Smart Process Management: Automated Generation of Adaptive Cases based on Intelligent Planning Technologies*

Arturo González-Ferrer, Juan Fdez-Olivares, Inmaculada Sánchez-Garzón, and Luis Castillo

Department of Computer Science and Artificial Intelligence, Universidad de Granada
{arturogf,faro,isanchez,l.castillo}@decsai.ugr.es

Abstract. This demo will present a tool able to extract the knowledge embedded in a BPMN process model, representing this knowledge by means of a Hierarchical Task Network planning domain. An intelligent planner can interpret this domain, obtaining automatically the corresponding process instance for the original model (considering process planning and resource allocation), that is finally deployed for execution into Bonita BPMS. This demo is directed to managers and decision makers in organizations, but it is also interesting for researchers on the field of business performance, optimization and Adaptive Case Management.

1 Introduction

Understanding and estimating the time and necessary resources to complete a business process is a key business challenge. Typically, managers have relied in project management tools (PMT) for planning purposes, but the interdependencies between time and resource constraints make it very difficult to analyze activity costs and resource requirements using traditional PMTs. The introduction of computer-aided Business Process Simulation (BPS) tools traditionally helped to capture the resource constraints, decisions rules and stochastic behavior of real situations. But, while the strength of BPS tools relays on their ability to incorporate stochastic situations in the model, their use imply that, to find the best resource allocation scenario, the manager has to determine various scenarios and simulate them. First, this is not very realistic, as the simulation relays on subruns for a specific scenario which is sometimes not repeatable, as the constraints change over time. Second, BPS tools are based on trial-and-error mechanisms that don't help the manager to do the correct allocation of resources to activities. Sometimes this circumstance can be a serious problem as the constraints get harder, making difficult to find a correct assignment. The framework presented in this paper tackle both BPS inconveniences exposed above by introducing an automated planning and scheduling (P&S) system into the BPM life cycle, capable of both interpreting and reasoning about an initial process model, providing support for decision making on key issues like tasks organization and resources allocation.

* This work has been partially supported by the Andalusian Regional Ministry of Innovation under project P08-TIC-3572.

What's more, our approach makes progress on offering support for Adaptive Case Management[15], since we can, from both a business process model (correctly transformed into a planning representation) and a given state of the world (some conditions of the process environment), obtain situated plans (i.e. adaptive business cases) for a specific given context.

2 The JABBAH framework

The JABBAH system provides a neat tool for analysts that need to perform resource allocation analysis and anticipated process planning on business workflows, embedding a non-trivial transformation of BPMN-expressed process models in terms of Hierarchical Task Networks. Some details about how it works are given in the next subsections.

2.1 Overview

The presented framework has been developed to give support to organization decision makers about business process planning and resource allocation. This is done augmenting the traditional BPM life-cycle by introducing a new stage: design, modeling, *planning*, execution, monitoring and optimization. This is achieved by means of an intelligent component based on artificial intelligence automated planning and scheduling techniques.

Our tool can automatically translate BPMN process models (serialized as XPDL) into a planning representation (following the HTN paradigm) in order to let the user to obtain a process plan automatically, respecting the constraints and control-flow information specified in the original process model. So, starting from a well-structured BPMN process model, we propose a knowledge acquisition stage that allows to build up a corresponding nested process model following a hierarchical structure, i.e. a tree model. This tree model can be built by means of a graph reduction process (by determining the most basic control flow patterns split-join, exclusive-OR and sequences) and the subsequent tree expansion of this reduced graph. The cause for this reduction/expansion process is that all these patterns can be easily represented with our HTN extension of PDDL [9].

Once the planning domain and problem have been generated, the IACTIVE™ intelligent planner[1] can interpret them, obtaining a corresponding plan that respects all the order, resource, and control flow constraints stablished in the process model. A back translation of this plan into XPDL is carried out and deployed into Bonita BPMS, so that the process can be ubiquitously executed and monitored by the participants involved.

2.2 Maturity

Our tool deals at the moment of writing this paper with a basic subset of BPMN elements, those considered to be mostly used by the majority of end-users (*Activities,*

[1] `http://www.iactive.es`

Gateways, Transitions, ActivitySets, Lanes, Participants, Parameters). Concerning the robustness of the framework, we have established some necessary conditions on the input process models, in order to guarantee that the translation is carried out correctly. They must be:

1. Well-structured (every opening gateway has a corresponding ending gateway).
2. Connected.
3. Their maximally connected components must be two-terminal.

As long as these properties are assured, the algorithms are carried out correctly and a corresponding HTN representation can be found for the input process model. Properties 1,2 are mainly directed to assure that the block detection is carried out correctly. Property 3 also allows to define collaborative process models, where different two-terminal components can represent process models of different areas of the organization that can be synchronized (i.e. by using BPMN *Associations* or *Message Flows*). In this case, every component is going to be represented as a different HTN, being the planning goal a composition of all the HTN's goals. The log manager included into the framework allows to identify possible problems that were found in the translation process.

Concerning complexity, the workflow patterns detection algorithm used by our tool is $O(n^2)$, being n the number of edges of the input process model graph.

2.3 Features and Availability

- Import standard XPDL 2.1 files.
- Support for embedded subprocesses.
- Build equivalent graph model and tree model by means of jgraph and jgraphT java libraries.
- Logging system with different message levels (severe/error/warning).
- Support for translating different maximally connected components of a process model (each as a different Hierarchical Task Network).
- Use a specific block detection algorithm for detecting workflow patterns, obtaining a tree model.
- Translate the tree model into planning domain and problem files expressed by using a hierarchical extension of the standard PDDL planning language.
- Achieve task resource allocation, by means of extended attributes 'Duration' (for activities) and 'Lane' (for participants) introduced at modeling time.
- Obtain process plan by means of IACTIVE intelligent planner.
- Display process plan as Gantt diagram.
- Translate back the process plan into Bonita XPDL and deploy it for ubiquitous execution into Bonita BPMS.

The source code of the JABBAH framework is available at its website[2].

[2] `http://sites.google.com/site/bpm2hth`

2.4 Use cases

To date, JABBAH has been tested in two case studies drawn from different domains, e-learning and e-health. The first model represents the whole process to develop and deploy a specific course within the e-learning center at the University of Granada. So, having an incoming course request, as well as some available workers with different capabilities each, we want to assign an activity to every worker, so that a plan over time can be obtained, providing the e-learning managers information about the workers allocation as well as the total make-span of the whole course development, which helps to do anticipated decision-making upon the course request.

The second one represents a general care-process starting from a patient admitted into the hospital and finishing when the health insurance billing for this patient takes place. In this second process, we can better observe how process planning is carried out, given different input parameters which can vary in real situations (is it an emergency? does it need an urgent operation? does the patient need special exam? does the patient need intensive care?) and how our tool is able to generate different process instances according to them. It can also be seen how our translation deals with recursion capabilities (by interpreting and translating correctly the embedded subprocesses in this process model). Both use cases can be explored in the website subsection ”Domain Repository”.

3 Script

A screencast for this demonstration can be seen in the ”Screenshots” subsection of the website, or directly in Youtube[3]. Also a sequence of the features shown in this demonstration can be followed in Figure 1:

Starting from a well-structured process model, for example, of patient admission into an hospital (a), that has been designed under some properties commented in this paper (b), and introducing some extended attributes related to scheduling and allocation (c), we show a tool that, given an XPDL file as input (d) can obtain a corresponding graph model (e), transform it into a semantically equivalent tree model (e). After this transformation, the corresponding planning domain and problem files can be generated (g), and interpreted by an intelligent planner that, on the one hand can obtain a Gantt diagram of the process plan (h), and later on deploy the plan into a BPMS as Bonita Open Solution (i, j).

4 Conclusion

In this demo, we present the JABBAH framework, that introducing a new planning step on the traditional BPM life-cycle -following a previous transformation of an input process model-, is able to offer new decision-making support capabilities to the organization managers. This is carried out following a different approach that traditional Business Simulation engines, by using Hierarchical Task Network planning techniques, offering attractive outcomes as resource allocation, computation of a total make-span or process planning for a given set of input parameters.

[3] `http://www.youtube.com/watch?v=FOHYsMWvS1c`

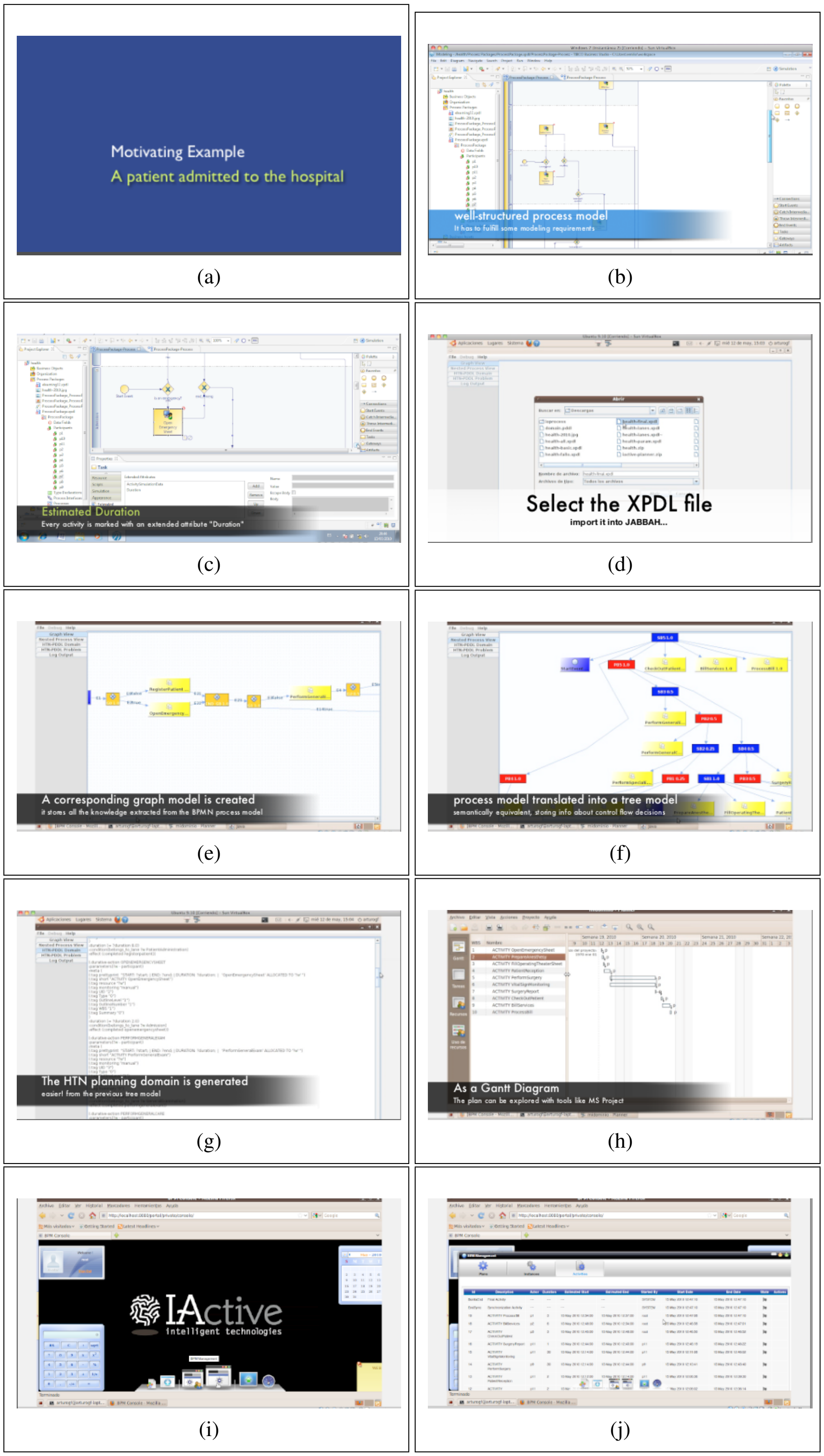


Fig. 1: Script for the demo

Smart Process Management: Automated Generation of Adaptive Cases based on Intelligent Planning Technologies*

Arturo González-Ferrer, Juan Fdez-Olivares, Inmaculada Sánchez-Garzón, and Luis Castillo

Department of Computer Science and Artificial Intelligence, Universidad de Granada, {arturogf,faro,isanchez,l.castillo}@decsai.ugr.es

Abstract. This paper presents a proposal for the seamless integration of Intelligent Planning techniques into the life-cycle of BPM. The integration is intended to leverage current BPM techniques by allowing them to manage smart processes as adaptive business cases that can be automatically generated from original process models and executed in standard BPM runtime engines. The integration of such intelligent techniques is based on a two-fold transformation process: from business models into planning domains, and from plan representations into executable processes.

1 Motivation

Adaptive Case Management [15] is being used by the Workflow Management Coalition as the brand new name of an emergent paradigm in current BPM standard aimed at supporting Human-Centric processes[5] for knowledge workers (highly qualified personnel of organizations, like experts or decision makers). The processes required by knowledge workers are collections of tasks, which usually are collaboratively performed and which necessarily require human interaction in order to control and manage their execution. Such processes commonly support decisions and help to the accomplishment of workflow tasks of knowledge workers in several application domains. For the sake of simplicity, we will designate these processes as *Smart Processes*.

Smart Processes may be viewed as business cases that demand some kind of intelligent management [15] since, on the one hand they are very difficult to foresee, as they respond to *unstructured* sets of procedures which reside either in experts'mind or in documents, what makes difficult to devise a priori which tasks to execute. On the other hand, they need to be adaptively generated, since they are unpredictable and they strongly *depend on the context* of the organization and *do not respond to a fixed pattern*. Finally, they have to be *flexibly and interactively executed* by humans since they are subject to change.

There is a general consensus on that BPM technologies should be improved in order to support this kind of processes since, at the time being, BPM is mainly focused on

* This work has been partially supported by the Andalusian Regional Ministry of Innovation under project P08-TIC-3572.

the management of static, repetitive, even perfectly predictable tasks/processes, mostly devoted to low qualification operators[15]. This is a widely known weakness and, because of this, it is also recognized that new techniques must be developed at both steps, process modeling/generation and process execution, in order to fully cover the needs of knowledge workers on Smart Processes.

In this sense, we present in this work a proposal that leverages the current BPM life-cycle in order to support smart processes through the development of Knowledge Engineering and intelligent planning techniques, focused on a two-fold transformation process. On the one hand, a transformation from business models into planning domains, in order to make the output of a business modeling tool interpretable by an intelligent planner. On the other hand, a transformation from a plan representation into an executable process, in order to make the output of the planner understandable by a BPM runtime engine.

The reason for the first transformation process is the fact that Artificial Intelligent Planning and Scheduling [10] AIP&S is a technology that clearly covers the above exposed demands for smart processes. Concretely, the hierarchical planning paradigm (mostly based on HTN, Hierarchical Task Networks [14,4]) has been proven in many applications([6,3,2,7]) to be successful on supporting the knowledge workers' effort. On the one hand, by modeling expert knowledge with planning domain models (which allow the description of actuation protocols or operating procedures represented as a hierarchy of tasks networks [13,4]). On the other hand, helping them to adaptively produce plans to support their decisions, as the result of a knowledge-driven planning process, guided by the knowledge represented in the planning domain. The reason for the second transformation process is that BPM has demonstrated to be much more appropriate for supporting the execution of the result of knowledge workers' effort, by providing technological infrastructures in order to interactively execute and monitor processes. Therefore, translating a generated plan into a BPM executable format will allow to execute plans on already existing standard platforms. Both transformations are fully automated, what allows to seamlessly integrate these techniques into the current BPM life-cycle, leading to an integrated framework for *Smart Process Management* that supports the automated generation of adaptive cases, from an original business process model based on AI P&S techniques.

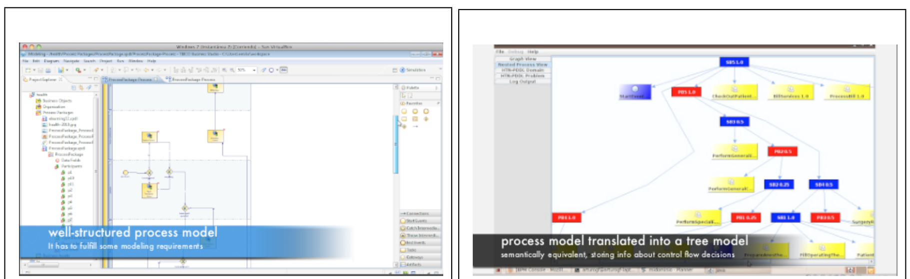
Next sections are devoted to briefly explain the most relevant aspects of the framework as well as its main advantages.

2 Integrating Intelligent Planning into the BPM life-cycle

AI Planning and Scheduling [10] and more concretely HTN planning [4,13] becomes the central technique for this work since it supports the modeling of planning domains in very similar terms to the ones used in standard BPM models. An HTN domain is a compositional hierarchy of tasks networks representing activities at different levels of abstraction (either compound or primitive tasks). A domain describes how every compound task may be decomposed into (compound/primitive) sub-tasks and the order that they must follow, by using different methods. An HTN planner interprets the set of task decomposition schemes and reasons about them in order to compose a suitable plan (a set of activities subject to order and temporal relations) such that its execution reaches a given goal, starting from an initial state (that represents an initial context as initial values for the properties of objects or resources involved in the activities). The HTN reasoning process is a knowledge and goal-driven process, guided by the procedural knowledge encoded in the domain. HTN techniques have been recently enhanced with valuable temporal and resource reasoning extensions [4], what allows to cope with a very rich temporal and resource representation, as well as to obtain plans that could be flexibly executed since they contain temporal constraints that can be adapted during plan execution.

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 (1) a planning domain that represents the original process model and (2) from an initial state that represents the context for which the business case has to be adapted. This aspect is the cornerstone of the proposal here presented, called Jabbah, a Knowledge Engineering for Planning tool that supports a three-step process that starts on an initial, already defined business process model, represented in XPDL using standard BPM modeling techniques. At a first step, the XPDL-based process model is transformed into an HTN planning domain and problem. Second, an HTN planner taking as input the domain and the problem (representing the context under which the case has to be enacted), generates a situated plan that represents the case to be executed. 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. Third, the plan is finally transformed into a process in an executable format (XPDL again), and this process is then used as input to a standard BPM runtime engine in charge of supporting its interactive, human-centered execution. Details about these steps are explained in the following.

Transformation from process models to planning domains. Given an XPDL process as input (which can be clearly seen as a graph), Jabbah proceeds by identifying common workflow patterns (that is, sequential, parallel, subprocess and conditional structures) as process blocks in the process model, and then generate a tree-like structure, much similar to HTN domains. The HTN target domain language (called HTN-PDDL) used in this work is a temporally extended, hierarchical extension of PDDL [9], the standard language of planning domains (see [4,7] for details about this representation). Concerning the representation of preconditions and effects found in planning representations, we only deal at the moment with the conditions that have been defined in the original BPMN model, as well as the task order established on it. However, the BPMN notation allows the inclusion of customized annotations, that could be used to augment the knowledge about preconditions and effects present on the process model (i.e. by using *extendedAttributes* for activity nodes). This said, the Knowledge Engineering process for transforming process models into planning domains consists of three different stages: **i)** Firstly, the XPDL document is parsed, transforming it into an intermediate data structure and graph model that can be easily managed throughout the next stages. **ii)** Then, the different blocks of workflow patterns (serial, parallel, subprocess and conditional blocks) are detected, distinguishing their kind from the knowledge acquired in the previous parsing stage, and build up an equivalent tree-like model. To do this, a collateral challenge is the transformation of the graph into a tree-like structure, which has been done using an algorithm described in [1]. This is carried on by arranging those workflow patterns hierarchically, but also keeping the semantic information (about control flow and decisions) present in the process diagram (see [11] for more details). c) Finally, we need to do a planning language generation phase, where we analyze the tree model that has been populated previously, trying to generalize common patterns found in the graph (i.e. serial or parallel split-joins patterns are always coded in the same way), and writing the HTN-PDDL code that corresponds to the tree-graph fragment analyzed.



(a) (b)

Fig. 1: A well-structured process model is designed with a BPM suite (a), and transformed later into a corresponding tree model by Jabbah (b), more appropriate to extract the HTN planning domain

By following this process, it is possible to generate domain and problem files which are given as input to an state-of-art HTN planner in order to obtain situated plans. We have used the IACTIVE$^{\text{TM}}$ planner for this work, a temporally extended HTN planner which uses HTN-PDDL as its planning language. Moreover, it has already been used in several applications [3,7,6]. 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. Given that the context parameters that guide the deliberative reasoning of the planning stage are included in the problem file, dynamic changes on the environment should be monitored, modifying the problem file accordingly, triggering a replanning stage to generate a new situated plan (the domain file would not be modified, in order to respect the original process model). Some approaches already exists for the monitored execution of plans [12], in order to handle any kind of exogenous events. The design of such execution monitor for the IACTIVE planner is being carried out at the moment of writing this paper.

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 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 in the process (allowing to start, finish, suspend or abort it) 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 [8] 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.

3 Conclusions

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. The main innovative aspects of this framework are both, the fully automated transformation from a business process model (represented in XPDL) into an HTN planning domain and the translation from plan representation into an executable process format, what allows to directly execute the result of a planning process in standard BPM runtime engines. The framework described presents significant advances in the field of BPM, since the seamless integration of the above explained techniques leverages the BPM life-cycle, allowing it to automatically carry out adaptive case generation (starting from an initial process model represented in BPM standard languages), and to execute cases using standard BPM technologies.

To date, JABBAH has been tested in two case studies drawn from different domains, e-learning and e-health. The first model represents the whole process to develop and deploy a specific course within the e-learning center at the University of Granada. So, having an incoming course request, as well as some available workers with different capabilities each, we want to assign an activity to every worker, so that a plan over time can be obtained, providing the e-learning managers information that helps to do anticipated decision-making upon the course request. The second one represents a general care-process starting from a patient admitted into the hospital and finishing when the health insurance billing for this patient takes place. In this second process, we can better observe how process planning is carried out, given different input parameters which can vary in real situations (e.g. is it an emergency? does it need an urgent operation?) and how our tool is able to generate different process instances according to them.

Source code and details about the Jabbah framework and the case studies commented are available at its website[1]. A demonstration screencast about its operation can be watched in the "Screenshots" subsection of the website, or directly in Youtube[2].

References

1. Bae, J., Bae, H., Kang, S., Y.Kim: "Automatic Control of Workflow Processes Using ECA Rules". IEEE Transactions on Knowlegde and Data Engineering 16(8) (2004)
2. Bresina, J.L., Jonsson, A.K., Morris, P., Rajan, K.: Activity planning for the mars exploration rovers. In: Proceedings of the ICAPS05. pp. 40–49 (2005)
3. Castillo, L., Fdez-Olivares, J., García-Pérez, O., Garzón, T., Palao, F.: Reducing the impact of ai planning on end users. In: ICAPS 2007, Workshop on Moving Planning and Scheduling Systems into the Real World. pp. 40–49 (2007)
4. Castillo, L., Fdez-Olivares, J., García-Pérez, O., Palao, F.: Efficiently handling temporal knowledge in an HTN planner. In: Proceeding of ICAPS06. pp. 63–72 (2006)
5. Dayal, U., Hsu, M., Ladin, R.: Business process coordination: State of the art, trends, and open issues. In: Proceedings of the 27th VLDB Conference (2001)
6. Fdez-Olivares, J., Castillo, L., García-Pérez, O., Palao, F.: Bringing users and planning technology together. Experiences in SIADEX. In: Proceedings ICAPS06. pp. 11–20 (2006)
7. Fdez-Olivares, J., Castillo, L., Cozar, J., Garcia-Perez, O.: Supporting clinical processes and decisions by hierarchical planning and scheduling. Computational Intelligence To Appear (2010)
8. Fdez-Olivares, J., González-Ferrer, A., Sanchez-Garzón, I., Castillo, L.: Integrating plans into BPM technologies for human-centric process execution. In: ICAPS 2010. Proceedings of Workshop on Knowledge Engineering for Planning and Scheculing (KEPS) (2010)
9. Fox, M., Long, D.: PDDL2-1: an extension to PDDL for expressing temporal planning domains. Tech. rep., University of Durham, UK (2001)
10. Ghallab, M., Nau, D., Traverso, P.: Automated Planning. Theory and Practice. Morgan Kaufmann (2006)
11. González-Ferrer, A., Fdez-Olivares, J., Castillo, L.: "JABBAH: A Java Aplication Framework for the Translation between Business Process Models and HTN". In: Proceedings of ICKEPS Competition (2009)
12. Leoni, M.D.: Adaptive process management in highly dynamic and pervasive scenarios. In: YR-SOC 2009. pp. 83–97
13. Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., Yaman, F.: SHOP2: An HTN Planning System. Journal of Artificial Intelligence Research 20, 379–404 (2003)
14. Sacerdoti, E.D.: The nonlinear nature of plans. In: Proceedings of IJCAI 1975. pp. 206–214 (1975)
15. WfMC: Group on adaptive case management (2010), http://www.xpdl.org/nugen/p/adaptive-case-management/public.htm

[1] http://sites.google.com/site/bpm2hth

[2] http://www.youtube.com/watch?v=FOHYsMWvS1c