Refining Business Process Ontology Model with Invisible Prime Tasks using SWRL rules

Kelly R. Sungkono
Department of Informatics Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
kelsungkono@gmail.com

Riyanarto Sarno
Department of Informatics Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@if.its.ac.id

Nurul Fajrin Ariyani
Department of Informatics Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
nurulfajrin@if.its.ac.id

***Abstract*—A process model is a model that is formed based on user design or event logs. The process model is used as a guidance for process performance analysis. In fact, there are several problems that occur in the modeling processes from the event log. One of them is detecting invisible prime tasks. An invisible prime task is a task that does not exist in the event log, but it is required in the process model to describe the actual processes. Many research suggested methods for detecting invisible prime tasks. However, none of them applied their method in the ontology. This research proposes a method of indicating skip and redo invisible prime tasks in ontology automatically. The method is written in SWRL rule. The experiment shows that the method can detect invisible prime tasks based on the event logs in BPMN ontology as the process model. This method also becomes the pioneer of invisible task detection in the ontology.**

Keywords—BPMN; invisible prime task; process discovery; ontology; SWRL

I. INTRODUCTION

Process models have been widely utilized by analysts in assessing process performance and addressing various related activity issues [1]. The areas of issues are fraud [2-3], business [4], communication [5], and environment [6]. Process models can be drawn manually by analysts or can be generated based on the event logs. Many research already suggested methods for depicting process model [1], [7-9].

During the implementation, there are several emerging issues. Several of them are dynamic business processes and invisible prime tasks. Invisible prime tasks are additional tasks that are not represented in the event log, but they are added to depict the processes clearly [9]. Those issues become challenges in modeling business process model.

To deal with dynamic business processes, building process model in ontology is chosen. It is because ontology is built in domain concept and relations, so it is more flexible in handling dynamic processes [10]. Many research proposed ontology structure for representing business processes, such as [11] and [12].

From various modeling process model methods, there are several methods that focus on detecting invisible prime tasks. Those methods are Alpha# [9], CHMM for Invisible Tasks [8], and Alpha$ [13]. Alpha# and Alpha$ methods detect invisible prime tasks by several rules that utilized relation of activities. CHMM for Invisible Tasks method detected invisible tasks based on Coupled Hidden Markov Model of activity relations. The similarities of those methods are the methods used static process models and the methods did not implement on the process ontology model.

This research proposed a method, namely Ontology with Invisible Tasks method for updating the process ontology model with invisible prime tasks automatically based on the event log. Ontology with Invisible Tasks method forms several SWRL rules and utilizes BPMN ontology and Public Workflow Ontology. The method was evaluated in Student Registration process ontology model. The main goal of the evaluation is adding appropriate invisible prime tasks in the process ontology model based on the event log on Student Registration.

II. LITERATURE REVIEW

A. Invisible Prime Tasks

Several conditions of processes need additional tasks for shows behavior of processes in process models. Those conditions are skip, redo, and switch [9]. Skip condition occurs when there is a relation of activities which jump over other activities based on a process model. Redo condition occurs when executed previous activities are implemented again as the next activities of others. The switch condition occurs when activities on one of the choice sequences have relationship with activities in another choice sequence.

The additional tasks that help depiction of those conditions in the process model are named invisible prime tasks or invisible tasks [9]. The simple examples of invisible tasks that used for depicting the conditions are shown in Fig. 1. The black rectangles of Fig. 1 denote the invisible prime tasks.

B. BPMN Ontology

Business Process Model and Notation or BPMN is included in standard types of process models. This standard type facilitates process delineation using graphic forms [14]. Several forms of BPMN that often used to build a process model are Events, Activities, Gateways, and Sequence Flows.

978-1-5386-2827-0/17/$31.00 ©2017 IEEE

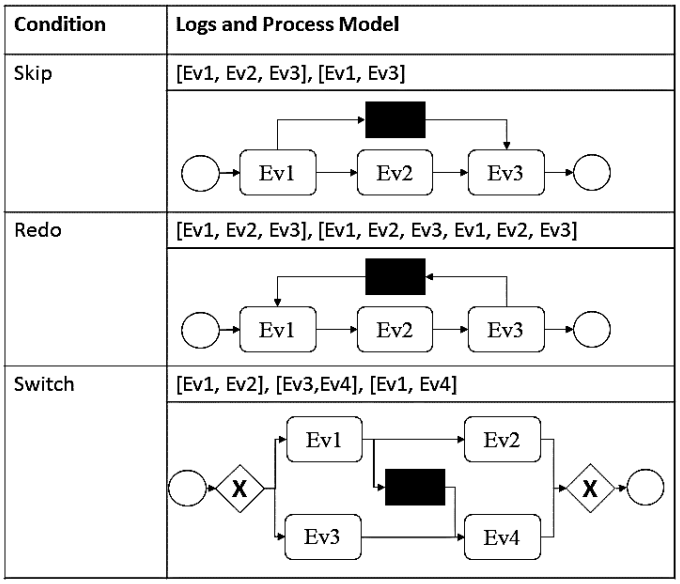

Fig. 1 Simply examples of invisible tasks.

Several forms of the events are a start event and an end event. Those are signified the beginning and the end of a process. Both of them are denoted as circles. Subsequently, activities are tasks or jobs that are implemented by resources in the processes. Activities are denoted as boxes. Afterward, gateways are starting gates or end gates of parallel processes. There are several types of parallel processes, such as XOR process, OR process, and AND process [1]. Gateways are denoted as diamond and there are different diamonds for every type of parallel processes. To connect activities, events or gateways, BPMN presents sequence flows. The flows are denoted by arrows.

With the rapid development of an ontology, [11] provides BPMN ontology structure. The names of classes in this ontology are adjusted with elements in Business Process Model and Notation.

C. Publish Workflow Ontology

Publish Workflow Ontology is an ontology which represents workflow of processes in ontology [15]. The forerunner of the establishment of this ontology is to store the journal publication processes.

During the implementation, this ontology undergoes some renewal. One of the renewal is adding class WorkflowExecution [16]. The goal of adding this class is representing the event log along with the workflow in the ontology. The general description of Publish Workflow Ontology is shown in Fig. 2. The classes of this ontology are denoted by rectangles and the object properties are denoted by arcs.

Because this research used BPMN ontology to represent the workflow of processes (process model), class Workflow and class Step are ignored. Class Step was replaced by Class Activities of BPMN Ontology.

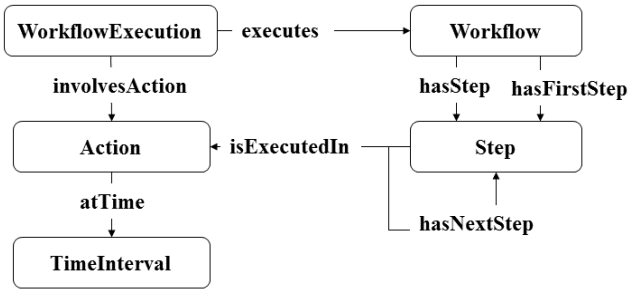

Fig. 2 Publish workflow ontology.

hasSon(?k, ?r), hasWife(?r, ?s) ->
hasDaughterInLaw(?k, ?s)

Fig. 3 SWRL rule for finding daughter in law.

D. SWRL

SWRL (Semantic Web Rule Language) is a group of notations to build rules in the ontology [17]. This language combines OWL DL with OWL Lite Sublanguages. The principle of this language is cause and effect. If the cause is right then the effect is right. The simple example of SWRL is described in Fig. 3. hasSon() and hasWide() represents the cause and hasDaughterInLaw() represents the effect. The meaning of a rule in Fig. 3 is individual s is daughter in law of individual k if individual r is the son of individual k and individual s is the wife of individual r.

SWRL (Semantic Web Rule Language) consists of several atoms that can be utilized in the rules, such as P(x,y), C(x), DifferentForm(x,y), and SameAs(x,y). DifferentForm which is used to distinguish between individuals is utilized in the proposed method of this research.

III. Proposed Method

A goal of this research is adding skip or redo invisible tasks in the process ontology model based on event logs using rules. The method which is named Ontology with Invisible Tasks Method for realizing the goal is consists of several steps. The general steps of the method are shown in Fig. 4.

A. Providing Invisible Prime Tasks

The first step is providing invisible prime tasks which will be used in SWRL rules for refining business process ontology model. Those invisible tasks are defined as individuals of Class Invisible_Task and the connectors between the invisible tasks and activities in process model are defined as individual of Class BaseElement. Class Invisible_Task is a class that is added by this research in business process ontology model.

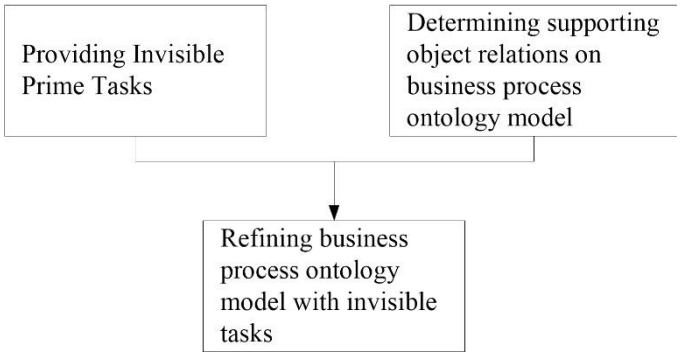

Fig. 4 General steps of ontology with invisible tasks method.

TABLE I. OBJECT PROPERTIES IN CLASS INVISIBLE_TASK

Domains	Object Properties	Ranges	Definition
Task, Invisible_Task	*outgoing*	Base Element	A task or an invisible task as the end activity of a relation has a connector.
Task, Invisible_Task	*incoming*	Base Element	A task or an invisible task as the pioneer activity of a relation has a connector.
Invisible_Task	*InvTaskof*	Task	Invisible task is the selected invisible task when the relation of task requires it.

A number of individuals in Class Invisible_Task is same as a number of activities in the business process model. The object properties that are attached to each individual in Class Invisible_Task are described in Table I.

B. *Determining Supporting Object Relations*

To facilitate in determining relationships between activities in a business process ontology model, several rules are proposed. The rules are defined in Table II. The first rule is used for determining sequence relation and other rules are used for determining parallel relations. This research appended hasStepPrevious as inverse object property of hasStepNext.

C. *Refining Business Process Ontology Model with Invisible Tasks*

A way to refining business process ontology model with Invisible Tasks is connecting corresponding invisible tasks with activities in the process ontology model. The corresponding invisible tasks are determined based on the processes in the event log.

This research proposed a rule to refining the business process ontology model. The result of the rules of a business process ontology model and the invisible prime tasks in the first step are utilized in this rule. The rule which is written in SWRL format is defined in Fig. 5. This rule checks relations of activities in the event log ontology that are not represented

TABLE II. RULES OF BUSINESS PROCESS ONTOLOGY MODEL

No.	Rules
1	Task(?a), Task(?b), outgoing(?a, ?arc_go), outgoingTask(?arc_go, ?b) -> hasStepNext(?a, ?b)
2	Task(?a), Task(?b), ExclusiveGateway(?ig), incoming(?ig, ?arc_in), outgoing(?ig, ?arc_go), incomingTask(?arc_in, ?a), outgoingTask(?arc_go, ?b) -> hasStepNext(?a, ?b)
3	Task(?a), Task(?b), InclusiveGateway(?ig), incoming(?ig, ?arc_in), outgoing(?ig, ?arc_go), incomingTask(?arc_in, ?a), outgoingTask(?arc_go, ?b) -> hasStepNext(?a, ?b)
4	Task(?a), Task(?b), ParallelGateway(?ig), outgoing(?ig, ?arc_go), outgoingTask(?arc_go, ?a), outgoing(?ig, ?arc_go1), outgoingTask(?arc_go1, ?b), DifferentFrom(?a,?b) -> hasStepNext(?a, ?b)

WorkflowExecution(?case), involvesAction(?case, ?act1), hasNextAction(?act1, ?act2), hasActivity(?act1, ?act_prev), hasStepNext(?act_prev, ?temp_act_next), hasActivity(?act2, ?act_next), hasStepPrevious(?act_next, ?temp_act_prev), DifferentFrom (?act_prev, ?temp_act_prev), DifferentFrom (?act_next, ?temp_act_next), Invisible_Task(?inv), InvTaskof(?inv, ?act_prev), incoming(?inv, ?e1), outgoing(?inv, ?e2) -> incoming(?act_next, ?e2), outgoing(?act_prev, ?e1)

Fig. 5 Rule of refining business process ontology model with invisible tasks.

in the business process ontology model and adds skip invisible tasks or redo activity tasks to depict those relations. Connectors between invisible tasks and those activities are depicted by SEQ individuals of Class BaseElement, object properties *incoming* and object properties *outgoing*.

IV. RESULT AND ANALYSIS

The experiment in this research used a registration process model which was built in the ontology. Fig. 6 shows the registration process model in BPMN format and Fig. 7 shows the classes of the registration process ontology model. The cases of the utilized event log ontology and the description of relations of those activities are shown in Fig. 8 and Table III.

Based on the Table III, the relations of activities in italic form are not depicted in the process ontology model. The addition of invisible tasks is needed to overcome this situation. The rules in Ontology with Invisible Tasks method were built in Protégé application and this experiment utilized Pellet reasoner to relate appropriate Pellet reasoner to relate appropriate invisible tasks with activities in the ontology model.

The results of the rules in this experiment are shown in Fig. 9 and Fig. 11. The updated ontology model in BPMN model format for all cases are described in Fig. 10. The results in Fig. 9 and Fig. 11 shows that the proposed rules can successfully depict the relation of activities in the cases using invisible prime tasks.

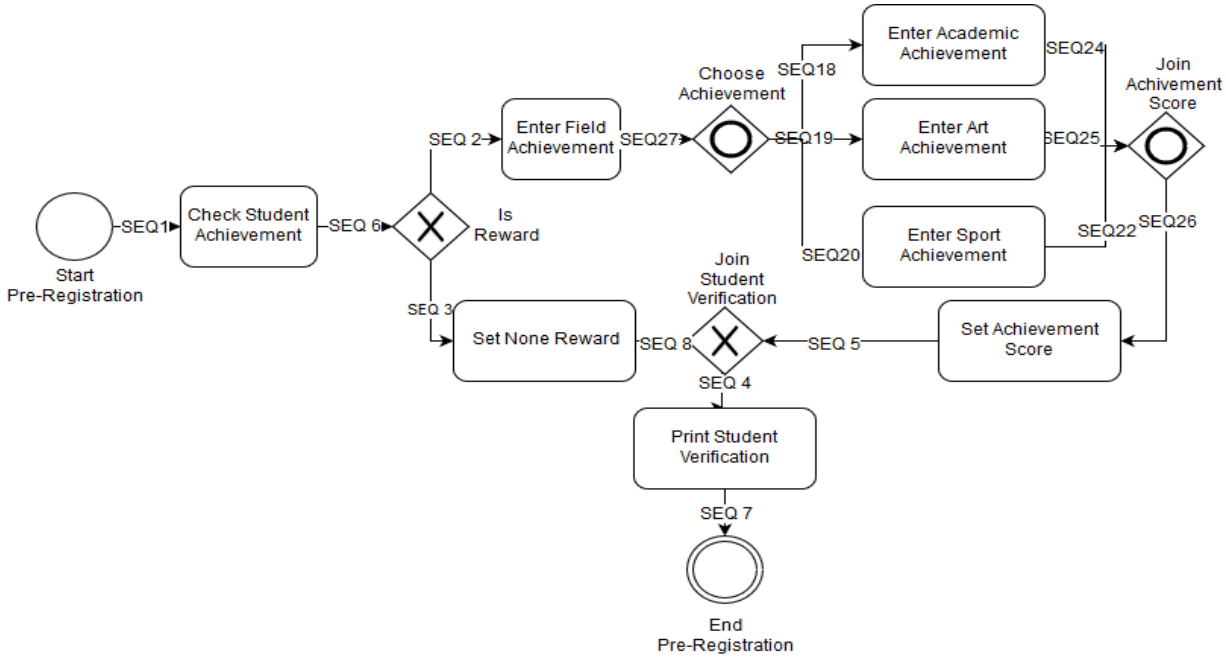

Fig. 6 Registration process model in BPMN model format.

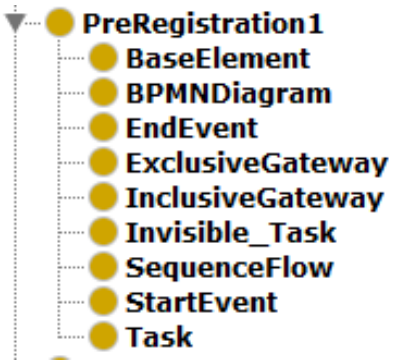

Fig. 7 Classes of registration process ontology model.

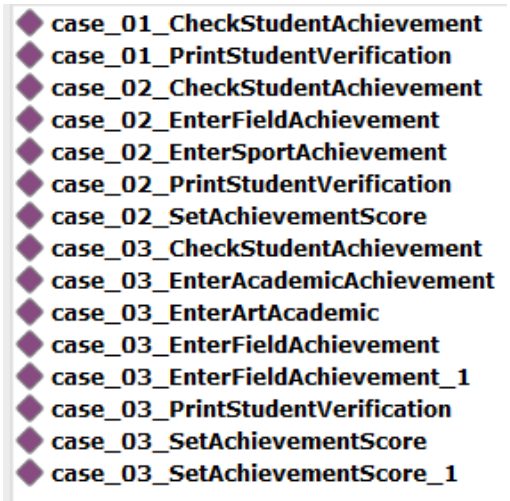

Fig. 8 Cases in registration event log ontology.

TABLE III. DESCRIPTION OF RELATIONS OF ACTIVITIES IN THE REGISTRATION EVENT LOG ONTOLOGY

Case	Relation of Activities
1	*Check Student Achievement -> Print Student Verification.*
2	Check Student Achievement -> Enter Field Achievement -> Enter Sport Achievement -> Set Achievement Score -> Print Student Verification.
3	Check Student Achievement -> Enter Field Achievement -> Enter Academic Achievement -> *Set Achievement Score -> Enter Field Achievement* -> Enter Art Achievement -> Set Achievement Score -> Print Student Verification.

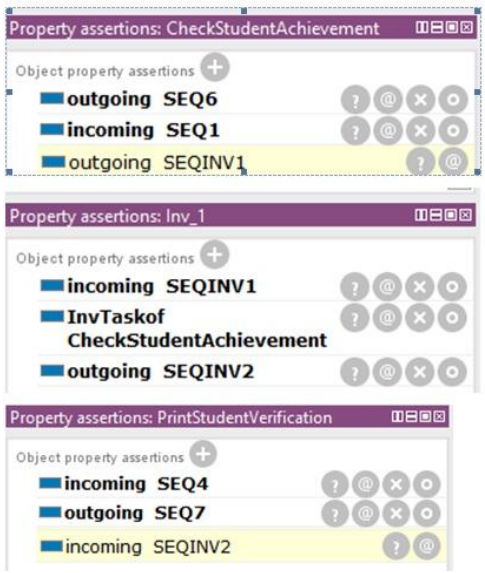

Fig. 9 Result of experiment using case 1 in protégé.

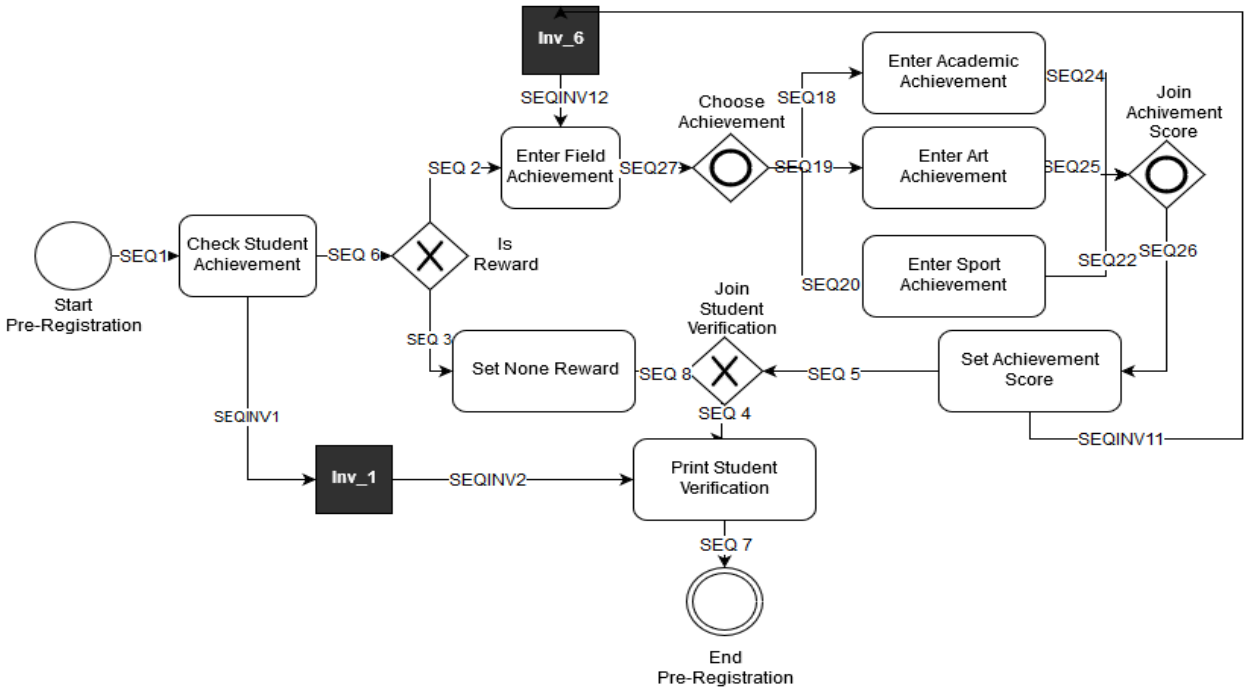

Fig. 10 Registration process model with invisible prime tasks.

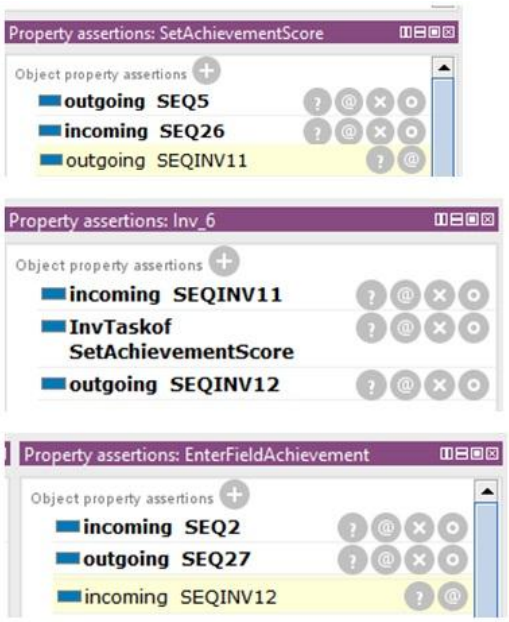

Fig. 11 Result of experiment using case 3 in protégé.

V. Conclusion

This research refines the business process ontology model with invisible tasks by Ontology with Invisible Tasks method. The method consists of several SWRL rules and additional individuals in Class Invisible_Task. Ontology with Invisible Tasks method was tested using a few processes in real-event log and Registration process ontology model.

The outputs in this experiment show that the rules in Ontology with Invisible Tasks method can automatically identify appropriate invisible prime tasks in the business process ontology model. Moreover, the obtained invisible tasks can simplify users and analysts to detect the changes of processes in the business process ontology model. Ontology with Invisible Tasks method is the pioneer method for detecting invisible tasks in the process ontology model because all of the other methods were applied in static process models, such as Petri Net, YAWL or BPMN.

Rule of Ontology with Invisible Tasks method forms excessive invisible tasks on parallel relations. Therefore, further development of this research is minimizing the use of invisible tasks on parallel relations. Moreover, applying the rules in a huge real-event logs and converting process ontology models to static process model can be the subject of future research.

Acknowledgment

Authors give a deep thank to Institut Teknologi Sepuluh Nopember and the Ministry of Research, Technology and Higher Education of Indonesia for supporting the research.

References

[1] R. Sarno and K. R. Sungkono, "Hidden Markov Model for Process Mining of Parallel Business Processes," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 4, pp. 290–300, 2016. [Online]. Available: https://doi.org/10.15866/irecos.v11i4.8700

[2] D. Rahmawati, M. A. Yaqin, and R. Sarno, "Fraud detection on event logs of goods and services procurement business process using Heuristics Miner algorithm," in *2016 International Conference on Information Communication Technology and Systems (ICTS)*, 2016, pp. 249–254. [Online]. Available: https://doi.org/10.1109/ICTS.2016.7910307

[3] R. Sarno, R. D. Dewandono, T. Ahmad, M. F. Naufal, and F. Sinaga, "Hybrid Association Rule Learning and Process Mining for Fraud Detection.," *IAENG International Journal of Computer Science*, vol. 42, no. 2, 2015.

[4] O. T. Baruwa and M. A. Piera, "Identifying FMS repetitive patterns for efficient search-based scheduling algorithm: A colored Petri net approach," *Journal of Manufacturing Systems*, vol. 35, pp. 120–135, Apr. 2015. [Online]. Available: https://doi.org/10.1016%2Fj.jmsy.2014.11.009

[5] V. R. L. Shen, H.-Y. Lai, and A.-F. Lai, "The implementation of a smartphone-based fall detection system using a high-level fuzzy Petri net," *Applied Soft Computing*, vol. 26, pp. 390–400, Jan. 2015. [Online]. Available: https://doi.org/10.1016%2Fj.asoc.2014.10.028

[6] A. Sanaa, S. Ben Abid, A. Boulila, C. Messaoud, M. Boussaid, and N. Ben Fadhel, "Modeling hydrochory effects on the Tunisian island populations of Pancratium maritimum L. using colored Petri nets," *Biosystems*, vol. 129, pp. 19–24, Mar. 2015. [Online]. Available: https://doi.org/10.1016%2Fj.biosystems.2015.02.001

[7] R. Sarno, Y. A. Effendi, and F. Haryadita, "Modified Time-Based Heuristics Miner for Parallel Business Processes," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 3, pp. 249–260, 2016. [Online]. Available: https://doi.org/10.15866/irecos.v11i3.8717

[8] R. Sarno and K. R. Sungkono, "Coupled Hidden Markov Model for Process Mining of Invisible Prime Tasks," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 6, pp. 539–547, 2016. [Online]. Available: https://doi.org/10.15866/irecos.v11i6.9555

[9] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, and J. Sun, "Mining process models with prime invisible tasks," *Data & Knowledge Engineering*, vol. 69, no. 10, pp. 999–1021, Oct. 2010. [Online]. Available: https://doi.org/10.1016%2Fj.datak.2010.06.001

[10] S. Fan, Z. Hua, V. C. Storey, and J. L. Zhao, "A process ontology based approach to easing semantic ambiguity in business process modeling," *Data & Knowledge Engineering*, vol. 102, pp. 57–77, Mar. 2016. [Online]. Available: https://doi.org/10.1016%2Fj.datak.2016.01.001

[11] C. Natschlager, "Towards a BPMN 2.0 Ontology," in *Lecture Notes in Business Information Processing*, Springer Berlin Heidelberg, 2011, pp. 1–15. [Online]. Available: https://doi.org/10.1007%2F978-3-642-25160-3_1

[12] D. Gašević and V. Devedžić, "Petri net ontology," *Knowledge-Based Systems*, vol. 19, no. 4, pp. 220–234, Aug. 2006. [Online]. Available: https://doi.org/10.1016%2Fj.knosys.2005.12.003

[13] Q. Guo, L. Wen, J. Wang, Z. Yan, and P. S. Yu, "Mining Invisible Tasks in Non-free-choice Constructs," in *Lecture Notes in Computer Science*, Springer International Publishing, 2015, pp. 109–125. [Online]. Available: https://doi.org/10.1007%2F978-3-319-23063-4_7

[14] M. Chinosi and A. Trombetta, "BPMN: An introduction to the standard," *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 124–134, Jan. 2012. [Online]. Available: https://doi.org/10.1016%2Fj.csi.2011.06.002

[15] A. Gangemi, S. Peroni, D. Shotton, and F. Vitali, "A pattern-based ontology for describing publishing workflows," in *Proceedings of the 5th International Conference on Ontology and Semantic Web Patterns-Volume 1302*, 2014, pp. 2–13.

[16] A. Gangemi, S. Peroni, D. Shotton, and F. Vitali, "The Publishing Workflow Ontology (PWO)," *Semantic Web*, vol. 8, no. 5, pp. 703–718, Apr. 2017. [Online]. Available: https://doi.org/10.3233%2Fsw-160230

[17] I. Horrocks *et al.*, "SWRL: A semantic web rule language combining OWL and RuleML," *W3C Member submission*, vol. 21, p. 79, 2004.