The Six CMMN Models used in the Online Survey

Mike A. Marin

February 16, 2017

1 Introduction

The CMMN models used in the experiment described in Chapter 8 (Empirical Validation of Case Management Metrics) of the main thesis were generated using the following procedure,

1. Table 1 was used to decide how many of each CMMN elements and annotators will be included in the resulting model.

2. Three samples CMMN models from the literature were used to establish a base line. *Sample A* is the the plaint receipt process used by the Central Intellectual Property and International Trade Court in Thailand from Marin and von Rosing [Mv p]. *Sample B* is the Write Document example from the CMMN Beta specification [OMG14a]. *Sample C* is the claims file management example from the CMMN 1.0 specification [OMG14b].

3. Mapped the three samples to Table 1 and calculated an *average model* (see Table 2).

4. Using the *average model* calculated a maximum value for each row that will be used to generate the CMMN models. The maximum was calculated by multiplying the elements of the *average model* by five. Five was selected as an arbitrary small number. The minimum was set to zero for all the different icons, with exception of the case plan model element that is required for a valid CMMN model.

5. The *average model* has a total number of icons *CTS* equal to 89 (see Table 2), so it was decided that *CTS* will remain constant at 90 for all the generated models. This forced the number of icons in the model to be constant at 90, and removed the total number of icons from being a factor on the complexity of the models.

6. Based on TableTable Table 2, a set of variables and the rules to generate them were defined in Table 3. The variables included the size, length and complexity variables.

7. A set of consistency constraints were developed as described in Table 4.

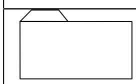
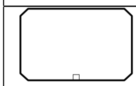
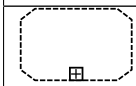
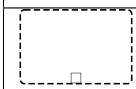
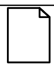
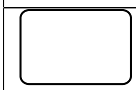
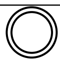
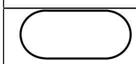
8. The constraints in Table 4 and the variables and rules in Table 3 were translated into a constraint model. MiniZic, a constraint programming modeling language [Net+07], was used to implement the constraint model.

9. Six solutions to the MiniZic model were used to describe the type and number of icons to create the six CMMN models. Each of the six models correspond to a solution to the set of constraints.

10. Minor manual adjustments were done to the generated models to adjust for variation on the type of icons used.

11. Finally, the resulting CMMN models were drawn using Inkscape.

Table 1: CMMN weights

$\mathcal{E} \cup \mathcal{A}$	**Description**	**Weight**
	case element	1
	stage element	1
	discretionary stage element	2
	plan fragment element	3
	case file item element	1
	task element	1
	discretionary task element	2
	event listener element	2
	milestone element	1
. . . .	connector (sentry) element	0
	collapsed planning table	1
	expanded planning table	2
■	autocomplete	2
⊞	collapsed	0

⊟	expanded	1
▷	manual activation	1
#	repetition	1
!	required	1
◊	entry criterion with associated connector	1
	entry criterion without a connector	2
♦	exit criterion with associated connector	1
	exit criterion without a connector	3
	non-blocking human	1
	process	0
	case referring to a case element not in this model	0
	case referring to a case element in this model	1
	participant	0
	timer	0

2 CMMN Samples

Table 2: Sample CMMN models

CMMN symbol or construct		Sample A	Sample B	Sample C	⌈Average⌉	Min	Max
	case plan model element	1	1	1	1	1	5
	stage element	5	1	3	3	0	15
	discretionary stage element	3	1	0	2	0	10
	plan fragment element	0	0	0	0	0	$0 \to 1$
	case file item element	7	1	0	3	0	15
	task element	13	2	2	6	0	30
	discretionary task element	1	8	5	5	0	25
	event listener element	0	0	0	0	0	$0 \to 1$
	user event listener element	0	0	2	1	0	5
	timer event listener element	1	1	0	1	0	5
	milestone element	3	2	3	3	0	15
	stage collapsed planning table	0	0	3	1	0	5
	stage expanded planning table	0	3	0	1	0	5
	human task collapsed planning table	0	0	1	1	0	5
	human task expanded planning table	2	0	0	1	0	5
■	autocomplete	5	2	0	3	0	15
⊞	collapsed	6	0	0	2	0	10
⊟	expanded	2	2	3	3	0	15
▷	manual activation	3	0	0	1	0	5
#	repetition	6	2	4	4	0	20
!	required	12	1	2	5	0	25
······◇	entry criterion with associated connector	15	8	6	10	0	50
◇	entry criterion without a connector	1	0	1	1	0	5
······◆	exit criterion with associated connector	1	3	2	2	0	10
◆	exit criterion without a connector	1	1	0	1	0	5
	non-blocking human marker	1	8	1	4	0	20
	process marker	1	0	3	2	0	10
	case marker (case not in this model)	0	0	1	1	0	5
	case marker (with case in this model)	0	0	0	0	0	$0 \to 1$
	blocking human marker	11	2	2	5	0	25
· · · · · · · ·	extra connector (sentry) element(2)	4	1	0	2	0	10
	TOTALS	105	50	45	75		375
	CS	54	29	24	39		195
	CAS	68	33	31	50		250
	CTS	122	62	55	89		445
	CL	5	3	3	-		-
	CC	93	62	47	77		385

3 Variables and constraints

Table 3: Variable and metrics calculation

Variable	Description	Calculation
$CS_{SC}[m]$	Number of cases	$min(CS_{SC}) \leq CS_{SC}[m] \leftarrow random() \leq max(CS_{SC})$
$CS_{SS}[m]$	Number of stages	$min(CS_{SS}) \leq CS_{SS}[m] \leftarrow random() \leq max(CS_{SS})$
$CS_{SDS}[m]$	Number of discretionary stages	$min(CS_{SDS}) \leq CS_{SDS}[m] \leftarrow random() \leq max(CS_{SDS})$
$CS_{SPF}[m]$	Number of plan fragments	$min(CS_{SPF}) \leq CS_{SPF}[m] \leftarrow random() \leq max(CS_{SPF})$
$CS_{DI}[m]$	Number of case file items	$min(CS_{DI}) \leq CS_{DI}[m] \leftarrow random() \leq max(CS_{DI})$
$PT\&MH[m]$	non-blocking human task	$min(PT\&MH) \leq PT\&MH[m] \leftarrow random() \leq max(PT\&MH)$
$PT\&MP[m]$	process task	$min(PT\&MP) \leq PT\&MP[m] \leftarrow random() \leq max(PT\&MP)$
$PT\&MC[m]$	case task referring to a case not in this model	$min(PT\&MC) \leq PT\&MC[m] \leftarrow random() \leq max(PT\&MC)$
$PT\&MC\&C[m]$	case task referring to a case in this model	$min(PT\&MC\&C) \leq PT\&MC\&C[m] \leftarrow random() \leq max(PT\&MC\&C)$
$PT\&MHB[m]$	blocking human task	$min(PT\&MHB) \leq PT\&MHB[m] \leftarrow random() \leq max(PT\&MHB)$
$CS_{PT}[m]$	Number of tasks	$PT\&MH[m] + PT\&MP[m] + PT\&MC[m] + PT\&MC\&C[m] + PT\&MHB[m]$
$PDT\&MH[m]$	discretionary non-blocking human task	$min(PDT\&MH) \leq PDT\&MH[m] \leftarrow random() \leq max(PDT\&MH)$
$PDT\&MP[m]$	discretionary process task	$min(PDT\&MP) \leq PDT\&MP[m] \leftarrow random() \leq max(PDT\&MP)$
$PDT\&MC[m]$	discretionary case task referring to a case not in this model	$min(PDT\&MC) \leq PDT\&MC[m] \leftarrow random() \leq max(PDT\&MC)$
$PDT\&MC\&C[m]$	discretionary case task referring to a case in this model	$min(PDT\&MC\&C) \leq PDT\&MC\&C[m] \leftarrow random() \leq max(PDT\&MC\&C)$
$PDT\&MHB[m]$	discretionary blocking human task	$min(PDT\&MHB) \leq PDT\&MHB[m] \leftarrow random() \leq max(PDT\&MHB)$
$CS_{PDT}[m]$	Number of discretionary tasks	$PDT\&MH[m] + PDT\&MP[m] + PDT\&MC[m] + PDT\&MC\&C[m]$ $+PDT\&MHB[m]$
$PE[m]$	plan event listener element	$min(PE) \leq PE[m] \leftarrow random() \leq max(PE)$
$PE\&MHB[m]$	user event listener element	$min(PE\&MHB) \leq PE\&MHB[m] \leftarrow random() \leq max(PE\&MHB)$
$PE\&MT[m]$	timer event listener element	$min(PE\&MT) \leq PE\&MT[m] \leftarrow random() \leq max(PE\&MT)$
$CS_{PE}[m]$	Number of event listeners	$PE[m] + PE\&MHB[m] + PE\&MT[m]$
$CS_{PM}[m]$	Number of milestones	$min(CS_{PM}) \leq CS_{PM}[m] \leftarrow random() \leq max(CS_{PM})$
$DCP_S[m]$	stage collapsed planning table decorator	$min(DCP_S) \leq DCP_S[m] \leftarrow random() \leq max(DCP_S)$
$DCP_H[m]$	human task collapsed planning table decorator	$min(DCP_H) \leq DCP_H[m] \leftarrow random() \leq max(DCP_H)$
$CAS_{DCP}[m]$	Number of collapsed planning table decorators	$DCP_S[m] + DCP_H[m]$
$DEP_S[m]$	stage expanded planning table decorator	$min(DEP_S) \leq DEP_S[m] \leftarrow random() \leq max(DEP_S)$
$DEP_H[m]$	human task expanded planning table decorator	$min(DEP_H) \leq DEP_H[m] \leftarrow random() \leq max(DEP_H)$
$CAS_{DEP}[m]$	Number of expanded planning table decorators	$DEP_S[m] + DEP_H[m]$
$CAS_{DAC}[m]$	Number of autocomplete decorators	$min(CAS_{DAC}) \leq CAS_{DAC}[m] \leftarrow random() \leq max(CAS_{DAC})$
$CAS_{DC}[m]$	Number of collapsed decorators	$min(CAS_{DC}) \leq CAS_{DC}[m] \leftarrow random() \leq max(CAS_{DC})$
$CAS_{DE}[m]$	Number of expanded decorators	$min(CAS_{DE}) \leq CAS_{DE}[m] \leftarrow random() \leq max(CAS_{DE})$
$CAS_{DMA}[m]$	Number of manual activation decorators	$min(CAS_{DMA}) \leq CAS_{DMA}[m] \leftarrow random() \leq max(CAS_{DMA})$
$CAS_{DRN}[m]$	Number of repetition decorators	$min(CAS_{DRN}) \leq CAS_{DRN}[m] \leftarrow random() \leq max(CAS_{DRN})$
$CAS_{DR}[m]$	Number of required decorators	$min(CAS_{DR}) \leq CAS_{DR}[m] \leftarrow random() \leq max(CAS_{DR})$
$SE\&OC[m]$	entry criterion sentry with associated connector	$min(SE\&OC) \leq SE\&OC[m] \leftarrow random() \leq max(SE\&OC)$
$SE[m]$	entry criterion sentry without a connector	$min(SE) \leq SE[m] \leftarrow random() \leq max(SE)$
$CAS_{SE}[m]$	Number of entry criteria sentries	$SE\&OC[m] + SE[m]$
$SX\&OC[m]$	exit criterion sentry with associated connector	$min(SX\&OC) \leq SX\&OC[m] \leftarrow random() \leq max(SX\&OC)$

$SX[m]$	exit criterion sentry without a connector	$min(SX) \leq SX[m] \leftarrow random() \leq max(SX)$
$CAS_{SX}[m]$	Number of exit criteria sentries	$SX\&OC[m] + SX[m]$
$OC[m]$	optional connector [sentry] element	$min(OC) \leq OC[m] \leftarrow random() \leq max(OC)$
$CS_{OC}[m]$	Number of connectors	$OC[m] + SE\&OC[m] + SX\&OC[m]$
$CAS_{MH}[m]$	Number of non-blocking human markers	$PT\&MH[m] + PDT\&MH[m]$
$CAS_{MP}[m]$	Number of process markers	$PT\&MP[m] + PDT\&MP[m]$
$MC_I[m]$	case marker (with case in m)	$PT\&MC\&C[m] + PDT\&MC\&C[m]$
$MC_O[m]$	case marker (with case not in m)	$PT\&MC[m] + PDT\&MC[m]$
$CAS_{MC}[m]$	Number of case markers	$MC_I[m] + MC_O[m]$
$CAS_{MHB}[m]$	Number of participant markers	$PT\&MHB[m] + PDT\&MHB[m] + PE\&MHB[m]$
$CAS_{MT}[m]$	Number of timer markers	$PE\&MT[m]$
$CC[m]$	CMMN Complexity	$CS_{SC}[m] + CS_{SS}[m] + (CS_{SDS}[m] \times 2) + (CS_{SPF}[m] \times 3) + CS_{DI}[m]$ $+CS_{PT}[m] + (CS_{PDT}[m] \times 2) + (PE[m] \times 2) + (PE\&MHB[m] \times 2)$ $+(PE\&MT[m] \times 2) + CS_{PM}[m] + DCP_S[m] + (DEP_S[m] \times 2) + DCP_H[m]$ $+(DEP_H[m] \times 2) + CAS_{DAC}[m] + CAS_{DE}[m] + CAS_{DMA}[m] + CAS_{DRN}[m]$ $+CAS_{DR}[m] + SE\&OC[m] + (SE[m] \times 2) + SX\&OC[m] + (SX[m] \times 3)$ $+CAS_{MH}[m] + MC_I[m]$
$CL[m]$	CMMN Length	$0 \leq CL[m] \leftarrow random() \leq CAS_{DE}[m]$
$CS[m]$	CMMN Size	$CS_{SC}[m] + CS_{SS}[m] + CS_{SDS}[m] + CS_{SPF}[m] + CS_{DI}[m] + CS_{PT}[m]$ $+CS_{PDT}[m] + CS_{PE}[m] + CS_{PM}[m] + CS_{OC}[m]$
$CAS[m]$	CMMN annotators size	$CAS_{DCP}[m] + CAS_{DEP}[m] + CAS_{DAC}[m] + CAS_{DC}[m] + CAS_{DE}[m]$ $+CAS_{DMA}[m] + CAS_{DRN}[m] + CAS_{DR}[m] + CAS_{SE}[m] + CAS_{SX}[m]$ $+CAS_{MH}[m] + CAS_{MP}[m] + CAS_{MC}[m] + CAS_{MHB}[m] + CAS_{MT}[m]$
$CTS[m]$	CMMN total size	$CS[m]+ CAS[m]$

Table 4: Constraints

Constrain	Description
$CTS = 90$	Average sample was 89, decided to use 90 instead
$(CAS_{MH}[m] + CAS_{MP}[m] + CAS_{MC}[m] + CAS_{MHB}[m]) = (CS_{PT}[m] + CS_{PDT}[m])$	All tasks must have types (guaranteed by generation)
$(DCP_S[m] + DEP_S[m]) \leq (CS_{SC}[m] + CS_{SS}[m] + CS_{SDS}[m])$	Must have less or equal stage planning tables than stages and cases
$(DCP_H[m] + DEP_H[m]) \leq (CAS_{MH}[m] + PT\&MHB[m] + PDT\&MHB[m])$	Must have less or equal human planning tables than human tasks
$(CAS_{DC}[m] + CAS_{DE}[m]) = (CS_{SS}[m] + CS_{SDS}[m] + CS_{SPF}[m])$	expanded and collapsed markers must match the number of stages and planning tables
$CS_{SC}[m] > 0$	must have a case in the model (guaranteed by generation)
$CAS_{DAC}[m] \leq (CS_{SC}[m] + CAS_{DAC}[m] + CS_{SDS}[m])$	cannot have more cases and stages than autocomplete decorators
$CAS_{DMA}[m] \leq (CS_{SS}[m] + CS_{PT}[m])$	cannot have more stages and tasks than manual activation decorators
$CAS_{DRN}[m] \leq (CS_{SS}[m] + CS_{PT}[m] + CS_{PM}[m])$	cannot have more stages, tasks and milestones than repetition decorators
$CAS_{DR}[m] \leq (CS_{SS}[m] + CS_{PT}[m] + CS_{PM}[m])$	cannot have more stages, tasks and milestones than required decorators
$DEP_S[m] \leq (CS_{SDS}[m] + CS_{SPF}[m] + CS_{PDT}[m])$	must have enough discretionary elements for expanded stages planning tables
$DEP_H[m] \leq (CS_{SDS}[m] + CS_{SPF}[m] + CS_{PDT}[m])$	must have enough discretionary elements for expanded human planning tables
$CAS_{DEP}[m] \leq (CS_{SDS}[m] + CS_{SPF}[m] + CS_{PDT}[m])$	must have enough discretionary elements for expanded planning tables
$OC[m] \geq DEP_H[m]$	must have enough extra connectors for expanded human planning tables
$((CS_{SDS}[m] + CS_{SPF}[m] + CS_{PDT}[m]) > 0) \wedge (CAS_{DEP}[m] > 0)$	need expanding planning tables when there are discretionary elements
$(CS_{DI}[m] + CS_{PE}[m] + DEP_H[m]) \leq CS_{OC}[m]$	must have enough connectors for case file items, events and human planning tables

$(CS_{DI}[m] + CS_{PE}[m]) \leq (SE\&OC[m] + SX\&OC[m])$	need entry or exit criteria with connectors for case file items and events
$CAS_{DRN}[m] \leq CAS_{SE}[m]$	repetitions require entry criteria
$(CAS_{SE}[m] \times 2) \leq (CS_{SS}[m] + CS_{PT}[m])$	no more than two entry criterion per task or sentry
$(CAS_{SX}[m] \times 2) \leq (CS_{SS}[m] + CS_{SDS}[m] + CS_{SC}[m])$	no more than two exit criteria per stage or case
$(PT\&MHB[m] + PDT\&MHB[m] + PE\&MHB[m]) = CAS_{MHB}[m]$	correct number of non-blocking human markers (tasks and events)

4 Resulting CMMN models

This section describes the resulting six models and the questions for each model. Each model has five questions (Activity period, Concurrency, Exclusiveness, Order, Repetition, and Notation) using the suggestions in [LG11; Mel+10; MRC07]. The questions and the reasoning for each question is as follows,

Q	Type	Question template	Reason for the question
1	Count	How many x are in this model?	1. Force the subject to scan the complete model. 2. Verify the subject has an understanding of the notation.
2	Order (start)	Is there any situation in which A start executing before B?	1. Comprehension of start order.
3	Order (Case completion)	Can case # complete if A does not executes?	1. Comprehension of case completion order.
4	Order (completion)	Is there any situation in which A completes execution before B?	1. Comprehension of completion order.
5	Order	Which x start when A start executing?	1. Comprehension of stage or case start order.
6	Perceived complexity	How easy to understand is this model? 2. Used to validate the pairwise comparison answers.	1. Gather the subject perceived complexity of the model.

Model	Q1		Q2	Q3	Q4	Q5	
	type	answer	answer	answer	answer	type	answer
Model 1	milestone	5	No	Yes	Yes	tasks	Z,AA
Model 2	n.d.stages	5	Yes	Yes	No	tasks	None
Model 3	c.f.items	4	Yes	No	No	stages	E
Model 4	d.items	16	Yes	No	No	stages	DD
Model 5	events	7	No	No	Yes	tasks	P,W,EE
Model 6	n.d.tasks	5	No	Yes	Yes	stages	None

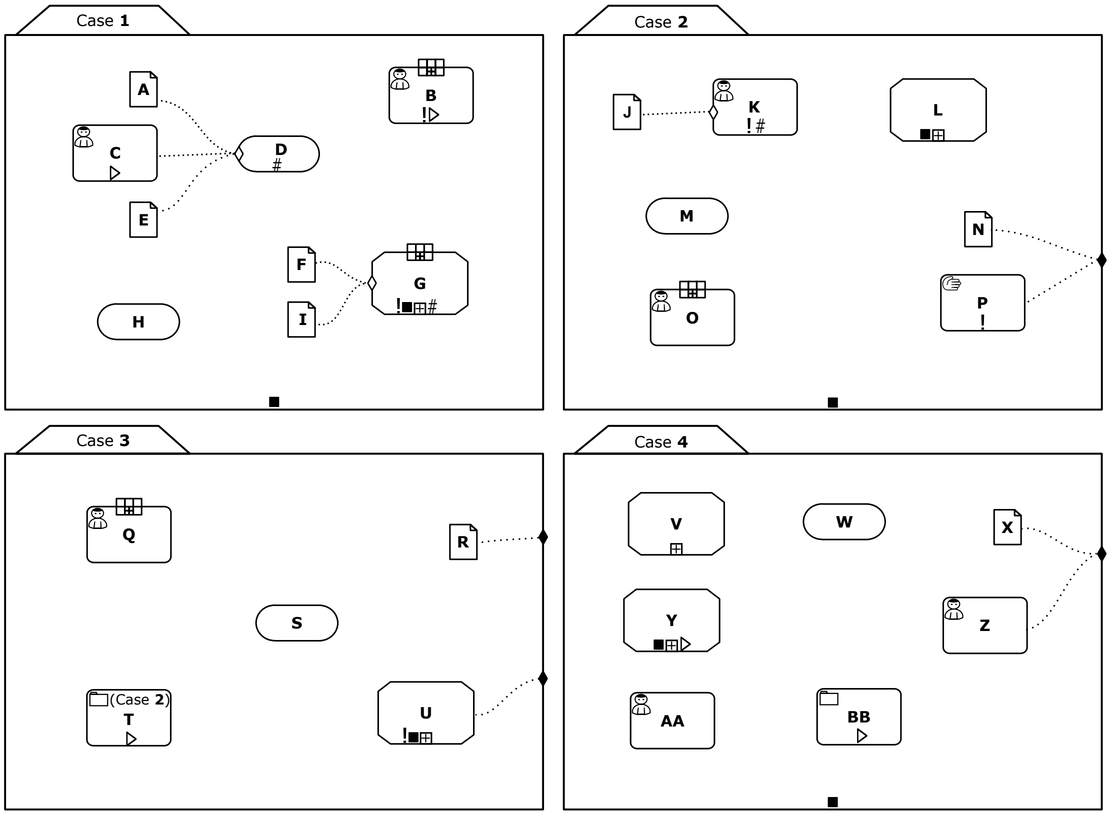


Figure 1: Model 1

Model 1 Questions:

1. (count) How many milestones are in this model? [5]
2. (order) Is there any situation in which O start execution before L? [No]
3. (order) Can case 1 complete if C does not execute? [Yes]
4. (order) Is there any situation in which G completes execution before B? [Yes]
5. (order) Which tasks start executing when case 4 start executing? V, Y, Z, AA, BB, None [Z,AA]
6. (Perceived complexity) How easy to understand is this model? 7-point Likert scale.

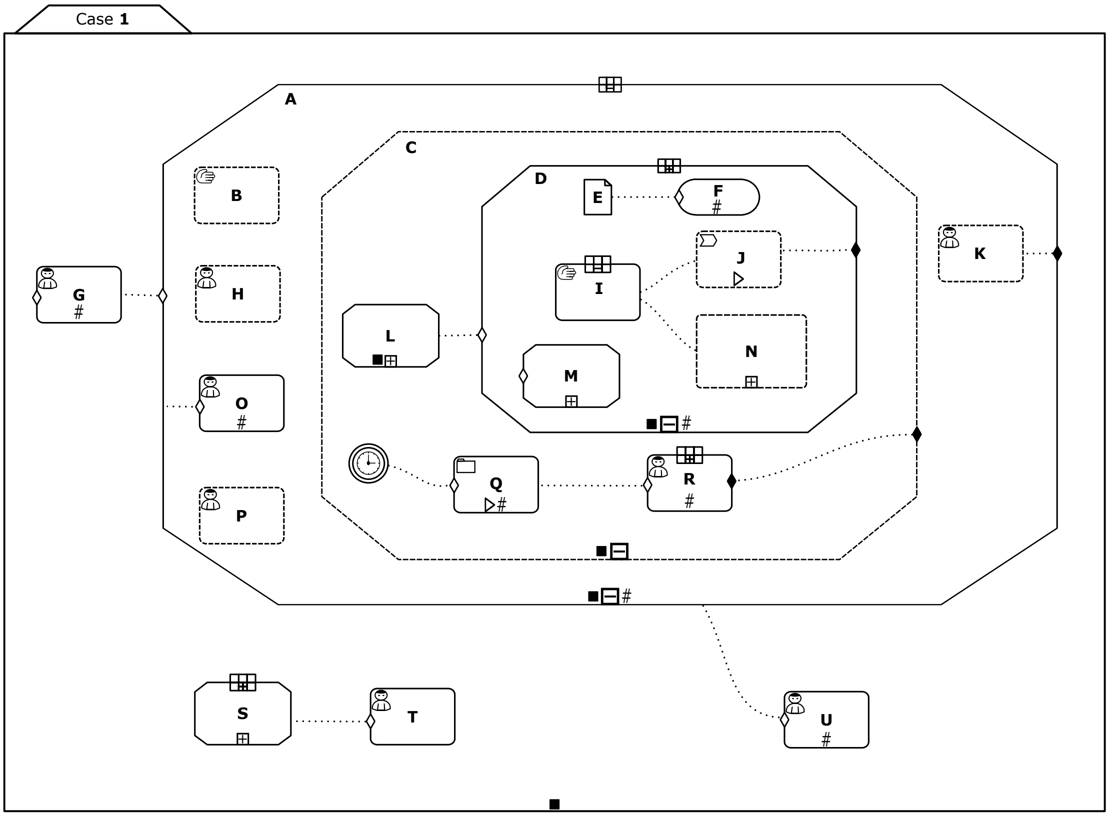


Figure 2: Model 2

Model 2 Questions:

1. (count) How many non-discretionary stages are in this model? [5]

2. (order) Is there any situation in which start execution before ? [Yes]

3. (order) Can case 1 complete if T does not execute? [Yes]

4. (order) Is there any situation in which D completes execution before M? [No]

5. (order) Which tasks start executing when case 1 start executing? A, G, S, T, U, None [None]

6. (Perceived complexity) How easy to understand is this model? 7-point Likert scale.

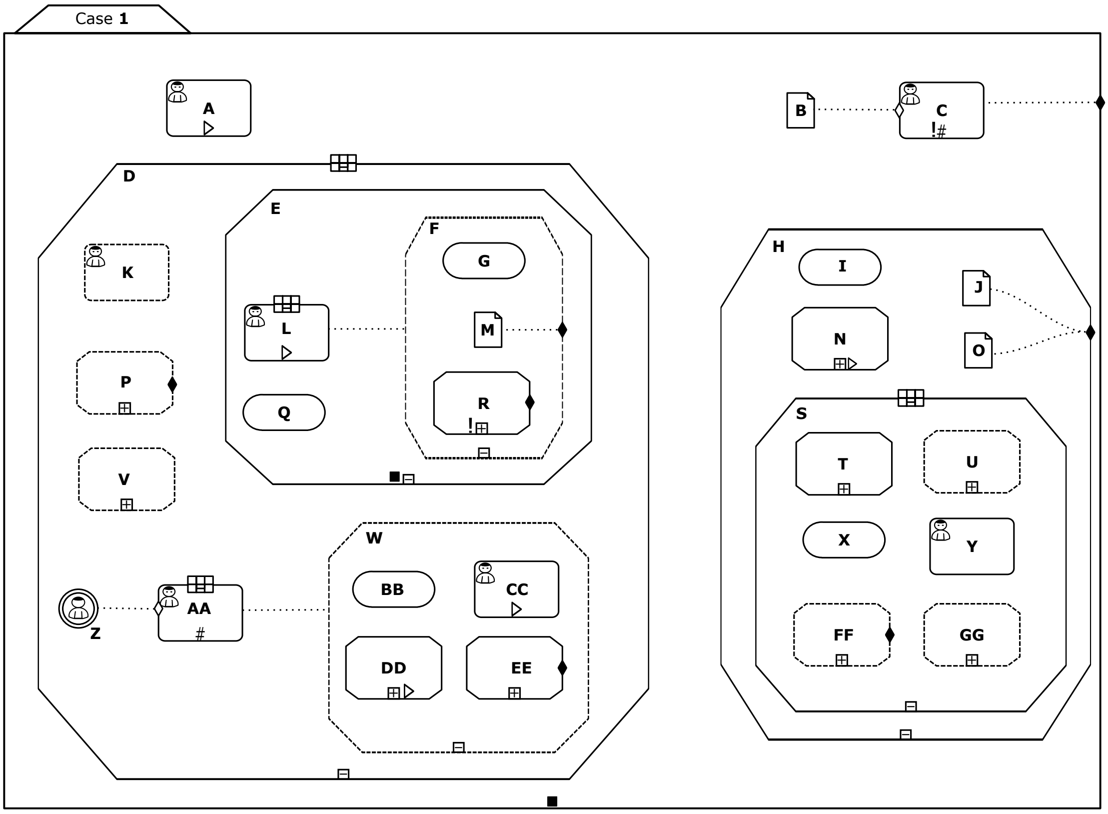


Figure 3: Model 3

Model 3 Questions:

1. (count) How many case file items are in this model? [4]
2. (order) Is there any situation in which C start execution before L? [Yes]
3. (order) Can case 1 complete if C does not execute? [No]
4. (order) Is there any situation in which W completes execution before CC? [No]
5. (order) Which stages start executing when D start executing? E, L, V, W, Z, None [E]
6. (Perceived complexity) How easy to understand is this model? 7-point Likert scale.

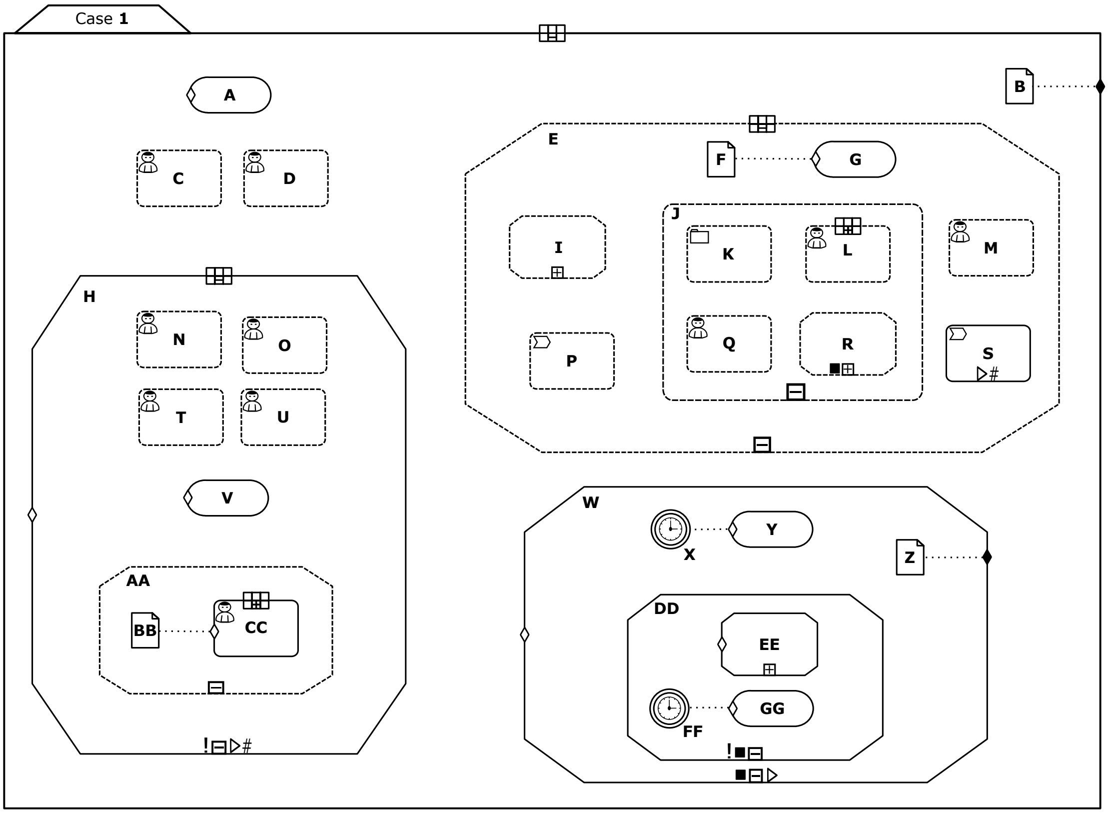


Figure 4: Model 4

Model 4 Questions:

1. (count) How many discretionary items are in this model? [16]
2. (order) Is there any situation in which CC start execution before C? [Yes]
3. (order) Can case 1 complete if H does not execute? [No]
4. (order) Is there any situation in which E completes execution before S? [No]
5. (order) Which stages start executing when W start executing? X, Y, DD, EE, FF, None [DD]
6. (Perceived complexity) How easy to understand is this model? 7-point Likert scale.

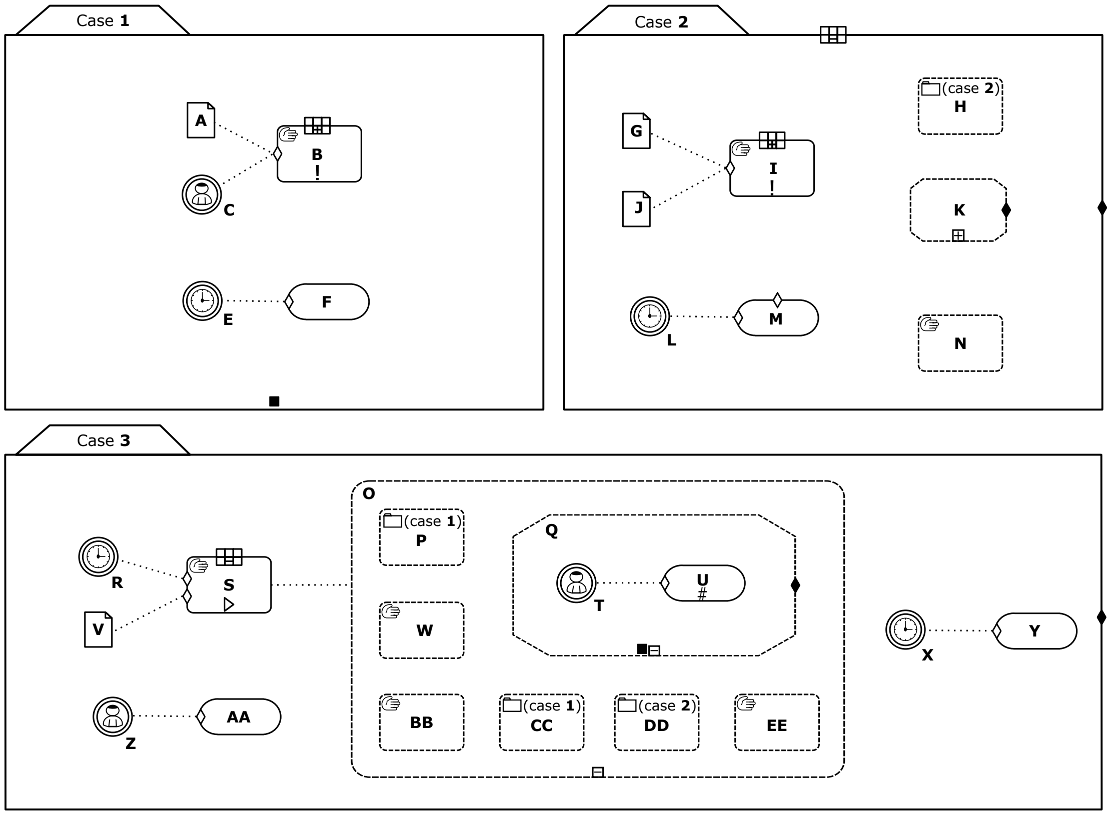


Figure 5: Model 5

Model 5 Questions:

1. (count) How many event listeners are in this model? [6]
2. (order) Is there any situation in which W start execution before P? [No]
3. (order) Can case 2 complete if I does not execute? [No]
4. (order) Is there any situation in which N completes execution before I? [Yes]
5. (order) Which tasks start executing when O start executing? P, Q, T, W, EE, None [P,W,EE]
6. (Perceived complexity) How easy to understand is this model? 7-point Likert scale.

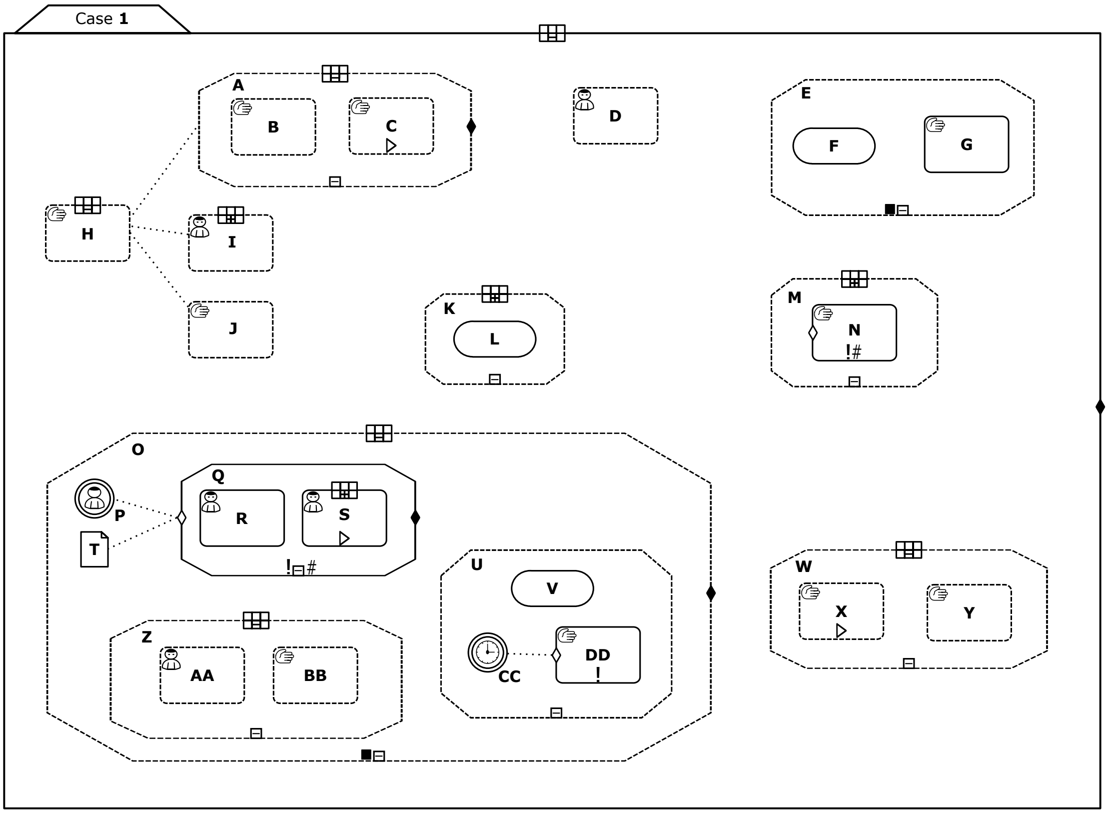


Figure 6: Model 6

Model 6 Questions:

1. (count) How many non-discretionary tasks are in this model? [5]
2. (order) Is there any situation in which S start execution before R? [No]
3. (order) Can case 1 complete if N does not execute? [Yes]
4. (order) Is there any situation in which Y completes execution before X? [Yes]
5. (order) Which stages start executing when O start executing? P, Q, T, Z, U, None [None]
6. (Perceived complexity) How easy to understand is this model? 7-point Likert scale.

Table 7: Metrics for the resulting CMMN models

Metric	Description	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
$iv.\star.CC$	CMMN Complexity	72	82	87	96	102	114
$iv.\star.CL$	CMMN Length	2	5	5	4	4	4
$iv.\star.CS$	CMMN Size	44	36	42	40	45	37
$iv.\star.CAS$	CMMN Annotators Size	46	54	48	50	45	53
$iv.\star.CTS$	CMMN Total Size	90	90	90	90	90	90
Sub-metric							
$iv.\star.CS.SC$	Number of cases	4	1	1	1	3	1
$iv.\star.CS.SS$	Number of stages	5	5	9	4	0	1
$iv.\star.CS.SDS$	Number of discretionary stages	0	1	7	4	2	8
$iv.\star.CS.SPF$	Number of plan fragments	0	1	0	1	1	0
$iv.\star.CS.DI$	Number of case file items	8	1	4	4	4	1
$iv.\star.CS.PT$	Number of tasks	10	7	6	2	3	5
$iv.\star.CS.PDT$	Number of discretionary tasks	0	5	1	11	8	10
$iv.\star.CS.PE$	Number of event listeners	0	0	1	2	7	2
$iv.\star.CS.PM$	Number of milestones	5	1	5	5	6	3
$iv.\star.CS.OC$	Number of connectors	12	13	8	6	12	6
$iv.\star.CAS.DCP$	Number of collapsed planning table decorators	4	3	0	2	2	4
$iv.\star.CAS.DEP$	Number of expanded planning table decorators	0	3	4	3	2	6
$iv.\star.CAS.DAC$	Number of autocomplete decorators	7	7	2	3	2	2
$iv.\star.CAS.DC$	Number of collapsed decorators	5	4	10	3	1	0
$iv.\star.CAS.DE$	Number of expanded decorators	0	3	6	6	2	9
$iv.\star.CAS.DMA$	Number of manual activation decorators	5	2	5	3	1	3
$iv.\star.CAS.DRN$	Number of repetition decorators	3	8	2	2	1	2
$iv.\star.CAS.DR$	Number of required decorators	5	0	2	2	1	3
$iv.\star.CAS.SE$	Number of entry criteria sentries	3	9	2	9	10	3
$iv.\star.CAS.SX$	Number of exit criteria sentries	4	4	7	2	4	4
$iv.\star.CAS.MH$	Number of non-blocking human markers	1	2	0	0	7	10
$iv.\star.CAS.MP$	Number of process markers	0	1	0	2	0	0
$iv.\star.CAS.MC$	Number of case markers	2	1	0	1	4	0
$iv.\star.CAS.MHB$	Number of participant markers	7	8	8	10	3	6
$iv.\star.CAS.MT$	Number of timer markers	0	0	0	2	4	1

5 MiniZic listing

```
int: min_CS_SC; % Min Number of cases
int: max_CS_SC; % Max Number of cases
var min_CS_SC .. max_CS_SC : CS_SC; % Number of cases

int: min_CS_SS; % Min Number of stages
int: max_CS_SS; % Max Number of stages
var min_CS_SS .. max_CS_SS : CS_SS; % Number of stages

int: min_CS_SDS; % Min Number of discretionary stages
int: max_CS_SDS; % Max Number of discretionary stages
var min_CS_SDS .. max_CS_SDS : CS_SDS; % Number of discretionary stages

int: min_CS_SPF; % Min Number of plan fragments
int: max_CS_SPF; % Max Number of plan fragments
var min_CS_SPF .. max_CS_SPF : CS_SPF; % Number of plan fragments

int: min_CS_DI; % Min Number of case file items
int: max_CS_DI; % Max Number of case file items
var min_CS_DI .. max_CS_DI : CS_DI; % Number of case file items

%int: min_PT2MH; % Min non-blocking human task
%int: max_PT2MH; % Max non-blocking human task
%var min_PT2MH .. max_PT2MH : PT2MH; % non-blocking human task
var min_CS_PT .. max_CS_PT : PT2MH; % non-blocking human task

%int: min_PT2MP; % Min process task
%int: max_PT2MP; % Max process task
%var min_PT2MP .. max_PT2MP : PT2MP; % process task
var min_CS_PT .. max_CS_PT : PT2MP; % process task

%int: min_PT2MC; % Min case task referring to a case  not in this model
%int: max_PT2MC; % Max case task referring to a case  not in this model
%var min_PT2MC .. max_PT2MC : PT2MC; % case task referring to a case  not in this model
var min_CS_PT .. max_CS_PT : PT2MC; % case task referring to a case  not in this model
```

```
%int: min_PT2MC2C; % Min case task referring to a case  in this model
%int: max_PT2MC2C; % Max case task referring to a case  in this model
%var min_PT2MC2C .. max_PT2MC2C : PT2MC2C; % case task referring to a case  in this model
var min_CS_PT .. max_CS_PT : PT2MC2C; % case task referring to a case  in this model

%int: min_PT2MHB; % Min blocking human task
%int: max_PT2MHB; % Max blocking human task
%var min_PT2MHB .. max_PT2MHB : PT2MHB; % blocking human task
var min_CS_PT .. max_CS_PT : PT2MHB; % blocking human task

int: min_CS_PT; % Number of tasks
int: max_CS_PT; % Number of tasks
var min_CS_PT .. max_CS_PT: CS_PT; % Number of tasks
CS_PT =  PT2MH + PT2MP + PT2MC + PT2MC2C + PT2MHB;

%int: min_PDT2MH; % Min discretionary non-blocking human task
%int: max_PDT2MH; % Max discretionary non-blocking human task
%var min_PDT2MH .. max_PDT2MH : PDT2MH; % discretionary non-blocking human task
var min_CS_PDT .. max_CS_PDT : PDT2MH; % discretionary non-blocking human task

%int: min_PDT2MP; % Min discretionary process task
%int: max_PDT2MP; % Max discretionary process task
%var min_PDT2MP .. max_PDT2MP : PDT2MP; % discretionary process task
var min_CS_PDT .. max_CS_PDT : PDT2MP; % discretionary process task

%int: min_PDT2MC; % Min discretionary case task referring to a case  not in this model
%int: max_PDT2MC; % Max discretionary case task referring to a case  not in this model
%var min_PDT2MC .. max_PDT2MC : PDT2MC; % discretionary case task referring to a case  not in
     this model
var min_CS_PDT .. max_CS_PDT : PDT2MC; % discretionary case task referring to a case  not in
     this model

%int: min_PDT2MC2C; % Min discretionary case task referring to a case  in this model
%int: max_PDT2MC2C; % Max discretionary case task referring to a case  in this model
%var min_PDT2MC2C .. max_PDT2MC2C : PDT2MC2C; % discretionary case task referring to a case  in
      this model
var min_CS_PDT .. max_CS_PDT : PDT2MC2C; % discretionary case task referring to a case  in this
      model

%int: min_PDT2MHB; % Min discretionary blocking human task
%int: max_PDT2MHB; % Max discretionary blocking human task
%var min_PDT2MHB .. max_PDT2MHB : PDT2MHB; % discretionary blocking human task
var min_CS_PDT .. max_CS_PDT : PDT2MHB; % discretionary blocking human task

int: min_CS_PDT; % Number of tasks
int: max_CS_PDT; % Number of tasks
var min_CS_PDT .. max_CS_PDT: CS_PDT; % Number of discretionary tasks
CS_PDT =  PDT2MH + PDT2MP + PDT2MC + PDT2MC2C + PDT2MHB;


int: min_PE; % Min plan event listener element
int: max_PE; % Max plan event listener element
var min_PE .. max_PE : PE; % plan event listener element

int: min_PE2MHB; % Min user event listener element
int: max_PE2MHB; % Max user event listener element
var min_PE2MHB .. max_PE2MHB : PE2MHB; % user event listener element

int: min_PE2MT; % Min timer event listener element
int: max_PE2MT; % Max timer event listener element
var min_PE2MT .. max_PE2MT : PE2MT; % timer event listener element

var int: CS_PE; % Number of event listeners
CS_PE =  PE + PE2MHB + PE2MT;

int: min_CS_PM; % Min Number of milestones
int: max_CS_PM; % Max Number of milestones
var min_CS_PM .. max_CS_PM : CS_PM; % Number of milestones

int: min_DCP_S; % Min stage collapsed planning table decorator
int: max_DCP_S; % Max stage collapsed planning table decorator
var min_DCP_S .. max_DCP_S : DCP_S; % stage collapsed planning table decorator

int: min_DCP_H; % Min human task  collapsed planning table decorator
int: max_DCP_H; % Max human task  collapsed planning table decorator
var min_DCP_H .. max_DCP_H : DCP_H; % human task  collapsed planning table decorator

var int: CAS_DCP; % Number of collapsed planing table decorators
CAS_DCP =  DCP_S + DCP_H;

int: min_DEP_S; % Min stage expanded planning table decorator
int: max_DEP_S; % Max stage expanded planning table decorator
```

```
var min_DEP_S .. max_DEP_S : DEP_S; % stage expanded planning table decorator

int: min_DEP_H; % Min human task  expanded planning table decorator
int: max_DEP_H; % Max human task  expanded planning table decorator
var min_DEP_H .. max_DEP_H : DEP_H; % human task  expanded planning table decorator

var int: CAS_DEP; % Number of expanded planing table decorators
CAS_DEP =  DEP_S + DEP_H;

int: min_CAS_DAC; % Min Number of auto complete decorators
int: max_CAS_DAC; % Max Number of auto complete decorators
var min_CAS_DAC .. max_CAS_DAC : CAS_DAC; % Number of auto complete decorators

int: min_CAS_DC; % Min Number of collapsed decorators
int: max_CAS_DC; % Max Number of collapsed decorators
var min_CAS_DC .. max_CAS_DC : CAS_DC; % Number of collapsed decorators

int: min_CAS_DE; % Min Number of expanded decorators
int: max_CAS_DE; % Max Number of expanded decorators
var min_CAS_DE .. max_CAS_DE : CAS_DE; % Number of expanded decorators

int: min_CAS_DMA; % Min Number of manual activation decorators
int: max_CAS_DMA; % Max Number of manual activation decorators
var min_CAS_DMA .. max_CAS_DMA : CAS_DMA; % Number of manual activation decorators

int: min_CAS_DRN; % Min Number of repetition decorators
int: max_CAS_DRN; % Max Number of repetition decorators
var min_CAS_DRN .. max_CAS_DRN : CAS_DRN; % Number of repetition decorators

int: min_CAS_DR; % Min Number of required decorators
int: max_CAS_DR; % Max Number of required decorators
var min_CAS_DR .. max_CAS_DR : CAS_DR; % Number of required decorators

int: min_SE2OC; % Min entry criterion sentry with associated connector
int: max_SE2OC; % Max entry criterion sentry with associated connector
var min_SE2OC .. max_SE2OC : SE2OC; % entry criterion sentry with associated connector

int: min_SE; % Min entry criterion sentry without a connector
int: max_SE; % Max entry criterion sentry without a connector
var min_SE .. max_SE : SE; % entry criterion sentry without a connector

var int: CAS_SE; % Number of entry criteria sentries
CAS_SE =  SE2OC + SE;

int: min_SX2OC; % Min exit criterion sentry with associated connector
int: max_SX2OC; % Max exit criterion sentry with associated connector
var min_SX2OC .. max_SX2OC : SX2OC; % exit criterion sentry with associated connector

int: min_SX; % Min exit criterion sentry without a connector
int: max_SX; % Max exit criterion sentry without a connector
var min_SX .. max_SX : SX; % exit criterion sentry without a connector

var int: CAS_SX; % Number of exit criteria sentries
CAS_SX =  SX2OC + SX;

int: min_PT2MHB_PDT2MHB; % min blocking-human-task
int: max_PT2MHB_PDT2MHB; % max blocking-human-task
var min_PT2MHB_PDT2MHB .. max_PT2MHB_PDT2MHB: PT2MHB_PDT2MHB; % blocking-human-task

int: min_OC; % Min optional connector [sentry] element
int: max_OC; % Max optional connector [sentry] element
var min_OC .. max_OC : OC; % optional connector [sentry] element

var int: CS_OC; % Number of connectors
CS_OC =  OC + SE2OC + SX2OC;

int: min_CAS_MH; % Min Number of non-blocking human markers
int: max_CAS_MH; % Max Number of non-blocking human markers
var int: CAS_MH; % Number of non-blocking human markers
CAS_MH =  PT2MH + PDT2MH;
constraint CAS_MH >= min_CAS_MH;
constraint CAS_MH <= max_CAS_MH;

int: min_CAS_MP; % Min Number of process markers
int: max_CAS_MP; % Max Number of process markers
var int: CAS_MP; % Number of process markers
CAS_MP =  PT2MP + PDT2MP;
constraint CAS_MH >= min_CAS_MP;
constraint CAS_MH <= max_CAS_MP;

int: min_MC_I;
int: max_MC_I;
var int: MC_I; % case marker with  case  in model
```

```
MC_I = PT2MC2C + PDT2MC2C;
constraint MC_I >= min_MC_I;
constraint MC_I <= max_MC_I;

int: min_MC_O;
int: max_MC_O;
var int: MC_O; % case marker _with  case not in m
MC_O = PT2MC + PDT2MC;
constraint MC_O >= min_MC_O;
constraint MC_O <= max_MC_O;

var int: CAS_MC; % Number of case markers
CAS_MC =  MC_I + MC_O;

var int: CAS_MHB; % Number of participant markers
CAS_MHB =  PT2MHB + PDT2MHB + PE2MHB;

var int: CAS_MT; % Number of timer markers
CAS_MT =  PE2MT;

%% Now calculate metrics %%

int: weight_CS_SC;
int: weight_CS_SS;
int: weight_CS_SDS;
int: weight_CS_SPF;
int: weight_CS_DI;
int: weight_CS_PT;
int: weight_CS_PDT;
int: weight_PE;
int: weight_PE2MHB;
int: weight_PE2MT;
int: weight_CS_PM;
int: weight_DCP_S;
int: weight_DEP_S;
int: weight_DCP_H;
int: weight_DEP_H;
int: weight_CAS_DAC;
int: weight_CAS_DC;
int: weight_CAS_DE;
int: weight_CAS_DMA;
int: weight_CAS_DRN;
int: weight_CAS_DR;
int: weight_SE2OC;
int: weight_SE;
int: weight_SX2OC;
int: weight_SX;
int: weight_CAS_MH;
int: weight_CAS_MP;
int: weight_MC_O;
int: weight_MC_I;
int: weight_PT2MHB_PDT2MHB;
int: weight_OC;

var 0..185: CC; % CMMN Complexity (max 1 case + 6*3 + 83*2)
CC = (weight_CS_SC * CS_SC) + (weight_CS_SS * CS_SS) + (weight_CS_SDS * CS_SDS) + (
     weight_CS_SPF * CS_SPF) + (weight_CS_DI * CS_DI) + (weight_CS_PT * CS_PT) + (weight_CS_PDT
      * CS_PDT) + (weight_PE * PE) + (weight_PE2MHB * PE2MHB) + (weight_PE2MT * PE2MT) + (
     weight_CS_PM * CS_PM) + (weight_DCP_S * DCP_S) + (weight_DEP_S * DEP_S) + (weight_DCP_H *
     DCP_H) + (weight_DEP_H * DEP_H) + (weight_CAS_DAC * CAS_DAC) + (weight_CAS_DC * CAS_DC) +
     (weight_CAS_DE * CAS_DE) + (weight_CAS_DMA * CAS_DMA) + (weight_CAS_DRN * CAS_DRN) + (
     weight_CAS_DR * CAS_DR) + (weight_SE2OC * SE2OC) + (weight_SE * SE) + (weight_SX2OC *
     SX2OC) + (weight_SX * SX) + (weight_CAS_MH * CAS_MH) + (weight_CAS_MP * CAS_MP) + (
     weight_MC_O * MC_O) + (weight_MC_I * MC_I) + (weight_PT2MHB_PDT2MHB * PT2MHB_PDT2MHB) + (
     weight_OC * OC);

%int: min_CL; % Min  CMMN Length
%int: max_CL; % Max  CMMN Length
var 0..45 : CL; % CMMN Length
constraint CL >= CAS_DE;
constraint CL <= CAS_DE;

var 1..90: CS; % CMMN Size
CS = CS_SC + CS_SS + CS_SDS + CS_SPF + CS_DI + CS_PT + CS_PDT + CS_PE + CS_PM + CS_OC;

var 0..90: CAS; % CMMN annotators size
CAS = CAS_DCP + CAS_DEP + CAS_DAC + CAS_DC + CAS_DE + CAS_DMA + CAS_DRN + CAS_DR + CAS_SE +
     CAS_SX + CAS_MH + CAS_MP + CAS_MC + CAS_MHB + CAS_MT;

var int: CTS; % CMMN total size
CTS = CS +  CAS;
constraint CTS = 90;
```

```
%% All tasks must have types (guaranteed by generation)
constraint (CAS_MH + CAS_MP + CAS_MC + PT2MHB_PDT2MHB) == (CS_PT + CS_PDT);

%% Correct number of non-blocking human markers (tasks and events)
constraint (PT2MHB_PDT2MHB + PE2MHB) = CAS_MHB;

%% Must have less or equal stage planning tables than stages and cases
constraint (DCP_S + DEP_S)  <= (CS_SC + CS_SS + CS_SDS);

%% Must have less or equal human planning tables than human tasks
constraint (DCP_H + DEP_H) <= (CAS_MH + PT2MHB + PDT2MHB);

%% expanded and collapsed markers must match the number of stages and planning tables
constraint (CAS_DC + CAS_DE) == (CS_SS + CS_SDS + CS_SPF);

%% must have a case in the model (guaranteed by generation)
constraint  CS_SC > 0 ;

%% cannot have more  cases and stages than auto complete decorators
constraint CAS_DAC <= (CS_SC + CS_SS + CS_SDS);

%% cannot have more stages and tasks than manual activation decorators
constraint CAS_DMA <= (CS_SS + CS_PT);

%% cannot have more stages, tasks and milestones than repetition decorators
constraint CAS_DRN <= (CS_SS + CS_PT + CS_PM);

%% cannot have more stages, tasks and milestones than required decorators
constraint CAS_DR <= (CS_SS + CS_PT + CS_PM);

%% must have enough discretionary elements for expanded stages planning tables
constraint DEP_S <= (CS_SDS + CS_SPF + CS_PDT);

%% must have enough discretionary elements for expanded human planning tables
constraint DEP_H <= (CS_SDS + CS_SPF + CS_PDT);

%% must have enough discretionary elements for expanded planning tables
constraint CAS_DEP <= (CS_SDS + CS_SPF + CS_PDT);

%% must have enough extra connectors for expanded human planning tables
constraint OC >= DEP_H;

%% need expanding planning tables when there are discretionary elements
constraint ((CS_SDS + CS_SPF + CS_PDT) > 0) /\ (CAS_DEP > 0);

%% must have enough connectors for case file items, events and human planning tables
constraint (CS_DI + CS_PE + DEP_H) <= CS_OC;

%% need entry or exit criteria with connectors for case file items and events
constraint (CS_DI + CS_PE) <= (SE2OC + SX2OC);

%% repetitions require entry criteria
constraint CAS_DRN <= CAS_SE;

% no more than two entry criterion per task and stages
constraint (CAS_SE * 2) <= (CS_SS + CS_PT);

% no more than two exit criteria per stage or case
constraint (CAS_SX * 2) <= (CS_SS + CS_SDS + CS_SC);


solve satisfy;
%solve minimize CC;
%solve maximize CC;

output["\nBasic Symbols",
"\nCS_SC=", show(CS_SC), "\t Number of cases",
"\nCS_SS=", show(CS_SS), "\t Number of stages",
"\nCS_SDS=", show(CS_SDS), "\t Number of discretionary stages",
"\nCS_SPF=", show(CS_SPF), "\t Number of plan fragments",
"\nCS_DI=", show(CS_DI), "\t Number of case file items",
"\nCS_PT=", show(CS_PT), "\t Number of tasks",
"\nCS_PDT=", show(CS_PDT), "\t Number of discretionary tasks",
"\nPE=", show(PE), "\t Event listener elements",
"\nPE2MHB=", show(PE2MHB), "\t User event listener element",
"\nPE2MT=", show(PE2MT), "\t Timer event listener element",
"\nCS_PM=", show(CS_PM), "\t Number of milestones",
"\nDCP_S=", show(DCP_S), "\t Stage collapsed planning table",
"\nDEP_S=", show(DEP_S), "\t Stage expanded planning table",
"\nDCP_H=", show(DCP_H), "\t Human task collapsed planning table",
"\nDEP_H=", show(DEP_H), "\t Human task expanded planning table",
"\nCAS_DAC=", show(CAS_DAC), "\t Number of auto complete decorators",
"\nCAS_DC=", show(CAS_DC), "\t Number of collapsed decorators",
```

```
"\nCAS_DE=", show(CAS_DE), "\t Number of expanded decorators",
"\nCAS_DMA=", show(CAS_DMA), "\t Number of manual activation decorators",
"\nCAS_DRN=", show(CAS_DRN), "\t Number of repetition decorators",
"\nCAS_DR=", show(CAS_DR), "\t Number of required decorators",
"\nSE2OC=", show(SE2OC), "\t Entry criterion with associated connector",
"\nSE=", show(SE), "\t Entry criterion without a connector",
"\nSX2OC=", show(SX2OC), "\t Exit criterion with associated connector",
"\nSX=", show(SX), "\t Exit criterion without a connector",
"\nCAS_MH=", show(CAS_MH), "\t Number of non-blocking human markers",
"\nCAS_MP=", show(CAS_MP), "\t Number of process markers",
"\nMC_O=", show(MC_O), "\t Case marker (case not in this model)",
"\nMC_I=", show(MC_I), "\t Case marker (with  case in this model)",
"\nPT2MHB_PDT2MHB=", show(PT2MHB_PDT2MHB), "\t Blocking human marker",
"\nOC=", show(OC), "\t Extra connector (sentry) element",
"\nMetrics:",
"\nCC=", show(CC),
"\nCL=", show(CL),
"\nCS=", show(CS),
"\nCAS=", show(CAS),
"\nCTS=", show(CTS),
"\nSub-Metrics:",
"\nCS_SC=",show(CS_SC),
"\nCS_SS=",show(CS_SS),
"\nCS_SDS=",show(CS_SDS),
"\nCS_SPF=",show(CS_SPF),
"\nCS_DI=",show(CS_DI),
"\nCS_PT=",show(CS_PT),
"\nCS_PDT=",show(CS_PDT),
"\nCS_PE=",show(CS_PE),
"\nCS_PM=",show(CS_PM),
"\nCS_OC=",show(CS_OC),
"\nCAS_DCP=",show(CAS_DCP),
"\nCAS_DEP=",show(CAS_DEP),
"\nCAS_DAC=",show(CAS_DAC),
"\nCAS_DC=",show(CAS_DC),
"\nCAS_DE=",show(CAS_DE),
"\nCAS_DMA=",show(CAS_DMA),
"\nCAS_DRN=",show(CAS_DRN),
"\nCAS_DR=",show(CAS_DR),
"\nCAS_SE=",show(CAS_SE),
"\nCAS_SX=",show(CAS_SX),
"\nCAS_MH=",show(CAS_MH),
"\nCAS_MP=",show(CAS_MP),
"\nCAS_MC=",show(CAS_MC),
"\nCAS_MHB=",show(CAS_MHB),
"\nCAS_MT=",show(CAS_MT), "\nEnd.\n"];
```

Listing 1: MiniZic constraint program listing

References

[LG11] R. Laue and A. Gadatsch. "Measuring the Understandability of Business Process Models - Are We Asking the Right Questions?" In: *Business Process Management Workshops.* Ed. by M. zur Muehlen and J. Su. Vol. 66. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2011, pp. 37–48. DOI: 10.1007/978-3-642-20511-8_4 (cit. on p. 7).

[Mv p] M. A. Marin and M. von Rosing. "Case Management Model and Notation". In: *The Complete Business Process Handbook, Body of Knowledge from Process Modeling to BPM, Volume II.* Ed. by M. von Rosing, A. W. Scheer, M. Dumas, and H. von Scheel. 1st ed. Morgan Kaufmann, (in print). Chap. Case Management Modeling and Notation (cit. on p. 1).

[Mel+10] J. Melcher, J. Mendling, H. A. Reijers, and D. Seese. "On measuring the understandability of process models". In: *Lecture Notes in Business Information Processing* 43 LNBIP (2010), pp. 465–474. DOI: 10.1007/978-3-642-12186-9_44 (cit. on p. 7).

[MRC07] J. Mendling, H. A. Reijers, and J. Cardoso. "What Makes Process Models Understandable?" In: *5th International Conference on Business Process Management*. Ed. by G. Alonso, P. Dadam, and M. Rosemann. Brisbane, Australia: Springer, 2007, pp. 1–16 (cit. on p. 7).

[Net+07] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. Duck, and G. Tack. "MiniZinc: Towards a Standard CP Modelling Language". In: *Principles and Practice of Constraint Programming – CP 2007*. Ed. by C. Bessière. Vol. 4741. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 529–543. DOI: 10.1007/978-3-540-74970-7_38 (cit. on p. 2).

[OMG14a] OMG. *Case Management Model and Notation (CMMN), FTF Beta 2*. Standard. Document dtc/2013-01-01. OMG, 2014, p. 94 (cit. on p. 1).

[OMG14b] OMG. *Case Management Model and Notation, version 1.0*. Standard. Document formal/2014-05-05. OMG, 2014, p. 96 (cit. on p. 1).