Automated Process Recognition Architecture for Cyber-Physical Systems

Dragos Repta*, Ioan Stefan Sacala*, Mihnea Moisescu*, Ioan Dumitrache*, and Aurelian Mihai Stanescu*
* University Politehnica of Bucharest, Faculty of Automatic Control and Computers, Bucharest, Romania
repta_dragos@yahoo.com, ioan.sacala@acse.pub.ro ioan.dumitrache@acse.pub.ro, mihnea.moisescu@upb.ro

***Abstract*—Systems that can tightly integrate physical with virtual components have represented a priority for the Systems Engineering Research. Research efforts have been concentrated in domains such as: Internet of Things, Internet of Services and recently in the domain of Cyber Physical Systems. The main objective of this paper refers to the investigation of applying process mining techniques on streams of data collected from existing sensing networks. A method based on automated planning problem is proposed and implemented.**

I. Introduction

In recent years, advances in various fields such as pervasive computing, sensor and computing technology, plan and activity recognition (especially human activity recognition) have opened several paths to achieve novel solutions for applying process mining techniques in various new settings.

In this paper, the practical aspects of using the process mining techniques are investigated, specifically the case in which data is collected from existing sensor networks. The objective will be to extract the process model that best explains the observed behavior – the result of applying a process discovery techniques – with the minimal effort on the part of the experts required to set up the system.

This problem can be aligned to challenges from the Process Mining Manifesto [1], specifically:

- C1 – Finding, Merging and Cleaning Event Data
- C7 – Cross-Organizational Mining – as it will be presented in the case study
- C11 – Improving Understandability for Non-Experts and the related guiding principle “GP5: Models Should Be Treated as Purposeful Abstractions of Reality”

II. Research Questions and Related Approaches

The Cyber-Physical System paradigm addresses the relation between “Cyber” (virtual, computation environment) and “Physical” (natural environment) components of ICT enabled systems.

Adopting the CPS paradigm in the context of enterprise systems must consider the following:

- the components heterogeneity in regard to interoperability and composition;
- the system components complexity and thus new modelling techniques
- complex interaction between human operators, equipment and software systems;
- stochastic nature of processes;
- Dynamic reconfiguration of systems
- Measure system state with the aid of sensing systems and changing state with the aid of actuation systems
- Adaptive control systems and networked control systems

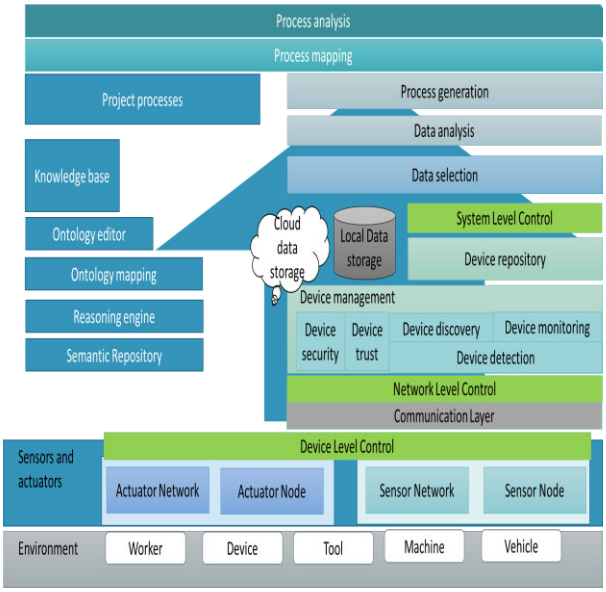

Figure 1. Generic CPS system architecture

In the field of process mining, from the early stages of its evolution, there has been a keen interest in the practical aspects and benefits of the developed methods and techniques, specifically in the case of process discovery techniques.

In [2] the authors analyze the usage of process mining techniques in a hospital setting the application of process discovery techniques in various Dutch organizations – public (municipalities) and private. The authors of [3] present an overview of the (perceived) success of process discovery techniques. Closest to this investigation – industrial setting and an existing sensor network - is the case study present in [4] involving the extraction of a process model from a wafer machine.

As all these practical (even simulated) case studies reveal – even though the process discovery algorithms have matured to the point of achieving their intended objective with minimal input from the human operators (parameter tuning), the main issues revolve around the pre-processing stages.

The “canonical” process discovery problem is defined in [5] as extracting a process model from an event log without any a-prori information and is the focus of the process mining research field. Event logs are considered the central data structure in the process mining field and are bags (multi-sets – sets in which each element has an associated cardinality) of event sequences, where each sequence represents a process realization / instance (or case according to the process mining terminology). Each sequence elements have at least a reference to an activity action type, although usually other event attributes are available such as the timestamp, the actors and/or resources involved in performing the action, etc. It should be noted that not all attributes can be used to extract useful information using the current process mining techniques.

The authors of [6] discuss some of the issues surrounding the resulting process models obtained through the usage of process discovery methods. They employ a "process model as a map" metaphor mentioning that the resulting process models should meet the expectations of the end-user, and, similar to maps, some of the aspects of the process (representing events or activities) should be abstracted-away(omitted) of aggregated into higher level representation.

This line of thinking aligns with the problem considered (scenario considered) in this paper – data collected from existing sensor networks – in that it becomes obvious that a pre-processing stage is required in order to achieve the abstraction level desired or required by the end-user, which in most cases is represented by a business analyst or system stakeholder. For example, in a logistic scenario, the data collected from a set of RFID readers connected through a public informatics infrastructure will be represented by a stream of events corresponding to passing of various objects through monitored gates. In this case, it is obvious that the business analyst / end-user / etc. would not be able to extract any useful information from a process model constructed directly from this data stream. As such, a set of data transformation steps will need to be employed to aggregate some RFID event patterns into higher level activity reference – for example a sequence involving 3 RFID events – one referring to a product entering a packing area, one from a container entering the same area and one from the container exiting the packaging area may refer to a single action.

Some of the methods of dealing with this abstraction problem, proposed in the field of process mining, have focused on the usage of clustering techniques [7] and extraction of common patterns [9][8]. These methods have the advantage of requiring limited domain-dependent, expert provided background explicit knowledge (usually in the form of user / operator intervention) thus simplifying their adoption.

However, as mentioned above, the final, discovered process model should be tailored according to needs of the end-user, and the previously mentioned methods rely on metrics based on event correlation frequency that might not suffice or provide relevant information to business experts. Furthermore, the limited expert input that these methods accept significantly hinders their flexibility and their ability to provide the correct level of abstraction and/or perform the desired aggregations in every application.

As an alternative to these methods, in this paper we investigate the possibility of reusing techniques developed in the fields of plan and activity recognition. Several reviews of these research fields, proving the growing interest in them, are presented in [10][11][12][13][14].

A key knowledge artifact, common to the majority activity and plan recognition methods is represented by an action library that specifies the necessary (partially-ordered) sequences of events that correspond to an activity or plan. But methods of this type have an important limitation caused by the considerable effort required in order to build these knowledge bases. Furthermore, in most cases, the created knowledge bases will be domain specific, being unable to be transferred from one system implementation to another.

An alternative explored in this paper involves a generative solution, in which the plans / partially-ordered sequences are constructed during the execution of the activity recognition procedure using simpler action models. Such an approach is similar to the one presented in [15], where the possible set of plans for an agent is constructed based on an action library represented using the PDDL notation. The described algorithm is capable of identifying the distribution of the achievable objectives based on a set of observations regarding the already performed actions and a cost function.

Thus, the problem of action recognition can be cast into a problem of constructing the (partially-ordered) sequence of actions whose execution determines a trajectory in the state space explaining the sequence of collected observations. Such an approach can be implemented using „model checking" techniques [16] that allow the exploration of a system's state space in order to check the validity of a set of properties, usually expressed as a temporal logic formula.

These techniques have been used to solve the automated planning problem, this scheme being investigated in a number of papers [4][5][6][7]. Broadly speaking, solving the automated planning problem using a "model checker" involves the conversion of the planning domain into a system model. Most "model checking" techniques are capable of providing a counter-example in case the verified (temporal) property is satisfied representing the solution to the planning problem.

A related approach can be found in the research into the (discrete event) system diagnosis problem. The system diagnosis problem refers to the identification of all behaviors of a system that can explain a set of observations and determining which of them are correct and faulty, a faulty behavior usually containing a set of forbidden actions.

Given this problem's similarity to model checking, system diagnosis techniques can be considered for solving the activity recognition problem. Such an approach is described in [10] and accepts as a system model a PDDL domain. Another similar solution for the system diagnosis problem is presented in [12]. The authors highlight the similarities between this problem and the automated planning problem and propose a PDDL transformation schema for it. After this transformation, the problem is solved using existing planners, leveraging the heuristics developed for efficient state space exploration developed in the last two decades. Similar approaches, relying on existing automated planners and PDDL transformation schemas are presented in [13]. Closely related to [13], in [14] the authors develop the LTS++ application, tackling the problem of generating hypotheses regarding the behavior of a dynamic system explaining a set of observations.

Another issue that should be solved before applying the process discovery techniques on the collected event stream involves Identifying the process instance associated to each event.

Existing informational systems processing sensor observations are not usually „process-aware" meaning that they do not include at least a partial description of the semantics of current business processes that can be used, at least in this situation, to properly associate each collected event to its corresponding process instance; For example, in the case of a RFID reader network, they won't be aware of the executed logistic processes; Coming back to the previous example involving RFID events, the data processing subsystem should be able to extract(not necessarily partition) from the collected stream the subsets of events corresponding to each process realization, from the perspective of the end user. As such, if the process model should be discovered from the perspective of an individual product, all RFID events pertaining to it (even those referring to the package containing the product) should be identified.

Compared to the activity recognition problem, in the case of process instance identification, the number of candidate solutions is significantly smaller and usually require the involvement of a human expert either directly – guiding a candidate solution removal procedure - and/or indirectly by specifying domain-dependent heuristics.

Given that a fully-automated solution for the instance identification cannot be developed, for the proposed system, a semi-automatic method based on event-correlation rules, as proposed in [16] will be used. This procedure involves a 3-step rule generation algorithm (atomic rules, conjunctive rules and disjunctive rules) guided by an (easily extensible) set of configurable heuristics and user decisions – the user sets the parameters of the heuristic rules (max. process instance length) or directly removes irrelevant correlation rules after each step.

A similar approach is presented in [17], based on a similar set of heuristic rules but lacking support for (pruning) composite correlation rules (conjunctive and disjunctive).

For the process discovery problem, in last two decades a significant number of methods have been proposed. A detailed description of this problem and proposed solutions is beyond the scope of this paper, but comprehensive reviews are available in [9]. For the proposed solution however, a class of methods described in [11] capable of accepting additional information – specifically the independence relationship between the log elements - is preferred.

III. Methodology

This analysis led to the development of a proposed solution whose structural architecture is depicted in Fig. 1. Several existing approaches for the activity recognition and process instance identification problem have been analyzed.

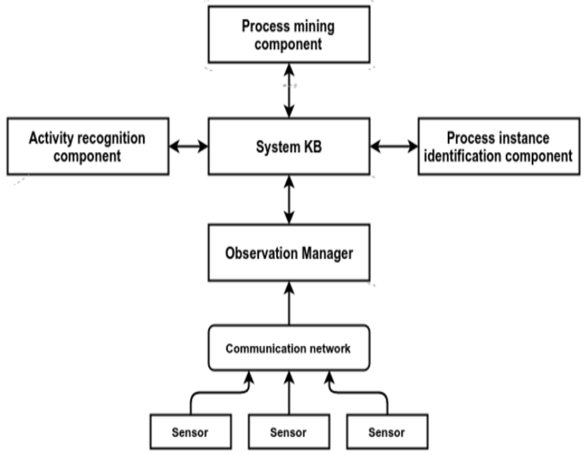

Figure 2. Structural architecture of proposed system

Semi-automatic completion of the planning domain and problem facilitated through the usage of an ontology as a mean of representing both the system's knowledge base and its working database – the information conversion steps are reduced to executing SPARQL-(Construct) queries and feeding the result sets to a templating engine

The usage of ontologies has been inspired by the research into Internet of Things middleware solutions

This analysis led to the development of a proposed solution whose structural architecture is depicted in Fig. 1. Several existing approaches for the activity recognition and process instance identification problem have been analyzed.

In order to solve the activity recognition problem, a method based on its transformation into an automated planning problem was chosen and implemented. The generation of the PDDL domain and problem files are done in automatic manner based on the information from the system's work ontology and a set of pre-determined action definition that constitute the domain-specific expert knowledge. The functional diagram of the instance identification component. The information flow depicted in this figure is summarized next.

Data is collected from an existing sensor network and is routed to the "Observation Manager" component where it is "semantically lifted" using a set of domain-specific transformation rules and the resulting OWL assertions are added to the system's "working" ontology; The actual process of retrieving this data is beyond the scope of this paper but it has been an intensely investigated subject in the Internet-Of-Things research community; The functionality of the "Observation manager" component is similar to that of the previous solution. As with the older version, for each sensing device, in the "Sensing Interface Manager" component, a "Sensing Interface Component" will be created. This interface is responsible with the transformation of the primary collected data into a set of uniform assertions that will be added to the system's work ontology. To initialize this interface, information regarding the sensing network's topology, stored in the system's work ontology can be used.

The system's knowledge base is represented as an OWL ontology backed by a relational database; Its terminological component is an extension of the SSN (Sensor Network Ontology) and DUL(Dolce Ultralite

Ontology) ontologies and should be used by domain experts to encode all the static aspects of the monitored environment; Its assertional component is used by the rest of the system's components as an information storage and retrieval medium(??);

The other three components of the system deal with each of the 3 sub-problems – process instance identification, activity recognition and process model discovery, the latter one using the results of the first ones. It should be noted that the instance identification and activity recognition steps can be performed in any order – depending on the domain(??). All these components extract information using domain-dependent SPARQL queries.

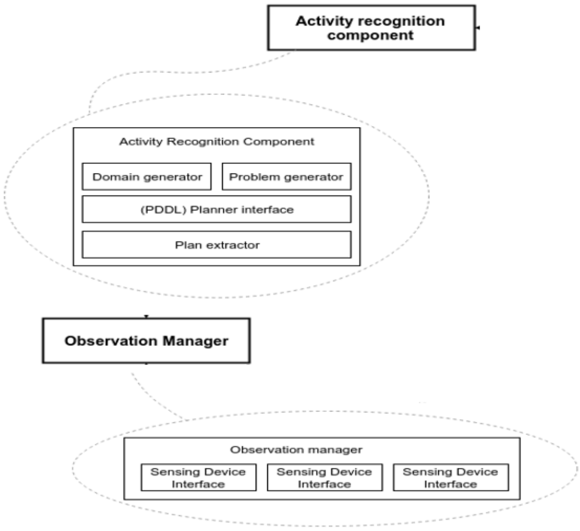

Figure 3. Components architecture for activity recognition

The generation of correlation relations is based on the semi-automatic iterative method mentioned in the previous section. This method requires input from a human operator in order to prune candidate event correlation relation candidates and to provide additional, domain-dependent, heuristic rules for removing erroneous relations (ex.: based on known constraints such as a maximum process duration or other temporal restrictions). After the final event correlation rule is identified it can be used to create "ProcessInstance" OWL individuals in the system's knowledge base used to related events or actions pertaining to the same process instances. It should be noted that this is not an event partitioning procedure as an event (or activity in the case of performing this step on the results of the activity recognition step) as the same item can be shared by more than one instances – for example, in the case of an RFID stream of events and processes referring to a single component, the events related to the container in which they are transported must be associated will all the components' process instances.

In order to determine the (partially-ordered) sequence of activities that corresponds to an event stream, the activity recognition component is used. As mentioned earlier, the events can refer the either the collected stream of data or the result of the instance identification procedure. The activity recognition procedure is performed using an external planner (LPG-td, SGPlan and MIPS-XXL have been tested) by generating a domain and planning problem file (in PDDL format) using the event data fetched from the system's KB. A set of simple syntactic processing rules are required to generate these files. The resulting plan, after a post-processing phase that removes redundant actions represents the sequence of activities that explain the observed behavior in the monitored system. As in the case of the process instance identification phase, the result of this phase is used to create a set of "Activity" OWL individuals in the system's KB. Another important result from this step is a set of independence relations between the items of plan (activities??) that will be exploited in the process discovery phase.

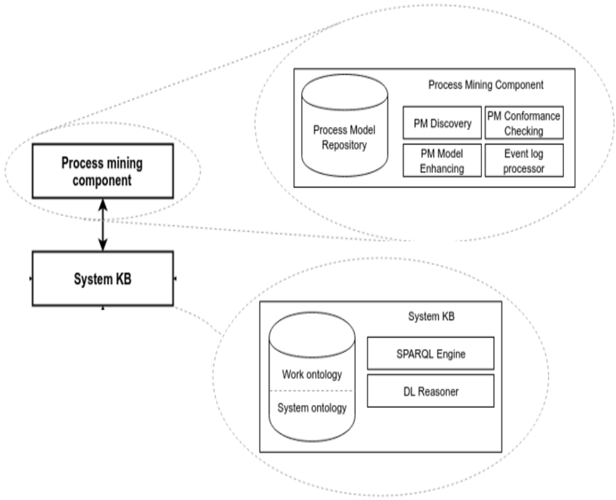

Figure 4. Components architecture for PROCESS MINING

Finally, the process model discovery component relies on the information produced by the other two components. As mentioned earlier there have been a multitude of methods developed in the process mining research field, but in this application, we preferred the method due to its ability to consider not only an event log but also a set of independence relations determined during the activity recognition phase.

IV. SOLUTION/DISCUSSION

In order to evaluate the proposed solution, a small test environment has been developed that simulates the interactions between 7 actors in a supply chain: 2 suppliers, one producer, a distribution center, 2 retailers and a transporter. For this system, several RFID readers, placed in various areas of this environment and GPS receivers placed on vehicles provide the input event stream. Notice that just detecting the movement of an object from one area to another will not be enough to infer the actions performed on it. For this, a PDDL description of the dynamic domain is required during the action recognition phase. In order to simulate the behavior of this system, a colored Petri Net was developed in CPN tools [http://cpntools.org/] and the event sequence was extracted from its simulation

The next step consists in the sematic lifting of the simulated observations, required before adding them into the system's work ontology. For each observation, using a set of syntactic rules, OWL Individuals are created. It is assumed that during the system's deployment, based on expert provided knowledge, the ontology has been populated with entities pertaining to the static aspects of the modeled environment.

During the next phase, the sequence of semantically lifted observations is transformed into high-level actions using the planning-based action recognition method. In the case of this supply chain, due to the presence of packing and unpacking actions, no correlation rule can be created in order to correctly partition the events into process instances. Consequently, the instance identification phase is executed based on the results obtained during the action recognition step. The observations are added to an existing (expert provided) set of action definitions as special, "constraint" actions that force the planner to generate a solution (a sequence of action instances) capable to explaining the collected events. After registering the resulting action instances in the system's work ontology, the instance identification phase can be performed. Using an implementation of the algorithm mentioned in the previous section, a set of candidate correlation rules have been identified. A human operator is required to specify the parameters of the heuristic rules guiding the candidate generation process, to purge irrelevant atomic and conjunctive rules and finally to select the correlation rule that best fits the process logic.

The application of the user selected event correlation rule leads to the creation of new OWL Individuals corresponding to the identified process instances that can be used to build the "event log" data structure, required for process discovery. For this, relatively small case, the POD tool was used which, besides the event log, can also accept a set of independence relationships between detected actions. These relations can be extracted from the plans generated during the action recognition phase. The result of running the POD tool on the simulated data is the Petri Net (process model) depicted in Fig 5

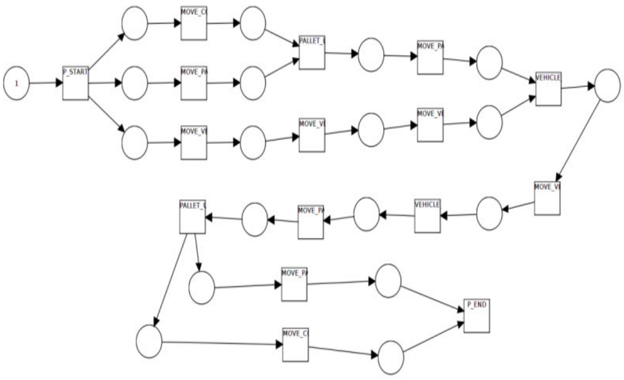

Figure 5. Process model

CONCLUSIONS

A solution to the problem of applying process mining techniques to events collected using sensor networks has been discussed, focusing on three main components: event representation, instance identification and activity recognition components

The need to use the event data acquired from sensors, in a conventional format, lead to the development of a system knowledge base, allowing domain-independent implementations.

Although the proposed solution proposes a domain-independent representation, it will need expert knowledge for correlation rules and domain definition

REFERENCES

[1] 1. Van Der Aalst, W., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., ... & Burattin, A. (2011, August). Process mining manifesto. In International Conference on Business Process Management (pp. 169-194). Springer Berlin Heidelberg.

[2] 2. Rebuge, Á., & Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. Information systems, 37(2), 99-116.

[3] 3. Mans, R., Reijers, H., Berends, H., Bandara, W., & Rogier, P. (2013). Business process mining success. In Proceedings of the 21st European Conference on Information Systems. AIS Electronic Library (AISeL).

[4] 4. Rozinat, A., de Jong, I. S., Gunther, C. W., & van der Aalst, W. M. (2009). Process mining applied to the test process of wafer scanners in ASML. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 39(4), 474-479.

[5] 5. Van Der Aalst, W. (2012). Process mining: Overview and opportunities. ACM Transactions on Management Information Systems (TMIS), 3(2), 7.

[6] 6. Van der Aalst, W. M. (2009). Process-aware information systems: Lessons to be learned from process mining. In Transactions on petri nets and other models of concurrency II (pp. 1-26). Springer Berlin Heidelberg.

[7] 7. De Medeiros, A. K. A., Guzzo, A., Greco, G., Van Der Aalst, W. M., Weijters, A. J. M. M., Van Dongen, B. F., & Saccà, D. (2007, September). Process mining based on clustering: A quest for precision. In International Conference on Business Process Management (pp. 17-29). Springer Berlin Heidelberg.

[8] 8. Bose, R. J. C., & van der Aalst, W. M. (2009, September). Abstractions in process mining: A taxonomy of patterns. In International Conference on Business Process Management (pp. 159-175). Springer Berlin Heidelberg.

[9] 9. Günther, C. W., & Van Der Aalst, W. M. (2007, September). Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In International Conference on Business Process Management (pp. 328-343). Springer Berlin Heidelberg.

[10] 10. Chen, L., & Khalil, I. (2011). Activity recognition: Approaches, practices and trends. Activity Recognition in Pervasive Intelligent Environments, 1-31.

[11] 11. Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. ACM Computing Surveys (CSUR), 43(3), 16.

[12] 12. Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology, 18(11), 1473-1488.

[13] 13. Ye, J., Dobson, S., & McKeever, S. (2012). Situation identification techniques in pervasive computing: A review. Pervasive and mobile computing, 8(1), 36-66.

[14] 14. Krüger, V., Kragic, D., Ude, A., & Geib, C. (2007). The meaning of action: a review on action recognition and mapping. Advanced Robotics, 21(13), 1473-1501.

[15] 15. Ramırez, M., & Geffner, H. (2010). Probabilistic plan recognition using off-the-shelf classical planners. In Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010) (pp. 1121-1126).

[16] 16. Baier, C., Katoen, J. P., & Larsen, K. G. (2008). Principles of model checking. MIT press.

17. Rozsnyai, S., Slominski, A., & Lakshmanan, G. T. (2011, July). Discovering event correlation rules for semi-structured business processes. In Proceedings of the 5th ACM international conference on Distributed event-based system (pp. 75-86). ACM.