

Supporting and assisting the execution of flexible healthcare processes

Steven Mertens, Frederik Gailly, Geert Poels

Department of Business Informatics and Operations Management

Faculty of Economics and Business Administration

Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium

{*steven.mertens,frederik.gailly,geert.poels*}@ugent.be

Abstract—Business process management has yet to fully conquer the healthcare sector. Although it has proven effective for more static and standardized healthcare processes, it does not offer a definitive answer for the dynamic, flexible and knowledge-intensive processes typically associated with the healthcare sector. Contemporary business process techniques, like imperative business process models (e.g., BPMN), do not allow for the flexibility required by these types of processes. In this paper we propose a new type of tool, a recommendation-based robust business process engine, in an effort to combat this problem. The goal is to not only support a more process-driven approach in healthcare, but also assist the users (i.e., doctors, nurses and other hospital personnel) during these processes. This should result in an improvement of the service levels and add another barrier to prevent doctor errors, without being detrimental to the empowerment of the personnel that characterizes the healthcare sector.

Keywords—Business process modeling, healthcare processes, recommendation systems, declarative process models, decision management

I. INTRODUCTION

Business process modeling (BPM) [1] has, generally speaking, been successfully adopted in many different sectors (e.g., manufacturing, sales, marketing...). However, healthcare is one of the few sectors that have yet to fully embrace the process-driven approach [2]–[4], even though some of the main concerns trending in eHealth are very similar to other sectors, namely cost reduction, efficiency and patient-orientation [4]. The primary cause for this is the flexible, dynamic and knowledge-intensive nature of healthcare processes [5], where deviations and variations are the rule, rather than the exception. Hospitals still depend heavily on the knowledge, skill and judgment of their personnel to handle the day-to-day workload in an acceptable manner. The doctors are heavily empowered and presumed to be fully aware of a patient's history, allergies, previously performed activities, emergent health problems and possible particular preferences. Yet, doctors usually do not treat just one patient at a time, but rather they treat many simultaneously. Additionally, they have to keep track of many possible diagnoses and treatments, the legislative rules and norms, the hospital's policy and availability of key resources. This results in a large variety of ways that a certain case could be and would be processed in different hospitals and even by different doctors in the same hospital [6]–[8].

The primary goals of healthcare processes, for both the patients as the doctors, are timeliness (i.e., the correct diagnosis and treatment in a timely manner) and efficiency (i.e., minimizing the waiting time and excess activities during process). Both are generally negatively influenced by the current variety of tacit processes, which rarely are the result of any research [6], as some variants will be less beneficial or even possibly harmful to the patient, while others are just less efficient [9]. However, as the solution space is large and dynamic, there exist many valid variations. Finding the optimal one is hard and time-consuming, while finding one that stays optimal over time is impossible.

The goal of this paper is to propose a process-driven tool to support flexible healthcare processes, called a recommendation-based robust business process engine. It is not our intent to eliminate all variation, nor is this workable, but rather to increase the transparency and quality of the decision making concerning which path to follow in what circumstances. The idea is to gain valuable insight as to why a certain variation is chosen in certain circumstances and what its consequences are. This information will then be offered to the users (i.e., doctors, nurses and other hospital personnel) at the moment that the decision needs to be made. The final call is still made by the user; the system only offers support and insight as it is not an expert system. This system thus attempts to avoid undesirable variations, but does not disallow them as they may be needed in certain extreme circumstances. This should result in an even better prevention of doctor errors, an increase in service levels and a decrease in steepness of the learning curve for beginning doctors as the knowledge of more experienced doctors is made more explicit. Ultimately, this translates to a higher patient and doctor satisfaction by saving lives, time and money.

This paper is structured as follows. Section II gives a high-level description of the system requirements in order to realize the above mentioned tool. In section III, a mock-up of the tool is presented. Finally, we conclude the paper and describe the future work in section IV.

II. SYSTEM REQUIREMENTS

A system as introduced above must meet certain requirements for it to be functional:

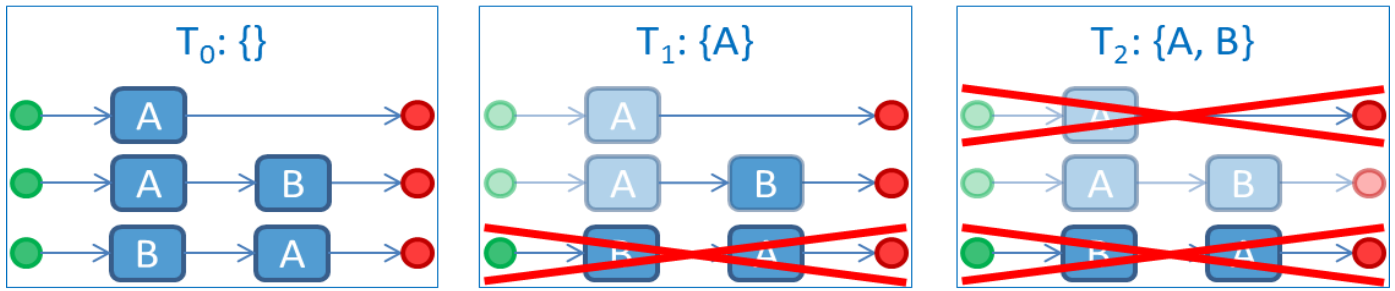


Fig. 1. Example of declarative run-time flexibility visualized with imperative models

1. Know all possible activities of the process
2. Keep track of the current state of process instances
3. Keep track of many and possibly complex constraints
4. Allow all valid variations to complete the process
5. Disallow next activities that make end-states unreachable
6. Log the usage data
7. Recommend best available next activities
8. Encapsulate the knowledge of the decision making process

Requirement 1 and 2 can be satisfied by a regular business process engine (e.g., jBPM) and, if business rules are supported, requirement 3 can also be fulfilled. A regular process engine uses an imperative process model to specify an exact path and forces the user to follow it. This model is designed at build-time¹ and thus offers no possibility of divergence from the path at run-time². This works well for predictable and stable processes, like production and administration processes, but is too restrictive for flexible processes [10].

To support flexible processes the system needs to allow the user to personalize the chosen execution path to its specific execution context at run-time. This means that all valid variations need to be supported and invalid execution paths need to be disallowed (i.e., requirement 4 and 5). We can accomplish this in two ways: imperatively enumerating all options or using a declarative model that implicitly contains all these variations. The former is very time consuming and needs to be done each time that the model is adjusted, which is not at all desirable in a dynamic environment. As stated by Goedertier, Vanhienen and Caron [11], dynamic, human-centric, knowledge-intensive and non-standardized business processes are most likely to require the run-time flexibility offered by declarative process modeling. In contrary to imperative approaches, declarative approaches determine only the activities that may be performed as well as constraints prohibiting undesired behavior [12]. This results in the specification of a collection of rules and constraints that leave enough freedom for various execution paths towards the process goals to exist. Furthermore, by explicitly defining these rules that would normally remain tacit we enhance the knowledge management capabilities of the organization, allow

for reuse of the rules in other process models, increase process compliance and improve the overall traceability [13]. We will refer to a process engine that uses a declarative model instead of an imperative model as a robust business process engine.

As illustration of how such a robust process engine works internally, consider the example execution of a very simple model with just two activities A and B. There are two rules: B can only be executed when it is preceded or succeeded by A and both activities can only be executed once. This leaves three valid variations that implicitly exist at the start of the execution, as visualized at T_0 in Fig. 1. At T_1 , the user chooses to execute activity A, which is allowed by the first two variations, but invalidates the third variation. Finally, at T_2 the user executes activity B, which in turn invalidates the first variation. At this time the process ends and in hindsight we can conclude that this particular user followed the second variation. It is important to notice that this choice was made at run-time, not at build-time. This is exactly the power of declarative models, as they allow for the path to be adjusted to the specific circumstances of each case at hand, while making sure that the rules are satisfied.

Requirement 6 states that the system should attempt to store all available relevant data for each terminated or running process instance (i.e., each patient case). This allows us to map and analyze this data so that we can gain more insight into the specifics of the process and possibly even improve upon it. For example, it could be used to count the number of historic cases impacted by a potential policy change in order to predict the impact it would have on current and future cases.

We would satisfy requirements 1 through 6 by creating a robust process engine with logging capabilities, but this system would be received badly due to its bad user friendliness. The user would be expected to know all of the possibly numerous variations himself to be able to choose the best one. To counter this lack of guidance, requirement 7 calls for the system to make recommendations in order to assist the user when making a decision. Similar approaches have been proposed for environments with less focus on flexibility and knowledge [14], [15]. The recommendations will be based on multiple criteria, like the overall completion time, the cost, the future flexibility, resource availabilities and historic compliance, so recommendations will even be possible with no historic data available. This offers some much needed guidance and insight to the user, but does not restrict the user's options as he can still choose any valid next activities. We merely try to provide

¹ Build-time: before deployment, when creating the model.

² Run-time: after deployment, when the model is used.

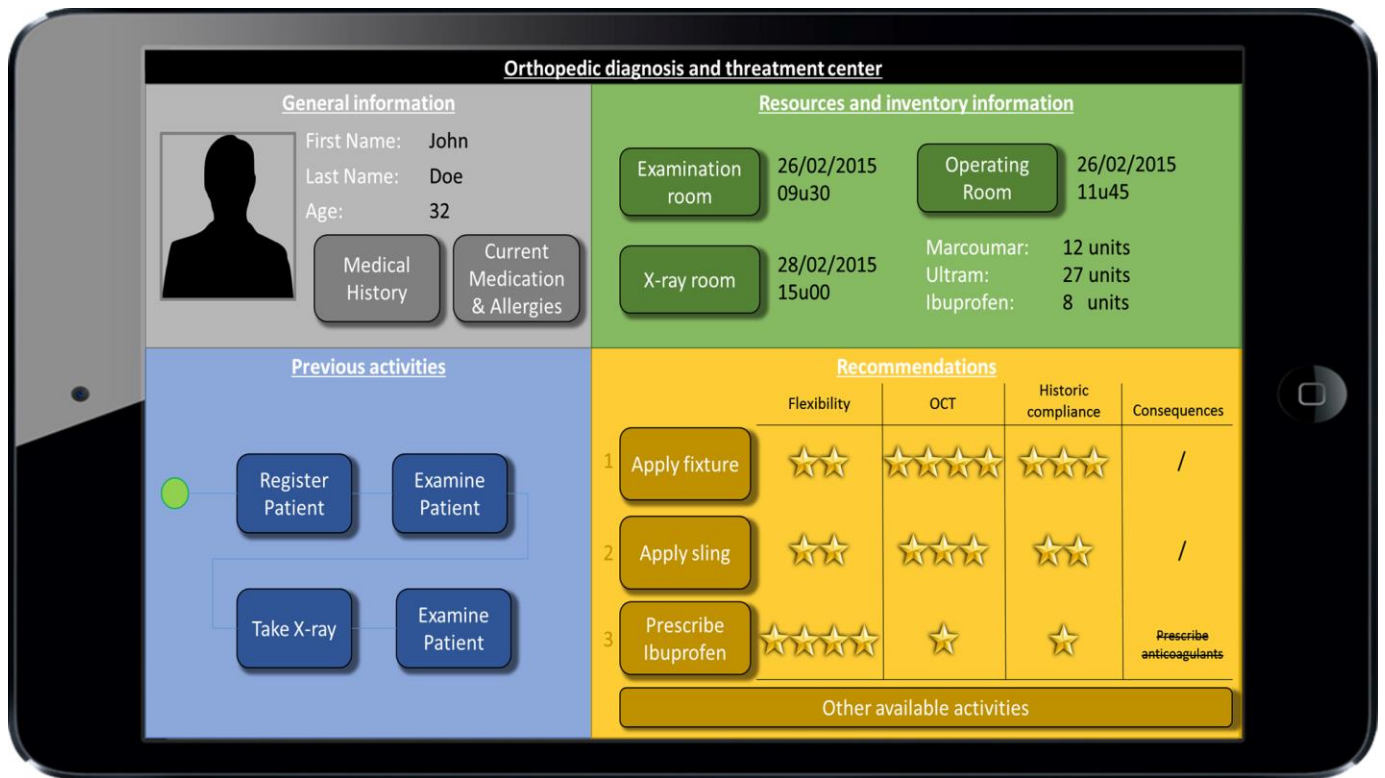


Fig. 2. A mock-up of how the envisioned system would look like to the user

support and assistance to the user during the decision making process, in contrast to expert systems, which attempt to automate the whole decision process. This should result in better informed users, more transparency and an overall improvement of the decision making process. However, it also adds a lot of complexity to the system, as it now needs to be able to analyze historic data and metadata about the process in order to calculate or predict certain criteria.

Finally, requirement 8 states that we also want the system to learn from the user. As the system is used in all sorts of situations, patterns will emerge. These patterns are the tacit knowledge that can be found inside the heads of the doctors, of which they might not even be fully aware of. By capturing these patterns we indirectly capture the evidence-based medicine knowledge as is known to the doctors, either consciously but also unconsciously. The younger doctors, that recently finished their studies, will probably contribute more about state-of-the-art techniques and evidence-based medicine, while older doctors contribute with their vast amount of practical experience. Practically, a combination of process and decision mining techniques will be used to identify interesting patterns. The knowledge that we are interested in is how the doctors decide on what the next step will be. The knowledge about the decision process is slowly built up as the system is being used. In the first phase, mere suspicions of patterns will be used to improve the scoring of recommendations. But ultimately, if the confidence and support (measured by historic compliance) of the patterns is high enough, these can be consolidated by domain experts and used explicitly throughout the process. This can be a big aid for young doctors with little

experience, as this knowledge normally only could have been learned by prolonged observation of how cases are handled by more experienced doctors or, regrettably so, by making costly mistakes and learning from them.

III. RECOMMENDATION-BASED ROBUST PROCESS ENGINE

As the healthcare processes that we are interested in are knowledge- and data-intensive, we want the system to have access to as much relevant data as possible. Specifically this means that we want to integrate the related systems, in order to access resource schedules, patient files and history, inventories... Fig. 2 shows a simple mock-up of how we envision the integrated system would look like from the point of view of the users.

The mock-up contains 4 main components: general information, resources and inventory information, previous activities and recommendations. First, the general information shows the available data from the patient's file. In the second component an overview of the resources availabilities and the inventory of relevant drugs can be consulted. The next component gives an overview of the already performed activities in the treatment of the patient (i.e., requirement 2). These activities are clickable, so that the data produced by each activity (e.g., scans and reports) can be reviewed to keep the doctor fully up-to-date of the specifics of the case. Although the system also offers guidance and assistance, the user is still able (and advised) to examine the relevant data himself to make an independent decision. These first three components allow the user to do this by offering the required background information. The last and most important component gives the

user recommendations on what to do next. This is a direct result of requirement 7, but indirectly also ensures requirements 1, 3, 4 and 5 are satisfied. It shows a number of top recommendations and their relative scores for each of the used criteria. This could possibly be enriched with the direct and indirect consequences of selecting a certain recommendation (e.g., which other activities will no longer be possible). Of course it is also possible to select lower ranked recommendations, but invalid next activities are made unavailable.

Requirements 6 and 8 are the only ones that are not directly visible to the (regular) user. The available process and application data is stored on the back-end of the system. This data is analyzed in the background in order to detect decision patterns. These in turn can be used to score recommendations (i.e., historic compliance), but their consolidation occurs separately. The patterns with a high enough confidence and support could, for example, be discussed regularly by all doctors of the department. The patterns that are confirmed can be used by the system to provide an explanation and justification of why a certain recommendations scores higher or lower than others.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have presented a process-driven system, the recommendation-based robust business process engine, which offers support and guidance during the execution of healthcare processes. It does not prohibit any valid action, but instead allows a maximum of freedom and flexibility to the users while making sure that everything is done by the book. Additionally, users will be helped through each the decision making process by being presented with recommendations and offered insight into what the consequences of a certain option will be. This does not only help the users, but also the patients and the whole organization as service levels and transparency will increase, another barrier preventing doctor error is added, knowledge management is improved and younger doctors gain an extra source of insight to learn their trade. The system can also be applied to other service branches. Any process that can be classified as a flexible, but still at least loosely framed [16], process that is knowledge-intensive should benefit from the techniques used by the tool.

The system itself is currently still a work-in-progress. The foundations of the techniques and algorithms behind the robust business process engine and the recommendation engine [17] have been implemented, though they are not yet able to deal with decisions. This is because current declarative process modeling languages (e.g., Declare) are missing the support for the decision logic as well. So in order to realize the system, we first will need to add this support to one of those languages. A proposal for the basics of the new language has been created and accepted for publication in the near future. In the next phase we will further expand this to a formal metamodel of the new language and give the system the proper support for this new language. Another important future step is to study and select different process mining and decision mining techniques [18] in order to integrate these into the system. They are

essential for the knowledge extraction and consolidation requirement and thus for the general learning capabilities of the system. If existing techniques are found to be insufficient in their current form, we will propose a tailored solution.

The system will also be thoroughly evaluated in different ways. First, the added value of recommendations will be evaluated through comparative studies. Next, when a prototype of the system is available, we will evaluate the system in an actual healthcare environment. This should give a clear view of the effectiveness of the proposed system. Additionally, we will also test the system in another service sector in order to evaluate its general usefulness and effectiveness.

REFERENCES

- [1] M. Weske, *Business process management: concepts, methods, technology*. Springer, 2007.
- [2] R. Lenz and M. Reichert, "IT support for healthcare processes - premises, challenges, perspectives," *Data Knowl. Eng.*, vol. 61, pp. 39–58, 2007.
- [3] P. Palvia, K. Lowe, H. Nemati, and T. Jacks, "Information technology issues in healthcare: hospital CEO and CIO perspectives," *Inf. Technol.*, vol. 30, 2012.
- [4] F. C. Payton, G. Paré, C. LeRouge, and M. Reddy, "Health care IT: Process, people, patients and interdisciplinary considerations," *J. Assoc. Inf. Syst.*, vol. 12, no. 2, 2011.
- [5] C. Di Ciccio, A. Marrella, and A. Russo, "Knowledge-intensive Processes: An overview of contemporary approaches," in *KiBP. CEUR Workshop Proceedings*, vol. 861, 2012, pp. 33–47.
- [6] J. E. Wennberg, "Unwarranted variations in healthcare delivery: implications for academic medical centres.," *Br. Med. J.*, vol. 325, pp. 961–964, 2002.
- [7] J. Appleby, V. Raleigh, F. Frosini, G. Bevan, H. Gao, and T. Lyscom, *Variations in health care: The good, the bad and the inexplicable*. London: King's Fund, 2011.
- [8] M. De Regge, P. Gemmel, I. Claerhout, and P. Duyck, "How standardized are standardized processes in hospitals: are we there yet?," in *Proceedings of the 21st EurOMA Conference*, 2014.
- [9] P. Gross, S. Greenfield, S. Cretin, J. Ferguson, J. Grimshaw, R. Grol, N. Klazinga, W. Lorenz, G. Meyer, C. Riccobono, S. Schoenbaum, P. Schyve, and C. Shaw, "Optimal methods for guideline implementation: conclusions from Leeds Castle meeting," *Med Care*, vol. 39, p. 8 Suppl 2 H85–92., 2001.
- [10] R. Lenz, M. Peleg, and M. Reichert, "Healthcare Process Support: Achievements, Challenges, Current Research," *Int. J. Knowledge-Based Organ.*, vol. 2, 2012.
- [11] S. Goedertier, J. Vanthienen, and F. Caron, "Declarative business process modelling: principles and modelling languages," *Enterp. Inf. Syst.*, no. November, pp. 1–25, Sep. 2013.
- [12] C. Haisjackl, I. Barba, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, and B. Weber, "Understanding Declare models: strategies, pitfalls, empirical results," *Softw. Syst. Model.*, Sep. 2014.
- [13] J. Krogstie, "Perspectives to process modeling," in *BPM. SCI*, vol. 444, Springer, 2013, pp. 1–39.
- [14] H. Schonenberg and B. Weber, "Supporting flexible processes through recommendations based on history," in *BPM. LNCS*, vol. 5240, vol. vol. 5240, Springer, 2008, pp. 51–66.
- [15] I. Barba, B. Weber, C. Del Valle, and A. Jiménez-Ramírez, "User recommendations for the optimized execution of business processes," *Data Knowl. Eng.*, vol. 86, pp. 61–84, Jul. 2013.
- [16] M. Dumas, W. M. P. van der Aalst, and A. H. ter Hofstede, *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, New York, USA, 2005.
- [17] S. Mertens, F. Gailly, and G. Poels, "Generating business process recommendations with a population based meta-heuristic," in *BPM Workshops. LNBIP*, vol. 202, Springer, 2014.
- [18] W. M. P. van der Aalst, *Process Mining*. Springer, 2011.