Evaluation of the Dynamic Construct Competition Miner for an eHealth System

David Redlich[1,2](✉), Mykola Galushka[2], Thomas Molka[2], Wasif Gilani[2], Gordon Blair[1], and Awais Rashid[1]

[1] Lancaster University, Belfast, UK
mr.redlich@gmail.com, {gordon,marash}@comp.lancs.ac.uk
[2] SAP Research Centre Belfast, Belfast, UK
{mykola.galushka,thomas.molka,wasif.gilani}@sap.com

Abstract. Business processes of some domains are highly dynamic and increasingly complex due to their dependencies on a multitude of services provided by various providers. The quality of services directly impacts the business process's efficiency. A first prerequisite for any optimization initiative requires a better understanding of the deployed business processes. However, the business processes are either not documented at all or are only poorly documented. Since the actual behaviour of the business processes and underlying services can change over time it is required to detect the dynamically changing behaviour in order to carry out correct analyses. This paper presents and evaluates the integration of the Dynamic Construct Competition Miner (DCCM) as process monitor in the TIMBUS architecture. The DCCM discovers business processes and recognizes changes directly from an event stream at run-time. The evaluation is carried out in the context of an industrial use-case from the eHealth domain. We will describe the key aspects of the use-case and the DCCM as well as present the relevant evaluation results.

Keywords: Business process management · Process discovery · Enterprise architecture · Complex event processing · eHealth

1 Introduction

The major objective of the European project TIMBUS is to enable Digital Preservation (DP) of business processes and services [5]. A major innovation of the project is to enable business process centric risk management to help identify critical parts of crucial business processes and services, which need to be preserved in order to ensure long-term availability and business continuity. The TIMBUS risk management process, based on the ISO 31000 standard, starts with first establishing the context of the target system. This means capturing all information about the business processes, including the behavioural information, the execution context, the legal context, the resources needed from the software, hardware and facility level to support the top level business processes, etc. The extracted context information is the basis for different actions: (1) reasoning

W. Abramowicz (Ed.): BIS 2015, LNBIP 208, pp. 115–126, 2015.
DOI: 10.1007/978-3-319-19027-3_10

about whether or not DP is feasible, (2) risk management, and (3) if required, for the execution of digital preservation process. Driven by requirements from three different business processes, coming from the domains of Civil Infrastructure, eScience and eHealth, the individual tools and the overall TIMBUS solution are required to be generic in order to be applicable to the different use-cases.

The first step in the TIMBUS risk management process is the availability of the control-flow information of the target business process and its performance and resource information, which is generally available in the process logs. Two types of Process Extractors have been developed in the project: (1) a statically operating genetic miner that works on the historic business process logs [14] and is suitable for long-life business processes that do not change over time, such as the one from Civil Infrastructure domain. However, due to the characteristics of the static genetic miner (non-deterministic as well as long and unpredictable execution time) it is not applicable for monitoring fast and dynamically changing processes such as the ones from the eHealth domain [3]. For these use-cases it is required to monitor the system for changes in the process which result in changes in the risk assessment and potentially entail a new preservation iteration. This is why the second type of Process Extractor has been included in the project: (2) the Process Monitor that detects changes in the business process during run-time.

In this paper we describe how the Dynamic Constructs Competition Miner (DCCM) has been integrated into the TIMBUS architecture as a Process Monitor and evaluate its suitability in the context of the DrugFusion use-case from the domain of eHealth. In the remainder of the paper we first introduce the DrugFusion use-case in Sect. 2, then explain important aspects of the DCCM and its integration into the TIMBUS architecture as a Process Monitor in Sect. 3. This is followed by an evaluation of the DCCM for the DrugFusion use-case in Sect. 4 where we explain findings and shortcomings of the solution. Then in Sect. 5, the paper is concluded by summarizing emerged results and findings.

2 eHealth Use Case (DrugFusion)

Each prescription drug package selling in Europe must contain information about how it works, what is the intended effect, and cautions for its use. Medical practitioners who are authorized to prescribe drugs try to identify the best treatment strategy by assessing a patient's condition and previously prescribed medicines. Such treatment may include a prescription of one or more drugs which need to be taken within the predefined time interval. Sometimes prescribed medications may cause an Adverse Drug Reaction (ADR) [7–9]. The study of ADRs is conducted in the field known as Pharmacovigilance. ADRs describe harms caused by taken medications at a normal dosage during normal use [2,12]. ADRs may occur in the following scenarios: a single dose, a prolonged usage of a drug or a result of combined use of two or more drugs (this scenario is targeted by the DrugFusion system described in this paper). ADRs expression has a different meaning than "side effect", since side effect might also imply that the effects can be beneficial.

A more general term, Adverse Drug Event (ADE) [6,11], refers to any injury caused by the drug (whether drugs were used at normal dosage and/or due to overdose) and any harm associated in such case, i.e. ADRs are a special type of ADEs.

DrugFusion has been developed and made available as a use case by an industrial partner in the TIMBUS project. The main objective was to provide a platform for avoiding ADEs. The high-level description of the DrugFusion process for discovering rules which help to predict ADEs is presented in Fig. 1. The ADE rules discovery process can be split into three distinct phases: *Creation of Dictionaries*, *Load of Adverse Event Report (AER)* and *Discovery of ADE Rules*. The majority of operations in the second and third phases are implemented using a map-reduce approach and run on a Hadoop cluster.

Creation of Dictionaries Phase. Begins with collecting data for products. A product is an abstraction of the following three types of data: indication, drug, and reaction. The indication data represents patients' diagnosis. The drug data consists of generic names of prescription medications. The reaction data describes undesirable effects on a patient's health, caused by applied treatment. This data is essentially used to create dictionaries for indications, drugs and reactions. The resulting dictionaries are created in parallel and under the supervision of three different groups of experts. Each dictionary is created through a number of iterations. When the scope of each dictionary is finalized, a specialized filter removes information which is not relevant for future lookup operations. The dictionary creation phase is concluded with mapping active ingredients to drugs represented in the dictionary. This process is fully automated, however, if the system cannot make a conclusive mapping decision, it requires an expert to perform the manual assignment.

Load of AER Phase. Begins with downloading drug usage data[1]. The following categories of report data are considered: demographic, indication, drug, and reaction. A loading process performs data cleansing and a replacement of indication, drug and reaction names with unique identifiers obtained from the already created dictionaries. The loaded data of each category is moved to a predefined location to enable loading the joined report. All operations involved in the loading of individual report data can be executed in parallel. The loading of the joined report combines demographic, indication, drug, and reaction information into individual cases using the unique event identifiers. This phase ends with a drug normalization step in which inconsistencies of the drugs' names are resolved. Usually, pharmaceutical companies have their own line of drugs for targeting generic groups of illnesses. These drugs have the same structure but different brand names. Such diversity in drug names causes a significant impact on the complexity of the ADE rules discovery algorithm which is addressed in the normalization step.

[1] DrugFusion downloads adverse events report published by United States Food and Drug Administration (http://www.fda.gov) every quarter.

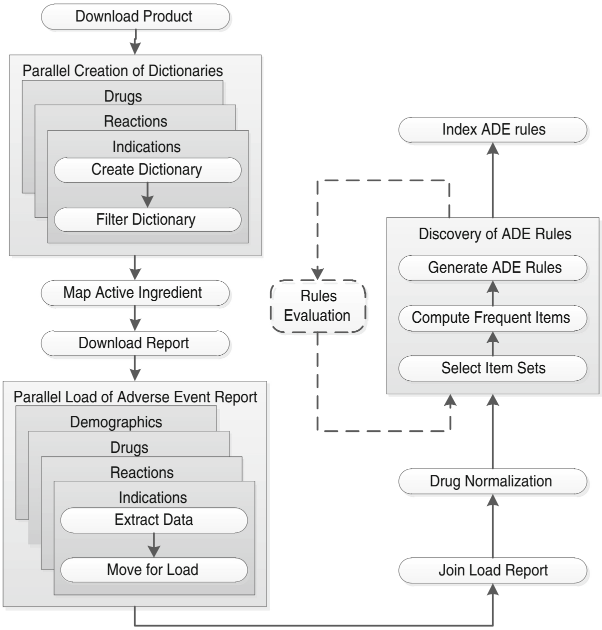

Fig. 1. DrugFusion business process.

Discovery of ADE Rules Phase is implemented by using a specifically modified Apriori Algorithm [15]. It includes: selection of item sets (these item sets combine patients' drug and reaction information), computing of frequent items (these computations run until all items are considered) and generating of ADE rules (where each rule is assigned with confidence and support values, respectively). If some of the obtained rules contradict existing medical observations, a group of experts performs an evaluation of the rules. It might trigger modifications in the items selection algorithm and a relaunch of the discovery process, which can be repeated a number of times. This phase as well as the DrugFusion process is terminated by indexing the obtained rules.

3 Dynamic Process Discovery for Digital Preservation

The goal of the TIMBUS project is to enable the digital preservation of business processes. An important part of this is to be able to make decisions about if, how, and when services/components/modules need to be preserved in order to ensure the execution and continuity of an organization's business functions. Business functions are typically implemented by business processes which are defined as *"...a series or network of value-added activities, performed by their relevant roles or collaborators, to purposefully achieve the common business goal"* [10]. In this paper we focus on the control-flow perspective of a business process, i.e. the activities (steps that represent the execution of actual work) and their execution order defined by control-flow elements, e.g. XOR-Split, XOR-Join, AND-Split, AND-Join.

A number of components have been developed to carry out these analysis on different abstraction levels of the business process and the underlying IT infrastructure, e.g. business process vs. resource level. These analyses operate either in an on-demand or a continuous fashion, depending on requirements of the TIMBUS project use-cases and on the level of automation of the employed methods. In many organizations, the involved business process models are not documented or the documented models deviate from the actual executed process, e.g. for the DrugFusion use case a documented process model is available (see Fig. 1) but does not accurately reflect the reality as will be shown in the evaluation, Sect. 4. The extraction of a business process model from a given event log without the usage of any a-priori information is addressed by *Process Discovery* algorithms [19], e.g. [14,16,18,20].

Of a high importance in the context of the TIMBUS project is the detection of changes in the business process that potentially have implications on the assessment of risks and the infrastructure which is to be preserved. For instance, the process of the DrugFusion use-case shown in Fig. 1 did initially operate without the "Rules Evaluation" carried out by experts in the "Discovery of ADE Rules" phase. The introduction of that step and the associated loop changed the behaviour of the process, thus increasing the severity of risks and introducing additional resources required for carrying out the "Rules Evaluation" activity. To detect these changes in the monitored system, a process monitoring component is required to be part of the TIMBUS solution. As opposed to traditional process discovery algorithms that calculate a process model from an input log in a static fashion, the process monitor has to work in a dynamic fashion, i.e. events are not recorded in a log but directly processed to changes of the system's "state". This event-based processing is an application of Complex Event Processing (CEP), a method to capture and filter low-level events and aggregate them to complex events representing high-level information about the system [13] - in this case information about the business process model.

3.1 Dynamic Constructs Competition Miner

The process monitor in the TIMBUS solution has been implemented with a modified Constructs Competition Miner (CCM) originally proposed in [16]. The original CCM is based on two fundamental steps:

(1) Footprint Creation, in which the footprint consisting of information on the global relations between any two elements of a set of activities is calculated from a given event log. In particular, two different relations are contained in the CCM footprint: (A) *appears before first*, a relative measure that records in how many cases an activity appeared before the first occurrence of another activity, and (B) *appears before*, a measure that records in how many cases an activity appears at some point before another activity (no matter if it was the first occurrence or not). For an example regard the following sequences of activity occurrences: $[DE], [EDE], [ED]$ are three instances of a process execution consisting of the two activities E and D, then appeared E before the first occurrence of D in

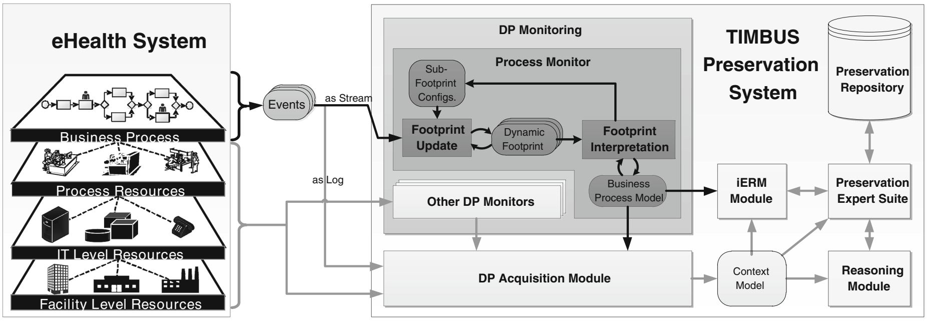

Fig. 2. The high-level view of the TIMBUS DP architecture with focus on the Process Monitor - Rounded corners: Models; Sharp corners: Agents

67 % of the cases, and D before the first appearance of E in 33 % of the cases (*appears before first* relation), whereas E appears before any D in 67 % of the cases, and D before any E in also 67 % of the cases[2] (*appears before* relation).

(2) Footprint Interpretation, in which the footprint is interpreted to a business process model construct. The CCM is technically able to identify and build a business process model based on the following constructs: *Sequence, Choice, Parallelism, Loop, Loop over Sequence, Loop over Choice, Loop over Parallelism* and constructs for single activities [16].

These two steps are repeatedly executed in a recursive fashion, with each step identifying a construct and splitting up the set of activities into subsets which are then again analysed the same way. This is carried out until the set of activities cannot be split up any further. The final result of the algorithm is a business process model representing the behaviour recorded in the log. For more details on the functionality as well as the footprint and business process model of the original CCM please see [16].

In [17] modifications to the CCM algorithm have been proposed that enable Dynamic Process Discovery in an event-based fashion as required for the Process Monitoring component in the TIMBUS solution. The result is the Dynamic Constructs Competition Miner (DCCM) in which the two steps, *Footprint Creation* and *Footprint Interpretation*, were completely separated, each with their own respective life cycle. Furthermore, the *Footprint Creation* was altered towards a *Footprint Update* method which is not analysing a complete log but operating on an event-by-event basis, updating the footprint with each event. Due to this a significantly low execution time for the event processing has been achieved and no record of previously occurred events has to be stored. Another feature of the *Footprint Update* is that the influence of older cases on the footprint gradually decreases until it eventually disappears [17], thus supporting the monitoring of dynamically changing processes. Since the *Footprint Interpretation* is

[2] For sequence [EDE] both relations are true: E appears before D and D appears before E.

relatively cost effective, it is not part of the event occurrence life cycle, but rather executed on-demand or in intervals (e.g. every 10 s). Additionally, other modifications making the original CCM fit for run-time application have been carried out and are described in [17], e.g. introducing a third relation, *direct neighbours*, to speed up the interpretation step. Especially DCCM's features of *robustness* (dealing with exceptional behaviour) and *scalability* (can easily manage 100 s of events/second) qualifies the DCCM for application as Process Monitor in the TIMBUS framework.

3.2 Integration in TIMBUS Project

The Digital Preservation (DP) framework developed within the TIMBUS [4] project provides a unique set of solutions going beyond the scope of existing DP approaches. It covers all aspects of traditional DP system such as preserving a digital content but also addresses enterprise risk analysis and business continuity planning. It covers a wider scope of DP processes, which includes intelligent Enterprise Risk Management (iERM) for automatic identification and prioritization of risks within an enterprise and ability to minimize those risks by taking a specific set of actions including DP. A high-level view of the TIMBUS DP analysis architecture is shown in Fig. 2. It consists of seven main modules: DP Monitoring, DP Acquisition, iERM, Context Model, Preservation Expert Suite, Reasoning, and Preservation repository.

A *Context Model* has been designed and developed in the project [1] that is meant to be populated with the complete context of the business process. It acts as a single data source in the TIMBUS solution for carrying out the risk management, reasoning and the digital preservation process. A number of static context extractors, for example, for software, hardware, and business process, etc., have been developed within the project and are part of the *DP Acquisition Module*. The aim of the DP Acquisition Module is to provide functionality for extracting the contextual information initially or on-demand, and then populating automatically this information into the Context Model. In the iERM Module the Context Model can be imported, annotated with different risk factors, and assessed according to the specified annotations. The assessment is performed in an interaction mode, where a risk expert can alter different model parameters and run simulations to identify the critical subset of resources and business processes, which require preservation. A generated Risk Assessment Report (RAR) is analysed by the Preservation Expert Suite (PES) with the help of the Reasoning module. PES combines a set of tools integrated to provide the centralized control for preservation and redeployment cycles. In the preservation cycle the PES transfers virtualized hardware components from a VM environment into the preservation repository. In the redeployment cycle PES performs a reverse operation where it transfers virtualized hardware components from the preservation repository back into the VM environment.

Since dynamic changes are expected in some of the use-cases, e.g. eHealth, a DP Monitoring Module is integrated in the TIMBUS architecture. It consists of different monitors, each monitoring a different aspect of the business process.

In Fig. 2 the agents and models of the Process Monitor, a module to discover the process behaviour and changes in the process at run-time, are displayed in more detail. The static process extractor employed in the DP Acquisition module could not be utilized for this task due to its non-deterministic behaviour and the non-compliance with run-time requirements (the extractor is based on a genetic algorithm [14]). Instead the Process Monitor was implemented with the Dynamic Constructs Competition Miner as introduced earlier. The Process Monitor module offers a RESTful Webservice API as interface to communicate with the service module. It provides for instance a method `onEvent(Event e)` which is invoked by the monitored system for each occurring event. Every event triggers a Footprint Update which causes a small alteration of the Dynamic Footprint. The Footprint Update module is highly efficient and takes only a small constant amount of time to execute the update on the footprint (100 s of events can be processed per second). The Dynamic Footprint is transformed to a Business Process Model by the Footprint Interpretation. Since this is relatively cost intensive (up to 3 s for very large processes with 100 activities on a normal machine) it is not executed for every event but runs decoupled in a second life cycle, e.g. scheduled for every 10 s. The Footprint Interpretation also formulates special requests (Sub-Footprint Configurations) for the Footprint Update if it is required (see [17]). Past events are discarded after they have been processed. Only the Dynamic Footprint and the current Business Process Model are stored in memory and as a result use only a very limited amount of memory.

The sum of small alterations on the Dynamic Footprint caused by individual events can eventually amount to a change in the Business Process Model, e.g. through introducing a new activity or new behaviour as seen in the evaluation section. The DCCM supports both, disregarding old behaviour and incorporating newly introduced behaviour. To identify a change in the process the Process Monitor always compares the previous Business Process Model with the newly interpreted Business Process Model at every execution of the Footprint Interpretation. If a change occurs the DP Acquisition Module is notified which requests the current Business Process Model with the method `BPModel getCurrentModel()` provided as a service and updates the Context Model accordingly. This service interface is also directly utilized by the iERM tool which can import the current business process model when performing a run-time risk analysis.

4 Evaluation

In this section we will carry out an evaluation of the DCCM integrated into the TIMBUS solution in the context of the DrugFusion use-case introduced in Sect. 2. In a first step we will examine what the actual business process looks like, i.e. we will hook the Process Monitor into the event stream produced by the use-case. The result after the first recorded 15 process instance executions is displayed in Fig. 3.

The "Creation of Dictionary" phase includes activities from "DownloadNdcRepository" to "MapActiveIngredient". The parallel execution of "Creating a Dictionary" and "Filtering a Dictionary" for Reactions, Drugs, and Indicators

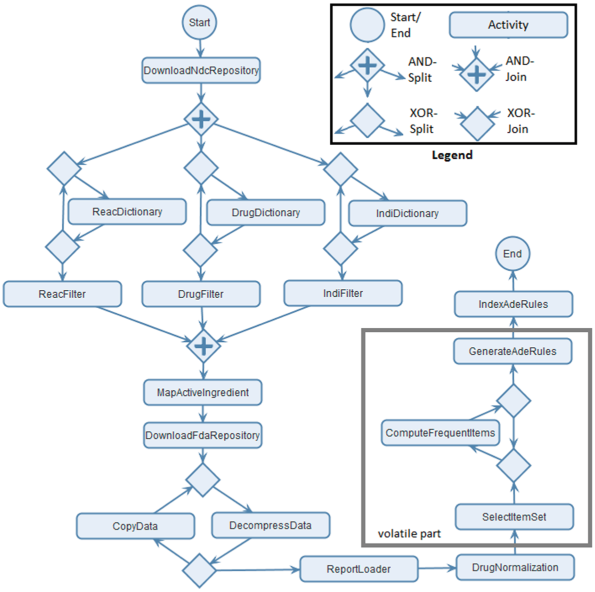

Fig. 3. Originally discovered DrugFusion process

has been identified correctly. However, the dictionary creation can be executed repeatedly until the filtering activity is executed for each type. This shows that the documented process deviates from the actual process execution.

The second phase of “Load Adverse Event Report” ranges in the discovered process from activity “DownloadFdaRepository” to “DrugNormalization”. Instead of a parallel behaviour for “Extracting Data” and “Moving Data” for the types Reactions, Drugs, Indicators, and Demographics, a loop over “DecompressData” and “CopyData” (representing the “Extracting Data” and “Moving Data” activities in the original process) is detected. After further investigation it was revealed that “Extract Data” and “Copy Data” are unified activities that will always fire the same event independent of the type of data to be loaded or moved. Because of this underspecification of events “DecompressData” and “CopyData” is recorded four times for each process execution. This revealed a shortcoming of the DCCM: the algorithm has difficulties to detect a loop over a sequence of activities if in every case the sequence is executed more than once. The reason for this behaviour originates from the general concept of the DCCM: it only looks at the global relation between two activities which in this case is indistinguishable from a normal “loop” (instead of “loop over sequence”), i.e. the footprints for the constructs “loop” and “loop over sequence” look exactly the same[3]. Since the DCCM is not able to distinguish these two constructs in this special case the footprint is interpreted as a normal loop. Note, that this

[3] Both relations “appears before first” and “appears before” are always true for activities within the loop, e.g. for both sequences [EDE] (normal loop) and [$EDED$] (loop over sequence) E “appears before” D and vice versa.

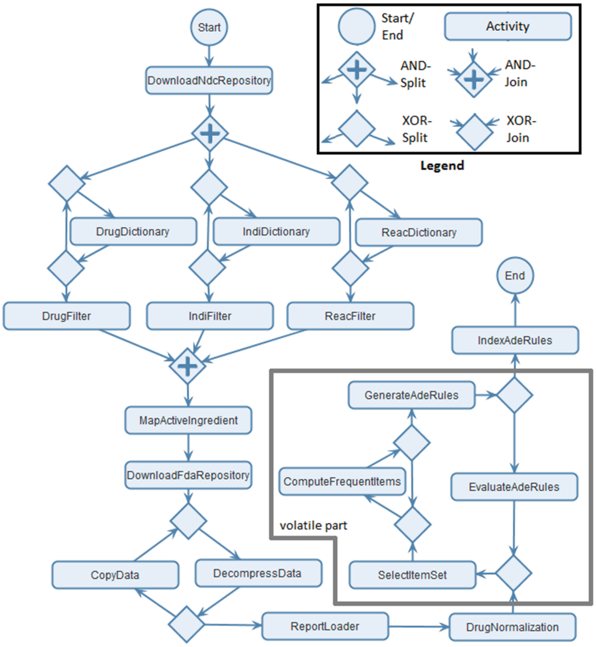

Fig. 4. DrugFusion process with evaluation step

behaviour only occurs if a loop is always looping at least once in every instance execution.

The last phase of the process “Discovery of Adverse Event Rules” is represented in the discovered process by activities from “SelectItemSet” to “GenerateADERules”. Here again, the discovered process deviates from the documented one since “ComputeFrequentItems” can in reality be executed multiple times before moving on in the process. Additionally, the activity “IndexAdeRules” has been detected at the end of the process.

In the second part of the evaluation an extra activity “Rules Evaluation” was introduced that would trigger a repeatedly execution of the “Discovery of ADE Rules” phase of the process. The new activity was detected immediately after the first occurrence and the additional loop behaviour shortly afterwards. The business process model discovered 5 instance executions after the change was implemented is shown in Fig. 4. Now, an additional loop over the activities from “SelectItemSet” to “GenerateADERules” can be identified in the discovered process as well as the additional activity “EvaluateADERules” that is executed every time before the “Discovery of ADE Rules” phase is repeated.

After the change was detected the DP Acquisition Module was immediately notified and as a result the Context Model was updated according to the detected change. With the updated Context Model, the iERM module as well as the Reasoning module of the TIMBUS solution were able to perform their analyses on the up-to-date information of the business process.

5 Conclusion

In this paper we presented the integration of the Dynamic Constructs Competition Miner into the TIMBUS solution as a process monitor and evaluated its application in the context of the use-case DrugFusion from the eHealth domain. The main behaviour of the business process of the DrugFusion process as well as its dynamically changing behaviour was discovered. However, due to the unified execution of some of the activities ("DecompressData" and "CopyData"), events were not uniquely mappable to the activities, which in turn resulted in loop behaviour of a constant length. In the light of this, it was identified that the DCCM has difficulties to detect a loop over a sequence of activities if the loop is repeated at least once in every instance. Although not critical for the risk assessment because the general loop behaviour was detected, this is not an optimal result since underspecified event creation and/or loops of constant length may occur in other use-cases as well. In contrast, other process discovery algorithms not based on global relations but on local relations, e.g. direct neighbours, should have less problems to detect the correct loop but do not comply to the run-time requirement. Resolving the issue of detecting a loop of a constant length over a sequence is considered future work for both, the process monitor in the TIMBUS project as well as the general DCCM. Another next step in the context of the TIMBUS project is the monitoring of changes for other aspects of a business process than the control-flow: performance, resources, state of single instances. If this is achieved a continuous automatic risk analysis can be performed.

Acknowledgments. Project partially funded by the European Commission under the 7th Framework Programme for research and technological development and demonstration activities under grant agreement 269940, TIMBUS project (http://timbusproject.net/).

References

1. Antunes, G., Caetano, A., Bakhshandeh, M., Mayer, R., Borbinha, J.: Using ontologies to integrate multiple enterprise architecture domains. In: Abramowicz, W. (ed.) BIS Workshops 2013. LNBIP, vol. 160, pp. 61–72. Springer, Heidelberg (2013)
2. Butt, T., Cox, A., Oyebode, J., Ferner, R.: Internet accounts of serious adverse drug reactions a study of experiences of Stevens-Johnson syndrome and toxic epidermal necrolysis. Drug Saf. **35**(12), 1159–1170 (2012)
3. Galushka, M., Gilani, W.: DrugFusion - retrieval knowledge management for prediction of adverse drug events. In: Abramowicz, W., Kokkinaki, A. (eds.) BIS 2014. LNBIP, vol. 176, pp. 13–24. Springer, Heidelberg (2014)
4. Galushka, M., Taylor, P., Gilani, W., Thomson, J., Strodl, S., Neumann, M.: Digital preservation of business processes with TIMBUS architecture. In: Proceedings of 9th International Conference on Preservation of Digital Objects IPRES2012, pp. 117–125 (2012)
5. Gilani, W., Redlich, D., Galushka, M., Molka, T., Du, Y.: TIMBUS: Digital preservation for timeless business processes and services. In: 23rd Proceedings of eChallenges Conference (e-2013) (2013)

6. Huang, Y., Lin, S., Chiu, C., Yeh, H., Soo, V.: Probability analysis on associations of adverse drug events with drug-drug interactions. In: BIBE 2007, pp. 1308–1312 (2007)
7. Jin, H., Chen, J., He, H., Kelman, C., McAullay, D., O'Keefe, C.: signaling potential adverse drug reactions from administrative health databases. IEEE Trans. Knowl. Data Eng. **22**(6), 839–853 (2010)
8. Jin, H., Chen, J., He, H., Williams, G., Kelman, C., O'Keefe, C.: Mining unexpected temporal associations: applications in detecting adverse drug reactions. IEEE Trans. Inf. Technol. Biomed. **12**(4), 488–500 (2008)
9. Ji, Y., Ying, H., Dews, P., Mansour, A., Tran, J., Miller, R., Massanari, R.M.: A potential causal association mining algorithm for screening adverse drug reactions in postmarketing surveillance. IEEE Trans. Inf. Technol. Biomed. **15**(3), 428–437 (2011)
10. Ko, R.K.L.: A computer scientist's introductory guide to business process management (BPM). Crossroads J., ACM **15**(4), 4 (2009)
11. Koutkias, V., Kilintzis, V., Stalidis, G., Lazou, K., Nis, J., Durand-Texte, L., McNair, P., Beuscart, R., Maglaveras, N.: Knowledge engineering for adverse drug event prevention: on the design and development of a uniform, contextualized and sustainable knowledge-based framework. J. Biomed. Inf. **45**(3), 495–506 (2012)
12. Krska, J., Cox, A.: Adverse drug reactions. Clin. Pharmacol. Ther. **91**, 467–474 (2012)
13. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing. Addison-Wesley Professional, Reading (2002)
14. Molka, T., Redlich, D., Drobek, M., Zeng, X.-J., Gilani, W.: Diversity guided evolutionary mining of hierarchical process models. In: Genetic and Evolutionary Computation Conference (GECCO 2015), ACM (2015) http://dx.doi.org/10.1145/2739480.2754765
15. Rao, S., Gupta, R.: Implementing improved algorithm over APRIORI data mining association rule algorithm. Int. J Comput. Sci. Technol. **1**, 489–493 (2012)
16. Redlich, D., Molka, T., Gilani, W., Blair, G., Rashid, A.: Constructs competition miner: process control-flow discovery of BP-domain constructs. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 134–150. Springer, Heidelberg (2014)
17. Redlich, D., Molka, T., Blair, G., Rashid, A., Gilani, W.: Scalable dynamic business process discovery with the constructs competition miner. In: Proceedings of the 4th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2014), CEUR 1293, pp. 91–107 (2014)
18. Van Der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
19. Van Der Aalst, W.: Process Mining - Discovery Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011)
20. Weijters, A., Van Der Aalst, W., de Medeiros, A.A.: Process Mining with the Heuristics Miner-algorithm. BETA Working Paper Series, WP 166, Eindhoven University of Technology (2006)