation Sets (see "Extended Presentation Sets" on page A-37).

Another input dependency is an update flag that shows that the proper cycle for the Human Resources Information System (HRIS) has been executed. This will give current information accessed directly for the status and certain attributes of line organization individuals.

Output Process Dependencies

Onscreen reports use the List, Detail, and Report screens (see "Extended Presentation Sets" on page A-37).

Batch Reports include Password Reauthorization List, generated monthly. This is computed on an 11-month warning, with a 12-month expiration for each individual account. *The format of this 11-month warning or notice normally would appear here.*

Output files with e-mail address updates are generated daily and sent to HRIS. *The format of this file normally would appear here.*

Output Processes

In general, every table (each table should more-or-less map to a logical object) should be listable from its home input screen based on full, partial, or "list them all" input of the table's primary key, and from this list a user should be able to pick an instance to expand or update. When a given object is listed, as many additional attributes of that object (besides its key) should be shown as possible in the list to help the user reference the object. This type of output should basically be a given for all tables. Below we specify additional outputs, primarily because they either have more complex selection criteria, are output only periodically, or are electronic files to be sent to other systems.

Reports show as "basic queries" which should produce a wide range of on-demand information to serve most any need, and then as specific report definitions grouped under the NWDB user access type that would be the main users.

Where a report's output method is listed as "Standard," we mean that the report could, at the user's choice and with the click of a button, be (a) printed as hard copy at a user-specified printer with an appropriate cover banner identifying the requesting user (b) e-mailed to an e-mail address that they provide, or (c) downloaded to a user-specified machine under a user-specified file name. (*Note:* this latter capability requires a machine set-up that will be controlled by NWDB administrators outside of the NWDB system itself. Only "trusted" machines will be allowed.)

In general, if any query generates more than 100 lines of output, the system should stop and ask the user if he or she wants to continue, and then again at 200 lines of output, etc. This check should be bypassed for certain large standard reports.

IDEF0 Decomposition

Deliverable 11

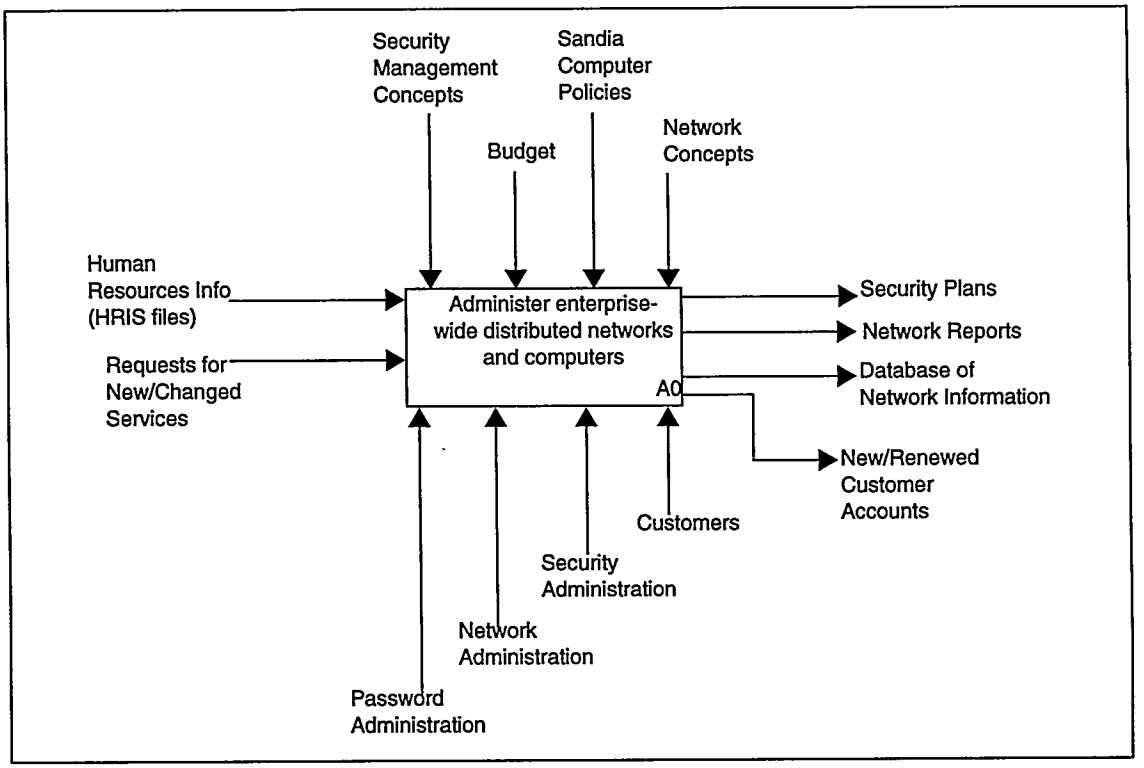
IDEF0 decomposition is a formal, hierarchical breakdown of the functions or processes that the software will provide. In addition to showing each process in progressively more detail, these graphical charts also emphasize:

- The inputs, which are transformed by the process
- The outputs that result from the process
- Inputs that are not transformed, but are controls on the process
- The "mechanisms" that are involved or use the processes

Note: Figures on the following pages include explanatory notes.

IDEF0 Decomposition

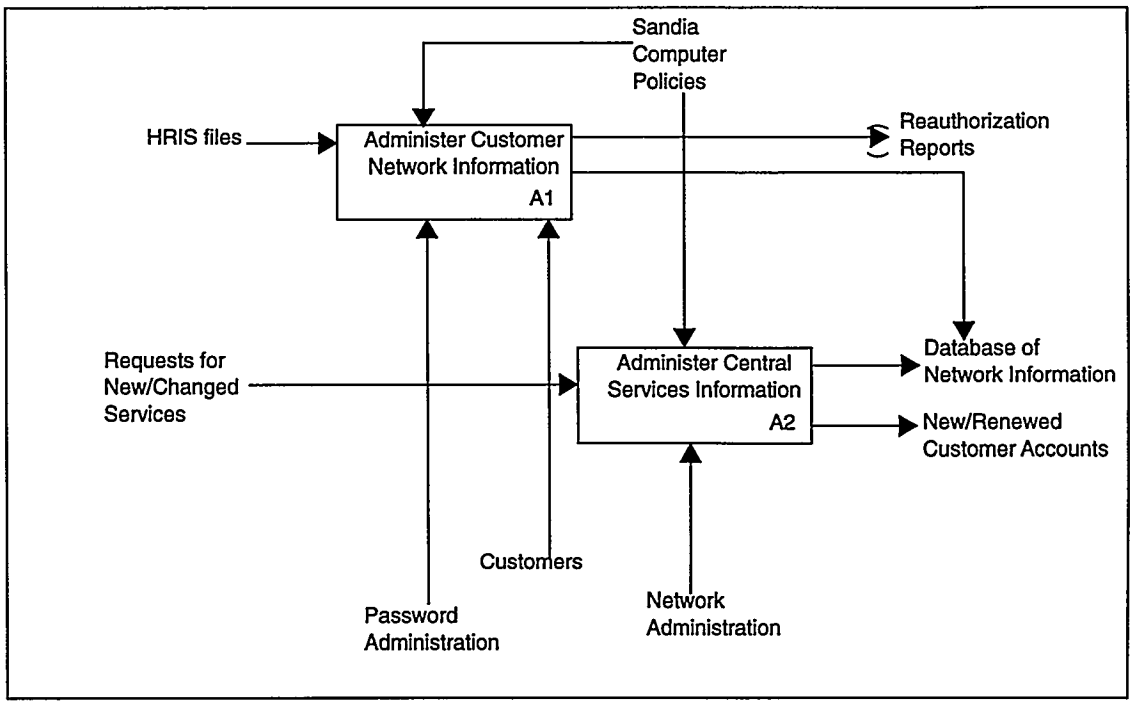
Figure 13 Node A-0, Network Database Example



Reading from the top, clockwise:

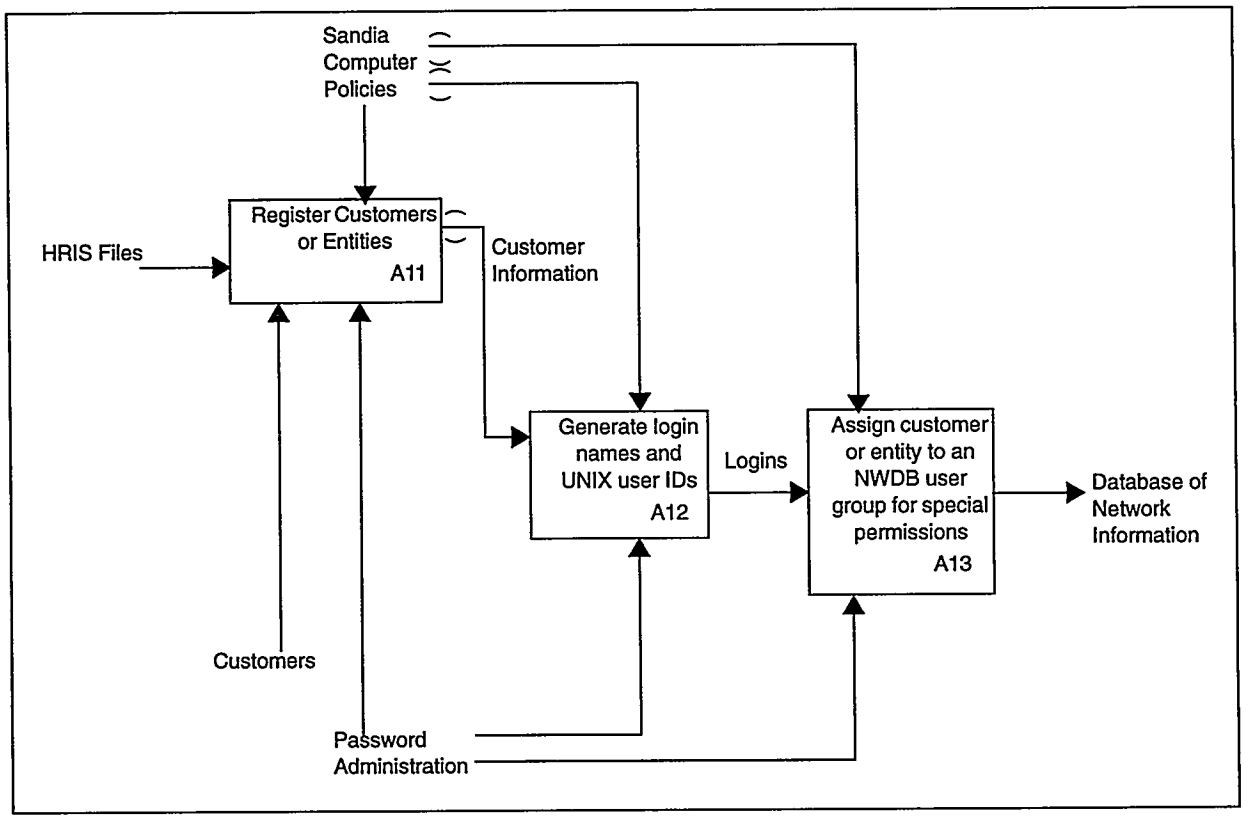
- The arrows entering the top of the box containing the process description show "controls" on this process.
- The arrows exiting to the left show the outputs.
- The bottom arrows show mechanisms, i.e., answers to who or what will act on this process.
- Finally the arrows entering from the right show those inputs that will be affected by the processes and transformed to an output.

Figure 14 Node A0, Administer Enterprise-Wide Distributed Networks



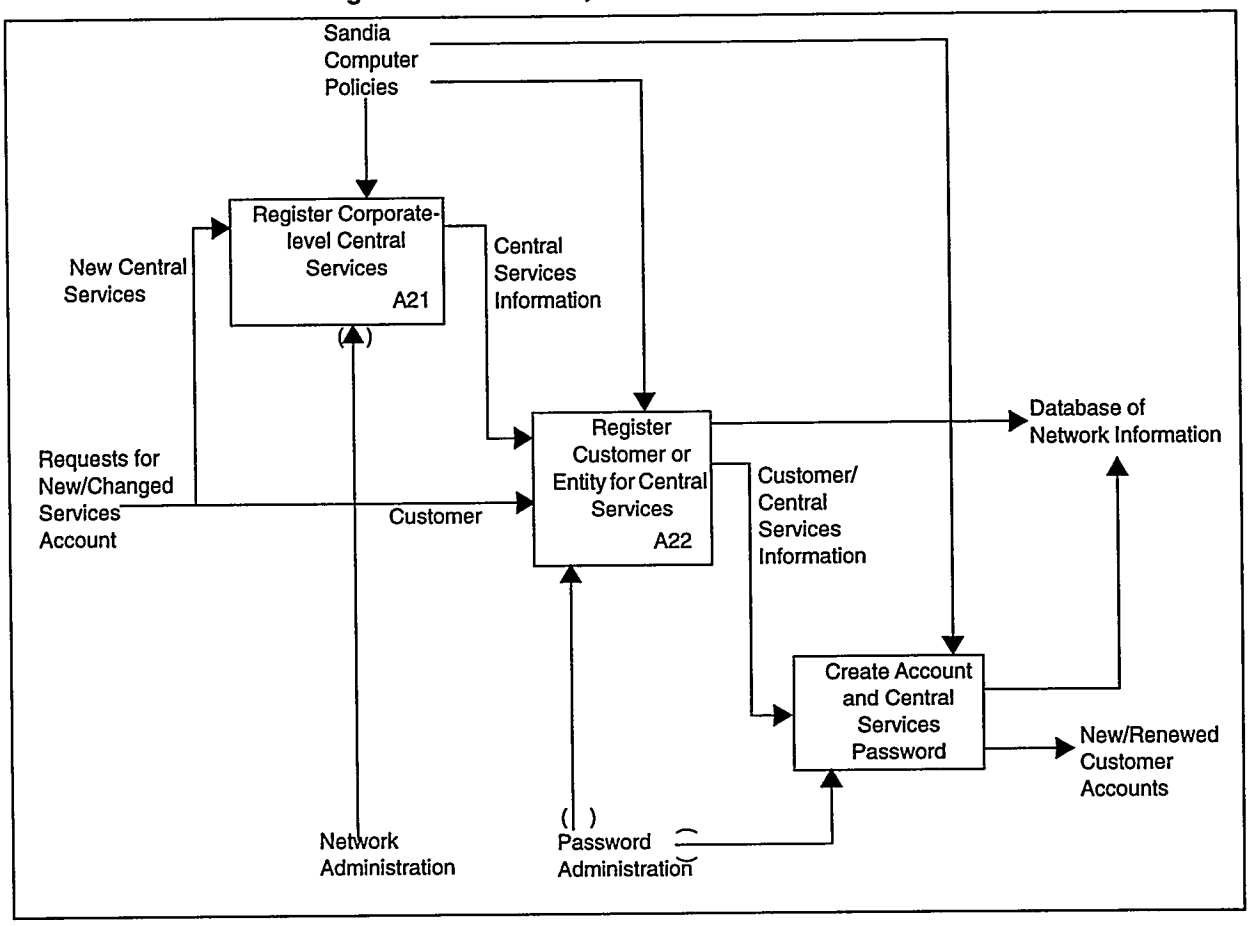
The A0 process shown here is decomposed into two subprocesses, A1 and A2. If an external entity was not noted with exact wording in the higher-level diagram, a tunnel marker displays around the arrowhead.

Figure 15 Node A1, Administer Customer Network Information



The A1 subprocess is further broken down into activities A11, A12, and A13. Each of these can be decomposed again until the granularity is such as to describe the software processes to be managed in writing and deploying the system.

Figure 16 Node A2, Administer Central Services Information



The A2 subprocess is further broken down into activities A21, A22, and A23.

Test Plan, Test Database, and Test Data

Deliverable 12

Test data may be acquired in the modeling phase as valid and invalid examples of each fact. The design of the test plan will validate the modeling examples, in addition to other tests as needed to check the algorithms and the output or results of the processes and the system. The test database will be a significant data set to establish and carry out the test plan. It must be a reusable baseline database, and it should be established early in the prototyping process.

The detailed specifications of screen functions and constraints for each process are written for the NWDB as numbered constraints for each screen. These constraints are essentially the test plan—to verify that the developer implemented every constraint as specified.

In addition, an experienced programmer will apply the testing technique called "error guessing." Without using any specific methodology, the veteran coder can "smell out" errors. Given a particular application, this tester can intuitively guess certain probable types of errors and test to expose these errors, if they are present (Reference 18). All tests will be run on the primary software platform (PC) and the other platforms prior to production.

The tester will document the results of all tests, using the Microsoft Word template (see "Test Incident Report" on page A-59).

Format and content of each report and datafile are validated by the recipients (customers).

Test Incident Report

tir#:
date:
platform:
application:
version:
function:
tester:
role:
retest date:
retester:
retest result:
legend:
severity level
0- informative
1- user interface not friendly (ex: buttons, menu not arranged well)
2- user interface is confusing (ex: directions not clear)
3- data is confusing (ex: information has to be derived)
4- data is incorrect
5- application response is inappropriate or incorrect
6- response is inordinately slow (ex: database access inefficient)
7- does not conform to software specifications
8- other
9- does not conform to datamodel
10- application aborts or does not return to user
****
incident#:
spec#:
error message:
severity:
description:
****
steps to duplicate error
incident#:
-
-
****
programmer's notes
incident#:
****
retest notes
incident#:
****

Test Data/Test Database Listing

Table 10 customer—Test Data

cust_id	stat_chg_agent_type	cust_status	stat_chg_date_time	stat_chg_agent	cust_mail_name	clearance_type	work_site_code	office_phone	office_fax	assigned_org	building_num	room_num	mail_stop	email po	in_hris_flag	log in_flag	cust_lookup_name
1	C	a	9/15/95 12:00:00 AM	1	David S Cuyler	Q	SA	000-0000	000-0000	00000	000	000	0000	PRINT	Y	Y	CUYLER, DAVID S.
2	p	a	10/15/95 12:00:00 AM	1	David P Duggan	Q	SA	505-844-9121	505-844-3333	13999	880	D555	0884	PRINT	Y	Y	DUGGAN, DAVID P.
3	p	a	10/15/95 12:00:00 AM	1	Donna B Black	Q	SA	505-845-9987	505-844-9344	04832	880	D99	0884	PRINT	Y	Y	BLACK, DONNA B.
4	p	a	10/15/95 12:00:00 AM	1	Mary F Johnson	Q	SA	505-945-8898	505-844-9344	04832	880	D99	0884	PRINT	Y	Y	JOHNSON, MARY F.

Table 11 account—Test Data

central_svc	login_name	acceptance_date	case	account_status	stat_chg_date_time	stat_chg_agent	card_serial_num	card_expire_date	set_up_cust_id	file_disp_flag	stat_chg_agent_type
LIS	dbblack	10/15/95 12:00:00 AM	0100589000	a	10/15/95 12:00:00 AM	1	0	10/15/95 12:00:00 AM	3	y	p
LIS	dscuyle	9/15/95 12:00:00 AM	3321400000	a	9/15/95 12:00:00 AM	1	0	9/15/95 12:00:00 AM	1	Y	p
LIS	mfjohns	10/15/95 12:00:00 AM	0100589000	a	10/15/95 12:00:00 AM	1	0	10/15/95 12:00:00 AM	4	y	p
SYBLIS	dscuyle	9/15/95 12:00:00 AM	3321400000	a	9/15/95 12:00:00 AM	1	0	9/15/95 12:00:00 AM	1	Y	p
Secure Cray	dduggan	9/15/95 12:00:00 AM	0100589000	a	9/15/95 12:00:00 AM	1	0	9/15/95 12:00:00 AM	2	Y	p

Table 12 login—Test Data

login_ name	login_ holder_t ype	login_ status	stat_chg_date_time	stat_ chg_ agent	unix_ user_id	acct_expire_date	stat_ chg_ agent_ type	login_ holder	clsfn_ level
dbblack	customer	a	9/12/95 12:00:00 AM	1	411	9/12/96 12:00:00 AM	p	3	s
dduggan	customer	a	9/12/95 12:00:00 AM	1	922	9/12/96 12:00:00 AM	p	2	s
dscuyle	customer	a	9/12/95 12:00:00 AM	1	1	9/12/95 12:00:00 AM	p	1	u
mfjohns	customer	a	10/12/95 12:00:00 AM	1	222	10/12/96 12:00:00 AM	p	4	s

Table 13 nwdb_user_grp—Test Data

nwdb_user _grp	grp_nm _ org	grp_nm_ phone	grp_nm_ mail_stop	grp_ca _org	grp_ca_ phone	grp_ca_ mail_sto p	grp_ca_ fax	grp_nm_ fax	grp_nm_ e_mail	grp_ca_ e_mail
LAN Points of Contact	13911	505-234-5677	1009	08534	510-294-2222	9001	510-294-9999	505-844-3333	dduggan@ sandia.gov	mjsimth@ sandia.gov
Password Administration	13316	505-234-5678	0661	08534	510-294-2222	9001	510-294-9999	505-844-3333	dduggan@ sandia.gov	mjsmith@ sandia.gov

Table 14 nwdb_user_grp_mbr—Test Data

nwdb_user_grp	mbr_cust_id
LAN Points of Contact	3
Password Administration	1
Password Administration	2
Password Administration	4

Table 15 cases—Test Data

case	statusedcase_status
0000000000	a
0100589000	a
0101010101	a
3321400000	a
8999111000	a
9999111000	a

Table 16 site—Test Data

site_code	describedsite_desc
SA	Sandia New Mexico
SL	Sandia California

Table 17 reference_record—Test Data

reference_record	last_cust _unix_id	last_entity_ unix_id	last_cust_id
1	985	444	4

Table 18 entity—Test Data

entity	login_flag
356	n
443	n

Detail Process Specifications & Algorithms

Deliverable 13

Processes, other than elementary processes and functions that models can generate, should be documented to the level of detail that would eventually assist in the generation of code. Several formats will suffice, until an automatic code generation tool is chosen or written.

NWDB Example

This is an optional deliverable, and the NWDB example did not supply this information. It did not have extensive process algorithms, and relevant material was documented in detail as processing constraints and presentation sets (see "Extended Presentation Sets" on page A-37 and "Extended Dependencies and Implementation Constraints" on page A-49).

External and Internal Interfaces

Deliverable 14

Most systems rely on data and controls from other systems, either by direct network links or file transfer. The interfaces must be documented to the level of detail required for future maintenance of the software. If the data must be converted for this application, a conversion map will show the input and transformation of each field, and the mapping to the new software.

The Conversion Map is optional.

This example extracts specific information from the NWDB. To view the complete deliverable for this subset of the NWDB, see the World Wide Web:

http:// sass902d.itd.sandia.gov/SILCExam.pdf

External and Internal Interfaces

The NWDB example includes these interfaces:

- *Inputs from Other Systems*—The Network Database receives, from Human Resources, basic data about customers who are registered in Human Resources (note that many customers are not registered in Human Resources), a list of valid contracts with begin and end dates, a list of valid buildings, a list of valid cases and case statuses, and the clearance level ("L" or "Q") of each cleared person.
- *Outputs to Other Systems*—The Network Database generates and sends Login Names to Human Resources to support e-mail, sends case charges for central service accounts, LAN connections, and CSU services to the Financial Information System, sends six types of files to the Domain Name servers to drive Internet traffic, and sends basic user, account, machine, and LAN connection data to the Trouble Ticket system for reference purposes.
- *Other Outputs*—The Network Database also generates about 30 other types of output, many of which are data files to feed various hosts and other network services.

The data must be converted for this application. The following conversion map illustrates the input and transformation of each field, and the mapping to the new software.

Table 19 Conversion Map

	Old Tables			New Tables		
Facts	Table(s)	Column Name of Sentence Subject	Column Name of Sentence Object	Table(s)	Column Name of Sentence Subject	Column Name of Sentence Object
A customer has a first name. On HR load, derive by parsing out name.	user_info rpt_perso		fname name	customer	cust_id	first_name
A customer has a middle name. On HR load, derive by parsing out name.	user_info rpt_perso		mname name	customer	cust_id	middle_name
A customer has a last name. On HR load, derive by parsing out name.	user_info rpt_perso		lname name	customer	cust_id	last_name
A customer is identified by a mail name. If the customer first, middle, or last name is changed, do the following: The customer's first name, a space, the first character of the middle name and a space (if there is a middle name), and last name should be concatenated to form the customer mail name.	user_info rpt_person		fname mname lname name	customer	cust_id	cust_mail_name

Table 19 Conversion Map (Continued)

	Old Tables			New Tables		
Facts	**Table(s)**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**	**Table(s)**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**
A customer is identified by a lookup name. If the customer first, middle, or last name is changed, do the following: The customer's last name, a comma, a space, first name, and if there is a middle name, another space, the first character of the middle name and a period should be concatenated to form the customer lookup name.	user_info rpt_person	[redacted]	fname mname lname name	customer	cust_id	cust_lookup_ name
A customer holds a clearance. Clearance code must be either "Q," "L," or "none." Clearance code should automatically default to "none" unless otherwise obtained from the security file.	clearance	[redacted]	clearance_type	customer	cust_id	clearance_type
A customer is typed by a customer type. Note: Customer type	user_info rpt_person	[redacted]	emp_type pertype nemtype	customer	cust_id	cust_type
[redacted]	user_info rpt_person	[redacted]	[redacted]	customer	cust_id	[redacted]
A customer works at a site, NWDB values not the same as HR. Need translate to HR codes. Must be upper case. Site is required for customers of type "sandian."	user_info rpt_person	[redacted]	site wrsitcd	customer	cust_id	work_site_code
A customer answers at a phone number. Phone numbers from HR should be stored without embedded dashes—just as a number.	user_info rpt_person	[redacted]	uphone ofcphone	customer	cust_id	office_phone
A customer receives FAXes at a phone number.	rpt_person	[redacted]	faxnum	customer	cust_id	office_fax
A customer works for an organization. A customer must work for an organization if the customer type is "sandian." Org. must be a five-digit number, with a leading zero as necessary. In the HR files org. has often not been zero-filled to the left. All org. numbers should be zero-filled to 5 places, e.g., "1" should become "00001," "1956" should become "01956." Also remove any section numbers, i.e., dashes and any numbers following the dashes.	user_info rpt_person	[redacted]	uorg asorg	customer	cust_id	assigned_org
A customer works in a building. Strip any leading "B" and any embedded spaces, blanks, "-" or "/."	user_info rpt_person	[redacted]	bld ofcbldg	customer	cust_id	building_num

Table 19 Conversion Map (Continued)

	Old Tables			New Tables		
Facts	Table(s)	Column Name of Sentence Subject	Column Name of Sentence Object	Table(s)	Column Name of Sentence Subject	Column Name of Sentence Object
A customer works in a room. Strip any embedded spaces, blanks, "-" or "/."	user_info rpt_person	[redacted]	room ofcroom	customer	cust_id	.room_num
A customer receives internal hard-copy mail at a hard-copy mail stop. Mail stop is required for customers of type "sandian." NWDB has 5 characters—need chop off 1.	user_info rpt_person	[redacted]	mailstop mailstop	customer	cust_id	mail_stop
A customer receives electronic mail at an E-mail post office. Don't load this from old NWDB files; this only comes in from HR anyway. Just load it anew on the first HR load to the new system.	rpt_person	[redacted]	emailpo	customer	cust_id	emailpo
A customer has an E-mail address. E-mail post office is required for customers of type "sandian." Create e-mail addresses whenever the customer's unclassified Login Name or E-mail Post Office changes as follows: if the unclassified Login Name is blank, blank out e-mail address. If the customer's unclassified Login Name is not blank and the customer's electronic post office is blank, take the customer's unclassified Login Name (such as "rsjoyce") and concatenate it with "@" and "sandia.gov" to yield a result such as "rsjoyce@sandia.gov." If the customer's unclassified Login Name is not blank and the customer's electronic post office is not blank, take the customer's unclassified Login Name (such as "rsjoyce") and concatenate it with "@," the customer's electronic post office (such as "snl2800"), a period, and "sandia.gov" to yield a result such as "rsjoyce@snl2800.sandia.gov."				customer	cust_id	email_addr
A customer may be registered in the HR system. This flag should be set to "Y" if and only if the customer appears in an HR file. This flag will automatically be set to "N" anytime a customer is added directly on-line, and then automatically set to "Y" only when and if they appear in an HR file.	- keep in future	derive on HR load		customer	cust_id	in_hris_flag
A customer is marked with a status. Note: Customer status	user_info rpt_person	[redacted]	status perstat	customer	cust_id	cust_status
A customer had their status last changed on a date.	user_info	[redacted]	datetrans	customer	cust_id	stat_chg_date_time

Table 19 Conversion Map (Continued)

	Old Tables			New Tables		
Facts	**Table(s)**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**	**Table(s)**	**Column Name of Sentence Subject**	**Column Name of Sentence Object**
A customer had their status last changed at a time.	- keep in future			customer	cust_id	stat_chg_date_time
A customer had their status last changed by an agent. whotrans is a Login Name. Some may be missing. Convert to customer ID for new system	user_info	[redacted]	whotrans	customer	cust_id	stat_chg_agent
none	user_info	acct_status		customer		
A customer is a subtype of login holder.	- by definition			login		
An NWDB user group may be contacted in California through a hard-copy mail stop.	- hand enter			nwdb_user_grp	nwdb_user_grp	grp_ca_mail_stop
A customer belongs to an NWDB user group. (load LAN POCs only)	- hand enter			nwdb_user_grp	nwdb_user_grp	nwdb_user_grp

List of Reused Functions

Deliverable 15

Sharable routines such as functions and methods will be documented. Those existing functions found in the repository will be listed, along with new sharable routines that are additions to the repository.

NWDB Example

Reused Object Types and Fact Types

1. HRIS information as shown in the conversion map (see "External and Internal Interfaces" on page A-64). In summary, the following HRIS tables are reused: case, building, clearance, contract, and basic person data.

Reused Functions

1. XVT/C function libraries

User Documentation

Deliverable 16

The purpose of user documentation is to present information in a concise, helpful, and organized manner that is necessary for accomplishing tasks. This resource includes information pertaining only to the software tool. Other documentation, such as policies, procedures, and tutorials, is outside the scope of the software project. The user documentation set includes:

- System Overview
  - Purpose
  - Scope
  - Intended audience
- Glossary of Terms
  - Field definitions with examples
- Task Procedures
  - Steps required to accomplish a task or solve a problem
- Quick Reference
  - Mechanics of operating the system, such as navigating, editing, entering special characters, searching, keyboard shortcuts
  - Structure of the user interface, such as flow or order of screens, general explanation of screen design such as menu bar symbols
- FAQ (Frequently Asked Questions)
  - Simple troubleshooting procedures
  - Pitfalls to avoid
  - Who to call for help
  - Documented bugs
  - Enhancements for future versions

Information pertaining to field domains, such as field length and type of characters allowed, should be embedded in the screen design.

Documentation Process

Deliverable 16

The following steps are provided in lieu of an extensive NWDB example.

1. Decide which documentation tool(s) will be used, based on the chosen platform(s) and database. The selected tool(s) will determine the documentation style. If necessary, a tool should be selected that will accommodate multiple platforms so only one version of the documentation is managed. The software developer shall inform the writer of any special considerations for interfacing the completed documentation with the source code for on-line help.
2. Match the documentation set to the level of user audience, as defined in the Problem Statement (Deliverable 1).
3. Decide which format(s) will be used to present the information. The on-line help format will always be employed. Other formats, such as hard copy documentation or HTML on the World Wide Web, may be required. If other formats are required, they should be automatically produced from the original documentation, so only one version is managed.
4. Create the System Overview, based on the Problem Statement (Deliverable 1).
5. Automatically extract object definitions and examples from the Conceptual Information Model (Deliverable 3) to create the Glossary of Terms. Considering the level of user audience, it may not be necessary to define all of the objects.
6. Automatically extract procedural information from the PAST Matrix (Deliverable 6) to create the Task Procedures. This should be combined with information presented in the Extended Presentation Sets (Deliverable 8) and the Extended Dependences & Implementation Constraints (Deliverable 10) to completely define the steps for each task.
7. Create the Quick Reference, which will include mechanics of the operating the system and the structure of the user interface. Some of this information will be found in the Extended Presentation Sets (Deliverable 8) and the Extended Dependencies & Implementation Constraints (Deliverable 10). Other information will be obtained from the software developer. The standard GUI template or guideline may also have a Quick Reference or provide information for the application reference.
8. Create the FAQ (Frequently Asked Questions) from various sources, such as the software developers, system testers, help desk, and users of the system. This document should be updated frequently. The system should be designed so that the Corporate Computer Help Desk can respond to the user questions first and refer calls to application maintenance personnel second. Access to system change request procedures and browsing to check for request status is one way to provide information on bugs and future enhancement information.

Future State Goal: Design user interface so that no additional user documentation is required.

Test Results

Deliverable 17

Following the test plan, a tester documents results in a standard template. If present, this would include the results from testing complex functions and algorithms.

The tests were performed as specified in the test plan and documented in the standard Test Incident Report template. These tests concentrated on the screen/data aspects of the system. Functions and algorithms are not present in a complex form in this system, so computerized testware design was not used. This would be required in other types of software.

Test Incident Report

tir#: 1000
date: 06/27/95
platform: pc
application: nwdb.exe
version:
function: NWDB User Groups
tester: fay
role: nwdbdev
retest date:
retester:
retest result:
legend:
severity level
0- informative
1- user interface not friendly (ex: buttons, menu not arranged well)
2- user interface is confusing (ex: directions not clear)
3- data is confusing (ex: information has to be derived)
4- data is incorrect
5- application response is inappropriate or incorrect
6- response is inordinately slow (ex: database access inefficient)
7- does not conform to software specifications
8- other
9- does not conform to datamodel
10- application aborts or does not return to user
****

incident#: 1
spec#: none
error message:
severity: 0- informative
description: I was able to invoke two copies of this screen, and query "User Group" == 'Test Group' on both screens. I then changed the "mail stop" on one screen . The second screen was not refreshed with the new information. This may not be in error but may confuse the user. (should the user not be allowed to query the same entity on two different screens?)

incident#: 2
spec#: gen-70
error message:
severity: 7- does not conform to software specifications
description: The system did not request a confirmation for a delete (see steps to duplicate error below)

incident#: 3
spec#: none
error message: 'wugrps_pbdelete' dependent foreign key constraint...constraint name = 'central sv952442517'...command has been aborted...sql server error...failed deleting nwdb_usedr_grp
severity: 1- user interface not friendly
description: the above error messages appeared when I attempted to delete a user group.
(see steps to duplicate error below)

incident#: 4
spec#: usergrp-30
error message:
severity: 7- does not conform to software specifications
description: mail stop is required to be 4 digits upon entry. I entered 2 digits and the system left filled with zeroes instead of displaying an error message
(see steps to duplicate error below)

incident#: 5
spec#: usergrp-40
error message:
severity: 7- does not conform to software specifications
description:: Since telephone numbers allow '-' and '(' and ')' I was able to input a value that only consists of '-' and '(' and ')' . ex: (----)--
(see steps to duplicate error below)
****

steps to duplicate error

incident#: 2
- add a test row
- query the row to confirm that it was added
- delete the row
- notice that there is no confirmation message
- query to confirm that the row was in fact deleted

incident#: 3
- query NWDB User group == 'Test Group'
- select delete
-notice the succession of error messages

incident#: 4
- add or change the "mail stop" as a two digit value
- click on "add" or "change"
-notice that the value is left padded with zeroes

incident#: 5
-" add" or "change" a test "user group"
- enter "voice","fax" values as '(-----)--'
- click on "add" or "change"
- notice the successful update
****

programmer's notes

incident#:

****

retest notes

incident#:

****

Appendix B

Software Products Used for NWDB Example

To produce the deliverables shown in Appendix A, the SILC team used a variety of software products. Table 20 lists products by deliverable. We used the word processors Microsoft Word and FrameMaker interchangeably.

Note: The "Generated" column shows deliverables that can be generated from the business rules (Deliverable 2).

Table 20 Software Products Used for NWDB Example

Deliverable	Generated	Tools
1 Problem statement		Microsoft Word Design/IDEF
2 Business rules		InfoModeler Corporate Repository and DBArtisan
3 Conceptual information model	✓	InfoModeler and filter
4 Logical information model	✓	InfoModeler and ERWIN/ERX
5 Elementary processes	✓	InfoModeler and DBArtisan
6 Presentation sets		MacDraw Pro and Microsoft Word
7 CRUD matrix	✓	Corporate Repository; Microsoft Excel
8 Extended presentation sets		MacDraw Pro and Microsoft Word
9 DB schema	✓	InfoModeler and ERWIN/ERX
10 Extended dependencies and implementation constraints		MacDraw Pro and Microsoft Word
11 IDEF0 decomposition		Design/IDEF
12 Test data, test plan, and test database	✓	GQL or DBArtisan

Table 20 Software Products Used for NWDB Example (Continued)

Deliverable	Generated	Tools
13 Detail process specifications and algorithms (may include design decisions)		n/a
14 External and internal interfaces (may include conversion map)		Microsoft Word
15 List of reused functions		Microsoft Word and Corporate Repository
16 User documentation		Microsoft Word Doc-to-Help
17 Test results		Microsoft Word

Distribution:

MS 0100 Document Processing, 7613-2
For DOE/OSTI (2)
MS 0619 Print Media, 12615
MS 0622 Hank Witek, 4606
MS 0622 L. Herb Pitts, 4400
MS 0629 Paul D. Merillat, 4800
MS 0629 Al Alvarado, 4800
MS 0629 Carol Jones, 4800
MS 0630 Michael J. Eaton, 4010
MS 0638 Dwayne Knirk, 12326
MS 0661 David K. Cuyler, 4923
MS 0661 Donna S. Eaton, 4816 (40)
MS 0661 Elisa Kephart, 4816
MS 0661 Gary Rivord, 4816
MS 0661 Irene Thurston, 4816
MS 0661 Joe Schofield, 4816
MS 0661 John Hatley, 4816
MS 0661 Michael H. Pendley, 4612
MS 0661 Shelley Eaton, 4816
MS 0661 Charlene Strickland, 4816
MS 0801 Melissa J. Murphey, 4900
MS 0803 John F.Jones, 4600
MS 0803 John K. Sharp, 4600
MS 0805 Dorothy S. Rarick, 4911
MS 0805 Nancy Marsh, 4911
MS 0806 Michael O. Vahle, 4616
MS 0806 Leonard Stans, 4616
MS 0807 R. Michael Cahoon, 4918
MS 0809 Melville E. Mefford, 4913
MS 0811 C. Douglas Brown, 4621
MS 0811 Lucien Don Daigle, 4815
MS 0812 Bill Mertens, 4923
MS 0812 Ronald Hall, 4923
MS 0813 Jim Hamilton, 4412
MS 0813 T.C. Adams, 4913
MS 0898 R. Dennis Rowley, 4605
MS 0899 Technical Library,4414 (5)
MS 1090 Grant C. Claycomb, 4814
MS 1090 Joseph A. Ruggles, 4812
MS 1090 Mary Lynn Clark, 4815
MS 1090 Scott Joyce, 4816
MS 1090 William D. Swartz, 4411
MS 1098 Andrea Cassidy, 4403
MS 1098 Jim Stromberg, 4815
MS 1098 Thomas L. Ferguson, 4813
MS 9018 Central Technical Files, 8523-2

Org.	Bldg.	Name	Rec'd by	Org.	Bldg.	Name	Rec'd by

Sandia National Laboratories