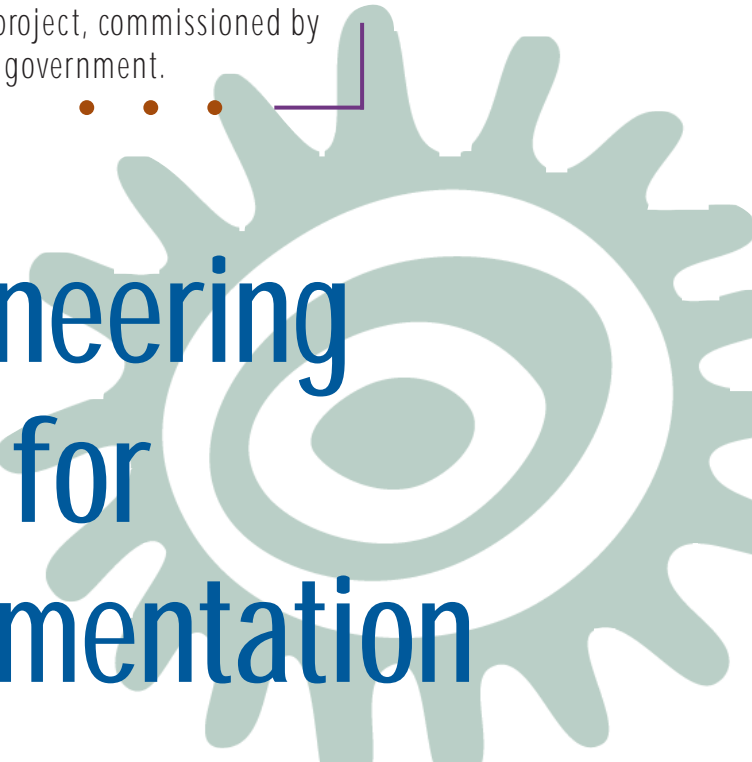


Case Study

Reverse-engineering a commercial client-server system from PeopleSoft yielded a valuable resource and proved to be cost-effective. The authors describe the motivations for, approach to, and results of this project, commissioned by the Commonwealth of Virginia's government.

Reverse-Engineering New Systems for Smooth Implementation



Peter Aiken and Ojelanki K. Ngwenyama, Virginia Commonwealth University
Lewis Broome, Innovative Business Solutions

The Commonwealth of Virginia's departments of Personnel and Training (DP&T) and Accounts (DOA) undertook a \$12 million effort in 1994 to replace their existing payroll and personnel information systems because the technology was dated, it no longer provided timely information support, and the databases were not integrated. Its systems had become inefficient and cost-prohibitive to operate and maintain, and management was concerned about keeping qualified technical expertise on staff to support them. Successful implementation of a new integrated human resource information system, or IHRIS, required top management to be sensitive to public-sector funding pressure and the political fallout of government program failures.^{1,2} (More information is available at <http://fast.to/peteraiken/>.) As a result, the transition team had to show rapid and credible progress implementing the new system.

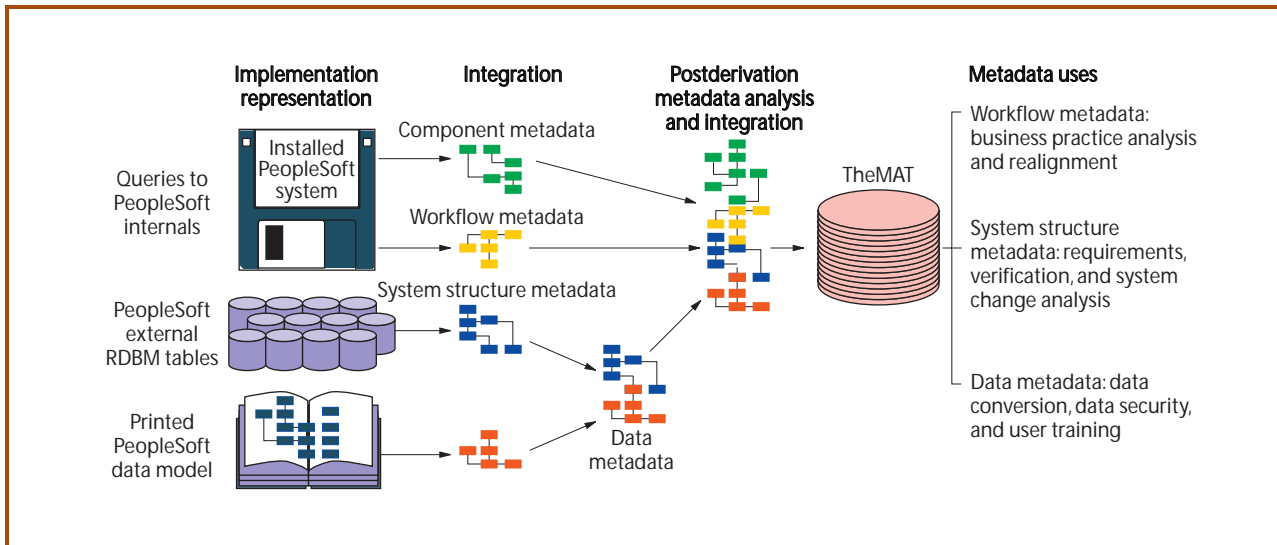


Figure 1. Reverse engineering, sources used to derive metadata, and its uses.

By 1997, management selected the PeopleSoft Human Resources, Benefits, and Pay modules as the basis for the new IHRIS system design. The client software was designed to run on PC desktops accessing a superserver providing relational data management and distributed workgroup servers hosting associated utilities. The PeopleSoft system was developed specifically for distributed client-server implementation, but the company expected the modules to be tailored to fit the needs of each organization.

To leverage the investment in its existing system, the DP&T/DOA decided to reverse-engineer the PeopleSoft modules. The initial motivation was to develop a detailed understanding of the modules in order to convert the two legacy systems' data. The goal of reverse engineering is to analyze a system to uncover facts about its design and functionality, often called metadata, to facilitate development and implementation activities. Metadata, or "data about data," is defined by the ISO description 11179 as "the information and documentation which makes data sets understandable and sharable for users."³ It also describes facts about the system components.⁴

Organizations concerned with effective and efficient operations can benefit tremendously from easily accessible system metadata. From a practitioner perspective, new systems are likely to be more easily reverse-engineered as system developers adopt architecture-based approaches to development. This architecture provided the IHRIS transition team with easily accessible and obtainable system metadata with a modest investment.

GETTING STARTED

An innovative industry-academic partnership

between the DP&T/DOA and the Virginia Commonwealth University's Information Systems Research Institute was formed to establish part of the IHRIS transition team. In the Spring of 1997, a reverse engineering team was organized to reverse-engineer the PeopleSoft modules and obtain the metadata. The RET consisted of four graduate students, each of whom had studied advanced analysis and design and database management. By August 1997, the team had provided more than 1,200 professional and 6,000 student project hours in support of IHRIS implementation. The RET logged 660 hours of total project time between April and September 1997 for an approximate total project cost of \$21,000 (600 student hours at \$20 per hour and 60 project leader hours at \$150 per hour).

Motivated by the need to better understand the system, the RET applied reverse engineering as a systematic approach to examine, document, model, and analyze the system. The team followed an iterative, evolutionary approach to derive the metadata; this was necessary because the process of analyzing and understanding the system was itself iterative. The RET would often begin with an understanding of part of the system, only to have that understanding evolve as new information emerged. The reverse-engineering analyses consisted of three steps applied repeatedly.

- ◆ Identify and describe a metadata requirement.
- ◆ Identify, derive, and integrate the metadata.
- ◆ Use the metadata to support an implementation task.

Figure 1 indicates graphically how these analyses were carried out and how the metadata was derived, analyzed, and integrated. Three metadata sources were the system itself, the system documentation, and other nonsystem sources.

Table 1
Comparison of System Characteristics

| System | Platform | OS | Age in 1998 | Structure | Approximate Number of Data | | | |
|-----------------|----------|--------|-------------|------------------------------|----------------------------|-----------------|----------|------------|
| | | | | | Physical Records | Logical Records | Entities | Attributes |
| Payroll (LS1) | Amdahl | MVS | 15 | VSAM/virtual database tables | 780,000 | 60,000 | 4/350 | 683 |
| Personnel (LS2) | Unisys | OS | 21 | DMS (network database) | 4,950,000 | 250,000 | 57 | 1,478 |
| PS-L | PC | Win 95 | New | Client-server RDBMS | 600,000 | 250,000 | 1,600 | 15,000 |
| PS-P | PC | Win 95 | New | Client-server RDBMS | 600,000 | 250,000 | 2,706 | 7,073 |

SYSTEM CHARACTERISTICS

The two systems to be replaced were used by DP&T/DOA to merge payroll and personnel records. The payroll system (LS1) was a partially implemented commercial system with in-house extensions. It processed the payroll for more than 130,000 state employees and was designed around four large VSAM files. At runtime, data was redefined by the system to appear as more than 350 unique, virtual database tables. The personnel system (LS2) was developed completely in-house using a platform-specific database, associated Mapper, and other software programs. The LS2 database maintained personnel benefits records for roughly 100,000 full-time employees and more than 50,000 part-time employees (130,000 of which had corresponding LS1 records), and about 100,000 retirees.

The data architecture of the PeopleSoft modules employs a relational data model and SQL to ensure compatibility with a number of commercial relational database management systems. The user interface employs rectangular data arrays and manipulation tools. Once users are taught how to use these tools, they can search for their data, within security constraints. The system also provides a user-friendly report generator that supports a wide range of data access, analysis, and presentation facilities.

Table 1 presents a comparison of some basic logical and physical characteristics of LS1, LS2, and the logical and physical models of the PeopleSoft application (PS-L and PS-P respectively).

EXTRACTING, MANAGING, AND INTEGRATING THE METADATA

Most of the reverse engineering was accom-

plished using the PeopleSoft system and a toolkit based on the Microsoft Office Suite. To manage the metadata in the most flexible and adaptable electronic format, the RET developed and implemented a metadata repository database called The Metadata Access Tool, or TheMAT, using MS Access and automated and manual procedures. Various utilities were combined using the integrated project management, word processing, spreadsheet, database, and presentation software capabilities of MS Office. TheMAT maintains the metadata for each system and permits data item mapping between systems.

Initially, the RET developed a hierarchical decomposition model of the PeopleSoft system based on its component structure, workflow, and data views. It then used this information to formulate specific queries about system objects, getting PeopleSoft to report metadata about its own structures. Until the RET obtained the desired results, it repeated the cycle of decomposing specific system objects and deriving enough information to structure specific queries to report the metadata. The team then statistically analyzed the metadata to determine its validity and relationships to other metadata. This analysis cycle is described in the boxed text entitled "Reverse Engineering" on p. 40.

Prior to V6, the PeopleSoft modules were delivered with limited CASE tool support and static technical documentation in PDF format. These were inadequate to support the IHRIS implementation. Additional CASE tool support, costing \$1,200 per seat, was purchased from another vendor, but this provided only marginally better functionality. To work with these models the RET had to print them, then manually cut and paste more than 200 pages together. More importantly, the paper and PDF formats prohibited browsing or computer-based analysis of the model content or structure. As the project

planning proceeded, requirements for more robust metadata emerged.

METADATA REQUIREMENTS

The RET's first priority was to analyze the transition team needs to understand the new system. The analysis focused on discovering the system core, as did similar investigations.^{5,6} Requirements analysis revealed the need to understand and describe the system structure, workflow, and data metadata. Structure metadata describes user system interaction according to interaction roles. Workflow metadata describes the processes supported by the system. Data metadata describes the data structure and relationships.

The IHRIS transition team wanted electronic metadata access when working individually and in groups. They wanted the ability to manipulate and analyze the metadata and to derive useful information from it when performing implementation tasks. They wanted to understand and describe the system in terms of structure, workflow, and data. These requirements and the metadata volume required a central repository and interactive graphical analysis tools.

Figure 2 illustrates the PeopleSoft enterprise software metadata that describe the Human Resources, Benefits, and Pay modules. This model describes the relationships represented by various types of metadata, the entity definitions, and the number of facts derived by the reverse engineering. The number of facts also indicates relative system component complexity. For example, the *records* and *fields* entities represent 2,705 Oracle tables, 7,973 columns, and 5,873 instances of *fields* occurring on *records* in the physical database. In this figure, the metadata representing *panels*, *menuitems*, *records*, and *fields* connect the three types of metadata.

System structure metadata requirements

One transition team requirement was to understand the system functionality. Understanding structure metadata was key to determining the collection of *panels* supporting specific IHRIS business activities. As implemented by DPT, the PeopleSoft system structure consists of a hierarchical arrangement of menu structures leading to more than 1,400 *panels*. The *panels* are the interfaces where users create, read, update, or delete data. To reach a given *panel*, users must sometimes navigate through a

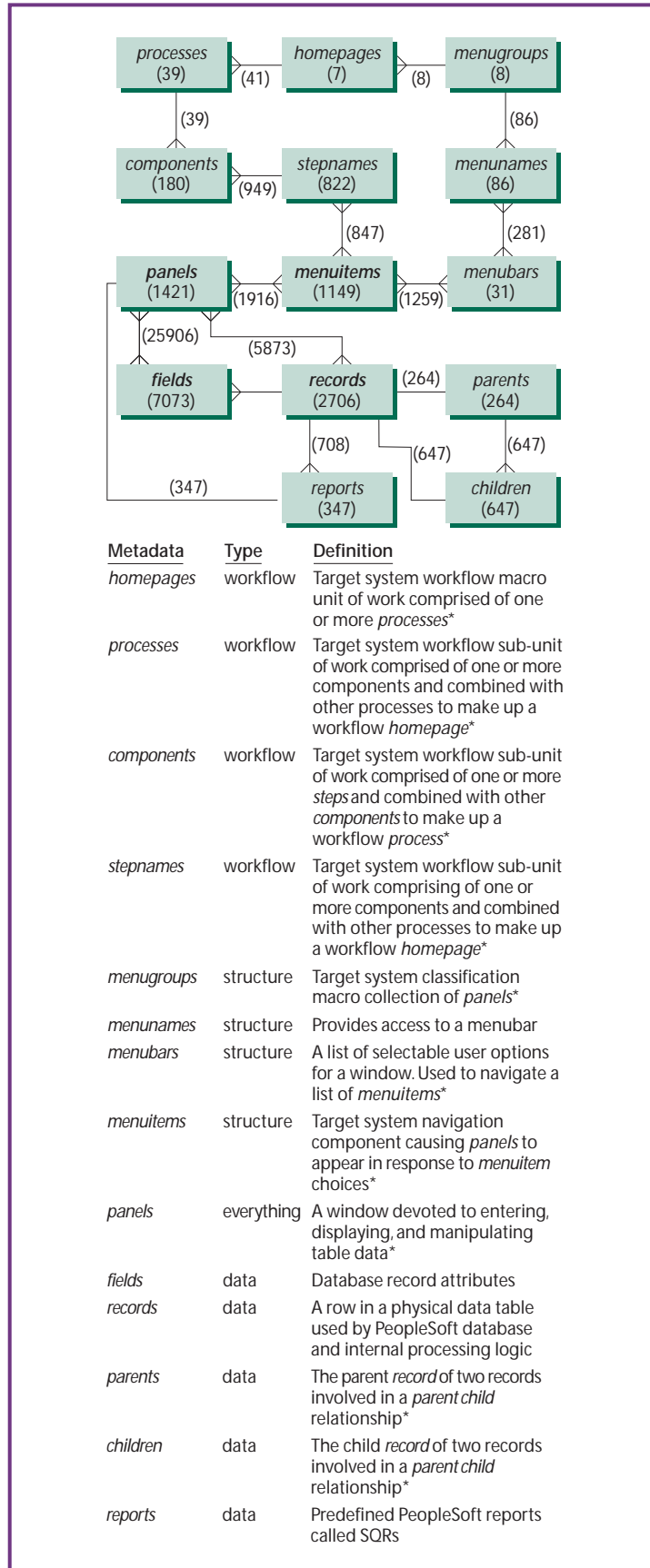


Figure 2. Logical metadata requirements, definitions, and volume. (Note: those definitions followed by (*) are taken directly from PeopleSoft documentation.)

REVERSE ENGINEERING

The Reverse Engineering Team developed a hierarchical decomposition model of the PeopleSoft system based on its component structure, workflow, and data views. This information was used to formulate specific queries about system objects, getting it to report metadata about its own structure. The metadata was then statistically analyzed to determine its validity and relationships to other metadata. The iterative reverse-engineering analysis cycle^{1,2} is summarized in Table A in six steps.

The last step usually resulted in more refined requests for additional metadata, and the cycle began again. This general model of extracting specific information from the system via SQL queries, restructuring it, and integrating it with existing TheMAT contents was repeated with many analysis variations to develop the metadata. The variations generally consisted of changing

the source of the analysis inputs to include different system objects, the system documentation, or the metadata itself. These analyses provided the RET with the facility to report on associations among any set of systems objects using SQL queries. In this way, RET members were able to define and document the PeopleSoft system metadata and relate these to the LS1 and LS2 metadata.

REFERENCES

1. P. Aiken, *Data Reverse Engineering*, McGraw Hill, New York, 1996.
2. P. Aiken, "The Reverse Engineering of Data," *IBM Systems J.*, Vol. 37, No. 2, 1998, pp. 246-269.

Table A

| Reverse-Engineering Analysis Steps | Illustration |
|--|--|
| Determine what type of metadata to derive for specific system implementation requirements. | Implementation needs indicate a requirement to understand the workflow metadata by obtaining a list of all combinations of <i>stepname</i> instances along with each associated <i>component</i> , <i>business process</i> , and <i>homepage</i> . |
| Formulate a query designed to extract specific metadata. | Using SQL, extract from the system all unique combinations of <i>homepages</i> , <i>processes</i> , <i>components</i> , and <i>stepnames</i> resulting in a 13,044-line report. |
| Export the extracted metadata to a spreadsheet for subsequent complexity analysis and validation. | Saving the query results into .xls format permits further statistical analysis of the metadata to determine information about metadata relationships, complexity, and occurrence frequency in order to confirm the extracted metadata correctness. |
| Import the validated metadata into TheMAT. | Once validated, the process metadata is moved into the MS Access database as a new stand-alone table. |
| Integrate the new metadata with the existing metadata. | The new table is formally associated with the existing metadata, in this case linking processes to <i>menugroups</i> via <i>homepages</i> and <i>stepnames</i> to <i>menubars</i> and <i>panels</i> via <i>menuitems</i> (as shown in Figure 2). |
| Provide the resulting, richer metadata to the requesting user, verifying the metadata and its utility. | Enhance TheMAT reporting capabilities, publish the next version, and work directly with the user group requesting the metadata to ensure that the metadata is accurate and meets their needs. |

four-tier hierarchical menu structure comprised of (from highest to lowest) *homepages*, *menugroups*, *menunames*, *menubars*, and *menuitems*. The resulting metadata supported the transition analysis by providing responses to the following questions:

- ◆ Given a *menuname*, what are the possible *menubar* choices?
- ◆ From what *menuitems* can this *panel* be accessed?

- ◆ Given a *menubar*, what are its associated *menuitems*?

The associations between higher and lower navigation components are usually one higher-level item related to many lower-level items. However, most implementations required support for many relationships. The RET identified and documented all the relationships among these components.

Workflow metadata requirements

A second implementation requirement focused on aligning the organizational business practices with the new modules' functionality. The workflow metadata provided the transition team with descriptions of the PeopleSoft business processes. The workflow metadata was used to compare the PeopleSoft process structure to the DP&T/DOA's. The metadata supported mappings linking the DP&T/DOA business events to PeopleSoft functionality. This provided a framework supporting the transition team's analysis and system-tailoring activities. This information was important to the transition team as they assessed how well the existing work procedures and practices were aligned with those supported by PeopleSoft. The team performed a gap analysis to determine how much an existing process had to be tailored to fit the new system; it also identified and analyzed gaps and developed solutions, modifying either system functionality or business practice or both. The metadata requirements focused on understanding the workflow structure implemented by IHRIS:

- ◆ Given a *business process name*, what are its *components*?
- ◆ Given a *component name*, what are the steps that form it?
- ◆ Given a *panel*, how many *process components* access it?

The metadata was structured and repeated a generally hierarchical arrangement among workflow, consisting of *homepages*, *processes*, *components*, *stepnames*, and *menuitems*. The *menuitems* link the workflow with the system structure metadata.

Data metadata requirements

Third, the IHRIS transition team needed to derive design information about the physical data architectures of the new system for comparison and mapping to the legacy system data. This metadata was needed to convert the LS1 and LS2 data structures to the PeopleSoft data structures. The LS1 and LS2 data structures were reverse-engineered, and the metadata derived from this process was analyzed and transformed into a standard format for mapping to the PeopleSoft database. It uncovered the need to maintain metadata on more than 2,700 *records* with a subset of more than 7,000 *fields*. This requirement alone represents more than 35,000 facts. Another aspect of the data structure was the 264 *records* that function as *parents* to more than 647 *children*. Because the data metadata also detailed which

panels contained which *fields*, the team could use this data structure to integrate the data, workflow, and structure metadata.

USING THE METADATA IN THE SYSTEM IMPLEMENTATION

The metadata derived by reverse engineering was widely used during the new system implementation. Several examples show how metadata provided valuable system and background information.

The system metadata in TheMAT was used to document discrepancies between capabilities and requirements in the PeopleSoft system. To do this analysis, the transition team demonstrated to users a series of panels that were organized from the system structure metadata. The users associated each system requirement by linking it to corresponding system components. Discrepancies were noted for subsequent investigation and resolution.

The transition team analyzed the discrepancies and evaluated proposed system changes, modifications, and enhancements. Various metadata types were used to assess the magnitude of proposed changes. For example, the IHRIS transition team was interested in the number of panels requiring modification if a given field length was doubled. This information was used to analyze the costs of changing the system versus changing the organizational practices.

Business practice analysis was conducted to identify gaps between the DP&T/DOA business requirements and the PeopleSoft system. The RET mapped the appropriate process components to specific existing user activities and workgroup practices. The mapping helped users focus their attention on relevant portions of the system. For example, the payroll clerks used the metadata to determine which panels belonged to them.

Business practice realignment addressed gaps between system functionality and existing work practices. Once users understood the system's features and could navigate through it, they compared the system's inputs and outputs with their own needs. If gaps existed, they used the metadata to assess the magnitude of proposed changes. This information was then used to forecast the cost of further customizing the system. In a number of instances, these forecasts provided justification for changing the business practice instead of the system.

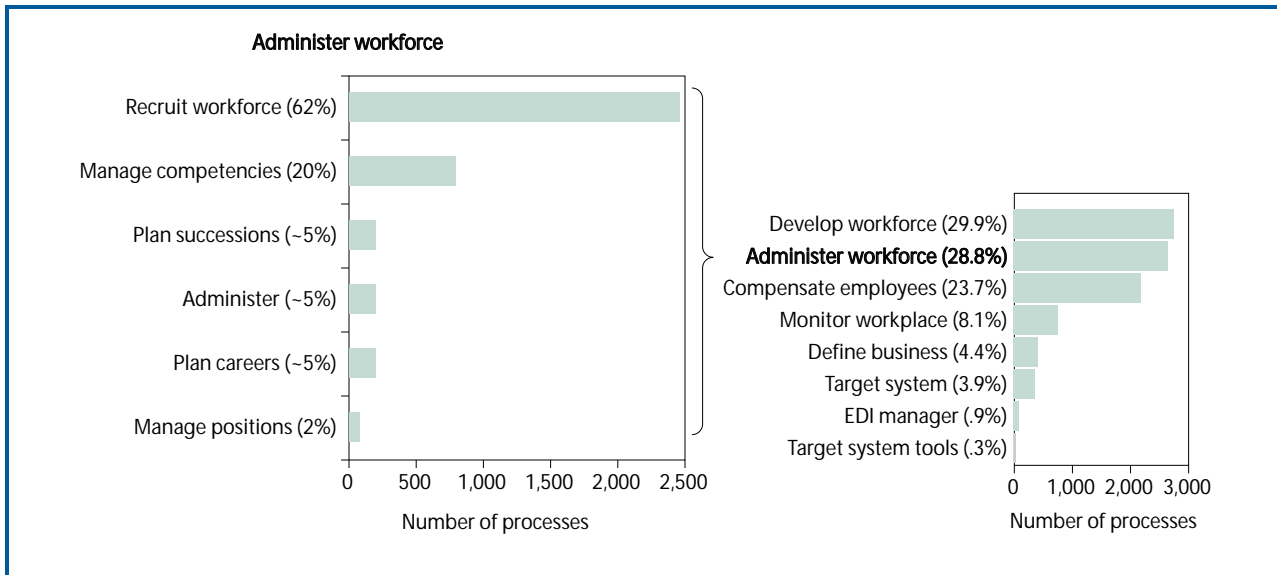


Figure 3. A statistically derived introduction to the PeopleSoft Administer Workforce module business process showing the complexity of the six components and how they fit into the homepage structure.

User training specialists also used the mappings between business practices and system functions to determine which combinations of *panels*, *menuitems*, and *menubars* were relevant to each user group. TheMAT was used to display panels in the sequence expected by the system users. By reviewing panels, users were able to swiftly become familiar with their areas. Additional capabilities for screen session recording and playback were integrated into the toolkit to permit development of system/user interaction “movies” and development of system test scripts.

Several additional metadata were incorporated into TheMAT that are not illustrated in Figure 2, including metadata describing LS1 and LS2, associations with system batch reporting programs, and user and user type metadata. The first supported the data conversion effort and assisted the transition team in developing physical data models. The data conversion subtask was the initial motivation for the metadata development work. More than 300,000 logical and millions of physical records had to be converted from LS1 and LS2 into formats usable by IHRIS. Each decision to convert a data item was recorded, permitting the tracking of the number of data items that had been mapped and converted. The metadata was queried to determine whether a specific data item had been associated with one or more IHRIS data fields, when the conversion had been accomplished, and often what code was used to accomplish the actual conversion.

The metadata helped the team to systematically organize the analysis of the PeopleSoft physical database design. A CASE tool, Visible Advantage, was integrated to extract the database design information directly from the physical database and integrate it into TheMAT, simplifying project documentation. Metadata was used to support the decomposition of the physical database into logical user views. We collected additional metadata to document how the system implements user requirements. This enabled us to track the status of individual requirements. Similarly, metadata was used as the basis for planning security access levels and privileges. Once we integrated the user metadata into TheMAT, we were able to use the system *menuitem/panel* hierarchy as the basis for defining database security views, each granting or denying *menuitem* or *panel* access according to various user profiles.

Statistical analysis was also useful for guiding metadata-based data integration from the two legacy systems. For example, the data metadata was used by the IHRIS transition team to map the legacy system data into PeopleSoft data structures. Statistical metadata summaries were also used by the RET to describe the system to users. For example, Figure 3 illustrates use of workflow metadata showing the number of processes in each homepage and the Administer Workforce PeopleSoft modules. It indicates the number of *components* associated with each homepage and that the Recruit Workforce was

the most complex component. These charts were used to show the Administer Workforce users why the recruiters were receiving separate training based on the relative complexity of the components comprising the Administer Workforce process.

The IHRIS project showed the effectiveness of managing a lot of information by maintaining a relatively small amount of metadata. While each of the three metadata types had independent value to the transition team, the sheer volume of system facts discouraged rapid comprehension. The integrated metadata made the system information accessible, encouraging implementation personnel to incorporate it as a regular part of their situation analysis and resolution activities. Storing the metadata in its most flexible and adaptable format permits different types of metadata to be integrated, bringing economies of scale to metadata management.

Based on this case study, we have found that the collection and management of metadata is not as expensive as one would expect. Some metadata can be maintained using CASE or metaCASE tools—especially if CASE tools were used to develop the system. Integrated metadata repositories permit systematic and flexible manipulation, management, and control of physical and logical data via SQL and graphical browsers. System metadata that is easy to create and maintain can be a valuable organizational asset. Reasonable investments in targeted reverse engineering of new systems can provide metadata that can be immediately reused as the systems are reengineered, greatly facilitating implementation. ❖

ACKNOWLEDGMENTS

This investigation is sponsored by the Virginia Department of Personnel & Training. Bill Girling, manager of systems, saw the need and developed the initiative. We thank Information Systems Research Institute research associates Leslie Borman, Sasipa Chankaoropkhun, Pawan Pavichitr, and Kim Boos for their professional contributions to IHRIS, ISRI research associates Lynda Hogdson and Sirirat Taweewattanaprecha for their support in preparing this article, and Pat Krebs at PeopleSoft for a technical review of the materials.

REFERENCES

1. S. Hsu, "Welfare Overhaul by Virginia Faces Computer Glitch: Massive Automation Program Behind Schedule; Costs Mount," *Washington Post*, 24 June 1996, p. A-1.
2. M. Hardy, "VA. Drops Pact on Medicaid: EDS' Failure to Finish System Blamed," *Richmond Times Dispatch*, 25 April 1997, p. A-1.
3. ISO 11179:195-1996 Information Technology - Specification and Standardization of Data Elements.

4. C. Hsu et al., "Metadatabase Modeling for Enterprise Information Integration," *J. Systems Integration*, Vol. 2, No. 1, Feb. 1992, pp. 5-37.
5. C. Hsu et al., "Core Information Model: A Practical Solution to Costly Integration Problems," *Computers and Industrial Engineering*, Vol. 1, 1994, pp. 75-83.
6. S. Weibel et al., "An Element Set to Support Resource Discovery: The State of the Dublin Core," *Intl. J. Digital Libraries*, Vol. 1, Issue 2, Jan. 1997, pp. 176-186.

About the Authors



Peter Aiken is a research director with Virginia Commonwealth University's Information Systems Research Institute. He was a computer scientist with the Defense Information Systems Agency from 1992 to 1997. His main interests are systems engineering, systems requirements, data reverse engineering, and

hypermedia-based software requirements engineering tools and techniques.

Aiken received a PhD from George Mason University.



Ojelanki K. Ngwenyama is an associate professor of information systems at Virginia Commonwealth University and a research professor at Aalborg University, Denmark. His research focuses on understanding how people in everyday social activity appropriate and innovate upon information technology to solve relevant problems.

Ngwenyama received an MBA from Syracuse University, an MS in computer and information sciences from Roosevelt University, and a PhD in computer sciences and information systems from the State University of New York.



Lewis Broome is a business analyst for Innovative Business Solutions inc. (IBSi) in Richmond, Virginia. He specializes in the application of information engineering to private businesses and public-sector entities.

Broome received an MS in business information systems from Virginia

Commonwealth University.

Address questions about this article to Aiken at Virginia Commonwealth University, Department of Information Systems, 1015 Floyd Avenue - Room 4170, Richmond, VA 23284-4000; paiken@computer.org.