

Represent The Discovered Proportional Information Control Net Using Extended XPDL Format

Dinh-Lam Pham, Hyun Ahn, Kyoung-Sook Kim and Kwanghoon Pio Kim*

*Data and Process Engineering Research Lab. Division of Computer Science and Engineering,
KYONGGI UNIVERSITY, 154-42 Youngtong-Gu Suwon-Si Gyeonggi-Do, 16627, Republic of Korea

* Corresponding Author: Kwanghoon Pio Kim

phamdinhlam@kgu.ac.kr, hahn@kgu.ac.kr, khmjmc@kgu.ac.kr, kwang@kgu.ac.kr

Abstract—The XML Process Definition Language (XPDL) is a widely used XML format for storing and exchanging business process information from various workflow systems. In this paper, we introduce the methodology to represent the run-time proportional Information Control Net (pICN) discovered from the business process event log by using XPDL format. More precisely, we propose an approach and algorithm to export pICN data from the graphical visualization model to XPDL format (version 2.1) with several extended attributes to store run-time proportional information of activities in the system. By storing the discovered pICN in an XPDL format, we do not have to rediscover it again in the next mining time since using the information that has been recorded. With the popular and widespread use of the XPDL format, we can use the discovered pICN model in various business process management and workflow systems.

Keywords: Process mining, XPDL format, pICN, Workflow system, business process.

I. INTRODUCTION

The workflow system includes the execution and automation of business processes. In particular, tasks, information, or documents transmitted from one person to another perform actions according to a set of procedural rules set by the business. By automation, business processes help coordinate tasks between people, synchronize data between application systems with the ultimate goal of improving performance, responsiveness, and profitability. Using a workflow management system, which helps to reduce work time, as well as reduce potential human errors, make work processes more efficient, each step of the business process is clearly defined, be monitored for execution time, and optimized for maximum performance. By optimizing processes, updating work every minute, managers and employees can get activities done faster and make smarter decisions [1][2]. One of the most widely used input formats in worldwide workflow systems is XPDL [3]. XPDL is a standard standardized by WfMC to help exchange business processes between different workflow systems. In the definition of XPDL, an XML structure is described to specify the declaration of a workflow process. XPDL is built to support process information exchange, including the graphical interface and semantics of a workflow business process system. XPDL is one of the best formats supporting information exchange in the business process model and notation (BPMN) systems; It is designed to store all aspects in a BPMN diagram. Beside, XPDL also includes elements to represent graphical

information (e.g., X and Y position of nodes) as well as other content that will be used to perform a process.

As we know, in the business process management life cycle, there are 6 process phases, that are process identification, process discovery, process analysis, process redesign, process implementation, process monitoring and controlling. In the process discovery phase, we discover the process model of the information system through its operation event log. The discovered model can be present in different types such as Petri net [4], process model notation, information control net (ICN) [5]. Furthermore, we can present the processes in the form of proportional ICN (pICN). The pICN model describes the ICN process model of the system and the proportional of running one activity followed by the others in the discovered model from the system event log history.

Discovering the pICN model from the log of the system often requires us to read the information contained in the log file, identify the components involved in the operation of the system (e.g., activities, performers, times) as well as implement the algorithms that will be applied to exploit information. This activity usually takes time and not only depends on the complexity of the log file but the size of the data. Therefore, the discovered model should be stored for later use to avoid re-mining the same knowledge. To solve this problem, we propose an approach representing the pICN model in XPDL format. Accordingly, the pICN model explored graphically will be represented in XPDL version 2.1. In the scope of this paper, assume that we have discovered the pICN model already, and we concern about how to represent it in XPDL format for later use in workflow systems. In terms of organizing the paper content, in section II, we survey some of the other research results related to the XPDL standard, pICN model. The next section describes the proposed approach in detail as well as the algorithm to represent pICN in the form of XPDL. Finally, we summarize the content of the paper and give some conclusions and future work.

II. RELATED WORK

In this section, we present summary of some research results related to the XPDL standard as well as the pICN model. In [6], **Robert Shapiro** conducted a comparison of XPDL, BPML, and BPEL. Their primary objective was to clarify the differences between the BPML, XPDL, and BPEL. In the

paper, they compared these languages in several different categories such as block structure, specialized atomic activities, activities and attributes, web services orchestration, transactions and exception handlers, high-level constructs. The results of their study are a good reference in selecting languages that suit different scenarios.

In [7], **Nicolas Guelfi** et al. presented a systematic approach generating XPDL specifications from UML activity diagrams. Their approach operates in two steps. In the first step, they mapped each UML activity element to a new XPDL element. In the second step, they applied some optimization rules to obtain an optimal XPDL specification. They also explained how possible to reuse established property proofs on activity diagrams in order to prove properties on the XPDL specification.

In [8], **Hye-Min Noh** et al. proposed an approach to generate an XPDL document from a business process model. In their approach, the business process model is represented by the extended UML activity diagram, and they also provided an implementation for the approach. The main key idea of their proposal is mapping elements between XPDL and the process model (mapping state, activity, synchronization, decision, and transition element).

In [9], **Haiping Zha** et al. introduced a methodology for converting XPDL into Petri nets. Their study aimed to automatically transform XPDL process models to Petri nets with equivalent structure and behaviors. In the study, they explained how to map from XPDL to Petri Nets through activities, links, routing structure, and successfully applied the solution to some sample processes.

In [10], **Pingpeng Yuan** et al. introduced a Workflow Transformation tool between XPDL and BPEL (business process execution language). The tool can automatically translate workflow expressed in terms of XPDL onto BPEL, and vice versa. In order to transfer information between XPDL and BPEL, The tool first analyses the structures of process and gets the result of a Map-like structure expressed with an adjacency matrix. Then it scans the adjacency matrix and identifies the structure of the block to generate XPDL or BPEL code.

In [11], **Sang-Young Lee** et al. conducted a survey of UML based XPDL document for business process modeling. In the paper, they also proposed an approach generating XPDL file from a business process model represented by the extended UML activity diagram. They described how to map the business process model (BPM) to the XPDL document by using a mapping table to map elements on both sides. Their approach can generate an XPDL file successfully from BPM presented by a UML activity diagram. However, their method can not map all the modeling elements of the UML activity diagram into corresponding XPDL entities.

In [12] [13] [14], **Ahn** et al. used the XPDL process not only for discovering the performer-activity affiliation networking knowledge but generating process entity hierarchies. They implemented a workflow-supported organizational intelligence system (wOIS-paan) for discovering the performer-activity knowledge and proposed the XPDL-based process entity generator by using a useful hierarchy generator to perform analytic of hierarchical properties among business process entities.

Through some of the studies just mentioned, we can see the significance as well as the popularity, importance, and the effectiveness of using XPDL format to store and display business process information. In the next section, we present the solution to represent pICN in XPDL format.

III. REPRESENT THE RUN-TIME PICN USING EXTENDED XPDL FORMAT

In this section, we explain the concept of run-time pICN and describe the method and algorithm to represent pICN in the XPDL format.

A. The Run-time Proportional Information Control Net (pICN)

To understand the pICN concept, first of all, we consider the Information Control Net (ICN) [5]. There are four types of

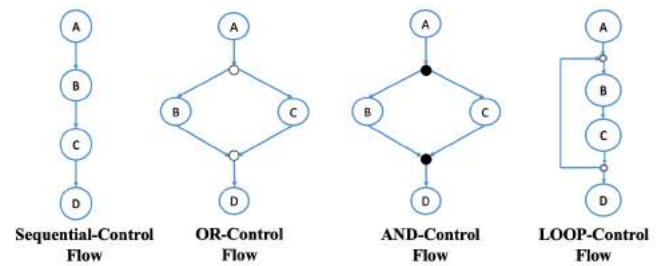


Fig. 1: The ICN Primitive Patterns

control-flow in the ICN-based process model. That is Sequential Control-flow (an activity follows activity by a sequential order). OR Control-flow (an activity has a disjunctive (or decision) transition follow by two or more activities. Only one of the followed activities will be executed next). AND Control-flow (an activity has a conjunctive (or parallel) transition follow by two or more activities. All of the followed activities will be executed at the next step). LOOP Control-flow (an activity has a loop transition. The activities inside the loop area will be executed repeatedly). Fig 1 illustrates the ICN primitive patterns. The Run-time pICN is the discovered ICN Γ from the business process event log, and Γ contains the proportional ε of activity followed by the other activity or Control-flow gate. Fig 2 is one example of pICN discovered

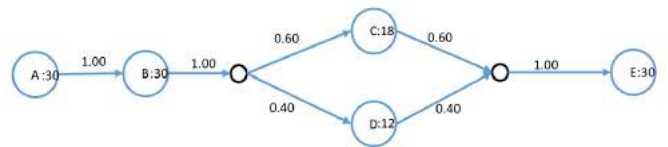


Fig. 2: Example of pICN Discovered from an Event Log

from an event log. In the figure, there are 5 activities (A, B, C, D, E), and 5 proportional values describing the transactions between activities (A $\rightarrow$ B:1.00, B $\rightarrow$ Or-Open:1.00, Or-Open $\rightarrow$ C:0.60, Or-Open $\rightarrow$ D:0.40, C $\rightarrow$ Or-Close:0.60, D $\rightarrow$ Or-Close:0.40, Or-Close $\rightarrow$ E:1.00). The occurrences of activities A, B, C, D, E are 30, 30, 18, 12, 30 respectively.

B. Method and Algorithm

For representing the activities of discovered pICN in XPD, we use the already defined structures in XPD format. However, to represent the proportional of transactions and the occurrences of activities, we have to extend attributes in the XPD output file. We name the extended attribute as **ProportionalAttribute** and **OccurrencesAttribute**. It implies the probability of activity followed by the other activity as well as the occurrences of activities discovered from an event log (Fig 3).

```
<xsd:element name="ProportionalAttribute">
  <xsd:complexType mixed="true">
    <xsd:attribute name="FromActivity" type="xsd:string" use="required"/>
    <xsd:attribute name="ToActivity" type="xsd:string" use="required"/>
    <xsd:attribute name="Value" type="xsd:decimal"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="OccurrencesAttribute">
  <xsd:complexType mixed="true">
    <xsd:attribute name="ActivityId" type="xsd:string" use="required"/>
    <xsd:attribute name="Value" type="xsd:int"/>
  </xsd:complexType>
</xsd:element>
```

Fig. 3: Attribute Extension in XPD Storing Transaction Proportional and Activity Occurrences

Similarly, while representing pICN gate in XPD, there is no LoopGate in the XPD format. To handle this problem, we add gate name attribute "LoopGate" while using activities notation to illustrate gates in pICN. An example of representing LoopGate in XPD is described in Fig 4.

```
<xpd:Activity Id="LOOP-OPEN4" Name="LoopGate">
  <xpd:Route ExclusiveType="Data" GatewayType="Exclusive"/>
  <xpd:TransitionRestrictions>
    <xpd:TransitionRestriction>
      <xpd:Join ExclusiveType="Data"/>
      <xpd:Split ExclusiveType="Data" Type="Exclusive">
        <xpd:TransitionRefs>
          <xpd:TransitionRef Id="LOOP-OPEN4ARAP"/>
          <xpd:TransitionRef Id="LOOP-OPEN4URAP"/>
        </xpd:TransitionRefs>
      </xpd:Split>
    </xpd:TransitionRestriction>
  </xpd:TransitionRestrictions>
  <xpd:NodeGraphicsInfo>
    <xpd:NodeGraphicsInfo BorderColor="#112233" FillColor="#221133"
      Height="40" LaneId="lane1" Shape="ExclusiveGateway" Width="40">
      <xpd:Coordinates XCoordinate="750" YCoordinate="530"/>
    </xpd:NodeGraphicsInfo>
  </xpd:NodeGraphicsInfo>
</xpd:Activity>
```

Fig. 4: An Example of Representing pICN LoopGate in XPD Format

To read the information from pICN and then transform it to XPD, we proceed as follows:

- Create Header for output XPD file γ
- Insert the open tag **xpd:WorkflowProcesses**
- Insert tag **xpd:Activities**
- Visit all ω components in pICN Γ . If ω is an activity then insert the tag **xpd:Activity** and its properties. If ω is a Control-flow gate, then check under OrGate, AndGate, LoopGate and process its information respectively.
- Visit all transactions θ in pICN Γ . Get information about fromActivity, toActivity, proportional and insert into file γ

- Insert the close tag **xpd:WorkflowProcesses**

In the **Algorithm 1**, we use pseudo code for describing how to read information from discovered pICN as well as represent its elements in XPD in detail. The input data is the graph of discovered pICN Γ ; the output data is the XPD file γ .

IV. CONCLUSION

In this paper, we have presented some concepts about XPD, ICN, and pICN. We also surveyed some other research results related to the exploitation and use of XPD data standards in workflow management systems for managing and exchanging business process information. Furthermore, we have proposed an approach to represent the data of discovered pICN, extracted from system process history, in XPD format. The interactive and performance discovered pICN would help analyze information system processes in various workflow systems supporting the XPD format.

In the future work plan, we will proceed to build a complete system, from discovering pICN to represent to XPD format to support the construction of a Process mining system.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant No.2017R1A2B2010697). This work also was supported by Kyonggi University's Graduate Research Assistantship 2019.

REFERENCES

- [1] A. Ghoneimy, A. Zinatbaksh, S. Tiwari, J. D. Wein, A. Jun, and R. S. Simhadri, "Workflow system and method," Jul. 22 2008, uS Patent 7,403,948.
- [2] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-science: An overview of workflow system features and capabilities," *Future generation computer systems*, vol. 25, no. 5, pp. 528–540, 2009.
- [3] Xpd format. [Online]. Available: <https://www.wfmc.org/standards/xpd>
- [4] J. L. Peterson, "Petri nets," *ACM Computing Surveys (CSUR)*, vol. 9, no. 3, pp. 223–252, 1977.
- [5] K. Kim and C. A. Ellis, *Section II / Chapter VII. An ICN-based Workflow Model and Its Advances, Handbook of Research on BP Modeling*. Hershey, Pennsylvania, USA: IGI Global, ISR, 2009.
- [6] R. Shapiro, "A comparison of xpd, bpm and bpm4ws: Cape visions," *Rough Draft*, pp. 1–17, 2001.
- [7] N. Guelfi and A. Mammari, "A formal framework to generate xpd specifications from uml activity diagrams," in *Proceedings of the 2006 ACM symposium on Applied computing*. ACM, 2006, pp. 1224–1231.
- [8] H.-M. Noh, B. Wang, C.-J. Yoo, and O.-B. Chang, "An extension of uml activity diagram for generation of xpd document," in *Asia-Pacific Web Conference*. Springer, 2005, pp. 164–169.
- [9] H. Zha, Y. Yang, J. Wang, and L. Wen, "Transforming xpd to petri nets," in *International Conference on Business Process Management*. Springer, 2007, pp. 197–207.
- [10] P. Yuan, H. Jin, S. Yuan, W. Cao, and L. Jiang, "Wftxb: a tool for translating between xpd and bpm," in *2008 10th IEEE International Conference on High Performance Computing and Communications*. IEEE, 2008, pp. 647–652.
- [11] S.-Y. Lee, "Survey of uml based xpd document for business process modeling," *International Journal of Control and Automation*, vol. 5, no. 2, pp. 11–20, 2012.
- [12] H. Ahn, M. Park, and K. P. Kim, "wois-paan-discovering performer-activity affiliation networking knowledge from xpd-based workflow models."
- [13] H. Ahn, K. Kim, and K. P. Kim, "Generating process entity hierarchies from xpd process models," in *ICEIS (2)*, 2018, pp. 690–695.
- [14] H. Ahn, H. Kim, and K. P. Kim, "Visualizing workflow-supported social networks and their degree centrality measures," *International Journal of Advances in Soft Computing & ICACIT*, vol. 6, no. 3, pp. 191–202, 2020.

Algorithm 1: Represent pICN in XPD format**Input:** Graph of activities in ICN model Γ **Output:** XPD file format γ of Γ

```

1 Initialize  $\gamma = \emptyset$ 
  /* Insert header format of XPD ver 2.1:
  <xpd:Pakage...>, <xpd:PackageHeader>,...
  */
2  $\gamma.InsertHeader()$ 
  /* Insert WorkflowProcesses attribute to  $\gamma$  */
3  $\gamma.InsertTagAttribute("xpd:WorkflowProcesses")$ 
  /* Insert Activities attribute to  $\gamma$  */
4  $\gamma.InsertTagAttribute("xpd:Activities")$ 
  /* Read all node (activities or gates) in  $\Gamma$ ,
  transfer it to tag then add to  $\gamma$  */
5 for  $\forall$  node  $\eta$  in  $\Gamma$  do
  /* Insert Activity attribute to  $\gamma$  */
  6  $\gamma.InsertTagAttribute("xpd:Activity")$ 
  7  $currentNode = \eta.GetCurrentNodeInfo()$ 
  8  $routeType = defaultRouteType, shapeType =$ 
     $defaultShapeType, gateType = \emptyset$ 
  9  $\gamma.InsertTagAttribute("OccurrencesAttribute")$ 
  10  $\gamma.InsertAttribute(type=occurrence,$ 
     $\Gamma.\eta.GetOccurrence)$ 
  11 if  $currentNode$  is OrGate then
  12    $routeType = "Exclusive"$ 
  13    $shapeType = "Exclusive Gateway"$ 
  14    $gateName = "OrGate"$ 
  15 if  $currentNode$  is AndGate then
  16    $routeType = "Parallel"$ 
  17    $shapeType = "Parallel Gateway"$ 
  18    $gateName = "AndGate"$ 
  19 if  $currentNode$  is LoopGate then
  20    $routeType = "Exclusive"$ 
  21    $shapeType = "Exclusive Gateway"$ 
  22    $gateName = "LoopGate"$ 
  23 if  $currentNode$  is OpenGate then
  24    $gateType = "Split"$ 
  25 if  $currentNode$  is CloseGate then
  26    $gateType = "Join"$ 
  27  $\gamma.InsertAttribute(type=activity, currentNode,$ 
     $routeType, shapeType, gateType)$ 
  /* Insert transition attribute to  $\gamma$  */
  28  $\gamma.InsertTagAttribute("xpd:Transitions")$ 
  29 for  $\forall$  edge  $\theta$  in  $\Gamma$  do
  30    $currentEdge = \theta.GetEdgeInfo()$ 
  31    $fromActivityID = currentEdge.GetFromID()$ 
  32    $toActivityID = currentEdge.GetToID()$ 
  33    $\gamma.InsertAttribute(type=transition, fromActivityID,$ 
     $toActivityID)$ 
  34    $\gamma.InsertTagAttribute("ProportionalAttribute")$ 
  35    $\gamma.InsertAttribute(type=proportional,$ 
     $\Gamma.\theta.GetTransctionProportional)$ 
35 Return  $\gamma$ 

```



Dinh-Lam Pham Dinh-Lam Pham is a full-time PhD. student in Kyonggi University, South Korea since March, 2018. He got the MSc in computer science, Thainguyn University, Vietnam (2010). He worked in Thainguyn University from 2008 to 2016 as an IT engineering. From 2016, he works in the Information Technology Institute, Vietnam National University, Hanoi. His research interests include Process Mining, Workflow Management System, Social Network, Intelligent process, Information Control Net.



Hyun Ahn Hyun Ahn is an assistant professor of computer science and engineering department at Kyonggi University, South Korea. He received his B.S., M.S., and Ph.D. degrees in computer science from Kyonggi University in 2011, 2013, and 2017, respectively. His research interests include workflow systems, BPM, business process intelligence, and process mining.



Kyoungsook Kim Kyoungsook Kim is an adjunct professor of the department of computer science and engineering at Kyonggi University, South Korea. She received her B.S. degree in computer science from Kyonggi University in 1984. She also received her M.S. and Ph.D. degrees in computer engineering from Kyunghee University in 2001 and 2019, respectively. She had worked as a researcher and developer at FAST Systems Inc. and OSI Corp. in South Korea. Her research interests include large-scale database systems and applications, enterprise information systems, real-time databases, distributed databases, temporal databases, federate databases, object-oriented database management systems, and the Internet of Things collaboration architectures and services.



Kwanghoon Pio Kim Kwanghoon Pio Kim is a full professor of computer science department and the founder and supervisor of the collaboration technology research laboratory at Kyonggi University, South Korea. He received B.S. degree in computer science from Kyonggi University in 1984. And he received M.S. degree in computer science from Chungang University in 1986. He also received his M.S. and Ph.D. degrees from the computer science department at University of Colorado Boulder, in 1994 and 1998, respectively. At Kyonggi University, he is Dean of the Computerization and Informatics Institute, and the director of the contents convergence software research center, as well. He had worked as researcher and developer at Aztek Engineering, American Educational Products Inc., and IBM in USA, as well as at Electronics and Telecommunications Research Institute (ETRI) in South Korea. In present, he is a vice-chair of the BPM Korea Forum. He has been in charge of a country-chair (Korea) and ERC vice-chair of the Workflow Management Coalition. He has also been on the editorial board of the journal of KSII, and the committee member of the several conferences and workshops. His research interests include groupware, workflow systems, BPM, CSCW, collaboration theory, Grid/P2P distributed systems, process warehousing and mining, workflow-supported social networks discovery and analysis, process-aware information systems, data intensive workflows, and process-aware Internet of Things.