



# Applications of Automated Planning for Business Process Management

Andrea Marrella<sup>1</sup>(✉) and Tathagata Chakraborti<sup>2</sup>

<sup>1</sup> Sapienza Università di Roma, Rome, Italy  
marrella@diag.uniroma1.it

<sup>2</sup> IBM Research AI, Cambridge, MA, USA  
tchakra2@ibm.com

**Abstract.** This is a brief summary of the applications of automated planning in the field of Business Process Management (BPM); and accompanies a tutorial with the same theme at the 19th International Conference on Business Process Management (BPM 2021). We hope that this report is able to quickly onboard newcomers into this field with a broad overview of the associated challenges and opportunities, as well as provide established practitioners in the field some new food for thought in terms of the state-of-the-art and the evolving nature of these problems.

## 1 Why Automated Planning for Business Processes?

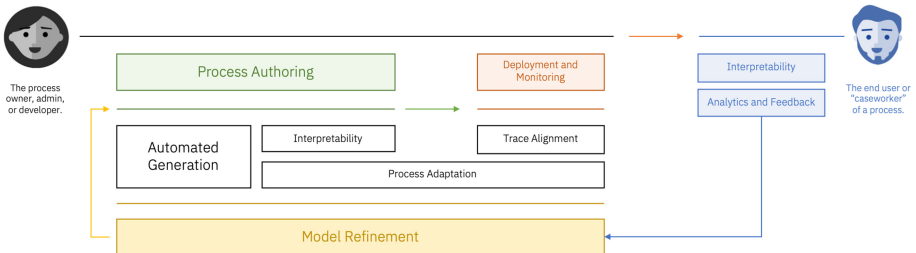
Automated planning deals with sequential decision-making of autonomous or semi-autonomous systems. Typically, a planning task takes as input a model of how the world works and a description of the task to be solved in that world; and produces as output a sequence of steps (plan) or a mapping from a world state to an action (policy). The representation of the world and task knowledge determines the flavor of planning. In this paper, we will focus on the subcategory “classical planning” or planning compilable to its classical form, where a goal state must be reached from a fully known initial state by applying planning steps having deterministic effects [9].

*The discipline of business process management (BPM) deals with the discovery, modeling, analysis, measurement, improvement, optimization, and automation of business processes [19].*

While each of these problems within the scope of BPM have a wide range of associated techniques attached to them, the field of automated planning has interesting touch points with every one of them. In the next section, we will describe briefly *how*. But before we get there, we discuss briefly *why*.

- The key advantage of modeling problems in this form is that it is **domain independent**, i.e., we can bring to bear decades of research and tools from the planning community (such as planners, domain-independent heuristics,

- editors, visualizers, and so on). The key challenge here is then the interfacing between a BPM practitioner and a representation the planner can understand. We will explore this in various contexts in the next section.
- The representations are **human-interpretable** and hence allows for iterative modeling and refinement with different stakeholders in the process;
  - Planning formalisms offer an **exponential scale-up** from the complexity of the representation to the complexity of the process. Though this means that classical planning, i.e., the simplest form of a planning problem, is NP-hard. This theoretical limit is rarely breached when it comes to applications of planning to BPM. Instead, an exponential scale-up means that:
    1. processes of much more sophistication can be composed the same amount of work as manual specification;
    2. much less work is required to specify processes of the same sophistication than manual specification; and
    3. coverage of a wide space of processes from the same domain-independent specification, i.e., a possibility of BPM-practitioners to go past hard-coded solutions to individual problems to domain-independent solutions using compilations to planning formulations.
  - Classical planning models constitute implicit **representations of finite state controllers**, and can be thus queried by standard verification techniques, such as Model Checking. In fact, this implicit representation is directly tied to the exponential scale-up since practitioners do not need to specify the control explicitly but rather only its declarative components.



**Fig. 1.** Different touch points of planning along the life-cycle of a business process.

## 2 Automated Planning for BPM

We will now reflect on how the advantages described above play out in various applications of planning in world of BPM. Figure 1 conceptualizes the various touch points of planning technologies along the life-cycle of a business process.

## 2.1 Automated Generation of Process Models

Current BPM technology is generally based on rigid process models making its application difficult in dynamic and possibly evolving domains, where pre-specifying the entire model is not always possible. In this context, the automated generation of process models not starting from an event log (that is often not available) but from the knowledge of the process context and goal is highly desirable. This ability to construct process models automatically from its individual declarative components is the primary application of planning in BPM and is among the many flavors of declarative modeling used in the field [10].

As an example, in [15], the authors applied planning for automated process model synthesis. Specifically, the use of planning enabled to build a plan that led from the world status to the goal status. That plan was the process model. Notice that, typically, a plan is a sequence of steps. Instead, the technique in [15] catered for partial orders allowing to encompass parallelism and (some) choice points. A particularly interesting design of that technique was that a partial knowledge of the world status was sufficient to synthesize the process model. For more examples for this type of application, we refer the reader to [14].

**Web Service Composition.** One important sub-theme in business process generation is the composition of existing web services to create new ones [20] – this creates complex semi-automated pathways among existing manual processes as well as augments manual paths with automation. Composing web services finds a ready ally in automated planning [2, 8]. We refer to [21] for a summary of work done in this area, and [25, 29] for a summary of challenges.

**Conversational Agents.** An emerging application of chatbots is goal-oriented conversation, i.e., conversational agents with underlying business processes. End-to-end learning models cannot specify such bots due to inability to connect the conversational elements to process constraints and execution. While traditionally the “dialogue tree” for such agents have been built manually, for example, using tools such as Watson Assistant or Google Dialogflow, emerging techniques built on automated planning [18, 24] has provided new pathways to domain authors to generate these structures automatically based off of their declarative components. This also has synergies with web service composition as well [5].

## 2.2 Trace Alignment

Within process mining, trace alignment is the problem of verifying if the observed behavior stored in an event log is compliant with the process model that encodes how the process is allowed to be executed to ensure that regulations are not violated. Trace alignment makes it possible to pinpoint the deviations causing nonconformity with a high degree of detail [1]. While there exist manifold explanations why a trace is not conforming, one is interested in finding the most probable explanation, i.e., one of the alignments with the least expensive deviations

(i.e., optimal alignments), according to some function assigning costs to deviations. The state-of-the-art techniques to compute optimal alignments against procedural [1] and declarative [7] process models provide ad-hoc implementations of the A\* algorithm. The fact is that when process models and event logs are of considerable size the existing approaches do not scale efficiently due to their ad-hoc nature and may be unable to accomplish the alignment task.

Among scalable approaches to solve the alignment task, in [11, 13] the authors have reduced trace alignment (let it be declarative or imperative) to a classical planning problem. This opportunity led to solving a number of additional problems. For example, it became possible to compute alignments in presence of coarse-grained timestamps - such that some events are marked as if they were contemporary, against the typical event log assumption [12].

### 2.3 Process Adaptation

Process Adaptation is the ability of a process to react to exceptional circumstances and to adapt/modify its structure accordingly [22]. While anticipated exceptions can be foreseen at design-time and incorporated into the process model as exception handlers, unanticipated exceptions refer to situations that emerge at run-time, thus requiring that BPM tools provide real-time monitoring and adaptation features to detect/repair them during process execution.

To overcome the limits of traditional process adaptation, which was based on an ad-hoc definition of exception handlers to build recovery procedures, in [16, 17] the authors presented a planning-based approach to adapt on-the-fly a running process instance requiring no predefined exception handler. Specifically, the SmartPM approach enables to automatically detect unanticipated run-time exceptions and exogenous events by monitoring the discrepancies between the expected reality, i.e., the (idealized) model of reality that reflects the intended outcome of the task execution, and the physical reality, i.e., the real world with the actual values of conditions and outcomes. If the gap between the expected and physical realities is such that the process instance cannot progress, the SmartPM approach resorts to classical planners to build a recovery procedure as a plan, which can thereby reduce the misalignment between the two realities, thus resolving exceptions that were not designed into the original process. In general, this falls under the broader theme of “replanning” in automated planning, and can be adopted to a wide range of problems including the specific case of automated web service composition [4] as discussed before.

### 2.4 Interpretability and Authoring Tools

The interpretability question for automated composition of process elements using automated planning boils down to *understanding the imperative consequences of declarative design*. These interpretability issues can occur at multiple stages:

**Process Definitions.** This is meant to reduce the level of expertise required to specify constructs that can interface easily with a planning representation. For example, in the context of web service composition, authors in [3, 23] tried out a tag based language that is at once easy to understand from the domain author’s point of view and at the same time easily compilable to a representation that an automated planner can consume. Authors in [23, 26] also demonstrate how the domain author can make use of not one but multiple compositions to understand how a process will evolve and use that knowledge to constrain their authoring problem. Authors in [18] do the same using a contingent plan instead.

**Process Understanding.** Once process elements have been defined, it must be ensured that the process author understands how those individual elements get composed into process controllers. Interestingly, the process author here can be the end-user themselves, as well as the usual developer or admin of the process.

In [27], authors introduced a vocabulary for triaging composed processes iteratively through a mixture of foils, landmarks (i.e. necessary steps), and abstractions. In this paradigm, the domain author fixes problems in the most simple abstraction of a process and then tests them in the fully composed process by querying it with “foils” or instantiations of the process that they feel should (or should not) be supported by the automated composition.

In [28], on the other hand, the explanations are for the end-user who wants to explore the inner workings of an “aggregated assistant” composed on the fly – by asking how certain things were done and why. Interestingly, such explanations are sometimes just a feature that is good to have (in terms of increased transparency and establishment of common grounds with the user) but they may also be required by law (e.g. GDPR rules may require establishing provenance and necessity of data flow in certain cases).

### 3 Conclusions

This concludes a whirlwind overview of the applications of planning for BPM. We encourage the reader to follow-up for more details with related surveys [6, 14, 30] and our tutorial on this topic at BPM 2021: [ibm.biz/bpm-2021-tutorial](http://ibm.biz/bpm-2021-tutorial).

**Acknowledgements.** The work of Andrea Marrella has been supported by the H2020 project DataCloud and the Sapienza grant BPbots.

### References

1. Van der Aalst, W., Adriansyah, A., van Dongen, B.: *Replaying History on Process Models for Conformance Checking and Performance Analysis*. Data Mining and Knowledge Discovery, Wiley Interdisciplinary Reviews (2012)
2. Araghi, S.S.: *Customizing the Composition of Web Services and Beyond*. U Toronto (2012). Ph.D. thesis

3. Bouillet, E., Feblowitz, M., Liu, Z., Ranganathan, A., Riabov, A.: A tag-based approach for the design and composition of information processing applications. In: OOPSLA 2008 (2008)
4. Bucchiarone, A., Pistore, M., Raik, H., Kazhamiakin, R.: Adaptation of service-based business processes by context-aware replanning. In: SOCA 2011 (2011)
5. Chakraborti, T., Agarwal, S., Khazaeni, Y., Rizk, Y., Isahagian, V.: D3BA: a tool for optimizing business processes using non-deterministic planning. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 181–193. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-66498-5\\_14](https://doi.org/10.1007/978-3-030-66498-5_14)
6. Chakraborti, T., Ishakian, V., Khalaf, R., Khazaeni, Y., Muthusamy, V., Rizk, Y., Unuvar, M.: From robotic process automation to intelligent process automation: emerging trends. In: BPM RPA Forum (2020)
7. De Leoni, M., Maggi, F.M., van der Aalst, W.M.: Aligning event logs and declarative process models for conformance checking. In: BPM 2012 (2012)
8. Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J.: Similarity search for web services. In: VLDB (2004)
9. Geffner, H., Bonet, B.: A Concise Introduction to Models and Methods for Automated Planning. Synth, Lectures on AI and Machine Learning (2013)
10. De Giacomo, G.: Artificial Intelligence-based declarative process synthesis for BPM. In: BPM 2021 (Keynote Talk) (2021)
11. De Giacomo, G., Maggi, F.M., Marrella, A., Patrizi, F.: On the disruptive effectiveness of automated planning for LTLf-based trace alignment. In: AAAI 2017 (2017)
12. de Leoni, M., Lanciano, G., Marrella, A.: Aligning partially-ordered process-execution traces and models using automated planning. In: ICAPS 2018 (2018)
13. de Leoni, M., Marrella, A.: Aligning real process executions and prescriptive process models through automated planning. *Exp. Syst. with App.* **82**, 162–183 (2017)
14. Marrella, A.: Automated planning for business process management. *J. Data Semantics* **8**(2), 79–98 (2019)
15. Marrella, A., Lespérance, Y.: A planning approach to the automated synthesis of template-based process models. *Serv. Oriented Comput. Appl.* **11**, 367–392 (2017)
16. Marrella, A., Mecella, M., Sardina, S.: Intelligent process adaptation in the SmartPM system. *ACM Trans. Intell. Syst. Technol.* **8**(2), 1–43 (2017)
17. Marrella, A., Mecella, M., Sardiña, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. *AI Commun.* **31**, 47–74 (2018)
18. Muise, C., et al.: Planning for Goal-Oriented Dialogue Systems. Technical Report (2020)
19. Nathaniel Palmer: What is BPM? <https://bpm.com/what-is-bpm>
20. Papazoglou, M.P., Georgakopoulos, D.: Service-oriented computing. *Commun. ACM* **46**(10), 24–28 (2003)
21. Rao, J., Su, X.: A survey of automated web service composition methods. In: Workshop on Semantic Web Services and Web Process Composition (2004)
22. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-30409-5>
23. Riabov, A.V., Boillet, E., Feblowitz, M.D., Liu, Z., Ranganathan, A.: Wishful search: interactive composition of data mashups. In: WWW (2008)
24. Rizk, Y., et al.: A unified conversational assistant framework for business process automation. In: AAAI Workshop on IPA (2020)
25. Sohrabi, S.: Customizing the composition of actions, programs, and web services with user preferences. In: ISWC (2010)

26. Sohrabi, S., Riabov, A., Katz, M., Udrea, O.: An AI planning solution to scenario generation for enterprise risk management. In: AAAI 2018 (2018)
27. Sreedharan, S., Chakraborti, T., Muise, C., Khazaeni, Y., Kambhampati, S.: D3WA+ - a case study of XAIP in a model acquisition task for dialogue planning. In: ICAPS 2020 (2020)
28. Sreedharan, S., Chakraborti, T., Rizk, Y., Khazaeni, Y.: Explainable composition of aggregated assistants. In: ICAPS Workshop on Explainable AI Planning (2020)
29. Srivastava, B., Koehler, J.: Web service composition - current solutions and open problems. In: ICAPS Workshop on Planning for Web Services (2003)
30. Vukovic, M., Gerard, S., Hull, R., et al.: Towards automated planning for enterprise services: opportunities and challenges. In: ICSOC 2019 (2019)