The Action Engine –
Turning Process Insights into Action

Peyman Badakhshan, German Bernhart, Jerome Geyer-Klingeberg*,
Janina Nakladal, Steffen Schenk, Thomas Vogelgesang
Celonis, Munich, Germany
*corresponding author: j.geyerklingeberg@celonis.com

***Abstract*—The Action Engine from Celonis is a new web application that translates findings from automatic discovery and rule-based process analysis into recommendations for operational support during process execution. The engine continuously analyzes data across different systems and processes, communicates personalized recommendations for process-related actions to human users or the digital workforce, and directly executes actions in the source system. This approach contributes to the important field of operational process support by operationalizing insights created through process analysis. The presented capabilities are especially valuable for active process mining users requiring fast suggestions where to take action for improvement of process performance during execution.**

Keywords: Process Mining, Operational Support, Recommendation, Action Engine, Recommendation System

I. Introduction

Process mining provides objective and fact-based insights derived from event logs stored in common IT systems. It enables the user to analyze and improve business processes by finding answers for both compliance-related and performance-related questions [1].

Various business processes are executed continuously in organizations, which usually run on various operational systems such as SAP, Salesforce, Service Now, Microsoft Dynamics, etc. Process mining tools empower organizations to bring all these systems together by building an integrating layer on top of various distributed systems, which allows to reconstruct as-is flows from end-to-end [1]. Process mining has proved to be an innovative and efficient method to discover, analyze, and predict business processes' behavior [2], [3]. It enables automated process model discovery and in consequence provides in-depth insights into business processes. Despite analyzing historical data and existing models, process mining is also considered as a tool that enables predicition of future process flows and operational decision support. For instance, decision in processes can be facilitated by predicting the completion time of running instances or forecasting of the next process activity that will be executed [4], [5]. Accordingly, a prediction engine is playing an important role in moving process mining tools to a new level of productivity and to extend the field of process enhancement. Figure 1 describes the prediction approach.

Building on this prediction approach, Celonis developed a new application called the *Action Engine*. On a high-level, this application aims to operationalize insights gained

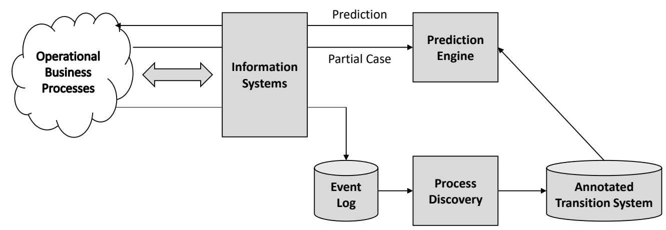

Fig. 1. Process mining prediction approach (adopted from [5])

from using process mining by providing predictions as well as recommending necessary actions to be taken to improve process performance. These findings are generated using process mining capabilities, including all the relevant information necessary to make a decision, which then allows taking action directly out of the Action Engine (e.g. execute a task in the source system or trigger a bot). The Action Engine also assists in identifying aspects of high priority where an action is needed and where action has a high impact on pre-defined process KPIs (Key Performance Indicators).

The Action Engine is a catalyst for continuous change in organizations and an enabler for business transformation through smart actions for a human and virtual workforce. As it guides them to make and execute optimal decisions. "Optimal" means that decisions not necessarily optimize one single transaction or one order only, but optimize a certain target process performance KPI across transactions (like throughput time, fulfillment rates, process costs etc.). As an example, if resources are sparse, it can be optimal to delay one customer order to prioritize a VIP order.

There are three main phases to accomplish the aim of operationalizing findings and sending recommendations to users in order to help them executing actions right away. First, the Action Engine autonomously and continuously analyzes an organization based on operational and historical data across various systems and processes. Second, it communicates the detected improvement opportunities to users – just-in-time in a personalized way. Third, it proposes action to a predefined user/user-group or actively executes this action in the source system (e.g. by triggering a bot or starting a workflow).

Action Engine therefore also helps to define the next-best activity in every business process. For example, in a Purchase-to-Pay process, Action Engine shows every purchaser what to do next for his/her purchase orders to increase productivity, minimize purchasing spend, and reduce working capital. For instance, based on the past orders and the master data, the Action Engine indicates where payment terms can be optimized. In an Order-to-Cash process, Action Engine shows every order manager what to do next for his/her customer orders to increase productivity, maximize speed, and deliver the order in full and on-time. For instance, based on actual cycle times, the Action Engine indicates credit checks to prioritize in order to meet the delivery date.

While the classical process analysis is often carried out by process owners, the Action Engine enables the simple and intuitive use of process mining for all groups of employees in an organization whose daily work is made easier by means of operative recommendations for action. The analysis capabilities can thus be scaled within the company.

In the remainder of this paper, we first introduce the Action Engine in more detail by explaining its approach, followed by a brief overview of the functionalities and main features. Then we provide insights on the maturity of the tool by describing use cases. At the end, we provide the screencast along with the demo access.

II. Innovation and Approach

The first step of the Action Engine is to operationalize the process insights in order to create intelligent predictions and recommendations as well as to trigger off specific activities. The recommendations are always personalized and assigned to a responsible employee. Thus, the capabilities of the Action Engine go beyond simple alerts and support the performance of actions equally by providing all necessary process information and the direct link to workflows. Figure 2 depicts the general concept of the Action Engine.

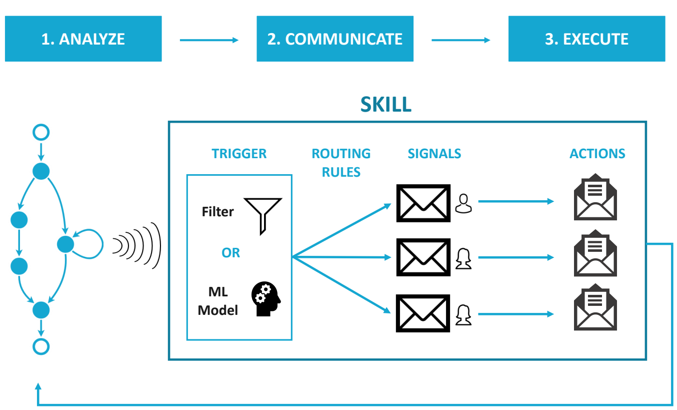

Fig. 2. Overview of Action Engine concept

In the first step, process data is extracted via event collection methods across various IT systems. In the second step, a skill is set up. A skill is defined as an object with the following characteristics: trigger, routing rules, signals, and actions. A trigger is a mechanism that activates the signals. A trigger can be defined via regular process mining filters (data model or table filters) or based on machine learning models. The different ways to define triggers are described in the next sections. A trigger initiates signals to be sent to users (human or bot). Routing rules define the recipient and the assignee for a task. The signal itself is the representation of a skill, which the end users interact with. Actions are recommended tasks the machine suggest to the end user. These actions are (mostly) executed in the operational source systems. There are three general ways to define triggers:

A. *Rule-based Triggers*

Using the process query language (PQL) developed by Celonis, rules can be derived that define recommended actions to avoid process inefficiencies in the future. The Action Engine continuously examines the process data in the background with the aid of this set of rules and forwards the resulting recommendations for action directly to the responsible end user. An exemplary use case are late checks of payment blocks for external invoices in paper form. These lead to a loss of cash discounts. Finding this inefficiency implies a rule to check such payment blocks earlier. The rule can be defined via PQL in the same way as an Event-Condition-Action formula. If the set of rules is in force, the end user receives a notification (by email, Slack or any other communication channel) in the case of an external invoice (as a condition and trigger of the rule) to check the payment block on time.

B. *Triggers by Classification of Comparable Processes*

Processes are classified into similar processes by attribute decisions, such as the material group or country where a purchase order has been placed, and advanced machine learning classifiers, especially Random Forest and Naive Bayes. Based on the analysis of the processes within a comparison group, predictions about the further course of the process and recommendations for current process flows are made. For example, the throughput time to delivery can be predicted for a current order on the basis of similar orders in the past (e.g. using a quantile of empirical throughput times). If the delivery is expected after the requested delivery date, early intervention is recommended to prevent late delivery. Free text orders are another use case. Based on similar orders in the past, a supplier can be recommended from whom similar/same articles were ordered. This reduces the research effort of the buyer. Various matching algorithms and also Natural Language Processing (NLP) is used for classification and the analysis of text strings, e.g. in the order text stored in the procurement system.

C. *Triggers by Predictive Process Monitoring*

Classified inefficiencies are predicted with the help of machine learning algorithms, especially Random Forest and Neural Networks. Deliveries are made too late, orders are cancelled, or invoices are not paid. To prevent this, a rule is created and a recommendation for action is initiated. In addition, the next activity of a process can be predicted, including the time it will take place, for example a delivery to the customer. On this basis, recommendations for optimization can be derived if, for example, the delivery arrives too late and this can be prevented.

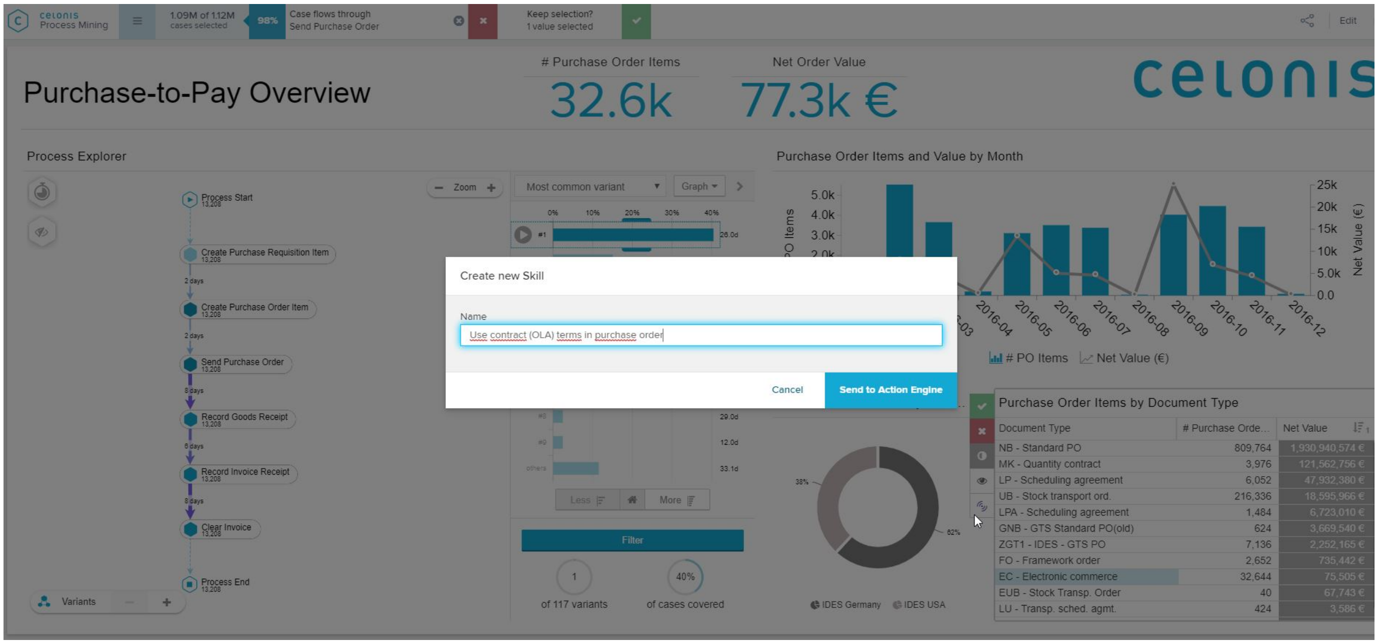

Fig. 3. Creating an Action Engine Skill

III. Main Features of the Application

For the case study, we use a demo data set of the Purchase-to-Pay (P2P) process obtained from a SAP ERP system. The features shown here can easily be applied to any other process. The Action Engine starts with the setup of a certain skill. A skill covers the configuration of triggers, routing rules, signals and actions. Triggers are set by certain filters (data model or table filter), classifier and machine learning methods applied on the process data that extract only the relevant incidents that should be addressed. Figure 3 shows how to set up a trigger directly from the insights gained in the analysis. By filtering within the analysis components, the process data is drilled down. By clicking on the little signal symbol on the left side of the bottom right table, the filter is automatically transformed into the respective PQL statement that is needed as a trigger. The data model filter selects specific contract terms for a purchase order to be sent out. Configuring the trigger and the subsequent signal that is sent out, the user defines which actions are generated for these signals and how these signals are sent out (routing rules). Once a skill is set up, the capturing, interpreting and communicating steps are executed automatically in a continuous manner. The end user finds his or her personal signal list depicting the purchase orders he or she should investigate and start an action.

There are various use cases in P2P. As an example, Peter as a purchaser spots contract terms that can be exploited. The Action Engine looks for similar purchase orders in the past and existing applicable contracts, and highlights if purchasing conditions can be improved. As a second example, the Action Engine indicates to Peter which purchase requests to prioritize in order to fulfill an important customer order. By analyzing historical cycle times for purchase request release and material procurement, Action Engine knows when to step into play to still meet the target customer delivery date. By working across processes, the application helps the workforce to work consistently towards customer satisfaction.

By clicking on an improvement opportunity, the user sees more detailed information (see Figure 4). For example, Peter recognizes that given his order quantity, the contract offers a 5% discount. Besides, he finds more information on the purchase order items corresponding to the purchase order in concern. The displayed information is fully flexible and can be adjusted easily to incorporate all context-relevant information necessary for Peter to make his decision.

To initiate an action, Peter can just click on "Add Contract" and is guided to the source system of the transaction (here: SAP). The appropriate SAP transaction is already executed and pre-filled with the right parameters. Thus, it is only one click to add the contract terms and realize the improvement opportunity. The important message here is that Action Engine triggers an arbitrary execution engine, such as the SAP system, and forwards all necessary parameters of the purchase order to execute the action. There is no need to manually collect the information. Having taken action, Peter can set the status of the item to "In Progress" to indicate that he has taken care of it. Once the item is resolved in the operational system, i.e. the contract terms are added successfully, the status is updated to "Resolved" automatically and the item vanishes from the list.

IV. Maturity and Case Studies

The Action Engine is an application that is available in the Celonis Intelligent Business Cloud since late 2018[1]. Currently

[1]https://www.celonis.com/press/celonis-intelligent-business-cloud-launched-as-first-saas-platform-for-supporting-business-transformation/

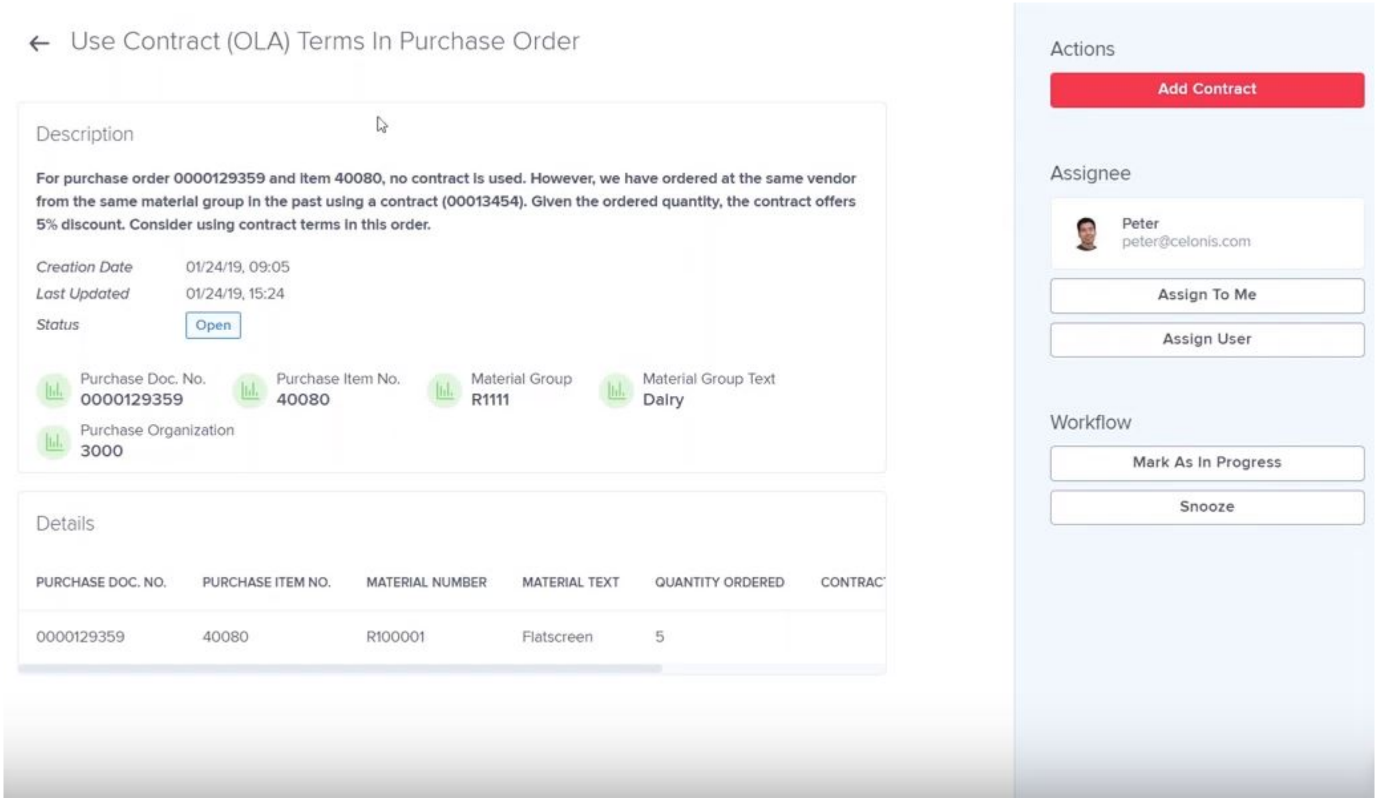

Fig. 4. Action window

it is used by more than 1,000 users across 10 different companies in various industries.

A first reference is Schukat Electronic, a German-based medium-sized wholesale distributor of electronic components with a workforce of 200 employees. Schukat's most important competitive advantage lies in a 24-hour delivery promise. Every purchase manager and every order manager has access to the Action Engine and receives signals for proactive action to avoid delayed customer orders on a daily basis. The Action Engine informs the order processing staff in real-time about the necessary measures to ensure fast deliveries already during process executions and also triggers SAP workflows. With this application, Schukat increased its on-time shipment of express orders to 100% and thus the Action Engine is a decisive success factor for competitive advantage.

A second reference is the Dutch publisher Elsevier. They apply the Action Engine for the submission-to-publication process. More than 300 journal managers have access to the application and receive notifications about where to take action to avoid delays in the review and publication process.

V. Screencast and Demo License

A screencast[2] explains the main features of the Action Engine. A free demo access and training material for the Action Engine is offered to the academic users via the Celonis Academic Alliance[3]. Commercial demos are also available via request[4].

Acknowledgment

All authors contributed equally to this paper. The list of authors is structured in an alphabetical order. The authors would like to thank Schukat Electronic for their cooperation on the development of the Celonis Action Engine.

References

[1] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[2] F. Veit, J. Geyer-Klingeberg, J. Madrzak, M. Haug, and J. Thomson, "The proactive insights engine: Process mining meets machine learning and artificial intelligence," in *Proceedings of the BPM Demo Track and BPM Dissertation Award*, R. C. et. al, Ed., 2017. [Online]. Available: http://ceur-ws.org/Vol-1920/BPM_2017_paper_192.pdf

[3] J. Geyer-Klingeberg, J. Nakladal, F. Baldauf, and F. Veit, "Process mining and robotic process automation: A perfect match," in *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018*, ser. CEUR Workshop Proceedings, W. M. P. van der Aalst et. al, Ed., vol. 2196. CEUR-WS.org, 2018, pp. 124–131. [Online]. Available: http://ceur-ws.org/Vol-2196/BPM_2018_paper_28.pdf

[4] W. M. P. van der Aalst, M. H. Schonenberg, and M. Song, "Time prediction based on process mining," *Inf. Syst.*, vol. 36, no. 2, pp. 450–475, 2011. [Online]. Available: https://doi.org/10.1016/j.is.2010.09.001

[5] W. M. P. van der Aalst, M. Pesic, and M. Song, "Beyond process mining: From the past to present and future," in *Proceedings of CAiSE 2010*, ser. Lecture Notes in Computer Science, B. Pernici, Ed., vol. 6051. Springer, 2010, pp. 38–52. [Online]. Available: https://doi.org/10.1007/978-3-642-13094-6_5

[2] http://bit.ly/2UTMqmg

[3] https://www.celonis.com/academic-alliance

[4] https://www.celonis.com/try-or-buy/