Case Management Modelling and Notation - A how to guide on the new OMG standard

Mike A. Marin & Prof. Mark von Rosing

1 Introduction

Case handling has been discussed in the literature since 2001, when van der Aalst and Berens described its unique requirements[1,2,3,]4,5. Case handling aims at supporting knowledge workers with a more flexible process execution by avoiding well known restrictions present in conventional BPM and workflow technology. Examples of such restrictions include rigid control flow, context tunneling and restricted data visibility. The central concept behind case management modeling is an information model useful to react to data conditions and data state change events. BPMN 2.0 doesn't have it, and without it, conditional events are not as usable, because the only data visible to them are process variables and properties. The information model includes both data and documents, so changes in values, metadata, and lifecycle state can all be used to model the case. Case handling is also about empowering participants in a process, by removing context tunneling and providing better support for exception handling[3]. Case handling was considered a difficult use case for process and workflow technology[4]; and among others, Swenson proposed Adaptive Case Management[5] to address the requirements described by van der Aalst *et al.*4. In this 'how to guide', we use case handling and case management interchangeable as they represent the same use case.

Before arriving to the case handling use case described by van der Aalst and Berens1 in 2001, other researchers have worked on increasing the flexibility of workflow and process technology. For example, Hull *et al.*[6] focused on declarative processes in 1998, Kappel *et al.*[7] focused on the interception between rules and workflow in 1998. Oba and Komoda[8] in 2001 worked on dynamically controlled processes for enterprise application integration.

In 2009, the Object Management Group started the process of standardizing a case management notation to complement its BPMN specification[9]. The result is the Case Management Modeling and

[1] W.M.P. Van Der Aalst and P.J.S. Berens, "Beyond Workflow Management: Product-Driven Case Handling," *Proceedings of the 2001 International ACM SIGGROUP*, New York: ACM Press, 2001, pp. 42–51.

[2] P. Athena. Case Handling with FLOWer: Beyond, Workflow. 2002.

[3] H.A. Reijers, J. Rigter, and W.M.P. Van Der Aalst, "The Case Handling Case," *International Journal of Cooperative Information Systems*, vol. 12, 2003, pp. 365–391.

[4] W.M.P. Van Der Aalst, M. Weske, and D. Grunbauer, "Case Handling: A New Paradigm for Business Process Support," *Data and Knowledge Engineering*, vol. 53, 2005, pp. 129–162.

[5] K.D. Swenson, *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*, Meghan-Kiffer Press, 2010.

[6] Hull, R., Llirbat, F., Simon, E., Su, J., Dong, G., Kumar, B., & Zhou, G. (1999). Declarative Workflows that Support Easy Modification and Dynamic Browsing. Language. *Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration*, WACC 1999.

[7] Kappel, G., Rausch-Schott, S., and Retschitzegger, W.Coordination in workflow management systems – A rule-based approach Coordination Technology for Collaborative Applications. *Springer Berlin Heidelberg*, 1998, 1364, 99-119.

[8] M. Oba and N. komoda. Multiple Type Workflow Model for Enterprise Application Integration. HICSS, 2001.

[9] OMG, *Case Management Process Modeling (CMPM) Request for Proposal*, Needham, MA: Object Management Group, 2009.

Notation (CMMN) specification[10]. This new specification looks at case management from a data centric perspective based on business artifacts[11], which is a line of research that originates with Niggam and Caswell[12] in 2003 and focus on the artifacts that knowledge workers use to accomplish their work, which in most cases are documents. This 'how to guide' introduces CMMN, describe how it works, where it can be applied, and provides an example.

2 Case Management

Case management is not about making processes more flexible; it is about empowering the workers by providing them with access to all the information about the case and giving them the discretion to plan the case execution, and decide which tasks to execute for a particular case instance. In a BPM system the process designer encode the business goal to be accomplished in the model of the process; therefore the workers executing activities in the process do not need to be aware of the business goal. In a case management system, in the other hand, the case worker is in control of the business goal. In BPM, the system is responsible to accomplish the business goal and uses workers to achieve that goal. In a case management system, the workers are responsible for the business goal and use the system as a tool to accomplish that goal. Case management relies more in the worker's judgment than in control flow.

On its introduction to case handling, Aalst *et al.*[4] describe four problems of the workflow approach that also applies to BPM. These problems describe how the workflow system constraint the workers, and relegate data to a secondary concern. The four problems are,

- The work is *straight-jacketed into activities*. Everything that does work in a process is modeled by an activity restricting the workers flexibility.
- R*outing is used for both distribution and authorization*; therefore workers can only access the work and data they are assigned to do and nothing else.
- Introduce *context tunneling* by focusing on the control flow and relegating data to a secondary concern.
- Focus is on what *should* be done instead of what *can* be done, creating inflexible processes.

Case management and in particular CMMN solve those problems by formalizing the concept of a case file that contains case data. Data is more important than control flow, and therefore the state of a case dependent more on the case data state than on the control flow state1. In most instances, case data is represented by documents, because workers commonly interact by using documents like spreadsheets, presentations, word processor documents, voice recordings, videos, pictures, etc. Therefore, in most situations the case data will be documents that are used by the workers to accomplish their business goal. In CMMN case data can be mapped to documents and folders via the Content Management Interoperability Services (CMIS)[13]. Although, CMIS is not required for a CMMN implementation, the use of a content management system is useful to support knowledge workers.

The focus on knowledge workers1[,4] is an important component of a case management system. The ability to add, modify, or remove data from a case file at any time during the process is a key feature of

[10] OMG, *Case Management Model and Notation (CMMN), version 1.0,* Needham, MA: Object Management Group, 2014.

[11] M. Marin, R. Hull, and R. Vaculín, "Data Centric BPM and the Emerging Case Management Standard: A Short Survey," *Business Process Management Workshops*, M. Rosa and P. Soffer, eds., Springer Berlin Heidelberg, 2013, pp. 24–30.

[12] A. Nigam, and N.S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal* 42(3), 428-445, 2003.

[13] OASIS. *Content Management Interoperability Services (CMIS),.OASIS 2012*.

a case management system. This introduces the need to provide collaboration technology like groupware as part of the case management runtime. Although CMMN does not prescribe the runtime environment, collaboration technology is useful to support knowledge workers.

The four problems described by Aalst *et al.*4 are addressed in CMMN as follows:

- Not all the work that happens in a case is modeled. In particular the interaction of workers with the case file and case data may not be modeled, or be partially modeled. In most situations, we don't model how the data gets into the case. This solves the *straight-jacketed into activities* problem of BPM. Case data can be added, removed, modified by the case workers at any time during the processing of a case without the need of an activity.
- Work distribution is independent from authorization; therefore case workers can be authorized to access the case file and its data even when they don't have work assigned. Case workers can search for cases they are assigned to, and they can interact with the data contained in that case file. This solves the *routing is used for both distribution and authorization* problem, and introduces the requirement for collaboration technology for case workers to collaborate and interact with case instances.
- Case data, and not control flow, is the focus of a case management system. Case workers have always access to the case file and all the case data, solving the *context tunneling* problem. The interaction of case workers with case data may trigger additional tasks or activities in the case.
- By allowing runtime planning and manually activation of tasks, case management focuses on *what can be done*. Tasks are enabled based on the case data available rather than on the tasks that have already been executed. Workers could collaborate via case data like comments and notes.

As described1[,4] case management is a paradigm shift from control flow to a data and worker centric paradigm. The expectation that workers will interact directly with the case file and case data reduces the need to fully model the process. Workers will be involved in case planning, will make decisions on what tasks should be executed, and will be able to add, remove, or modify case data for a particular case instance. Providing this level of flexibility and support to knowledge workers requires a combination of technologies, including BPM, collaboration, and content management technologies.

2.1 Case Management Modeling and Notation

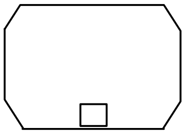
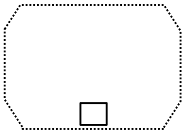
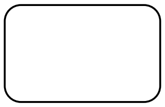
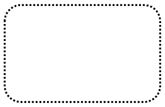
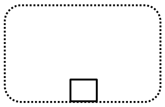
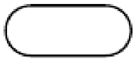
A case, in CMMN, is composed of an information model, a behavioral model, and a set of roles that operates on both models. The case file describes the information model, which corresponds to the business artifacts being handled by the case, and the case plan describes the behavioral model. Case management is inherently human centric, and so, not everything in the case needs to be modeled; only those aspects that must be enforced or that can help the human to advance the case must be modeled. During the execution of a case instance, the users can add, modify, or delete data from the case file; can modify the case execution by planning or deciding on optional activities, etc. The result is that both the information and the behavior of a case can vary greatly between two instances. The ability for workers to adjust the case behavior by doing planning is an important component of CMMN. Case planning is handled by providing planning tables containing discretionary items that can be added to the instance execution plan at runtime by the workers. The end result is that workers can add new tasks to a case instance at their own discretion.

CMMN is based on the Guard-Stage-Milestone (GSM) approach for declarative artifact lifecycle defined by Hull *et al.*[14]. Guards in CMMN are a combination of an event and a condition, where both are optional, but one must exist. They are used to define entry and exit criterion for tasks, stages, and milestones. *Guards* are modeled as a diamond (◊) decorator representing entry criteria, *stages* in GSM can be equated to CMMN stages and tasks. GSM *milestones* are modeled as milestones in CMMN.

Event-Condition-Action (ECA) rules can also be mapped into CMMN, which is not surprised because the GSM semantics is based on ECA rules[15]. The *event* is modeled using an optional connector; the *condition* by the entry criteria using the diamond (◊) decorator; and the *action* can be a stage, a task, or a milestone to which the entry criteria is attached to.

CMMN models can invoke BPM processes via process tasks, and for that reason an implementation may include integration with a BPM system. Those processes, could be process fragments to help the worker achieve a particular outcome, or fully developed processes the organization has implemented. The CMMN notation is designed to be compatible and complementary to BPMN, because the OMG expects vendors to implement the two specifications in the same tool set.

CMMN introduces the concept of discretionary elements, which are modeled but are not included in the execution plan of a case instance. The only way a discretionary item will be executed in a case instance is if a worker adds it to the case plan. Adding discretionary elements to the case plan is called planning. Planned elements are executed as soon as the entry criterion is met, unless it requires manual activation. Table 1, shows the main CMMN notational elements, and Table 2 shows the decorators in the notation.

Planned	Discretionary	Description
		*Stage*. Container of stages, tasks, plan fragments, milestones, and events. Allows decomposing the model into manageable sets. Represent the stages in GSM.
		*Task*. Represents the execution of actual work. Similar to BPMN tasks or activities. In a pure GMS model, a task will be considered the same as a stage.
		*Plan fragment*. Provides a grouping mechanism for discretionary items.
		*Milestone*. Indicate an accomplishment during the process. Correspond to milestones in GSM.

[14] R. Hull, E. Damaggio, F. Fournier, M. Gupta, I. Fenno(Terry) Heath, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P. Sukaviriya, and R. Vaculin, "Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles," *Web Services and Formal Methods*, M. Bravetti and T. Bultan, eds., Springer Berlin Heidelberg, 2011, pp. 1–24.

[15] R. Hull, E. Damaggio, R.D. Masellis, F. Fournier, M. Gupta, F.T. Heath III, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P.N. Sukaviriya, R. Vaculın, and R. De Masellis, "Business Artifacts with Guard-Stage-Milestone Lifecycles: Managing Artifact Interactions with Conditions and Events," *Proceedings of the 5th ACM international conference on Distributed event-based system*, New York, NY, USA: ACM, 2011, pp. 51–62.

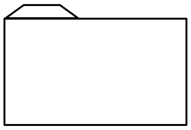
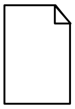
◎	*Event*. Similar to BPMN events.
	*Case model*. The complete behavioral model of the case being modeled is described inside the case model.
	*Case file item*. Represent the items in the information model, and normally stored in the case file.
- - - - - - - - - - - - -	*Connector*. Connectors are optional, and are used in two situations. First, to visualize the event portion of an entry or exit criterion, in which case the connector ends on the entry or exit criterion decorator. Second, to visualize the discretionary items associated with a human task planning table, in which case the connector connects a human task with a planning table to discretionary tasks, stages, or plan fragments.

Table 1 – CMMN Notational elements

Decorator	**Description**
⊞	*Planning table*. Present when planning is allowed in the case model, stage, or human task. For planning to be allowed, there must be discretionary stages, tasks, or plan fragments in the scope of the planning table.
◊	*Entry criterion*. Present when the stage, task, or milestone contains a guard that must be triggered for the stage, task, or milestone to be enabled. Represent the guards in GSM, and the event-condition in an ECA rule.
♦	*Exit criterion*. Present when the stage, task, or case model contains a guard that will force termination of the stage, task, or case model if triggered.
■	*Auto complete*. Indicates the stage or case model will complete when all the required stages, tasks, and milestones inside are completed. If the decorator is not present, the stage or case model requires manual completion after all the required stages, tasks, and milestones inside are completed.
▷	*Manual activation*. Indicates that the stage or task must be manually initiated after the entry criterion has been satisfied. If the decorator is not present, the stage or task will automatically start executing when the entry criterion is satisfied.
!	*Required*. Indicates that the stage, task, or milestone must be executed for the scope (stage or case model) to complete.
#	*Repetition*. Indicates the stage, task, or milestone can be repeated multiple times.

Decorator	Description
⊞	*Collapsed*. Similar to BPMN. Applicable to stages, planning tables, and plan fragments.
⊟	*Expanded*. Similar to BPMN. Applicable to stages, planning tables, and plan fragments.
👤	*Blocking human task*. Indicates a task that is executed by a human, and the engine must wait until the human signalizes the task has been completed.
🖐	*Non-blocking human task*. Indicate a task that is handled to a human and the engine should not wait to consider it complete.
🗀	*Case task*. Indicate a task that is implemented by invoking another case.
▷	*Process task*. Indicates a task that is implemented and modeled using a process notation, like BPMN.

Table 2 – CMMN Decorators

Case management looks at a process from the perspective of the knowledge workers, with the goal of enabling them to efficiently collaborate to achieve a business goal. CMMN achieves that by allowing runtime planning of cases, having the concept of manually activated tasks and stages, and reacting to creation, update, and delete of case data. Case planning itself can be modeled in advance, by indicating tasks and stages that allow planning with a planning table. However, any worker in a role that allows planning can do planning at any moment during the case instance execution. Planning is based on the concept of discretionary tasks and stages that are modeled to be used at the discretion of the workers working a case instance. Manually activated tasks and stages are those for which the entry criterion has been met, but they are only executed if a worker decides to do so. Case data can be both structured like a properties in the case file, and unstructured like documents and folders. Workers in the case have access to the case data, and assuming enough privileges, they can add, modify, or delete data. Changes to the case data can trigger entry criterion in tasks, stages, or milestones.

Although, CMMN is a modeling notation and only describe the runtime behavior of the modeled elements, it is clear that an effective implementation should provide a role based collaboration environment for workers to interact with the case and its data. Case data can be documents that in this context refer to any type of digital asset, including pictures, video, voice recordings, spreadsheets, presentations, etc. Implementations will probably include or interact with a content management system. The integration with collaboration and content management technology means that not everything that happens in a case needs to be modeled, because workers can affect the case in multiple ways via the collaboration and content platforms.

Figure 1 shows the minimum possible case model. In CMMN, roles and case data are defined but not modeled. Therefore, the minimum case model in Figure 1 may have a set of roles and case data. The case runtime may provide collaboration capability for the workers to collaborate between them, and to interact with the case instance. Workers can add, update, and remove data from the case instance. The workers could also add comments for other workers as part of the case data. Assuming the case data is implemented via documents, this minimum case model in Figure 1 looks very similar to a team space in a content management system or in a collaboration system. As a result, the minimum case model in Figure 1 can be useful for workers to solve some simple cases.

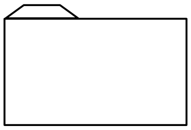


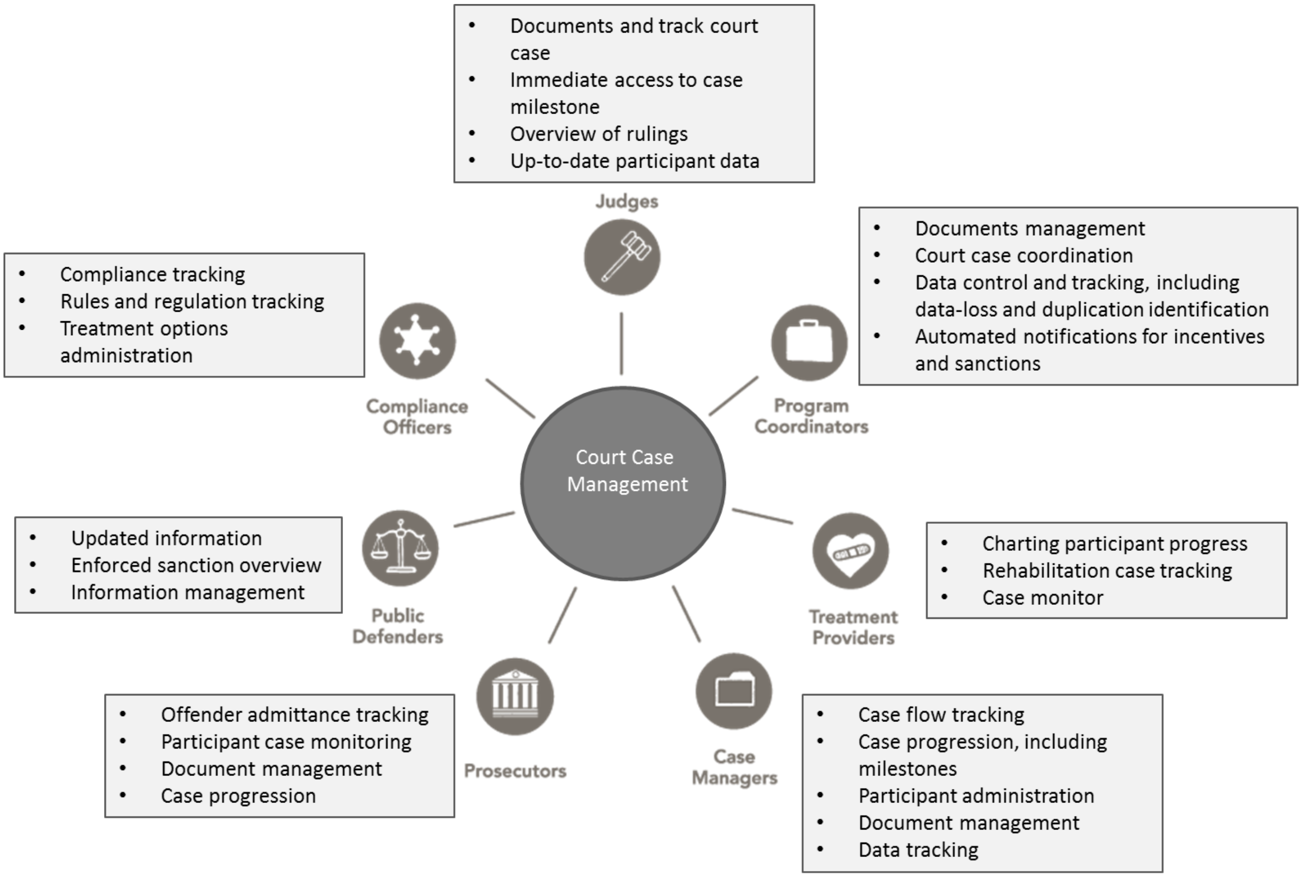
Figure 1. Minimun case model

2.2 Example

In this section, we illustrate how CMMN supports case handling around the civil procedure of a court case. By choosing an industry with a high degree of complexity, we wish to illustrate the capabilities of CMMN. We have chosen a court environment because the users are specialized knowledge workers, their workflows are complex, and the document management capabilities are highly sophisticated.

2.2.1 The court system

Case in a court environment is about managing information and documents around multiple roles in a complex workflow from Judges, Program Coordinators, Case Managers, Treatment Providers, Prosecutors, Compliance Officers, Public Defenders and many others. In the criminal system, they have to interact with multiple external defendants. For example, prisoners, illegal immigrants, people under hospital order, etc. What makes court case management interesting is that each worker needs real-time access to the full case information, and that information can easily change as new evidence is filed, evidence is deemed inadmissible, milestones are reached, notifications are generated, etc. Figure 2 provides an overview of a criminal system roles and their specific case handling needs[16]. The same complexity and similar requirements are present in the civil system.



Documents and track court case
Immediate access to case milestone
Overview of rulings
Up-to-date participant data
Judges
Compliance tracking
Rules and regulation tracking
Treatment options administration
Documents management
Court case coordination
Data control and tracking, including data-loss and duplication identification
Automated notifications for incentives and sanctions
Compliance Officers
Program Coordinators
Court Case Management
Updated information
Enforced sanction overview
Information management
Charting participant progress
Rehabilitation case tracking
Case monitor
Public Defenders
Treatment Providers
Offender admittance tracking
Participant case monitoring
Document management
Case progression
Prosecutors
Case Managers
Case flow tracking
Case progression, including milestones
Participant administration
Document management
Data tracking

Figure 2 Typical roles, their requirements, and needs around a court case management system[16].

In a court case management system each worker plays a unique role, requires a unique and flexible workflow, and has unique information requirement. Each worker needs real-time access to the relevant information in the context of a case. The system must handle content management requirements like email notifications, document storage, retention policies, and versioning. It must also handle the collaboration requirements of court workers. It must integrate information and workflow of external parties like drug labs, assessment tools, other agencies and justice partners. The process must react to changes in the case data.

The content management requirements include handling the storage and retention policies of the legislation, the court cases and the verdicts. Case management in a court system could be very complex as it needs to coordinate a large set of parties. Figure 3 is an example of a layered architecture view of a civil system.

[16] LEADing Practice Justice Standards LEAD ID#-IS60015. From http://www.leadingpractice.com/

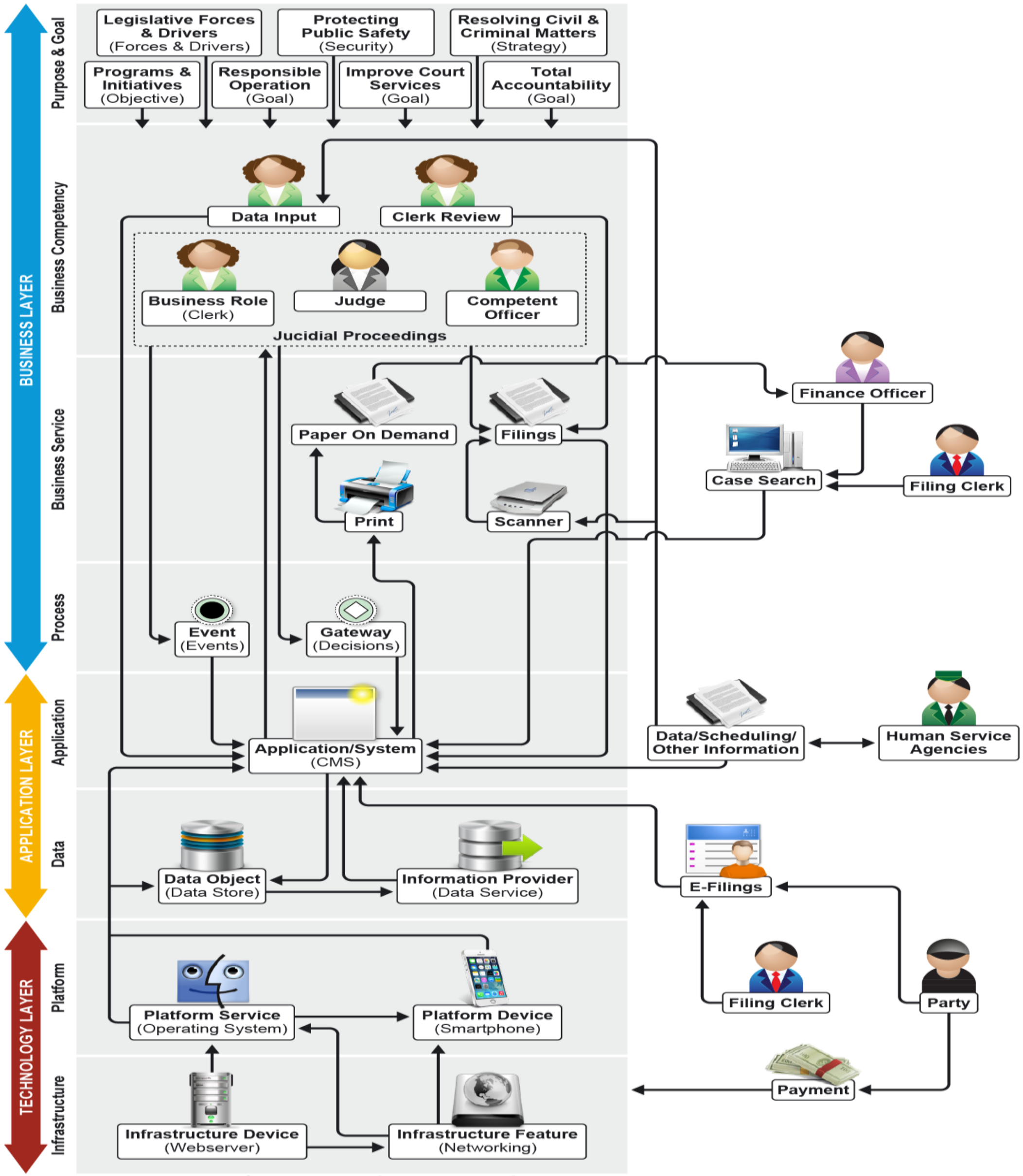

BUSINESS LAYER
APPLICATION LAYER
TECHNOLOGY LAYER
Purpose & Goal
Business Competency
Business Service
Process
Application
Data
Platform
Infrastructure
Legislative Forces & Drivers (Forces & Drivers)
Protecting Public Safety (Security)
Resolving Civil & Criminal Matters (Strategy)
Programs & Initiatives (Objective)
Responsible Operation (Goal)
Improve Court Services (Goal)
Total Accountability (Goal)
Data Input
Clerk Review
Business Role (Clerk)
Judge
Competent Officer
Jucidial Proceedings
Finance Officer
Paper On Demand
Filings
Case Search
Filing Clerk
Print
Scanner
Event (Events)
Gateway (Decisions)
Application/System (CMS)
Data/Scheduling/ Other Information
Human Service Agencies
Data Object (Data Store)
Information Provider (Data Service)
E-Filings
Filing Clerk
Party
Platform Service (Operating System)
Platform Device (Smartphone)
Payment
Infrastructure Device (Webserver)
Infrastructure Feature (Networking)
(©)LEADing Practice Justice Reference Content [#LEAD-IS60015]

Figure 3 Layered architecture view illustrating court case handling[17].

Figure 3 shows that multiple roles are involved; numerous tasks, information aspects and stages must be coordinated. This makes the modeling of case handling difficult to do in BPM. Collaboration technology and content management technology can help in this scenario and there is no need to model them. A case management model in a court environment therefore has to include the ability for:

- **Roles** including role specification, documents they create the tasks they execute, and the stages they can complete.
- **Information definition** including the case file items, which could be documents. The role that

[17] LEADing Practice Justice Standards LEAD ID#-IS60015. From http://www.leadingpractice.com/

produces the document, the task in which the document is produced.

- **Milestone specification** including the description of the milestones as well as the condition that should be meets for the milestone to be achieved.
- **Stage specification** including describing the stages, and when they are executed.
- **Tasks specification** describes the tasks and when they are executed
- **Case Model** describes the semantics, with all roles involved, information, tasks and the stages collapsed.

2.2.2 The court plaint receipt example

In this example, we will use the *plaint receipt process* used by the Central Intellectual Property and International Trade Court in Thailand. According to Rungruangpattana, and Achalakul[18,19,20], the plaint receipt process fits well the case management paradigm. In this section, we extracts the main modeling elements, while keeping it at a high level by omitting some details from the original descriptions by Rungruangpattana and Achalaku[l]18[,19,20].

The plaint receipt process uses most CMMN constructs with the exception of plan fragments, stages with plan tables, and exit criteria for tasks. Plan tables are used in tasks only. Those constructs were not required to model the process.

As a convention to simplify referencing the different notational elements, each notational element label includes a short indicator at the end of the label enclosed in parenthesis. In summary,

- Case file items are numbered from I1 to I8
- Stages are numbered from S1 to S5
- Discretionary stages are numbered from dS1 to dS3
- Tasks are numbered from T1 to T13
- There is a single discretionary task numbered as dT1
- There is a single event numbered as E1
- Milestones are numbered from M1 to M3

2.2.3 Plaint receipt

The *plaint receipt process* starts by the litigant party filing a plaint. The plaint is reviewed by a competent officer that classifies the case and inspects it. The competent officer may request the party to amend the plaint, in which case, the party is notified by the filing clerk that the plaint needs to be amended. In that situation, the party may submit an amended plaint. This part of the process may happen multiple times, and the party may submit amended plaints at any time, even if it is not

[18] P. Rungruangpattana and T. Achalakul, “The Software Prototype of Civil Court Case Management in Thailand,” *International Journal of Software Engineering and Its Applications*, vol. 3, Jul. 2009, pp. 45–58.

[19] P. Rungruangpattana and T. Achalakul, “The Software Prototype of Civil Court Case Management in Thailand,” *Proceedings of the 2008 Advanced Software Engineering and Its Applications*, Washington, DC, USA: IEEE Computer Society, 2008, pp. 221–225.

[20] P. Rungruangpattana and T. Achalakul, “The design framework of the civil court case management system in Thailand,” *TENCON 2007-2007 IEEE Region 10 Conference*, IEEE, 2007, pp. 1–4.

requested by the filing clerk.

When the plaint is accepted, a competent officer charges the court fees and the party provides the payment to the finance officer that issues a receipt of court fees. After payment has been received, a competent officer assigns a Judge to the case. The Judge review the case and issue an order, which could be to accept the case, reject the case, or return the case to the party for more information.

If the case is accepted by the Judge, witnesses are summoned and interviewed by the judge, which also collect and review the evidence. There may be multiple witnesses and evidence to be reviewed by the Judge. With that information the judge issues a sentence. If the case is rejected, the judge records that fact in the judgment list.

If the case is returned to the party for more information, the litigant party is notified by the filing clerk to amend the plaint. The party may submit an amended plaint that is reviewed by the competent officer. If the party does not amend the claim in the allocated timeframe, the competent officer writes a report to the Judge indicating the amended plaint was not received in time. In either of those two situations, the judge is again requested to issue another order, which again could be to accept the case, reject the case, or return the case to the party for more information.

2.2.4 Roles

In CMMN, roles are defined at the case level. Roles can be used as performers for human tasks, to do case planning, and to raise user events. However, roles do not have a graphical notation, and not everything a role does is modeled. In particular, a role may be allowed to do planning that may not be explicitly modeled; a role may also add, create, modify, or remove documents from the case file which may not be explicitly modeled. The plaint receipt process prototype described by Rungruangpattana, and Achalakul uses six roles. For simplification, the Authority role was omitted. Table 3, describes the remaining roles, the documents they create, the tasks they execute, and the stages they can complete.

Role	Creates	Execute tasks	Complete stages
Competent officer	Accepted Amended Plaint (I4) Report on Plaint (I5)	Classify the case (T1) Inspect the plaint (T2), including planning Asses plaint docs to charge court fee (T3) inform litigant to amend the plaint (dT1) Assign a judge (T5) Accept the plaint (T12) Do report to the judge (T13)	Inspect plaint (S1) (S4)
Filing clerk		Inform litigant to amend the plaint (T11) Inform litigant to amend the plaint (dT1)	Prototype of Civil Court Case Management in Thailand
Finance	Receipt (I8)	Issue a receipt of court fee (T4)	

Role	Creates	Execute tasks	Complete stages
officer			
Judge	Sentence (I7)	Manually schedule Process plaint (S3) Order (T6), including planning Examine witnesses (T7) Review evidence (T8) Sentence (T9) Sentence (T10)	Process plaint (S3) (S5)
Party	Plaint (I1) Amended plaint (I2) Payment (I3) Amend Plaint (I6)		

Table 3 – Roles

2.2.5 Information model

Table 4, describes the different case file items. In this example, the case file items are documents. The table describes the role that produces the document, and the task in which the document is produced. Note that some documents originate outside the case management system; and for those documents produced within a task in the system, there is no explicit modeling of that fact.

Case file item	Produced by	Produced at task
Plaint (I1)	Party	
Amended plaint (I2)	Party	
Payment (I3)	Party	
Accepted Amended Plaint (I4)	Competent officer	Inspect the plaint (T2) Accept the plaint (T12)
Report on Plaint (I5)	Competent officer	Do report to the judge (T13)
Amend Plaint (I6)	Party	
Sentence (I7)	Judge	Sentence (T9) Sentence (T10)

Case file item	Produced by	Produced at task
Receipt (I8)	Finance officer	Issue a receipt of court fee (T4)

Table 4 – Information model

2.2.6 Milestones

Milestones represent accomplishments during the process of the case. Due to the large variations between case instances, milestones are important to understand the progress of a particular case instance. Table 5, describes the milestones and the condition that should be meet for the milestone to be achieved.

Milestone	Entry criteria	Description
Inspected plaint (M1)	Completion of stage S1	Triggered when a plaint has been accepted
Judge assigned (M2)	Completion of task T5	Triggered when a Judge has been assigned to the case
Case decided (M3)	Creation of I7	Triggered when the case have been decided and a sentence has been produced

Table 5 – Milestones

2.2.7 Stages

Stages are behavioral containers, and they are used to manage the complexity of the model by using nesting to hide details. Table 6, describes the stages in this example, and when they are executed (entry criteria).

Stage	Entry criteria	Description
Inspect plaint (S1)	Arrival of a plaint (I1)	Required stage, but not auto-complete; because T1, T2, and dT1 may complete before an I2 arrives
Collect payment (S2)	Completion of Inspected plaint (M1)	Required and auto-complete stage
Process plaint (S3)	Completion of Judge Assigned (M2)	Required and manually activated stage
(S4)	None. Starts when dS1 transitions to active	Required stage, but not auto-complete; because T7 add T8 can be executed

Stage	Entry criteria	Description
		multiple times and T7 is optional
(S5)	None. Starts when dS3 transitions to active	This stage has an exit criteria that is triggered when T13 is completed
Accept (dS1)	None. Discretionary and may be added to the plan during the execution of T6.	Auto-complete discretionary stage
Reject (dS2)	None. Discretionary and may be added to the plan during the execution of T6.	Auto-complete discretionary stage
Return (dS3)	None. Discretionary and may be added to the plan during the execution of T6.	Auto-complete and repeatable discretionary stage. It is repeatable because the plaint (I1) could be returned multiple times to the party

Table 6 – Stages

2.2.8 Tasks

Tasks in CMMN are similar to tasks or activities in BPMN. Table 7, describes the tasks and when they are executed (entry criteria).

Task	Entry criteria	Description
Classify the case (T1)	None. Created when S1 becomes active	Requited, and blocking human task
Inspect the plaint (T2)	T1 completes OR a new I2 is created	Requited, repeatable, and blocking human task; because multiple I2s may be created. First time is expected to execute based on T1 completion, and if there is a problem with plaint the case worker can plan dT1 in which case a new I2 may be submitted by the party. Note that if the task is being executed and a new I2 is present, then I4 may be created by the competent officer during the execution of this task
Asses plaint docs to charge court fee (T3)	None. Created when S2 becomes active	Requited, and blocking human task

Task	Entry criteria	Description
Issue a receipt of court fee (T4)	T3 completes AND I3 is created	Required tasks implemented by a process
Assign a judge (T5)	T4 completes	Requited, blocking human task
Order (T6)	S3 becomes active OR a new I4 is created OR a new I5 is created	Required, repeatable, and blocking human task. This task is expected to execute the first time when a judge is assigned, and after that each time a new I4 or I5 are created
Examine witnesses (T7)	None. Created when S4 becomes active	Repeatable, manually activated, and blocking human task
Review evidence (T8)	None. Created when S4 becomes active	Requited, repeatable, manually activated, and blocking human task
Sentence (T9)	S4 completes	Requited, and blocking human task. Note that I7 will be created by the judge during the execution of this task
Sentence (T10)	None. Created when dS2 becomes active	Requited, and blocking human task. Note that I7 will be created by the judge during the execution of this task
Inform litigant to amend the plaint (T11)	None. Created when S5 becomes active	Requited, and non-blocking human task
Accept the plaint (T12)	New I6 is created	Optional and blocking human task. Note that I4 will be created by the competent officer during the execution of this task
Do report to the Judge (T13)	Timer E3 completes	Optional and blocking human task. Note that I5 will be created by the competent officer during the execution of this task
Inform litigant to amend the plaint (dT1)	None. Discretionary task	Non-blocking human task

Table 7 – Tasks

2.2.9 The plaint receipt process CMMN model

The following CMMN model describes the plaint receipt process. This section presents the model and informally describes the semantics. Figure 4, provides the case model with all the stages collapsed. The arrival of the Plaint (I1) document starts the case. The case is created and the Plaint (I1) document added to the case file. That can be done manually, although some tools, like IBM Case Management[21], can be configured to automatically create the case when a document arrives to a content repository via any means including email. The arrival of Plaint (I1) to the case file triggers the entry criteria of the Inspect plaint (S1) stage that start execution. For the case to complete, Inspect plaint (S1), Collect payment (S2), and Process plaint (S3) must execute, indicated by the exclamation sign (!).

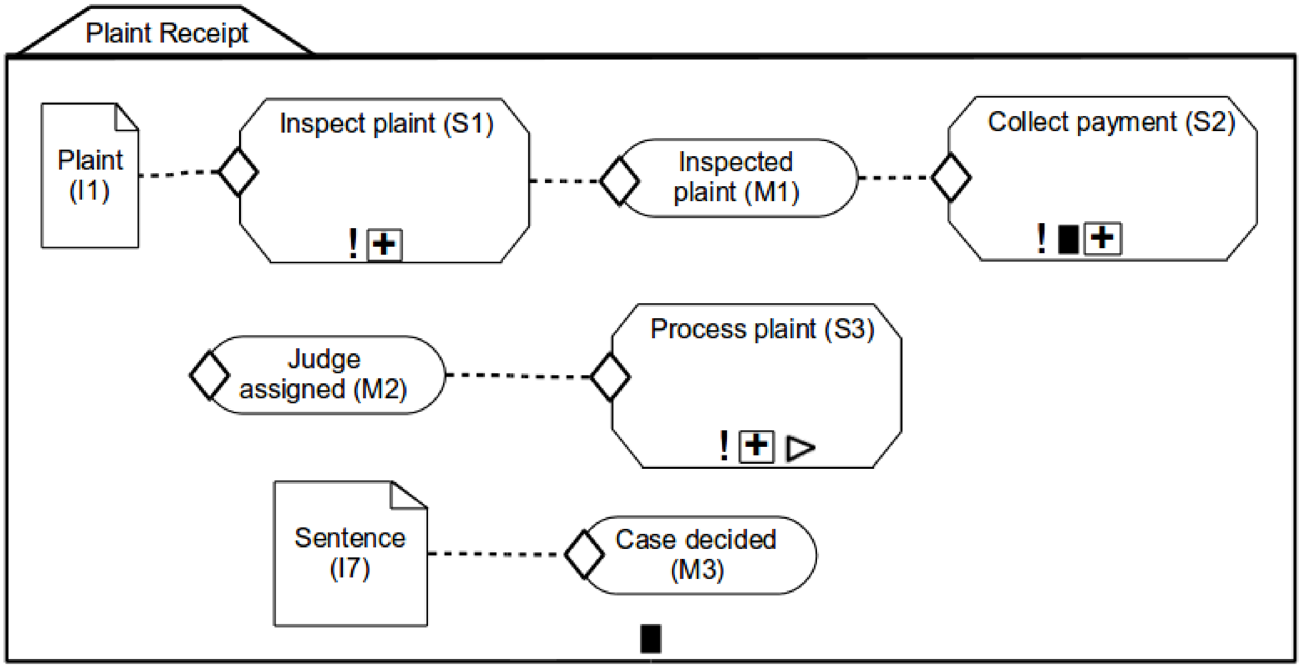


Figure 4 – Plaint receipt process

Note the connector (dotted line) between Plaint (I1) and the entry criterion in Inspect plaint (S1). That connector indicates that a state change in Plaint (I1) generates an event that is expected in the entry criterion of Inspect plaint (S1). In this case, the entry criterion is expecting a document of type Plaint (I1) to be created. There may also be a condition in the entry criterion to Inspect plaint (S1), for example the type of Plaint (I1) document. The connector between for Inspect plaint (S1) and milestone Inspected plaint (M1) indicates that a state change inside the Inspect plaint (S1) is being expected in the milestone entry criterion. By looking at Figure 4, we don't know which CMMN element inside S1 may generate the event expected by the milestone entry criterion. A connector between a task or stage and an entry or exit criteria does not means the completion of the task or stage will provide the expected event. By the description in Table 5, we know the expected event in M1 is completion of S1. Connectors are optional in CMMN, and could be omitted from the model.

[21] W.-D. Zhu, B. Benoit, B. Jackson, J. Liu, M. Marin, S. Meena, J.F. Ospina, and G. Rios, *Advanced Case Management with IBM Case Manager*, IBM Redbooks, 2014.

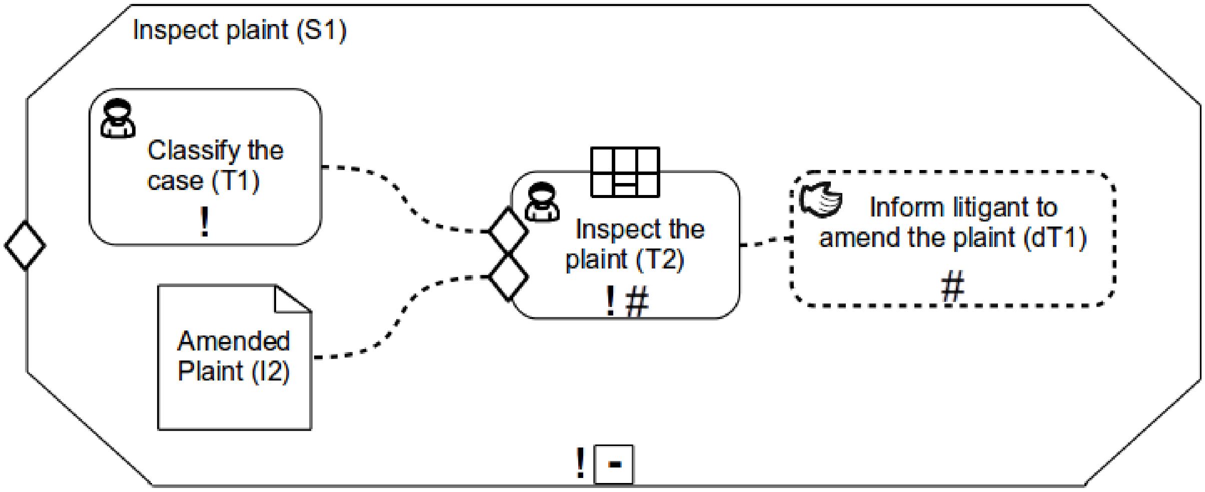


Figure 5 – Expanding stage Inspect plaint (S1)

In Figure 5, Classify the case (T1) becomes available for execution when the stage Inspect plaint (S1) starts executing. Inspect the plaint (T2) is repeatable, and become available for execution when Classify the case (T1) completes or Amend Plaint (I2) arrives to the case. T2 is repeatable, because multiple I2s may arrive. Inform litigant to amend the plaint (dT1) is a discretionary non-blocking human task, that may be used by the competent officer if needed.

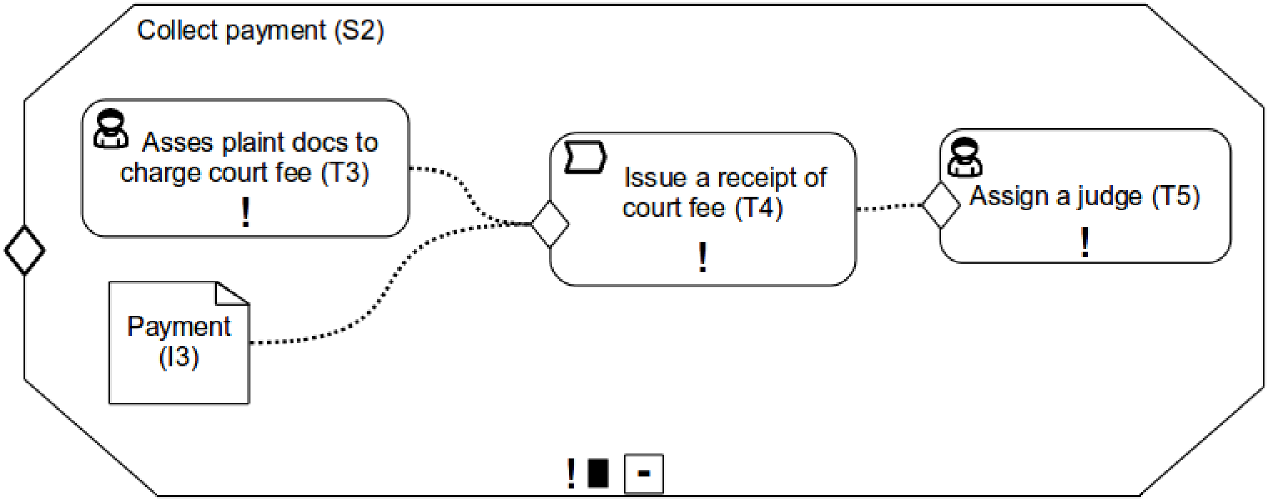

Collect payment (S2)
Asses plaint docs to charge court fee (T3)
!
Payment (I3)
Issue a receipt of court fee (T4)
!
Assign a judge (T5)
!
! ■ -

Figure 6 – Expanding stage Collect payment (S2)

As modeled in Figure 6, the entry criterion of Issue a receipt of court fee (T4) requires an event from both a Payment (I3) and the Asses plaint docs to charge court fee (T3). Collect payment (S2) is required for the case to complete and will automatically complete when T3, T4, and T5 complete.

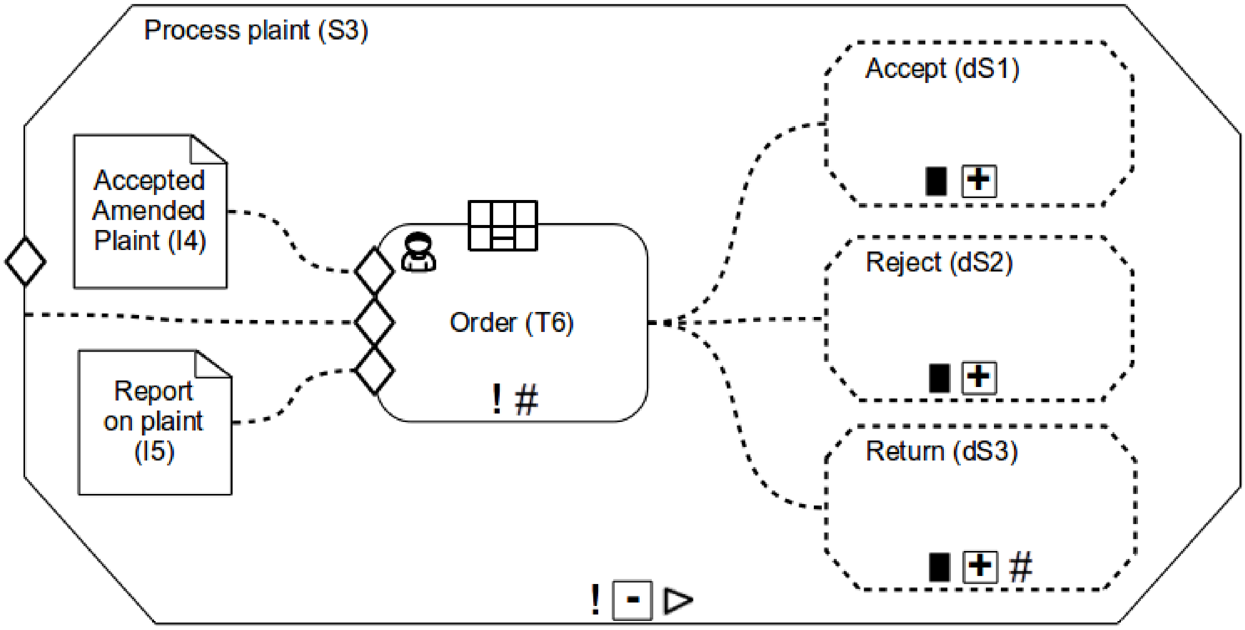


Figure 7 – Process plaint (S3)

Process plaint (S3) becomes available when a Judge is assigned as modeled in Figure 4, and it is manually activated by the Judge. As modeled in Figure 7, Order (T6) is the main task. T6 is required and can be executed multiple times. As described in Table 7, T6 initially start execution when S3 start executing; and then T6 executes again every time an Accepted Amended plaint (I4) or a Report on plaint (I5) arrives to the case. The Judge doing T6 can select one of the discretionary stages associated with T6 planning table by doing planning.

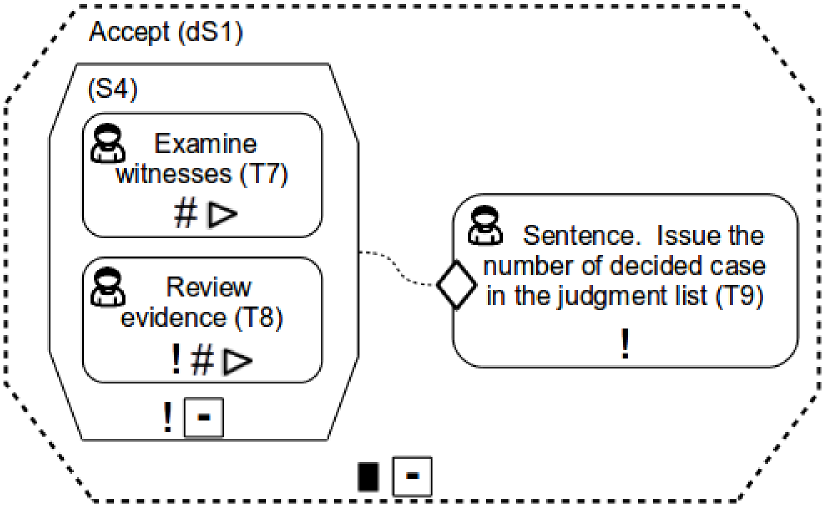

Accept (dS1)
(S4)
Examine witnesses (T7)
#
Review evidence (T8)
! #
!
Sentence. Issue the number of decided case in the judgment list (T9)
!

Figure 8 – Expanding discretionary stage Accept (dS1)

The Accept (dS1) discretionary stage modeled in Figure 8 starts by instantiating a required stage (S4). S4 contains two repeatable blocking human tasks. Examine witness (T7) is optional, and if used, it must be activated manually. Review evidence (T8) is required, but still must be activated manually. Because, both T7 and T8 are repeatable, then S4 was modeled as requiring manual completion. This avoids premature completion of S4. For example, let say that T8 is completed, and so, no required tasks are remaining in S4 that condition under auto complete will signalize to the case management system to complete S4; however in this case more witnesses may be scheduled to be examined (T7) in the future, and so, we don't want S4 to auto-complete.

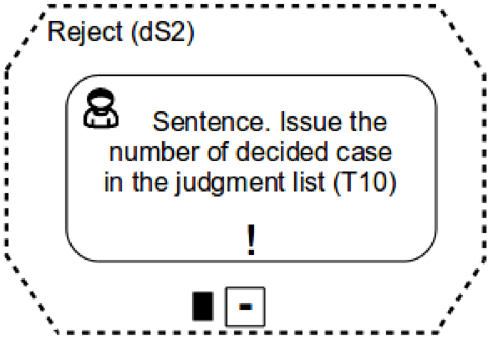

Reject (dS2)
Sentence. Issue the number of decided case in the judgment list (T10)
!

Figure 9 – Expanding discretionary stage Reject (dS2)

The Reject (dS2) discretionary stage modeled in Figure 9 contains a single task. The Sentence, issue the number of decided case in the judgment list (T10) task is required and after completion, Reject (dS2) will automatically complete.

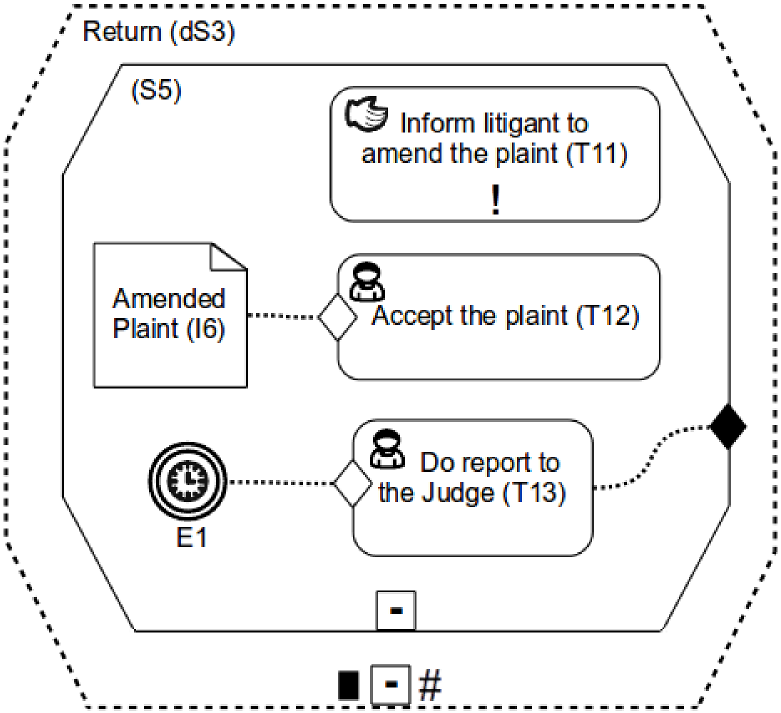

Return (dS3)
(S5)
Inform litigant to amend the plaint (T11)
!
Amended Plaint (I6)
Accept the plaint (T12)
E1
Do report to the Judge (T13)
#

Figure 10 – Expanding discretionary stage Return (dS3)

The Return (dS3) discretionary stage modeled in Figure 10 consists of a single stage (S5). S5 was required, because a discretionary stage cannot have an exit criterion. Note that Return (dS3) is repeatable, and so, it can be planned multiple times. The contained stage (S5) has an automatic non-blocking human task to inform the litigant to amend the plaint (T11). When an Amended plaint (I6) arrives, it is processes by accept the plaint (T12) task. Note the amended plain (I6) is the same case file item as I2. If the timer E1 expires, then do reports to the Judge (T13) may be executed, and that may trigger the exit criterion of S5. If the exit criterion is met then all the activity inside S5 will forced to terminate and S5 will complete.

3 Conclusion

This ‘how to guide’ described the main characteristics of *Case Management* and how it can be used to solve the *case handling* requirements. We introduced the CMMN specification, and described its relationship to BPM, collaboration, and content management technology. In particular, we described the differences between case management and BPM, because CMMN is not intended to replace BPMN but it is complementary to BPMN. CMMN is expected to be used in conjunction with BPMN.

We used the plaint receipt process followed by the Central Intellectual Property and International Trade Court in Thailand to illustrate the CMMN modeling capabilities. The plaint receipt process was introduced in the literature as a case management use case by Rungruangpattana, and Achalakul[18,19,20]. In the example, we presented the case modeling notation and how it forces a structured way of thinking, working and modeling; which is different than traditional BPM. Traditional workflow and BPM technology requires that all aspects of the problem be modeled. CMMN, in the other hand, rely on the worker’s judgment and collaboration technology to minimize modeling and make the process more flexible and data centric.