Database Support for PC Software Configuration and Version Management in the START Network

S. Auerbach *, A. Buchmann†, I. Buchmann *, T. Kudraß †, K. Lange *, C. Lier *, K. Weidmann *

* START Informatik GmbH
Mainzer Landstr. 341 , D-60326 Frankfurt

† Technical University Darmstadt, Dept. of Computer Science,
Database Systems Division, Frankfurter Str. 69a, D-64293 Darmstadt

Abstract. Configuration management problems are discussed in the context of one of Europe's largest wide area networks, the START travel services network. Problems are caused mainly by the size of the network, its heterogeneity, its growth rate and the availability requirements of day-to-day operations which are stretching the current file-based configuration management system that is used to control the remote update of software on the point of sale PCs. A database solution for support of a new configuration and version management process is presented and two pilot implementations, one relational and one object-oriented, are discussed.

1 Introduction and Environment

Travel services, ranging from airplane, rail and ferry tickets to tour packages, car rentals and tickets to special events are offered by a variety of service providers. To make the services of railways, airlines and tour operators readily available to the public through the travel agencies and to allow for the appropriate payments and accounting, START operates a wide area network with more than 26.000 terminal nodes worldwide at the point of sale, facilitates the interoperability among service providers and supplies the necessary back-office functionality. The nodes at the point of sale are PCs equipped with a special purpose networking board and may run under either SCO UNIX, XENIX or OS/2. Because of equipment amortization laws and the logistics of replacing and installing such a large number of PCs, at any given point in time there may be a variety of hardware configurations in the field, ranging from low-end Intel 80286-based machines to powerful pentium-based LAN servers. To make the situation even more complex, travel agencies often require additional software that is installed directly by the user on the PC. The heterogeneity in hardware and user preferences is also reflected in the multiple versions of the operating systems that must be supported on the machines in the field. At the center of the WAN are two powerful Siemens H120-S mainframes running above 200 transactions per second with a third mainframe, an H130-E for backup. The mainframes provide the interoperability between the PCs in the travel agency and the more than 70 hosts of the service providers and carry out the necessary accounting.

In 1993 more than 5.3 million direct tour bookings were handled through the START network alone in Germany. It serves also as a gateway to the Amadeus reservation system routing 30.5 million bookings to Amadeus in 1993. The network, currently one of Telekom's 10 biggest customers, is being converted to packet switching technology. Figure 1 illustrates schematically the configuration of the START WAN.

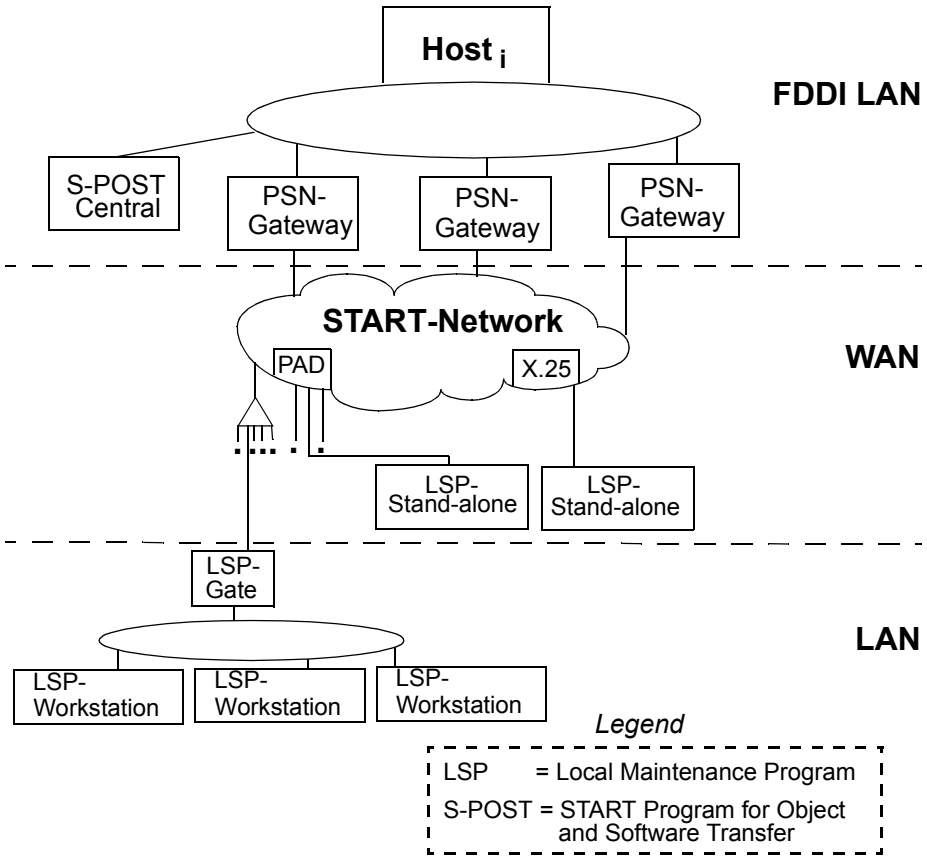

Fig. 1. Configuration of the START WAN

2 The configuration management problem

To maintain a network of this size and heterogeneity running with acceptable level of reliability is a difficult task. New software, updates and bug fixes must be installed in the point of sale machines with a minimum of disruption for the user and the day-to-day operations on the host. Further, users should not be burdened by the installation of software, a task that would be too taxing for the less sophisticated users and could thus lead to a loss of network reliability. Therefore, software is installed automatically mainly during off-hours under careful load-control by the hosts to avoid service degradation due to the transfer of large software packages over a low-bandwidth WAN.

Scheduling of software updates depends on the status of the machines in the field, information about new sales or lease agreements, new software releases and compatibility information among the new software, the specific PC's capabilities and possibly with third party software known to be installed on a particular PC. For example, it may not be possible to upgrade the version of some customized software in a given operating system environment, because a given software module it uses frequently is not yet available for the new OS version.

In addition, many special situations must be handled. These include the limited release and distribution of new software to pilot customers, special handling for teaching and demo equipment, and software customization for some of the larger clients.

The huge task of maintaining, controlling and providing the configuration management of such a network is compounded by the possibility of component or network failure and human error, which may range from disconnecting a PC overnight to inadvertently damaging a file necessary for communication between the PC and the host, or not updating the configuration information after a repair.

3 The current system

At present, software maintenance is carried out through the ASP system ("Automatische Software Pflege"). This system consists of a host-based portion that initiates and controls the software maintenance process and a local counterpart (LSP) on the PC that establishes the communication and carries out the installation locally. The information necessary to control the software update process is kept in the form of files that are periodically updated and compared and from which extracts are used to drive the software actualization process. Because of the file-based nature of the current system and a cumbersome input and output process, combined with a lack of consistency enforcement capabilities, the quality of the configuration management data kept on the central ASP suffers both in consistency and timeliness. As the WAN expands at a rather brisk pace and more service providers offer their travel services through START, the existing system will slowly reach its limits. Therefore, a new system is being developed in which the critical data handling problems for controlling the configuration management process are addressed. The remainder of this paper describes the new system, its design and pilot implementations that were realized in conjunction with the Technical University of Darmstadt.

4 S-POST - The New System

To address the handling and consistency problems mentioned in section 3, a new system called S-POST (START Program for Object and Software Transfer) was designed [1]. Since most of the data handling problems arise on the host side, a new central component was conceived based on a relational DBMS. Figure 2 shows an outline of the schema. The information handled can be roughly grouped into terminal and network information; information describing the properties of software packages; maintenance, version and transient status information; and special requirements information.

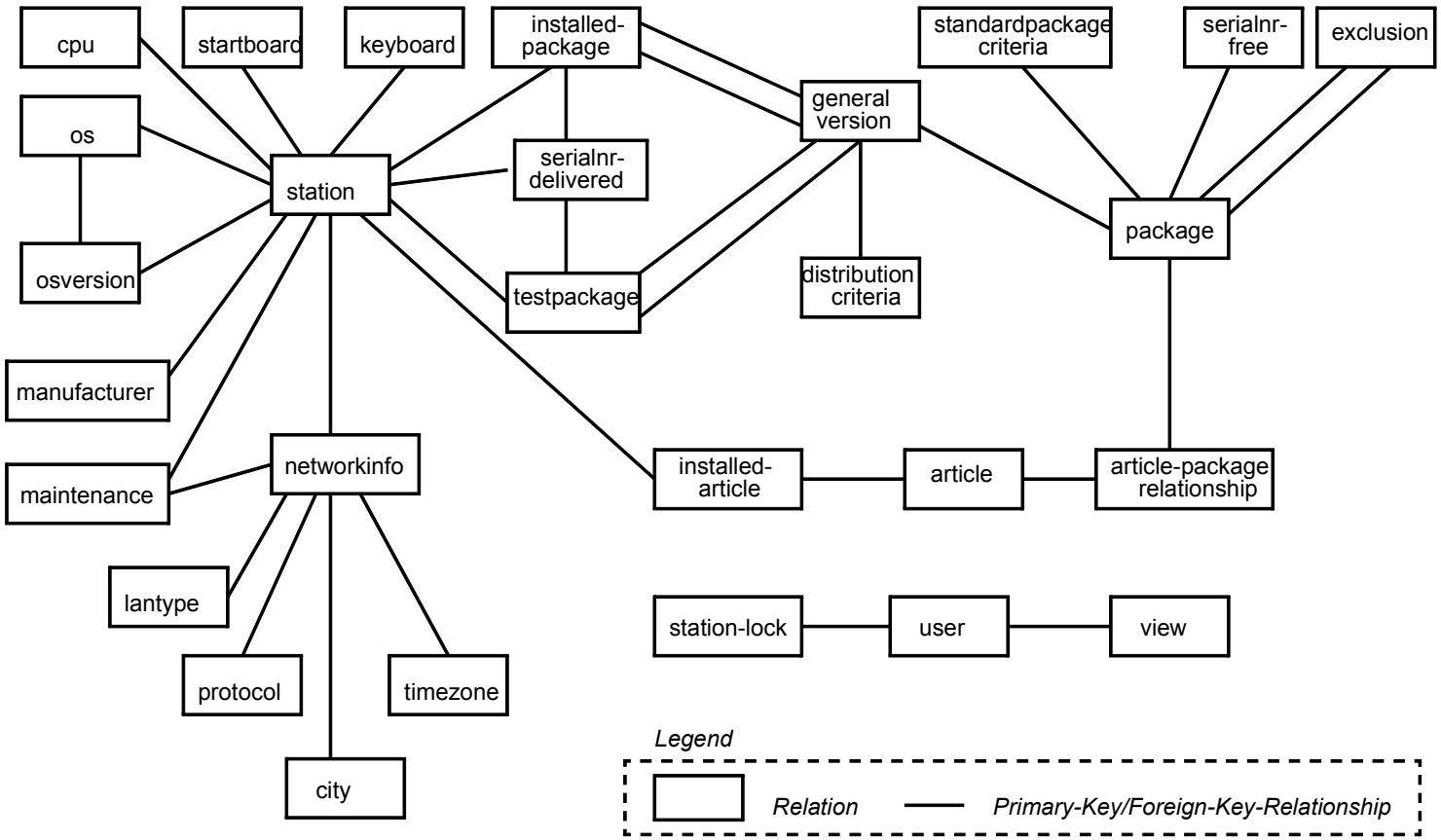

Fig. 2. The Schema of the S-POST Database

Terminal equipment and network information (also considered as core data) comprises all the hardware data, such as type of CPU, whether a system is standalone or part of a LAN, whether it is a gateway, the type of LAN, communication protocols, location, serial numbers, operating systems and its version, etc. Core data further comprise all the basic software that is installed once and not updated or changed remotely.

Software Information describes the configuration of a software product, its versions and its packaging into files for distribution. Each software product may be configured differently according to hardware capabilities, operating system and kind of service required by a customer. In many cases, other software products may depend on a particular feature of another product. If this feature is not available in the basic configuration but only in a larger configuration, the system must ensure that the minimally required functionality is always installed. Similarly, exclusion dependencies may exist. These dependencies among software products are important and must be tracked. They may also be transitive and the formation of any cycles must be detected. It is further necessary to document carefully in the database how a given software product is broken up into packages for distribution and the order in which these packages must be distributed and installed.

Maintenance, version and transient status information comprises all the operational information that is needed for day-to-day operations. It includes all the information about the software maintenance process proper, the recent version history at each node and such transient status information as reachability of a node or the necessary locks that must be placed on a system while the software maintenance process is under way. The whole set of core and maintenance data is referred to as base data.

The special requirement information is important to provide much of the flexibility needed to handle special cases. This information is defined by the user (in this case user refers to someone responsible for day-to-day operations of the START network) according to special criteria. For example, if a new software product is released it is not installed on all PCs across the board, but in a controlled form. Therefore, distribution criteria that may comprise certain cities or selected clients acting as test-sites must be defined. These testing criteria may also have a certain validity interval, after which they should be removed. In the S-POST system, different distribution or locking criteria may be specified. The distribution and locking criteria are stored in the database in the form of SQL-like selection predicates that are parsed and executed.

Finally, one of the major requirements of the new system is to assure data consistency in a far higher degree than is possible under the current file-based solution. This can be achieved by exploiting the consistency enforcement mechanisms of the DBMS, such as key uniqueness, foreign-key and domain specifications. More complex consistency criteria can be realized through triggers offered by today's DBMS.

The database of the S-POST system provides commonly used data for all components of the new system. The automatic maintenance module is responsible for periodically checking all PCs, and when new software needs to be installed, it sends the corresponding information to the PC and starts the local maintenance program which brings the

PC up-to-date. To be able to track the update process and do trouble shooting, a separate supervision module is implemented. This module allows the supervisor to look at all the data und provides the tools to generate statistics and reports.

A number of daily check jobs update the base data and look for error situations and inconsistencies.

Introduction of new versions, exclusion dependencies and distribution criteria is done by a database administration program that allows to manipulate the data interactively.

5 The Pilot Implementations

Two pilot implementations of sizeable portions of the S-POST system were realized by teams of students at the Technical University of Darmstadt. Within these pilot projects the schema was designed using a variant of the Coad Yourdon methodology [2], leading to a refinement of the original system specification [1]. The object-oriented design was mapped both into a relational and into an object-oriented schema. During this mapping the existing operational experience and information about access patterns were heavily considered to guarantee a good performance. The relational schema was implemented in INFORMIX-Online Version 5.0 [3] while the ObjectStore Release 3.0 [4] was used for the object-oriented implementation.

In the relational realization the configuration and distribution information is defined by the network administrators in a manner that is conceptually close to a view definition. This approach allows the use of the same basic maintenance procedures to all systems rather than having different processes for normal distribution, pilot system distribution, training software etc. This unified handling results in an easy definition of new patterns or exception handling capabilities without one special-purpose module undoing what was done by another. However, it means storing the view definitions (which are typically metadata) as data in the configuration management database and processing these data in the same applictions as the base data. To do this efficiently, a parser was developed that translates the stored view definitions into dynamic SQL statements and translates the external names into the corresponding internal names to avoid or speed-up the joins.

Consistency may be violated either by modifying the base data or the configuration and distribution criteria defined over them. Consistency among base data is enforced immediately after updates on them using the trigger mechanisms provided by Informix. Consistency between the base data and the selection criteria defined over them cannot be enforced with Informix's trigger mechanism since the trigger mechanism does not allow dynamic SQL statements in the action part. Therefore, a hybrid solution had to be chosen in which the insert, delete and update operations were embedded in C functions that implement the consistency checks.

Object-Oriented DBMS are often touted as being ideally suited for supporting software engineering applications in general and configuration management systems in particular. Therefore, it was interesting to compare the facilities offered by the rela-

tional system and those offered by a typical C++-based OODBMS. The OODBMS proved much more powerful for expressing typical aggregations, such as all the software that needs to be installed on a certain node or a LAN consisting of a gateway, non-gateway nodes, printers, etc. The OODBMS also supports better the ordering relationships and dependencies. On the negative side, the lack of a powerful query language with which to define views dynamically made it more difficult to express in a flexible and user-friendly way the distribution criteria. On the interface side, currently available OODBMSs require much more ad-hoc implementation than the relational systems.

6 Outlook

Several trends can be recognized in the development of software in general and the software for the tourism industry in particular that will effect the version control strategies.

- The increase in computing power of the PCs at the point of sale is forcing a shift of the functionality currently handled by big mainframes to the PCs. A typical example are consistency checks during the input process instead of shipping inconsistent data and verifying them on the mainframe. This means that certain sets of data, particularly those that are relatively static, such as the list of airports and railway stations, must be shipped with the software to the terminal nodes.
- The growing popularity of graphical interfaces and the increased functionality of the software running on the PCs has lead to much larger files needing to be transferred.
- The growing number of service providers and the diversification of services offered makes it difficult to guarantee immediate and universal distribution of every service and the corresponding software and data to every PC. Therefore, the possibility of on-demand transfer of certain software and/or the associated data may be necessary.
- The emergence of lazy vs. eager update strategies in database systems and the triggering mechanism to enforce them may lead to new strategies for version control.

These trends must be balanced against the capabilities and costs of the underlying communication network. The new S-POST system considers many of the above trends and provides the means for a flexible version and configuration management. Special emphasis was placed in the easy definition of distribution criteria and the description of the packaging process of the software into distribution units.

Acknowledgements

The authors wish to acknowledge the enthusiasm and many hours of hard work of the students who implemented the pilot solutions.

References

[1] Auerbach, S., Buchmann, I., Merhazion, E., Stöck, S.: Das START-POST Projekt. Systemspezifikation Version 1.0., START Informatik GmbH, 1994.

[2] Coad, P., Yourdon, E.: Object Oriented Analysis; Prentice Hall, 1991.

[3] Informix Software, Inc.: The Informix Guide to SQL: Reference, Sept. 1992.

[4] Object Design, Inc.: ObjectStore Release 3.0 for UNIX Systems, Dec. 1993.