Current DeciClare support of DeciClareEngine

Steven Mertens

Department of Business Informatics and Operations Management
Faculty of Economics and Business Administration
Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
steven.mertens@ugent.be

The current implementation of the DeciClareEngine-tool supports the 18 most important constraint templates of DeciClare. Support for more templates will be added later. Table 1 presents the currently supported constraint templates, each of which is accompanied by an example that illustrates its meaning and parameters. The parameters I and B define the logical operators that can be used in the parameter expressions of the template because of branching. The letter used refers to the option of using either the inclusive disjunction (I) or both the inclusive disjunction and conjunction (B).

Although supported by the DeciClareEngine, Table 1 omits the timing parameters required for each of the templates, except for the untimed RespondedPresence, AtLeastLag and ActivityAuthorization templates, to avoid overloading the table. These are two parameters expressing the time interval in which the constraint applies. For example, Response(AtLeast(e1, 1), AtLeast(e2, 1), 2, 7) expresses that if e1 is executed, then e2 must be executed at least once after at least 2 time units and at most 7 time units after e1 was executed. The lower- and upper-bound time parameters can be 0 and infinity, respectively, reducing them to the examples presented in Table 1.

Finally, DeciClare uses concepts from the DMN decision modeling language to express decision logic. This decision logic is used to define when a constraint can be activated and subsequently deactivated. Therefore, a constraint does not have to be satisfied until the activation decision evaluates as true (i.e., decision-dependent). The conditions of a decision can consist of any combination of values of the available data attributes, chained using the logical conjunctive and disjunctive operators (e.g., 'a = True' and ('b = False' or 'c = "clavicle"')). When a constraint has a trivial decision (i.e., one that always evaluates as true), it applies generally (i.e., decision-independent). To illustrate the concept of activation decisions, consider the decision-dependent healthcare rule for the reaction to trauma cases with internal bleeding. The rule would consist of a constraint Response(AtLeast(Doctor examination, 1), AtLeast(Surgery, 1), 0, 10) and an activation decision that evaluates as true when it is a trauma case and the patient exhibits signs of shock and/or heavy local inflammation or pain. This implies that when a patient is a trauma victim and is showing signs of shock and/or heavy local inflammation or pain, surgery has to be performed within 10 time units after the examination by a doctor. The constraint could be activated during such an examination, prompting emergency surgery.

Table 1. The constraint templates supported by DeciClareEngine

Class	Template	Example
Existence	AtLeast(I, n)	AtLeast(Apply bandage or Apply cast, 2): the sum of the occurrences of Apply bandage and the occurrences of Apply cast has to be at least 2
	AtMost(B, n)	AtMost(Examine patient and Take X-ray, 3): the number of co-occurrences of Examine patient and Take X-ray must be at most 3
	AtLeastChoice(B, n)	AtLeastChoice(Prescribe SAID painkillers or Prescribe NSAID painkillers, 1): at least one of the two activities has to be executed (independent of how many times)
	AtMostChoice(B, n)	AtMostChoice(Prescribe SAID painkillers or Prescribe anticoagulants, 1): at most one of the two activities can to be executed (independent of how many times)
	First(I)	First(Register patient, Examine patient): either Register patient or Examine patient has to be executed before any other event can be executed
	Last(I)	Last(Apply sling or Examine patient): either Apply sling or Examine patient has to be executed as the final event
Relation	RespondedPresence(B_1, B_2)	RespondedPresence(AtLeast((Take X-ray and Apply sling) or Apply bandage, 1), AtLeast(Examine patient or Perform surgery, 1)): if Take X-ray and Apply sling or just Apply bandage are executed at least once, then Examine patient or Perform surgery must also be executed at least once somewhere in the instance (before, during or after)
	Response(B_1, B_2)	Response(AtLeast((Take X-ray and Apply sling) or Apply bandage, 1), AtLeast(Examine patient or Perform surgery, 1)): if Take X-ray and Apply sling or just Apply bandage are executed at least once, then Examine patient or Perform surgery must also be executed at least once at some time after Take X-ray and Apply sling or after Apply bandage
	ChainResponse(B_1, B_2)	ChainResponse(AtLeast((Take X-ray and Apply sling) or Apply bandage, 1), AtLeast(Examine patient or Perform surgery, 1)): if Take X-ray and Apply sling or just Apply bandage are/is executed at least once, then Examine patient or Perform surgery must also be executed at least once immediately after Take X-ray and Apply sling or after Apply bandage
	Precedence(B_1, B_2)	Precedence(AtLeast((Take X-ray and Apply sling) or Apply bandage, 1), AtLeast(Examine patient or Perform surgery, 1)): Take X-ray and Apply sling or just Apply bandage can be executed only after Examine patient or Perform surgery has already been executed at least once
	ChainPrecedence(B_1, B_2)	ChainPrecedence(AtLeast((Take X-ray and Apply sling) or Apply bandage, 1), AtLeast(Examine patient or Perform surgery, 1)): Take X-ray and Apply sling or just Apply bandage can be executed only immediately after Examine patient or Perform surgery has been executed at least once
	AtLeastLag(B_1, B_2, n)	AtLeastLag(AtLeast(Give painkiller, 1), AtLeast(Take anticoagulants, 1), 12): if Give painkiller is executed at least once, then at least 12 time units have to have passed before Take anticoagulants can be executed at least once
Resource	AtLeastAvailable(I, n)	AtLeastAvailable(nurse, 2): there should be at least 2 nurses available to the process (but they are allowed to be in use by the process).
	AtMostAvailable(I, n)	AtMostAvailable(nurse, 4): there should be no more than 4 nurses available to the process
	AtLeastUsage(act, I, n)	AtLeastUsage(Register patient, receptionist, 1): at least 1 receptionist is needed to execute Register patient
	AtMostUsage(act, I, n)	AtMostUsage(Register patient, receptionist, 2): no more than 2 receptionists should be used to execute Register patient
	(No)ActivityAuthorization(I, act)	ActivityAuthorization(doctor, Clinical exam): a doctor is allowed to execute a Clinical exam