The Process Highlighter: From Texts to Declarative Processes and Back

Hugo A. López[1,2], Søren Debois[1], Thomas T. Hildebrandt[3], and Morten Marquard[2]

[1] IT University of Copenhagen, Denmark
{hual,debois}@itu.dk
[2] DCR Solutions A/S, Copenhagen, Denmark
{hual,mm}@dcrgraphs.net
[3] Copenhagen University, Denmark
hilde@di.ku.dk

Abstract. The adoption of formal models by process specialists has faced two challenges: First, it requires process specialists to get training in formal modeling. Second, the resulting specifications bear little resemblance wrt. the original descriptions. We introduce a tool that supports translations between natural language descriptions and declarative process models. The resulting models are given in a graphical formalism, DCR Graphs. Traceability is at the core of the tool: Later changes in the process model due to, e.g., ambiguity resolution are traced *back* into the text. This allows users to either correct and complete their descriptions, or to derive models more refined than the text. In this paper, we describe the mechanics of the tool and provide examples of its use. Finally, we report on experiences using the tool in a Danish Municipal government.

1 Introduction

This paper describes an implementation of tool support for a method for extracting declarative process models from textual process descriptions. The tool aims at reducing the gap between textual descriptions of processes and their interpretations in a declarative process model. The tool is particularly aimed at supporting users who are *not* trained in formal methods.

Both the tool and the method exploit the regularity of contemporary constraint-based process notations with trace semantics such as DECLARE [1,8] or DCR graphs [7,3]: That constraints are expressed as relations between a fixed number of activities. This regularity is actually present and respected already in textual descriptions of processes. For instance, suppose our text says:

"Payout requires pre-approval by the line manager" (1)

Then activities "Pre-approval" and "Payout" are related by a DCR or DECLARE *condition*-relation. Thus, there is a correspondence between the constraint in the declarative process model on the one hand and this particular sentence on

F. Casati et al. (Eds.): Proceedings of the Dissertation Award and Demonstration, Industrial Track at BPM 2018, CEUR-WS.org, 2018. Copyright © 2018 for this paper by its authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

the other. The highlighter tool reported on here helps to establish and maintain that correspondence. It allows users to "mark up" a textual process description, as if with a yellow marker; the tool then generates a formal declarative DCR process model from that markup. The correspondence between model and text is bi-directional: later changes to the model may change the markup and vice versa.

The purpose of the tool is to allow domain experts with some familiarity with the modelling notation, to easily *construct and maintain* declarative models. The tool is currently being used at both industrial and academic environments, and user feedback indicates that both the development method and the tool are effective in establishing and maintaining a correspondences between a textual process descriptions and a formal constraint-based process models.

2 Example Usage

We illustrate the tool via an example process description from the BPM Academic Initiative[4], presented in Fig. 1:

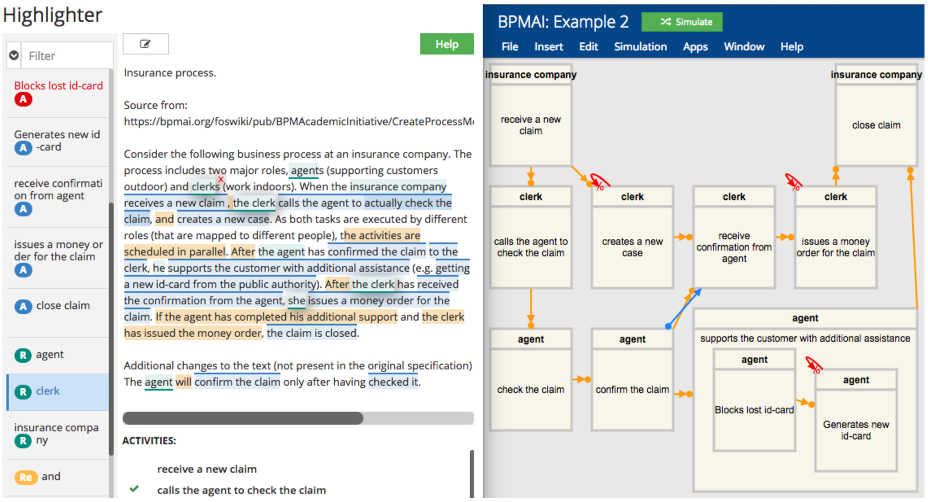
> The process includes two major roles, agents (supporting customers outdoor) and clerks (work indoors). When the insurance company receives a new claim, the clerk calls the agent to actually check the claim, and creates a new case. As both tasks are executed by different roles (that are mapped to different people), the activities are scheduled in parallel. After the agent has confirmed the claim to the clerk, he supports the customer with additional assistance (e.g. getting a new id-card from the public authority). After the clerk has received the confirmation from the agent, she issues a money order for the claim. If the agent has completed his additional support and the clerk has issued the money order, the claim is closed.

Fig. 1: Insurance process description

DCR Graphs are the formal foundations of the process engine developed by DCR Solutions. For sake of space, we do not provide formal definitions, referring the reader to [7,4] instead. As the name suggests, a DCR graph is a (directed multi-)graph: nodes of the graph are activities in a process, and edges denote constraints between activities. Activities are there to be executed, and relations regulate the state and executability of activities:

- A *condition* relation from $A \rightarrow\!\bullet B$ means that B can only execute if A already has been executed, or has been excluded (see below).
- A *response* relation $A \bullet\!\!\rightarrow B$ indicates that whenever A executes, B changes state to *pending*, that is, it needs further execution (or exclusion) of B for the workflow to be complete.
- An *exclusion* (resp. *inclusion*) relation $A \rightarrow\!\% B$ (resp. $A \rightarrow\!+ B$) indicates that whenever A is executed, B is excluded from (resp. included in) the workflow: An excluded activity cannot execute and is ignored as condition and not required to execute if pending, unless it is re-included by an include relation.

[4] https://bpmai.org/foswiki/pub/BPMAcademicInitiative/CreateProcessModels/ex2_insurance_process.pdf



Insurance process.

Source from:
https://bpmai.org/foswiki/pub/BPMAcademicInitiative/CreateProcessM

Consider the following business process at an insurance company. The process includes two major roles, agents (supporting customers outdoor) and clerks (work indoors). When the insurance company receives a new claim , the clerk calls the agent to actually check the claim, and creates a new case. As both tasks are executed by different roles (that are mapped to different people), the activities are scheduled in parallel. After the agent has confirmed the claim to the clerk, he supports the customer with additional assistance (e.g. getting a new id-card from the public authority). After the clerk has received the confirmation from the agent, she issues a money order for the claim. If the agent has completed his additional support and the clerk has issued the money order, the claim is closed.

Additional changes to the text (not present in the original specification)
The agent will confirm the claim only after having checked it.

ACTIVITIES:

receive a new claim

✓ calls the agent to check the claim

Fig. 3: DCR Highlighter: filter (left), textual (centre) and model (right) views

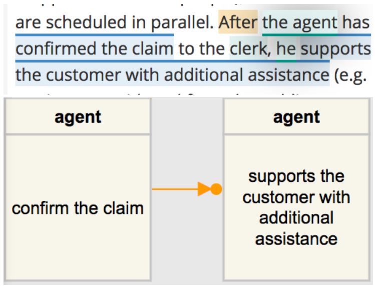

are scheduled in parallel. After the agent has confirmed the claim to the clerk, he supports the customer with additional assistance (e.g.

Fig. 2: Correspondence between text and DCR graphs

From texts to processes. We start by creating process models from texts. Fig. 1 contains three explicit roles (agent, clerk, customer) and one implicit role (insurance company). Using the tool, we mark the text of each role and mark it up as such.

We proceed by identifying activities. For instance, fragments "confirm the claim" and "support customer" are marked as activities in the tool (see top of Fig 2). Changes in the title of each activity to improve readability are supported, e.g., "confirm the claim" vs. "confirm<u>ed</u> the claim".

We proceed to find relations. The word "After" in line 5 of Fig. 1 suggests causality between agents' activities "confirm claim" and "support customer". When marking up "After" as a relation, the tool allows setting the type of relation, as well as which activities are related. Such mark-up creates a relation in the graph (a condition arrow). Fig. 2 shows the highlighter in action in an excerpt of Fig. 1; we see the text fragment at the top, and (part of) the generated graph at the bottom. The full mapping is shown in Fig. 3.

Advanced use. Constraints may be explicit (e.g. "after", "and", "if-then"), or implicit (i.e.: the first highlighted comma). Other aspects may hinder the mapping between texts and DCR graphs:

i. A sentence might denote the refinement of an activity. i.e.: "The evaluation process might contain evaluations regarding chronic and lifelong conditions of the patient". Here, activity (evaluation) needs to be refined into several (separate assessments for lifelong and for chronic diseases).
ii. A sentence might denote internal activities (performed by a single role) as well as communication activities (i.e.: lines 2–3 in Fig. 1).
iii. Elements identified in textual paragraphs might be referenced later.
iv. Relations might include one-to-one, many-to-one and one-to-many cardinality between activities.

The tool supports these scenarios. First, multiple markings of the same text will denote groups (abstractions) and refinements (nested activities). Second, multiple highlights denote communication activities, one role per marked text. Third, the tool allows the creation of references to existing elements. Finally, the creation of relations within the text allows users to include more than one relation per marked text, with possibly different actors and types involved.

2.1 ... "and back": Support for Model Changes

In practice, both models and texts may evolve over time. These changes do not break correspondences with the model. The filter view (c.f.: Fig. 3) shows the mapping between components and highlighted texts. Elements in the model that do not have a mapping to the text are shown in red. In the example, those elements are the new activities that refine additional support. This distinction helps users identify deviations from the model, and to correct them either by linking the model components to the text, or by updating the text description.

3 Concluding Remarks

Despite a raising interest in the alignment of process models and textual descriptions [6,5,9], authors are not aware of other works exploring such alignment with *declarative* process models. At its current stage, the highlighter depends on the knowledge of the process modeler to resolve inherent ambiguities coming from the interpretation of natural language descriptions. In future work, we would like to explore (semi)-automated declarative process discovery from texts, via the integration of the highlighter tool with NLP techniques [6].

The highlighter tool has been validated in industrial and academic environments. First, the tool has been introduced to a small number of end-users from the Syddjurs municipality in Denmark, who have used it to build models corresponding to the municipal governments internal guidelines on processes for implementing Section 42 of the Danish Consolidation Act on Social Services, which describe rights to compensation for loss of earnings awarded to parents of children with long-term illness or disabilities. Users had a background in law and social work, and had previously received training in process modelling using DCR graphs. Users' interaction with the tool resulted in their independent process models[5] End-users reported that the highlighter was effective and easy to use for the intended purpose[6]. Finally, the highlighter tool is currently being used in B.Sc. and M.Sc. courses on business process modelling at the ITU.

3.1 Availability, Documentation and Screencast

The tool has been implemented by DCR Solutions A/S as part of their set of tools for declarative process modelling and execution, and is part of their commercial on-line offering [2]. The highlighter tool is free for non-commercial and academic use. Documentation, installation and further examples of use are available in our wiki site[7]. A screencast documenting its usage is available at `https://youtu.be/JB9ueRu_asE`.

Acknowledgments. Work supported by the Innovation Fund Denmark project Eco-Know.org (7050-00034A). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement BehAPI No.778233.

References

1. Aalst, W.M.P.v.d., Pesic, M.: DecSerFlow: Towards a Truly Declarative Service Flow Language. In: WSFM. pp. 1–23. LNCS, Springer, Berlin, Heidelberg (Sep 2006)
2. Debois, S., Hildebrandt, T.T., Marquard, M., Slaats, T.: The DCR graphs process portal. In: BPM (Demos). CEUR Workshop Proceedings, vol. 1789, pp. 7–11. CEUR-WS.org (2016)
3. Debois, S., Hildebrandt, T.: The DCR Workbench: Declarative Choreographies for Collaborative Processes. In: Behavioural Types: from Theory to Tools, pp. 99–124. River Publishers (2017)
4. Debois, S., Hildebrandt, T.T., Slaats, T.: Replication, refinement & reachability: complexity in dynamic condition-response graphs. Acta Informatica pp. 1–32 (2017)
5. Delicado, L., Sánchez-Ferreres, J., Carmona, J., Padró, L.: NLP4BPM - natural language processing tools for business process management. In: BPM (Demos). CEUR Workshop Proceedings, vol. 1920. CEUR-WS.org (2017)
6. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: CAiSE. pp. 482–496. Springer (2011)
7. Hildebrandt, T., Mukkamala, R.R.: Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. In: PLACES. EPTCS, vol. 69, pp. 59–73 (2010)
8. Pesic, M., Schonenberg, H., Aalst, W.M.P.v.d.: DECLARE: Full Support for Loosely-Structured Processes. In: EDOC. pp. 287–287 (Oct 2007)
9. Riefer, M., Ternis, S.F., Thaler, T.: Mining process models from natural language text: A state-of-the-art analysis. Multikonferenz Wirtschaftsinformatik (MKWI-16) pp. 9–11 (2016)

[5] `http://www.dcrgraphs.net/Tool?id=7119`

[6] However, they did report that some aspects of the guidelines, e.g., the criteria an assessment of a child's special needs must meet to be lawful, were not straightforward to map to the BPM concepts of "Activity". This concern is orthogonal to the tool itself: such difficulty would manifest also in a pen-and-paper model.

[7] `http://wiki.dcrgraphs.net`