AI Magazine Volume 14 Number 3 (1993) (© AAAI)

Tennessee Offender Management Information System

David Reynolds and Tim Beck

■ This article describes the integration of a knowledge-based system with a large COBOL-DB2–based offender management system. The knowledge-based application, developed for the purpose of offender sentence calculation, is shown to provide several benefits, including a shortened development cycle, simplified maintenance, and improved accuracy over a previous COBOL-based application.

The state of Tennessee manages 20 correctional institutions, 39 field offices for parole and probation, and 16 community corrections grant programs. Sentences for the 50,000 offenders vary from community work release and probation to lifelong incarceration. Tennessee was one of 38 states required by court order to improve prison conditions and reduce overcrowding; it is the target of over 300 inmate lawsuits each year. Under a Federal Court Consent Decree in February 1990, the Tennessee Department of Correction (TDOC) hired Andersen Consulting to design, install, and implement the automated Tennessee offender management information system (TOMIS). Completed in June 1992, TOMIS manages the entire correctional process from sentencing through incarceration to release. The new $14 million system is the largest and most comprehensive computer system ever developed in the field of corrections.

Problem of Sentence Calculations

Among the many problems facing TOMIS was calculating the various types of sentences for offenders, one of the most complicated functions performed by the Department of Correction. The importance of this function is obvious in terms of determining the accurate release dates for offenders. Sentences were originally calculated by TDOC using a program coded in a traditional third-generation coding language (COBOL). Because of the complexity of the sentencing laws, this program provided correct results only 80 percent of the time, forcing sentence-management personnel to check all results by hand. All of an offender's sentences must then be recalculated each time something, such as good-behavior credits, occurs to change a criminal's release date. Correctional officers, judges, and offenders had been computing sentences manually, risking errors by inconsistently applying the sentence-calculation rules. Expert help was in short supply because only a few people in the state fully understood the end-to-end sentencing process. It was not uncommon for an offender to be penalized because of an incorrect interpretation of the law. Even after a sentence had been calculated, there was often great confusion. Families struggled to understand when offenders would be released, and judges wondered how much time convicted criminals would actually spend behind bars.

In July 1990, the general design of a more accurate COBOL-based sentence-calculation subsystem was begun. Tennessee's current automated sentencing process was inaccurate and unreliable. During the detailed design effort, it became apparent that the sentencing rules were much more complicated than previously realized and that the current sentencing process was not documented anywhere except in the state's law documentation (Tennessee Code Annotated). The only true method of calculating sentences was understood by two analysts in the sentence-management department. Based on these issues, the project team searched for an alternative method of automat-

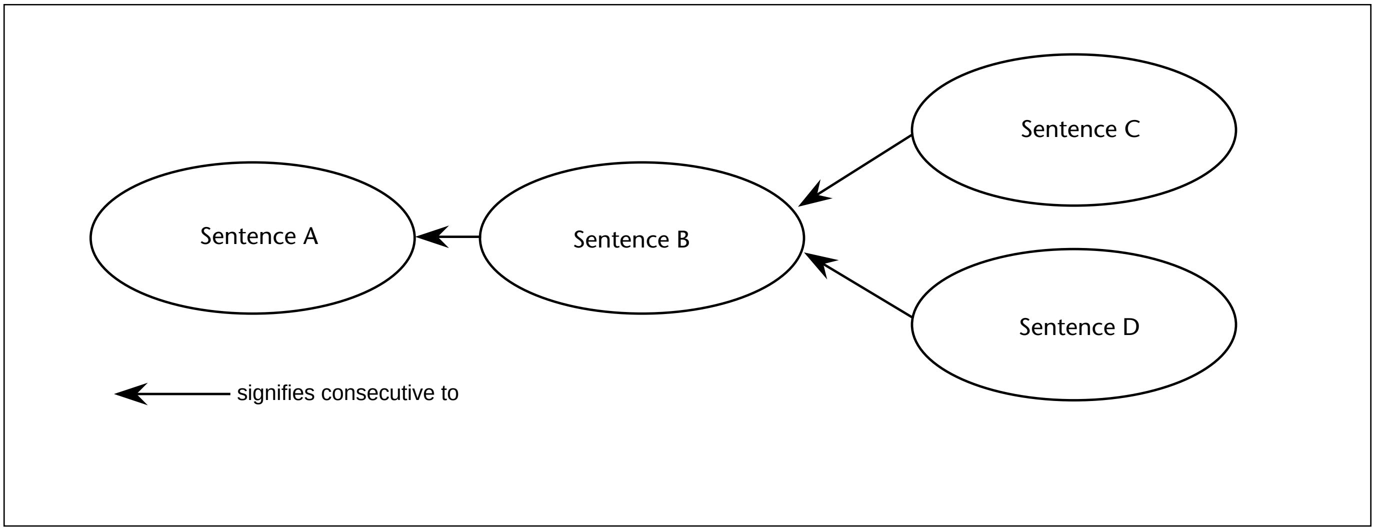

Figure 1. Consecutive Sentences of an Offender.
Sentences C and D are consecutive to sentence B, and sentence B is consecutive to sentence A.

ing the calculation of sentences. This analysis revealed the need for a knowledge-based system. The knowledge-based system would have the technical sophistication to incorporate the vast and complex rules of sentencing an offender and easily embody the iterative combination of the expert's knowledge of sentencing and the ever-changing laws and guidelines.

Conviction and sentencing is a three-step process: First, information on the offender's criminal and employment histories, educational background, and medical records is collected and made available for consideration. Then, the judge sentences the offender based on the applicable sentence laws for the date the offense was committed. Finally, based on the laws the individual was sentenced under and the coordination of multiple sentences, calculations including 10 to 12 different dates are produced for each sentence of an offender. In addition, summary offender sentence dates are calculated based on the underlying individual sentence dates. Sentencing information must be captured at the beginning of the process. Data gathering is done through online data entry of the judgment order, which provides the baseline information needed to calculate sentences.

However, the intricacies of the sentencing laws cause the calculations to be much more complex than is readily apparent. Each offender can have an unlimited number of sentences. Sentences can be forced to run consecutively to a previous sentence, which means the dates of a sentence are dependent on the related sentence dates of the offender's previous sentence. For example, to calculate sentences A, B, C, and D of an offender, as shown in figure 1, it must be determined which sentence is not consecutive to any others. In this case, A is the sentence that must first be calculated because its dates do not depend on a previous sentence. Once A is figured, B can be calculated based on A's calculations. Then, either C or D can be calculated based on B's calculations. Each sentence calculation hinges on its type and its consecutive sentence's type. Five different types of sentences exist in Tennessee based on the legislated sentence laws of the state. These sentence laws are reform 1194, judge, class X, determinate, and indeterminate.

Nuances of the laws, the occurrence of related sentences, and various other factors contribute to the sentence-calculation complexity, for example, the different types of credits that an offender can receive to reduce the sentence length. These credits include a prisoner's performance sentence credit, a prisoner's sentence-reduction behavior credit, a prisoner's sentence-reduction program credit, a graduate equivalency degree credit, a literacy program credit, a drug-alcohol program credit, and a good-conduct credit.

Adding to this burden are any changes that need to factor into the sentencing equation. Offender release dates are recalculated when an offender receives credits for good behavior or work programs, credits are removed for disciplinary actions, parole eligibility is extended because of disciplinary actions, or time is added because of escape or parole or probation violations. Initial offense, sentence, and credit data for newly arriving offenders also require recalculating an offender's sentences. Sen-

Offender Release Dates	Involuntary Manslaughter	Assault and Battery	Burglary, 1st Degree	Escape
_				
Sentence Effective	03/23/1975	11/25/1986	11/25/1986	05/25/1994
Expiration	11/25/1986	05/25/1991	05/25/1994	05/25/1996
Full-Term Expiration	10/25/1995	10/25/2000	10/25/2003	10/25/2005
Mandatory Parole	05/25/1986			
Regular Parole	10/25/1981			
Probationary Parole	10/25/1980			
Release Eligibility		04/26/1982	03/21/1983	10/26/1983
Safety Valve	10/25/1978	09/19/1979	04/03/1980	08/13/1980

Table 1. Summary of Release Dates for Example Scenario.

tences are recalculated for commissioner additional sentence orders and a governor's pardon or commutation. Parole board date order changes, probation judgments, and new laws and sentencing guidelines enacted each year by the state legislature also affect sentence calculations. Because offenders are often sentenced under multiple laws, these changes can create a complex equation for judges and offenders to decipher. These complex calculations made the use of knowledge-based technology necessary and the traditional language (COBOL) completely inadequate for the programming task.

As an example of a possible sentencing problem, consider the following scenario: On 23 March 1975, John Doe was arrested for involuntary manslaughter, which was allegedly committed on 17 January of the same year. He was sentenced on 8 June 1975 to serve a minimum of 10 years to a maximum of 20 years under the indeterminate sentencing law that was in effect at the time of his offense. Under the indeterminate sentencing law, Doe received the following dates: sentence effective date, 03/23/1975; expiration date, 04/23/1986; full-term expiration date, 03/23/1995; mandatory parole date, 10/23/1985; regular parole date, 03/23/1981; probationary parole date, 03/23/1980; and safety valve date, 03/23/1978.

The *sentence effective date* is the date that his sentence begins. This date is prior to the date that his sentence was actually imposed because of jail credit received while he waited for trial. The *full-term expiration date* is the date the sentence expires if no credits are awarded. Under the indeterminate sentencing law, Doe received three different parole dates. On the earliest of these parole dates, he would be eligible for release from prison to serve the remainder of his sentence in the community. Finally, because of overcrowding in the prison, Doe received a *safety valve date,* which is a fraction of his time to serve until parole. Under these conditions, he would be eligible for release on this date.

On 2 July 1982, while serving his sentence at XYZ Prison, Doe escaped and remained at large in the community until 8 February 1983, at which time he was arrested for committing first-degree burglary and assault and battery. He was returned to prison, where his previous sentence dates were extended by 216 days (the number of days he was on the lam). On 2 March 1983, Doe was convicted of burglary and assault and battery and received sentences of 8 and 5 years, respectively, under the judge sentencing law. These sentences were deemed to run concurrently with each other and consecutively to his previous indeterminate sentence. Additionally, he received a two-year judge sentence for his escape, which was said to run consecutively to all previous sentences. Table 1 summarizes the release dates for all sentences, and figure 2 depicts the coordination of Doe's sentences. As seen in table 1, the offender received the same basic dates for his last three sentences as he did for his first sentence. The parole dates are the exception; they were replaced by the release eligibility date

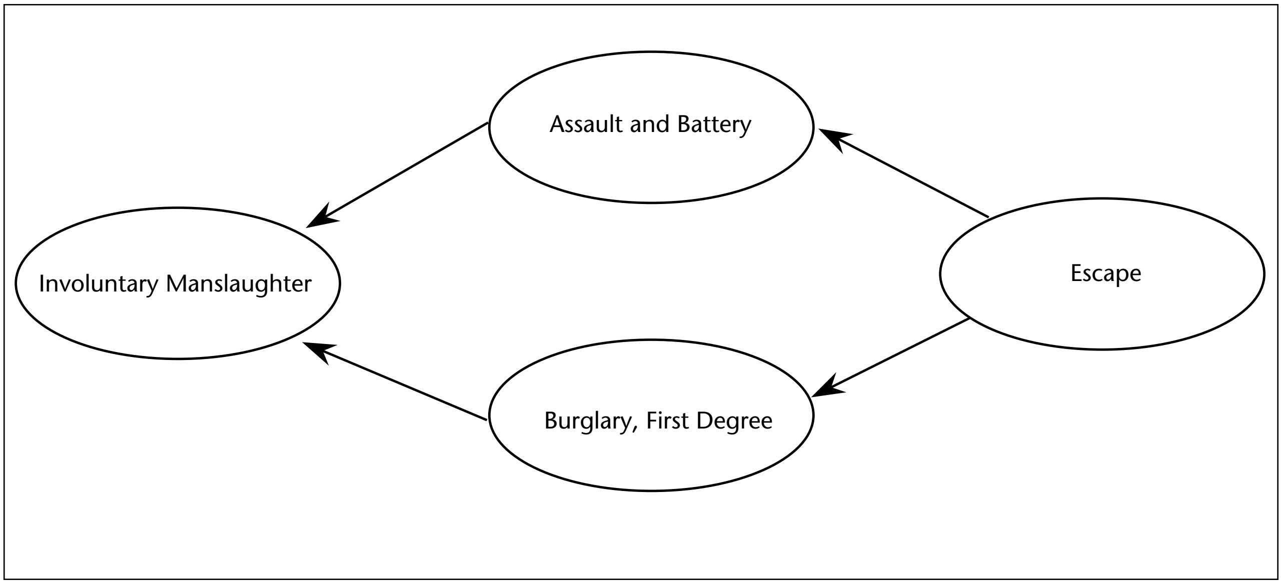

Figure 2. Coordination of Sentences for Example Scenario.

under the judge sentencing law.

To compute the sentences, the knowledge base went through the following process: First, the order in which the sentences could be processed was determined through pattern matching. This process continues throughout the calculations. Once a valid sentence was determined, the appropriate backward-chaining rules were invoked. For this scenario, rules for indeterminate nonconsecutive, judge consecutive to indeterminate, and judge consecutive to judge were used. Finally, for sentences deemed to run consecutively to multiple sentences, the correct sentence to which the consecutive sentence should be added had to be determined. These were the conditions for the sentence given for the escape because it was deemed to run consecutively to all previous sentences. This calculation process is more clearly defined in the next section.

Development

The AION development system (ADS-PC) was the chosen knowledge-based technology. ADS can run in a mainframe environment in batch mode, a requirement for the TOMIS sentence-calculation knowledge base. ADS also supports pattern matching and backward-chaining rules, which are vital for the complex functions of the sentencing laws. Because of the number of rules involved in the sentence-calculation process, it was important to isolate logical groups of rules so that the layout of sentence laws was simple and readily maintainable. ADS facilitates this process with the use of isolated entities called *states*. These states, which can contain processes, functions, rules, and so on, allowed for a logical grouping of sentencing rules that was easily maintainable. Additionally, ADS was the most cost-effective knowledge-based technology for TOMIS's sentence-calculation subsystem. Finally, although ADS is a proven and advanced technology, it also offers a low learning curve, which was an added benefit.

During the new sentencing analysis, a state system analyst and an Andersen senior analyst laid out approximately 2000 rules for the knowledge base. Concurrent with this process, the architecture of the knowledge base was decided on. The designed architecture consists of two state hierarchies. The first hierarchy, shown in figure 3, is for process control and includes states for entering an offender's sentences, processing the sentences, and storing the results. The *process state* is where the order of the sentences for an offender is determined. This determination is made using a single pattern-matching rule, which states that an offender's sentence can be calculated only if (1) it is nonconsecutive (independent of the offender's other sentences) or (2) it is a consecutive sentence, and it's related sentences have already been calculated.

The second hierarchy is for calculating an offender's sentences once an order is determined. This hierarchy begins, as shown in figure 4, with an entry processing state that declares global calculation parameters; sets up goals of calculations; and determines sentencing dates whose calculations are common to

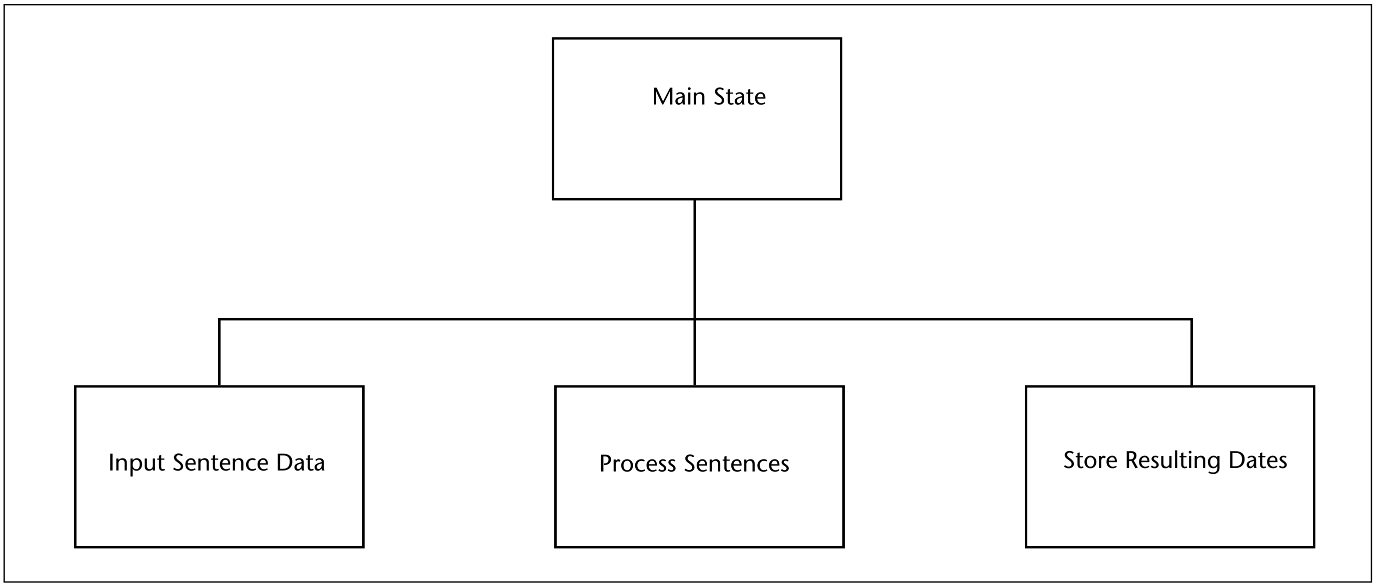

Figure 3. Process Control State Hierarchy.

all sentencing laws, regardless of type. The next state that a sentence enters depends on its type. States contain entry conditions to ascertain the sentence type being processed. Two variables determine which state a sentence enters next: (1) the type of sentence, whether life, habitual, or neither, and (2) the consecutive sentence's type and whether its consecutive sentence is life, habitual, or neither. Life and habitual sentences differ from normal sentences in that they never expire; however, the offender can be released on parole for these sentences. The possibilities of sentences are as follows:

Reform 1194	Indeterminate
Reform 1194 Life	Indeterminate Habitual
Judge	Determinate
Judge Life-Habitual	Determinate Life
Class X	Determinate Habitual
Class X Life-Habitual	

These states contain entry conditions to determine sentence type and backward-chaining rules to find necessary parameters to solve the sentence equations.

For example, if sentence A from figure 1 is judge life, and sentence B is determinate life, sentence A is calculated first. In figure 4, sentence A enters the entry processing state and then the judge state. Processing continues through the nonconsecutive judge life-habitual state because this sentence has no consecutive sentences. Once all calculations are complete for sentence A, processing begins for sentence B through the entry processing state and enters the determinate state. Sentence B would then enter a state called determinate life consecutive to judge life-habitual, using calculations from sentence A's dates to determine B's sentence-date calculations. In figure 4, only the paths for judge and judge life-habitual are shown. Calculations must follow these complex paths because each combination of sentence types contains unique calculations.

The design of the knowledge base led to a highly structured environment. Each state in the second hierarchy contains certain entry conditions that reduce the complexity of rule premises. In fact, because of entry conditions, many rules require no premise. The design streamlines maintenance by providing a logical grouping of rules easily extensible to new laws and modifications.

Architecture

The TOMIS technical architecture is mainframe based: an MVS-ESA system running under DB2, using CICS-VS as its online monitor. The system was developed using FOUNDATION, Andersen Consulting's computer-aided software engineering tool. The TOMIS sentence-calculation knowledge base was developed on a PS-2 model 70 using ADS-PC.

Test cycles were created to test every variation of a sentence calculation, and the results were validated against hand calculations made by sentencing experts in Tennessee's sentence-management department. Once this automated testing process was successful for all possible sentence combinations, the knowledge base

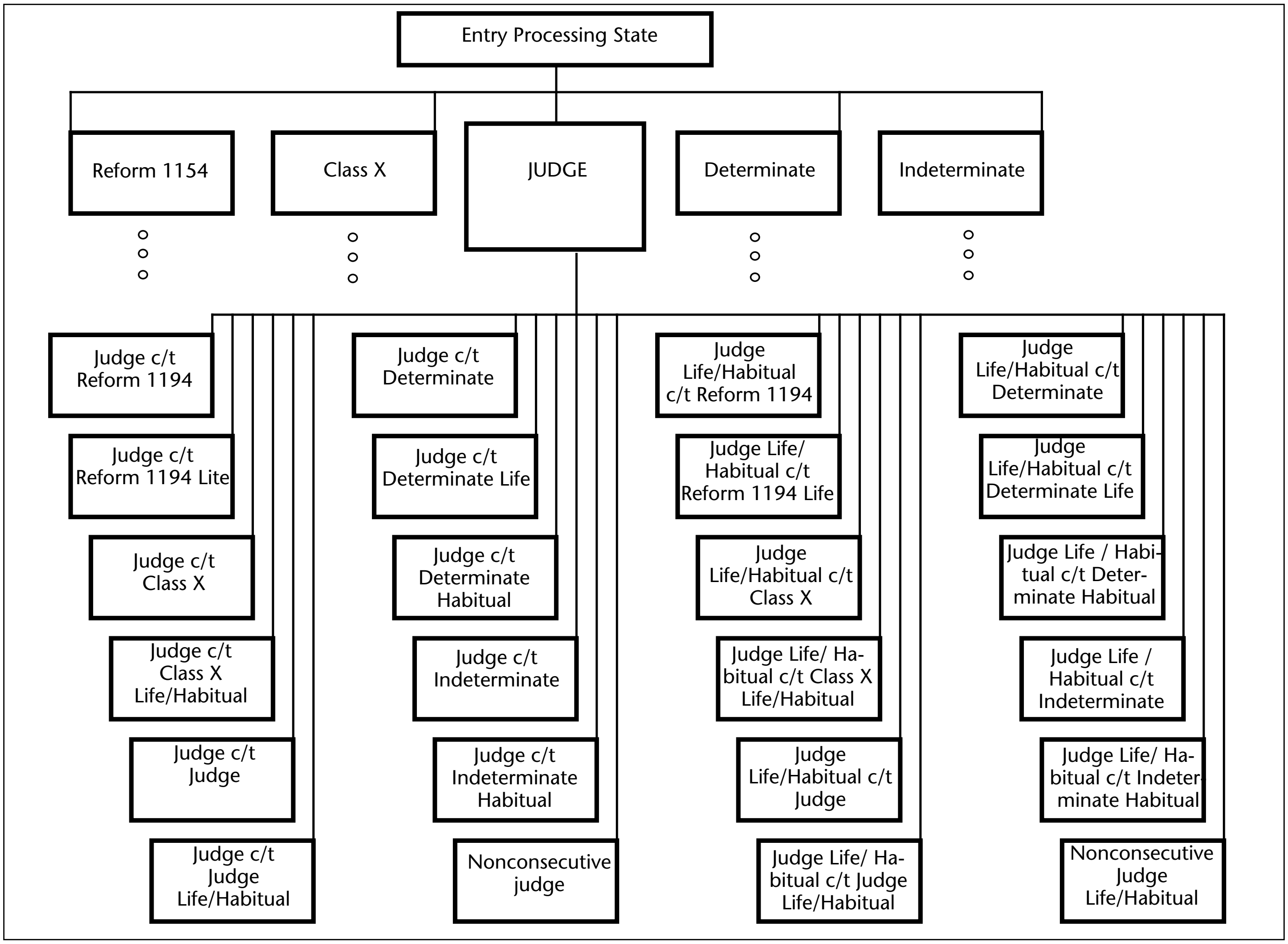

Figure 4. State Hierarchy for Sentence-Calculation Rules (c/t = consecutive to).

was transferred to the mainframe, and all test cycles were rerun. Because of performance requirements, the knowledge base was compiled from Pascal source code using AION's high-performance option to create an executable load module for faster processing. This tool increased performance by tenfold.

Throughout the month, sentences are added, modified, and deleted on the TOMIS mainframe database. If a modification occurs, a sentence must be recalculated, and a recalculation flag is set for an offender. An overnight batch job runs each night to compute offender sentences. All offenders with their recalculation flag set have their sentences calculated for the first time or recalculated to incorporate any sentence changes in the offender sentence dates. A COBOL program extracts all necessary sentencing information from the DB2 database on the mainframe and stores it in flat files, as shown in figure 5. The knowledge base then reads the data from the flat files, processes all sentences, and stores the calculations in additional flat files. A COBOL program reads the updated sentence data from the flat files and updates the DB2 database on the mainframe. The mainframe version of the knowledge base functions as the calculation engine to process sentence calculations.

Sentence-Calculation Workstation

A personal computer (PC) version of the sentence-calculation knowledge base was developed, enabling sentence-management personnel to determine the effects of modifications to an offender's sentence parameters without affecting mainframe production data. Originally, such procedures were manually calculated, often resulting in inconsistent and inaccurate data. Management also wanted a projection tool for reporting statistics on an offense statute basis to help determine global

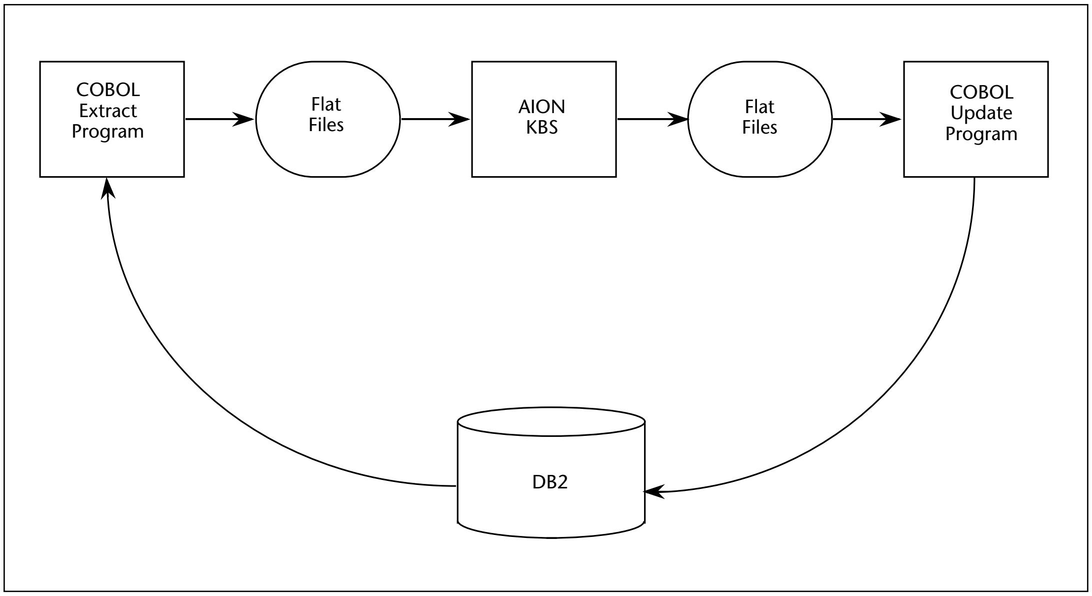

Figure 5. Information Flow of Sentence-Calculation Data.

changes to sentence laws for the entire offender population. The workstation version of the knowledge base was developed to provide sentencing personnel with a responsive tool to facilitate this what-if type of analysis.

The workstation uses a graphic user interface to provide two primary functions: offender calculations and offense statute groups. The *offender calculation* function allows sentence-management personnel to view an offender's sentence dates, make modifications and recalculate an offender's sentence dates, and view a graphic representation of the interaction of all sentences for an offender. Sentences can be made to run consecutively to a previous sentence to determine the effect of changes in the flow of offender sentences. The *offense statute group* function allows personnel to calculate the parole-eligibility dates for a group of offenders. The offenders are grouped by offense statute (for example, all drug-related crimes). Statistics are calculated to report how many offenders would be eligible for release by a given date. This tool allows TDOC to determine the effects of legislation changes to ease overcrowding in state institutions.

Each month, data needed for the calculation of offender sentence dates are downloaded from the mainframe production DB2 database to the PC DBASE database. Sentence calculations can now be performed as requested, providing sentence-management personnel with immediate results without affecting production data. The workstation provides TDOC with the ability to manipulate data in an isolated environment to perform what-if analysis on sentences and offense statute analysis on the total population. The mainframe version functions as a calculation engine, but the PC version is used by TDOC personnel to understand sentences and perform analysis.

Deployment

TOMIS has been in use since February 1992. The total system required approximately 23,000 person-days for development at an estimated cost of $14 million. The development of the knowledge base subsystem required 2 months of user documentation, 3 months of design and development involving a programmer, an analyst, and a technical architect, and 2 months of acceptance testing involving an analyst and a user, a total of about 300 person-days. The addition of a user interface and offense statute functions for the workstation version of the knowledge base required another 100 person-days.

The mainframe component of the knowledge base executes as an overnight batch program 365 times a year. About 10 percent of all sentences are recalculated daily, all sentences are recalculated at the beginning of each month, and 7500 new sentences are calculated monthly. The workstation version is currently used to conduct what-if scenarios on habitual offenders by the Parole Review Board and sentence-management personnel. In the future, the workstation will be used both on judges' desktops and by sentence-management personnel who will be able to fully benefit from its ability to forecast the effects of changes to offenders' sentences.

TOMIS triggered organizationwide change, pairing reengineered policies and processes with the new technology. TOMIS reduced the manually intensive, paper-driven activities in TDOC and provides facilities for better management of prison activities and programs. The knowledge-based application calculates sentences and release dates and has increased accuracy from 80 percent to 100 percent. By eliminating the need for manual calculation of sentences, which took an average of 5 hours for each sentence, the TOMIS knowledge base saves TDOC more than 37,500 hours of manual effort each month. Because the knowledge of sentence-calculation experts has been embedded in over 2000 knowledge-based rules, personnel with only limited knowledge can properly calculate the sentence dates of an offender. Sentence calculations can now be performed accurately without relying on a few who understand the complicated process. The workstation version will be used to test changes to laws and guidelines, allowing the state to gauge the impact of changes before they are in production.

Judges, offenders and their families, and prison personnel benefit from the reduced complexity of the system, with the guarantee that all offenders are treated consistently and with the reduced opportunity for incidents and offender lawsuits. Accounting for changing laws and sentencing guidelines entails a simple change to the knowledge base rather than a change to complex COBOL code. TOMIS was designed to provide solutions for other states as well. Its functions address many of the issues for which other states are under court orders. The system was developed with standard system analysis and design techniques, resulting in reusable and maintainable software.

Maintenance

The state system analyst involved in the development of the knowledge base is responsible for its maintenance. The analyst has been trained in the layout of the rules within the knowledge base, which facilitates modifications and additions. For a modification, the analyst needs only to find which rules the change affects by looking at the layout of rules in the knowledge base and updating accordingly. Once the change is made, new test conditions and test data are developed to ensure the accuracy of the modification. Finally, regression testing is performed to ensure no adverse effects to previous results. Since its deployment in February 1992, only one modification to the knowledge base has been necessary.

Additions to the system are primarily expected in one of two forms. The simplest form would be the addition of a new date to be calculated for one or more sentencing laws. In addition to the new input-output requirements, this modification would be implemented by adding a new goal to the backward-chaining calculations and the addition of appropriate date-calculation rules in the affected states.

The second expected addition and the more complex of the two would be the adoption of a new sentencing law. In this case, a new sentence type would be added to the five main types of sentencing laws. This addition would most easily be done by copying an existing type, along with its subsequent paths, and then modifying all related states. The new type must also be added to all existing paths as a possible consecutive sentence state. New test cycles would be created to include all new states and expected results calculated by hand. As described earlier, regression testing would also be performed to ensure that there were no adverse effects on existing states.

Conclusion

TOMIS has replaced manually intensive, paper-driven activities with computer-based functions. It has reduced errors, streamlined business activities, and provided more accurate and timely information. Determining and maintaining each offender's sentence is an extremely sophisticated process because of complex sentence laws and ongoing legislative activities forcing changes in the sentence statutes. The TOMIS sentence-calculation knowledge base automates this process to provide precise information for release determination, parole and probation eligibility, and population forecasting.

Acknowledgments

Thank you to all those who made the TOMIS sentence-calculation subsystem a success, including Brandon Powers and Michal King from the state of Tennessee and David Wilkins, David DeLozier, Katrina Liberto, Matt Remanick, and Brian Scammon from Andersen Consulting. A special thanks goes to Katrina Liberto for her assistance in preparing this article.