COURT INFORMATION TECHNOLOGY OFFICERS CONSORTIUM

Managed Randomized Judicial Assignments in a Case Management System

by Barbara Holmes, Administrative Office of Pennsylvania Courts
717-795-2000, barb.holmes@pacourts.us

February 13, 2008

Technology Experience Bulletin, TEB: 2008-02

In developing a statewide criminal Common Pleas Case Management System (CPCMS) for Pennsylvania, development staff sought to provide many of the functions in existing case management systems throughout the state. The legacy Philadelphia CMS employed a managed, randomized Judicial Assignment function. After reviewing this judicial assignment function, the Administrative Office of Pennsylvania Courts (AOPC) sought to develop an improved function within the client/server architecture of CPCMS.

The Philadelphia mainframe based function provided the ability to assign percentages to Judges within a given case type. This allowed trial court administrators to increase or decrease the percentage allocation by Judge in order to balance case load without removing the element of randomness. However, if the percentage allocated to a particular Judge was increased or decreased, users had to manually calculate and adjust the percentage of all other Judges in the group. The current CPCMS implementation offers an automated reallocation feature. It provides for the grouping of cases into differentiated case management groups known as Event Tracks. The system has the capability to group event tracks into Judicial Assignment groupings so that assignments can be randomized across tracks.

A review of available literature on randomization in computing indicates that the "randomize" functions provided with most programming languages suffer from inherent problems of predictability and repeatability that make them ineffective for all but the most casual uses. With the comparatively small sample of Judges in most courts, the ineffective nature of this randomization was particularly noticeable. CPCMS development staff then performed additional research to determine how to remedy this problem and provide for a more robust implementation of a randomization function appropriate to the importance of judicial assignments.

The final implementation uses the existing random number function within Sybase to provide just the seed value for a java implementation of the "Mersenne Twister" algorithm for random number generation. It was first formally presented in 1998 (M. Matsumoto and T. Nishimura, Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Trans. on Modeling and Computer Simulations, 1998.) and is generally accepted as one of the best choices available for non-cryptographic applications. Features (adapted from "Dept. Math. Hiroshima-Univ, Homepage of Makato Matsumoto"):

- It was designed with consideration on the flaws of various existing generators.
- The algorithm is coded into a C-source downloadable that is widely available.
- Far longer period and far higher order of equidistribution than any other implemented generators. (It is proved that the period is 2^19937-1, and 623-dimensional equidistribution property is assured.)
- Fast generation. (Although it depends on the system, it is reported that MT is sometimes faster than the standard

ANSI-C library in a system with pipeline and cache memory.) (Note added in 2004/3: on 1998, usually MT was much faster than rand(), but the algorithm for rand() has been substituted, and now there are no much difference in speed.)

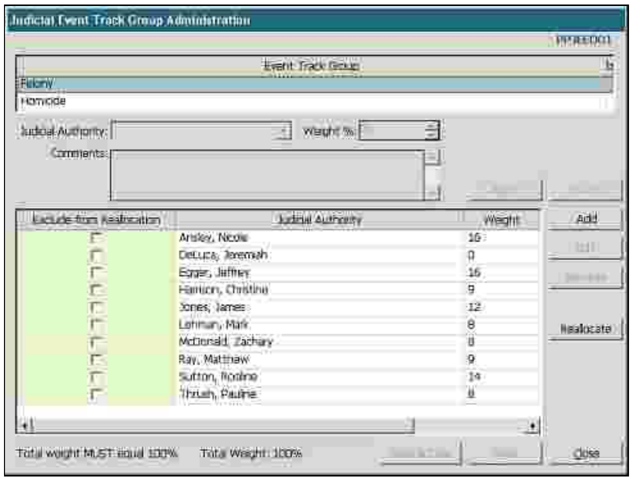
- Efficient use of memory. (The implemented C-code mt19937.c consumes only 624 words of working area.)

Court Context

The Administrative Office of Pennsylvania Courts Judicial Automation department has been responsible for development of case management systems for Magisterial District Justices, the Court of Common Pleas Criminal Division, and the Pennsylvania Appellate Courts (Supreme, Superior, and Commonwealth).

Philadelphia is the largest county using the CPCMS with over 3,000 users in the First Judicial District. The system, overall, supports approximately 7,000 users via the client/server application while many more use the Secure Web applications to obtain access to criminal case management data.

The Judicial Assignment Process

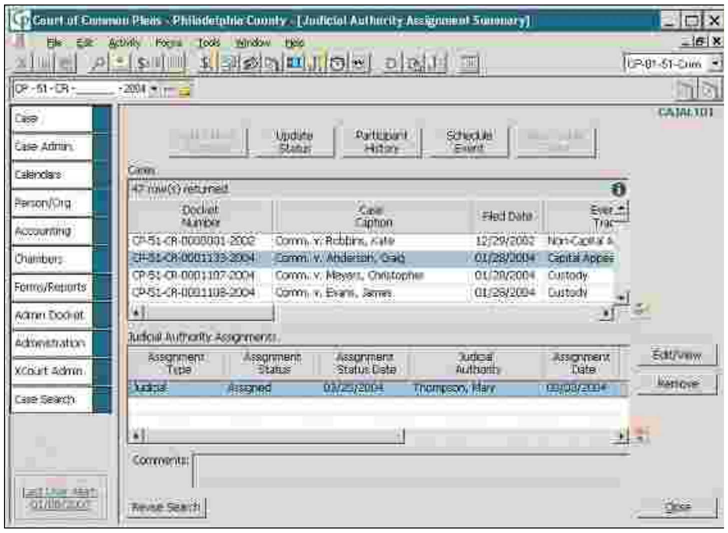


Event Track Group
Felony
Homicide

Exclude from Reallocation	Judicial Authority	Weight
	Ansley, Nicole	16
	DeLuca, Jeremiah	0
	Eggar, Jeffrey	16
	Harrison, Christina	9
	Jones, James	12
	Lehman, Mark	8
	McDonald, Zachary	8
	Ray, Matthew	9
	Sutton, Rosline	14
	Thrush, Pauline	8

Total weight MUST equal 100%   Total Weight: 100%

Figure 1 This screen allows for a weighted allocation of assignments to control the volume of cases a Judge receives while still providing for randomization of assignment. The Reallocate button automatically recalculates percentages when the weighting is changed for single or multiple Judges. Excluding those Judges from Reallocation allows their weighting to stay the same, while redistributing any needed change proportionally among the other Judges.



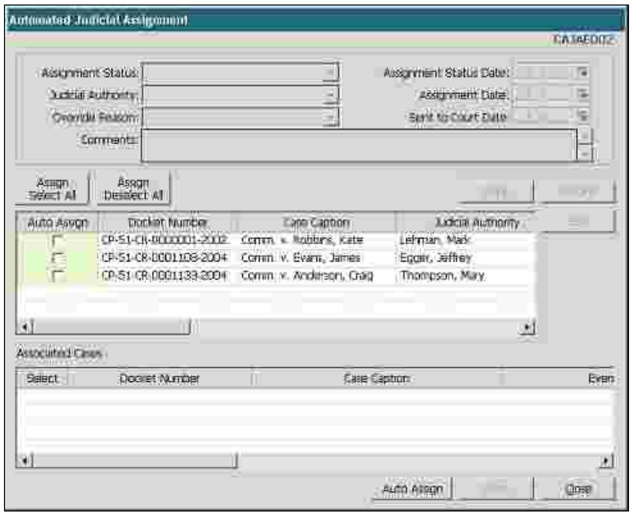
Cases
47 row(s) returned

Docket Number	Case Caption	Filed Date	Event Trac
CP-51-CR-0000001-2002	Comm. v. Robbins, Kate	12/29/2002	Non-Capital A
CP-51-CR-0001133-2004	Comm. v. Anderson, Craig	01/28/2004	Capital Appea
CP-51-CR-0001107-2004	Comm. v. Meyers, Christopher	01/28/2004	Custody
CP-51-CR-0001108-2004	Comm. v. Evans, James	01/28/2004	Custody

Judicial Authority Assignments

Assignment Type	Assignment Status	Assignment Status Date	Judicial Authority	Assignment Date
Judicial	Assigned	03/25/2004	Thompson, Mary	03/03/2004

Figure 2 A base screen shows all those cases awaiting Judicial Assignment. Note that the event tracks on this base screen vary. The event track determines the pool of Judges who may hear this case based on the Event Track Group (for example, all Major Felonies or Homicides).



Auto Assign	Docket Number	Case Caption	Judicial Authority
	CP-51-CR-0000001-2002	Comm. v. Robbins, Kate	Lehman, Mark
	CP-51-CR-0001108-2004	Comm. v. Evans, James	Eggar, Jeffrey
	CP-51-CR-0001133-2004	Comm. v. Anderson, Craig	Thompson, Mary

Associated Cases

Select	Docket Number	Case Caption	Even

Figure 3 This screen brings up the selected cases and allows them to be auto-assigned in batch. If the assignment needs to be manually processed or modified, the user can also do this here. The user can also auto-assign and select other cases that the defendant might have for assignment to the same Judge.

Tips for Implementation

Tip 1: Gather and Validate User Requirements for the Implementation.
A specific user group may have different or additional requirements for your assignment implementation.
Two interesting requirements of the implementation in CPCMS are visible in the data model shown in Tip 2:

- One requirement was the ability to place a Judge “on hold” for assignments, for example, when he/she was ill for a long period. This is reflected in the JudicialAssignHoldDate Table. Later, the key of this table was changed to store a history of “on hold” periods for a specific Judge.
- A second requirement provided that the system retain a history of the automated assignments or “spins” that a Judge received, even if they were overwritten by the user. This allows validating the accuracy of the randomization. The JudicialAssignSpinHistory show in Tip 2 provides for this.

Tip 2: Define Access/Entry Paths to Increase Use-ability. The CPCMS design provided for accessing the assignment function CPCMS will provide two primary entry paths into the Judicial Assignment Process. These include the ability to create a Judicial Assignment while viewing a Case Calendar Event (Case Calendar Event Summary) based on the fact that many Judicial Assignments are made at formal arraignment or Judicial Conference and the ability to access this function from its own base screen.

Tip 3: Develop an Effective Data Model for the Implementation. The following table definitions are used to support the CPCMS implementation. Note the term “spin” was used based on the original mainframe implementation. Primary key columns are noted with an asterisk.

EventTrackJAGroup_ULKP

***EventTrackGroup**	**lookupvalue_short**	**not null,**
***CourtOffice**	**varchar(30)**	**t null,**
IsActive	bit	not null,
CreateDt	autodate	not null,
CreateUser	autousername	not null,
LastUpdateDt	autodate	not null,
LastUpdateUser	autousername	not null

Constraints:

(CourtOffice) references CourtOffice (CourtOffice)

JudicialAssignList_ULKP

***EventTrackGroup**	**lookupvalue_short**	**not null,**
***CourtOffice**	**varchar(30)**	**not null,**
***JudgeNm_ID**	**identifier**	**not null,**
Weight	tinyint	not null,
RangeStart	tinyint	not ull,
RangeEnd	tinyint	not null,
CommentTxt	varchar(500)	null,
CreateDt	autodate	not null,
CreateUser	autousername	not null,

LastUpdateDt	autodate	not null,
LastUpdateUser	autousername	not null

Constraints:

(EventTrackGroup, CourtOffice) references EventTrackJAGroup_ULKP (EventTrackGroup, CourtOffice)

(JudgeNm_ID) references ParticipantName (ParticipantNm_ID)

JudicialAssgnSpinStatus_LKP

***SpinStatus**	**lookupvalue_short**	**not null,**
IsActive	bit	not null,
CreateDt	autodate	not null,
CreateUser	autousername	not null,
LastUpdateDt	autodate	not null,
LastUpdateUser	autousername	not null

JudicialAssignSpinHistory

***EventTrackGroup**	**lookupvalue_short**	**not null,**
***CourtOffice**	**varchar(30)**	**not null,**
***JudgeNm_ID**	**identifier**	**not null,**
***SpinDtTime**	**datetime**	**not null,**
SpinStatus	lookupvalue_short	not null,
Weight	tinyint	not null,
CreateDt	autodate	not null,
CreateUser	autousername	not null,
LastUpdateDt	autodate	not null,
LastUpdateUser	autousername	not null

Constraints:

(SpinStatus) references JudicialAssgnSpinStatus_LKP (SpinStatus)

(EventTrackGroup, CourtOffice, JudgeNm_ID) references JudicialAssignList_ULKP (EventTrackGroup, CourtOffice, JudgeNm_ID)

JudicialAssignHoldDate

***Participant_ID**	**identifier**	**not null,**
HoldStartDt	datetime	not null,
HoldEndDt	datetime	null,
CommentTxt	varchar(500)	null,
CreateDt	autodate	not null,

```
    CreateUser           autousername                    not null,
    LastUpdateDt         autodate                        not null,
    LastUpdateUser       autousername                    not null
Constraints:
(Participant_ID) references Participant (Participant_ID)


AssignmentOverrideReason_LKP
    *OverrideReason       lookupvalue_med                 not null,
    CourtOfficeSort       CourtOfficeSort                 not null,
    IsActive              bit                             not null,
    CreateDt              autodate                        not null,
    CreateUser            autousername                    not null,
    LastUpdateDt          autodate                        not null,
    LastUpdateUser        autousername                    not null
```

Tip 4: Carefully Select the Randomization Approach. As noted, a review of the available literature on randomization in computing indicates that the "randomize" functions provided with most programming languages suffer from inherent problems of predictability and repeatability that make them ineffective, especially in relatively small samplings.

Tip 5: Plan for Multiple Usages. Since this function is designed to be customized and used somewhat differently by various courts within the Common Pleas system, differences in use were considered in the design. Many courts use it with all percentages the same because they do not wish to weight the percentages, for example. This is especially true in courts where there are fewer Judges and less likelihood of an overwhelming caseload based on longer running cases. Also, for many smaller courts, all the event tracks are combined into a single Event Track Group.

Summary

Designing a managed, randomized judicial assignment function involves planning and the development of requirements as well as consideration of the reallocation and randomization techniques to be used.

Resource and Jurisdiction Contacts

Amy Ceraso, Director of Judicial Projects, Administrative Office of Pennsylvania Courts, 412-565-3013, amy.ceraso@pacourts.us

Ralph Hunsicker, Senior Projects Director, Administrative Office of Pennsylvania Courts, 717-795-2000, ralph.hunsicker@pacourts.us