A Process Warehouse Model Capturing Process Variants

Lisana Berberi[a], Johann Eder[*,a], Christian Koncilia[a]

[a] Department of Informatics-Systems, Alpen-Adria Universität Klagenfurt, Austria

Abstract. *Process Warehouses are a well-established means for analysing the execution of business processes and the computation of key performance indicators. We propose a new model for process warehouses which is better suited to cope with business processes which have many variants. We present a meta-model of processes with a notion of generic activities which then is used to automatically generate a generalisation hierarchy for process variants along which OLAP operations can be performed.*

Keywords. Business Process Model • Process Warehouse • Process Variant

1 Introduction

Process Warehouses (Benker 2016; Eder et al. 2002; Pau et al. 2007) are an appropriate means for analysing the performance of business process execution using well established data warehouse technology and on-line analytical processing (OLAP) tools. In particular, they allow the definition, computation and monitoring of key performance indicators along several dimensions. Typical dimensions in process warehouses are process, time, actor, geographic location. While most of the dimensions are organized in hierarchies supporting roll-up and drill-down operations, the process dimension usually is relatively flat, often comprising just two levels: activity and process, sometimes augmented with a part-of hierarchy but typically without a generalization hierarchy.

Some aspects of processes are still poorly supported such as processes with variants (Döhring et al. 2014) or interorganisational processes (Groiss and Eder 1997). Frequently processes exist in several different variants or versions within the same enterprise and even more so between enterprises. These variants are due to different regulations in different countries, variations due to different requirements for different branches of an enterprise (Kop et al. 2005; Mayr et al. 2007) or different decision histories and responsibilities. Variants also arise due to process evolution and the arising differences add additional complexity to modelling temporal data warehouses (Eder et al. 2001). Even if all these variants were expressed in a single process definition (with the excessive use of xor-splits), the resulting processes are large, difficult to understand and to communicate and overloaded, and new process definitions still comprise of all the past processes definitions they should replace. Analysing sets of process variants is cumbersome with current process warehouse technology.

In this paper we propose the core of a process warehouse model with a generalization hierarchy of processes which captures process variants. This generalization hierarchy can be generated from a meta-model of business process models which introduces the notion of generic activities which generalize a set of activities (e. g., pay by credit card, by check, or by third-party (PayPal) could all be generalized to an activity payment).

Based on given hierarchies of activities we define generalizations of processes for the "process" dimension of a process warehouse. This hierarchy can the be used to roll-up or drill down when analysing the logs of the executions of the various process variants and it makes it much easier to compare key-performance indicators between different variants at different levels of genericity.

* Corresponding author.
E-mail. johann.eder@aau.at

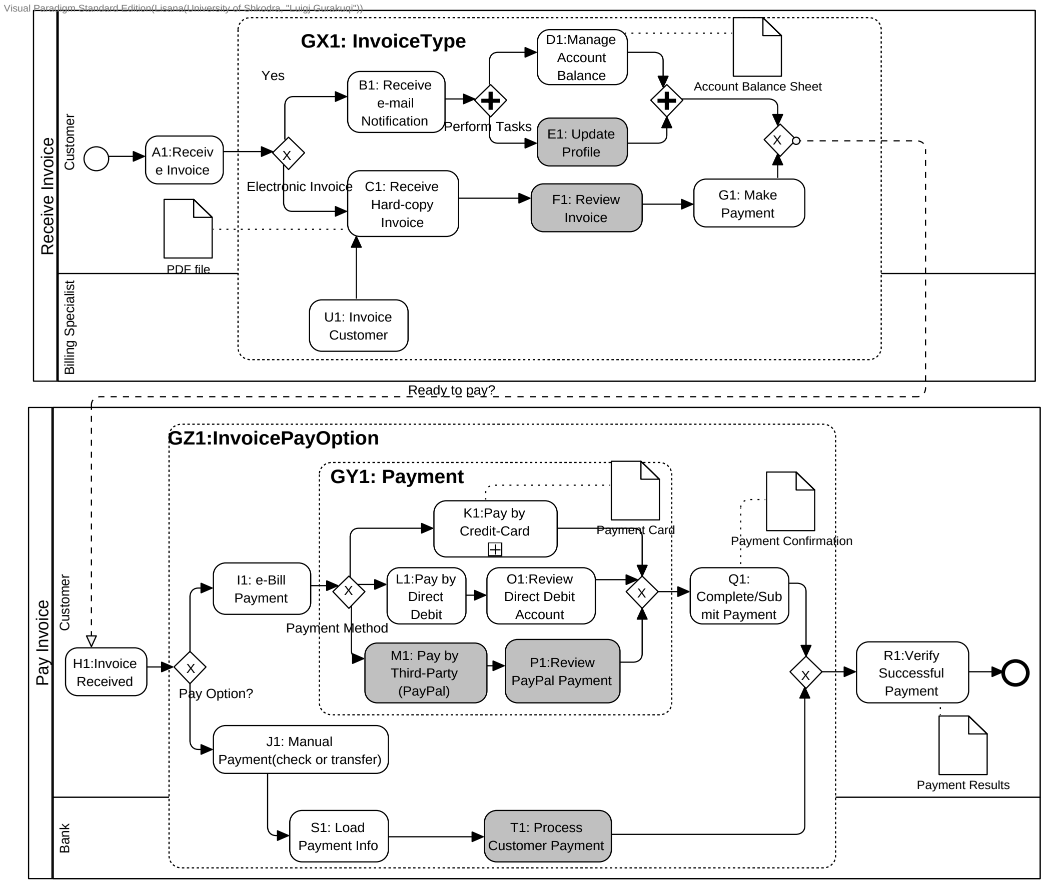

Figure 1: Processing Customer Invoice Payment in Texas/USA.

2 Motivating Example

Let's assume we have a core business process, e. g., Processing Customer Invoice Payments of a financial administration agency that is modelled as an interaction between two processes named *ReceiveInvoice* and *PayInvoice*.

ReceiveInvoice process consists of a set of activities that checks if a customer has requested electronic or hard-copy invoice for goods or services, while *PayInvoice* process consists of a set of other activities that submit or complete with the payment after a customer invoice is received.

We illustrate our approach of generic processes and generic activities with 2 variants of this business process in two different states Texas and New York as shown in Figures 1 and 2. The differences between variants are highlighted with light gray.

Both variants start with Activity *A1: Receive Invoice* followed by a decision point (depicted with an X diamond) where one of the outgoing activities i. e., *B1: Receive e-mail Notification* or *C1: Receive Hard-copy Invoice* is to be executed depending on the type of the invoice.

After receiving e-mail notification two parallel activities should be executed: *D1: Manage Account Balance* or *E1: Update Profile* in variant 1 whereas in variant 2 *D2: Manage Account Balance* and *E2: Review Payment History*.

In variant 1 if the received invoice is a hard-copy invoice which is issued by a billing specialist when activity *U1: Invoice Customer* is enacted. Afterwards *F1: Review Invoice* followed by *G1: Make Payment* is executed.

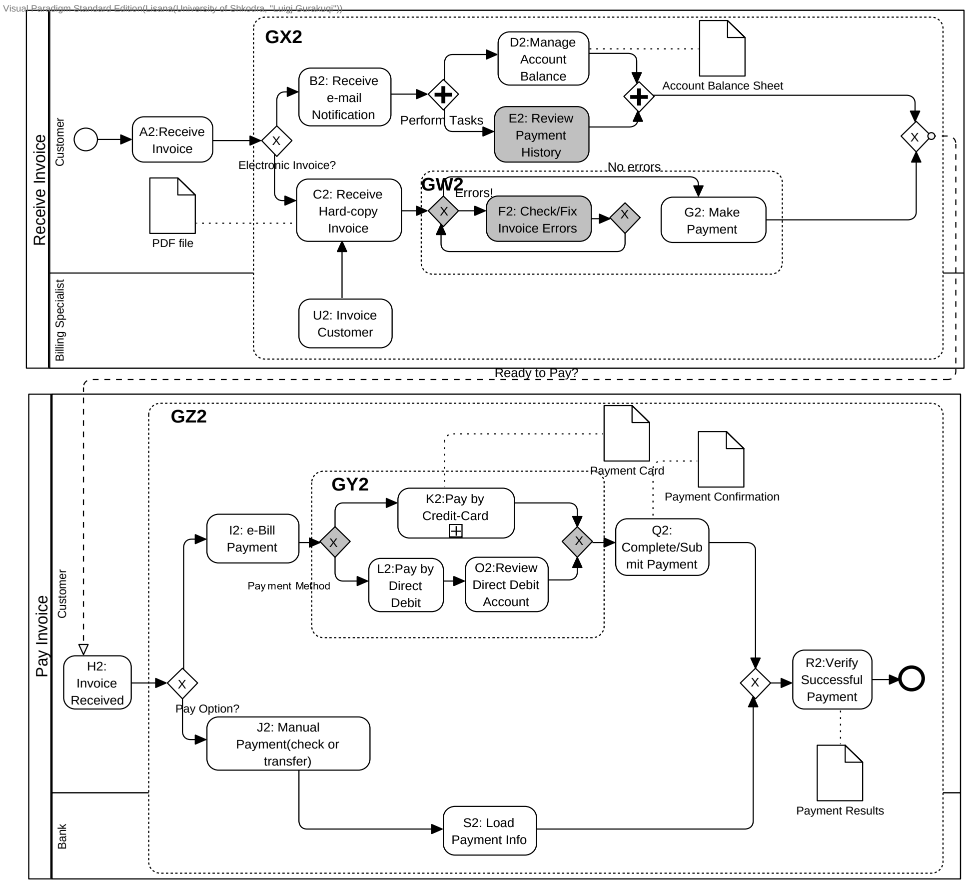

Figure 2: Processing Customer Invoice Payment in New York/USA.

Whereas, in variant 2 if the received invoice is a hard-copy invoice then a loop decision point that checks for the invoice errors is executed. If no errors are found then the flow is shifted to the second process *PayInvoice*, otherwise *F2: Check-/Fix Invoice Errors* is executed several times until the invoice errors are fixed.

The invoice may be paid manually or electronically, depending on this option two activities might be executed: *I1: e-Bill Payment* or *J1: Manual Payment (check or transfer)* in both variants. If electronic payment is chosen, then in variant 1 three possibilities are offered *(K1: Pay by Credit Card* or *L1: Pay by Direct Debit* followed by *O1: Review Direct Debit Account* or *M1: Pay by Third-Party (PayPal)* followed by *P1: Review PayPal Payment)* and only two in variant 2 (pay by credit card or by direct debit).

The sub process *K1: Pay by Credit Card* checks if it's enough credit to pay the invoice order. If yes, then activity Charge credit is executed otherwise Notify client is executed. But, if manual payment is chosen then in variant 1 activity S1: Load Payment Info is executed from a bank representative, followed by *T1: Process Customer Payment*, whereas only activity *S2: Load Payment Info* is executed in variant 2. After selecting the payment method activity *Q1: Complete/Submit Payment* is executed, a payment confirmation document is generated.

Finally, both variants end with executing *R1: Verify Successful Payment* activity.

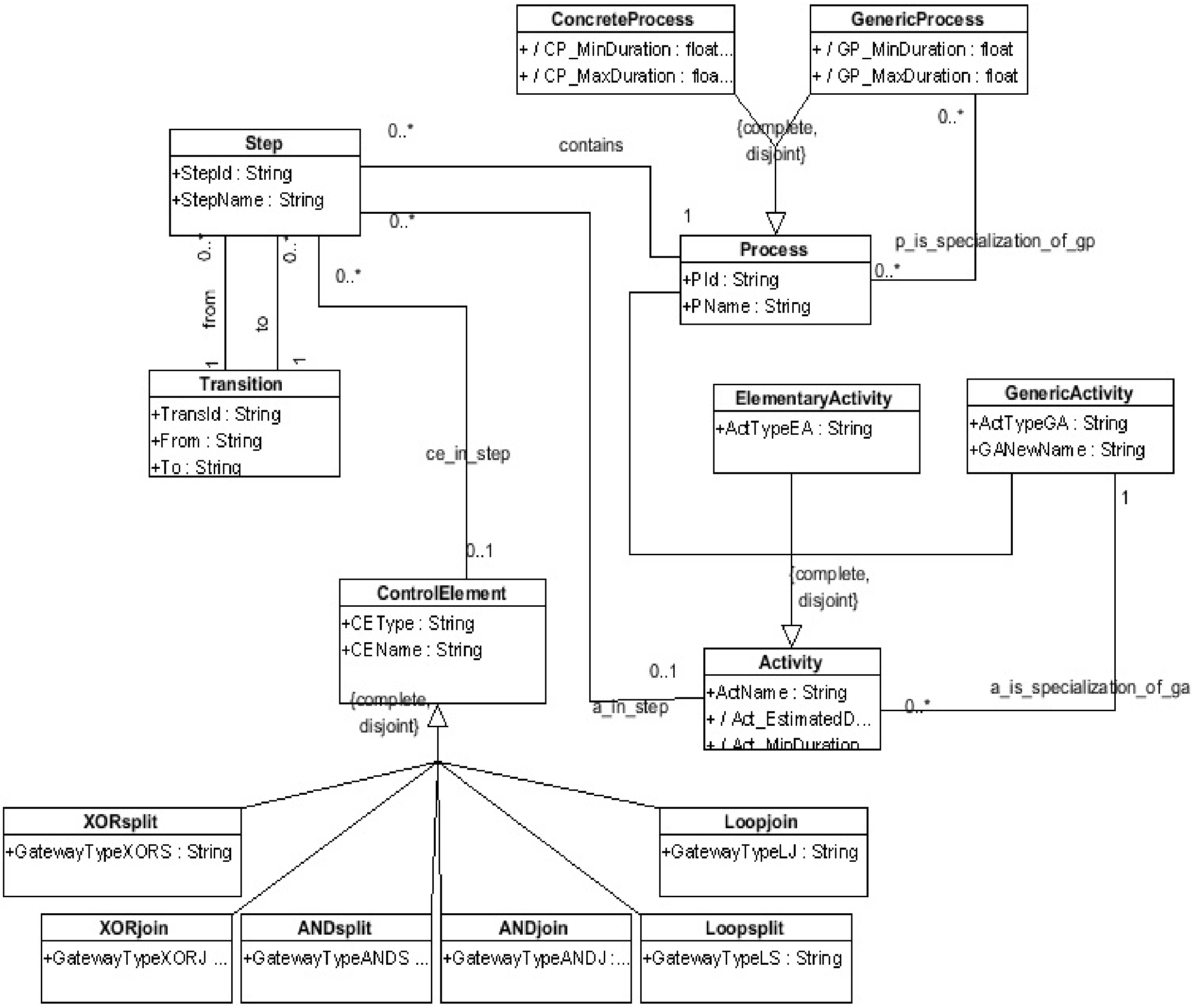

Figure 3: Process meta-model capturing modelling elements.

3 Process Meta-Model with Generic Activities

A *process* is a collection of activities, participants, and dependencies between activities. Activities correspond to individual steps in a business process, participants (software systems or users) are responsible for the enactment of activities, and dependencies determine the execution sequence of activities and the data flow between them (Eder et al. 2002). The process meta-model shown in Figure 3 captures process model elements and their variants from a design-time perspective. A *step* is a concrete invocation of activity in a process and it can be either *Activity* (e. g., activity A) or *ControlElement* (e. g., XORSplit). Each step has different predecessors (from-relationship) and successors (to-relationship), which is expressed by the association class Transition.

An *activity* (the smallest unit of work scheduled by a workflow engine during process enactment) can be of the following subtypes: *elementary activity*, *generic activity* or *process (sub-process)*, where each of these elements are represented through respective classes in the meta-model in a generalization hierarchy (cf. Fig.3). An elementary activity is an atomic/uncompounded activity,

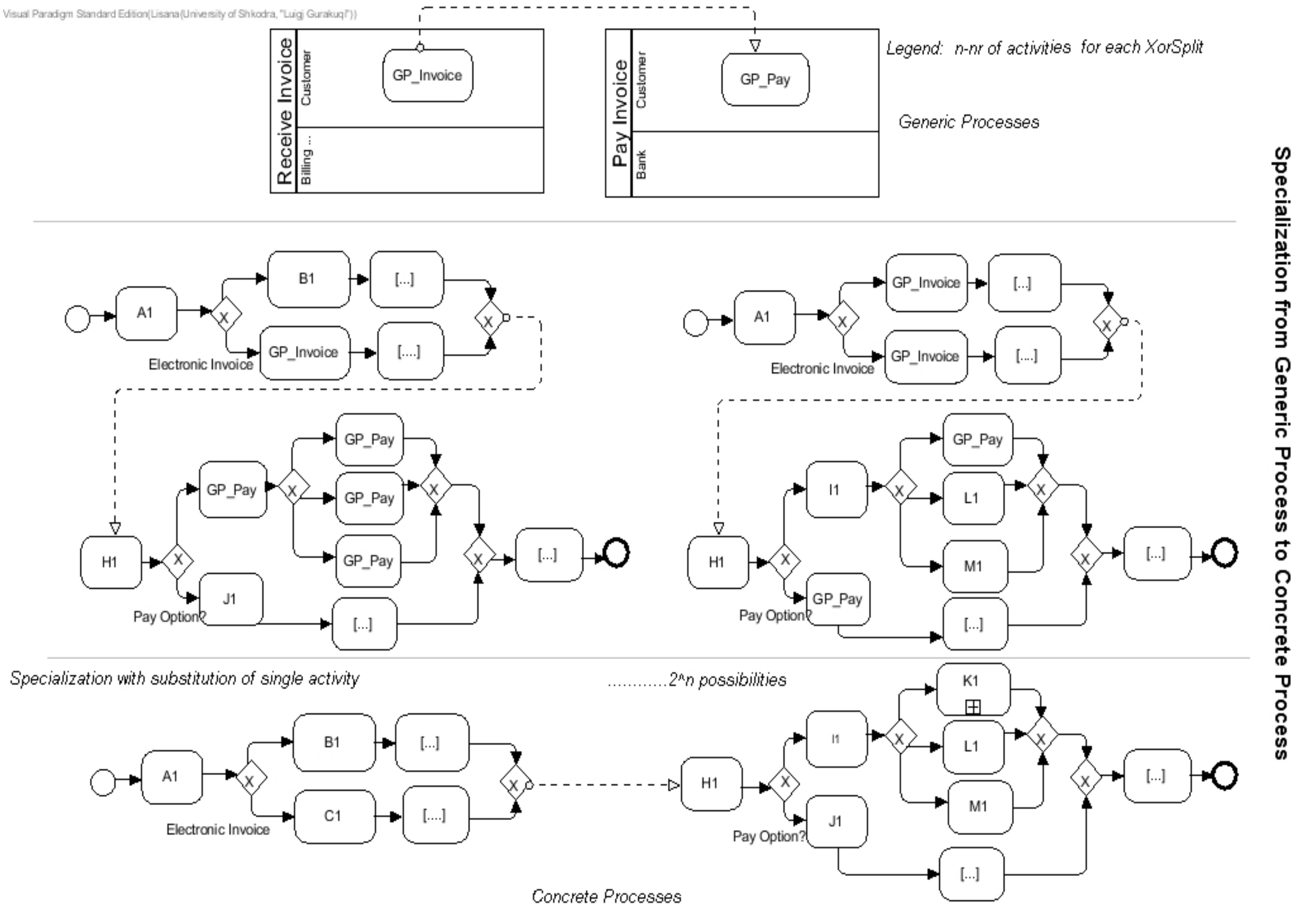

Figure 4: An excerpt of specialization from Generic Processes to Concrete Processes.

e. g., activity *A1: Receive Invoice*. A *sub-process* (complex activity) is composed of other activities, e. g., *K1: Pay by Credit-Card* composed of activities *KA1: Check Credit*, *KB1: Charge Credit* and *KC1: Notify Client*. A *process* in a process definition consists of many steps which are logically related in order to achieve a common goal, represented with class *ConcreteProcess (CP)* in our model. Here, we introduce two notions: generic activities and generic processes.

A *generic activity (GA)* is defined as a step in a process that might be realized by different activities, e. g., GX1, GX2 as depicted with a dotted line rectangular in Figs. 1and 2. In the metamodel a generic activity and its specializations are related by 'a_is_specialization_of_ga' in Fig. 3. For example, the subprocess in Fig. 1 Electronic Invoice and activities B, C are specializations of the generic activity GX1.

A *generic process (GP)* is defined as a process that contains at least one generic activity, e. g., GP_Pay consists of generic activities GY1 and GZ1 as shown in Fig. 1. CP and GP are modelled as disjoint subtypes of Process super type class, thus a CP cannot contain any GA.

A *substitution* replaces a generic activity g in a process model with one of the activities a, which are specializations of g. A process P is a *specialization* of a generic process G, if P can be derived from G by substituting one of the generic activities g of G with one of g's specializations. e. g., the process *PayInvoice* is a specialization of the generic process GP_Pay. Fig. 4 shows specializations by substitution of a single activity from GP to CP.

For the purposes of capturing process variants, the identification of multiple appearances of the same activity is very important, as it allows to aggregate measures of the same activity in different positions of different workflow variants.

4 The Generic Process Warehouse Model for Process Variants

Process warehouses typically feature the dimensions process, organization (department hierarchy), actor, geolocation, time. The process dimension usually only represents the part-of decomposition of processes. With the introduction of generic processes as discussed in the preceding section we are able to also provide consolidation hierarchies for process variants. The PWH schema is derived based on the workflow meta model. A Process dimension will be derived with respective attributes (cf. section (a) of Fig. 5) that stores information about which are the steps, activities, processes that are planned for a process enactment to achieve a specific goal. Some other typical dimensions from resource or organizational perspectives might be Participant (alternatively named agent) and Geolocation. We decided to separate these two dimensions to express the fact that different users can belong to different departments in different processes.

Participant stores information about users (human or system) with specific role, e. g., billing specialist, that are responsible for the execution of activities. Geolocation stores information about the geographic locations of a branch structured in different organization units (departments) where a user belongs. Time is another important dimension for our process warehouse that stores information about time needed to execute an activity etc. Hierarchy consolidation paths from the highest level to the lowest one are defined for every dimension table, to show the relationship between their relative attributes. At the lowest level of process dimension is step within the workflow schema. To consolidate the steps to the relevant activities and to consolidate the activities to the corresponding processes of different variants, two types of hierarchies are defined (cf. Fig. 5).

In process dimension the hierarchy coloured with light gray colour express the fact that a step is rolled-up to an activity, then an activity to a generic activity, whereas the hierarchy coloured with light blue colour express the fact that a step is rolled-up to a process, then a process to a generic activity or to a generic process. An example of instances (process attributes members) is shown in section (b), i. e., some possible execution paths (an excerpt of them) e. g., ‘A1⟶B1⟶[D1E1]⟶H1⟶I1⟶K1⟶Q1 ⟶R1’ is rolled-up to higher levels ‘A1⟶B1⟶[D1E1]⟶H1⟶GZ1⟶R1’ (cf. from light gray coloured hierarchy) and ‘A1⟶GX1⟶GP_Pay’ (cf. from from light blue coloured hierarchy) etc., and then every of them is rolled-up to the highest level that contains only generic processes, i. e., ‘GP_Invoice⟶GP_Pay’.

Time has two hierarchies named month and week hierarchy with specific consolidation paths e. g., month hierarchy where a day is rolled-up to a month, a month to a quarter and a quarter to a year, as shown in section b of Figure 5.

The *Participant* dimension with the lowest level of the participant him/herself is consolidated to a combination of users and roles, expressing the fact that a user can have a specific role in an organization unit. The *Geolocation* dimension with the lowest level organization unit (OU) consolidated to a super organization unit (superOU), e. g., Management Department can have a higher hierarchy Financial Department. From superOU we consolidate to a branch, a branch to a city and a city to a country to express the fact that different departments can be part of different branches, and different branches can be located to different cities and to different countries, e. g., the customer payment process in USA and Canada.

So, typical OLAP roll-up and drill down operations can be performed, i. e., from a specific activity or a process we can roll-up to a generic activity and the other way round, from generic activity we can drill-down to activity or process.

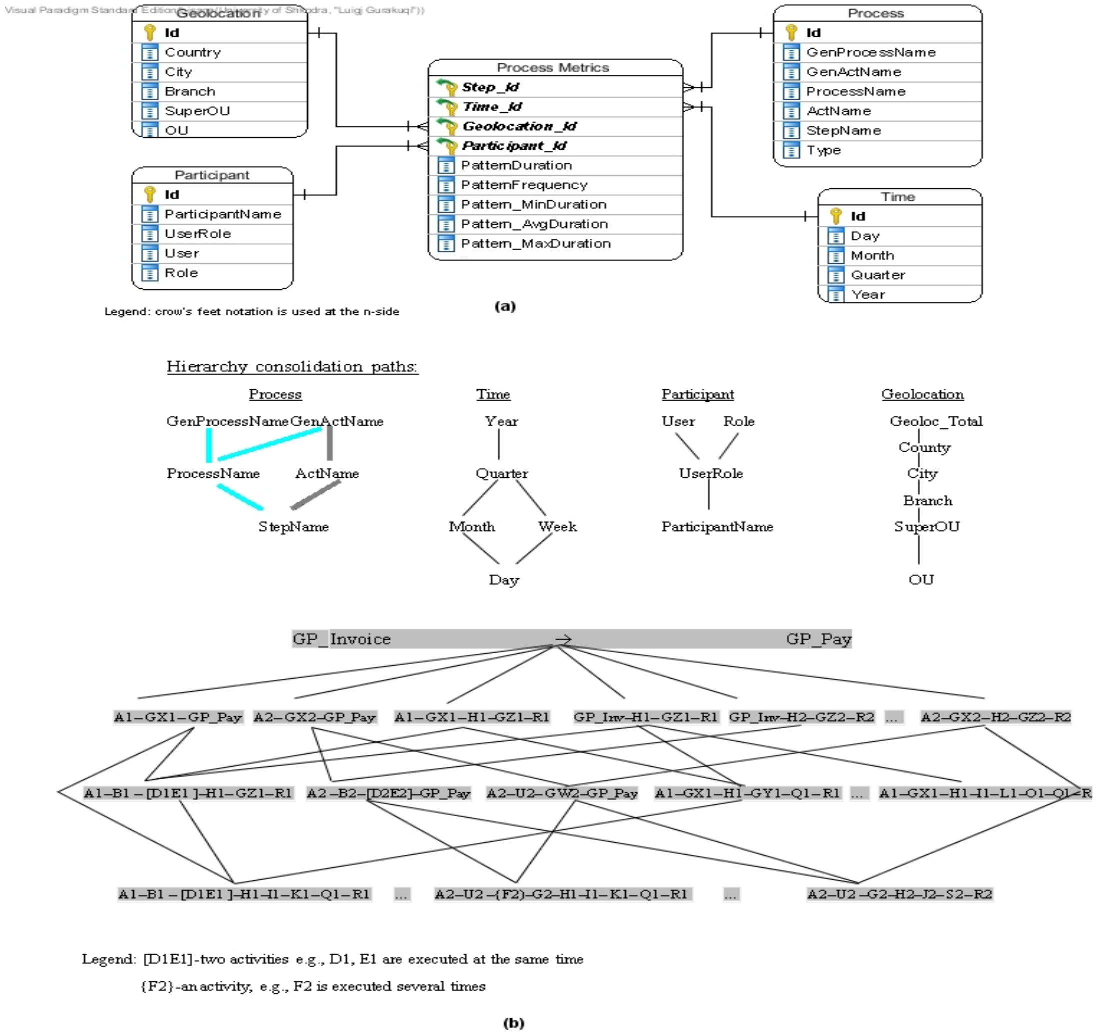

Figure 5: The process warehouse: (a) star schema and (b) example of instances.

From a process we can roll-up to a generic process and drill down the other way round.

Now we can express structural and complexity metrics in process models, called process metrics, e. g., key performance indicator (KPI) for measuring the duration (in time units) or cost of each process patterns (path) along different variants, i. e., Duration of process path starting from activity H1 if customer selected to pay by credit-card and ended to activity R1 (H1⟶I1⟶K1⟶Q1⟶R1). Other KPIs might be defined to calculate the most frequent pattern/behaviour after a split condition type is selected in process variants. The generic process structure consolidates different variants in a multidimensional way, e. g., to derive the average, minimum and maximum duration of payments across all different variants of the payment process.

5 Related Work

Enhancing analysis of business processes by employing data warehousing and data mining technologies has attracted research over the last 15 years, however, the problem of how to deal with process variants is still not covered satisfactorily.

Some studies (Eder et al. 2002; Koncilia et al. 2015; List et al. 2002; Pau et al. 2007) have been proposed on designing data warehouses for workflow audit trails called 'workflow logs' to exploit the valuable information they contain. (Eder et al. 2002; Pau et al. 2007) construct the respective logical data warehouse models using ADAPT (Application Design for Analytical Processing Technologies) notation.

(Benker 2016) recently proposes to derive data warehouse structures from the meta model of BPMN in order to be portable between application domains and to be stable in case of changing workflows. The data warehouse schema, however, does not support analysis of process variants.

Approaches like (Niedrite et al. 2007; Shahzad and Zdravkovic 2012) presented a goal-driven (i. e., a goal is a state of a process in terms of the quality of the service property that is intended to be achieved) requirement analysis together with the method to obtain the conceptual model of a data warehouse.

To apply state-of-the-art analysis in different workflow application domains, especially in Surgical Workflows, multidimensional modelling seems a promising solution as it allows viewing data from different perspectives and at different granularities. (Mansmann et al. 2007a,b) designed a surgical process recording scheme as UML class diagram. For the convergence of business process model and multidimensional model they found common abstraction between them.

A generic solution for warehousing business processes validated over HP and its customer processes is proposed in (Casati et al. 2007). They abstracted process models and then mapped the process progression (i. e., associating the start and the completion of each step) to events occurring in the source systems.

(Koncilia et al. 2015) focused on analysing complex workflow logs, i. e., different splits/joins and loops, by means of OLAP tools. They defined different sequence of events captured from the trace Log after importing into the DW.

Recently, process mining approaches propose techniques for multiple inter-related processes that may change over time (Aalst 2013) in form of process cube constructed by 3 dimensions, i. e., class type, event class, and time window. In (Liu et al. 2011) a novel E-Cube model is presented which combines complex event processing (CEP) and OLAP techniques for efficient multi-dimensional event pattern analysis at different abstraction levels. They built an event pattern hierarchy that integrates complex event patterns specified using sequence, negation and concept abstractions.

A multidimensional event log (MEL) analysis is proposed by (Vogelgesang and Appelrath 2015) which maps the structure of event logs to a data cube and organizes instances i. e., cases and events on different levels.

An evaluation of various process warehousing approaches through a comprehensive survey is discussed by (Shahzad and Johannesson 2009).

6 Conclusions

We argued that process variants can be efficiently and effectively analysed with a process warehouse using specialization/generalization hierarchies between processes and activities. The introduction of a genericity relationship between activities and generic activities allows to generate such generalization hierarchies for processes to structure the "process" dimension of process warehouses, which then can be used to analyse process metrics with the usual OLAP operations such as to roll-up and drill down the dimension hierarchy. In particular it allows to analyse variants of the same process as individuals together, or partitioned in similarity groups.

References

van der Aalst W. M. P. (2013) Process Cubes: Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining. In: Asia Pacific Business Process Management. Springer Verlag, pp. 1–22

Benker T. (2016) A Generic Process Data Warehouse Schema for BPMN Workflows In: Business Information Systems, BIS, Proceedings Springer International Publishing, pp. 222–234

Casati F., Castellanos M., Dayal U., Salazar N. (2007) A Generic Solution for Warehousing Business Process Data. In: Proceedings of the 33rd International Conference on Very Large Data Bases. VLDB Endowment, pp. 1128–1137

Döhring M., Reijers H. A., Smirnov S. (2014) Configuration vs. adaptation for business process variant maintenance: an empirical study. In: Information Systems 39, pp. 108–133

Eder J., Koncilia C., Morzy T. (2001) A Model for a Temporal Data Warehouse. In: Open enterprise solutions: systems, experiences and organizations Workshop (OES-SEO 2001), pp. 48–54

Eder J., Olivotto G. E., Gruber W. (2002) A Data Warehouse for Workflow Logs. In: Proc. Int. Conf. on Engineering and Deployment of Cooperative Information Systems. Springer, pp. 1–15

Groiss H., Eder J. (1997) Workflow systems for inter-organizational business processes. In: SIGGroup Bulletin 18, pp. 23–26

Koncilia C., Pichler H., Wrembel R. (2015) A Generic Data Warehouse Architecture for Analyzing Workflow Logs. In: Advances in Databases and Information Systems. Springer, pp. 106–119

Kop C., Vöhringer J., Hölbling M., Horn T., Mayr H. C., Irrasch C. (2005) Tool Supported Extraction of Behavior Models.. In: ISTA Vol. 63, pp. 114–123

List B., Schiefer J., Tjoa A. M., Quirchmayr G. (2002) Multidimensional business process analysis with the process warehouse. In: Knowledge Discovery for Business Information Systems. Springer, pp. 211–227

Liu M., Rundensteiner E., Greenfield K., Gupta C., Wang S., Ari I., Mehta A. (2011) E-Cube: Multi-dimensional Event Sequence Analysis Using Hierarchical Pattern Query Sharing. In: Proc. of the 2011 ACM SIGMOD International Conference on Management of Data. ACM, pp. 889–900

Mansmann S., Neumuth T., Scholl M. H. (2007a) Multidimensional data modeling for business process analysis. In: Conceptual Modeling-ER 2007. Springer, pp. 23–38

Mansmann S., Neumuth T., Scholl M. H. (2007b) OLAP Technology for Business Process Intelligence: Challenges and Solutions In: Data Warehousing and Knowledge Discovery: 9th International Conference, DaWaK 2007 Springer, pp. 111–122

Mayr H. C., Kop C., Esberger D. (2007) Business process modeling and requirements modeling. In: Digital Society, 2007. ICDS'07.. IEEE

Niedrite L., Solodovnikova D., Treimanis M., Niedritis A. (2007) Goal-driven Design of a Data Warehouse-based Business Process Analysis System. In: Proceedings 6th WSEAS Int. Conf. On Artificial Intelligence, Knowledge Engineering and Data Bases - Volume 6. World Scientific, Engineering Academy and Society (WSEAS), pp. 243–249

Pau K. C., Si Y. W., Dumas M. (2007) Data warehouse model for audit trail analysis in workflows. In: Student Workshop of the 2007 IEEE International Conference on e-Business Engineering (ICEBE 2007). IEEE

Shahzad K., Johannesson P. (2009) An evaluation of process warehousing approaches for business process analysis. In: Proceedings of the International Workshop on Enterprises & Organizational Modeling and Simulation. ACM

Shahzad K., Zdravkovic J. (2012) Process warehouses in practice: a goal-driven method for business process analysis. In: Journal of Software: Evolution and Process 24(3), pp. 321–339

Vogelgesang T., Appelrath H.-J. (2015) A Relational Data Warehouse for Multidimensional Process Mining. In: International Symposium on Data-Driven Process Discovery and Analysis. Springer, pp. 155–184