EduClare – An Intelligent Tutoring Chatbot for Teaching Declarative Process Modeling

Sabine Nagel[1,*], Anna Wolters[1], Dennis M. Riehle[1] and Patrick Delfmann[1]

[1]*University of Koblenz, Universitätsstr. 1, 56070 Koblenz, Germany*

Abstract

In this work, we introduce EduClare, a web-based intelligent tutoring chatbot to support Declarative Process Model (DPM) education in the modeling language *Declare*. This innovative hybrid chatbot combines principles from intelligent tutoring systems and large language models to create an interactive and user-friendly educational tool. In addition to explaining fundamental concepts of *Declare*, the chatbot generates an arbitrary number of tasks with increasing difficulty levels, covering basic reasoning, model execution, and active modeling of declarative specifications, with a focus on constraint interplay and inconsistency. Users benefit from instant answer validation, feedback, and the ability to ask for clarification, receiving customized guidance throughout their entire learning process.

Keywords

Declare, Declarative Process Models, Education, Intelligent Tutoring Systems, Chatbot

Metadata description	Value
Tool name	EduClare
Current version	1.0
Legal code license	CC 4.0 BY-NC-ND
Languages, tools and services used	Java, JavaScript, LangChain4j, gpt-4o
Supported operating environment	N/A
Download/Demo URL	https://educlare.de/
Documentation URL	https://uni-ko.de/educlare-git
Source code repository	https://uni-ko.de/educlare-git
Screencast video	https://uni-ko.de/educlare-screencast

1. Introduction and Related Work

Declarative Process Models (DPMs) offer a flexible approach to process modeling by specifying a set of constraints that implicitly define process behavior rather than explicitly modeling a fixed sequence of activities [1]. However, the implicit nature of DPMs introduces significant challenges in comprehension and application [2], which stresses the need for effective educational tools. Existing approaches, such as the one developed by De Smedt et al. [3] primarily focus on

ICPM 2024 Tool Demonstration Track, October 14-18, 2024, Kongens Lyngby, Denmark

*Corresponding author.

snagel@uni-koblenz.de (S. Nagel); awolters@uni-koblenz.de (A. Wolters); riehle@uni-koblenz.de (D. M. Riehle); delfmann@uni-koblenz.de (P. Delfmann)

0000-0003-4838-8246 (S. Nagel); 0000-0002-4075-5737 (A. Wolters); 0000-0002-5071-2589 (D. M. Riehle); 0000-0003-4441-0311 (P. Delfmann)

exploring hidden dependencies within DPMs. More recently, Nagel and Delfmann [4] proposed an e-learning framework designed to support learners in gradually becoming familiar with *Declare*. This framework addresses the common challenges identified in understanding *Declare* by offering a structured set of tasks that cover various aspects of the modeling language. However, there remains a significant gap in the availability of interactive, technology-enhanced educational tools for teaching *Declare*.

Within the field of technology-enhanced education, Pedagogical Conversational Agents (PCAs) [5] are widely used. PCAs, which are conversational agents (CAs) applied in educational contexts, offer an interactive learning experience that aligns with the ICAP framework, emphasizing the importance of active engagement in learning [6]. Recent advancements in CAs, particularly those using Large Language Models (LLMs), have opened new possibilities for increasing the interactivity and effectiveness of educational tools. While CAs have been explored in the context of business process modeling, their application to declarative process modeling has been relatively limited. Notable exceptions include Declo, a chatbot introduced by Alman et al. [7] for defining DPMs using natural language, and the CA developed by Fontenla-Seco et al. [8] for supporting declarative process mining. Still, the development of tools for DPM education remains unexplored.

To address this gap, we present EduClare, an intelligent tutoring chatbot designed to teach the declarative modeling language *Declare*. Our solution applies the latest advances in CA research and provides a structured, task-based learning experience that adapts to the learner's progress. The remainder of this paper is structured as follows. In Section 2 we describe the application, its functionality, and architecture in more detail. In Section 3 we explain usage and report on the results of a first usability study to discuss the tool's maturity. We conclude and discuss directions for future work in Section 4.

2. Tool Description

2.1. Functionality

EduClare is a comprehensive educational tool designed to support learning *Declare* through an interactive and personalized experience. The tutoring chatbot guides users through three main phases: (1) introducing the tutoring and explaining relevant *Declare* concepts, (2) providing tasks of different types and with increasing difficulty as part of the task-based learning and (3) interactively answering questions to provide continuous support. The task-based learning phase builds on previous work [4], where we developed the foundation for an e-learning concept for teaching *Declare* and provided a comprehensive collection of tasks covering concepts such as constraint definitions, model behavior, and inconsistency within DPMs. In the following, we will describe the tool's functionality in more detail.

Introduction In this phase, the chatbot assesses the user's prior experience with *Declare*. New users are provided with a detailed introduction, while more experienced users can choose to skip this section or only refresh their prior knowledge. This phase is interactive, with the chatbot regularly prompting users to ask questions to ensure that uncertainties are immediately addressed.

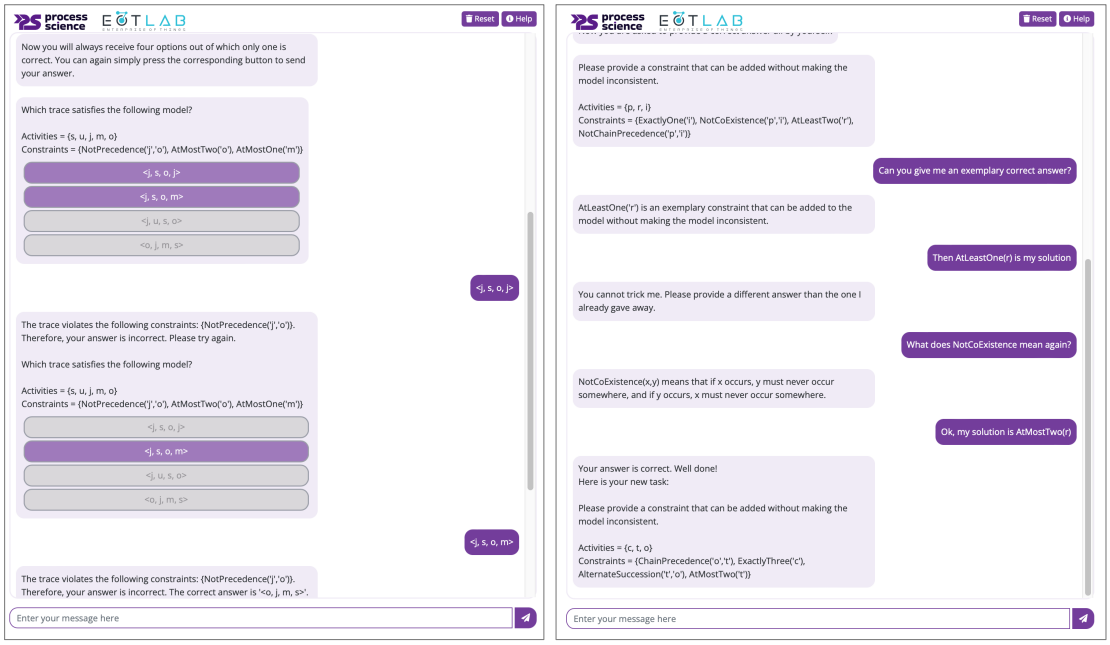

Figure 1: Exemplary Chatbot Conversations

Task Generation The chatbot can dynamically generate an arbitrary number of tasks with varying difficulty levels to help users improve and test their understanding of *Declare*. This includes (1) *inference* tasks where users must infer an activity based on a provided model and corresponding statement, (2) *execution* tasks, where users are required to provide valid/invalid traces for a model and vice versa, (3) *inconsistency* tasks where users are provided with a consistent model and either have to maintain consistency or cause inconsistency by adding a constraint and (4) *redundancy* tasks where users are asked to add or remove constraints to/from a model without changing its behavior. Furthermore, EduClare currently supports three task types (true/false, choice and input) to reinforce comprehension through both recognition and recall.

Instant Answer Validation and Feedback Any answers given by the user are evaluated on the fly and the chatbot provides instant feedback on each task, indicating whether the answer was correct and providing explanations when mistakes were made (cf. Figure 1).

Customized Guidance and Continuous Support Users can interrupt the task flow at any time to interact with the chatbot and receive customized guidance, as illustrated in Figure 1. Here, possibilities include asking for a new task, an exemplary answer or hint, and asking any other task-related questions, such as the definition of *Declare* templates used in a task. The chatbot's ability to maintain conversation history allows it to provide contextually relevant answers and continue discussions seamlessly.

Learning Progression The chatbot supports learning progression as the tutoring process is designed based on levels, allowing users to gradually build their skills. Each level is

designed to incrementally increase in difficulty (beginner, intermediate, and expert) and interaction complexity (true/false, choice, input). Eligibility to advance to a new level is based on a pre-defined number of consecutive correct answers, but users can decide to keep practicing at any point, which enables continuous learning.

2.2. Architecture

EduClare is implemented as a Java SpringBoot web application. To integrate LLMs, we used the LangChain4j[1] framework and selected OpenAI's GPT-4o model as the underlying LLM. We will now describe the architecture and design of our application (cf. Figure 2) in more detail.

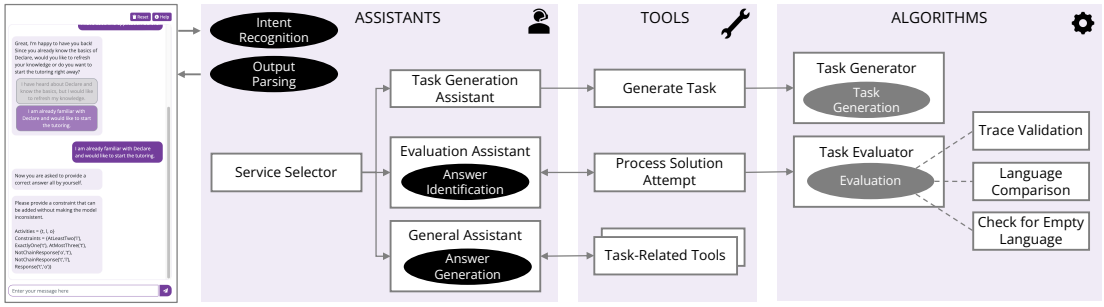

Figure 2: Overview of Chatbot Architecture

Frontend The frontend was developed using JavaScript, HTML and CSS. It allows the user to interact with the tutoring chatbot, either in textual form via the input field or by selecting one of the provided options. The chatbot additionally allows customized interactions, such as allowing users to input shorthand commands or full sentences, which are then correctly interpreted by the system. This allows for more flexibility and accessibility.

Assistants Assistants manage the interaction between users and the system's underlying tools and algorithms. The *Service Selector* is the first to process user messages by determining their intent, i.e., whether it is requesting a new task, responding to a task, or asking a general question. Based on this intent, the *Service Selector* directs the request to the appropriate assistant. The *Task Generation Assistant* handles requests for new tasks by invoking the *Task Generator Tool*, formatting the output for user display. The *Task Evaluation Assistant* manages task responses by identifying and evaluating user-provided artifacts using specific algorithms. The *General Assistant* addresses all other queries, such as answering general questions or providing exemplary answers, relying on predefined tools or the language model for more complex responses. To increase the validity of the answer to any general question, a system prompt was designed that provides the LLM with general information on *Declare* as well as further instructions on how to communicate with the users. Therefore, any general question that is not covered by a tool is directly

[1] https://docs.langchain4j.dev

processed by the prompted LLM. Each assistant ensures user interactions are efficiently processed and appropriately handled.

Tools The tools within our tutoring chatbot architecture perform specific tasks as directed by the assistants. They are defined by natural language descriptions that enable assistants to select and invoke the correct function for user requests. For example, the *Task Generator Tool* creates new tasks based on user context, while the *Task Evaluation Tool* validates user responses by comparing them with expected outcomes or calling the corresponding evaluation algorithm. This ensures efficient system performance by reducing reliance on the LLM for straightforward tasks. Lastly, tools accessed by the *General Assistant* handle routine requests, like providing exemplary answers or repeating tasks.

Algorithms We developed various Java algorithms to generate and evaluate tasks. For task generation, we created specific algorithms tailored to each task category, which enables the chatbot to create arbitrary amounts of tasks covering a variety of *Declare*-related concepts with progressively increased difficulty. To generate and assess both correct and incorrect answers, we employed additional algorithms using deterministic finite automata (DFA) via dk-brics-automaton[2]. By transforming each *Declare* model into a corresponding DFA, we validated traces against models, compared languages of different models (e.g., to check for redundant constraints), and ensured model consistency (as inconsistent models have an empty language).

3. Usage and Maturity

A demo version of EduClare is available here[3], a local version can be obtained from our Git repository and requires an own OpenAI API key.

To gain first insights into the usability of our initial concept, we conducted a small study with four PhD students that had varying levels of prior experience with *Declare*. It comprised of an exploration phase, where the participants were instructed to try out the chatbot on their own and explore its functionality, usability and intuitiveness, while voicing their thoughts . To gain additional insights, we then asked some follow-up questions, focusing on encountered difficulties, suggestions for improvement or extension and general feedback. Overall, the tutoring chatbot, its innovative approach and overall usability was perceived positively. Users appreciated the chatbot's ability to make *Declare* more accessible, with two participants mentioning that they found the interaction with the chatbot more engaging and enjoyable compared to conventional methods, such as textbooks or literature. The chatbot's guidance throughout the learning process was also well-received, with one participant mentioning an increased curiosity to explore beyond the necessary tasks, indicating a high level of user engagement. Additionally, some feedback suggested areas for improvement, such as shortening introductory messages and incorporating visual aids, as well as further customization based on individual learning styles. Furthermore, suggestions were made for introducing gamification elements and adjusting task difficulty based on user experience. The feedback also indicated interest in including visual representations of *Declare*, which could be a direction for future development.

[2] https://www.brics.dk/automaton/
[3] https://educlare.de/

4. Conclusion

In this work, we developed an intelligent tutoring chatbot with a hybrid architecture to support *Declare* education. Our approach integrates a structured learning path with task-based learning and continuous interactive support, tailored to different experience levels. The LLM-based chatbot effectively generates and evaluates tasks, while assisting users in learning *Declare*. Future work will address current limitations, such as a risk of producing incorrect outputs, by iteratively improving and expanding the chatbot's functionality. This includes visual DPM representations, multilingual support, and personalized learning experiences. That way, we aim to not only enhance Business Process Management (BPM) education but also provide a valuable tool for process mining researchers.

Acknowledgments

This paper was funded by the Deutsche Forschungsgemeinschaft (grant number DE 1983/9-3) and the Federal Ministry of Education and Research (BMBF) under grant number 16DHBKI039.

References

[1] C. Di Ciccio, M. Montali, Declarative Process Specifications: Reasoning, Discovery, Monitoring, in: Process Mining Handbook, volume 448, Springer Int. Publ., 2022, pp. 108–152.

[2] C. Haisjackl, I. Barba, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, B. Weber, Understanding Declare models: strategies, pitfalls, empirical results, Software & Systems Modeling 15 (2016) 325–352.

[3] J. De Smedt, J. De Weerdt, E. Serral, J. Vanthienen, Gamification of Declarative Process Models for Learning and Model Verification, in: Business Process Management Workshops, volume 256, Springer International Publishing, Cham, 2016, pp. 432–443.

[4] S. Nagel, P. Delfmann, Towards an E-Learning Approach for Declarative Process Modeling, in: Proceedings of the BPM 2024 Blockchain/RPA/CEE/Educators/Industry Forum, Krakow, Poland, 2024.

[5] F. Weber, D. Rüttimann, T. Wambsganss, M. Söllner, Pedagogical Agents for Interactive Learning: A Taxonomy of Conversational Agents in Education, in: ICIS Proceedings, 2021.

[6] M. T. H. Chi, R. Wylie, The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes, Educational Psychologist 49 (2014) 219–243.

[7] A. Alman, K. J. Balder, F. M. Maggi, Declo: A Chatbot for User-friendly Specification of Declarative Process Models, in: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2020, 2020, pp. 122–126.

[8] Y. Fontenla-Seco, S. Winkler, A. Gianola, M. Montali, M. Lama, A. Bugarín-Diz, The Droid You're Looking For: C-4PM, a Conversational Agent for Declarative Process Mining, in: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM 2023, Utrecht, Netherlands, 2023.