Generating Process Entity Hierarchies from XPDL Process Models

Hyun Ahn[1], Kyoungsook Kim[2] and Kwanghoon Pio Kim[1]
[1]*Department of Computer Science, Kyonggi University, Korea, Republic of*
[2]*Department of Computer Engineering, Kyunghee University, Korea, Republic of*

Keywords: Business Process Intelligence, Hierarchical Property, Process Entity Hierarchy, Hierarchy Generator, XPDL 2.0.

Abstract: Business process intelligence enables us to discover a variety of deep insights about business process execution, and it provides a set of useful methods for related decision-making activities. The hierarchical information that this paper focuses on is an important sort of information and it ought to be used in analyzing hierarchical properties of business processes. In this paper, we present a useful hierarchy generator to make it easier to perform analytics of hierarchical properties among business process entities. To this end, we define an abstracted meta-model that represents hierarchical relations among entity types in XPDL process models. According to the relational rules of the meta-model, a process entity hierarchy can be organized, analyzed, and visualized.

1 INTRODUCTION

Business process management (BPM) technologies and systems have been widely adopted in the various industries where business processes are considered as a critical factor in determining the enterprise value. Accordingly, the importance of intelligent methods for effectively analyzing business process increases to enrich the managing capabilities of process-oriented organizations.

Business process intelligence (Grigori et al., 2004) and process mining (van der Aalst and Weijters, 2004) both are the highly active research fields with the aim of analyzing business processes and discovering valuable business process-related knowledge. In this regard, a variety of approaches and methods are proposed based on the fundamental aspects of the business process, such as control-flow, data-flow, resource, and performance.

The hierarchical structure that this paper focuses on is also an important kind of analytical property pertaining to business process modeling and execution. The main purpose of the analysis of hierarchical information structure is to understand the structural characters of information elements and to evaluate each element by its proportion. Likewise, hierarchical structures and analytical properties exist in business processes and in the environment that manages them (e.g., the explicit hierarchies among entity types in business process models).

In this paper, we present a useful process entity hierarchy generator to make it easier to perform analytics of hierarchical structures consisting of business process entities in XPDL (XML Process Definition Language) process models (WFMC, 2012). More specifically, the generator is supported by the relation rules based on the abstracted meta-model in the hierarchy generation phase. Additionally, as a validation step, we present a simple running example using the sample models to generate, analyze, and visualize the process entity hierarchy.

2 RELATED WORK

In this paper, we concentrate the hierarchical structure and property implied the XPDL-based process models. In this regard, this section briefly introduces the previous works regarding the business process model analysis.

Existing methods and techniques for business process model analysis range from the fundamental property analysis (van der Aalst and van Hee, 2004; Murata, 1989; Li et al., 2004) to the advanced and sophisticated solutions (Dijkman et al., 2012; Khlif et al., 2017; Awad, 2007; Kunze et al., 2011) to meet practical objectives and to achieve advantages in managing business processes. Two prior works (van der Aalst

Ahn, H., Kim, K. and Kim, K.
Generating Process Entity Hierarchies from XPDL Process Models.
DOI: 10.5220/0006756106900695
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 690-695
ISBN: 978-989-758-298-1

and van Hee, 2004; Murata, 1989) described the series of concepts and formal methods for fundamental properties of the process model, such as reachability, soundness, and liveness. In other hands, (Li et al., 2004) presented the mathematical basis for measuring the performance metrics including workload, and turnaround time on the extended Petri nets.

While those prior works mentioned above contributed to the foundation of modeling and verifying business process models, more and more researchers have been interested in developing methodologies to efficiently manage a large collection of business process models. (Awad, 2007) introduced the visual language for expressing queries over a repository of business process models. The author argues that this language allows the reusability of business process model without redundant efforts of modeling. Similarly, (Kunze et al., 2011) presented the measure of model similarity to search similar business process models, which satisfies certain properties.

Although the other works are not included in this paper due to the limitation of space, there is a voluminous set of contributions which focus on business process analysis. However, the hierarchical property of business process has been considered as an insignificant feature in the literature. In only a few studies, the hierarchy concept has been used to express subprocess (van der Aalst and van Hee, 2004) and resource models (Du and Shan, 1999; Kim et al., 2005), but neither of them addresses the analysis with the hierarchical structure composed of process entities.

3 AN ABSTRACTED META-MODEL

In order to generate and analyze process entity hierarchies, we formerly devised the abstracted meta-model (Ahn et al., 2016) from the meta-model defined in the XPDL 2.1 specification (WFMC, 2012). This section describes the details of the abstracted meta-model and hierarchical relations among entity types.

3.1 The XPDL Standard

XPDL is the popular one of the standards for interchanging business process definitions between heterogenous BPM products. It was introduced by the WfMC (Workflow Management Coalition) in 1998 and is currently standardized to the version 2.2. Figure 1 shows the meta-model of XPDL 2.1 specification representing primitive entity types and their relations. These entity types are broadly related to three main aspects: Control-flow, resource, and data.

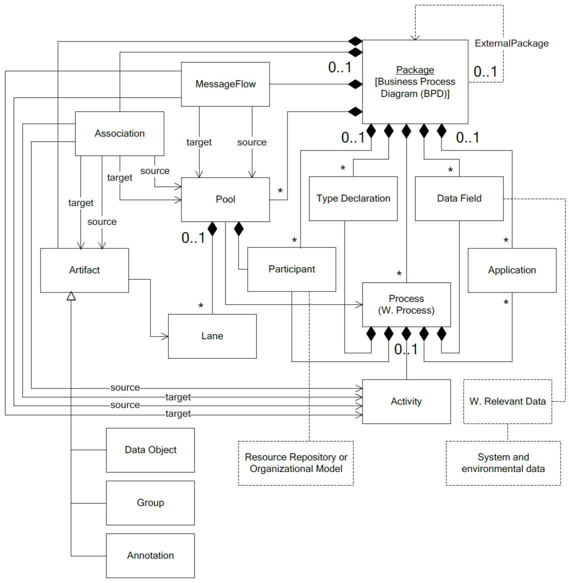


Figure 1: The meta-model of the XPDL 2.1 standard.

3.2 The Abstracted Meta Model and Hierarchical Relations

To facilitate the analytics of process entity hierarchy, we analyzed the meta-model (shown in Figure 1) and devised the abstracted meta-model representing the hierarchical relations between entity types.

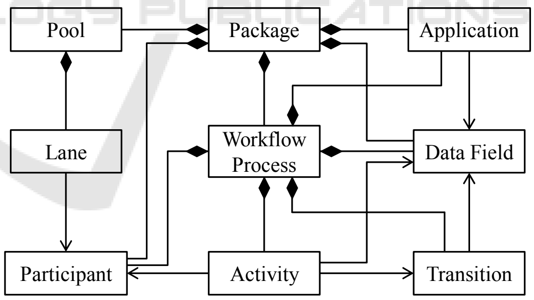


Figure 2: Abstracted meta-model representing hierarchical relations (composition and directed association) between XPDL process entity types.

The abstracted meta-model includes essential entity types which satisfy a hierarchical property and are can be used as a construct to build a process entity hierarchy. The definitions of entity types are described as follows:

- Package: A package corresponds to a set of business processes. In general, it means a grouping of business processes in a specific business domain. Therefore, a package is mapped to the root element in an XPDL-formatted process model.

- Workflow Process: It refers to a business process. In an XPDL file, a <WorkflowProcess> element is a container for the set of its activities and their temporal ordering.
- Activity: A typical activity represents a logical unit of work in a business process. Also, the routing activities (e.g., OR split/join, AND split/join) and events both are types of activity, but since these types do not imply a hierarchical property or related semantics we only take account of typical activities in constructing a process entity hierarchy.
- Transition: Transitions are core elements of control-flow aspect in a workflow process. Each transition connects from a predecessor activity to a successor activity. In cases of transitions connected from a disjunctive routing activity, a transition condition is attached to a corresponding <Transition> element. Accordingly, the transition is activated only when the transition condition holds true.
- Pool: This entity type is the top-level container for resources allocated in a business process. In business process modeling systems, it is used to graphically augment the view of the resources to a business process diagram. According to the syntax of XPDL, a <Pool> element includes a set of <Lane> elements as child nodes.
- Lane: Lanes are used to describe responsibilities of a certain set of activities in a business process. Therefore, a lane is often relevant to a business role (e.g., administrator and contract manager), a system (e.g., a legacy system), a business unit (e.g., department and project team).
- Participant: A participant is a resource capable of performing an activity. It can be not only a human resource but also a machine resource.
- Data Field: Data fields are created and consumed within each business process execution. They are used as variables to exchange intermediate results between activities or evaluate transition conditions.
- Application: This entity type refers to a software program or a service invoked from business process enactments. They provide functionalities required to carry out tasks involved in a business process.

Hierarchical relations the meta-model includes are divided into two relation types: Composition and directed association. A composition (depicted as –◊) refers to an explicit hierarchical relationship between a superior entity type and subordinate entity type. These relationships are explicitly represented by hierarchical XML elements in an XPDL file. For instance, each <Package> element has a set of subordinate <WorkflowProcess> elements. A directed association (depicted as →) refers to an association relationship that is navigable in only one direction and is interpreted as an implicit hierarchical relationship, which is not explicitly represented in an XPDL file, but it implies the semantic information related the hierarchical properties of corresponding business processes. For instance, an <Activity> element includes reference information about an identifier of a <Participant> element instead of explicit element inclusion.

4 GENERATING PROCESS ENTITY HIERARCHIES

According to the abstracted meta-model, we implement a prototype system for generating a process entity hierarchy that is based on the relational rules of the meta-model. This section describes the hierarchy generation steps with the implemented system. Additionally, we present a running example to validate the feasibility of analytics of process entity hierarchies.

4.1 Process Entity Hierarchy Generator

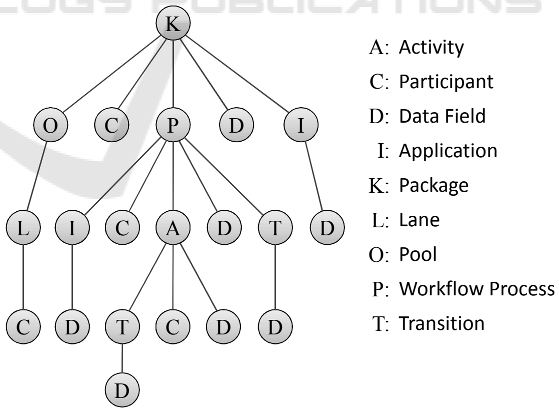


Figure 3: Tree representation of the hierarchical relations of the abstracted meta-model.

The implemented generator is constrained by the rules of hierarchical relation for the compliance with the meta-model that we devised. Figure 3 shows the entity tree representing hierarchical relations between entity types. A hierarchy generation is implemented by traversing the entity tree from the first selected entity type to a specific entity type that is the final destination (except the root entity type). Additionally, all the entity types, excluding the leaf entity types, can be the root entity type of a generated hierarchy since the minimum required levels of hierarchy is two levels.

By using the graphical user interface provided by the system, users can easily and interactively organize a new hierarchy. The graphical navigation functionality offers users available options of entity types in the phase of hierarchy generation. The simple example of organizing a hierarchy is shown in Figure 4, and its screen capture shows the situation of the hierarchy generation. The hierarchy consists of three layered entity types (package, pool, lane), and it is currently incomplete and has only one available entity type (participant) that the user can select.

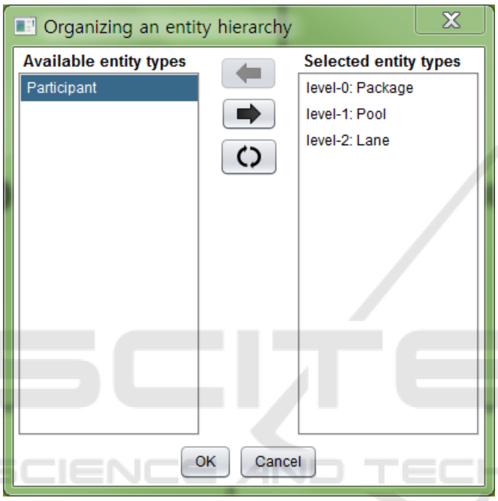


Figure 4: Screen capture of organizing a hierarchy using the graphical user interface.

After the user completes the hierarchy generation, the hierarchy is automatically fleshed out to the full process entity hierarchy through the mapping of entity information in XPDL models to the hierarchy. Figure 5 shows the generated process entity hierarchy. Consequently, the user can perform analytics of hierarchical properties by applying analysis techniques to the process entity hierarchy.

4.2 Running Example

To validate how useful the hierarchy generator is, we demonstrate a simple running example using sample XPDL process model.

The sample model contains one process package (Commercial Banking) including two business processes (Fund Transfer, Data-driven Fund Transfer), and other entity information. From the view of resources, the sample model also includes three pools, eight lanes, and nine participants.

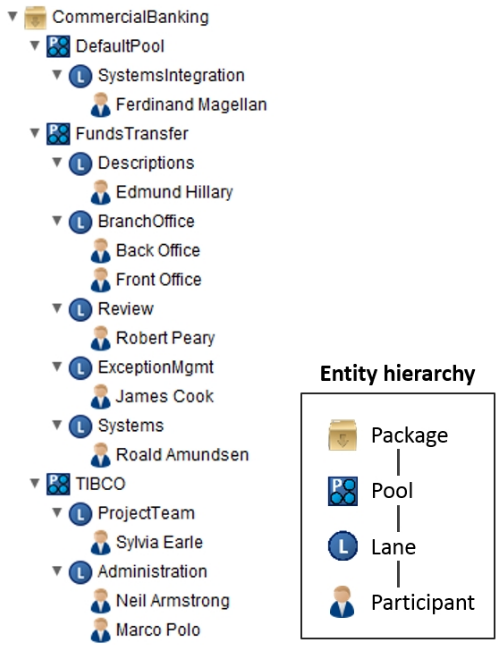


Figure 5: Generated process entity hierarchy (Package–Pool–Lane–Participant).

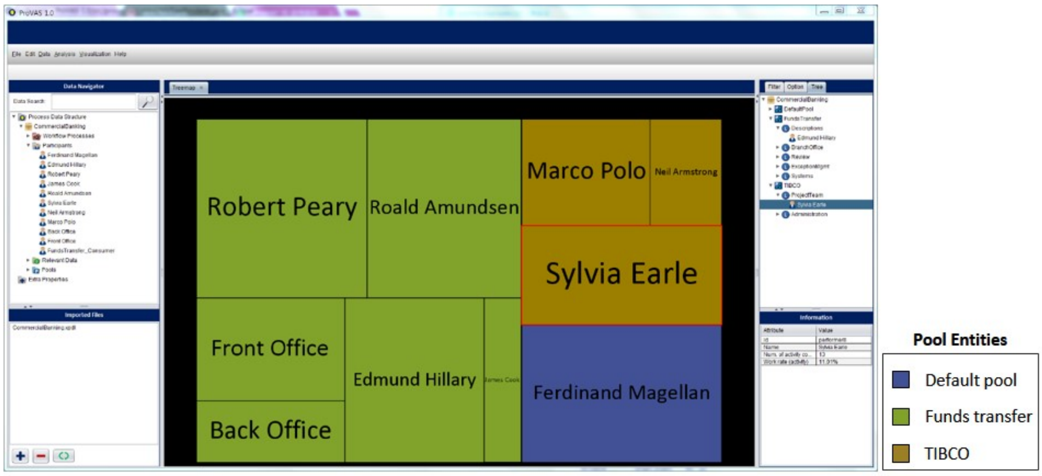
The results of visual analytics executed by the implemented system with the running example is shown in Figure 6. At first, entire entity information included the input XPDL file is collected to the system through the parsing phase (left side).

To analyze a hierarchy of resource entity types, the process entity hierarchy (Package–Pool–Lane–Participant) is generated by the GUI-based hierarchy organizer and afterward concretized with the corresponding collected entity information (right side).

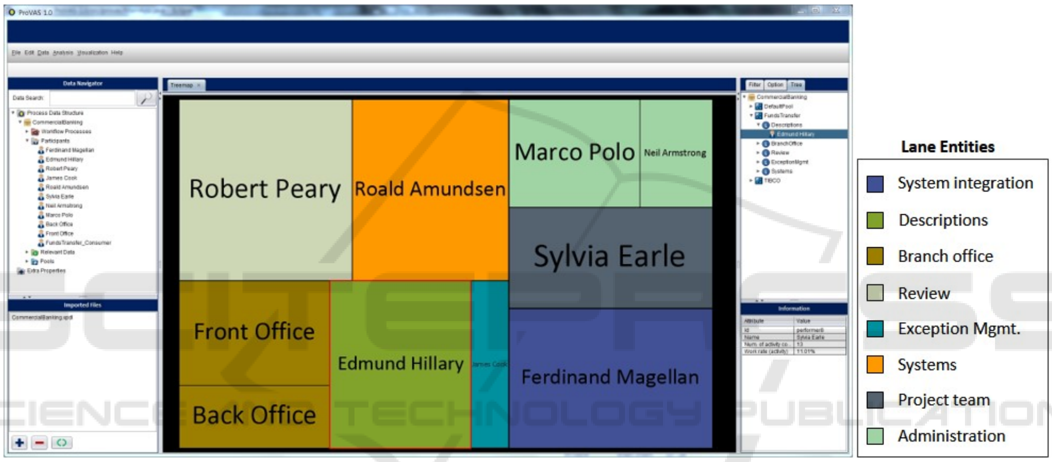
After the step of hierarchy generation is completed, the system visualizes the process entity hierarchy applied the pseudo-measures of work rate, which means the portions of activity instances performed by each participant. The visualization exploits the treemap method (Bruls et al., 2000) and the color filter (by each pool entity), and the node size represents the work rate measures. Even though the visualization result is less effective to demonstrate the hierarchy and related properties, we confirmed that the feasibility of the process entity hierarchy generation through this running example.

5 CONCLUSION

To address analytical properties of the hierarchical structure of business processes, we proposed the XPDL-based process entity generator in this paper.



(a)



(b)

Figure 6: Visualizations of the analysis result of the generated process entity hierarchy applied the participants work rate (activity instance) measures with two color coding methods: The pool-based color coding (a), the lane-based color coding (b).

The generator is governed by the rules of hierarchical relations defined in the abstracted meta-model. Conclusively, through the running example, we confirmed that the generator is fruitful to easily create a process entity hierarchy from XPDL process models.

As a future work, we are planning to devise a comprehensive meta-model to ensure that the hierarchy generator is compatible with other standards, including BPMN, BPEL, and XES. As another issue, we are interested in an effective visualization of the process entity hierarchy. The visualization issue is not regarded as the main concern in this paper, but it is also crucial to perform business process analytics. However, the presented visualization results are limited in effectively displaying a process entity hierarchy and its analysis result. To overcome this shortcoming, we will be studying new visualizations that effectively conveys the information or knowledge of hierarchical properties of business processes.

ACKNOWLEDGEMENTS

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). Also this research was partially supported by the contents convergence software research center at Kyonggi University funded by the GRRC program (Grant No. 2017-0194) of Gyeonggi Province, South Korea.

REFERENCES

Ahn, H., Park, M., and Kim, K. P. (2016). Analyzing hierarchical properties of entities in xpdl-based workflow models. In *Information Science and Technology, 2016. 11th Asia Pacific International Conference on*, pages 63–65.

Awad, A. (2007). Bpmn-q: A language to query business processes. In *EMISA*, volume 119, pages 115–128.

Bruls, M., Huizing, K., and Van Wijk, J. J. (2000). Squarified treemaps. In *VisSym*, pages 33–42. Springer.

Dijkman, R. M., La Rosa, M., and Reijers, H. A. (2012). Managing large collections of business process models-current techniques and challenges. *Computers in Industry*, 63(2):91–97.

Du, W. and Shan, M.-C. (1999). Enterprise workflow resource management. In *Research Issues on Data Engineering: Information Technology for Virtual Enterprises, 1999. RIDE-VE'99. Proceedings., Ninth International Workshop on*, pages 108–115. IEEE.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., and Shan, M.-C. (2004). Business process intelligence. *Computers in industry*, 53(3):321–343.

Khlif, W., Khlif, W., Ben-Abdallah, H., Ben-Abdallah, H., Ben Ayed, N. E., and Ben Ayed, N. E. (2017). A methodology for the semantic and structural restructuring of bpmn models. *Business Process Management Journal*, 23(1):16–46.

Kim, K.-H., Yoo, H.-J., and Kim, H.-S. (2005). A process-driven e-learning content organization model. In *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*, pages 328–333. IEEE.

Kunze, M., Weidlich, M., and Weske, M. (2011). Behavioral similarity–a proper metric. *Business Process Management*, pages 166–181.

Li, J., Fan, Y., and Zhou, M. (2004). Performance modeling and analysis of workflow. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 34(2):229–242.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

van der Aalst, W. M. P. and van Hee, K. M. (2004). *Workflow management: models, methods, and systems*. MIT press.

van der Aalst, W. M. P. and Weijters, A. J. M. M. (2004). Process mining: a research agenda. *Computers in industry*, 53(3):231–244.

WFMC (2012). Xml process definition language version 2.2.