1 Impacts of Change Operation Add

Add Context Adding a context implies that instances of the added process will be monitored by the changed ISC. This does not have impacts on the condition and action parts of the ISC structure, but affects the selection of event types instead. The versioning strategy will create a new version and use the same mechanism (i.e., router) to correlate facts with the appropriate version. Migration will create a new version and use the state of the old version. The state includes all facts of the ISC old contexts and the values of the shared variables. Then, facts of process instances related to the added context that are received before t_c, are searched in the working memory and tested against the conditions. Shared variables are then recalculated based on those facts and the facts of the old state. This migration technique is good in complexity, but can be problematic if the ordering of the facts is important. For example, assume an ISC that states that only the first three clients buying a TV will benefit a discount. We assume that there are already two TV buyers a_1 and a_2, and the ISC changed to include laptop buyers. Assume further that before the change there were already two laptop buyers b_1 and b_2 arrived before a_2. Then the migration technique will look for the first arrived out of b_1 and b_2 and add him to both TV clients. This is unfair since both laptop buyers arrived before a_2. The ISC will fire and stop giving discounts. The second migration technique will not consider the old state, but will search from scratch for all facts in the working memory in the right order. This technique is fair but its complexity is not optimal.

Add Connection Adding a connection refers to either adding new event type or a time trigger. The impact of adding an event type is similar to adding a context. However, adding a new time trigger will simply increase the number of times the ISC will be checked.

Add Condition Adding a new condition to an existing ISC assumes that imposes a new constraint on the event types and the connections specified with it. However, as most conditions use shared variables, then the added one might either act on an existing one or create a new variable (e.g., a counter for patients). It is also possible that the entire condition is already modeled and used by another ISC and, therefore, can be directly linked as shared nodes with the changed ISC. A condition can be added either in conjunction or disjunction. the latter can be implemented as two different ISC that lead to the same actions. So in the following we focus on adding a condition in conjunction. With the versioning change strategy, a router is necessary to correlate the events with the corresponding version. The new version uses the same old structure but linked to the new condition structure as additional constraint for the event types. Regarding migration strategy, the order were to insert the new condition among the existing ones is very important. For this purpose, the new condition structure is linked to the old ISC version after the last existing condition. This helps reusing the facts that satisfied all conditions of the old version by filtering them through the added condition. Checking the new condition before the old ones

will prevent reusing facts of the latter as the checking follows a certain order. With respect to actions that fired before the change, if the facts used to fire the ISC also match the new condition, then no compensation action is needed. However, in the opposite case, a compensation action becomes required. For clean state strategy, shared variables need to be reinitialized and the router need to ignore future events of instances started before t_c.

Add Behavior adding a behavior does not impact the ISC structure but the action part instead (RHS). The versioning will trigger the added actions only for the new process instances (as for clean state), and the old actions only for the old process instances. For the ISC instances that fired before the changes, they can be migrated by enacting the new actions if possible.

Add ISC Adding new ISC imposes new restrictions on the process instances. The ISC change needs to be first checked whether or not it conflicts with existing ISCs. In case it does not conflict, the ISC is added to the ISC base and the process instances are checked against the new rule. Since the ISC monitoring engine is only aware of the execution events it receives, and keeps track of the relevant events for only a frame of time, then it is not always possible to include the process instances that started before the deployment of the new ISC in the monitoring of that ISC.