Assisted Declarative Process Creation from Natural Language Descriptions*

Hugo A. López[1,2] Morten Marquard[2] Lukas Muttenthaler[3] Rasmus Strømsted[2]

***Abstract*— In this paper, we report recent advances on user support for declarative process generation from natural language descriptions. The Process Highlighter is a hybrid-modelling tool that facilitates the (manual) creation of Dynamic Response Condition (DCR) graphs directly from text documents, supporting non-technical users in the adoption of declarative process models. While some process descriptions are a few paragraphs long, others, such as the ones coming from municipal governments and legal bodies might contain several pages. Some aspects that undermine the adoption of hybrid modelling techniques and their promised one-to-one correspondence between texts and process models are the length of the texts, the inconsistent use of terms, and the difficulty in identifying textual elements that correspond to elements in a declarative process model. To mitigate these risks, we have implemented major additions in the Process Highlighter for industrial usage. The principal change is the inclusion of Natural Language Processing (NLP) techniques to support users in the identification of roles, activities and constraints. This, combined with the modelling, simulation and verification tools already existing in the framework, support the users in providing process models that are better aligned with their specifications, in a shorter time. These features are motivated from empirical observations of the use of the Process Highlighter in groups of caseworkers and students of process engineering in Danish universities.**

***Index Terms*— Natural Language Processing, Business Process Management, Declarative Process Models, DCR graphs.**

I. Introduction

Process Digitalisation describes the transformation of inter-organisational processes from paper-based descriptions to digital-based ones, with the promise that once in their digital form, they will be subject to analysis and automation. In particular, the digitalisation of the public sector requires a joint effort between specialists in law and in computer science. In Denmark, these two sectors have collaborated actively for more than 30 years, which has resulted in modernisation programs that pointed to the use of new technologies to cut on bureaucracy. Since 2018, all new legislations should be ready for digitalization [2], which among its principles, includes 1) the drafting of simple and distinct rules to contribute with a more uniform and digital administration, and 2) considerations on how to enable automated digital case processing while still taking into account the legal rights for citizens and businesses. Digitalisation of laws also poses challenges in the adaptive case management systems supporting digital case processing. Firstly, how to identify the information in paper-based descriptions that is important in a digital process. Secondly, how to mitigate the inherent ambiguity coming from natural language descriptions. Thirdly, how to streamline process digitalisation efforts, when paper-based process descriptions normally spans over several pages long.

The project EcoKnow[1] focuses on solving these challenges via declarative process models. The Dynamic Condition Response (DCR) graphs [9], [15] is a theory originally introduced for the formalisation and mechanisation of adaptive case management (ACM) processes. The theory has evolved into a commercial offering that supports digitalization initiatives of one of the biggest Danish developer houses, who has integrated DCR graphs in a case management solution used in 70% of Danish central government institutions[2]. As a way to reduce the gap between textual process descriptions and their formal interpretation, the Process Highlighter [12] (or simply "the highlighter") was created. One of the main uses of the tool is the agile creation of process models: from a natural language description (for example, a law), users can identify the activities, roles, and relations involved in a business process, by simply marking them in the text. Resulting models are *executable*, which means that they can be plugged into an ACM system. The highlighter has been used as an educational tool in the software engineering and business process management courses at the IT University of Copenhagen, and the Technical University of Denmark,

After a year since its initial release, we have collected experiences from the use of the highlighter in these different realms. They come in the form of empirical evaluations of the tool [3], and discussions with users. Most evaluations have been positive, and our exploratory results suggest that the hybrid modelling approach in the highlighter provides cognitive support to process modellers and contributes to an enhanced modelling experience. The highlights used to mark-up specific fragments of the process description can be associated with a phenomenon referred in Cognitive Psychology as the isolation effect [10], which has been shown to increase the reader attention on specific parts of the text [5], and make sense of the process description. However, our evaluations also raised questions regarding the use in

[1]Computer Science Department, IT University of Copenhagen, Denmark. Email: hual@itu.dk

[2]DCR Solutions A/S, Copenhagen, Denmark. Emails: {hual,mm,rs}@dcrsolutions.net

[3]Department of Computer Science, University of Copenhagen. Email: mnd926@alumni.ku.dk

*Authors in alphabetical order

[1] www.ecoknow.org

[2] http://www.kmd.dk/indsigter/fleksibilitet-og-dynamisk-sagsbehandling-i-staten

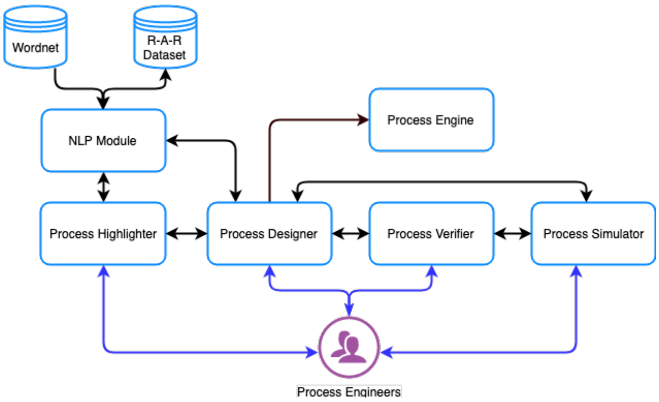


Fig. 1. System Overview

industrial-size cases, and suggested plans for improvement. While some elements in a declarative process model might be easier to identify, some others, such as the constraints between activities, are not obvious for users not trained in declarative process models before. Moreover, the one-to-one correspondence between texts and processes claimed by the highlighter is threatened if 1) users do not find ways to extend the original descriptions to account for novel constraints, or 2) they need to manually mark all terms, including those that have been identified in earlier parts processed by the user.

In this paper we present efforts in providing assistance for users to automate process highlights. We believe that these additions will help users to increase the speed in digitalising process models, support the traceability of models with respect to requirements, and maintain models aligned with specifications.

II. System Overview

Fig. 1 shows the components of the most recent version of the DCR process portal [6]. Each box represents a component, and the lines connecting them denote their interactions. The artefacts used in each component are different:

- Natural language text and process models in the process highlighter,
- A process model in the process designer and in the process engine,
- A process model and a set of predefined properties in the process verifier, and
- A process model and a set of scenarios in the process simulator.

The DCR process portal offers modules for modelling, simulation, verification, and execution. The modelling phase considers an incremental construction of process descriptions and their corresponding models. Modellers might choose to start with an initial process description that will be refined in the interaction with the model, or alternatively, start modelling and document the process afterwards. The alignment is performed via *highlights*: engineers mark how different parts of the requirements correspond to elements in the process model, and vice-versa. Ideally, this mapping should be bijective [4]. This means, that if there is a constraint described in natural language, its constituents should be formally defined in the process model, and that all constraints in the model have a natural language explanation in the process description. In reality, both process models and their documentation may be changed at different stages (e.g.: due to the interaction with other artefacts). The highlighter provides traceability for those changes and it will inform which parts of the model require a proper documentation.

A novel aspect in this version of the Process Highlighter is the inclusion of a Natural Language Processing (NLP) module. Its main purpose of supporting the automation of alignments between process models and documentation. Secondly, it helps identifying difficult language patterns that undermine the understandability of process models. Such patterns are for instance, the definition of activities with non-compliant labelling (an issue presented first for imperative process models [11], also present in declarative models).

The NLP module relies on a mixed dataset. On the one hand, it uses the Wordnet pre-trained dataset [14]. WordNet provides a lexical database of English, where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets). The process highlighter presents suggestions for process model components based in this data. In addition, the NLP module interacts with a dataset containing roles, activities, and relations (R-A-R in Fig. 1). This dataset is trained via user interactions: each suggestion accepted/rejected via the process highlighter changes the accuracy of the suggestions presented. The NLP module combines both datasets when providing suggestions.

While the interaction between the highlighter and the designer results in a process model, little is known regarding the quality of the model generated. The portal provides two forms for ensuring quality in declarative process models. For structural correctness, the process verifier checks for reachability properties, as well as deadlocks and livelocks. For semantic correctness, the workbench implements tooling supporting test-based modelling, and the simulation tool [13] verifies whether positive scenarios (behaviour that should be supported) are accepted, and whether negative scenarios (behaviour that should be restricted) are impossible to replicate. The interplay between simulation and verification tools will modify the process model, for instance, by including more constraints to forbid negative scenarios from happening.

Finally, each process model can be instantiated and executed in a case management system [17], keeping track of each instance, the activities executed, and the documents associated for each instance [18].

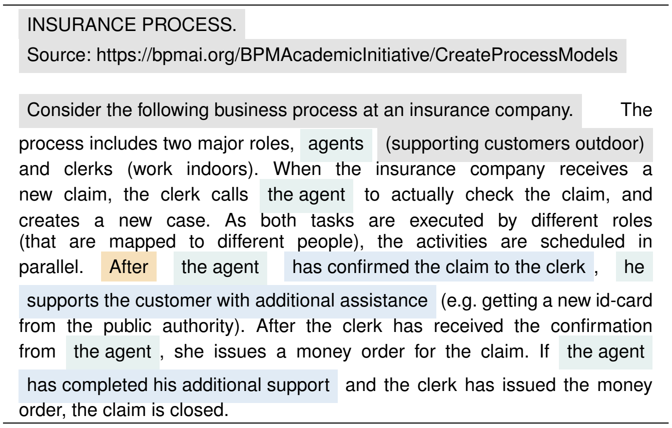
In this paper, we will concentrate on the connections between the Process Highlighter and the NLP module.

III. The Structure of a Highlighted Text

Fig. 2 showcases a short but representative example of a process description of an insurance process [12]:

The process highlighter supports four kind of markings:

- **Roles** (highlighted in green): They include titles, names, groups, departments and organisations. The same role may be expressed in singular and in plural forms. In



INSURANCE PROCESS.
Source: https://bpmai.org/BPMAcademicInitiative/CreateProcessModels

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.

Fig. 2. Insurance process description

some cases, a co-reference (e.g.: *"he"*) may be used to refer to previously defined roles (e.g.: *"the agent"*).

- **Activities** (highlighted in blue): represent arbitrary tasks or events. They can be written in active/passive voice, and in negated form. Moreover, they may be prone to *aliasing*, that is when two syntactically different tasks refer to the same activity, as in *"(the agent) supports the customer with additional assistance"* and *"(the agent) has completed his additional support"*.
- **Relations** (highlighted in yellow) they denote causal/temporal information, and link one or many activities together. In a text, relations are often given by modal verbs, as in *"shall"*, *"must"*, or *"can"*, or as adverbs, as in *"when"*, *"after"*, and *"if/then"* patterns. Relations impose constraints between activities, as in *"(the agent) has confirmed the claim to the clerk"* and *"supports the customer with additional assistance"*.
- **Non-processual information** (highlighted in grey): Texts may include information that do not refer to elements in the declarative process, for instance, the source where the document was retrieved.

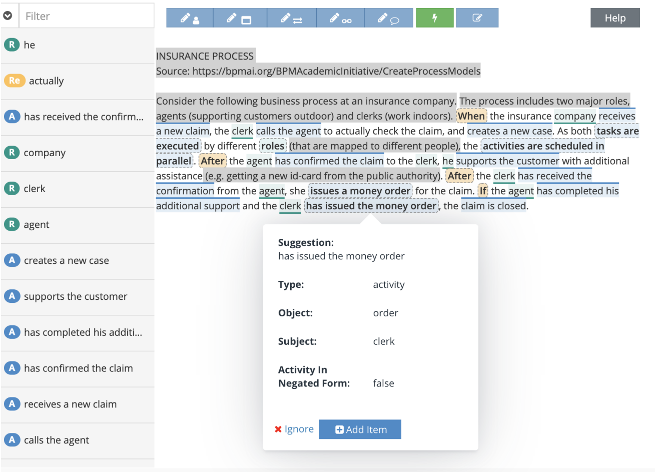
The first three markings existed in the original presentation of the highlighter. To create a process model from its description, the user had to manually search for each text pattern. This freedom allows for multiple marking styles. Some users mark whole paragraphs while others follow a strict marking pattern, highlighting only the most relevant words.

IV. Functionality

In this paper we have focus on facilitating the construction of a process model from a textual process description. For simplicity, we assume an empty process model for this example (this does not have to be the case in general). To start, the user to filters out (e.g.: use the comment functionality) all the non-processual information from the process description. We assume an empty process model for this example.

The modelling process considers the following steps:

1) Elicitation of process components from texts.



INSURANCE PROCESS
Source: https://bpmai.org/BPMAcademicInitiative/CreateProcessModels

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.

Suggestion:
has issued the money order

Type: activity

Object: order

Subject: clerk

Activity In Negated Form: false

Fig. 3. Visualisation of suggested highlights

2) Elicitation check (discard false-positives, complete the model with false-negatives).
3) Quality-control: enrich the process model to respect scenarios and preserve structural properties.
4) Maintain the alignments to reflect changes in the process model.

We proceed to describe each of the phases.

a) Elicitation of process components: The mapping from roles, activities, and relations can be done manually or semi-automatically. While manually identifying roles and activities is relatively simple in small descriptions, it might become time-consuming for lengthy process descriptions. This is noticeable even in a simple example as the one in Fig. 3. To preserve the correspondence, an engineer will need to mark six occurrences and three variants of *"agent"*, although semantically they correspond to the same role. Similarly happens with activities *"has completed additional support"*, and *"supports the customer with additional assistance"*. Moreover, our empirical observations [3], concluded that users tend to do all the modelling of relations in the modeller, partly because they find it difficult to find a proper language construct that describes relations in a process model.

The semi-automatic method makes use of the NLP module to perform extraction, visualisation and learning of roles, activities and constraints. Instead of performing model extraction, our method provides alignment suggestions. They allow the user to focus on specific text fragments that denote process elements. This, we hypothesise, will increase his/her understanding of the process model. Suggestions retrieved can be discarded, accepted, or merged with existing activities. These decisions are used by the highlighter as training data for the R-A-R dataset to improve future suggestions. Fig. 3 presents a screenshot of how suggestions for the text in Fig. 2 appear to the user.

b) Elicitation Check: A semi-automated mapping of process components to natural descriptions may potentially retrieve false positives and false negatives. False positives are inherent to the starting dataset used. Pre-trained and general purpose-datasets such as WordNet aim to cover most of everyday English and do not include domain-specific terminology for process descriptions. That is evidenced in our example, via the suggestion of *"roles"* as a role. False negatives appear due to the inability to suggest all relations in the model. NLP methods are particularly suited to capture syntax-directed constraints (e.g.: *"when"*, *"after"*), but not semantic rules (as in *"all previous activities should be done in parallel"*). To mitigate the impact of false positives, the suggestions for the highlighter compare both the suggestions from the Wordnet dataset with entries in the R-A-R dataset. Each accepted/rejected suggestion is stored in the dataset, allowing us to calculate confidence levels that suit the process domain. For false negatives, we require the user to manually add such constraints to the model. Such manual interventions are then used for training of the R-A-R dataset.

c) Quality Control & Documentation Maintenance: The workbench has a number of tools to check for correctness of the generated processes. First, the *dead-end analyzer* implements the checks for deadlock and livelock-freedom for DCR graphs in [15]. The second refers to semantic correctness. Test-driven development [16], [20] allows engineers to define scenarios describing valid and invalid process traces. Scenarios will be tested against the current model, suggesting whether some constraints are necessary (resp. missing).

V. Related work

Despite a raising interest in the alignment of process models and textual descriptions [1], [7], [8], [19], the authors are only aware of a single work exploring the alignment of natural language and declarative process models [1]. We mostly differ on the scope. While the objective of [1] is to generate models, we are aiming at supporting user creation of models. In addition, we differ in the target language used for extraction (Declare in [1], and DCR in our case).

VI. Concluding Remarks

This work reports initial efforts in optimising the process of process modelling for declarative processes. In particular, we have extended the DCR Process Highlighter with label, role and constraint discovery methods, using a range of NLP techniques. They answer to empirical observations that stressed the difficulties users face in aligning words in a text and their corresponding elements in a process model.

The process highlighter is in a mature stage, and it supports both academic and industrial users. The NLP module, and the automatic suggestions have been recently released in June 2019. Due to the short release time, we have not yet validated the advantages of automatic suggestions with bigger datasets, but in our current experiments we have been able to identify roles, activities and constraints that closely resemble those established by humans[3]. The tool is integrated in the industry offering of DCR solutions (`http:://dcrgraphs.net`), and it is available for free for non-commercial use. A screencast describing its use is available at `https://youtu.be/9N6ZEnO1FkA`.

Acknowledgments. Work supported by the Innovation Fund Denmark project EcoKnow.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] van der Aa, H., Ciccio, C.D., Leopold, H., Reijers, H.A.: Extracting declarative process models from natural language. In: CAiSE. LNCS, vol. 11483, pp. 365–382. Springer (2019)

[2] Agency for Digitisation: Vejledning om digitaliseringsklar lovgivning. Ministry of Finance, Denmark, `https://digst.dk/media/16953/vejledning_om_digitaliseringsklar_-lovgivning_maj_2018_tg.pdf`

[3] Andaloussi, A.A., Buch-Lorentsen, J., López, H.A., Slaats, T., Weber, B.: Exploring the modeling of declarative processes using a hybrid approach ((to appear) 2019), international Conference of Conceptual Modeling

[4] Bench-Capon, T.J.M., Coenen, F.P.: Isomorphism and legal knowledge based systems. Artificial Intelligence and Law 1(1), 65–86 (Mar 1992)

[5] Cashen, V.M., Leicht, K.L.: Role of the isolation effect in a formal educational setting. Journal of Educational Psychology 61(6, Pt.1), 484–486 (1970)

[6] 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)

[7] 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 (2017)

[8] Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: CAiSE. pp. 482–496. Springer (2011)

[9] Hildebrandt, T., Mukkamala, R.R.: Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. In: PLACES. EPTCS, vol. 69, pp. 59–73 (2010)

[10] Hunt, R.R.: The subtlety of distinctiveness: What von Restorff really did. Psychonomic Bulletin & Review 2(1), 105–112 (1995)

[11] Leopold, H., Mendling, J., Günther, O.: Learning from quality issues of bpmn models from industry. IEEE Software 33(4), 26–33 (2015)

[12] López, H.A., Debois, S., Hildebrandt, T.T., Marquard, M.: The process highlighter: From texts to declarative processes and back. In: BPM (Demos). CEUR Workshop Proceedings, vol. 2196, pp. 66–70. CEUR-WS.org (2018)

[13] Marquard, M., Shahzad, M., Slaats, T.: Web-based modelling and collaborative simulation of declarative processes. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM. pp. 209–225. Springer International Publishing, Cham (2015)

[14] Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)

[15] Mukkamala, R.R., Hildebrandt, T., Slaats, T.: Towards trustworthy adaptive case management with dynamic condition response graphs. In: EDOC. pp. 127–136. IEEE (2013)

[16] Slaats, T., Debois, S., Hildebrandt, T.T.: Open to change: A theory for iterative test-driven modelling. In: BPM. LNCS, vol. 11080, pp. 31–47. Springer (2018)

[17] Slaats, T., Mukkamala, R.R., Hildebrandt, T., Marquard, M.: Exformatics declarative case management workflows as dcr graphs. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM. pp. 339–354. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

[18] Strømsted, R., López, H.A., Debois, S., Marquard, M.: Dynamic evaluation forms using declarative modeling. In: van der Aalst et al., W.M.P. (ed.) BPM (Industry). CEUR Workshop Proceedings, vol. 2196, pp. 172–179. CEUR-WS.org (2018)

[19] Sànchez-Ferreres, J., van der Aa, H., Carmona, J., Padró, L.: Aligning textual and model-based process descriptions. Data & Knowledge Engineering (09 2018)

[20] Zugal, S., Pinggera, J., Weber, B.: Creating declarative process models using test driven modeling suite. In: CAiSE. pp. 16–32. Springer (2011)

[3] For the most updated benchmarks, see `https://wiki.dcrgraphs.net/highlighter-suggest`