CITOC
COURT INFORMATION TECHNOLOGY OFFICERS CONSORTIUM

Data for Case Flow Management Improvement

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

February 13, 2008

Technology Experience Bulletin, TEB: 2008-01

In Joint Application Development sessions, users from the Pennsylvania Appellate Court Case Management System (PACMS) project requested the inclusion of case flow management processes. As a result of this review, a Case Event process was built into the PACMS system. Subsequently in building the Common Pleas Criminal Case Management System (CPCMS), lessons learned from the PACMS project were taken into consideration in building a refined model. The resulting case flow management model is very flexible and robust and forms the basis for case management screens, reporting, and external message production.

Central to the concept of case flow management in PACMS and CPCMS is the concept of Event Tracks. Event Tracks are used by the courts for differential case management within the system and represent varied case lifecycles and/or scheduling requirements. Each Event Track may have a varied set of Case Event, Next Event combinations or sequences that track important milestones and move a case along through standardized Case Processing Statuses, mapping to statistical milestones. The Case Processing Statuses are associated to changes in overall Case Status.

CPCMS allows for customized docket entries for internal use by the court in recording actions on a case. These can be created automatically from many processes within the system, including scheduling and dispositions, among others. Docket entries can, in turn, create Case Event, Next Event combinations that move the case along in its lifecycle. User created docket entries may also have associated Case Event, Next Event Sequences and Case Processing and Case Status changes.

Court Context

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

Judicial Automation at the AOPC employs a full development staff. Throughout 15 years of developing software packages, beginning with the Magisterial District Justice (MDJ) System, AOPC has refined a repeatable process for software development, enhancement, and acquisition that involves an iterative lifecycle, joint application development (JAD) with user groups, screen prototyping, cross-disciplinary software development teams, and structured testing.

Both the PACMS and the CPCMS employ Event Tracks and their related Case Event, Next Event sequences to provide case management and tracking functionality.

Tips for Implementation

Tip 1: Establish an effective data model for case flow management. Figure 1 is a high level diagram of the CPCMS Case Event model.

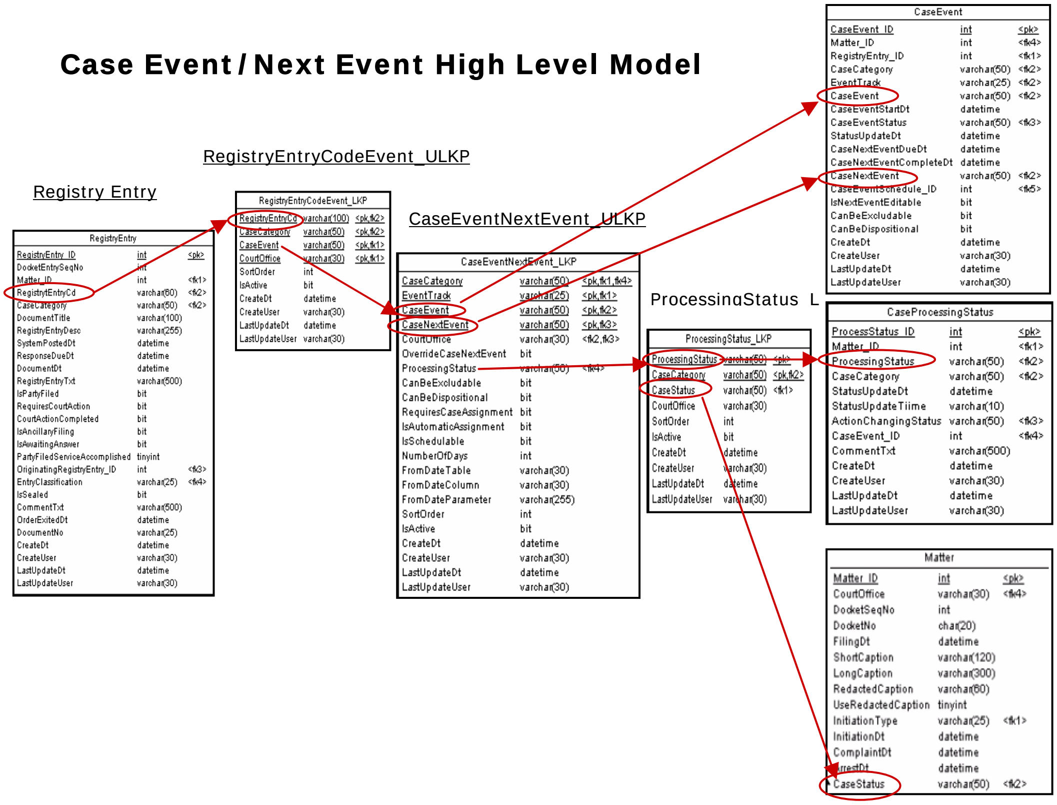
Tip 2: Provide for flexibility in completing case event/next event sequences. In our first implementation of this concept in PACMS, we put the structure in place to update the status of Case Event/Next Event sequences as "Failed" or "Completed." However, though the application provided for completing a particular Case Event/Next Event sequence based on the occurrence of the Next Event, we never fully implemented the status changing functionality. In CPCMS, we implemented the ability of varied insertion of new Case Events to update the status of prior sequences.

In order to determine what should happen to Case Events, a table holding 'Status Changing Events' was implemented. Case Events that are inserted have a next event of 'Awaiting Completion,' unless the Next Event is 'No Scheduled Next Event.' In these events, the CaseEventStatus defaults to 'Completed.'

There is a table in the model called CaseEventNextEventStat_ULKP. These records indicate what should happen when a Status Changing Event occurs. In general, the Status Changing Event is the CaseNextEvent. For example, on the Guilty Plea Filed event, there is a Next Event of "Sentencing Scheduled." When the registry entry of "Sentencing Scheduled" is created, it inserts the event that completes any outstanding Next Events of "Sentencing Scheduled." Events can also be manually marked "Awaiting Completion" or "Completed" by end-users.

This implementation provides us with the capability, for example, for a status record of 'Case Dismissed' to mark any outstanding events as 'Failed' (see Figure 2).

Case Event / Next Event High Level Model



RegistryEntry		
RegistryEntry_ID	int	<pk>
DocketEntrySeqNo	int	
Matter_ID	int	<fk1>
RegistrytEntryCd	varchar(60)	<fk2>
CaseCategory	varchar(50)	<fk2>
DocumentTitle	varchar(100)	
RegistryEntryDesc	varchar(255)	
SystemPostedDt	datetime	
ResponseDueDt	datetime	
DocumentDt	datetime	
RegistryEntryTxt	varchar(500)	
IsPartyFiled	bit	
RequiresCourtAction	bit	
CourtActionCompleted	bit	
IsAncillaryFiling	bit	
IsAwaitingAnswer	bit	
PartyFiledServiceAccomplished	tinyint	
OriginatingRegistryEntry_ID	int	<fk3>
EntryClassification	varchar(25)	<fk4>
IsSealed	bit	
CommentTxt	varchar(500)	
OrderExitedDt	datetime	
DocumentNo	varchar(25)	
CreateDt	datetime	
CreateUser	varchar(30)	
LastUpdateDt	datetime	
LastUpdateUser	varchar(30)	

RegistryEntryCodeEvent_LKP		
RegistryEntryCd	varchar(100)	<pk,fk2>
CaseCategory	varchar(50)	<pk,fk2>
CaseEvent	varchar(50)	<pk,fk1>
CourtOffice	varchar(30)	<pk,fk1>
SortOrder	int	
IsActive	bit	
CreateDt	datetime	
CreateUser	varchar(30)	
LastUpdateDt	datetime	
LastUpdateUser	varchar(30)	

CaseEventNextEvent_LKP		
CaseCategory	varchar(50)	<pk,fk1,fk4>
EventTrack	varchar(25)	<pk,fk1>
CaseEvent	varchar(50)	<pk,fk2>
CaseNextEvent	varchar(50)	<pk,fk3>
CourtOffice	varchar(30)	<fk2,fk3>
OverrideCaseNextEvent	bit	
ProcessingStatus	varchar(50)	<fk4>
CanBeExcludable	bit	
CanBeDispositional	bit	
RequiresCaseAssignment	bit	
IsAutomaticAssignment	bit	
IsSchedulable	bit	
NumberOfDays	int	
FromDateTable	varchar(30)	
FromDateColumn	varchar(30)	
FromDateParameter	varchar(255)	
SortOrder	int	
IsActive	bit	
CreateDt	datetime	
CreateUser	varchar(30)	
LastUpdateDt	datetime	
LastUpdateUser	varchar(30)	

ProcessingStatus_LKP		
ProcessingStatus	varchar(50)	<pk>
CaseCategory	varchar(50)	<pk,fk2>
CaseStatus	varchar(50)	<fk1>
CourtOffice	varchar(30)	
SortOrder	int	
IsActive	bit	
CreateDt	datetime	
CreateUser	varchar(30)	
LastUpdateDt	datetime	
LastUpdateUser	varchar(30)	

CaseEvent		
CaseEvent_ID	int	<pk>
Matter_ID	int	<fk4>
RegistryEntry_ID	int	<fk1>
CaseCategory	varchar(50)	<fk2>
EventTrack	varchar(25)	<fk2>
CaseEvent	varchar(50)	<fk2>
CaseEventStartDt	datetime	
CaseEventStatus	varchar(50)	<fk3>
StatusUpdateDt	datetime	
CaseNextEventDueDt	datetime	
CaseNextEventCompleteDt	datetime	
CaseNextEvent	varchar(50)	<fk2>
CaseEventSchedule_ID	int	<fk5>
IsNextEventEditable	bit	
CanBeExcludable	bit	
CanBeDispositional	bit	
CreateDt	datetime	
CreateUser	varchar(30)	
LastUpdateDt	datetime	
LastUpdateUser	varchar(30)	

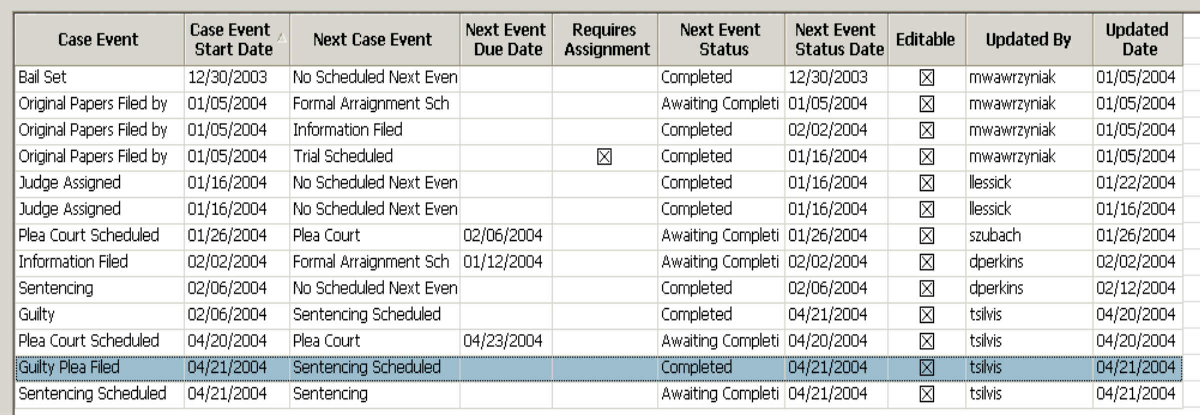
CaseProcessingStatus		
ProcessStatus_ID	int	<pk>
Matter_ID	int	<fk1>
ProcessingStatus	varchar(50)	<fk2>
CaseCategory	varchar(50)	<fk2>
StatusUpdateDt	datetime	
StatusUpdateTiime	varchar(10)	
ActionChangingStatus	varchar(50)	<fk3>
CaseEvent_ID	int	<fk4>
CommentTxt	varchar(500)	
CreateDt	datetime	
CreateUser	varchar(30)	
LastUpdateDt	datetime	
LastUpdateUser	varchar(30)	

Matter		
Matter_ID	int	<pk>
CourtOffice	varchar(30)	<fk4>
DocketSeqNo	int	
DocketNo	char(20)	
FilingDt	datetime	
ShortCaption	varchar(120)	
LongCaption	varchar(300)	
RedactedCaption	varchar(60)	
UseRedactedCaption	tinyint	
InitiationType	varchar(25)	<fk1>
InitiationDt	datetime	
ComplaintDt	datetime	
ArrestDt	datetime	
CaseStatus	varchar(50)	<fk2>

Figure 1. A high level diagram of the CPCMS Case Event model. Whenever a Registry Entry is created, the application looks in RegistryEntryCodeEvent_ULKP to determine if any CaseEventNextEvent Records need to be inserted. If the CaseEventNextEvent has a Processing Status, the application inserts a row in CaseProcessingStatus and updates Matter with this Process Status ID. The application then takes the Case Status of the Processing Status and updates the Matter table with this new (or maybe same) Case Status.

Tip 3: Provide for variation by Case Type, Category, and Track. Because cases of different types have different case life cycles, your data model must allow for varied case event/next event sequences by category or type. Both PACMS and CPCMS further vary the sequences based on Event Track. This provides differences in life cycle for Accelerated Rehabilitative Disposition or Plea Tracks, for example.

Tip 4: Carefully review the Docket/Registry, Case Event/Next Event Sequences, and their related Case Processing and Case Status Changes. Events need to be triggered by some action of the user, either performing a function in the system or through creating a docket entry that is significant in the case's life cycle and has been determined to move the case along the event track. This is accomplished by associating events to registry entries. Registry entries that advance or divert the case on the Event Track will create events on the case. Based on the Event Track, due dates can be calculated for events that need to be tracked. It is important for the proper relationships to be established in the data for the functionality to work properly (see Figure 3).



Case Event	Case Event Start Date	Next Case Event	Next Event Due Date	Requires Assignment	Next Event Status	Next Event Status Date	Editable	Updated By	Updated Date
Bail Set	12/30/2003	No Scheduled Next Even			Completed	12/30/2003	☒	mwawrzyniak	01/05/2004
Original Papers Filed by	01/05/2004	Formal Arraignment Sch			Awaiting Completi	01/05/2004	☒	mwawrzyniak	01/05/2004
Original Papers Filed by	01/05/2004	Information Filed			Completed	02/02/2004	☒	mwawrzyniak	01/05/2004
Original Papers Filed by	01/05/2004	Trial Scheduled		☒	Completed	01/16/2004	☒	mwawrzyniak	01/05/2004
Judge Assigned	01/16/2004	No Scheduled Next Even			Completed	01/16/2004	☒	llessick	01/22/2004
Judge Assigned	01/16/2004	No Scheduled Next Even			Completed	01/16/2004	☒	llessick	01/16/2004
Plea Court Scheduled	01/26/2004	Plea Court	02/06/2004		Awaiting Completi	01/26/2004	☒	szubach	01/26/2004
Information Filed	02/02/2004	Formal Arraignment Sch	01/12/2004		Awaiting Completi	02/02/2004	☒	dperkins	02/02/2004
Sentencing	02/06/2004	No Scheduled Next Even			Completed	02/06/2004	☒	dperkins	02/12/2004
Guilty	02/06/2004	Sentencing Scheduled			Completed	04/21/2004	☒	tsilvis	04/20/2004
Plea Court Scheduled	04/20/2004	Plea Court	04/23/2004		Awaiting Completi	04/20/2004	☒	tsilvis	04/20/2004
Guilty Plea Filed	04/21/2004	Sentencing Scheduled			Completed	04/21/2004	☒	tsilvis	04/21/2004
Sentencing Scheduled	04/21/2004	Sentencing			Awaiting Completi	04/21/2004	☒	tsilvis	04/21/2004

Figure 2 Case Event/Next Event Sequences and their status.

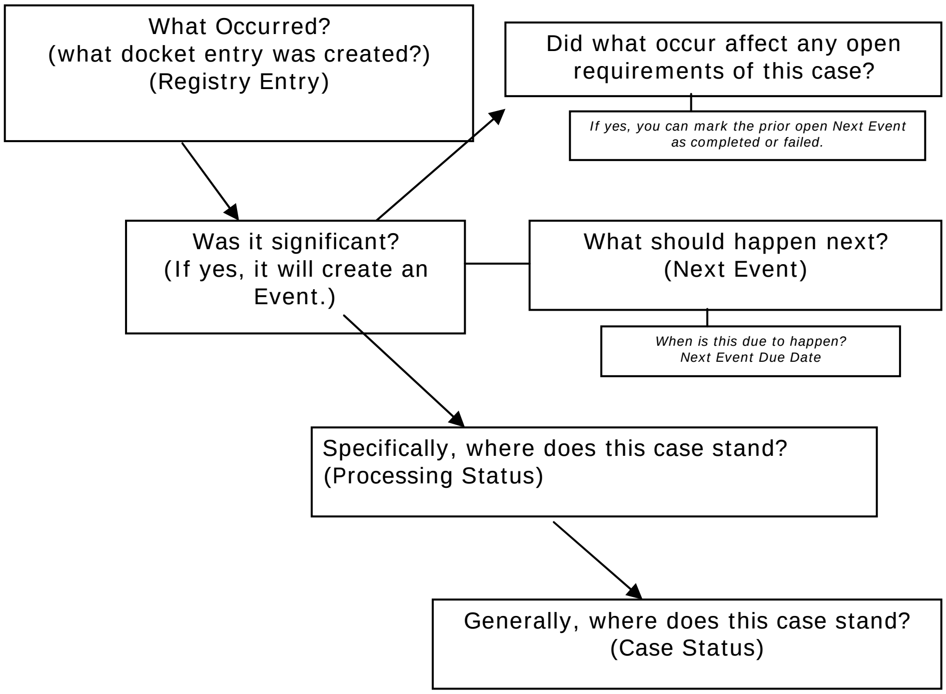

What Occurred?
(what docket entry was created?)
(Registry Entry)
Did what occur affect any open requirements of this case?
If yes, you can mark the prior open Next Event as completed or failed.
Was it significant?
(If yes, it will create an Event.)
What should happen next?
(Next Event)
When is this due to happen?
Next Event Due Date
Specifically, where does this case stand?
(Processing Status)
Generally, where does this case stand?
(Case Status)

Figure 3 Work flow functionality.

Summary

The Case Event/Next Event model used in CPCMS provides for prompting users to schedule, assign Judges and perform other case management activities as well as serving as a basis for reporting and for the creation of Case Event related external messages, such as our Case Initiation and Warrant messages. This model is extremely flexible and in many instances can be customized quickly to avoid additional coding. Because Case Events are more generic than Registry Entries, they can speed up reporting. They provide the basis for recording major case milestones, known as Case Processing Statuses, which in turn provide for changing overall Case Status.

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

Disclaimer: The advice and opinions represented in this bulletin are based on the experiences of the Administrative Office of Pennsylvania Courts (AOPC). Such recommendations may not be suitable for other jurisdictions, and are only offered in the spirit of sharing experience as information to others considering the installation of similar technologies and adoption of similar processes.