Interconnecting Processes through IoT in a Health-Care Scenario

Monica Vitali
Department of Electronics, Information, and Bioengineering
Politecnico di Milano
Milan, Italy
Email: monica.vitali@polimi.it

Barbara Pernici
Department of Electronics, Information, and Bioengineering
Politecnico di Milano
Milan, Italy
Email: barbara.pernici@polimi.it

***Abstract*—In the context of health and social care, it is important to provide assistance to individuals in their own home by coordinating the activities of several actors. The coordination of processes of different organizations is a complex task which requires continuous attention. Process mining techniques have been widely used to discover hidden information useful to understand the behaviour of a single process. Thanks to the spread of Internet of Things (IoT), this complex activity can benefit from the huge amount of data collected for other purposes. In this paper we propose techniques for exploiting the information available in the environment, considering several sources of information to analyse and interconnect business processes. We apply the proposed approach to a health-care scenario enriched with IoT devices. The proposed techniques allows the discovery of interconnections between processes and external factors which have an impact on them.**

I. Introduction

Modern organizations have to deal with several complex activities usually modelled as business processes. The outcome of a process and its behaviour during its execution can be measured and evaluated in the light of the goals of the organization, defined at several layers (operational, tactical, strategical). The issue of process modelling has been widely studied in the literature, but it is still in evolution [1][2]. Modelling a business process is a non trivial activity and needs a deep knowledge of the organization and its procedures. Modern organizations often interact with external partners to accomplish their job. The ability of an organization to reach its goals can be influenced by such interactions. These interactions, sometimes explicitly modelled, but most of the time implicit, have an effect on the processes executed in the organization and on their results. As an example, a delay in the furniture of raw materials by a supplier affects the production plan. This delay can be due to factors that, if identified in advance, can be prevented. The best way to model this relation would be to design the interactions among processes and to coordinate in this way the interconnections. However, most of the time, organizations and most of all their employees, are not willing to change consolidated procedures for the sake of coordination with external entities. Process Mining [3] is a discipline aiming at extracting information from the process execution, looking at the process log, for improving the business process management. However, it does not take into account other important information as the external factors which can affect its behaviour.

In fact, relations between business processes of cooperating organizations can be direct due to a designed coordination between the two processes under analysis, or indirect when no explicit relation exists. In this last case we talk about hidden relations. Discovering hidden relations is a complex activity which can benefit from the huge amount of data available in the environment where the process is executed. Knowledge about the process can be derived from the process itself (e.g. process logs and traces) such as in process mining, or from other processes with which it interacts. Finally, interactions between processes can be also derived from the environment in which they are executed. Thanks to the spread of Internet of Things (IoT), a huge amount of sensors and devices collect information about several aspects that can have a direct or indirect impact on the processes. This information can be consulted for better managing the processes and their evolution in time, and for understanding their behaviour and results.

In a previous work [4], we introduced the initial idea of Processes in Events (PiE) for discovering interconnections between processes through events of different types, and to exploit them for process analysis. The main contribution of this paper consists in proposing techniques for analysing the impact of external events on the processes of an organization by exploiting the information generated by the surrounding environment and by the processes themselves. In particular, techniques are addressed to discover hidden and indirect interconnections among processes through events from different sources. The information acquired from the analysis can be exploited for deeply understanding the process behaviour, for process enhancement, and for detecting possible causes of failures, preventing and reacting to them through adaptation. The proposed approach is discussed in a health and social care scenario considered in the validation, but can be generalized and applied to other contexts.

The paper is organized as follows. In Sect. II we discuss the state of the art related to our proposal. In Sect. III we describe a health-care scenario where independent organizations need to interact. In Sect. IV we propose techniques for discovering interconnections between processes of different organizations, through the analysis of events. We provide an example of application in Sect. V and we discuss the results in Sect. VI.

II. State of the Art

Business processes are usually monitored through the production of logs, where information about the enactment of the activities composing the process are reported, including start and end time of each activity for each process instance. This information is useful to reason about the process conformance and behaviour. The set of techniques for extracting information about process execution is known as Process Mining. An overview of process mining is discussed in [3], where the author provides motivations and possible applications of process mining techniques, and in [5] where open issues in process mining are discussed, taking into account the context in which the process operates. Some of the open issues are dealing with complex event logs, cross-organizational mining, and combining process mining with other analysis. In [6], the authors use process mining to perform what-if analysis for evaluating possible scenarios, and to learn improvements from past mistakes. All these techniques consider the mapping between events in the log and activities in the process model as given. However, this association can be not trivial, since it is usually not explicit in the log. In [7], the authors propose an iterative semi-automatic methodology for mapping events generated by logs with activities in the business processes, based on the temporal analysis of events and activities occurrences. The issue of evolution in services has been discussed in [8], where it is considered from the technical perspective of the service versioning management. An initial approach to the automatic detection of process evolution is considered in [9], where process mining analyses updated stream of events (activity execution traces) to capture process evolution (Streaming Process Discovery). Authors use Heuristic Miner to build a directed graph, where a node is an activity and an edge is a dependency associated to a measure. In [10], the issue of concept drifts detection and management is addressed performing a temporal analysis of the event logs both at a local and global level. The existing approaches consider only events internal to the processes, without taking into account interconnections between processes and dependencies with events generated by the external environment. In particular, techniques for process mining have been widely applied in the field of healthcare. In [11], authors claim the importance of conducting Business Process Analysis (BPA) of healthcare processes for improving their efficiency and reliability. This analysis should be based on a reliable model. The proposed framework uses process mining techniques such as sequence clustering to derive process models, main paths, variants, and exceptions from logs produced during real executions. In their work, the rich set of events available from the environment is not considered. Recently, in [12], the authors have analysed the problem of detecting interactions among actors within a given context of a series of activities. However, a process based perspective is missing.

Event analysis has been studied also in the area of Complex Event Processing (CEP) [13], looking for event patterns involving the occurrence of multiple, related events [14][15]. In [16], the authors use a systematic approach to derive some additional information from discovering events occurring in a process. This analysis is related to the process enhancement application of process mining. Authors use the information to build decision trees to answer questions about possible process characteristics. A similar approach for process diagnosis is described in [17]. In [18], missing events are reconstructed from other traces in order to extend process logs. In [19], the goal is supporting process manager to change processes according to their context, understanding the relations between internal and external variables.

In this work, the innovative proposal of the research is to investigate the mutual interconnections between different processes and the influence from external factors considering not only events produced by process event logs, but also focusing on the wide set of events produced by devices distributed in a sensing environment in which the processes operate, discovering their relations and influence on the processes. In this way we address some of the open issues in process mining [5], such as cross-organizational mining and combining process mining with other analysis. Our approach is context-aware in the sense that the context is considered for understanding and predicting the behaviour of the processes, but the context is not predefined since it depends on the devices used to monitor the environment in which the processes operate and is subject to modification during the processes life-cycle.

III. Motivating Use Case and Running Example

In this section we illustrate a use case, in which the proposed techniques can be usefully applied, which has inspired this research. The scenario is part of a research project, financed by the Lombardy region in Italy, called Attiv@bili. The aim of the project is to improve the coordination of several organizations cooperating directly and indirectly in the health and social care of citizens, each one with its set of business processes describing their procedures. In this context, an Ambient Assisted Living (AAL) scenario has been developed, where organizations need to share common information coming from several sources helping them in providing a better assistance. In the scenario we have an assisted person living in his own home and needing health and social care for conducting an independent life. To reach this goal, the patient's residence is enriched with sensors. These devices produce events accessible by the different independent stakeholders involved in the health and social care activities. Other events may be manually created and inserted in the system through a web interface by operators, such as reports or registration of a service provided to the patient. Two processes, belonging to two different organizations, have been considered in the validation. The former is a health care process: an operator visits the patient once a day, checks the general state of the patient, helps him in the daily body care procedures, and finally verifies that all the drugs needed are available (if needed an order is submitted to integrate missing drugs). The operator has to register his/her entrance and exit using an access control device installed in the

TABLE I
DEVICE EVENTS IN THE AAL SCENARIO

Type	Description	Source	Category
W1 w_std	watch std read	watch	regular
W2 w_unf	watch unfasten	watch	random
W3 w_off	watch off	watch	random
W4 w_outp	patient out	watch	random
W5 w_motion	patient motionless	watch	random
T1 oper_in	operator entrance	totem	random
T2 oper_out	operator exit	totem	random
I1 ord_reg	regular drug order	web interface	random
I2 ord_urg	urgent drug order	web interface	random

house. The second process is provided by a social cooperative whose operators deliver the drugs needed for the patients.

The two processes are monitored through process logs registering the execution of their activities. There are three devices to monitor the environment: a *wearable wrist watch (d1)*, monitoring physiological parameters and activities of the subject (fall and stillness detection, exit and entrance in the house); a *totem (d2)* for the registration of the exit and entrance of operators; and a *web interface (d3)* for submitting reports about the patient conditions or needs. Each of these devices can produce events of several types. The set of device events is reported in Tab. I.

In this paper we focus our attention on the analysis of the processes to discover hidden relations between processes of independent stakeholders. The acquired knowledge is an important source for understanding the process behaviour and for improving the quality of the results. For instance we can show that the successful drugs delivery process depends not only on the issued orders, but is also related to the monitored state of the patient.

IV. PROCESSES IN EVENTS: MANAGING PROCESS EXECUTION

In this section we propose the techniques that can be adopted in PiE for business process analysis of interconnections in a sensing environment. The proposed approach is based on the application of data mining techniques to the set of events produced by both the processes involved and the sensors distributed in the environment. Knowing these correlations, paths between activities belonging to different processes can be retrieved and used for analysing the process behaviour. The analysis is based on information related to the time in which the events occur. Time-series data analysis enables the discovering of frequent patterns between events occurrences. An overview of techniques available for time-series mining is discussed in [20]. Here we take into account the time component as a dimension for discovering relations between processes, but other dimension can be considered. We consider to operate in a dynamic environment, in which relations are continuously updated since the set of events and their mutual effects can change in time. This aspect has to be accounted during the analysis. We focus on discovering hidden relations between event types using a probabilistic approach for modelling the discovered dependency.

The approach consists of three steps, depicted in Fig. 1, where circles indicate event types generated during the process execution, and diamonds represent event types generated by devices in the environment. The color of the event indicates that it is generated outside the organization. The starting point for the analysis are the logs of events produced by both the processes and the sensors in the observed environment where we have process logs of processes inside and outside the organization, and event logs collecting devices events in an observed environment. The steps are:

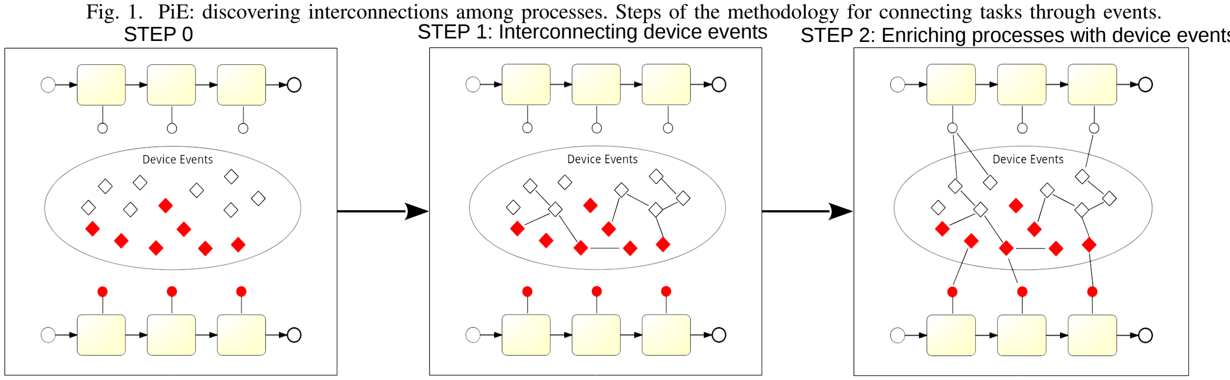
- Step 0: the system is composed of two unrelated processes, in which activities are coupled with the events in the process log associated to them. There is also a pool of events generated by sensors and devices in the environment. We consider the association between process events and activities as given and expressed in the process log. In case the correspondence is not immediate, techniques as [7] can be used.
- Step 1 - Interconnecting device event types: through an analysis of the event log, correlations are discovered between device events of two different types.
- Step 2 - Interconnecting device and process events: for each process in the scenario, connections are discovered between process events and device events types considering their temporal analysis. The outcome of this analysis, combined with the result of Step 1, produces an oriented graph $\mathcal{G}$ of relations among events of processes and devices.

The oriented graph $\mathcal{G}$, incrementally produced after Step 1 and Step 2, provides a deeper knowledge about the behaviour of the processes which can be analysed for understanding process failures and for discovering the causes of the observed results.

The techniques applied in Step 1 and Step 2 to discover the causes that bring to the generation of an event by analysing the sequences of events in the history are similar. We start from the device event instances and we try to discover recurrent patterns between them considering their event type using a generalization mechanism. The relation between events is represented in a graph $\mathcal{G}$ where nodes correspond to event types and an oriented edge expresses a temporal relation, meaning that an instance of an event is raised by the occurrence of an event instance belonging to another type. The edge is also annotated with a value, expressing the confidence of the detected dependency. In the following we describe the algorithm used to build the graph $\mathcal{G}$. As we assume that dependencies vary on time, the graph is continuously updated, taking into consideration present and historical values and providing a mechanism to mitigate noise and errors.

We define $p(et_i|et_j)$ as the probability of observing an event e_i of type et_i after an event instance e_j of type et_j. At each step, the analysis is limited to a time window W,

Fig. 1. PiE: discovering interconnections among processes. Steps of the methodology for connecting tasks through events.



outside which connections are not evaluated. The information discovered in previous steps is considered during the update of the discovered relations, as shown later in this section.

The value of $p_W(et_i|et_j)$ is computed as:

$$p_W(et_i|et_j) = \frac{\sum\limits_{\forall i \in E_i} o_W(e_i \Leftarrow e_j)}{|E_j|} \quad (1)$$

where E_j is the number of events e_j : $e_j.type = et_j$ preceding an event e_i : $e_i.type = et_i$ with $e_i.t - e_j.t < W$. The value $\sum\limits_{\forall i \in E_i} o_W(e_i \Leftarrow e_j)$ evaluates for each case in which an event of type et_i is observed, if it is preceded by an event of type et_j in the considered time window, weighting the observation with the time distance between the two events:

$$o_W(e_i \Leftarrow e_j) = \begin{cases} 1 - \frac{(e_i.t - e_j.t)}{|W|} & if\ e_i.t > e_j.t \\ 0 \ \ if\ e_i.t < e_j.t \end{cases} \quad (2)$$

where $|W|$ is the dimension of the time interval considered in the window used to normalize the importance of the observation. The confidence of this relation can be evaluated as:

$$C_W(et_i|et_j) = p_W(et_i|et_j) - p_W(et_j|et_i) \quad (3)$$

where $C_W(et_i|et_j) \in [-1, 1]$. The function $C_W(et_i, et_j)$ measures unidirectional relations between a couple of events, discarding events which influence each other without a specific temporal predominance. The confidence value associated to a couple of event types can dynamically change in time. Since the system is subject to noise, we update the value of the confidence link taking into account both the most recent value and the historical value computed in the past.

$$C(et_i|et_j) = (1-\alpha) \cdot C_W(et_i|et_j) + \alpha \cdot C_{W-1}(et_i|et_j) \quad (4)$$

As it can be observed, the value of confidence of the relation is updated weighting the present and the history. The importance given to the two components depends on the value of the weight α. Here we want the update to be more dependent on the recent value when the information in the history is old (no recent updates have been performed), and more dependent on the past information when a recent update has been done, so that we can say that the historical value is reliable. We define α as:

$$\alpha = \frac{t\_previous\_update(C(et_i|et_j))}{t\_end(W-1)} \quad (5)$$

where $t\_previous\_update(C(et_i|et_j))$ is the time where the previous update of the confidence between two event types has been performed, and $t\_end(W-1)$ is the end of the previous window interval.

The algorithm for computing and updating the relations between events is shown in Alg. 1. At first, the conditional probability $p_W(et_i|et_j)$ between all the device event types are analysed (Step 1), then each process event type is related to all the available device event types to discover process to device events relations (Step 2). In this second step, a post analysis is needed for avoiding redundancy. In fact, for all activities preceding or following a device event, the confidence value will be significant, but the only one which should be considered as possible cause/effect of the event is the one more proximate to the event generation. This information is implicitly considered in the computation of the confidence, since its value takes into consideration the time expired between the observation of the activity execution and of the event registration. According to this, when using the graph, only the activity event type with the highest conditional probability value is connected to each device event type. Direct relations between events of different processes are not considered, since they can be retrieved through classic process mining techniques. Here we want to emphasize the importance of the information in the execution environment for supporting the process execution. The proposed approach provides also an instrument to automatically detect which are the relevant events in the observed environment. In fact, if at the end of the computation a node of the graph is not connected, it adds no information on the behaviour of the observed processes.

It is worth noticing that the confidence value of a relation is not dependent on the support of the relation (number of time this relation is observed). In fact, we aim to model even relations between rare event categories. The algorithm is able to do that, without removing a rare connection. The confidence

Algorithm 1 PiE - Event Graph

Input: the set of all device events in the window W $\mathbb{ED}_W$
Input: the set of all process events in the window W $\mathbb{EP}_W$
Input: an initial graph $\mathcal{G}$
Output: $\mathcal{G}'$: an updated graph of events

extract $\mathbb{ED}^K_W$ from $\mathbb{ED}_W$ as the set of event types in $\mathbb{ED}_W$
extract $\mathbb{EP}^K_W$ from $\mathbb{EP}_W$ as the set of event types in $\mathbb{EP}_W$
$\forall\ et_i, et_j \in \mathbb{ED}^K_W$
    evaluate $p_W(et_i|et_j)$ and $p_W(et_j|et_i)$
    Compute $C_W(et_i|et_j)$
    Compute $C(et_i|et_j)$
    $\mathcal{G}'$ = updateEdge(et_i, et_j, $C(et_i|et_j)$, $\mathcal{G}$)
$\forall\ et_i, et_j : et_i \in \mathbb{EP}^K_W, et_j \in \mathbb{ED}^K_W$
    evaluate $p_W(et_i|et_j)$ and $p_W(et_j|et_i)$
    Compute $C_W(et_i|et_j)$
    Compute $C(et_i|et_j)$
    $\mathcal{G}'$ = updateEdge(et_i, et_j, $C(et_i|et_j)$, $\mathcal{G}$)
    Compute $C_W(et_j|et_i)$
    Compute $C(et_j|et_i)$
    $\mathcal{G}'$ =updateEdge(et_j, et_i, $C(et_j|et_i)$, $\mathcal{G}'$)

value of a relation is updated only for relations observed in the window W. If the relation is not observed, the old relation remains valid. Otherwise, it is updated with a value depending only on the confidence and not on the support of the relation.

V. Preliminary Results: Ambient Assisted Living

In order to validate the proposed approach, we applied the techniques described in Sect. IV for building a graph of events interconnections. The events considered are coming from the logs of the two processes and from the events produced by the IoT devices in the environments (represented as circles and diamonds in Fig. 2). The experimentation phase of Attiv@bili has not provided enough data to perform a complete analysis. Therefore, starting from the real set of events, from the process models, and from the observation of the real behaviour of the sensors, we have produced synthetic logs integrating real observation with realistic ones. We believe that this approach does not affect the validation since our goal is not to validate the existence of an interconnection between processes in the specific scenario, but that through our approach, we are able to detect such interconnections if they exist. In building the log, we have considered that the health operator process is executed everyday, randomly selecting the starting time between 9:00 a.m. and 5:00 p.m.. For each activity in the process we have considered an average duration and we have used again random generation for changing their starting time in each trace. Finally, we have considered the execution of the drug order activity occurring with a probability of 50%, triggering in this way the production of an event of drug order e^{D3}_{I1} accompanied with a probability of 30% by an urgent drug order event e^{D3}_{I2}. This last event causes the execution of an instance drug delivery process. The starting time of this process has been randomly selected between 10 and 30 minutes after the order placement. The events coming from IoT have been reproduced from the observation of real traces. For instance, every time an operator arrives or leaves, a registration of access is performed (e^{D2}_{T1} and e^{D2}_{T2}) as well as the body care operation requires the patient to take the watch off and the corresponding event is registered (e^{D1}_{W2}). Between the environment events, some are produced only when a specific condition applies, while other are produced regularly (e.g. the watch std event e^{D1}_{W1} is produced regularly every 5 minutes while the watch is properly working).

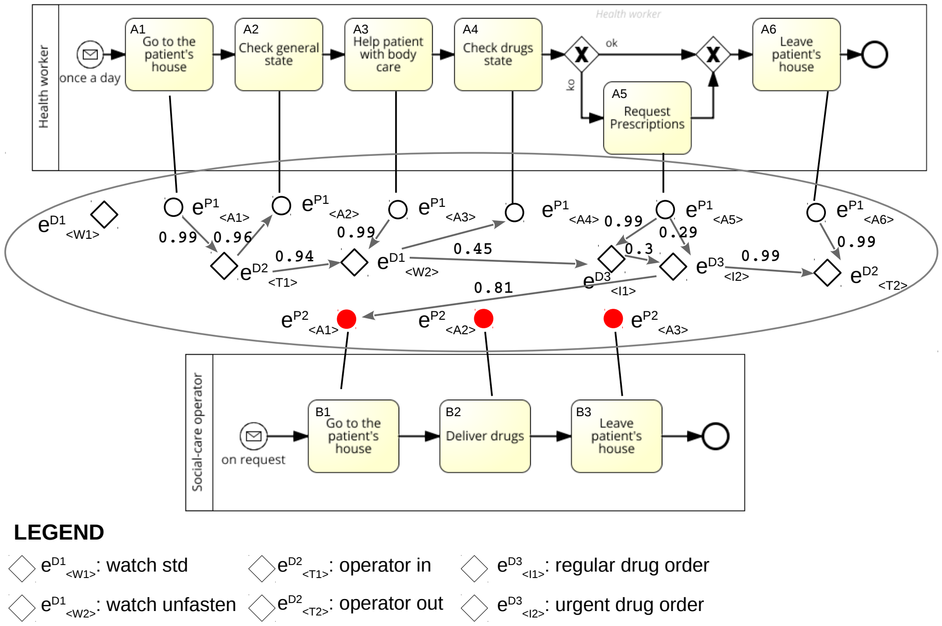
We have performed an analysis of 1000 days, extracted recurrent relations, connected process events to activities and analysed the paths interconnecting the processes. Results are shown in Fig. 2. In the analysis we have not considered connections between events belonging to the same process, since it is not the focus of the analysis in this paper. As it can be seen, using the proposed approach we have discovered strong relations between process and environment events. For instance, entrance and exit of an operator are followed by registrations with a confidence of 0.99. Also, the body care activity is followed by the watch unfasten event with a confidence of 0.99. The analysis reflects the probability of execution of the *Request Prescriptions* activity and the probability of generation of an urgent order event. Event if this event is rare, whenever it is observed, a new instance of the Social Care process is started with a confidence equal to 0.81. As a consequence, we derive the information that the social care operators are active only for delivering urgent drugs. This enables the interconnection of two processes, otherwise independent. As it can be observed, the watch standard event e^{D1}_{W1} is not related to any other event. This is due to the nature of this event, which is produced on a regular base and is observed both before and after the other events in the considered window. Therefore, no causality can be discovered for this kind of event. The algorithm is able to detect this condition and to behave properly.

To prove the importance of considering the context in which processes are executed, we have performed the same analysis removing the information provided by the urgent order device event from the dataset. In this case, no relation can be discovered between the two processes and they appear to be completely unrelated. It is clear from this result that through additional information provided by the environment in which the processes are executed, it is possible to discover relations which are hidden when this information is missing.

VI. Final Remarks

In this work we presented the PiE techniques for analysing processes in a health care scenario and discovering hidden interconnections through the events produced in an IoT environment. Events, coming from different sources, provide important information for understanding and managing the process execution, by enriching standard process mining techniques with a new set of approaches. We proposed an approach based on the temporal events analysis for building a graph of dependencies between the whole set of events, showing how this additional knowledge can be used to discover interconnections

Fig. 2. Discovering interconnections in an AAL real case study



between processes of cooperating organizations. We validated our approach showing the effectiveness in discovering existing relations in an AAL scenario.

In future work we are willing to apply our approach to complex scenarios with several interacting processes and a wider set of environment event types. Finally, we want to refine the presented technique for discovering relations between events with other kinds of analysis, looking also at the information contained in the event instances (e.g. location and values), not considered in this paper. We believe that an integration of several techniques can provide a more general solution.

REFERENCES

[1] M. Dumas, W. M. Van der Aalst, and A. H. Ter Hofstede, *Process-aware information systems: bridging people and software through process technology*. John Wiley & Sons, 2005.

[2] M. Weske, *Business process management: concepts, languages, architectures*. Springer Science & Business Media, 2012.

[3] v. d. W. Aalst, "Process mining: Overview and opportunities," *ACM Transactions on Management Information Systems*, vol. 3, no. 2, p. 1, 2012.

[4] M. Vitali and B. Pernici, "Pie-processes in events: Interconnections in ambient assisted living," in *On the Move to Meaningful Internet Systems: OTM 2015 Workshops*, 2015, pp. 157–166.

[5] W. M. Van Der Aalst and S. Dustdar, "Process mining put into context," *IEEE Internet Computing*, no. 1, pp. 82–86, 2012.

[6] W. M. van der Aalst, W. Z. Low, M. T. Wynn, and A. H. ter Hofstede, "Change your history: Learning from event logs to improve processes," in *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on*, 2015, pp. 7–12.

[7] T. Baier, A. Rogge-Solti, J. Mendling, and M. Weske, "Matching of events and activities-an approach based on behavioral constraint satisfaction," in *SAC 2015, Salamance, Spain*, 2015.

[8] V. Andrikopoulos, S. Benbernou, and M. P. Papazoglou, "On the evolution of services," *Software Engineering, IEEE Transactions on*, vol. 38, no. 3, pp. 609–628, 2012.

[9] A. Burattin, A. Sperduti, and W. M. van der Aalst, "Control-flow discovery from event streams," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 2420–2427.

[10] R. P. Bose, W. M. Van Der Aalst, I. Zliobaite, and M. Pechenizkiy, "Dealing with concept drifts in process mining," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 154–171, 2014.

[11] Á. Rebuge and D. R. Ferreira, "Business process analysis in healthcare environments: A methodology based on process mining," *Information Systems*, vol. 37, no. 2, pp. 99–116, 2012.

[12] A. Senderovich, J. Mendling, A. Rogge-Solti, A. Mandelbaum, and A. Gal, "The ROAD from Sensor Data to Process Instances via Interaction Mining," in *28th International Conference on Advanced Information Systems Engineering CAiSE16*, 2016, p. in press.

[13] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Computing Surveys (CSUR)*, vol. 44, no. 3, p. 15, 2012.

[14] A. Margara, G. Cugola, and G. Tamburrelli, "Learning from the past: automated rule generation for complex event processing," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, 2014, pp. 47–58.

[15] Y. Taher, M. Parkin, M. P. Papazoglou, and W.-J. van den Heuvel, "Adaptation of web service interactions using complex event processing patterns," in *Service-Oriented Computing*. Springer, 2011, pp. 601–609.

[16] M. d. Leoni, F. M. Maggi, and W. M. P. V. d. Aalst, "An alignment-based framework to check the conformance of declarative process models and to preprocess event-log data," *Inf. Syst.*, vol. 47, pp. 258–277, 2015.

[17] R. P. Bose and W. M. van der Aalst, "Discovering signature patterns from event logs," in *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, 2013, pp. 111–118.

[18] F. Mannhardt, M. de Leoni, and H. A. Reijers, "Extending process logs with events from supplementary sources," in *Business Process Management Workshops*. Springer International Publishing, 2014, pp. 235–247.

[19] K. Ploesser, J. Recker, and M. Rosemann, "Supporting context-aware process design: Learnings from a design science study," in *Business Process Management Workshops*, 2010, pp. 97–104.

[20] T.-C. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.