Extracting Models from Spoken Text - A Research Proposal

Stefan Haag[1], Carlo Simon[1]

Abstract: Interviews between modeling and domain experts are often used as a source for model development which is an iterative and time-consuming procedure. The research plan aims to improve and automate this procedure which would enable domain experts to create models themselves and free time and mental capacity for modeling experts to concentrate on the process, not its model's implementation. The proposed system takes as its input (transcriptions of) spoken language in interview situations and builds models during the course of the interview while detecting ambiguities, syntactic or semantic errors, and providing feedback, alternatives or possibly corrections. The goal is a functional prototype which will be described and developed in a dissertation project.

Keywords: Process Extraction, Natural Language Processing, Conceptual Modeling, Petri Nets

1 Introduction

Process knowledge of employees or domain experts often lies hidden in process-unaware information systems, unstructured data, or established processes instead of being available in exact descriptions [vdAN08]. If suitable digital traces exist, processes can be mined [vdA16]. Otherwise, more traditional approaches of building process models are needed, like conducting interviews between modeling and domain experts. To simplify this iterative and time-consuming procedure we aim to answer the following research question:

Can an NLP-assisted system, that interacts like a modeling expert in an interview situation, support domain experts in developing sound (process, data, and organizational) models by speaking to the system, and without initial direct involvement of modeling experts?

The project follows the seven guidelines for Design Science Research according to [He04]. The proposed system and its components represent the *artifact*. Enabling domain experts to create models in absence of modeling experts exhibits the *problem's relevance*. The *evaluation* shows whether the system actually behaves comparably to a modeling expert. The *contribution* consists of solutions for the three topics described in the next paragraph and the system's implementation. *Rigor* is achieved by comparing the system's components to existing solutions and using current technologies that seemingly have not been used in this context before. The implementation constitutes a *search process* as different approaches will be examined for each of the components. *Communication* includes presentations to practitioners, researchers, and also students for single components, their implementation steps, and the final evaluated system.

[1] HS Worms, FB Informatik, Erenburgerstr. 19, 67549 Worms, Deutschland, {haag,simon}@hs-worms.de

Related work considers pre-processed data from textual descriptions (cf. [No20, FKM05]). However, ad hoc speech is often unstructured, incompletely phrased, shows grammatical errors, slips or filling words. Current NLP technologies like Transformers [Va17] that may aid in the *transcription* are unused as of now. The model is created while domain experts speak. This necessitates a *feedback loop* incorporating a *flaw detection* to inform users about issues and let them update it. Such an error handling has not yet been examined.

The remainder of the paper describes the project vision, and gives a first overview of the project's phases, an interim specification of the components, and related work.

2 Project Vision

The goal of this project is to establish a prototypical system that supports domain experts conducting process modeling tasks. Spoken language gets transcribed by third-party speech recognition while experts describe a process. A text-to-model solution extracts all readily available information from the transcript. This information is transferred into rich event logs (containing a maximum of available information) that in turn result in workflow nets by use of process mining techniques. Other model representations to be constructed and used for cross-checking are tables and ontologies. The representation is converted into a displayable model that is refreshed each time new or updated information can be integrated, thus, providing feedback to the users. A flaw detection checks the model for issues - be it ambiguity, grammar, model correctness and completeness, or any other kind of detectable flaw. Two consequences follow from such a detection. (1) The system attempts a correction, presents alternatives and recommendations, or describes the problem as is. The issue, the system's decision, and information about it are annotated in the model. (2) Users are called to action to either accept a recommendation, to correct the flaw, or to ignore it.

3 Project Phases, Components, and Related Work

The project as outlined in figure 1 directly touches three of *25 Challenges of Semantic Process Modeling* [MLP14]. A first key aspect relates to how to *Transform Text to Model (C13)*, i.e., how to derive process models from (transcriptions of spoken) natural language. The second key aspect is an evaluation that needs to *Verify Model Correctness (C14)* and *Validate Model Completeness (C15)* with respect to the semantics of the activity labels.

These two aspects can be partitioned into the working packages *(1)* - *(10)* spread over five phases. Each of these phases yields a component which combined constitute the prototype's goal: *Aid users to interactively create (process, data, and organizational) models from spoken natural text*. The implemented system is evaluated in a dedicated last phase.

Phase 1 covers *(1) textual input*. It deals with acquisition of spoken interviews, their transcription and pre-processing. Ideally, at least two people will describe the same process in order to learn about differences in their descriptions, and how to handle ambiguities.

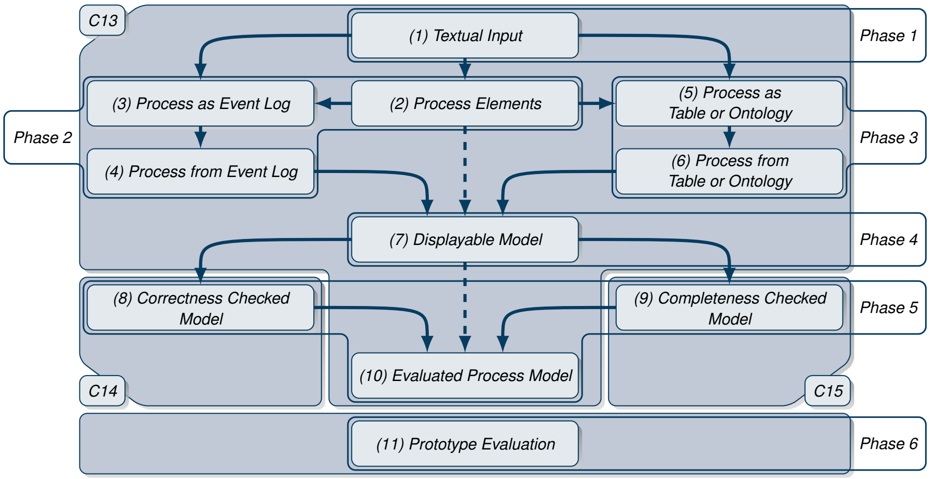

Fig. 1: Proposed Research Plan

Phase 2 provides workflow nets [vdAS11] and possible additional information about data and organizational structures. In order to enable further mathematical analysis and deep simulation, the processes will be formalized as high-level Petri nets [Re13, GL81]. This expands the possibilities to use the models in fields like forecasting, workflow management, or plant control [SHZ21, vdAS11, Zu80]. Hence, *(2) process elements* are extracted from the textual input to create *(3) rich event logs* that contain all information that can be obtained directly, but also derived or inquired in phase 1. Annotations about the latter are also included in the logs. Then, a model *(4) representation* is computed by use of Process Mining [vdA16]. Although process extraction from natural language is a field actively being researched on, it is still a complex task which often requires pre-processing of input data [FKM05, YBL10] or active user intervention [Ko05a, KHO11], yields incomplete processes [GKC07], or generally reveals highly relevant practical problems [FMP11].

Phase 3 reuses the process elements to describe processes in form of *(5) tables* that allow for structured process recording [KD10] and *(5) ontologies* which have an inherent ability to carry semantics [KO05b, RFP08]. To compare the thus established process models with those of phase 2, they also have to be converted into a compatible *(6) representation*.

Phase 4 focuses on how to create a *(7) displayable model* that needs to be enhanced by additional information and by incorporating user feedback. The use of frameworks like PM4Py [BvZvdA19] or an import into an existing (process) modeling software like the integrated management system Process-Simulation.Center [Si22] would be feasible options.

Phase 5 establishes the flaw detection. Evaluation of *(8) correctness* with respect to a given specification can be performed by, e.g., a logic of actions [Si01] or invariant analysis [La97].

Semantics are the basis for checking *(9) completeness*. The proposed workflow is iterative, thus, users check the semantics implicitly and continuously. Still, a formal instance has to be provided. Possibilities include using error patterns [GL11] or ontologies [TF07]. Also, verification tools like LoLA have been proposed [Sc00]. Every time the displayable is updated, the model gets checked for flaws, correctness, and completeness. If an issue is found, the users receive feedback about it including - if possible - a proposed correction or an alternative which they are free to accept or temporarily ignore. Rephrasing an already given but erroneous statement updates the model. In combination, a fully *(10) evaluated process model* is established.

Phase 6 does not contribute to the system itself but a *(11) prototype evaluation* is performed. This includes presentation of the rationales for selecting the chosen approaches. Also, practitioners will be questioned as to how the system behaves compared to modeling experts.

Closing Remark

This paper outlines quite a large vision the authors aim to realize. However, as the dissertation project is in its preparation phase, adjustments in scope and subject seem expectable. It may be feasible, e.g., to concentrate on one or several partial aspects depending on a possible research context. To this end, remarks and recommendations would be highly appreciated.

Bibliography

[BvZvdA19] Berti, A.; van Zelst, S.; van der Aalst, W. M. P.: Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science. In: International Conference on Process Mining (Aachen). CEUR-WS, Aachen, Germany, pp. 13–16, 2019.

[FKM05] Fliedl, G.; Kop, C.; Mayr, H. C.: From textual scenarios to a conceptual schema. Data & Knowledge Engineering, 55(1):20–37, 2005.

[FMP11] Friedrich, F.; Mendling, J.; Puhlmann, F.: Process Model Generation from Natural Language Text. In: CAiSE. Springer, Berlin, Heidelberg, Germany, pp. 482–496, 2011.

[GKC07] Ghose, A.; Koliadis, G.; Chueng, A.: Process Discovery from Model and Text Artefacts. In: Congress on Services. IEEE CPS, Washington, DC, USA, pp. 167–174, 2007.

[GL81] Genrich, H. J.; Lautenbach, K.: System Modelling with High-Level Petri Nets. Theoretical Computer Science, 13(1):109–135, 1981.

[GL11] Gruhn, V.; Laue, R.: Detecting Common Errors in Event-Driven Process Chains by Label Analysis. EMISA, 6(1):3–15, 2011.

[He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design Science in Information Systems Research. MIS Q., 28(1):75–105, mar 2004.

[KD10] Krumnow, S.; Decker, G.: A concept for spreadsheet-based process modeling. In: International Workshop on Business Process Modeling Notation. Springer, Berlin, Heidelberg, Germany, pp. 63–77, 2010.

[KHO11] Koschmider, A.; Hornung, T.; Oberweis, A.: Recommendation-based editor for business process modeling. Data & Knowledge Engineering, 70(6):483–503, 2011.

[Ko05a] Kop, C.; Vöhringer, J.; Hölbling, M.; Horn, T.; Mayr, H. C.; Irrasch, C.: Tool Supported Extraction of Behavior Models. In: Information Systems Technolopgy and its Applications. GI, Bonn, Germany, pp. 114–123, 2005.

[KO05b] Koschmider, A.; Oberweis, A.: Ontology Based Business Process Description. In: Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability. CEUR Workshop Proceedings, Aachen, Germany, 2005.

[La97] Lautenbach, K.: Action logical correctness proving. In: Symposium on Mathematical Modelling. ARGESIM, Wien, Austria, pp. 141–148, 1997.

[MLP14] Mendling, J.; Leopold, H.; Pittke, F.: 25 Challenges of Semantic Process Modeling. International Journal of Information Systems and Software Engineering for Big Companies, 1(1):78–94, 2014.

[No20] Nolte, F. R.: Text to Process Model: Automating Process Model Creation form Text. PhD thesis, WWU Münster, 2020.

[Re13] Reisig, W.: Understanding Petri Nets. Springer, Berlin, Heidelberg, Germany, 2013.

[RFP08] Rebstock, M.; Fengel, J.; Paulheim, H.: Ontologies-Based Business Integration. Springer, Berlin, Heidelberg, Germany, 2008.

[Sc00] Schmidt, G.: Grundlagen der Aufbauorganisation. Verlag Dr. Götz Schmidt, Gießen, 4 edition, 2000.

[SHZ21] Simon, C.; Haag, S.; Zakfeld, L.: Requirements for Highly Integrated Management Systems - Simulation Expands Past-oriented Documentation to Future-oriented Optimization. In: International Conference on Advances in System Simulation (Barcelona, ESP). ThinkMind, Wilmington, DE, USA, pp. 58–64, 2021.

[Si01] Simon, C.: A Logic of Actions and Its Application to the Development of Programmable Controllers. PhD thesis, Universität Koblenz-Landau, 2001.

[Si22] Simon, C.: , Group for Applied Process Simulation. `https://www.hs-worms.de/en/gaps/` (last accessed 01.05.2022), 2022.

[TF07] Thomas, O.; Fellmann, M.: Semantic business process management: Ontology-based process modeling using event-driven process chains. International Journal of Interoperability in Business Information Systems, 2:29–44, 2007.

[Va17] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Kaiser, A. N. Gomez L.; Polosukhin, I.: Attention is All you Need. In: 31st Conference on Neural Information Processing Systems. Curran Associates, Inc., Red Hook, NY, USA, 2017.

[vdA16] van der Aalst, W. M. P.: Process Mining. Springer, Berlin, Heidelberg, Germany, 2 edition, 2016.

[vdAN08] van der Aalst, W. M. P.; Nikolov, A.: Mining E-Mail Messages: Uncovering Interaction Patterns and Processes using E-Mail Logs. International Journal of Intelligent Information Technologies, 4(3):27–45, 2008.

[vdAS11] van der Aalst, W. M. P.; Stahl, C.: Modeling Business Processes - A Petri Net-Oriented Approach. The MIT Press, Cambridge, 2011.

[YBL10] Yue, T.; Briand, L. C.; Labiche, Y.: An Automated Approach to Transform Use Cases into Activity Diagrams. In: European Conference on Modelling Foundations and Applications. Springer, Berlin, Heidelberg, Germany, pp. 337–353, 2010.

[Zu80] Zuse, K.: Petri-Netze aus der Sicht des Ingenieurs. Vieweg+Teubner, Wiesbaden, Germany, 1980.