Conformance Checking Evaluation of Process Discovery Using Modified Alpha++ Miner Algorithm

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

Abstract*—Process mining techniques extract business processes from event logs. Process discovery, one of process mining techniques, has aim to mine a process model from event log. To help its process, process discovery uses some algorithms such as alpha, alpha+ and alpha++ where each algorithm has some advantages and limitations in analyzing event log. Alpha++ miner algorithm is considered the most advance improvement of alpha and alpha+ algorithm, therefore we use alpha++ as basic algorithm to our Modified alpha++ Miner. The proposed method in this paper uses Modified alpha++ Miner to generate process model, determine sequential and parallel relation from process model using temporal causal relation and control flow, and then evaluate the process model in terms of fitness value, validity and completeness. This paper will be focused on the advantages of Modified alpha++ Miner to discover business process.

Keywords-- Alpha++ Algorithm; Time Interval; Alpha Miner; Process Discovery; Process Mining; Conformance Checking

I. INTRODUCTION

Process mining techniques extract business processes from event logs. Process mining is divided into three types, namely process discovery, model extension/ enhancement and conformance checking. From those three, mining a process model from event log is main focus of process discovery [1]. To analyze the event log, process discovery uses technique or algorithm, such as α (Alpha), α+, α++, and heuristics miner [2]. Each of this algorithm has some advantages compared to the others, but it also has its limitations [3]. Limitations of α, α+, and α++ algorithm can be found in related works part of this paper.

The proposed method in this paper uses Modified α++ Miner (MA++M) algorithm to generate the process model as well as their relation based on event log. Compared to α++, this algorithm can detect overlapping process in time interval and identifies it as a parallel process. Because of this advantage, MA++M algorithm will generate less trace than the α++ algorithm. We use α++ algorithm because at current time the α++ is the latest version of modified α algorithm with big improvement of either α or α+ and MTBAM algorithm is also based on α algorithm, short description related to α++ will be written in related works section of this paper.

Main focus of this research is modifying α++ algorithm to discover process model and then evaluate the process model in terms of fitness value, validity and completeness, which is better known as conformance checking.

II. RELATED WORK

Related work of this paper explains and does comparing the existing process discovery algorithm α++ based on the earlier α and α+ algorithm. α+ algorithm itself is based on the algorithm called α.

A. α++ Algorithm

α algorithm is the most commonly used algorithm in process discovery to analyze and discover activity from cases in the event log. However, α algorithm has some disadvantages in the implementation. In [4], the limitations of α algorithm is further exposed, α algorithm cannot mine duplicate and hidden tasks, non-free choice [5], cannot deal with noisy data, and data with time constraint and mining loops (L1L and L2L). α+ which is the improvement of α algorithm can detect short loop like length one loop. Lastly α++ algorithm that can detect implicit dependencies, discover length one loop (L1L) and length two loop (L2L) and can reconstruct workflow-net (WF-net) to non-free choice. However, there is a disadvantage of α++ algorithm, that is it will produce traces according to parallel cases in the event log. The bigger the number of the parallel case, the higher the number of traces generated from α++ algorithm [6]. Further explanation of α++ algorithm and α algorithm can be found in [7].

B. Modified Time-Based Alpha Miner Algorithm (MTBAM)

MTBAM algorithm uses temporal causal relation and control flow to differentiate sequential and parallel relation as explained in [6]. As mentioned before in introduction, this algorithm can generate less traces than α algorithm using time interval so the fitness value is higher than α algorithm. We adopt this temporal causal relation and control flow to determine the parallel and sequence of process model based on their time interval into this modified α++ miner algorithm.

III. CONFORMANCE CHECKING EVALUATION

In this section, conformance checking evaluation is explained. We only use three evaluation criteria in this research. They are fitness value, validity and completeness. Process model from discovery process using Modified α++ Miner algorithm are evaluated based on this criteria.

A. Fitness of Process Model

Conformance checking is closely related to fitness. Fitness is kind of evaluation of a process model which has a focus on all activities of process model should correctly parsed based on event log. The process model which has all the activities that can be correctly parsed based on the event log has a high fitness value. On the contrary if many of the activities cannot be correctly parsed into the process model then the fitness value will be low.

Fitness value ranges from 0 (no activities including cases and traces parsed based on event log into process model) to 1 (all activities including cases and traces parsed correctly based on event log into process model) [6 , 8]. Generally, all α, α+ and α++ algorithm have high fitness value because of the relations in process model. They cover all relations, sequence and parallel without giving the threshold. We can calculate the fitness value using (1).

$$Xf = \frac{CapturedCasesinEL}{LogCasesinEL} \qquad (1)$$

where:

Xf : fitness value of process model
CapturedCasesinEL : total number of cases that can be parsed into the process model
LogCasesinEL : total number of cases in event log

B. Validity of Process Model

Event log that has been formed into a process model must be evaluated in terms of dependency graph, causal net and semantic. If all of them are in correct way, then it is meaning of validity [1 , 9]. Semantic process model can be obtained if process model enriches with split and join. To fulfil the evaluation criteria of validity, there are three steps to evaluate the validity of process model:

1. Determine dependency graph
2. Transform (1) and generate causal net
3. Create semantic process model
4. Obtain final model of event log which follows the validity condition correctly

C. Completeness of Process Model

Completeness is a notion referring to a problem in computational complexity theory. In process mining, concept of completeness related to minimum number of traces needed to discover process model correctly [10].

Process discovery also implement the idea of completeness. Process discovery works correctly when the event log meets the idea of the completeness of the algorithm. Event log which do not meet the idea of completeness are called incompleteness. Incompleteness occurs because there is too little data on event log which leads to misunderstandings in process discovery. So the result of process model is not as it should be [11].

Definition 1. There are event log (EL), process model (PM), and activity (Act). Notion of completeness for Act, EL and PM are:

$$Act \in PM, iff\ Act \in EL$$
$$(Act_1 \rightarrow Act_2) \in PM, iff\ (Act_1 > Act_2) \in EL\ \ where\ i\ in\ Act_i\ is\ an\ identifier\ of\ an\ activity.$$
$$(Act_1 \parallel Act_2) \in PM, iff\ [(Act_1 > Act_2) \in EL\ and\ (Act_2 > Act_1) \notin EL] \vee [\{Act_1 @ Act_2 \vee Act_{1_f} Act_2 \vee Act_1 \Diamond Act_2 \vee Act_{1_p} Act_2 \vee Act_1 \square Act_2\} \in EL\ ].$$

Given a process model (PM), the number of activities (n) in the i-branch (i), and the number of activities which has branch activities (p). Process model (PM) consists of parallel AND, OR and XOR. The number of traces required for the process:

Formula 1. Parallel AND relation

$$AND_T = \sum_{i=1}^{p} \sum_{j=i+1}^{p} n_i \times n_j \qquad (2)$$

Formula 2. Parallel OR relation

$$OR_T = \left(\sum_{i=1}^{p} \sum_{j=i+1}^{p} n_i \times n_j \right) + 1 \qquad (3)$$

Formula 3. Parallel XOR relation

$$XOR_T = \sum_{i=1}^{p} if\ R_i == sequence, 1\ else\ 0 \qquad (4)$$

Formula 4. Process model (PM) consists of a parallel set activity which each branch of parallel set has another parallel or sequence relation. To mine the process model with parallel set (P_T), the number of traces required:

$$P_T = AND_T + OR_T + XOR_T \qquad (5)$$

IV. Proposed Method

Fig. 1 explains our proposed method of this research. This research focuses on discovering process model from Modified α++ miner algorithm. After we discover the process model, we evaluate the fitness, validity and completeness.

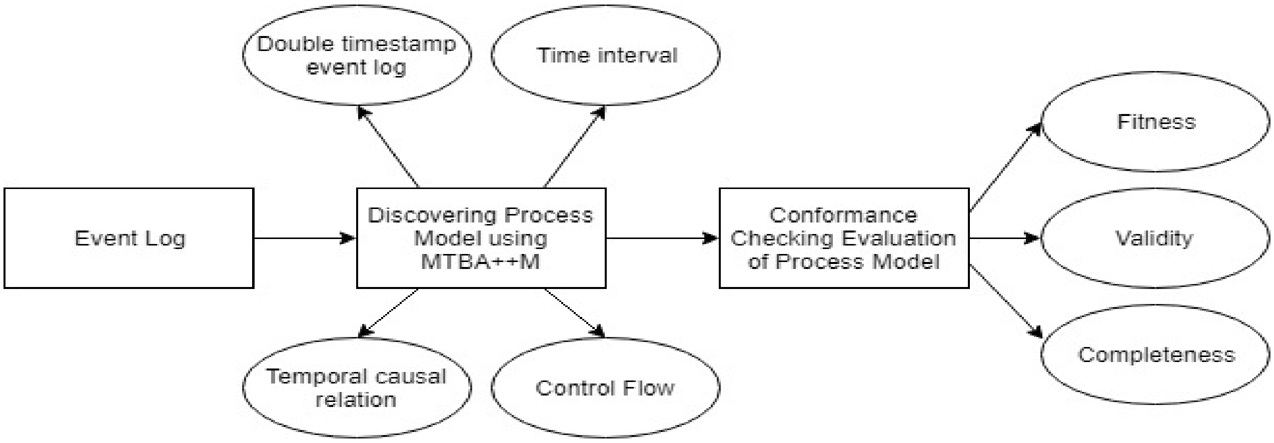

Double timestamp event log
Time interval
Event Log
Discovering Process Model using MTBA++M
Conformance Checking Evaluation of Process Model
Fitness
Validity
Completeness
Temporal causal relation
Control Flow

Fig. 1. Proposed Method

Step 1. Input data of this process discovery is event log. Event log has two types in reality; single timestamp and double timestamp event log. Generally, to discover the process model, we use Case ID, Activity and timestamp.

Step 2. After the input is ready to mine, Modified α++ Miner algorithm is used to discover the process model. To discover process model, as we explained in Section II, we adopt the steps of Modified Time-based Alpha Miner (MTBAM) algorithm to determine the sequence and parallel relation of process model which are temporal causal relation and control flow [6] [12] [13] [14].

Main differences between MTBAM and MA++M are in terms of detecting implicit dependency, discovering length one loop (L1L) & length two loop (L2L) and reconstruct the workflow-net (WF-net) into non-free choice.

There are 13 steps used in Modified α++ Miner algorithm:

Step 1. Determine the sequence relation and parallel relation from each case

Step 2. Delete duplicate relations (sequence and parallel) from all cases

Step 3. Get all traces of event log which include sequence and parallel relations

Step 4. Create set of transition in WF-net

Step 5. Create set of input and output transition from event log

Step 6. Create set of L1L and L2L from event log

Step 7. Detect all implicit dependencies from WF-net

Step 8. Create gantt chart for all traces

Step 9. Create the places

Step 10. Discover the process model

Step 11. Determine the type of parallel relation AND, XOR or OR

Step 12. Add input activity, output activity and sequence relations into process model

Step 13. A process model is complete

Step 3. Conformance Checking Evaluation of Process Model. After process model is generated, fitness value, validity and completeness of process model need to be measured.

V. Experimental Result

A. Event Log

We use Yarn Manufacturing Process event log in this research. Yarn Manufacturing Process consists of 14 activities, 8 traces and 50 cases. This event log is double timestamp event log which has start time and complete time. The relations in this event log are sequential and parallel relations. Table I shows event log of Yarn Manufacturing Process. For this experiment, we only use case id, activities and timestamp.

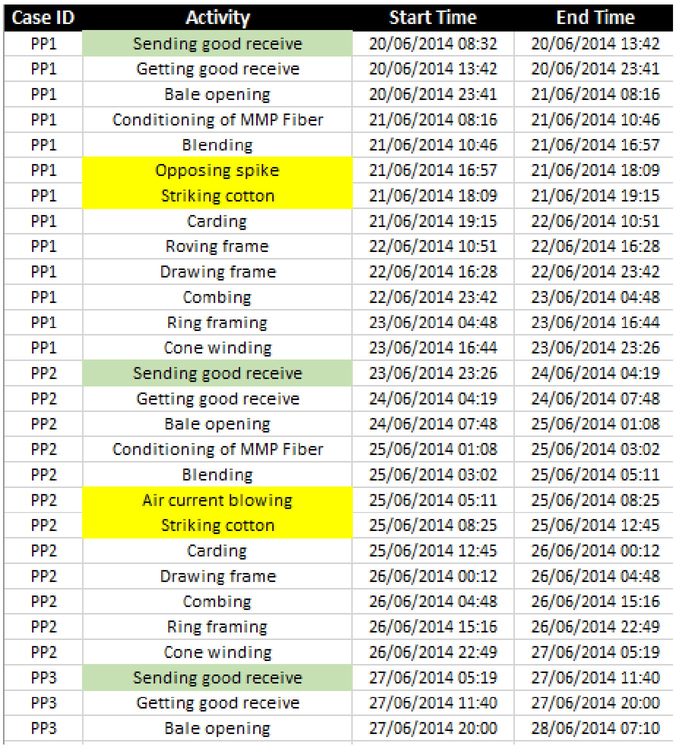
B. Experimental Result

We mine the process model using MA++M algorithm and event log in Table I. Fig. 2 shows the result after MA++M is run. Process model in Fig. 2 contains sequential relation, parallel XOR and OR, and also L1L and L2L. Table II explains activities listed in the process model in Fig. 2.

Based on Fig. 2, activity ‘OpposingSpike’ and activity ‘AirCurrentBlowing’ are in parallel relation XOR. Meanwhile, activity ‘DrawingFrame’ and activity ‘RovingFrame’ are in parallel relation OR. We get the knowledge that using MA++M, we can discover the process model which contains L1L, L2L and non-free choice. α, α+ and α++ algorithm cannot handle those three problems.

α algorithm can only discover relations of process model correctly. Fig. 3 shows process model discovery using α algorithm. Based on Fig. 2 and Fig.3, Modified α++ Miner algorithm is more powerful than α, α+ and α++ algorithm because all the conditions in the event log can be handled well. Activities listed in the process model in Fig. 3 also as same as explained in Table II.

TABLE I. EVENT LOG OF YARN MANUFACTURING PROCESS



Case ID	Activity	Start Time	End Time
PP1	Sending good receive	20/06/2014 08:32	20/06/2014 13:42
PP1	Getting good receive	20/06/2014 13:42	20/06/2014 23:41
PP1	Bale opening	20/06/2014 23:41	21/06/2014 08:16
PP1	Conditioning of MMP Fiber	21/06/2014 08:16	21/06/2014 10:46
PP1	Blending	21/06/2014 10:46	21/06/2014 16:57
PP1	Opposing spike	21/06/2014 16:57	21/06/2014 18:09
PP1	Striking cotton	21/06/2014 18:09	21/06/2014 19:15
PP1	Carding	21/06/2014 19:15	22/06/2014 10:51
PP1	Roving frame	22/06/2014 10:51	22/06/2014 16:28
PP1	Drawing frame	22/06/2014 16:28	22/06/2014 23:42
PP1	Combing	22/06/2014 23:42	23/06/2014 04:48
PP1	Ring framing	23/06/2014 04:48	23/06/2014 16:44
PP1	Cone winding	23/06/2014 16:44	23/06/2014 23:26
PP2	Sending good receive	23/06/2014 23:26	24/06/2014 04:19
PP2	Getting good receive	24/06/2014 04:19	24/06/2014 07:48
PP2	Bale opening	24/06/2014 07:48	25/06/2014 01:08
PP2	Conditioning of MMP Fiber	25/06/2014 01:08	25/06/2014 03:02
PP2	Blending	25/06/2014 03:02	25/06/2014 05:11
PP2	Air current blowing	25/06/2014 05:11	25/06/2014 08:25
PP2	Striking cotton	25/06/2014 08:25	25/06/2014 12:45
PP2	Carding	25/06/2014 12:45	26/06/2014 00:12
PP2	Drawing frame	26/06/2014 00:12	26/06/2014 04:48
PP2	Combing	26/06/2014 04:48	26/06/2014 15:16
PP2	Ring framing	26/06/2014 15:16	26/06/2014 22:49
PP2	Cone winding	26/06/2014 22:49	27/06/2014 05:19
PP3	Sending good receive	27/06/2014 05:19	27/06/2014 11:40
PP3	Getting good receive	27/06/2014 11:40	27/06/2014 20:00
PP3	Bale opening	27/06/2014 20:00	28/06/2014 07:10

TABLE II. CODE AND ACTIVITY NAME OF YARN MANUFACTURING PROCESS

Code	Activity Name
A	SendingGoodReceive
B	GettingGoodReceive
C	BaleOpening
D	ConditioningofMMFFiber
E	Blending
F	OpposingSpike
G	AirCurrentBlowing
H	StrikingCotton
I	Carding
J	DrawingFrame
K	RovingFrame
L	Combing
M	RingFraming
N	ConeWinding

C. Conformance Checking Evaluation

After discovering process model, we do the conformance checking evaluation of process model discovered by MA++M algorithm. We implement the (1), (2), (3), (4) and (5).

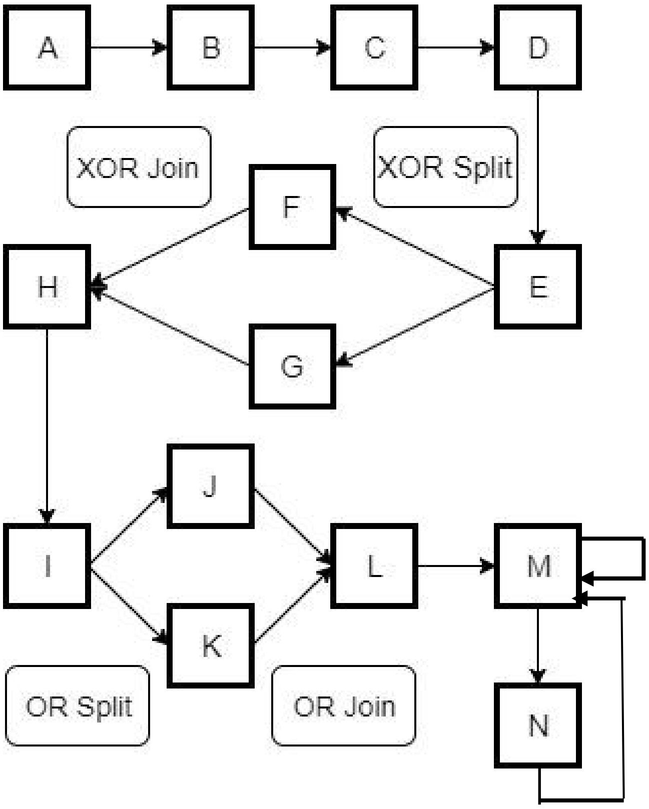


Fig. 2. Process Model of Yarn Manufacturing Process using MA++M algorithm

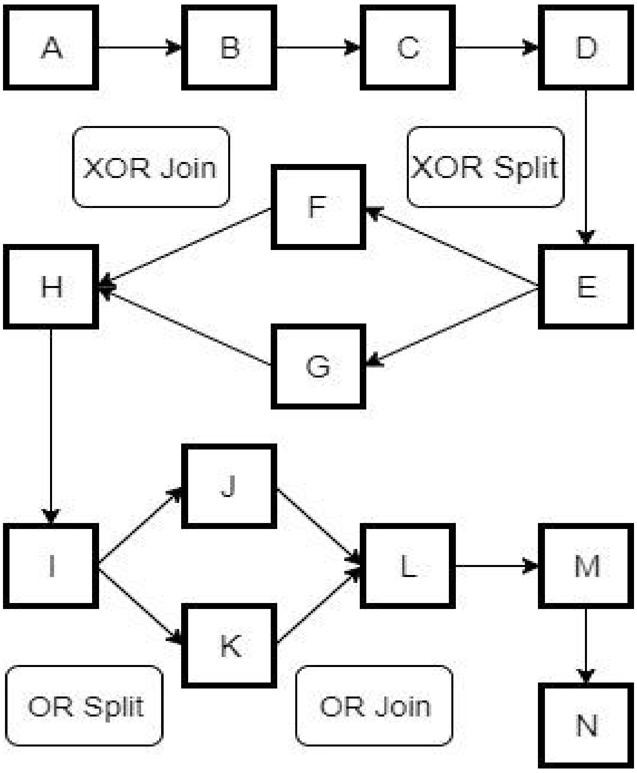


Fig. 3. Process Model of Yarn Manufacturing Process using α algorithm

a. Fitness

First evaluation is calculating the fitness value. We calculate fitness using (1) and we also compare the fitness between MA++M algorithm and MTBAM algorithm.

Based on Table III, result of fitness value of MA++M is higher than MTBAM because all cases in Yarn Manufacturing Process parsed into the process model based on event log. Meanwhile, MTBAM cannot parse the case where there are L1L and L2L.

TABLE III. FITNESS VALUE OF MA++M AND MTBAM

Fitness value of MA++M algorithm	**Fitness value of MTBAM algorithm**
1.0	0.92857

b. Validity

To get the validity of process model, we need to obtain the dependency graph of process model as generated using Modified α++ Miner algorithm. As we get the dependency graph, we need to transform it into causal net. Table IV shows dependency graph of process model generated by MA++M algorithm. After obtaining the dependency graph, we transform the dependency graph into causal net as shown in Table V. In causal net, input activity and output activity must be correct for each activity.

The final step of validity is connecting the gateway parallel and sequence relation of process model correctly so that we can create the semantic process model. Fig.4 shows the result where process model has correct validity. Based on validity terms, the result of process model by Modified α++ Miner algorithm generated the semantic process model in a valid way.

c. Completeness

Completeness is measured by calculating all the relations of business process to obtain the total minimum traces. They are needed to mine the process model correctly. Event log of Yarn Manufacturing Process can fulfill the idea of completeness if it has two traces calculated using (5).

Process model in Fig. 4 has two parallel relations; XOR relation and OR relation. We calculate the completeness using formula of completeness for XOR and OR (3) and (4).

The conditional XOR relation in the process model has one activity in each of its branch, then the number of traces based on (4) is

$$XOR = 1 + 1 = 2$$

Meanwhile, conditional OR relation also has one activity in each branch, then the number of trace based on Formula 3 is

$$OR = (1 \text{ x } 1) + 1 = 2$$

Based on (5), the number of traces of Yarn Manufacturing Process which consist of parallel sets are selecting the highest number of traces of each parallel sets. In this case, the answer is two.

Based on the calculation, the event log of Yarn Manufacturing Process for process discovery has 8 traces and 50 cases. Therefore, the event log of Yarn Manufacturing Process fulfills the idea of completeness, because it contains 8 traces of the 2 traces it should have.

TABLE IV. DEPENDENCY OF YARN MANUFACTURING PROCESS

Input Set	**Activity**	**Output Set**
{∅}	SendingGoodReceive	{GettingGoodReceive }
{SendingGoodReceive}	GettingGoodReceive	{BaleOpening}
{GettingGoodReceive }	BaleOpening	{ConditioningofMMF Fiber}
{BaleOpening}	ConditioningofMMFFiber	{Blending}
{ConditioningofMMF Fiber}	Blending	{OpposingSpike}
{Blending}	OpposingSpike	{AirCurrentBlowing}
{Blending}	OpposingSpike	{StrikingCotton}
{OpposingSpike}	AirCurrentBlowing	{Carding}
{OpposingSpike}	StrikingCotton	{Carding}
{StrikingCotton}	Carding	{DrawingFrame, RovingFrame}
{Carding}	DrawingFrame	{Combing}
{Carding}	RovingFrame	{Combing}
{DrawingFrame, RovingFrame}	Combing	{RingFraming}
{Combing}	RingFraming	{ConeWinding}
{RingFraming}	ConeWinding	{∅}

VI. CONCLUSION

In this research, we propose the modification of α++ algorithm which is the extended version of α and α+ algorithm. The modification of α++ is known as Modified α++ Miner (MA++M) algorithm which uses time interval in discovering the process model. Our experimental results explained that the result of process model can discover the sequence, parallel, length one loop (L1L) and length two loop (L2L) correctly. After discovering process model, the result is then evaluated using conformance checking which consists of-

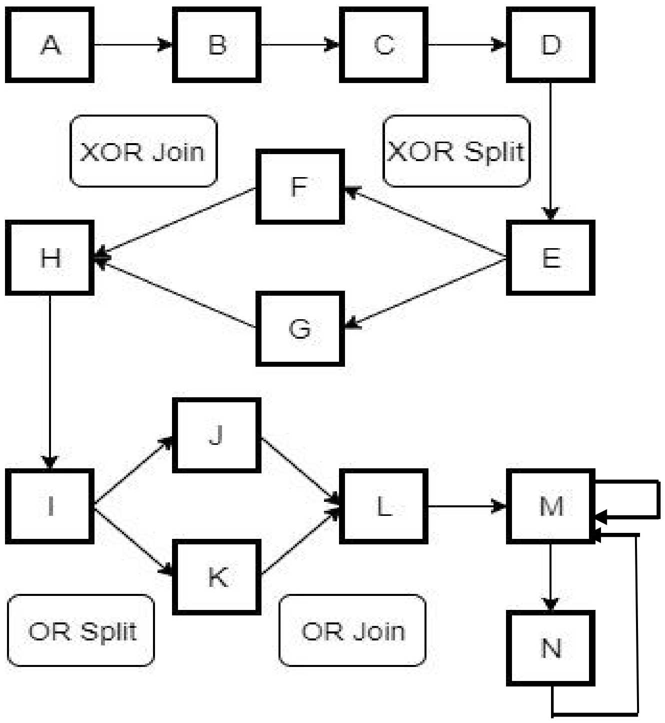


Fig. 4. Final process model of Yarn Manufacturing Process which has correct validity

three evaluation criteria; fitness value, validity and completeness. Final process model has high fitness value, can generate the semantics in a valid way and also fulfills the idea of completeness.

TABLE V. CAUSAL NET OF YARN MANUFACTURING PROCESS

Input Set	Activity	Output Set
{{∅}}	SendingGoodReceive	{{GettingGoodReceive}}
{{SendingGoodReceive}}	GettingGoodReceive	{{BaleOpening}}
{{GettingGoodReceive}}	BaleOpening	{{ConditioningofMMFFiber}}
{{BaleOpening}}	ConditioningofMMFFiber	{{Blending}}
{{ConditioningofMMFFiber}}	Blending	{{OpposingSpike}}
{{Blending}}	OpposingSpike	{{AirCurrentBlowing}}
{{Blending}}	OpposingSpike	{{StrikingCotton}}
{{OpposingSpike}}	AirCurrentBlowing	{{Carding}}
{{OpposingSpike}}	Striking_Cotton	{{Carding}}
{{StrikingCotton}}	Carding	{{DrawingFrame, RovingFrame}}
{{Carding}}	DrawingFrame	{{Combing}}
{{Carding}}	RovingFrame	{{Combing}}
{{DrawingFrame, RovingFrame}}	Combing	{{RingFraming}}
{{Combing}}	RingFraming	{{ConeWinding}}
{{RingFraming}}	ConeWinding	{{∅}}

REFERENCES

[1] 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 (3), pp. 249-260, March 2016*. http://doi.org/10.15866/irecos.v11i3.8717

[2] R. Sarno and Y. A. Effendi, “Hierarchy Process Mining from Multi-Source Logs,” *Telecommunication, Computing, Electronics and Control (TELKOMNIKA), Vol 15, No 4, 2017.*
DOI: http://dx.doi.org/10.12928/telkomnika.v15i4.6326

[3] Banerjee and P. Gupta. “Extension of alpha algorithm for process mining”

[4] L. Wen, Wil M.P. van der Aalst, J. Wang, and J. Sun, “Mining process models with non-free-choice constructs”.

[5] R. Sarno, P. L. I Sari, D. Sunaryono, B. Amaliah, and I. Mukhlash, “Mining decision to discover the relation of rules among decision points in a non-free choice construct,” *Proceedings of International Conference on Information, Communication Technology and System (ICTS), 2014.* http://dx.doi.org/10.1109/ICTS.2014.7010557

[6] Y. A. Effendi and R. Sarno, “Modeling Parallel Business Process Using Modified Time-based Alpha Miner”, *International Journal of Innovative Computing, Information and Control (IJICIC), Vol.14, No.4, August 2018*

[7] W.M.P. van der Aalst, Process mining: discovery, conformance and enhancement of business processes (Springer Science and Business Media, 2011). http://dx.doi.org/10.1007/978-3-642-19345-3

[8] Y. A. Effendi and R. Sarno, “Discovering optimized process model using rule discovery hybrid particle swarm optimization,” *3rd International Conference on Science in Information Technology (ICSITech), pp. 97-103, 2017.* DOI: 10.1109/ICSITech.2017.8257092

[9] Y. A. Effendi and R. Sarno, “Discovering process model from event logs by considering overlapping rules,” *4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 1-6, 2017.* DOI: 10.1109/EECSI.2017.8239193

[10] S. Huda, R. Sarno, and T. Ahmad, “Increasing Accuracy of Process-based Fraud Detection Using a Behavior Model,” *International Journal of Software Engineering and Its Applications, 10 (5), pp. 175-188, 2016*. https://doi.org/10.14257/ijseia.2016.10.5.16.

[11] R. Sarno, W. A. Wibowo, D. Sunaryono, A. Munif. “Developing Workflow Patterns Based on Functional Subnets and Control–Flow Patterns”. International Conference on Science in Information Technology (ICSITech), 2015. DOI : https://doi.org/10.1109/icsitech.2015.7407771

[12] S. S. Pinter and M. Golani, “Discovering workflow models from activities’ life spans”, *Computers in Industry, vol. 53, pp. 283-296, 2004*

[13] R. Sarno, Kartini, W. A. Wibowo, and A. Solichah, “Time Based Discovery of Parallel Business Processes,” *International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. *28 – 33, 2015.* DOI: 10.1109/IC3INA.2015.7377741

[14] I. G. Anugrah, R. Sarno, R. N. E. Anggraini, “Decomposition Using Refined Process Structure Tree (RPST) and Control Flow Complexity Metrics”. *International Conference on Information & Communication Technology and Systems (ICTS),* 2015.
DOI: https://doi.org/10.1109/icts.2015.7379899