SWRL Rules for Identifying Short Loops in Business Process Ontology Model

Yutika Amelia Effendi, Riyanarto Sarno
Department of Informatics
Faculty of Information Technology, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
e-mail: yutika.effendi@gmail.com, riyanarto@if.its.ac.id

***Abstract*—A set of linked activities which is produced for specific service is the definition of the business process model. A business process model is usually generated from event logs of an organization or is formed based on user design. In real life, business process model is used as estimation, prediction, calibration, and optimization of process performance. However in fact there are several problems that occur in the steps of modeling the business process model from event log, such as short loops, invisible prime tasks, and non-free choice. Not all process have a sequence of activities occuring one way (from start activity to end activity). There is a condition called short loops. This condition happens where an activity or several activities in a process are executed more than one time before terminate the process. Several research solved the problem of short loops using matrix and formula. However, none of them applied their method in ontology. This research proposes a method of identifying short loops in ontology. The method is written in SWRL rule. The experiment shows that the method can identify and automate identification of short loops based on the event logs in Business Process Ontology Model as the business process model. This method also becomes the pioneer of short loops detection in ontology.**

Keywords—Business Process Model and Notation; Short Loops; Ontology; SWRL; Semantic Web; Protege

I. INTRODUCTION

A set of linked activities which is produced for specific service is the definition of the business process [1]. Business process has information about where and when the activities are executed, input and output of activity, initial condition before activity is executed and final condition after activity is executed. The characteristics of the business process itself are: have a specific purpose, a specific input, a specific output, utilize resource, have an activity that can be executed in a certain order [2], and it can involve more than one organization.

Business process model can represents business process correctly. There are a lot of ways to represent business process model, such as BPMN, UML, Causal Net, BPEL, EPC, PNML, etc [1]. Each type of model has different characteristic, such as Petri Net uses token to connect the activity in the business process model.

In real life, business process model is used as estimation, prediction, calibration, and optimization of process performance [2]. However in fact there are several problems that occur in the steps of modeling the business process model from event log, such as short loops, invisible prime tasks, and non-free choice. Not all process have a sequence of activities occuring one way (from start activity to end activity). There is a condition called short loops [3]. This condition happens where an activity or several activities in a process are executed more than one time before terminate the process. Length-one-loop and length-two-loop are two types of short loops.

Business process model can be obtained from the event logs using process discovery techniques. Event logs are a document that stores records of production process activities. Process discovery has several algorithms: Alpha, Alpha+, Alpha ++, Genetic Algorithm, Heuristics Miner, Temporal Activity-based Process Discovery, etc. Each algorithm has its own advantages and disadvantages. Thus, the algorithm needs to be modified to overcome its shortcomings. The Alpha+ algorithm [4], which is an improvement of Alpha can discover short loops, but still lacks noise, incompleteness, and conditional OR behavior. Modified Time-based Heuristics Miner algorithms can overcome noise, discover short loops, and conditional OR.

Although there are several research solved the problem of short loops using matrix [3] and formula [4], none of them applied their method in ontology [5]. This research proposes a method of identifying short loops in ontology. The method is written in SWRL rule. The experiment shows that the method can identify and automate identification of short loops based on the event logs in Business Process Ontology Model as the business process model. This method also becomes the pioneer of short loops detection in ontology.

The remainder of this paper is organized as follows. In Section II, we review some research which have been done in identifying short loops and BPMN ontology. We present our proposed method in Section III. The results and analysis are explained in Section IV. Last, this paper is concluded with conclusions and sketched future works in Section V.

II. LITERATURE REVIEW

Section II gives a short summary of Short Loops, BPMN ontology, Publish Workflow Ontology, and SWRL.

A. Short Loops

Not all process have a sequence of activities occuring one way (from start activity to end activity). There is a condition where an activity or several activities in a process are executed more than one time before terminate the process. This condition is called as loop. There are two types of loop: long loop and short loop [3].

Long loop is a condition where more than two activities executed repeatedly in a process. Meanwhile, short loop is a condition where one or two activities executed repeatedly. If only one activity was repeated, this condition is called as length one loop. Otherwise, if two activities was repeated, the condition is called as length two loop. Fig.1 and Fig.2 show the example of a process model having a length one loop condition and a length two loop condition.

There are an arc in Figure 1 that connects activity E with itself. It signifies length one loop condition. Meanwhile, an arc that connects activity E to activity D in Figure 2 illustrates a condition in which those activities can be executed repeatedly. This condition is called a length two loop condition.

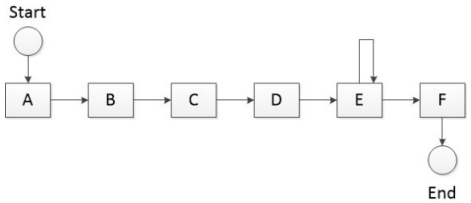

Fig.1. A process model containing a length one loop condition

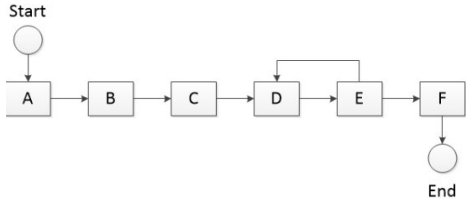

Fig.2. A process model containing a length two loop condition

B. BPMN Ontology

There are two types of BPMN ontologies have been researched before [6]. sBPMN is the first BPMN ontology. This ontology specifies a semantically enhanced BPMN [7]. The ontology is adapted the final release of BPMN 1.0 and developed within the SUPER project1. The classes of this ontology resemble the elements of BPMN. Flow Objects, Connecting Objects, Swimlanes, Artifacts and Processes are several types of categories divided based on the classes. The whole of sBPMN ontology contains 50 axioms and 95 classes.

C. Francescomarino, et al. introduced the second BPMN ontology which is adapted BPMN 1.1 [8]. This specification of second BPMN ontology is also followed the BPMN elements. This ontology contains 95 classes, 70 data properties, 439 class axioms and 108 object properties. Flow Objects, Connecting Objects, Swimlanes, Artifacts and Processes are also several types of categories divided based on the classes. In other publications, for example [9], this research is proposed the ontology called BPMNO. Meanwhile, A. Gangemi, et al. explained that the ontology does not include a description of all properties in the BPMN specification [10]. This reason is caused because some describe the execution behavior of the process. Another reason is some cannot be explained based on the limitations of OWL, such as default values. BPMNO is also not focused on modeling the dynamic behaviour of diagrams or how the flow proceeds within a process.

Therefore, former releases of BPMN become the guidance of development of both sBPMN and BPMNO. Based on this background, so their classes are also mainly adapted the BPMN elements. However, in BPMN 2.0, BPMN metamodel provided the basis for the development of BPMN 2.0. The advantages of this new ontology are having different and more extensive structure which better visualize the BPMN specification.

C. Publish Workflow Ontology (PWO)

Publishing Workflow Ontology (PWO) is developed to accommodate workflow requirements. The PWO is entirely based on the ontology patterns [11]. The description of the logical steps in a workflow is allowed in this ontology, such as the process of a document publication. Each step may require one or more activities (actions or events) which take place in a particular phase of the workflow. PWO has been implemented based on the ontology patterns [12]. Table I explains patterns have been used in PWO.

TABLE I. A SUMMARY OF THE MAIN ENTITIES OF PWO WHICH HAVE RELATIONS TO SOME ENTITIES INTRODUCED IN THE PATTERNS USED

PWO entity	Pattern entity
Workflow	Plan (basic plan description, via basic plan)
WorkflowExecution	PlanExecution (basic plan execution, via basic plan)
Step	Task (task role, via control flow)
Action	Action (task execution and plan execution) TimeIndexedSituation (time-indexed situation)
hasStep	definesTask (basic plan description)
hasFirstStep	definesTask (basic plan description)
executes	satisfies (basic plan)
hasNextStep	directlyPrecedes (sequence)
hasPreviousStep	directlyFollows (sequence)
needs	isDefinedIn o describes (basic plan description)
produces	isExecutedIn (o hasParticipant) (task execution and participation)

D. SWRL

An expressive OWL-based rule language is known as the Semantic Web Rule Language (SWRL) [7]. Users can build and write rules in SWRL which will be described in terms of OWL concepts. Using SWRL, the deductive reasoning capabilities are more powerful than using stand alone OWL. Semantically, SWRL and OWL are built based on the same description logic foundation which can provide similar strong formal guarantees when executing inference.

A SWRL rule consists of two parts, namely an antecedent part (*body*) and a consequent part (*head)*. Both antecedent part and consequent part consist of positive conjunctions of *atoms*:

atom ^ atom .... -> atom ^ atom

All the atoms in the antecedent part must be true so that a SWRL rule can be run and has a meaning. Not only atoms in the antecedent part, but also atoms in the consequent part must be properly defined. However, negated atoms or disjunction are not supported by SWRL. An expression of the form of an atom:

p(arg1, arg2, ... argn)

where *p* referred to a predicate symbol, *arg1*, *arg2*, ..., *argn* referred to the terms or arguments of the expression.

Classes, properties or data types of OWL can be included with the predicate symbols in SWRL. Individuals or data values of OWL can be categorized as arguments or variables referring to them. All variables in SWRL are universally quantified, with their scope limited to a given rule [8].

E. Algorithms used to identify short loops

Research related to identification of short loops, such as Modified Time-based Heuristics Miner algorithm [3] and Alpha+ algorithm [4]. The summary of them will be explained in Table II.

TABLE II. A SUMMARY OF TWO ALGORITHMS THAT HAVE BEEN USED TO IDENTIFY SHORT LOOPS

Method	Explanation of Method	Advantages and Disadvantages
Alpha+ algorithm	- Define the activity relation in the event log - Create a free Transition set of L1L or Length One Loop - Initialise a set, named F_{L1L} that contains the arcs of Length One Loop	Advantage: Alpha+ is an improvement of Alpha which can discover short loops Disadvantage: Must redefine the completeness of event logs, so that increase time complexity of algorithm
Modified Time-based Heuristics Miner algorithm	- Define the activity relation in the event log - Use the formula at step 2 (checking short loops) to discover the short loops	Advantage: Can discover short loops and parallel OR Disadvantage: Must redefine the activity relation, so that increase time complexity of algorithm

III. PROPOSED METHOD

This subsection presents the proposed method and SWRL rules for business process model and short loops which are used in this research.

A. Steps to Identify the Short Loops

Although there are several research solved the problem of short loops using matrix and formula, none of them applied their method in ontology. Therefore, this research proposes short loops identification algorithm by using ontology because ontology has object properties which are used to define all relations of individuals (in this case is activity).

There are four main steps of this research:

1. BPMN process model is converted into Business Process Ontology Model.
2. Event logs used in this research based on Publish Workflow Ontology. Event logs are not the result of the convert process but they adopt the pattern from the workflow.
3. Identify all activities in business process model which contain short loops (length one loop and length two loops).
4. Build rules in terms of OWL concepts to provide more powerful deductive reasoning capabilities and new knowledge using SWRL rules. Two ontologies are used in making SWRL rules.
5. The rules are process models that has been updated.

B. SWRL Rules

Rules are a way of expressing additional things that can be inferred from the dataset, in this case is event logs. Some inferences can be gleaned from the ontology model itself, but others may not be expressible in the ontology language (usually OWL) and require a more functional representation.

SWRL can also be seen as an extension to OWL. Semantically it layers nicely on top of OWL and is normally used for statements that OWL cannot express. Syntactically it does not layer that nicely on top of OWL because it introduces a completely new syntax (note e.g. the explicit variables).

In this research, we present two rules; rules for process model and rules for identification of short loops. These rules can identify length one loops and two loops automatically in Protege (an open source ontology editor and framework for building knowledge management system).

Rule for process model:

- Task(?a), Task(?b), outgoing(?a, ?arc_go), outgoingTask(?arc_go, ?b) -> hasStepNext(?a, ?b)
- 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)
- 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)

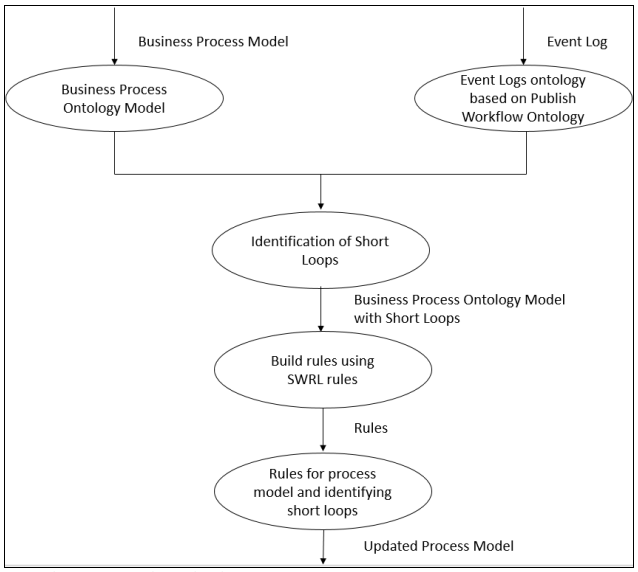

Fig.3. The proposed method

- Task(?a), Task(?b), InclusiveGateway(?ig), outgoing(?ig, ?arc_go), outgoingTask(?arc_go, ?a), outgoing(?ig, ?arc_go1), outgoingTask(?arc_go1, ?b), DifferentFrom (?a, ?b) -> hasStepNext(?a, ?b)

Rule for detecting short loops:

- SWRL rule for identifying Length One Loop

  WorkflowExecution(case_01), involvesAction(case_01, ?act1), hasNextAction(?act1, ?act2), hasActivity(?act1, ?act_prev), outgoing(?act_prev, ?arc1), outgoingTask(?arc1, ?temp_act_next), hasActivity(?act2, ?act_next), incoming(?act_next, ?arc2), incomingTask(?arc2, ?temp_act_prev) -> incoming(?act_next, ?e2), outgoing(?act_prev, ?e1)

- SWRL rule for identifying Length Two Loop

  WorkflowExecution(case_02), involvesAction(case_02, ?act1), hasNextAction(?act1, ?act2), hasActivity(?act1, ?act_prev), outgoing(?act_prev, ?arc1), outgoingTask(?arc1, ?temp_act_next), hasActivity(?act2, ?act_next), incoming(?act_next, ?arc2), incomingTask(?arc2, ?temp_act_prev) -> incoming(?act_next, ?e2), outgoing(?act_prev, ?e1)

IV. RESULT AND ANALYSIS

In this subsection, a real-life data set is used to evaluate our proposed method and we present the final result in SWRL rules in Protege 5.1.0.

A. Evaluation Setup

In this research, a business process model of School Pre-Registration Admission Process is used to evaluate our proposed method. The event logs of business process model contain the information, such as the case ID and the activities.

Fig.4 shows the business process model used in this evaluation. If we do simulation, business process model in-

TABLE III. FRAGMENT OF EVENT LOGS FROM SCHOOL PRE-REGISTRATION ADMISSION PROCESS

Case ID	Activity	Start Time
Case_01	Start Pre-Registration	6/20/2014 8:32
Case_01	Check Student Achievement	6/20/2014 13:42
Case_01	Check Student Achievement	6/20/2014 17:33
Case_01	Is Reward	6/20/2014 23:41
Case_01	Join Student Verification	6/21/2014 9:30
Case_01	Print Student Verification	6/21/2014 10:46
Case_01	End Pre-Registration	6/21/2014 16:57
Case_02	Start Pre-Registration	6/21/2014 18:09
Case_02	Check Student Achievement	6/22/2014 4:14
Case_02	Start Pre-Registration	6/22/2014 10:51
Case_02	Check Student Achievement	6/22/2014 13:44
Case_02	Is Reward	6/22/2014 16:28
Case_02	Enter Field Achievement	6/22/2014 23:42
Case_02	Choose Achievement	6/23/2014 4:48
Case_02	Enter Academic Achievement	6/23/2014 16:44
Case_02	Join Achievement Score	6/23/2014 23:26
Case_02	Set Achievement Score	6/24/2014 5:40
Case_02	Join Student Verification	6/24/2014 7:48
Case_02	Print Student Verification	6/25/2014 1:08
Case_02	End Pre-Registration	6/25/2014 3:02
.....	.....	.....
Case_14	.....	.....

- Fig.4 can generate the event logs in Table III. From Table III, we get the information that there are length one loop and length two loop in business process model.

Then we convert the BPMN process model into Business Process Ontology Model and present the event logs based on Publish Workflow Ontology. Data Property, Class, Object Property in Protege based on PWO are shown in Fig.5, Fig.6, and Fig.7.

Fig.8 and Fig.9 present the Individual named Case ID 01 and Case ID 02 in Protege. These two are the cases in the event log of School Pre-Registration Admission Process. Based on these, we identify the short loops and build the rules using SWRL rules in Protege.

Individuals in Protege not only contain name of activities and case ID, but also the arcs of business process model. The arcs are divided into two types: incoming and outgoing. The incoming is used for the relations which lead to activity (input), meanwhile the outgoing explains the relations which come out from activity (output). Example of the arcs are shown in Fig.10 and named SEQxx.

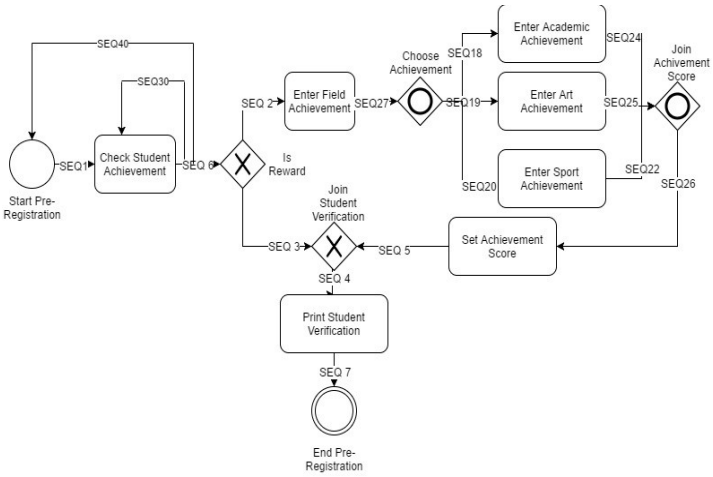

Fig.4. Business Process Model with Short Loops

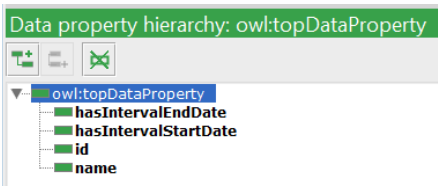

Fig.5. Data Property in Protege based on PWO

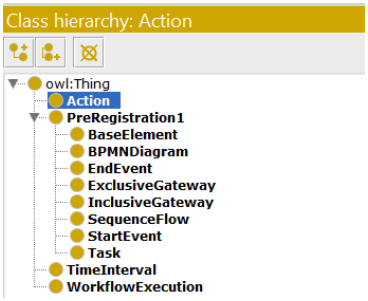

Fig.6. Class in Protege based on PWO

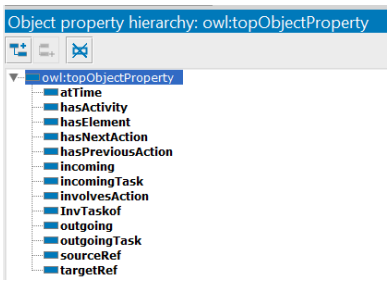

Fig.7. Object Property in Protege based on PWO

B. *Results and Discussion*

After we write the rules for identifying the short loops, then to run the rules we need to start the reasoner. A reasoner (semantic reasoner or reasoning engine or rules engine) is a free software which able to infer logical consequences from a set of asserted facts or axioms. Generalizing an inference engine and providing a richer set of mechanisms to work with are the notion of a reasoner. In this research, we use Pellet reasoner to run the rules. In Fig 11, Pellet reasoner need to be started to run the rules. After running the rules, we can provide the result of the rules we build.

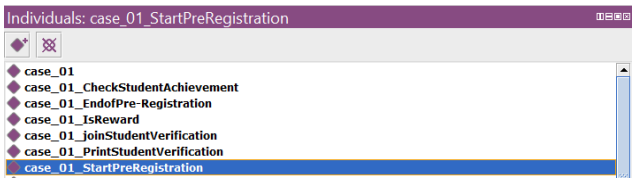

Fig.8. Individual named Case ID 01 in Protege

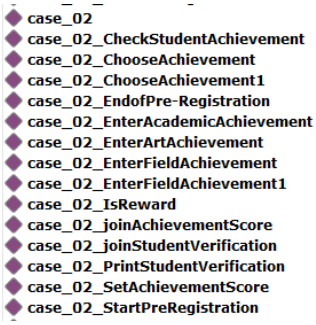

Fig.9. Individual named Case ID 02 in Protege

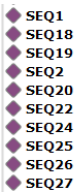

Fig.10. Individual named SEQxx in Protege

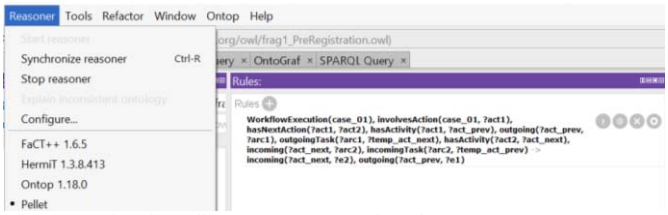

Fig.11. Starting the Pellet Reasoner to run the rules

Based on Case 01 in Table III, we get the information that there is length one loop. Length one loop happens in activity Check Student Achievement, previous activity is Student Pre-Registration, and next activity is Is Reward. Fig.12a shows the reasoner result of activity Start Pre Registration in Case 01 when rule is running. We get the new knowledge that activity Start Pre Registration has activity Check Student Achievement as the next actions.

Meanwhile, reasoner result of activity Check Student Achievement in Case 01 when rule is running presented in Fig.12b. The new knowlegde we obtain is lenght one loop happened in activity Check Student Achievement. Activity Check Student Achievement has two activities as its next action, namely Check Student Achievement and Is Reward. It means that after executing the activity Check Student Achievement, this activity leads to Check Student Achievement itself or activity Is Reward. The condition where an activity in a process is executed more than one time before leads to another activity in a process is called Length One Loop.

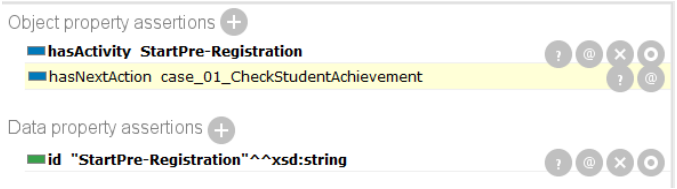

Fig.12a. Reasoner result of activity Start Pre Registration in Case 01 when rule is running

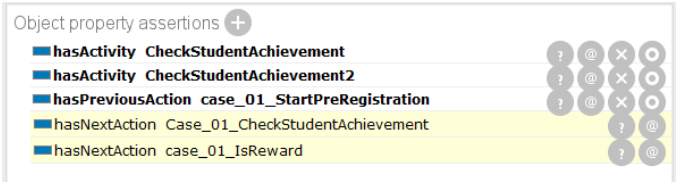

Fig.12b. Reasoner result of activity Check Student Achievement in Case 01 when rule is running

Based on Case 02 in Table III, we get the information that there is length two loop. Length two loop happens in activity Student Pre-Registration and next activity is Check Student Achievement. Fig.13a shows the reasoner result of activity Check Student Achievement in Case 02 when rule is running. We get the new knowledge that activity Check Student Achievement has activity Start Pre Registration and Is Reward as the next action.

Meanwhile, reasoner result of activity Start Pre Registration in Case 02 when rule is running presented in Fig.13b. The new knowlegde we obtain is lenght two loop happened in activity Start Pre Registration. Activity Start Pre Registration has Check Student Achievement as next action and has Start Pre Registration as previous action. It means that after executing the activity Start Pre Registration, this activity leads to Check Student Achievement and Check Student Achievement leads to activity Start Pre Registration itself and Is Reward. This condition is called Length Two Loop.

CONCLUSION

This paper presents the method which can identify and automate identification of short loops based on the event logs in Business Process Ontology Model as the business process model. Although business process model is always used in organization, in fact there are several problems that occur in the steps of modeling the business process model from event log, such as short loops. There are several research solved the problem of short loops using matrix and formula, but none of them applied their method in ontology.

The work was tested on a real-life event log, a business process model of School Pre-Registration Admission Process. The result shows that our method is able to identify process model with short loops (Length One Loop and Length Two Loop) in ontology using SWRL rules.

Future work aims at converting back Business Process Ontology Model into BPMN process model, after short loops are identified and process model is updated. This result will present in business process model.

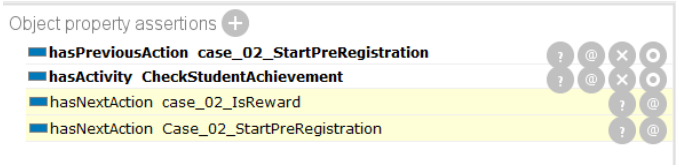

Fig.13a. Reasoner result of activity Check Student Achievement in Case 02 when rule is running

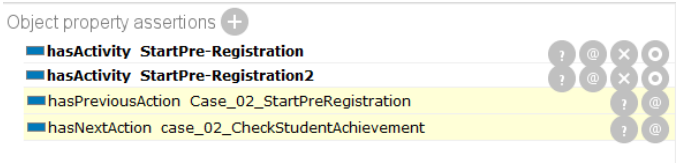

Fig.13b. Reasoner result of activity Start Pre Registration in Case 02 when rule is running

ACKNOWLEDGMENT

Authors give a deep thank to Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember for supporting this research.

REFERENCES

[1] R. Sarno, W. A. Wibowo, Kartini, F. Haryadita, Y. Effendi, and K. Sungkono, "Determining Model Using Non-Linear Heuristics Miner and Control-Flow Pattern," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 14 (1), 2016. http://doi.org/10.12928/telkomnika.v14i1.3257

[2] R. Sarno, E. W. Pamungkas, D. Sunaryono, and Sarwosri, "Business process composition based on meta models," International Seminar on Intelligent Technology and Its Applications (ISITIA), 2015. http://doi.org/10.1109/isitia.2015.7219998

[3] R. Sarno, Y. Effendi, and F. Haryadita, "Modified Time-Based Heuristics Miner for Parallel Business Processes," International Review on Computers and Software (IRECOS), vol. 11 (3), pp. 249-260, March 2016. http://doi.org/10.15866/irecos.v11i3.8717

[4] W. M. P. Van der Aalst, "Process Mining - Discovery, Conformance and Enhancement of Business Processes", Netherlands: Springer, pp. 95-125, 2011.

[5] H. A. Hartanto, R.Sarno, and N. F. Ariyani, "Linked Warning Criterion on Ontology-Based Key Performance Indicators", International Seminar on Application for Technology of Information and Communication (ISemantic), 2016

[6] C. Natschlager, "Towards a BPMN 2.0 Ontology", 2011. DOI: 10.1007/978-3-642-25160-3_1

[7] W. Abramowicz, A. Filipowska, M. Kaczmarek, and T. Kaczmarek, "Semantically enhanced Business Process Modelling Notation. Work on Semantic Business Process and Product Lifecycle Management (SBPM)", 2007

[8] C. Francescomarino, C. Ghidini, M. Rospocher, L. Serafini, and P. Tonella, "Reasoning on Semantically Annotated Processes". 6th Int. Conf. on Service-Oriented Computing, 2008

[9] P. Ciccarese and S. Peroni, "The Collections Ontology: creating and handling collections in OWL 2 DL frameworks". Semantic Web, 5(6): 515–529, 2016 http://dx.doi.org/10.3233/SW130121

[10] A. Gangemi, S. Peroni, D. Shotton, and F. Vitali, "A pattern-based ontology for describing publishing workflows", Proceedings of the 5th International Workshop on Ontology and Semantic Web Patterns, 2014

[11] S. Peroni, "Example of use of PWO : publishing domain. figshare", 2015 http://dx.doi.org/10.6084/m9.figshare.1449043

[12] A. Gangemi, S. Peroni, D. Shotton, and F. Vitali, "The Publishing Workflow Ontology (PWO)", 2015