8th International Conference on Advanced Science & Technology, March 1992

A Process-Modeled System for Designing the 5ESS® Switching System Database

Eisuke Muroga, Yao-Nan Lien, and Ronald J. De Lange

AT&T Bell Laboratories
200 Park Plaza
Naperville, IL 60566-7050

ABSTRACT

Spartan is a data design automation system for the 5ESS Switch real-time memory-resident database. The 5ESS Switch architecture and application software requires the database to be designed with speed, space efficiency, and correctness. The database design process is comprised of five unique phases, each with its own requirements: Investigation, Design, Engineering, Approval, and Verification. The design of Spartan incorporates distinctions between the process phases, yet provides a single environment to support all the phases. The Spartan System has been implemented and adopted, resulting in an improvement of 5ESS Switch database quality, and a reduction of its design time interval and cost.

1. Introduction

The software to process telephone calls on a 5ESS Switch is highly dependent on a real-time, memory-resident database [Barc82, Dela85]. This database allows each switch to be customized for a particular telephone office. The performance and the reliability of the switch is highly dependent on the correctness, speed, and efficiency of this database.

The 5ESS Switch database uses the relational database model. Any semantic rule covering the legal values in the database can be defined by *population rules*, including the cross-references between office features, views, relations, attributes, and domains.

The unique aspects of the 5ESS Switch database necessitate the creation of a specialized data design support system:

1. The database is extremely large and complex, comprised of about 1000 relations and views, as well as hundreds of attribute domains. In addition, there are thousands of population rules used to enforce the integrity of the relations in the database.
2. Stringent memory space and speed requirements to support a real time system (tuple access time is in the μs range) makes access and storage tuning necessary.
3. The database is distributed among the multi-processor architecture of the 5ESS Switch.
4. The database is constantly evolving, and all structure changes must be coordinated with many other changes.

The data design and development process in 5ESS has several phases: Investigation, Design, Engineering, Approval, and Verification. Before changing the database structure or population, a data designer needs to perform an initial investigation of the existing designs, since comprehending a database of 1000 relations is a non-trivial task. Furthermore, with several hundred engineers simultaneously developing 5ESS Switch software, it is necessary for all data designs to be coordinated, ensuring that there are no conflicts or overall

® Registered trademark of AT&T.

inefficiencies. This process was previously done manually, aided by a collection of tools supporting isolated tasks. Thus, this process was time-consuming and required a verification process that was highly dependent on human experts to ensure its quality.

The objective of the Spartan System is to provide an integrated design environment that encompasses all data design and development phases to provide guidance, assistance, and automation for the data designer, resulting in efficient and error-free changes to the 5ESS Switch database. The Spartan System provides a menu-driven interactive environment with a similar "look-and-feel" user interface to support all five phases. It also provides a consistent working environment and a mechanism to store and share information between the five phases. This coherence ensures that data designers can work under the same environment as other data designers, whether working on the same project or on different projects. Spartan also offers a wide selection of previously-unavailable tools that support data design and testing.

2. The Spartan System

2.1 General Architecture

The Spartan System was conceived as a system to encapsulate the entire data design process, but it was found that each phase of the process had unique needs. Thus the design of Spartan incorporates distinctions between the process phases, yet provides a single, consistent environment for all the phases.

The architecture of the Spartan System is designed to support separate, yet interdependent process phases. The key feature of the Spartan architecture is how various data is stored, accessed, and shared by each of the modules. The data in the private databases are used only within a particular module, whereas the shared databases are used not only as a common repository of data, but as the underlying link between the modules.

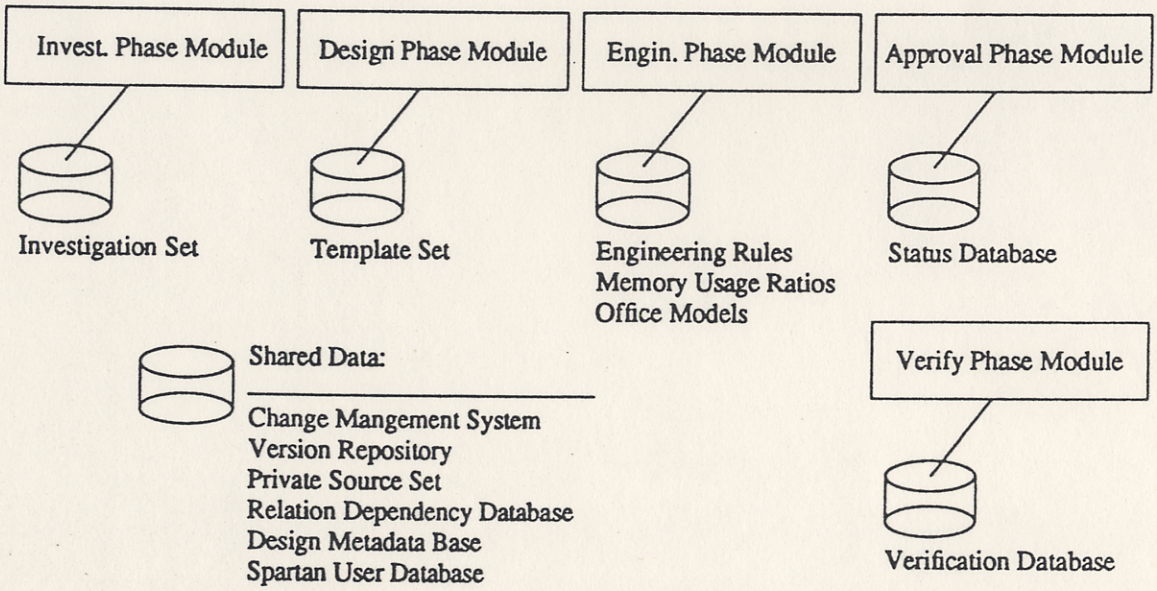

Figure 1. Spartan Database Architecture

The shared databases are:

1. *Change Management System* - a database system that controls all changes to 5ESS Switch software and data-definition source code [Basi85].

2. *Version Repository* - a read-only repository containing snapshots of a complete version of 5ESS Switch source code [Basi85].
3. *Private Source Set* - a set of source files privately held by each data designer.
4. *Design Metadata Base* - a database containing metadata describing the designs contained in the Version Repository.
5. *Relation Dependency Database* - a database containing dependency information between all relations.

2.2 Investigative Phase

The Spartan System supports the Investigation phase by providing access to information on past data designs. The importance of providing this access to the designs is twofold:

1. The data designer can reuse parts of existing designs.
2. The data designer can see what other parts of the database will be affected by any changes the designer makes.

A database of information regarding all existing relations and domains is maintained. To assist the data designer in browsing through the existing data designs, the Spartan System incorporates a set of investigation tools.

There are three basic tools that allow the data designer access to the *Design Metadata Base*. These include two browsers: one for relations, and one for domains. Each browser can display either relations or domains that meet a set of criteria selected by the data designer. The predefined criteria can be chosen and combined from a menu of criteria, much like a query. The set of relations or domains that result from the query selections is kept in a temporary private database, called the *Investigation Set*.

The third investigation tool is a *relation dependency analyzer* (RDA). A change made to a relation is likely to impact another relation that is dependent on the first relation. Thus, the data designer must find all relations that are dependent upon the relation being modified. It would be a complex and time consuming task to identify all inter-relation dependencies manually. Spartan can construct for the data designer the entire transitive closure of the dependency by abstracting the dependency relationship from population rules. This dependency information is stored in the *Relation Dependency Database*. Users can choose to browse for the dependency interactively through the RDA or to see a set of dependencies as a dependency graph.

2.3 Design Phase

The Spartan System provides an interactive development environment to privately develop and analyze the physical and logical designs of new or modified relations. Each new or modified relation in the switch database must be considered in terms of where the data is to reside (which processors), how the data is to be accessed, and how the data is to be stored.

Existing relation or domain physical definitions can also be modified. Spartan provides a set of templates to help data designers quickly construct an initial physical design for new relations or domains. Spartan provides the mechanism to use a copy of a relation or domain definition file from the Version Repository. The data designer is placed in an interactive mode using the Relation Definition Language (RDL) to update the template copy and manage it in the data designer's *Private Source Set*.

After the data designer has created new domains or relations and modified copies of any existing data structures to be changed, the *Private Source Set* is then syntax checked and a private *Design Metadata Base* is constructed or updated to provide a platform to describe the data's logical relationships.

RDL syntax and physical design engineering rules are referenced from the *Engineering Rule Database* stored within the Spartan System. A syntax directed interface/editor is provided that has knowledge of all

relations and domains defined in the database via the private *Design Metadata Base*. This interface/editor is used to define logical data constraints and inter-relation dependencies.

Once the basic design has been privately implemented, design documentation can be automatically generated and reviewed by the design team. By providing these capabilities in a private environment, many different design alternatives can be quickly defined and considered.

2.4 Engineering Phase

The Spartan System Engineering phase is designed to support data designers in optimizing the physical aspects of their designs for speed and space efficiency.

Spartan provides an interactive ability to forecast memory usage based on the number of anticipated tuples for a new relation. Alternatively, Spartan can estimate memory usage based on simulated or real data provided as input. This can be especially useful when modelling different types of switch configurations. Using simulated or real data as input, Spartan can also characterize the populated relation in terms of data clustering around logical keys.

For modified relations, the number of anticipated tuples and/or the memory utilization rate (MUR) calculated from data collected during the 5ESS Switch's transition from one software release to another is used to forecast memory usage. The *Memory Utilization Ratios* database is used to obtain the MUR data.

Each relation structure definition can be viewed on the data designer's terminal with the bit layouts of each attribute displayed. Attributes can be moved around interactively to find the minimum memory usage. Spartan can also automatically suggest a relation structure layout that minimizes memory usage. By using the private *Design Metadata Base* and *Private Source Set*, the engineering aspects of the design can be trialed to find the optimal physical design.

The data designer can use default values or specify a sub-set of engineering characteristics for a particular relation. Spartan, referencing the *Engineering Rules* database, will suggest the optimal relation access method to be used by the 5ESS Switch Data Base Manager. Spartan can also summarize existing engineering for similar relations using the *Design Metadata Base*, which can be used to model a new relation.

2.5 Approval Phase

Since the 5ESS Switch software is driven by the database, any change to the database potentially impacts many software modules that are owned by different organizations. Coordination among the data designers and the impacted organizations is essential to maintain consistency between the database and all associated software modules. Spartan supports the approval phase by maintaining a database of change information, responsible parties, etc. A common interface is also provided, by which all database changes can be coordinated.

2.6 Verification Phase

Under the current 5ESS Switch data development process, the integrity constraints of each relation are defined in a set of population rules using PRL [DiPi88]. These population rules can be used for data integrity maintenance as well as for documentation described as follows:

1. All 5ESS Switches and all testing laboratories of the same software release populate their databases, called *Office Dependent Data* (ODDs), according to the same set of relation definitions and population rules [Baue85].
2. After an ODD is populated, an ODD verification process is carried out to check its integrity using a executable set of population rules.

Although many commercial databases use a query-language-based integrity constraint system (e.g. SQL)

[Date89, Codd90], the specialized constraint language PRL was created to better handle the needs of the 5ESS Switch database. The constraint language is used to define thousands of rules, making it a very large and complex rule-set. The PRL is compiled into an executable constraint-checking program, allowing a populated database to be checked against these rules.

To verify the correctness of an ODD, the Spartan System is invoked to check every tuple of each relation, which may be replicated in multiple processors. The Spartan System will flag an error if an inconsistency is found between the tuple and the corresponding rule. After the check is completed, the errors found must be analyzed to determine their root causes, and then be corrected. There are four possible causes:

1. *population rule error:* the population rule itself is not correct.
2. *data entry error:* the tuple is incorrect due to an error introduced during the data entry process.
3. *data administration tools error:* the tuple is incorrect due to an error introduced during an off-line data modification or creation process.
4. *simulated error:* the error is embedded into the database intentionally for testing purpose.

3. Concluding Remarks

Before the Spartan System was adopted, the data design and definition process was minimally documented and supported. Each developer had to discover what the processes were and what engineering guidelines were applicable to their design. Data design was complex, tedious, and very time consuming.

By formalizing the process via the Spartan System, the processes to design data are well defined, with known entrance and exit criterion. Providing a common framework and languages specific to data design via *Private Source Sets* facilitates private, quick, interactive definition as well as tools that ensure the quality of the 5ESS Switch database and its data. The resulting automated code generation, verification processes, and common interfaces have significantly reduced the design interval and costs, and have improved the quality of data designs much earlier in the software development process.

There are efforts to fully automate the Engineering phase, so that the data designer need only specify some high-level parameters. Ultimately, these engineering parameters also will be automatically generated for the data designer.

References

[Barc82] D. K. Barclay, E. R. Byrne, and F. K. Ng, "A Real-time Database Management System for No. 5 ESS," Bell System Technical Journal Vol. 61, No. 9, Pt. 2, Nov. 1982.

[Basi85] R. G. Basinger, J. A. Herndon, B. Kaskey, J. A. Lindner, and J. M. Milner, "The 5ESS Switching System: System Development Environment," AT&T Technical Journal, Vol. 64, No. 6, Pt. 2, 1985.

[Brai90] R. M. Braida, B. N. Desai, P. Fu, and D. W. Walker, "RDL Reference Manual and Design Guidelines for Operational Views," AT&T Bell Laboratories internal document, 1990.

[Davi81] J. H. Davis, J. Janik, Jr., D. Royer, and B. J. Yokelson. "No. 5 ESS - System Architecture," Proceedings of International Switching Symposium, 1981.

[Dela85] J. P. Delatore, R. J. Frank, H. Oehring, and L. C. Stecher, "The 5ESS Switching System: Operational Software," AT&T Technical Journal, Vol. 64, No. 6, Pt. 2, 1985.

[DiPi88] J. E. Di Pirro, "Population Rule Language (PRL) Reference Manual," AT&T Bell Laboratories internal document, 1988.