

APPLYING SOFTWARE ENGINEERING PRACTICES TO THE DEPLOYMENT OF
NORTEL NETWORKS' PRESIDE FOR TELPHONY NETWORKS

by

Scott B. Bradley

s.bradley@computer.org

In partial fulfillment of requirements for

TS 5130: System Development Theory and Practice

Capella University

Sept. 18, 2003

Abstract

Nortel Networks' Preside for Telephony Networks is a complex distributed system used for the management of voice networks. Preside installations are generally described as long, complex, and inefficiently procured. In the disciplines of systems and software engineering, there are a number of tools available for addressing issues of development complexity and efficiency. By approaching system deployment as an aspect of the overall development process, this study applies those tools to the installation process and presents new approaches to the task of system delivery to the customer. The specific tools to be considered in this study are value-based engineering and requirements triage. The application of these engineering tools will demonstrate that Preside deployment would benefit from modularized installation and testing, and phased delivery.

Table of Contents

Executive Overview	4
Technical Overview	5
Overview of the Installation Process	7
Engineering Analysis	8
Value based engineering in system deployment	8
Requirements triage for phased deployment	14
Conclusion	17
References	19

Applying Software Engineering Practices to the Deployment of Nortel Networks' Preside for Telephony Networks

Executive Overview

The Preside system was originally designed and built ad hoc by Nortel on a large customer site in order to meet the requirements of that particular customer. Preside is inherently modular in its design as a synthesized collection of programs and processes, and so appears to be a flexible and adaptable system by nature. However, the flexibility of Preside from the view of systems design has been imposed on the system post hoc, rather than designed into the system. As a result, each new installation is a new exercise in re-engineering and adapting the system to new customer requirements. The result is one not of building a new system according to a requirements specification but one of shaping requirements to fit into the flexible boundaries of the system, and in turn molding that system to fit those requirements.

Significant to understanding the complexity of Preside deployment is noting the relation of Preside engineering methodologies to recent software industry trends toward COTS (Commercial Off-The-Shelf) and middleware technologies. These technologies and their growing popularity are introduced in the *Computer* article "Rigorous evaluation of COTS middleware technology," in which the authors state:

Middleware refers to a broad class of software infrastructure technologies that use high-level abstractions to simplify construction of distributed systems. Over the past decade, the adoption of commercial off-the-shelf middleware products across the software industry has gathered significant momentum. (Gorton, 2003, p. 50).

Running counter to this technological trend toward high-level abstraction, however, is the design approach behind Nortel's Preside for Telephony Networks. Rather, Preside has been engineered close to the hardware for which it was designed ad hoc, in turn requiring that new deployments of Preside be implemented within a similar hardware environment, which must be integrated into the customers' existing networks. The performance and flexibility expectations for the Preside system, however, have not been diminished due to this low-level design mentality behind the system. Rather, Preside has been expected to fit a wide variety of system solutions, each in turn requiring special consideration for the needs of the local system.

While often marketed as a turn-key system, Preside does not fit into the category of COTS or middleware software. Rather, it is a complex system built of a select combination of hardware and software which is integrated into the customer's existing network. Furthermore, this integration requires extensive custom configuration for each site, resulting in a re-engineering of the system from existing components for each new installation. Preside is an ad hoc solution which has consequently been molded into a one-solution-fits-all application, treated as a middleware solution, but in fact far from being an abstraction-oriented middleware product. Thus, the turn-key marketing approach has been a misleading one which has also led to miscommunications and misunderstandings on the customer's part regarding the nature of

Preside installation. Preside system installations generally run schedules into the months rather than days or weeks, and often remain incomplete by the time the installer leaves the customer site. Much of the system configuration and remaining installation activities are done remotely by the Preside support team, or by the customer's own system administrators.

The assessment above indicates a number of reasons for placing value in a prospective engineering analysis of the deployment of Preside for Telephony Networks. Systems and software engineering teach us that deployment is part of the development process. Some contemporary engineering methodologies even centralize deployment by involving the customer through the development and delivering the product in distinct phases. Some of the reasons that Preside deployment would benefit from an engineering analysis include:

- The inherently complex nature of a Preside installation.
- The tendency for miscommunications with the customer regarding the nature of Preside installations.
- The current non-ideal length of most Preside installations.
- The incompleteness with which Preside installations are often carried out.

The bottom line for the justification of an engineering analysis of Preside deployment comes down to value: The prospect of increased value for both Nortel and Nortel's customers. Potential benefits of this analysis, if practically applied to deployment of Preside, include:

- Simpler installations.
- Greater communication and understanding between Nortel and its customers.
- Reduced deployment schedules.

Working with this potential in mind for increasing the value that Preside deployment might provide to Nortel and to Nortel's customers, this study will unfold by presenting an overview of the Preside system and its installation, and by making a high-level, preliminary engineering assessment of areas for improvement of Preside deployment. While the Preside system is laden with system complexities, most apparent during the deployment stage, such complexity is not a necessity. As Brooks (1995) states, "complexity is the most serious inherent difficulty, but not all complexity is inevitable." (p. 211). By presenting an analysis of the Preside system, this study will serve to begin the process of system simplification through the use of such common engineering tools as modularization and development phasing.

Technical Overview

The system of concern in this engineering analysis is a voice network management system called Preside for Telephony Networks. This system is used to gather data from the telephony network and to present it in the form of various graphics interfaces and reports for the purpose of

network management and traffic and data analysis. Figure 1 illustrates the Preside system in its basic or standard configuration at the level of the system's primary physical components.

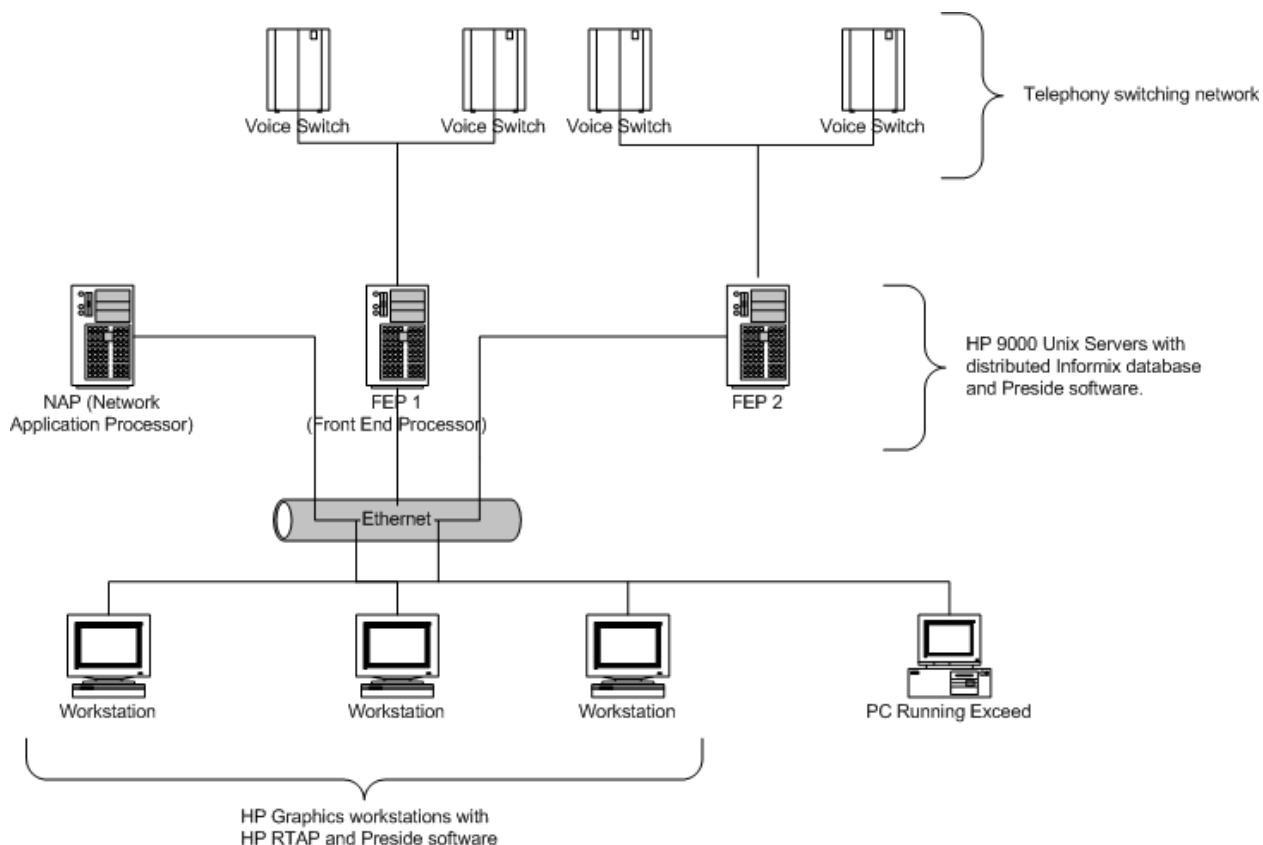


Figure 1. Nortel's Preside for Telephony Networks

This system operates as follows. The voice switches (typically Nortel's DMS or Meridian switches) send out ASCII data that includes alarm events, logs, and traffic statistics. This data is gathered by the management system and is parsed for useful information, which is in turn placed in a variety of tables in the Informix database. In addition, an HP real time graphic application called RTAP is integrated into the system for the purpose of displaying alarms to the users, typically in a 24/7 operating environment.

The role of the installation engineer on customer site typically includes:

- Act as on-site contact for Nortel to the customer
- Handle the installation and configuration of new system related hardware into the customer's existing network. Equipment typically includes HP servers and workstations. Configuration includes TCP/IP configuration and troubleshooting.

- Plan and install the operating system which is the HP-UX brand of the UNIX operating environment.
- Plan and implement file system sizing according to the individual machine needs and the needs of the overall system.
- Install third party software. This includes planning the sizing of the Informix database, as well as installation of Informix and initialization of the database. HP RTAP is also installed, with its own real-time database on each user workstation.
- Installed the Nortel software and provided necessary feature configuration, and site specific custom configuration.
- Test the system manually before hand-over to the site administrator.

Overview of the Installation Process

The following steps outline the primary activities that occur during a Preside installation. Each step varies in complexity for every project and may require a number of days, or even weeks, for completion. These steps are:

1. Setup hardware
2. Install Unix
3. Setup Informix database
4. Install HP RTAP
5. Install Preside
6. Run acceptance testing

These primary installation activities are shown here not only to demonstrate the activities of the installation engineer on customer site. Important for the present study, this high-level view of installation activities demonstrates a basic breakdown of the major components in a Preside system. Those primary components are: the system hardware, the operating system (HP UNIX), the distributed database (Informix), the real-time graphic interface (HP RTAP), the Preside software itself. The fact that the activities of the installation process itself coincide with the primary system components will be instrumental in analyzing the system from an engineering perspective and for demonstrating methods for improving the overall deployment process. This assessment follows with an engineering analysis in two parts. The first of these will be a modular analysis according to a method of value-based engineering, while the second will approach modularization by the methodology of requirements triage.

While Figure 1 above provides a good picture of the overall physical system of Preside, it provides little means for an analysis of the system along the lines of software functionality. The architecture of the software system is not the hardware configuration itself, rather, “the software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.” (Bass 2003, p. 21). For this reason, the analysis below will approach the system from the perspectives of software functionality and data flow. The result will be a stronger evaluation of the overall Preside system from the perspective of software engineering.

Engineering Analysis

The primary analysis in this study will proceed by the use of existing software engineering tools toward the end of simplifying and improving the Preside deployment process. A valuable tool available to software systems developers is that of staged or phased delivery. According to McConnell (1998), some of the benefits of staged product delivery are the following:

- Critical functionality is available earlier
- Risks are reduced early
- Problems become evident early
- Status-reporting overhead is reduced
- Staged delivery makes more options available
- Staged delivery reduces the possibility of estimation error
- Staged delivery balances flexibility and efficiency (pp. 55-56)

If we are to see these benefits work in the deployment of Preside, we must answer the question: How do we go about creating a phased deployment methodology out of this complex system? The following sections will address this question with the approaches of value-based engineering and software triage.

Value based engineering in system deployment

In their “Value-Based Software Engineering: A Case Study,” Boehm and Huang (2003) present the idea of an earned-value system of software development. By developing within the value based framework, developmental accomplishments are measured by the return value they bring to the company. The major tenets of this system are presented below:

1. Develop a set of tasks and associated budgets and schedules
2. Assign each task an earned value (EV), usually based on its budget
3. As the project proceeds, review these quantities:

- a. The budgeted cost of work scheduled
- b. The budgeted cost of work performed
- c. The actual cost of the project

Project progress is then evaluated based on these value oriented analyses (Boehm, 2003, p. 34). The significance of this engineering approach for the current high-level study of Nortel's Preside is the application of value based assessment for the purposes of phased deployment. By evaluating the valued significance of the Preside components, we can, in turn, place a value on the deployment of those components and further shape our phased deployment model. Such an approach is significant for a rollout that meets customer requirements over a series of deployment and delivery phases to culminate in a complete and satisfying product.

Ignoring the significance of the value of individual components of the system could be detrimental to the overall customer satisfaction of the system in any phased deployment. State Boehm and Huang, "Given that today's software has a major influence on system costs, schedule, and value, and that software decisions intertwine inextricably with system-level decisions, we can no longer afford to follow [a value-neutral approach]." (2003, p. 34) Applied to deployment of Nortel's Preside, the software components that make the overall system work can be seen to bring varying degrees of value to the overall value of the complex system.

Furthermore, since Preside is itself only one part of an overall, and generally complex and valuable, telephony network, albeit a valuable part as the management system of that network, the value contribution of the Preside components can be extrapolated to be understood as contributed value of the overall networked system. The high dollar value and revenue potential, for a telephony network provides strong incentive for evaluating Preside deployment from a value-based perspective. This is true even for the smallest networks such as the Spanish startup Jazztel, or the campus wide telephony system of Saudi Aramco, but is even more true for such large contracts as Worldcom and Sprint with dozens or more telephony switches over wide-spread geographic areas. For such larger systems, the incentive is even greater to evaluate the Preside system from a value-based perspective since, while the budget for Preside becomes a smaller factor in the budget of the overall system, the added value can be shown to increase dramatically.

As an exercise in examining the potential for the modularization of system deployment along the lines of component value, we will examine the difference between a high-level view of Preside as a black box, or turn-key system, and as a component-built system made up of an Informix database, an RTAP system, and the Nortel Preside software. This second view of the system will give us insight as to how the overall system might be seen in terms of its component values, and in turn how deployment might be phased accordingly.

To begin with, the standard customer view of the Preside system might be modeled something like the illustration shown in Figure 2. In this turn-key view of the system, Preside is a black box in relation to the overall telephony network and is merely understood by the end user as the means of accessing switch data. This view is appropriate as an executive view of Preside in relation to the overall telephony network and is one which places high value on the raw data

which is streamed from the switch and includes the various logs and reports important to understanding the switch activities.

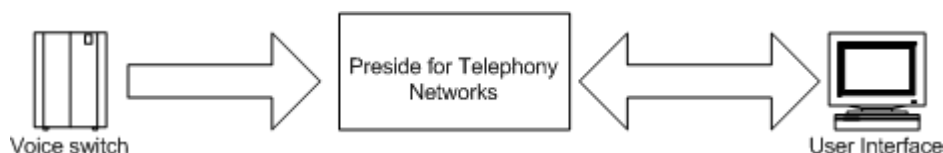


Figure 2. High-level of Preside within the overall telephony network

From an executive level view of the overall system, this black-box oriented view of Preside might be appropriate given the understanding that the switch data is the highest value contributor in the overall system equation. A higher level view would, in fact, reveal that it is not the switch data itself, but the traffic which is carried over the switches which is the highest value contributor to the overall system. It is, after all, the telephony traffic, not the related data, that is the profit earning “product” of the overall system. However, within the scope of our current assessment, neither the traffic, nor its related raw data can demonstrate the contributive value of the Preside system. Both of these components of the overall network system are assumed components for a Preside installation, and must be treated as such for an engineering analysis. Thus, to get to the value contribution of Preside itself, we must begin to look at the system at a lower level, breaking down the components of Preside and evaluating their individual contributive value. To demonstrate the potential for such an approach to an engineering analysis of the Preside system, we will look at a high level view of Preside components as shown in Figure 3.

From the view in Figure 3, we can see that the end-user view of the system does not provide direct access to the high-value raw switch data, but in fact provides an organized database of that raw data, and in turn, a front end to the organized data. While, as we have observed above, the raw switch data provides a high value contribution to the overall system, that data is only truly useful to operators trained in the observation of such raw data. While this data is in plain ASCII format, it remains unorganized, un-prioritized, and too massive to be utilized to any effective result. The value that Preside adds to the overall system is Preside’s ability to parse, organize, prioritize, and to present the raw switch data in a usable format. As we can see from the high-level component view of Preside that is presented in Figure 3, this value is not added at once, in singular black-box fashion, but is added in stages which are intrinsic with the components that make up the Preside system. For this reason, we can view the components of the system as providing particular value according to the relative accomplishment of the overall system goal of presenting data to the end user.

It is noteworthy that in our system evaluations according to both Figure 2 and Figure 3, the primary aspect of evaluating contributed value is in relation to system data. This is due to the fact that, although telephony traffic is inherently the most valuable aspect of a telephony system, the data with which Preside works is the data which enables the customer to evaluate and manipulate telephony traffic toward its profit potential. With this in mind, it is easily seen that the more organized and accessible this data, the higher its potential value. Thus, looking again at Figure 3, we can begin to assess component values according to data organization.

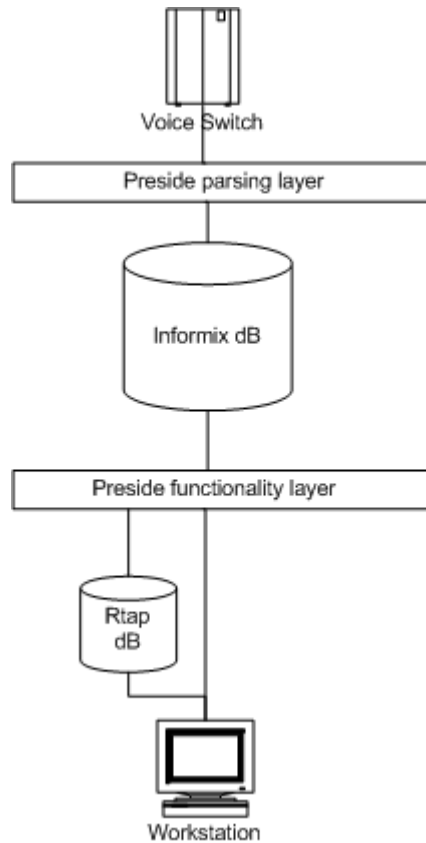


Figure 3. Functional components view of Preside

To begin this assessment, we will observe Figure 4, which further details the components of Figure 3. In this illustration, the larger arrows represent data pipelines whereby specific configurations within the Preside software manipulate data into organized formats which are contained in the system components of the Informix database, and the workstation interface. These data tunnels are labeled 1 and 2 respectively, and given a complete and working installation of Preside, represent the overall data flow within the Preside system. However, in order to break the system down into its value-oriented components, we must view this data flow in terms of the value contribution it brings in its various stages. These stages are represented in Figure 4 by the zones labeled A, B, and C. It can easily be seen that, moving left to right through the data flow zones, we move through iterations of added value, since:

- A. At this stage, data is in its most raw format. The data itself at this stage is of high value, but the return on that value is low due to the raw nature and relative un-usability of that data.

- B. At this stage, the Preside parsers have organized the raw data into the Informix database. While relatively user-unfriendly in format, the data is now highly accessible through database queries.
- C. In the final data flow stage, Preside as an overall working system shows its highest value potential as the data is presented in highly organized, prioritized and accessible format in the forms of graphic interface, lists, and reports.

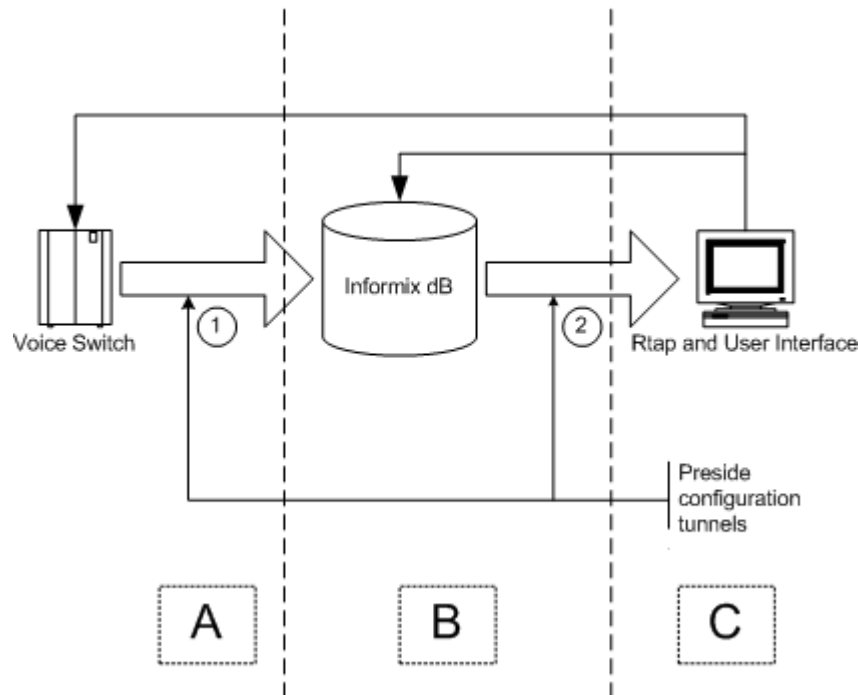


Figure 4. Preside's primary components and corresponding data flow

In order to understand how this relative value assessment of data flow stages might translate into phased deployment, we again observe Figure 4, now taking note of the data pipelines represented by the smaller arrows of alternate data routes represented along the top of the diagram. These tunnels represent the prospective notion that, although the Preside system might not be completely functional, various components within the system might still be accessible to the user, albeit at varying degrees of value. A phased system rollout could take advantage of this modular nature of the overall system's value by providing, in stages, a working system with each stage representing a higher level of system value. From the perspective that we have been viewing the system in the above evaluation, these stages correspond to the system components of (1) Switching network, (2) Informix database, and (3) User front end, and in turn, to the respective data formats of (1) Raw switch data, (2) Database data, and (3) Graphic and report organized data.

The realization of such a phased rollout would thus require (1) a prioritization analysis of the user data at both the raw data level and the database level, and (2) the investment of developing simple scripts which could make use of that prioritized data. This relatively small investment

would provide the return of phased deployment along the lines of increasing system value according to data usability. Table 1 demonstrates the data levels to be analyzed, the corresponding development required, and the potential value added in terms of data usability.

Data	Development investment	Return on Investment
Raw switch data	Parser scripts powered by GPL or a parser capable language like Perl	Access to highest priority raw data in crude parsed format
Parsed data in Informix dB	Query scripts powered by fast prototype language such as Perl or Python combined with SQL	Data queries of the same information that would eventually be presented in the final Preside interface.

Table 1. Phased deployment analysis according to data-oriented value added

The benefits of the phased deployment according to the above study of value-based analysis include:

1. System data becomes accessible and usable to the customer at an early stage of deployment, rather than only in the final stage of deployment.
2. Given the assumption that this accessible data will then be used, at least minimally, by the customer during the remainder of deployment, user feedback and final shaping of system requirements can begin early in the deployment process rather than near the end.
3. Perceived delays due to lagging deployment schedules will be minimized due to data accessibility early in the deployment process.

Furthermore, the tools which would be developed to serve this phased deployment would be valuable additions to the overall system in their own right, providing additional workarounds and means for debugging during times of system failure.

For a complete assessment of the value-oriented phased approach to development that we have examined above, further analysis of the system would be required at a lower level of system modularization. Such an analysis lies outside the scope of this paper, but it is hoped that this high level view of a value-based analysis provides a solid conceptual starting point for modularizing the system and phasing deployment along value-based lines. In fact, it is anticipated that a lower level assessment of Preside components would render much of the investment presented in Table 1 above as unnecessary. Given the author's understanding of the complexity of the Preside system at lower levels than presented above, it is quite possible that the tools for such a deployment already exist as pieces within the Preside puzzle, and only need to be deployed accordingly with perhaps slight modification to enable the independence of their deployment from the overall system.

Requirements triage for phased deployment

Further analysis of the system into phased deployment of the system at a lower level along the lines of the above high-level assessment according to component value could benefit from what Alan M. Davis calls “the art of requirements triage.” In his article, Davis defines triage as “the process of determining which requirements a product should satisfy given the time and resources available.” (Davis, 2003, p. 42). This process, according to Davis, requires grouping requirements into three categories: (1) Those requirements that must be satisfied, (2) requirements that need not be satisfied, and (3) requirements that “the product could incorporate, but that the development team must first carefully weigh against available resources.” (Davis, 2003, p. 42). For Davis, this process requires the following activities:

- Establish relative priorities for requirements. This may include establishing priority-related interdependencies as well.
- Estimate the resources needed to satisfy each requirement. This may include establishing resource-related interdependencies as well.
- Select a subset of requirements that optimizes the probability of the product’s success in its intended market, whether commercial or internal, relative to the resource constraints.

In application to the deployment of Preside, or the deployment of any complex system for which installation requires the re-engineering of existing components, the triage process will occur both prior to deployment in the planning stages, and during deployment itself. The gathering of system requirements prior to installation can help to form the shape of installation phases to come, and the gathering of customer feedback throughout the deployment of the system would further serve to refine those requirements. This on-site refinement of requirements and triage organized phased development would be made possible by the successive improvements of data usage with each phased rollout as described in the above section. Applying Davis’ primary triage activities to Preside deployment, we might focus, in turn, on the following activities as a method of refining the definitions of our deployment stages:

- Establish relative priorities of system components and component functionality along the lines of added system value. Also determine related component interdependencies.
- Estimate the time required for applied resources (usually a single on-site installation engineer) to implement prioritized components and related dependencies.
- Group prioritized components into phases and schedule the project accordingly.

To see how this approach might benefit a lower level assessment of phased deployment, we will consider, as an example, a single component within the Preside software suite. The component to be examined is the traffic report view which provides a GUI interface of traffic reports based

on stored traffic data and user definable criteria such as a date range or switch name. The data flow of this single system component can be modeled much like the overall system, and is demonstrated for our purposes of analysis in Figure 5.

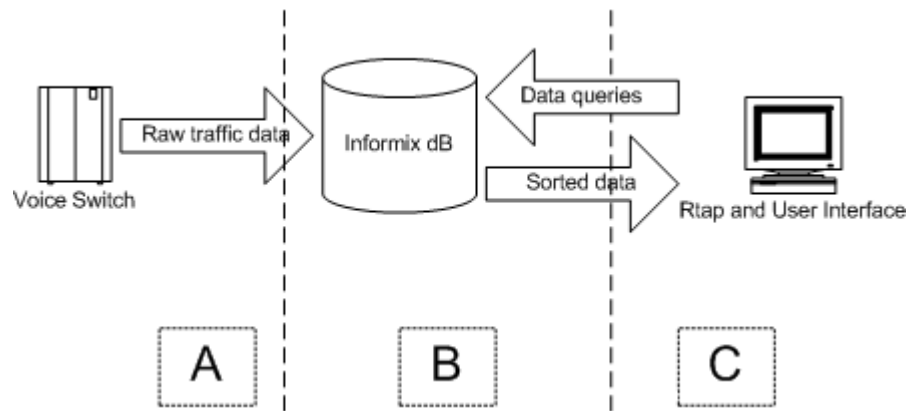


Figure 5. Component data flow for traffic statistics within Preside for Telephony Networks

For the analysis of requirements toward developing a phased rollout of this system component, we apply our first activity stated above which is to “establish relative priorities of system components and component functionality.” This example assumes that some aspect of traffic reporting will be valuable through all phases of deployment (relative to other system components) so we will proceed by establishment of priorities of functionalities of the traffic report component. By order of relative importance, component functionalities are listed below:

1. Provide access to traffic statistics provided by the switch
2. Provide access to parsed and organized traffic data through user definable queries
3. Present traffic data in an interactive GUI format with user definable parameters

By applying these requirements, prioritized by the process of triage, to a phased deployment model, we can see demarcation of deployment phases along similar lines to those discovered in the above high-level analysis of system deployment according to value-based considerations. Phased deployment for our present example, in terms of data access, could be procured through the following resources, requiring limited development investments similar to those shown in Table 1:

1. Terminal access to raw traffic reports which might be stored in text files and made useful through basic text based searches using simple tools created from basic Unix technologies such as Sed or Grep, or simple Perl based search scripts.
2. Script based access to parsed database which incorporates SQL for user definable data queries.
3. Full GUI formatting of traffic data through the completed installation of Preside.

Figure 6 demonstrates how, through the course of such a phased deployment, the user's point of view (POV) relative to traffic data will change as value is progressively added to the system.

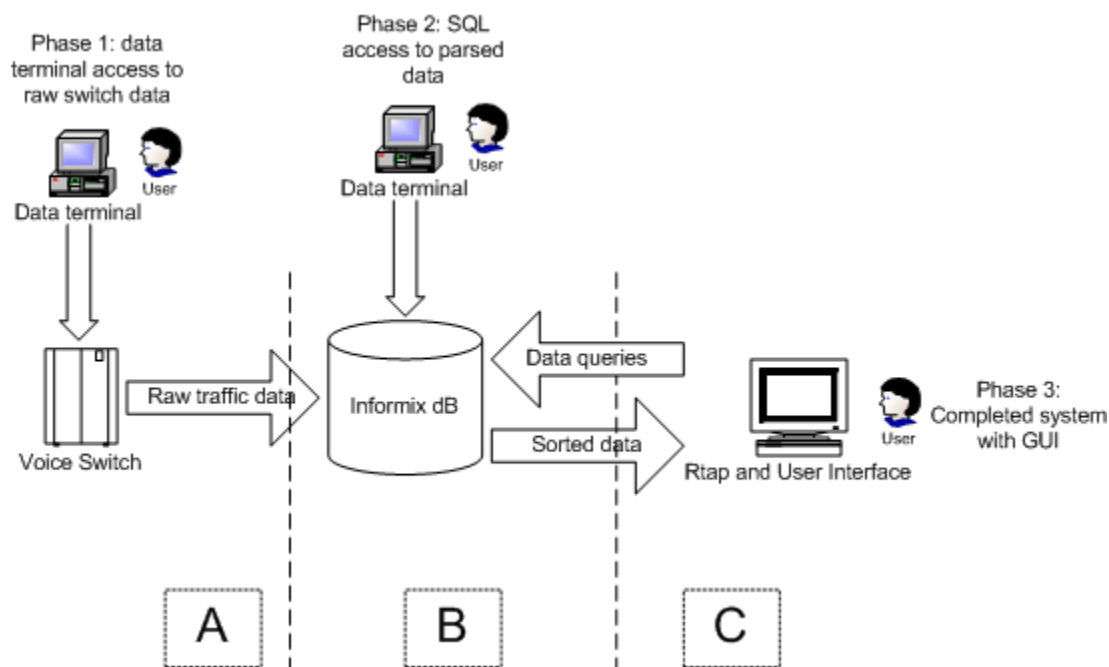


Figure 6. Progression of user POV and value development of traffic report data

The phased approach provides a means for increased customer involvement with the system throughout the deployment process, resulting in a higher level of user feedback and mutual learning for both Nortel and the customer. Furthermore, given the determination that, relative to this example, traffic statistics are determined to be a valuable asset to the customer's bottom line, then the phased approach increases the customer's bottom line by making that data available at an earlier date.

The purpose of this example using the single component of traffic statistic reporting has been to demonstrate the applicability and benefits of system evaluation for phased deployment at a lower level than that demonstrated above. A systematic evaluation of the Preside system would reveal the many components that make up Preside which then could be considered for independent and for phased rollout. Apperly (2003) gives us some criteria for further evaluating system components. According to Apperly, these are the characteristics of a component:

- Components are units of deployment
- Each component has a published interface
- Communications standards are important for components
- Components are assembled to realize solutions (Apperly 2003, p. 121)

Software engineering tools, such as value-based analysis, and requirements triage, among the many other tools available to the system engineer, give us a means for analyzing the system for the purpose of engineering improvements. As deployment is an aspect of system engineering, and as engineering tactics take on a particularly significant role in the deployment of a complex system such as Nortel's Preside, so engineering tools and practices can serve to better the deployment process as we have observed in these sections on phased development along the lines of value-based analysis and requirements triage.

Conclusion

Deployment of Preside could benefit from a number of principles taught to us by Software Engineering and Software Project Management. Within the scope of this paper, such principles include the adaptation of the following:

1. Modularized installation and testing
2. Phased rollout with proposed phases:
 - a. Informix install and data fill with automated testing by SQL enabled script
 - b. RTAP configuration: automated script back end with user observation of graphic front end.

This study is only a brief look into the potential for applying software engineering methods to the deployment of Nortel's Preside for Telephony Networks. The principles assessed in this study might also be applied to other complex systems which require the re-engineering of each deployment from existing software modules. For further examination of adding value to Preside deployment through software engineering methods, other methods and tools should be considered besides those examined above. Two methods that seem immediately apparent are the ideas of object orientation and UML (Unified Modeling Language) modeling. The application of these popular methods might serve two value adding purposes:

1. Further define the modularity of Preside, thus improving capabilities for the phased methods described above.
2. Enhance Preside related communications on all business levels, including executive, administrative, management, and technical levels.

This second aspect is important for overcoming those communications barriers which have been cited above as obstacles in the Preside deployment process. The UML is particularly suited for improving business and technical communications, and is also suited for the modeling of modular systems. Preside is a complex set of individual programs working together to solve a common set of problems. UML style modeling would serve to reduce this complexity and promote multiple levels of enhanced communication. In his book *UML in a nutshell*, Sinan Si Alhir states, "*Programs* are collections of problem-solving efforts (or projects). They are coordinated and related so that each project may benefit from the work of other projects in the program. One of the critical criteria necessary for achieving this objective is communication. If communication is possible across projects, it may be more likely that work and knowledge will

be readily shared and work effort reduced. The role of the UML is to enable and facilitate such communication.” (1998, p. 16).

Alhir’s definition of a program (1998) clearly applies to Nortel’s Preside which stands to benefit from the communication efforts inherent in such techniques as UML modeling. Similarly, while the above efforts demonstrated through such techniques as value-based analysis, requirements triage, and web-based procedures, have focused on the elements of value, requirements, and efficiency respectively, underlying each of these approaches is an inherent notion of communications through engineering efforts. In the end, every system requires the use of a company’s most valuable resource: people. People are required to design the system, to test the system, to deploy the system, and to operate and administer the system. For this reason, Preside for Telephony Networks, like any complex software system, can benefit from the lessons of systems and software engineering which serve to enhance communication and understanding among the developers and customers alike.

References

- Alhir, S. S. (1998). *UML in a nutshell*. Sebastopol, CA: O'Reilly.
- Apperly, H., Hofman, R., Latchem, S., Maybank, B., McGibbon, B., Piper, D. & Simons, C. (2003) *Service-and-component-based development*. Boston: Addison-Wesley.
- Bass, L., Clements, P. & Kazman, R. (2003). *Software architecture in practice*. Boston: Addison-Wesley.
- Boehm, B. & Huang, L. G. (2003, March). "Value-based software engineering: A case study." *Computer*, 36(3), 33-41.
- Brooks, F. P. (1995). *The mythical man-month*. Reading, MA: Addison-Wesley.
- Davis, A. M. (2003, March). "The art of requirements triage." *Computer*, 36(3), 42-49.
- Gorton, I., Liu, A. & Brebner, P. (2003). "Rigorous evaluation of COTS middleware technology." *Computer*, 36(3), 50-55.
- Kruchten, P. (2000). *The rational unified process: an introduction (2nd edition)*. Boston: Addison-Wesley.
- McConnell, S. (1998). *Software project survival guide*. Redmond, WA: Microsoft.
- Norris, M., Davis, R. & Pengelly, A. (2000). *Component-based network system engineering*. Boston: Artech House.