Using Knowledge Engineering for Planning Techniques to leverage the BPM life-cycle for dynamic and adaptive processes

Juan Fdez-Olivares and **Arturo González-Ferrer** and **Inmaculada Sánchez-Garzón** and **Luis Castillo**
Department of Computer Science and Artificial Intelligence
University of Granada

Motivation

The approach here presented deals with the development of AI P&S Knowledge Engineering techniques in order to: (1) automatically generate planning domains from expert knowledge described in BPM (Muehlen and Ho 2006) process models,(2) automatically generate plans, upon these domains, that can be interpreted as business process, and (3) automatically transform these plans back into executable business processes.

The interest of this work is focused on business processes the deployment and execution of which strongly depends on the given context of an organization and, therefore, do not respond to a fixed pattern. An example of such processes may be the organizational process to manage the collaborative creation of e-learning courses within a virtual learning center (a special case of product development processes). Upon a customer request, the manager of the organization needs an estimation of the tasks to be accomplished, the resources of the organization to be used in the elaboration of the course, as well as the time needed to deploy the requested course. Under these circumstances, since the final workflow to be carried out cannot be easily devised a priori, managers and decision makers rely on either project management or business process simulation tools to support decisions about activity planning (in order to find adequate dependencies between tasks and their time and resources constraints), by determining various scenarios and simulate them. But these tools force a knowledge worker to carry out a trial-and-error process that may become unrealistic when either the number of alternatives courses of action makes unmanageable to ascertain which tasks should be considered or the constraints get harder. Therefore, it is widely recognized (WfMC 2010) that, at this step, the life-cycle of BPM presents some weak points and new techniques must be developed at the process modeling/generation step, in order to fully cover the needs of knowledge workers for dynamic, adaptable processes.

From the AI P&S point of view, the need to obtain a context dependent, concrete workflow from a given business process model can be seen as the problem of obtaining a situated plan from an original process model. That is, a plan that represents a case for a given situation, and such that its composing tasks and order relations, as well as its temporal and resource constraints, strongly depend on the context for which the plan is intended to be executed. This is not a trivial problem which requires at least two strong requirement in order to be solved. On the one hand, since the (possibly nested) conditional courses of action that may be found in a process model lead to a vast space of alternative tasks and possible orderings, it is necessary to carry out a search process on this space in order to determine the sequence of actions to be included in the situated plan. On the other hand, the search process necessarily has to be driven by the knowledge of the process model, which in most cases takes a hierarchical structure. Therefore, considering that there is a structural similarity between BPM process models and HTN domain models, we opted in a previous work (Gonzalez-Ferrer, Fdez-Olivares, and Castillo) for the development of Jabbah: a Knowledge Engineering for Planning tool that allows to automatically extract and represent HTN planning knowledge from a business process model. Hence, by using Jabbah in order to generate HTN domain and problem files, from an original process model, it is possible to carry out a knowledge-driven HTN planning process that results in the generation of situated plans, that is, plans customized for a given situation.

These plans can be used either for supporting decision making about activity planning or process validation based on use-case analysis, leveraging the current BPM life-cycle at its process modeling/generation step. Furthermore, Jabbah has been extended (among other features below explained) in order for the business cases obtained to be executed in standard BPM runtime engines. Therefore, Jabbah fulfills, by using AI P&S knowledge engineering techniques, the needs of knowledge workers not yet completely covered by BPM technologies, in the management of dynamic, adaptable processes. Definitively, Jabbah supports most of the BPM life-cycle, from adaptive case generation (starting from a given process model represented in BPM standard languages), to the execution of such processes.

Transformation processes

The BPM models given as input to Jabbah are represented in XPDL, a standard BPM language that is a XML serialization of BPMN, the standard graphical representation for process models. See (Gonzalez-Ferrer, Fdez-Olivares, and Castillo) for a description of the relevant XPDL entities managed by Jabbah.

Transformation from process models to planning domains. Given an XPDL process as input, Jabbah proceeds by identifying common workflow patterns (that is, sequential, parallel, subprocess and conditional structures) as process blocks in the process model, and then maps them into HTN decomposition schemes (a decomposition scheme is formed by a compound task and its associated decomposition methods, each one comprising a set of subtasks arranged by order constraints). Hence, it is possible to use a state-of-art HTN planner that takes this domain representation as input and use its output as activity plans helpful for management tasks. We have used the IACTIVE™ planner for this work, a temporally extended HTN planner which uses an HTN planning language that is a hierarchical extension of PDDL (we call it HTN-PDDL, see (Gonzalez-Ferrer, Fdez-Olivares, and Castillo) for more details).

In its first version, Jabbah was capable of detecting, on the one hand, sequential and parallel blocks, translating them into decomposition schemes with one single method, with subtasks arranged by either sequential or parallel order constraints, respectively. On the other hand, conditional blocks were mapped into decomposition schemes comprising as many methods as the number of alternative courses of action defined by the conditional gateways (see (Gonzalez-Ferrer, Fdez-Olivares, and Castillo) for more details). At present, Jabbah has been extended in order to identify both, subprocess relations between process activities, and more complex synchronization mechanisms between parallel branches described in a process model. Regarding the later extension, once a synchronization between parallel blocks has been detected (represented as a data flow between two activities), Jabbah generates the necessary predicates in the effects of the data producer as well as in the preconditions of the consumer, in order to establish a causal relationship between tasks in the HTN domain.With respect to the former extension it is important to note that it can only by managed by hierarchical planning approaches, and it endows Jabbah with a greater expressivity, allowing it to deal with a wider set of more realistic process models.

By following this process, it is possible to generate problem and domain files which are given as input to the HTN planner in order to obtain situated plans. These plans are generated by the planner for a given context represented in the problem file, and they can be interpreted as adaptive business cases since they are direct and automatically obtained from the initial process model. Furthermore, these plans can also be seen as process instances of the original process model. Thus, next we briefly describe how these plans are transformed back into XPDL process instances in order to be understandable, and so executable, by a BPM runtime engine.

Transformation from plans to executable process models. Given an XPDL process instance as input, BPM engines are commonly endowed with the necessary machinery in order to interactively execute every task (allowing to start, finish, suspend or abort it) in the process by following an execution model based on state-based automata. The plans generated by the planner, using the planning domains and problems generated by Jabbah, are represented in XML as a collection of *Task nodes* where every node contains information about: *actions (activities)* and their parameters; temporal information as *earliest start and earliest end* dates for the execution of every activity; *order dependencies* between actions which allow to establish sequential and parallel runtime control structures; and *metadata* which allow to represent additional knowledge like the user-friendly *description* of a task, its *type* (manual, auto) or its *performer* (that is, the participant of the activity). It is worth to note that metadata are generated at domain generation phase and are automatically extracted and generated by Jabbah.

Starting from this XML plan representation, we have implemented as an extension of Jabbah a translation process that automatically generates XPDL processes which can be directly executed in a BPM runtime engine and users can interact with them on an underlying BPM console (see (Fdez-Olivares et al. 2010) for more details). This process has three main steps: (1) generation of XPDL DataFields and Participants from the problem and domain files; (2) generation of XPDL activities from the information about actions, temporal constraints and metadata in the plan; (3) generation of XPDL transitions from the order dependencies between the actions of the plan.

Notes on experiments

We have applied the transformation processes here described in some experiments, by representing the whole process to develop and deploy a specific course within an e-learning center. Having this process model and an incoming course request, as well as some available workers with different capabilities, we have generated its corresponding HTN planning domain. Then, we have obtained a plan by using the temporally extended HTN planner and we have translated it into an executable XPDL process. Finally, this process has been used as input to a standard BPM runtime engine and console. In conclusion, the system not only allows to support the generation of dynamic, adaptive processes in order to support decision making, in addition it provides the necessary functionalities to execute these processes under user request.

References

Fdez-Olivares, J.; Sanchez-Garzon, I.; Gonzalez-Ferrer, A.; and Castillo, L. 2010. ”Integrating plans into BPM technologies for Human-Centric Process Execution”. In *ICAPS 2010. KEPS Workshop*.

Gonzalez-Ferrer, A.; Fdez-Olivares, J.; and Castillo, L. ”JABBAH: A Java Aplication Framework for the Translation between Business Process Models and HTN”. In *ICKEPS 2009*.

Muehlen, M., and Ho, D. T.-Y. 2006. *Business Process Management Workshops, LNCS 3812*. Springer. chapter ”Risk Management in the BPM Lifecycle”, 454–466.

WfMC. 2010. http://www.xpdl.org/nugen/p/adaptive-case-management/public.htm.