

Supporting Blended Workflows

Davide Passinhas¹, Michael Adams², Bernardo Oliveira Pinto¹, Ricardo Costa¹, António Rito Silva¹, and Arthur H.M. ter Hofstede^{2,3}

¹ INESC-ID/IST/Technical University of Lisbon, Av. Prof. Dr. Cavaco Silva,
2744-016 Porto Salvo, Portugal

{davide.passinhas,bernardo.pinto,ricardo.antunes.costa,rito.silva}@ist.
utl.pt

² Queensland University of Technology, Brisbane, Australia
{mj.adams,a.terhofstede}@qut.edu.au

³ Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract. Despite advances in the field of workflow flexibility, there is still insufficient support for dealing with unforeseen exceptions. In particular, it is challenging to find a solution which preserves the intent of the process as much as possible when such exceptions are encountered. This challenge can be alleviated by making the connection between a process and its objectives more explicit. This paper presents a demo illustrating the blended workflow approach where two specifications are fused together, a “classic” process model and a goal model. End users are guided by the process model but may deviate from this model whenever unexpected situations are encountered. The two models involved provide views on the process and the demo shows how one can switch between these views and how they are kept consistent by the blended workflow engine. A simple example involving the making of a doctor’s appointment illustrates the potential advantages of the proposed approach to both researchers and developers.

1 The Blended Workflow Approach

The blended workflow engine supports a novel approach [1] to workflow management systems that provides end users with two views of the same workflow instance, a “classical” workflow view based on an explicit process model and a view which shows objectives and their relationships. By executing the workflow instance according to the former view, end users’ work is guided by a definition of what is the standard behaviour, specified by explicit organisational rules, whereas the latter view empowers end users to use their tacit domain knowledge to handle unexpected situations.

The two views are supported by two different specifications, one activity-based and the other goal-based. Both specifications share a common data model. As a workflow instance progresses toward its completion via consecutive instantiations of its data model, the two specifications describe a prescriptive and a descriptive way of instantiating the data model.

The two specifications are consistent from a semantic point of view: any workflow instance that is executing according to one of the specifications can continue executing according to the other. Therefore each view can present the current state of the workflow instance to end users, who can switch between the two views without having to redo any of the work they have done while using the other view. However, it is possible to produce more information using the goal-based specification, so that additional knowledge that may be useful when dealing with unexpected situations can be acquired. Additionally, it is possible to relax some restrictions when executing according to the goal view or to skip the execution of activities in the activity view.

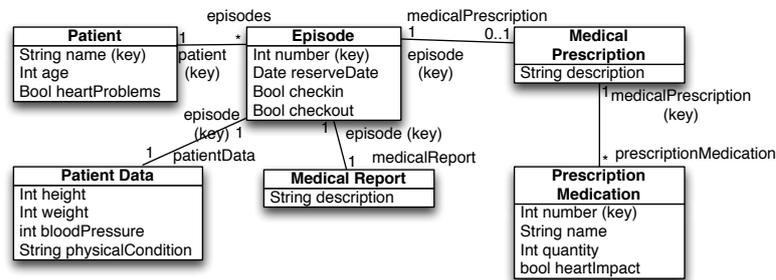


Fig. 1. Doctor Appointment Example - Data Model

Figure 1 presents the data model for a Doctor Appointment workflow specification. The creation of a workflow instance corresponds to the creation of an **Episode** instance and its association to an existing **Patient** instance. The workflow instance progresses by creating instances of the other entities. Eventually a **Medical Report** will be written and the patient checks out.

Figure 2 presents the activity-based specification of the Doctor Appointment example. The specification follows a typical BPMN⁴-like activity-based specification enriched with pre- and post-conditions, denoted respectively by PRE and POS. For each activity, its pre- and post-conditions describe what should be the data model state immediately before and after the execution of the activity, respectively. An activity-based specification can execute in an activity engine in isolation, ignoring its set of pre- and post-conditions, but pre- and post-conditions are necessary for the blended workflow to keep both views synchronized, as we shall show in the demo.

The specification shows that the **Booking** activity creates an instance of **Episode** and sets its **reserveDate** attribute. Note that the creation of an **Episode** instance requires a **Patient** instance and a value for the **number** attribute, which

⁴ <http://www.bpmn.org/>

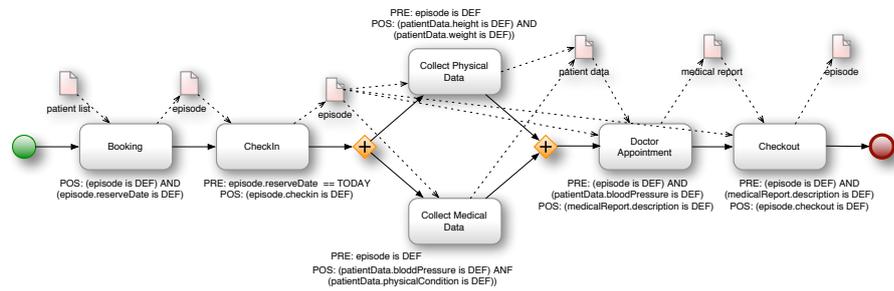


Fig. 2. Doctor Appointment Example - Activity Specification

constitute the key attributes of *Episode*. As another example, note that the execution of *Collect Medical Data* only requires an instance of *Episode*. This relaxed restriction occurs because the blended workflow allows activities to be skipped. In this case it would be possible to skip the *Booking* and *Checkin* activities, but to execute *Collect Medical Data* if an *Episode* instance is created.

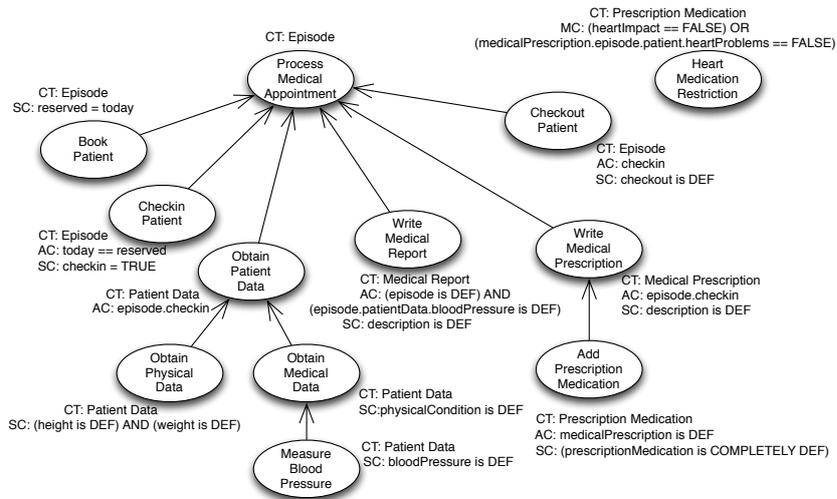


Fig. 3. Doctor Appointment Example - Goals Specification

Figure 3 presents the goal-based specification that adapts from [2] two kinds of goals: achievement and maintain. Achievement goals are specified in a tree and have activation conditions (AC), which should hold true to allow the goal to

become active, and success conditions (SC), which define when a goal is achieved. To achieve a goal it is also necessary that all its sub-goals have succeeded. Additionally, a goal has a context (CT) that represents the data model entity which is the root of its condition's evaluation. A maintain goal describes a data model invariant.

The achievement goal **Write Medical Report** has the **Medical Report** entity as context, and requires both an **Episode** instance and a value for the patient's blood pressure to become active. Upon the goal achievement, the **description** attribute is defined. The maintain goal **Heart Medication Restriction** states that it is not possible to prescribe drugs that may adversely impact the hearts of patients with heart problems.

When executing a workflow instance in the activity view end users can: (1) execute an activity; or (2) skip the execution of an activity. In both situations the process instance progresses according to the control-flow definition. If an activity's pre-condition does not hold when it is enabled according to the control-flow specification, the end user can execute a pre-activity to satisfy the pre-condition.

When executing a workflow instance in the goal view end users can: (1) activate a goal; (2) achieve a goal; (3) skip a goal; (4) redo a goal; (5) disable a goal's activation condition; (6) disable a maintain goal; or (7) create a new goal. When a goal becomes active, its sub-goals also become active. If a goal's success condition holds when it is activated, then it is immediately achieved. When a goal is achieved, the activity view evaluates whether any of the enabled activities can complete. The redo of a goal preserves the state: the same set of conditions continue to hold.

2 Demo Script

In this section we describe the demo script and its objectives. For a screen-cast please visit URL: <http://www.youtube.com/watch?v=Anb4kuXtBgc>. The tool does not include a worklist manager that assigns workitems to users based on their roles because it is not relevant for the blended workflow proof of concept. Therefore in the demo the doctor is able to execute any workitem, either goal or activity.

Second Opinion case

1. Start a **Doctor Appointment** process for Patient **Davide Passinhas**;
2. In the activity view, execute all activities until **Doctor Appointment** activity is enabled;
3. An unexpected situation occurs: during the **Doctor Appointment** activity the patient's condition deteriorates, and the doctor decides to measure his blood pressure again: (1) the doctor switches to the goal view, activates goal **Measure Blood Pressure** and redoes it;

4. Due to this situation the doctor decides to seek a second opinion from a colleague, so in the goal view she: (1) creates a new entity, **Second Opinion**, defines a relationship with entity **Medical Report**, and then creates a new goal, **Second Opinion**, which has a success condition requiring a **Second Opinion**'s **description** attribute to be defined; (2) activates goal **Write Medical Report**, which also activates goal **Second Opinion**; (3) goal **Second Opinion** is enabled by the blended workflow engine; (4) goal **Second Opinion** is achieved by the user; (5) goal **Write Medical Report** is enabled by the blended workflow engine; and (6) goal **Write Medical Report** is achieved by the user;
5. The doctor switches to the activity view and verifies that the **Doctor Appointment** activity is completed, and the **Checkout** activity is enabled for execution;
6. The doctor decides to prescribe a medication. She switches again to the goal view: (1) activates goals **Write Medical Prescription** and **Add Prescription Medication**, with the latter being enabled by the blended workflow engine; (2) achieves goal **Add Prescription Medication**; (3) activates new goal **Add Prescription Medication**, and prescribes a medication that may adversely impact on the heart; (4) tries to achieve goal **Add Prescription Medication** with the prescribed medication, but receives an error message because the maintain goal **Heart Medication Restriction** is violated; (5) the doctor decides to take the responsibility to prescribe the drug anyway, and so disables the maintain goal **Heart Medication Restriction**; and (6) executes goal **Add Prescription Medication**;
7. The doctor switches to the activity view and executes the **Checkout** activity, finishing the process.

Urgency case

1. Start a **Doctor Appointment** process for Patient David Martinho;
2. The **Booking** activity is executed and the **reservedDate** is set for 10 days later;
3. Due to urgency of the patient's condition, the patient requires an examination by the doctor before the **reserveDate** but the **Checkin** activity is not enabled for execution;
4. The doctor switches to the goal view, activates the **Checkin Patient** goal and overrides the goal's activate condition, '**today == reserveDate**', so that the goal can become enabled;
5. The doctor executes the **Checkin Patient** goal. The blended workflow engine automatically completes the **Checkin** activity and enables activities **Collect Physical Data** and **Collect Medical Data**;
6. In the activity view **Collect Physical Data** and **Collect Medical Data** activities are skipped by the doctor, due to the urgency;
7. The doctor executes the pre-activity in the **Doctor Appointment** activity to fill in the missing data, the blood pressure data, and continues execution.

The current version of the tool is a second prototype that extends a first prototype [3] implemented one year ago. It was implemented in JAVA using the FénixFramework⁵ for the domain specification and persistency, Vaadin⁶ for the user interfaces, and the YAWL [4] engine to execute the activity specification. The code is publicly available in a GitHub repository⁷.

3 Acknowledgement

The research was partially supported by the Portuguese Foundation for Science and Technology, FCT-MCTES (INESC-ID multiannual funding) through the PIDDAC Program funds.

References

1. Silva, A.R.: A blended workflow approach. In Daniel, F., Barkaoui, K., Dustdar, S., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., eds.: Business Process Management Workshops. Volume 99 of Lecture Notes in Business Information Processing., Springer (2012) 25–36
2. Van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: Proceedings of the Fifth IEEE International Symposium on Requirements Engineering. RE '01, Washington, DC, USA, IEEE Computer Society (2001) 249–262
3. Pinto, B.O., Silva, A.R.: An architecture for a blended workflow engine. In Daniel, F., Barkaoui, K., Dustdar, S., van der Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., eds.: Business Process Management Workshops. Volume 100 of Lecture Notes in Business Information Processing., Springer (2012) 382–393
4. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N., eds.: Modern Business Process Automation: YAWL and its Support Environment. Springer (2010)

⁵ <https://fenix-ashes.ist.utl.pt/trac/fenix-framework>

⁶ <https://vaadin.com/home>

⁷ <https://github.com/socialsoftware/blended-workflow>