TENNESSEE OFFENDER MANAGEMENT INFORMATION SYSTEM

David Reynolds

Andersen Consulting
Suite 1100
424 Church Street
Nashville, Tennessee 37219

Tim Beck

Tennessee Department of Correction
Ground Floor
Rachel Jackson Building
Nashville, Tennessee 37243

Abstract

This paper describes the integration of a knowledge-based system (KBS) within a very 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.

System Background

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. The State of Tennessee was one of 38 states required by court order to improve prison conditions and/or reduce overcrowding, and 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) employed Andersen Consulting to design, install, and implement the automated Tennessee Offender Management Information System (TOMIS). Completed 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 the problem of 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). Due to the complexity of the sentencing laws, this program provided correct results only 80% 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 an event, 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 as 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 due to 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 TOMIS began the general design of a more accurate COBOL-based sentence calculation subsystem. The State of Tennessee's current automated sentencing process was not accurate and was 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 automating 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, while easily embodying the iterative collection 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

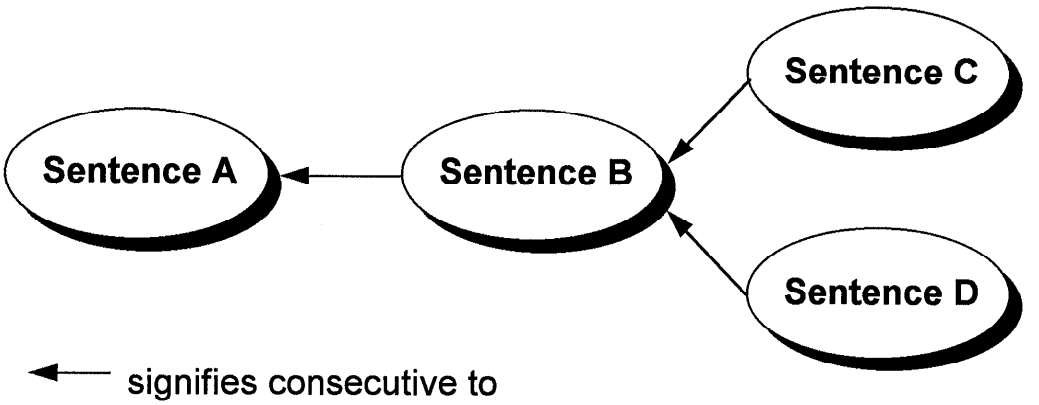

Figure 1. Consecutive Sentences of an offender. Sentences C and D are Consecutive to Sentence B, and Sentence B is Consecutive to Sentence A.

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 ten to twelve 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. This is done through on-line 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 apparent. Each offender can have an unlimited number of sentences. Sentences can be forced to run consecutive to a previous sentence. This means the dates of a sentence are dependent upon the related sentence dates of the offender's previous sentence. For example, to calculate sentences A, B, C, and D of an offender 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 now 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. One example of such factors are the different types of credits that an offender can receive to reduce their sentence length. These include:

- Prisoner's performance sentence credit
- Prisoner's sentence reduction behavior credit
- Prisoner's sentence reduction program credit
- GED credit
- Literacy program credit
- Drug/Alcohol program credit
- 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, or parole eligibility is extended due to disciplinary action or time is added because of escape or absconding on parole or probation. Initial offense, sentence, and credits data for newly arriving offenders also requires the recalculation of an offender's sentences. Sentences 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 affect sentence calculations also. 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 3GL (COBOL) completely inadequate for the programming task.

As an example of a possible sentencing problem, consider the following scenario. On March 23, 1975, John Doe was arrested for committing Involuntary Manslaughter on January 17 of the same year. He was sentenced on June 8, 1975 to serve a minimum of 10 years to a maximum of 20 years under the Indeterminate sentencing law which was in effect at the time of his offense. Under the Indeterminate sentencing law, John 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
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, due to jail credit received while waiting for trial. The Full-Term Expiration Date is the date the sentence expires if no credits were awarded. Under the Indeterminate sentencing law, John received three different parole dates. On the earliest of these parole dates, John would be eligible for release from prison to serve the remainder of his sentence in the community. Finally, due to overcrowding in the prison, John received a Safety Valve Date, which is a fraction of his time to serve to parole. Under these conditions, John would be eligible for release on this date.

On July 2, 1982, while serving his sentence at XYZ Prison, John escaped and remained at large in the community until February 8, 1983, at which time he was arrested for committing Burglary-1st Degree 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 March 2, 1983, John was convicted of Burglary and Assault and Battery, and received sentences of eight and five years, respectively, under the Judge sentencing law. These sentences were deemed to run concurrently to each other and consecutively to his previous Indeterminate sentence. Additionally, John received a 2-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 graphically depicts the coordination of John's sentences. As can be seen from Table 1, the offender received the same basic dates for his last three sentences as he did for his first sentence. The exception to this are the parole dates, which were replaced by the Release Eligibility Date 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 via 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 consecutive to multiple sentences, the correct sentence to which the consecutive sentence should be added had to be determined. This was the case for the sentence given for Escape, since 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

Table 1. Summary of Release Dates for Example Scenario

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

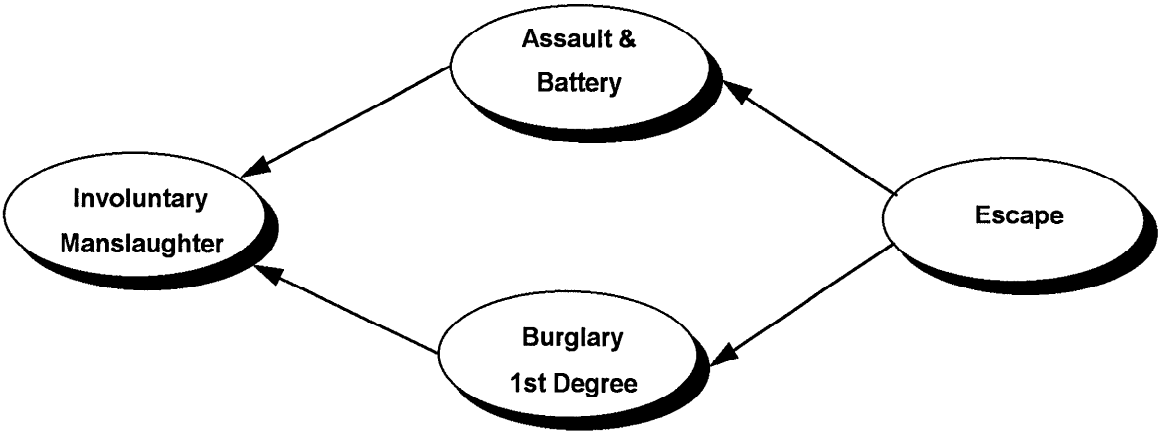

Figure 2. Coordination of Sentences for Example Scenario

rules vital for the complex functionality 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 through the use of isolated entities called "states". These states, which can contain processes, functions, rules, etc., allowed for a logical grouping of sentencing rules which was easily maintainable. Additionally, ADS was the most cost effective knowledge-based technology for the purposes of TOMIS' sentencing 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 systems analyst and an Andersen senior analyst laid out approximately 2,000 rules for the knowledge-base. Concurrent to this process, the architecture of the knowledge-base was decided upon. The designed architecture consists of two state hierarchies. The first hierarchy. shown is 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 is accomplished with a single pattern-matching rule, which states that an offender's sentence can be calculated only if: (1) it is non-consecutive (independent of the offender's other sentences) or (2) if 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 order is determined. This hierarchy begins 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 all sentencing laws, regardless of type. The next state a sentence enters

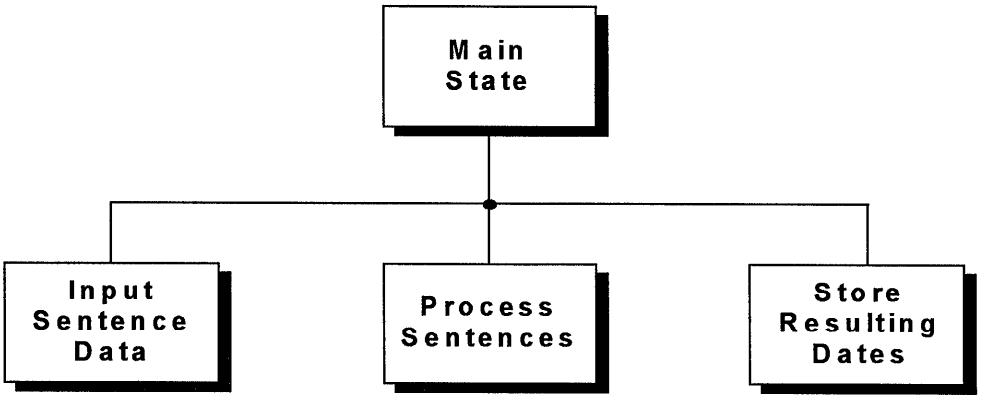

Figure 3: Process control state hierarchy.

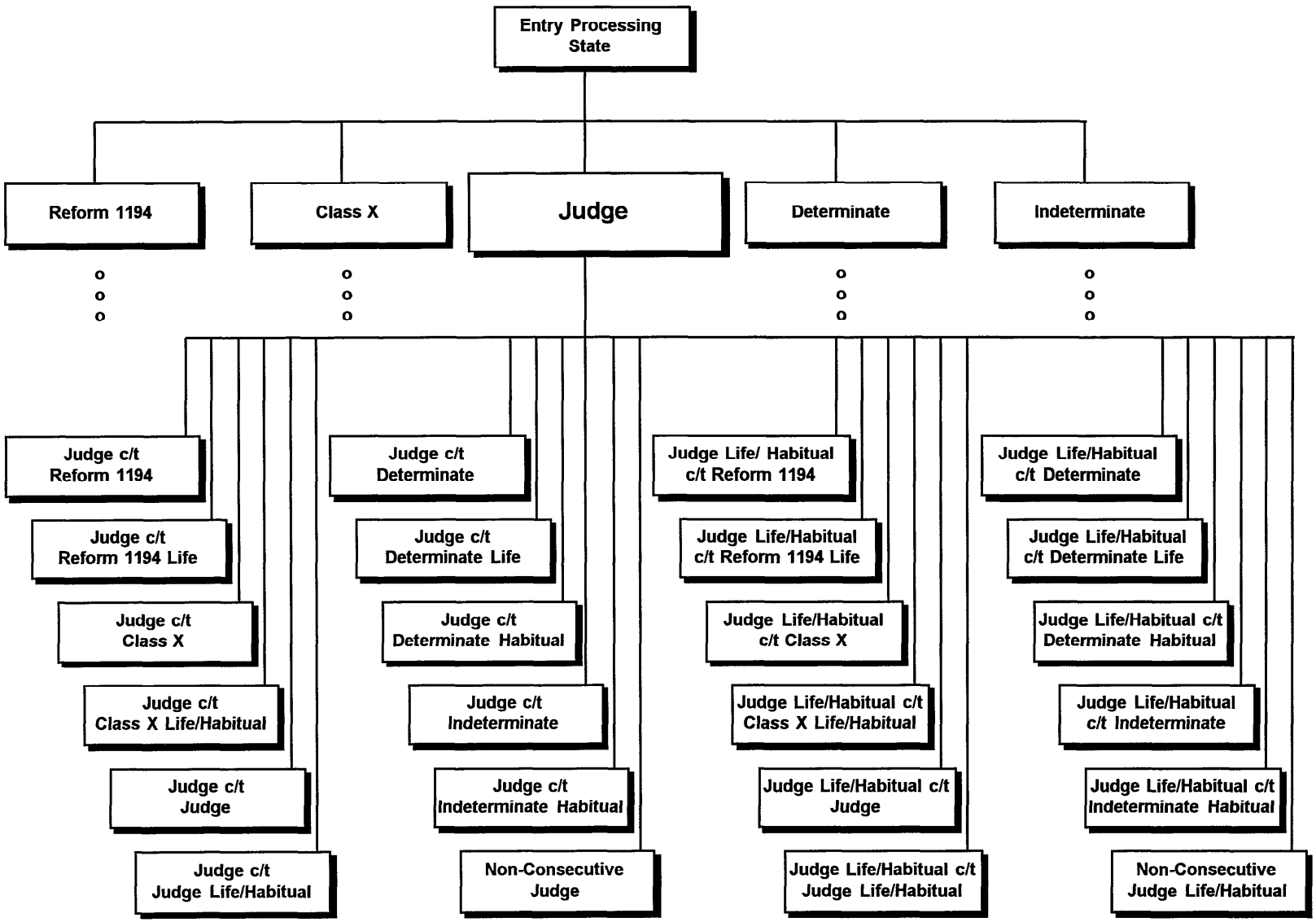

Figure 4. State Hierarchy for Sentence Calculation Rules ('c/t' signifies consecutive to).

depends on its type. States contain entry conditions to ascertain the sentence type being processed. Two variables determines which state a sentence enters next: 1) if the sentence is life, habitual, or neither and 2) its consecutive sentence's type, and if its consecutive sentence is life, habitual, or neither. Life and habitual sentences differ from normal sentences in that they will 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 Non-consecutive Judge Life/Habitual state, since 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, due to 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 and using CICS/VS as its on-line monitor. The system was developed using FOUNDATION, Andersen Consulting's CASE tool. TOMIS' sentence calculation knowledge-base was developed on a PS/2 Model 70 using the AION Development System (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 of the State of Tennessee's sentence management department. Once this automated testing process was successful for all possible sentence combinations, the knowledge-base was transferred to the mainframe and all test cycles were re-run. 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 will have their sentences calculated for the first time or recalculated to incorporate changes to any sentences 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 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 changes to sentence laws for the entire offender population. The workstation version of the knowledge-base was developed to provide sentencing personnel a responsive tool to facilitate this "what if" type analysis.

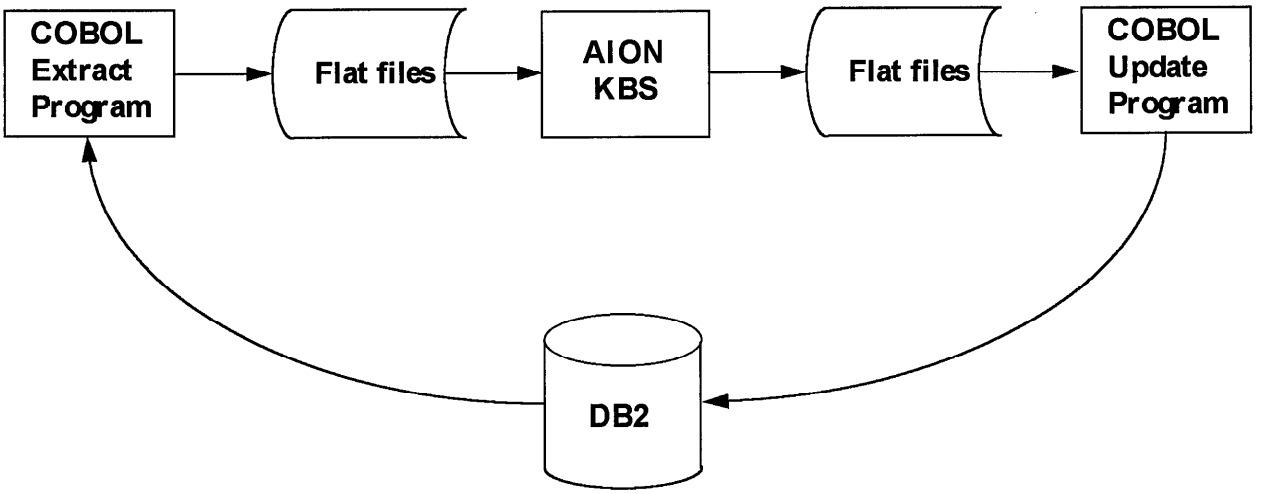

Figure 5. Information Flow of Sentence Calculation Data.

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

Each month, data needed for the calculation of offender sentence dates is 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. While the mainframe version functions as a calculation engine, the PC version is used by TDOC personnel to understand sentences and perform analysis.

Deployment

TOMIS has been in production 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 two months of user documentation; three months of design and development involving a programmer, analyst and technical architect; and two months of acceptance testing involving an analyst and 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% of sentences are recalculated daily, all sentences are recalculated at the beginning of each month, and 7,500 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 by sentence management personnel. In the future, the workstation will be used both on judges' desktops and by sentence management personnel to fully benefit from its ability to forecast the effects of changes to offenders' sentences.

TOMIS triggered organization-wide change, pairing re-engineered 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% to 100%. By eliminating the need for manual calculation of sentences, which took an average of five hours per sentence, the TOMIS knowledge-base saves TDOC more than 37,500 hours of manual effort each month. Since the knowledge of sentence calculation experts has been embedded in over 2,000 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 the 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 systems analysis and design techniques resulting in reusable and maintainable software.

Maintenance

The state systems 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, 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 I/O 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 form of an 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 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 above, regression testing would also be performed to ensure no adverse affects to 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; 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 paper.