Article

Functional Integration with Process Mining and Process Analyzing for Structural and Behavioral Properness Validation of Discovered Processes from Event Log Datasets

Kwanghoon Pio Kim [1,†,‡] 0000-0001-6320-2556

1   Data and Process Engineering Research Lab., Division of Computer Science and Engineering, KYONGGI UNIVERSITY; kwang@kgu.ac.kr

Abstract: Process (or business process) management systems fulfill defining, executing, monitoring and managing process models deployed on process-aware enterprises. Accordingly, the functional formation of the systems is made up of three subsystems such as modeling subsystem, enacting subsystem and mining subsystem. In recent times, the mining subsystem has been becoming an essential subsystem. Many enterprises have successfully completed the introduction and application of the process automation technology through the modeling subsystem and the enacting subsystem. According as the time has come to the phase of redesigning and reengineering the deployed process models, from now on it is important for the mining subsystem to cooperate with the analyzing subsystem; the essential cooperation capability is to provide seamless integrations between the designing works with the modeling subsystem and the redesigning work with the mining subsystem. In other words, we need to seamlessly integrate the discovery functionality of the mining subsystem and the analyzing functionality of the modeling subsystem. This integrated approach might be suitable very well when those deployed process models discovered by the mining subsystem are complex and very large-scaled, in particular. In this paper, we propose an integrated approach for seamlessly as well as effectively providing the mining and the analyzing functionalities to the redesigning work on very large-scale and massively parallel process models that are discovered from their enactment event logs. The integrated approach especially aims at analyzing not only their structural complexity and correctness but also their animation-based behavioral properness, and becomes concretized to a sophisticated analyzer. The core function of the analyzer is to discover a very large-scale and massively parallel process model from a process log dataset and to validate the structural complexity and the syntactical and behavioral properness of the discovered process model. Finally, this paper writes up the detailed description of the system architecture with its functional integration of process mining and process analyzing. And more precisely, we excogitate a series of functional algorithms for extracting the structural constructs as well as for visualizing the behavioral properness on those discovered very large-scale and massively parallel process models. As experimental validation, we apply the proposed approach and analyzer to a couple of process enactment event log datasets available on the website of the 4TU.Centre for Research Data.

Keywords: structured information control net; process mining; process analyzing; structural analysis; behavioral analysis; process rediscovery

1. INTRODUCTION

A (business) process management system (BPMS) is defined as a system that fully automates the definition, analysis, deployment, execution, and monitoring & controlling of work procedures in a process-aware enterprise. One of the essential components of BPMS is the modeling subsystem (Buildtime) that is substantially supported by a graphical and formal methodology of the process model. The conventional modeling subsystem is equipped

with a series of functional components supporting from modeling a process model with graphical notations to analyzing and deploying it onto the enacting subsystem (Runtime). In recent times, keeping abreast of the process mining [1][2] with the emerging concept of enterprise big data, the advanced BPMSs have begun to be furnished with the process mining functionality [3] as an essential subsystem (Miningtime). In other words, the time of the process reengineering and redesigning has come truly in the process-aware enterprises and organizations. This atmosphere booming process redesigning and reengineering is becoming a catalyst for triggering emergence of the concept of process mining and process-centered knowledge mining that rediscovers several perspectives of processes such as control flow, data flow [4], resource allocation planning [5], social, and organizational perspectives from their execution histories and traces collected at runtime. In consequence, it ought to be coming the time of the functional integration of the rediscovery functionality of the Miningtime subsystem and the analysis functionality of the Buildtime subsystem. Actually, the motivation and necessity of this functional integration came up with a couple of experimental validation studies of the process mining algorithm. Note that one of the minded process models is made up of 624 business activities and so it is almost impossible to fulfill any further meaningful analytical work due to the huge number of activities.

The purpose of this paper is to integrate the verification functionality of the Buildtime subsystem that is exclusively charged for visual verification and structural analysis [6][7][8] of the process models rediscovered from their enactment event trace datasets by the Miningtime subsystem. That is, we are able to accomplish the enterprise-wide goal which is minimizing the loss of effectiveness and the depreciation of efficiency in controlling the life-cycles of processes. The process designer is able to inspect the syntactical correctness as well as the structural performance of the corresponding process models prior to redesigning and reengineering them. This inspection ought to be much more effective if the rediscovered process models are especially characterized with very large-scale and massively parallel structures [9]. The Miningtime subsystem's functionality is to rediscover a process model from its enactment event trace dataset and to represent the rediscovered one via both the textual and graphical representation language and notation like XPDL [10] and BPMN[1] [11], respectively. The Miningtime subsystem eventually transforms the graphical model into a textual model represented by a standardized process definition language like XPDL that is the standardized XML specification released by the process management coalition. Before deploying the rediscovered process model, it is very important to inspect its syntactical correctness as well as its structural properness to minimize the loss of effectiveness and the depreciation of efficiency in managing the life-cycle [12] of a rediscovered process model.

The most important benefit of the proposed approach ought to be the seamless integration of the process mining functionality and the process analyzing functionality. Once we discover a process model, then we need to analyze various aspects of the discovered process model. Especially, this paper concerns about the cases of when the discovered models are very large-scale and massively parallel in terms of their structural building blocks. In particular, we assume that those process models are represented as structured information control nets and are rediscovered from their enactment event trace datasets by the process mining and analyzing system. That is, in order to efficiently and seamlessly verify the structural correctness (matched pairing and proper nesting) [13][14] of the discovered very large-scale and massively parallel process models, we develop a process mining and analyzing system based upon the functional integration approach proposed in this paper. The implemented system supports the structural verification and the visual simulation [15] with analytical control-path animation. The structural components of rediscovered process model are made up of the entity types such as activity, role, actor, invoked applications and relevant data, and the association types such as control-flow association, data-flow association, actor-role association, activity-role association and activity-application association. In other words, the system not only applies the structural verification functionality to the associative components of a discovered process model, but also applies the behavioral simulation and animation functionality to the entity components of the rediscovered process model. We assume that the rediscovered process model is textually represented in the

---

[1]   XPDL stands for XML process definition language, and BPMN stands for business process modeling notation that is the standardized graphical notation released by the BMIDTF (business modeling integration domain task force) of the object management group.

standardized XML-based process definition language (XPDL)[10], and that it is also graphically represented in a form of the structured information control nets (SICN) [16][14].

In terms of organizing the paper, we describe a functional architecture integrating the process analyzing and the process mining right after the literature surveys in the next section. In the consecutive sections, we specify the formal and graphical structural characteristics of the very large-scale and massively parallel process models, describe the functional details and design artifacts of the architectural components, and carry out a couple of experimental deployments and analytics for those very large-scale and massively parallel process models, which are discovered from a dataset that is released from the 4TU.Centre for Research Data [17].

2. RELATED WORKS

This paper aims to design and implement an XPDL-based process mining and analysis system with supporting the information control net process discovery functionality. The system seamlessly performs the multi-functionality of discovering, analyzing and simulating very large scale and massively parallel process models. Almost all of the previous tools and systems for the process analysis and simulation activities could be applied to the discovered and explored process models by the process mining systems. However, it should not be easy for the process simulation activity and the process mining activity to be seamlessly integrated with each other.

— Firstly, we try to investigate the state-of-the-art in the process analysis and simulation tools and systems, separately. The authors of [6] defined the concept of the structured process model and its properties, which are the properties that can be checked up by the system proposed in this paper, and described a taxonomy for analyzing unstructured processes, which are characterized by the properties of improper nestings or mismatched split-join pairings.

— In [4] and [8], the authors proposed a functional mechanism for analyzing XPDL-based process models and their relevant data modification sequences, and we use the functional mechanism in implementing the simulation-based performative analysis function of the system in this paper.

— The authors of [18] gave a definite intuition to the system of this paper, in which the authors described a template that was built in the simulation language, Arena. The positive effect of using the template is in decreasing the gap between the conceptualization activities and their real process models to be analyzed.

— The necessity of the system is given by [19], in which the authors emphasized the importance of the comparison analysis between designed processes and redesigned processes by using the process analysis and simulation tools and systems. The authors also discussed a number of analysis tools that are relevant for the business process field, evaluated their applicability for business process analysis and simulation, and formulated the technical recommendations and further research issues. The system proposed in this paper is based upon the theory of the information control net modeling methodology.

— Whereas [7] presented a tool of process analysis and simulation based upon the theory of Petri net modeling methodology, which is named as Yasper [7] as a tool for modeling, analyzing and simulating process models based on Petri nets.

— The authors of [20] proposed a fully automatic method to simplify BPML(business process model and notation) process models and described a two-phase iterative algorithm to achieve this simplification, which follows a heuristic approach that makes intensive use of a pattern repository.

— The study of [21] introduced a goal-driven process evaluation method based on process mining for healthcare processes. The proposed method comprises the following steps: defining goals and questions, data extraction, data preprocessing, log and pattern inspection, process mining analysis and generating answers to questions, evaluating results, and initiating proposals for process improvements. Additionally, the authors applied the proposed method to the surgery process of a university hospital in Turkey as a case study.

— The authors of [22] proposed three strategies (based on exhaustive search, genetic algorithms and a greedy heuristic) that use event data to automatically derive a process model from a configurable process model that better represents the characteristics of the process in a specific branch. Also, they implemented these strategies and tested them in both business-like event logs as recorded in a higher educational enterprise resource planning system and a real case scenario involving a set of Dutch municipalities.

    — In [23], the authors introduced the method of Accimap from the discipline of accident analysis to analyze the diagnosis results of process mining by creating a complaint handling service process management Accimap model and carrying out across different system levels. Additionally, they performed a case study in a big manufacturing company in China to verify the proposed method and approach. The case study identified 42 complaint handling process management factors and created the complaint handling process management Accimap model as a final outcome.

    — In the study of system inference, the authors of [2] proposed an integrated approach with process mining and fuzzy methods to build a system structure of the fuzzy discrete event system specification (Fuzzy-DEVS) model from system behavior. The proposed approach consists of three stages: (1) extraction of event logs from data by using the System Entity Structure method; (2) discovery of a transition system, using process discovery techniques; (3) integration of fuzzy methods to automatically generate a Fuzzy-DEVS model from the transition system. Finally, it took a plugin in the process mining framework (ProM) environment for inferring a Fuzzy-DEVS model from an event log dataset and carried out a simulation by using the SimStudio tool.

    — And other research outcomes [24][25] based on the Petrinet-based process modeling and analyzing methodology, like from the research group of the Eindhoven first and now of the Aachen University, from the research group of the Melbourne School of Engineering, and from the research group of the University of Camerino, provided us the essential intuitions as well as the functional scopes and definitions for specifying the very useful functional requirements and specifications of the system implemented in this paper.

Conclusively speaking, it might be the first trial in the literature so far for seamlessly integrating the SICN-based process analysis functionality with the process mining functionality. If those SICN-based process models, which are characterized with the properties of very large scale and massively parallel structures, are discovered from process enactment event log datasets, then this seamless integration plays much more important and meaningful in terms of performing their verification analysis and checking up simulation-based performative properness. We assume that the original process models embedded in the corresponding process enactment event log datasets are unknown.
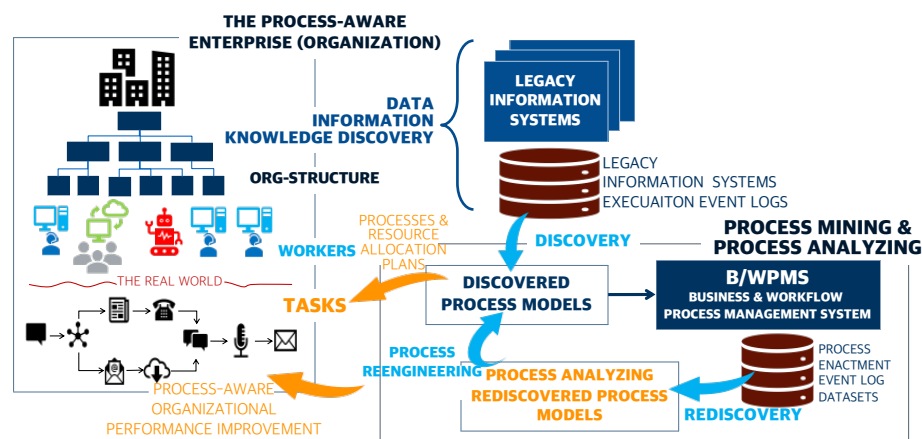

Figure 1. Mining and Analyzing Activities on a Process-Aware Enterprise: The Rationale of Functional Integration of Process Mining and Process Analyzing

3. AN INTEGRATED FUNCTIONAL ARCHITECTURE

So far, we suggested the necessity of the conceptual integration with the analysis function of the Buidtime subsystem and the discovery function of the Miningtime subsystem. Based upon the clear and reasonable conceptual goal, we need to devise a concrete integration approach and architecture in this section. Additionally, we describe the details of implementing the devised integrated architecture as a system. Consequently, in the

next consecutive subsections we excogitate the details of the concrete goal and functional scope for developing the integrated architecture and system supporting the process mining functionality and the sophisticated process analyzing functionality.

Figure 1 illustrates a situational snapshot of the process-aware enterprise and organization working with the Miningtime subsystem supporting the process discovery and rediscovery functions [26][27] and the Buildtime subsystem supporting the structural analysis functions as well as the visual-simulation functions. Particularly, we emphasize that the discovery is to explore business-activity processes from the event logs of the executions of the traditional information systems, while the rediscovery is to do mining the enacted business-activity processes from the event logs and traces stored whenever the corresponding business-activity processes are enacted and executed by their process management system. In this section, we focus on proposing a concrete integration approach of the rediscovery function through the Miningtime subsystem and the analysis and simulation functions through the Buildtime subsystem.

Basically, the functional integration of the process mining and the process analyzing is to pursue the seamless operations from the rediscovery functions to the simulation-based analysis functions. More precisely, the process mining is to rediscover process models from process enactment event log datasets. The process analyzing is to fulfill the structural and simulation-based analysis on those rediscovered SICN process models. Note that in this paper we especially focus on the process visual-simulation analysis that supports the simulation-based performative properness on the rediscovered process models. The rediscovered process models of the process mining are represented in a mathematical form of the structured information control nets (SICN) [14] and in a graphical form of the GraphML [28] file format with the Graphviz graph visualization open source library. Our experimental cases are also based upon those SICN process models discovered from XES-formatted datasets of process enactment event logs and traces, which are chosen from the 4TU.Centre for Research Data [17]. Therefore, the functional integration between the process mining and the process analyzing ought to be realized seamlessly through transforming the graphical form of Graphviz and GraphML into the textual form of XPDL [10] of the discovered SICN process models.
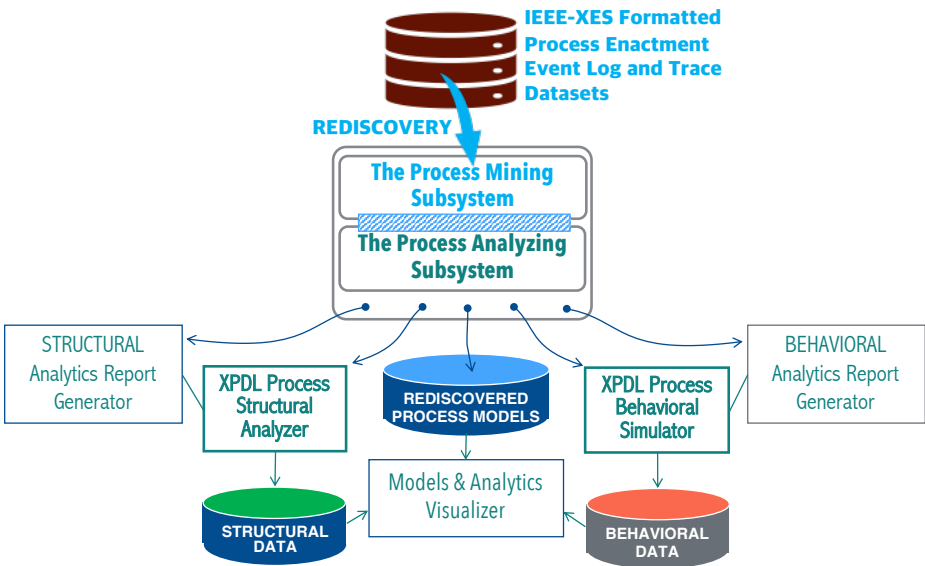

Figure 2. The Ultimate Architecture of the Functional Integration of Process Mining and Process Analyzing

The overall integrated architecture and its components are illustrated in Figure 2. The upper part of the figure shows an integrated architecture consisting of the process mining subsystem and the process analyzing subsystem, and the lower part is the functional components of the integrated architecture and system, such as report generators, structural analyzers, performative simulator and visualizer. Especially, the XPDL-based process instance tracing functionality of the performative simulator ought to be a focal function that is provided by the integration of the process mining and the process analyzing. Conclusively, the scope and goal of the paper

is to develop the XPDL-based process's structural analyzer and performative simulator in conjunction with the SICN-based process mining system. In the consecutive sections, we describe the design and implementation details of these architectural components. The eventual goal of the integrated architecture is to analyze the structural and behavioral aspects of a SICN-based process model rediscovered from an enactment event log dataset.

4. PROCESS MINING FUNCTIONALITY

The crucial advantage of using the analysis functionality being conjunct with the mining functionality ought to be that we can be automatically analyzing such rediscovered process models. Especially, when the underlying process models of the datasets are unknown and these rediscovered process models are very large scale and massively parallel process models, the users can realize the real values and merits of the process analysis functionality. This subsection describes the architectural details of the process mining functionality as a process rediscovery subsystem (Miningtime) that provides the input to the process analysis subsystem (Buildtime).
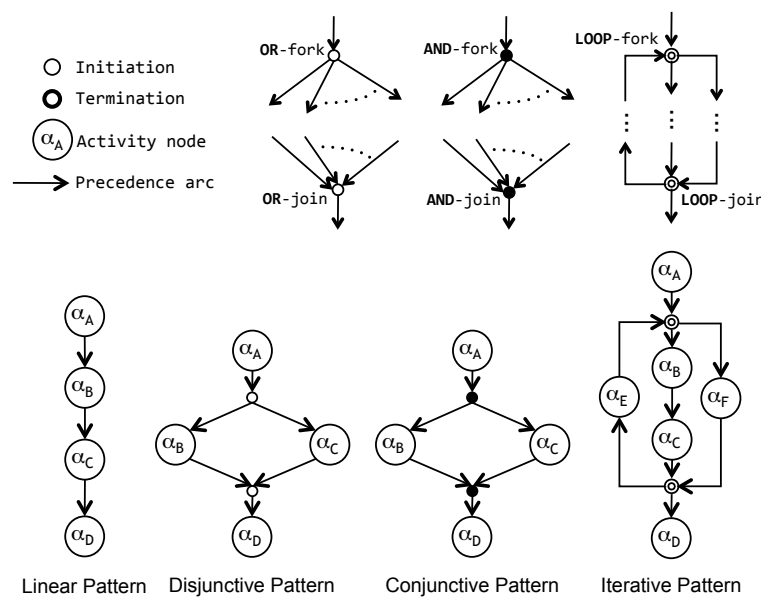

Figure 3. The Structural Control-Flow Primitives in Structured Information Control Net Process Models

4.1. Structural and Behavioral Process Patterns

The structured information control net (SICN) process model [6][14] is a mathematical and graphical graph model to build the theoretical formalism of process-aware business activities. It is well known as the fittest methodology to describe and analyze control and information flows among the process-aware business activities by capturing temporal transitions and associations among the essential entities, such as business activities, roles, actors, applications and repositories. Furthermore, it has been used within actual as well as hypothetical automated offices to yield a comprehensive description of activities, to test the underlying office description for certain flaws and inconsistencies, to quantify certain aspects of office information flows, and to suggest possible office restructuring permutations. In this subsection, especially, we focus on the process-centered (control-flow) structural attributes in a process model of the structured information control nets.

The structural and behavioral complexity of a SICN process model is determined by the control-flow attributes that are built through the combinational permutation of the four types of the structural transition patterns, such as sequential, disjunctive, conjunctive and iterative transition patterns, as graphically shown in Figure 3 and formally defined in [Definition 1]. That is, the structure of a SICN process model is built by a set of business activities connected by temporal orderings (control flow) called activity transitions. The business activities can be temporally related to each other by combining sequential transition type, disjunctive transition
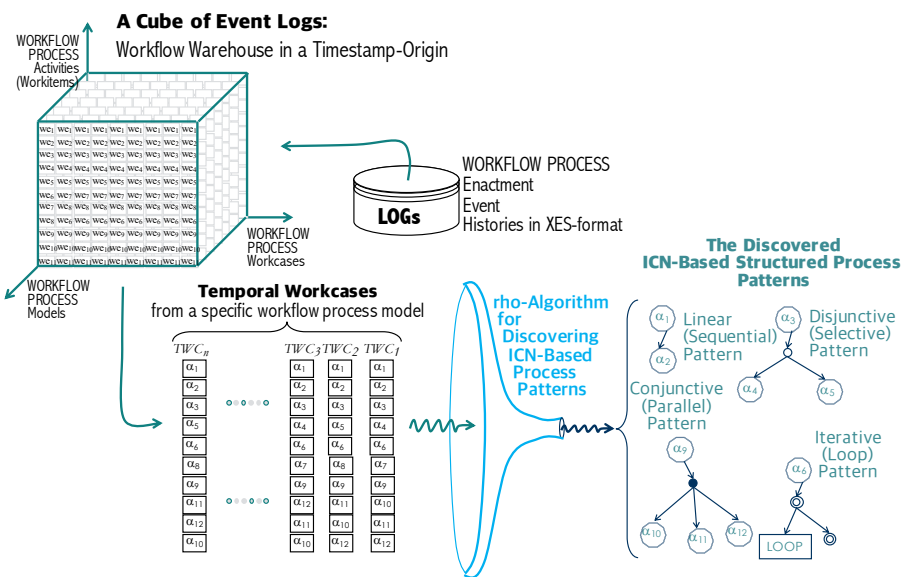
Figure 4. The Process Mining Framework

type (after activity α_A, do activity α_B or α_C, alternatively) with predicates attached, conjunctive transition type (after activity α_A, do activities α_B and α_C concurrently), and iterative transition type. Naturally, the composite transition types (except the sequential transition type) are represented by the so-called gateway activity types with split and join properties. In the SICN process model, the gateway activity types must keep the properties of matched pairings and proper-nestings with splits and joins.

[Definition 1] Structural Control-Flow Patterns in Structured Information Control Nets. The basic control-flow structure of a SICN process model is formally defined through 4-tuple $\Gamma = (\delta, \kappa, \mathbf{I}, \mathbf{O})$ over a set of $\mathbf{A}$ activities (including a set of group activities), a set $\mathbf{T}$ of transition conditions, where

— $\mathbf{I}$ is a finite set of initial input repositories, assumed to be loaded with information by some external process before execution of the model;

— $\mathbf{O}$ is a finite set of final output repositories, which is containing information used by some external process after execution of the model;

— $\delta = \delta_i \cup \delta_o$: Control-Flow Structural Attributes
where, $\delta_o : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an activity to its set of (immediate) successors, and $\delta_i : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an activity to its set of (immediate) predecessors;

— $\kappa = \kappa_i \cup \kappa_o$: Transition-Condition Associative Attributes
where $\kappa_i : \mathbf{T} \longrightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of an activity $(\alpha \in \mathbf{A})$ to its incoming transition-conditions on each arc, $(\delta_i(\alpha), \alpha)$; and $\kappa_o : \mathbf{T} \longrightarrow \wp(\mathbf{A})$: is a multi-valued mapping function of an activity $(\alpha \in \mathbf{A})$ to its outgoing transition-conditions on each arc, $(\alpha, \delta_o(\alpha))$.

4.2. The Process Mining Approach: ρ-Algorithm

This subsection simply describes a process mining framework that is proposed and implemented by the authors' research group. As shown in Figure 4, the framework is able to rediscover all the four types of primitive process patterns in a structured information control net process model from a process enactment event log dataset by applying the concept of mass with their enactment event log occurrences. The conceptual initiation of the process mining framework is introduced in this subsection. Especially, the core component of the framework is the ρ-Algorithm [26][29]. In the ρ-Algorithm, the symbol and name of *rho* (ρ) comes from the A programming

language (APL) firstly released in 1960's. The function ρ, coded like ρX in APL, implies that it gives the number of elements in X, from which the concept of ***mass*** comes. The central idea of the rediscovery algorithm of the framework is exactly same to the implication of the APL function, rho (ρ). The ρ-operator gives the masses (occurrences) of all the activities in the event log dataset of a corresponding process model. In the paper, the emphasis is placed on rediscovering very large scale and massively parallel process models that are usually built by a combination of the four types of primitive process patterns, such as linear, disjunctive (exclusive-OR), conjunctive (parallel-AND) and repetitive (iterative-LOOP) process patterns, that are formally defined in the previous section.
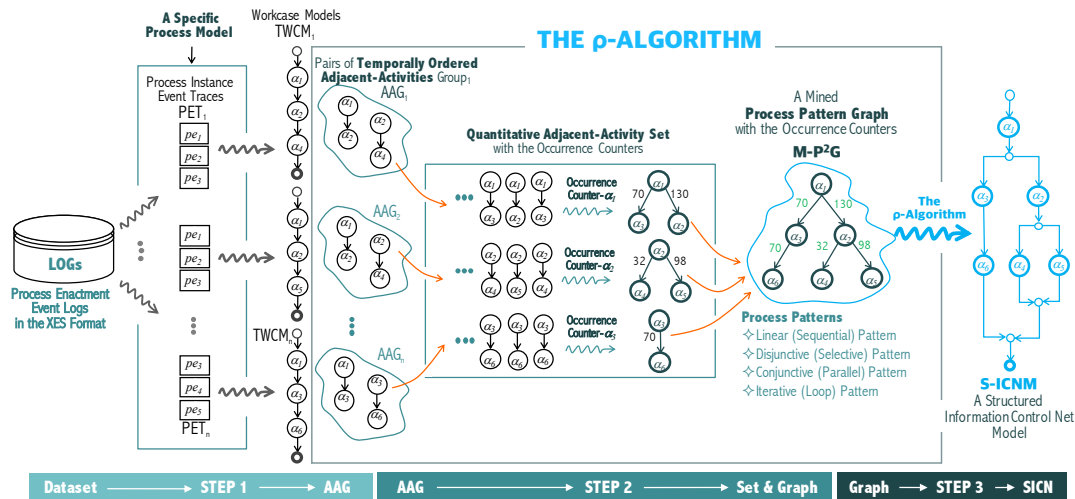

Figure 5. The ρ-Algorithm with Three Steps of Graphical Transformations

The overall algorithmic approach is a stepwise mining procedure with a series of graphical transformations to be used for rediscovering all the types of the primitive process patterns that constitute a structured information control net process model. That is, as shown in Figure 5, the ρ-Algorithm consists of three major transformation steps, such as STEP-1, STEP-2 and STEP-3, from forming temporal workcases out of the process enactment event logs to discovering a structured information control net process model as a result. The first transformation of the algorithm is to discover the enacted workcases from the event logs, each of which can be modeled into a temporal workcase model. At the same time, it is necessary to count the occurrence of each temporal workcase with its activities. The second transformation of the algorithm is to discover an activity-driven pattern graph by integrating all the members of the adjacent-activity set and calculating the occurrences of the temporal workcases. In terms of rediscovering the structured information control net process model from the corresponding activity-driven pattern graph, we develop the third transformation of the algorithm that is able to transform to an any combinational number of AND/OR/LOOP primitive process patterns.

- *STEP-1: Groups of Temporally Ordered Adjacent-Activity Pairs*: The first step of the ρ-Algorithm is to mine a group of temporally ordered adjacent-activities pairs from temporal workcases of the process instance event logs. Also, each of the temporal workcases is formally represented by one of the workcase model types introduced in the conceptual framework. That is, a temporal workcase represents an ordered enactment sequence of activity event logs, each of which is formed with its activity identifier and its time-stamp extracted from its corresponding process enactment event log.
- *STEP-2: Quantitative Adjacent-Activity Set and Process Pattern Graph*: The STEP-2 of the ρ-Algorithm is to build all the groups of temporally ordered adjacent-activity pairs, each of which corresponds to a process instance event trace. The eventual output of this algorithm is a quantitative adjacent-activity set named as adjacencyList β. This set is built from all the groups of temporally ordered adjacent-activity pairs through an internal transformation procedure.
- *STEP-3: Rediscovering Structured Process Patterns*: The final step (STEP-3) of the ρ-Algorithm is to discover all the primitive process patterns of a structured information control net process model from

the mined process pattern graph discovered from all the groups of temporally ordered adjacent-activity pairs. The eventual goal of the ρ-Algorithm is accomplished through this step. Note that the structured information control net process model must be satisfied with the proper nesting as well as the matched pairing properties in forming gateway activities in each primitive process graph pattern.

Based upon the ρ-Algorithm, the authors' research group has successfully developed a process mining system that is able to rediscover a structured information control net process model coping with all the four types of process patterns from a dataset of process enactment event logs. By using the process mining system, this paper carries out a series of experimental analyses by deploying the implemented system onto a dataset provided by the BPI challenges of 4TU.Centre for Research Data [17] and formatted in the IEEE XES standardized format [30]. Summarily, the following are the characteristics of the ρ-Algorithm and the process mining framework:

— First, the process mining framework is able not only to rediscover the process patterns but also to discover the enactment proportions [27] of the process patterns from a dataset of the IEEE XES-formatted enactment event logs of a corresponding process model.
— Second, the process mining framework is theoretically supported by the information control nets modeling methodology [16] of process models.
— Third, the essential algorithm of the process mining approach is named as ρ-Algorithm (rho-Algorithm) that is able to rediscover a structured information control net model with the enactment occurrences of the activities associated with an underlying process model.
— Fourth, the ρ-Algorithm is firstly developed in the process management and mining literature as the process mining algorithm that discovers a structured process model theoretically supported by the structured information control net modeling methodology.
— Fifth, the ρ-Algorithm is able to rediscover all the primitive process patterns such as linear (sequential), conjunctive (parallel-AND), disjunctive (exclusive-OR), and repetitive (iterative-LOOP) process patterns and discover the enactment occurrences and proportions of each branch of the process patterns.

5. PROCESS ANALYZING FUNCTIONALITY

Once a SICN process model is rediscovered from a dataset, then it is analyzed by the structural analyzer and the visual simulator implemented in the paper after being transformed into its XPDL process model. Therefore, this subsection describes the details of the algorithms of the structural analyzer and the visual simulator. The structural analyzer is based on the standardized formats of XPDL and extracts the structural attributes of the rediscovered SICN process model, which are defined in the first subsection. The next consecutive subsections describe the architectural and functional components of the structural analyzer and the visual simulator as well.

5.1. Architectural Components of the Process Analyzer

This subsection describes the detailed design specifications of the process analyzer with supporting structural and behavioral analysis in a fashion of visual simulation. As already illustrated in Figure 2, the functional architecture of the process analyzer is composed of two groups of architectural components. One is the XPDL process structural analyzer and its related components, the other is the XPDL process performative simulator and its related components. Actually, there are five functional components with three database connection agents: model and analytics visualizer, XPDL process structural analyzer, XPDL process performative simulator, structural analytics report generator, and performative analytics report generator. The database schema for the process analyzer consist of rediscovered XPDL process models database schema, structural database schema and performative database schema. As we know that the rediscovered process model database schema is for storing XPDL process models rediscovered and transformed by the process mining functionality described in the previous subsection. The structural database schema and the performative database schema preserve the structural analysis and the behavioral analysis results, respectively, including the statistical data of the structural attributes and the behavioral attributes of those XPDL-based process models.

The *rediscovered process models and analytics visualizer* is in charge of the overall control of the analyzer including user management, session & access control, and XPDL process management, in particular. Assume

---

Algorithm 1 The XPDL process Model Generation Algorithm

Require: An activity graph of a rediscovered SICN process model, Γ;
Ensure: The XPDL process model, γ of Γ;
  1: Initialize $\gamma \leftarrow \emptyset$;
  2: $\gamma.InsertHeader()$; ▷ Insert the header format of XPDL ver 2.1: <xpdl:Pakage...>, <xpdl:PackageHeader>
  3: γ.InsertTagAttribute("xpdl:Processes");   ▷ Insert the processes attribute to γ
  4: γ.InsertTagAttribute("xpdl:Activities");   ▷ Insert the activities attribute to γ
  5: **for** ($\forall$ node η in Γ) **do** ▷ Read all the nodes (actvities or gateways) in Γ and then add all of their tags to γ
  6:      γ.InsertTagAttribute("xpdl:Activity");   ▷ Insert the activity attribute to γ
  7:      currentNode $\leftarrow \eta$.GetCurrrentNodeInfo();
  8:      routeType $\leftarrow$ defaultRouteType; shapeType $\leftarrow$ defaultShapeType; gateType $\leftarrow \emptyset$;
  9:      **if** (currentNode is OrGateway) **then**
10:          routeType $\leftarrow$ "Exclusive-OR";
11:          shapeType $\leftarrow$ "Exclusive-OR Gateway";
12:      **else if** (currentNode is AndGateway) **then**
13:          routeType $\leftarrow$ "Parallel-AND";
14:          shapeType $\leftarrow$ "Parallel-AND Gateway";
15:      **else if** (currentNode is LoopGateway) **then**
16:          routeType $\leftarrow$ "Iterative-LOOP";
17:          shapeType $\leftarrow$ "Iterative-LOOP Gateway";
18:      **end if**
19:      **if** (currentNode is OpenGateway) **then**
20:          gateType $\leftarrow$ "Split";
21:      **else if** (currentNode is ClosedGateway) **then**
22:          routeType $\leftarrow$ "Join";
23:      **end if**
24:      γ.InsertAttribute(type=activity, currentNode, routeType, shapeType, gateType);
25: **end for**
26: γ.InsertTagAttribute("xpdl:Transitions");   ▷ Insert the transition attribute to γ
27: **for** ($\forall$ edge θ in Γ) **do**
28:      currentEdge $\leftarrow \theta$.GetEdgeInfo();
29:      fromActivityID $\leftarrow$ currentEdge.GetFromID();
30:      toActivityID $\leftarrow$ currentEdge.GetToID();
31:      γ.InsertAttribute(type=transition, fromActivityID, toActivityID);
32: **end for**
33: **Return** γ;   ▷ Finally output the XPDL process model

---

that the rediscovered process model is based upon the structured information control net methodology, and the process mining subsystem supports the *Export-to-XPDL* function that is automatically translating the graphical representation of a rediscovered SICN process model into a textual representation of its corresponding XPDL process model. Since then, all the functions and operations, such as verifying, analyzing, reporting and visualizing operations, to be applied into the XPDL process models are controlled and managed *via* this visualizer. The *structural and performative report generators* are to generate and visualize the structural analysis and performative simulation results and statistics of each structural component in an XPDL process model. This generator is able to produce two-level structural and performative simulation results and statistics. One is the process-level analytical results and statistics, and the other is the package-level analytical results and statistics. For the sake of providing the complete version of process analyzing functionality, the *XPDL process structural analyzer*, *XPDL process animator* including the *XPDL process instance tracer* perform the following verifiable analysis functions:

— Structural properness verification: Checking up whether the rediscovered process model is keeping the rules of proper-nestings and matched pairings in building its gateway-type activities.
— Associative relationship verification: Checking up whether the rediscovered process model is keeping the correct associative attributes in building activity-to-role associations, activity-to-program associations, activity-to-data associations, and role-to-actor associations.
— Performative properness verification: Checking up whether the rediscovered process model is keeping the efficiency in managing the dynamic aspects at the package level, process level, activity level and the performer level performative indicators.

346   — Behavioral simulation verification: Based upon the rediscovered process model with their enactment event
347   histories, checking up the enactment histories by tracing a process instance of the corresponding model *via*
348   step-by-step clicking operations.

5.2. Generation of the Rediscovered XPDL Process Models

350   The initiation step of the process analyzing approach is to perform a transformation function that generates
351 the rediscovered XPDL process model from the graphical form of the rediscovered SICN process model. The
352 algorithm of this transformation step is represented in Algorithm 1 named as the XPDL process model generation
353 algorithm. Note that the tagged constructs of XPDL are described in the next section of the process analyzing
354 functionality. As you can see in the Algorithm 1, it requires an activity graph of the rediscovered SICN process
355 model, which is the internal graph model of the ρ-Algorithm, as input (Γ), and ensures the XPDL process model
356 corresponding to the rediscovered SICN process model, as output (γ), finally. The XPDL process model plays
357 very important role as a bridge between the process mining functionality and the process analyzing functionality.
358 Therefore, the focal point of the functional integration of the process mining and the process analyzing ought
359 to be this algorithm and its output, the XPDL process model, because it is possibly realized to complete the
360 seamless integration through this algorithm and its output.

5.3. The Structural Analysis of Rediscovered XPDL Process Models

362   This subsection describes the structural attributes used in rediscovered XPDL process models, the
363 specifications of which was released by the international standardization organization of workflow management
364 coalition. The XPDL version 1.0 is the XML-formatted version of the workflow process definition language
365 (WPDL). Recently, the workflow management coalition had released the new specifications of XPDL reflecting
366 the OMG's standardized graphical notation of the process modeling notation (BPMN) as the XPDL version 2.0.
367 In this paper, we consider the XPDL version 2.1 as the textual format of a SICN process model, and it will be the
368 input of the structural analyzer. In particular, the control-flow structural attributes can be formed by the From
369 and To properties in the transition attribute. Conclusively, the structural attributes of the XPDL version 2.1 are
370 summarized in Table 1. Note that, because of the page limitation, we simply introduce the structural attributes
371 and their properties in this paper.

372   Based upon the structural attributes and their occurrences in a rediscovered XPDL process model (or
373 package), the XPDL process structural analyzer is able to produce the two-level analytical statistics such as the
374 package-level analytical statistics and the process-level analytical statistics. In the package-level, it supports
375 the activity-type structural analysis, component-type structural analysis, component-type's usage ratio analysis,
376 and the subprocess usage ratio analysis. It also supports the activity-related structural attributes analysis and
377 the associative relationship attributes analysis such as activity-to-program, activity-to-role, activity-to-data, and
378 role-to-performer associations. Note that a process package comprises a group of process models, and the XPDL
379 schema is formatted from a pair of the package-level tags like `<package> ... </package>`. The following
380 are the analytical statistics to be analyzed and produced by the structural analyzer, which ought to be extensibly
381 represented in the rediscovered XPDL process models through their corresponding extended tags:

382   (1) Package-level Structural Analysis Statistics

383   — The number of process models in a corresponding process package
384   — The number of activities in each model and their usage ratios
385   — The number of roles in each model and their involvement ratios
386   — The number of invoked applications in each model and their usage ratios
387   — The number of relevant data types in each model and their usage ratios
388   — The number of subprocesses in each model and their usage ratios
389   — The usage ratio of each model as subprocesses

390   (2) Process-level Structural Analysis Statistics

391   — Species of structural patterns and their usage ratios
392   — The number of participants (actors or performers) and their participation ratios
393   — The number of roles and their involvement ratios
394   — The number of invoked applications and their usage ratios

Table 1. Structural Attributes in the XML Process Definition Language

Package	process	Activity	Transition	Application	Data-field	Participant
-Identifier -Name -Descrip. -Ex_Attr.	-Identifier -Name -Descrip. -Ex_Attr.	-Identifier -Name -Descrip. -Ex_Attr.	-Identifier -Name -Descrip. -Ex_Attr.	-Identifier -Name -Descrip. -Ex_Attr.	-Identifier -Name -Descrip. -Ex_Attr.	-Identifier -Name -Descrip. -Ex_Attr.
-XPDL -Source -Creation -Version -Author -Country -Publ_St -Conform -Priority	-Cre_Date -Version -Author -Code_P -Country -Publ_St -Priority -Limit. -Val_Fr -Val_To	-A_Mode -Split -Join -Priority -Limit. -St_Mode -Fi_Mode -Deadline			-Data_Ty	-Par_Ty
-Respon. -Ext_Pac	-Params. -Respon.	-Perf. -Tool -Subflow -Act_Set -A_param	-Conditi. -From_Tr -To_Tr	-Params.	-Ini_Val.	
-Docum. -Icon	-Docum. -Icon	-Docum. -Icon				
-Cost_Unt	-Dur_Unt -Duration -Wait_T -Work_T	-Cost -Duration -Wait_T -Work_T				

395   — The number of relevant data types and their usage ratios
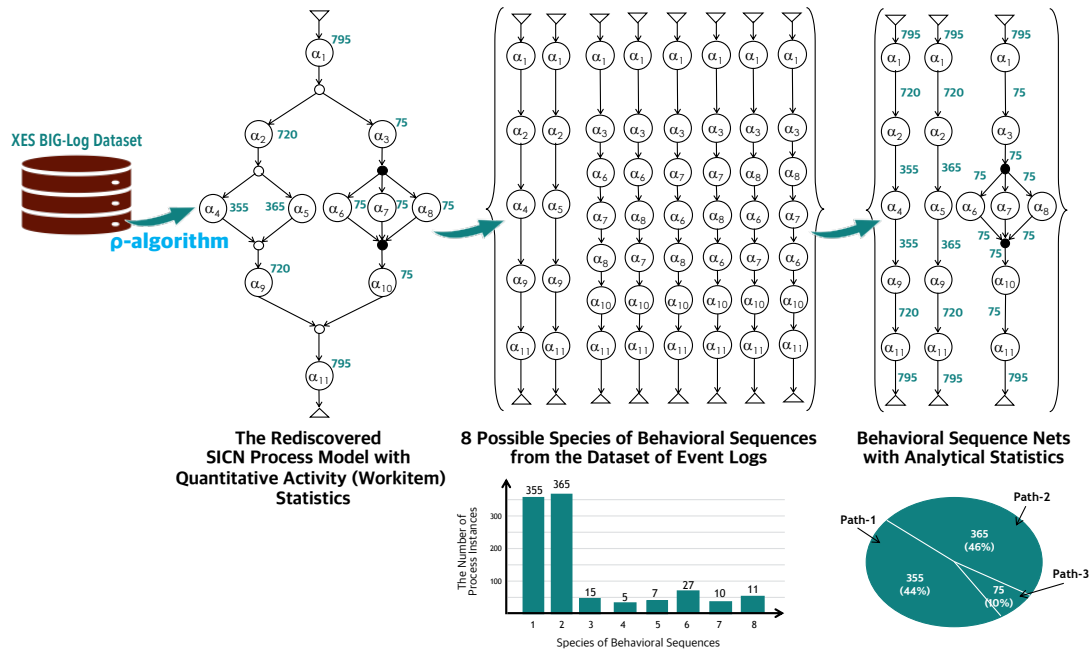396   — The number of subprocesses and their usage ratios

Figure 6. The Behavioral Analysis and Statistics of the Rediscovered Process Model

---

Algorithm 2 The Behavioral Sequence Analysis Algorithm

Require: A rediscovered SICN process model, Γ;
Ensure: A Set of Behavioral Sequence Nets, Ω of Γ;
1:   initialize $\Omega \leftarrow \emptyset$;
2:   **Procedure BSAFunc (in** s $\leftarrow \{\alpha_I\}, \Omega$);   ▷ Recursive procedure for discovering Ω
3:   **begin**
4:   $v \leftarrow s; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{v\}$;
5:   **while** ($u \leftarrow \delta_o(s) \,!= \alpha_F$) **do**
6:      **if** (u is a sequential activity?) **then**
7:         $w \leftarrow u; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{w\}$;
8:         $\Omega.\rho_o(v) \leftarrow w; \Omega.\rho_i(w) \leftarrow v; \Omega.\beta_o(v) \leftarrow \kappa_o(s); \Omega.\beta_i(v) \leftarrow \kappa_i(s)$;
9:         $\phi_o((v, o \in \rho_o(v))) \leftarrow$ occurrences of $o \in \rho_o(v)$;
10:       $\phi_i((\varepsilon \in \rho_i(v), v)) \leftarrow$ occurrences of v;
11:     **else if** (u is a conjunctive AND-split-gateway activity?) **then**
12:       $w \leftarrow u; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{w\}$;
13:       $\Omega.\rho_o(v) \leftarrow w; \Omega.\rho_i(w) \leftarrow v; \Omega.\beta_o(v) \leftarrow \kappa_o(s); \Omega.\beta_i(v) \leftarrow \kappa_i(s)$;
14:       **for** (each of $\forall \alpha \in \delta_o(u)$) **do**
15:          $x \leftarrow \alpha; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{x\}$;
16:          $\Omega.\rho_o(w) \leftarrow w; \Omega.\rho_i(x) \leftarrow w$;
17:          $\Omega.\beta_o(w) \leftarrow \kappa_o(u); \Omega.\beta_i(w) \leftarrow \kappa_i(u)$;
18:          $\phi_o((w, o \in \rho_o(w))) \leftarrow$ occurrences of $o \in \rho_o(w)$;
19:          $\phi_i((\varepsilon \in \rho_i(w), w)) \leftarrow$ occurrences of w;
20:       **end for**
21:       **for** (each of $\forall \alpha \in \delta_o(u)$) **do**
22:          **call Procedure BSAFunc (in** $s \leftarrow \alpha, \Omega$);
23:       **end for**
24:     **else if** (u is a disjunctive OR-split-gateway activity?) **then**
25:       $w \leftarrow u; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{w\}$;
26:       $\Omega.\rho_o(v) \leftarrow w; \Omega.\rho_i(w) \leftarrow v; \Omega.\beta_o(v) \leftarrow \kappa_o(s); \Omega.\beta_i(v) \leftarrow \kappa_i(s)$;
27:       **for** (each of $\forall \alpha \in \delta_o(u)$) **do**
28:          **call Procedure BSAFunc (in** $s \leftarrow \alpha, \Omega$);
29:       **end for**
30:     **else if** (u is either an OR-join-gateway or an AND-join-gateway activity?) **then**
31:       $w \leftarrow u; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{w\}$;
32:       $\Omega.\rho_o(v) \leftarrow w; \Omega.\rho_i(w) \leftarrow v; \Omega.\beta_o(v) \leftarrow \kappa_o(s); \Omega.\beta_i(v) \leftarrow \kappa_i(s)$;
33:       $\phi_o((w, o \in \rho_o(w))) \leftarrow$ occurrences of $o \in \rho_o(w)$;
34:       $\phi_i((\varepsilon \in \rho_i(w), w)) \leftarrow$ occurrences of w;
35:     **end if**
36:     $s \leftarrow u; v \leftarrow w$;
37:   **end while**
38:   $w \leftarrow u; \Omega.A^{bs} \leftarrow \Omega.A^{bs} \cup \{w\}$;   ▷ where u is the final activity, α_F
39:   $\Omega.\rho_o(v) \leftarrow w; \Omega.\rho_i(w) \leftarrow v; \Omega.\beta_o(v) \leftarrow \kappa_o(s); \Omega.\beta_i(v) \leftarrow \kappa_i(s)$;
40:   $\phi_o((w, o \in \rho_o(w))) \leftarrow$ occurrences of $o \in \rho_o(w)$;
41:   $\phi_i((\varepsilon \in \rho_i(w), w)) \leftarrow$ occurrences of w;
42:   **Return** Ω;   ▷ Finally output a set of behavioral sequence nets

---

5.4. The Behavioral Analysis of Rediscovered Process Models

The behavioral analysis of a rediscovered XPDL process model is based upon the primitive control-flow process patterns introduced in [Definition 1] in the previous subsection. The control-flow attributes of the rediscovered process model are also composed by the combinational permutation of the four types of the structural transition patterns, such as sequential, disjunctive, conjunctive and iterative transition patterns. Therefore, the behaviors (*i.e.* execution sequences of process instances) of the rediscovered process model is analyzed and measured by tracing these combinational primitive process patterns. Figure 6 illustrates how the behavioral analysis and statistical measurement works on the rediscovered process model. The ρ-Algorithm rediscovers a SICN process model from a dataset of event logs. Assume that the rediscovered SICN process model is made up of two disjunctive process patterns and a single conjunctive process pattern as shown in the figure. The algorithm discovers not only the SICN process model but also the execution occurrences of its process instances and workitems, in addition. From the rediscovered SICN process model and its execution statistics, the process analyzing function generates the corresponding XPDL process model with extended tags of the statistical attributes and analyzes the behavioral sequences and quantitative statistical measurement, as well. [Definition 2]
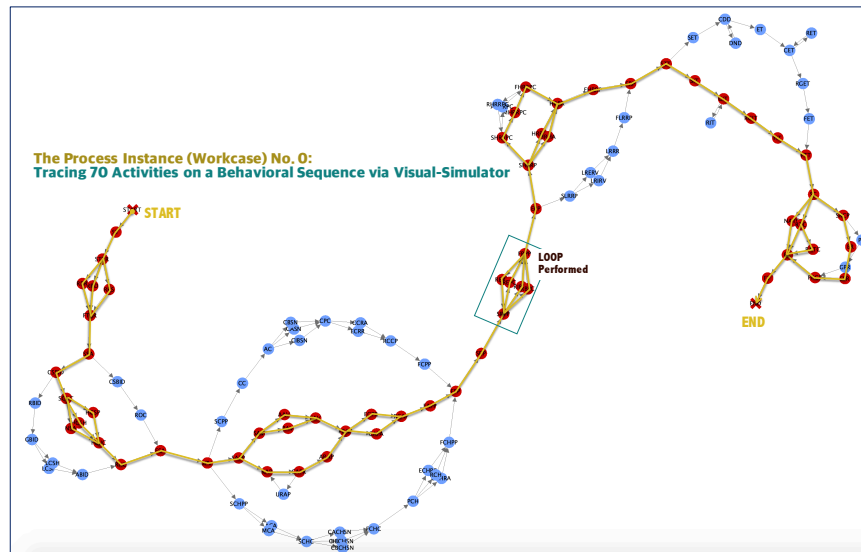
Figure 7. The Interactive and Visual Analysis of Behavioral Sequences of the Rediscovered Process Model

411 defines the behavioral sequence net that is the formal representation of a behavioral sequence and its execution
412 statistics (*i.e.* the number of occurrences). Also, the **Algorithm 2** named as the behavioral sequence analysis
413 algorithm is for generating a set of behavioral sequence nets from a rediscovered SICN process model. Note that
414 the algorithm deals with only three process patterns, such as sequential, conjunctive-AND and disjunctive-OR
415 process patterns, except the iterative-LOOP process pattern, because the iterative-LOOP process pattern is treated
416 as a special type of disjunctive-OR process pattern that ought to be properly dealt by the disjunctive-OR routines
417 of the algorithm. Practically, the XPDL standards do not specify the iterative-LOOP transition type as a normal
418 process pattern, either.

419 **[Definition 2] Behavioral Sequence Net** of a rediscovered SICN (or XPDL) process model model. Let Ω
420 be a **BSN**, a behavioral sequence net, that is formally defined as $\Omega = (\rho, \kappa, o)$ over the discovered workitems,
421 $\mathbf{A}^{bs}$, and the discovered control-transitions, $\mathbf{T}^{bs}$, where

  − $\rho = \rho_i \cup \rho_o$ where, $\rho_o : \mathbf{A}^{bs} \longrightarrow \wp(\mathbf{A}^{bs})$ is a multi-valued mapping of a workitem to its immediate
    successor, and $\rho_i : \mathbf{A}^{bs} \longrightarrow \wp(\mathbf{A}^{bs})$ is a multi-valued mapping of a workitem to its immediate predecessor;
  − $\beta = \beta_i \cup \beta_o$ where, $\beta_i : \mathbf{A}^{bs} \longrightarrow \mathbf{T}^{bs}$ is a single-valued mapping of a workitem to its connected
    control-transition, $\tau, (\nu \in \rho_i(\alpha), \alpha) \in \mathbf{T}^{bs}$; and $\beta_o : \mathbf{A}^{bs} \longrightarrow \mathbf{T}^{bs}$ is a single-valued mapping of a workitem
    to its connected control-transition, $\tau, (\alpha, \nu \in \rho_o(\alpha)) \in \mathbf{T}^{bs}$, where $\alpha \in \mathbf{A}^{bs}$;
  − $o = o_i \cup o_o$ where, $o_i : \mathbf{T}^{bs} \longrightarrow \mathbf{N}$ is a single-valued mapping of an incoming control-transition ($\tau \in \beta_i$) to
    its number of occurrences; and $o_o : \mathbf{T}^{bs} \longrightarrow \mathbf{N}$ is a single-valued mapping of an outgoing control-transition
    ($\tau \in \beta_o$) to its number of occurrences;

430 The behavioral analysis functionality illustrated in Figure 6 and devised in Algorithm 2 is able to produce a
431 set of behavioral sequence nets from the rediscovered SICN/XPDL process model. Our emphasis is placed on
432 the quality analysis of rediscovered process model by using the proposed concept and its algorithm. That is, the
433 behavioral analysis aims at improving the quality of the rediscovered process model. And so naturally it is raised
434 for the following questions: what is the quality of a process model? And how can we evaluate the quality of a
435 process model? This paper tries to answer for those questions through the concept of the behavioral sequence
436 nets supported by the functional integration of the process mining and the process analyzing functionalities. We
437 would insist that the quality of a rediscovered process model be defined by the degree of the discrepancy between
438 the discovered process model as it is built by the Buildtime subsystem and the rediscovered process model as it is
439 rediscovered by the Miningtime subsystem. The discrepancy, as you can easily imagine, is caused by disjunctive
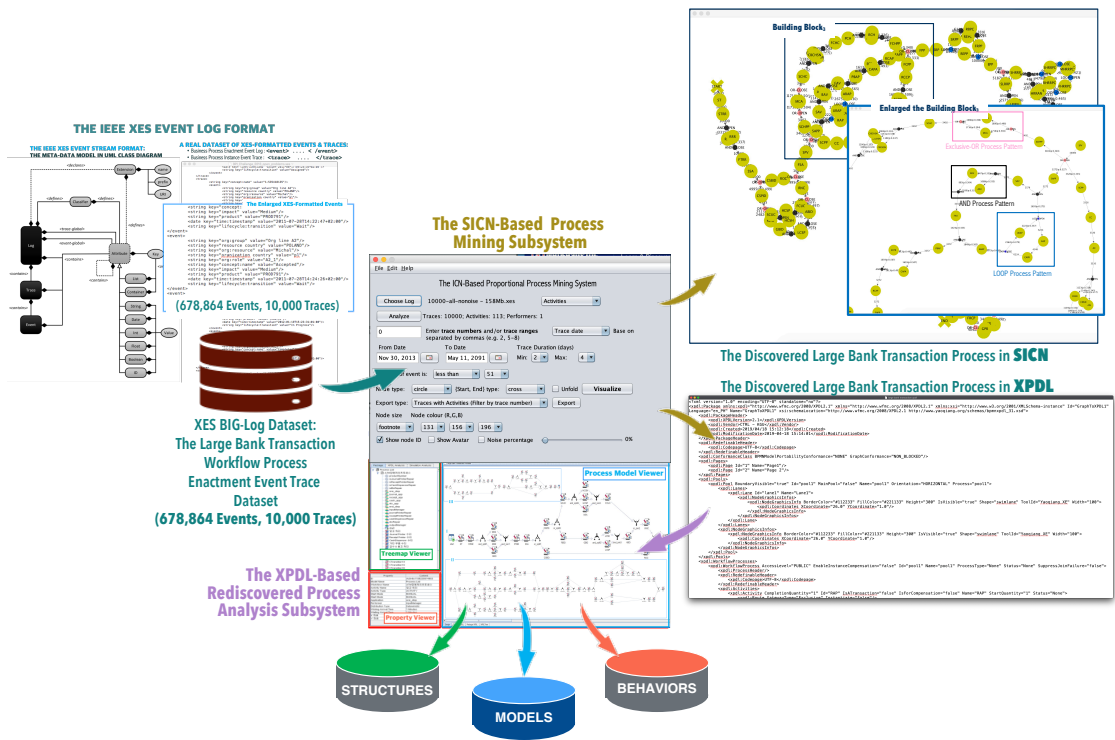
Figure 8. The Implemented Systematic Framework of the Integrated Functional Architecture

and conjunctive process patterns presented on the corresponding process model. The number of disjunctive process patterns on the model will effect on the number of *mutually exclusive behavioral sequences*, whereas the number of the conjunctive process patterns on the model will effect on the number of *mutually inclusive behavioral sequences*. Additionally, the statistical measurements related to the behavioral sequences play very valuable role in evaluating the quality of the rediscovered process models.

Especially, each of the behavioral sequences can be traced through the interactive and visual operations provided by the process analyzer. Figure 7 depicts a situational snapshot of performing the interactive and visual analysis through the behavioral sequence tracer. The graph model of the behavioral sequence net rediscovered from the dataset of the large bank transaction process enactment event logs. Note that the details of the dataset and its rediscovered SICN process model is described in the experimental section of this paper. As shown in the figure, the behavioral sequence net corresponding to the process instance No. 0 includes 65 workitems (business activities) with their temporal enactment sequence. Through the interactive and visual analysis on those process instances rediscovered from datasets, we are able to evaluate the properness and quality of the process instances and their models, eventually. The implementation details of the interactive and visual analysis functionality for analyzing the behavioral sequences and its practical applications are described in the experimental studies of the next section.

5.5. Implementation of the Integrated Functional Architecture

This section describes the implementation details of the integrated functional architecture designed in the previous section. The major systematic framework is made up of a structural analyzer and a behavioral simulator with a process instance tracer, and these are operable on those rediscovered SICN and XPDL process models from the process enactment event log datasets by the process mining approach developed in this paper. Figure 8 illustrates the concrete systematic framework of the functional integration of the process mining and the process analysis and its experimental results with a specific dataset. As shown in the figure, it is realized for the process mining functionality to be integrated with the process analyzing functionality through seamlessly sharing the XPDL process models transformed from the rediscovered SICN process models. The figure uses a series of captured screens produced through one of the experiments described in the next section. The dataset is in the

IEEE XES standardized log format and contains 678,864 event logs recorded from enacting 10,000 workcases (*i.e.* process instances) of the large bank transaction process model; from the dataset, the process mining subsystem rediscovers the structured information control net process model consisting of 113 activities and their occurrences, and the process analyzing subsystem produces the textual form of the rediscovered XPDL process model; by analyzing the rediscovered XPDL process model, the process analyzing subsystem eventually validates the structural properness as well as the behavioral properness of the rediscovered process model characterized with the very large scale and massively parallel structural properties.

6. EXPERIMENTAL VALIDATION STUDIES

This section carries out a couple of feasibility and validation studies of the proposed conceptual architecture: functional integration with the process mining and the process analyzing. One is for the structural analysis That is, as the implemented conceptual architecture, the systematic framework is implemented and applied to a couple of real datasets obtained from the 4TU.Centre for Research Data[17], each of which is formatted in the IEEE-XES standardized event log format and contains very large scale event logs spawned from enactment histories of very large scale and massively parallel process models. Figure 9 is the captured screens of the datasets and the following are the brief descriptions of the datasets used for the experimental validation studies:
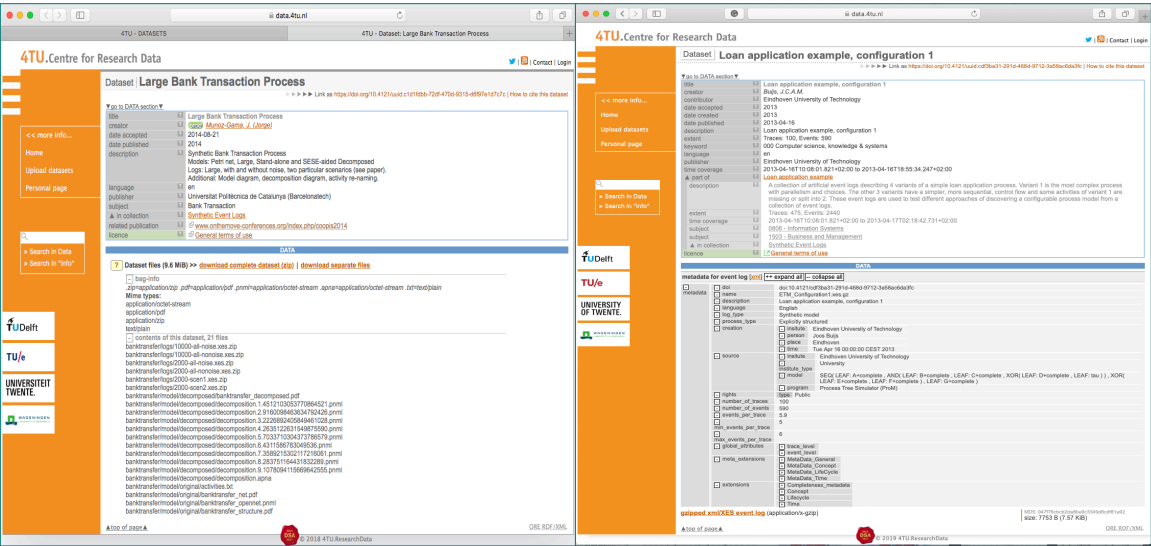
Figure 9. The Detailed Descriptions of the Datasets on the 4TU.Centre for Research Data

— Experimental Dataset-1: Mining and analyzing a structured information control net process model from the enactment event log dataset of the Large Bank Transaction Process Model:

· Title: Large Bank Transaction Process
· Creator: Munoz-Gama. J. (Jorge)
· Date accepted and published: 2014-08-21
· Description: Synthetic Bank Transaction Process

∗ Models: Petri net, Large, Stand-alone and SESE-aided Decomposed
∗ Logs: Large, with and without noise, two particular scenarios
∗ Additional: model diagram, decomposition diagram, activity re-naming.

— Experimental Dataset-2 – Mining and analyzing a structured information control net process model from the enactment event log dataset of the Loan Application Example (Configuration 1) Process Model:

· Title: Loan Application Example, Configuration 1
· Creator: Buijs, J.X.A.M.
· Date accepted and published: 2013-04-16

495 · Description: A collection of artificial event logs describing 4 variants of a simple loan application
496 process. Variant 1 is the most complex process with parallelism and choices. The other 3 variants
497 have a simpler, more sequential, control flow and some activities of variant 1 are missing or split into
498 2. These event logs are used to test different approaches of discovering a configurable process model
499 from a collection of event logs.

6.1. Validation of the Rediscovered SICN Process Models

501 First of all, it is necessary to validate the rediscovered process models of structured information control nets
502 (SICN), which are mined from the experimental datasets, respectively. The process mining subsystem of the
503 integrated functional architecture and its implemented system is applied to the experimental dataset-1 as well as
504 the experimental dataset-2 and successfully mined the rediscovered SICN process models.
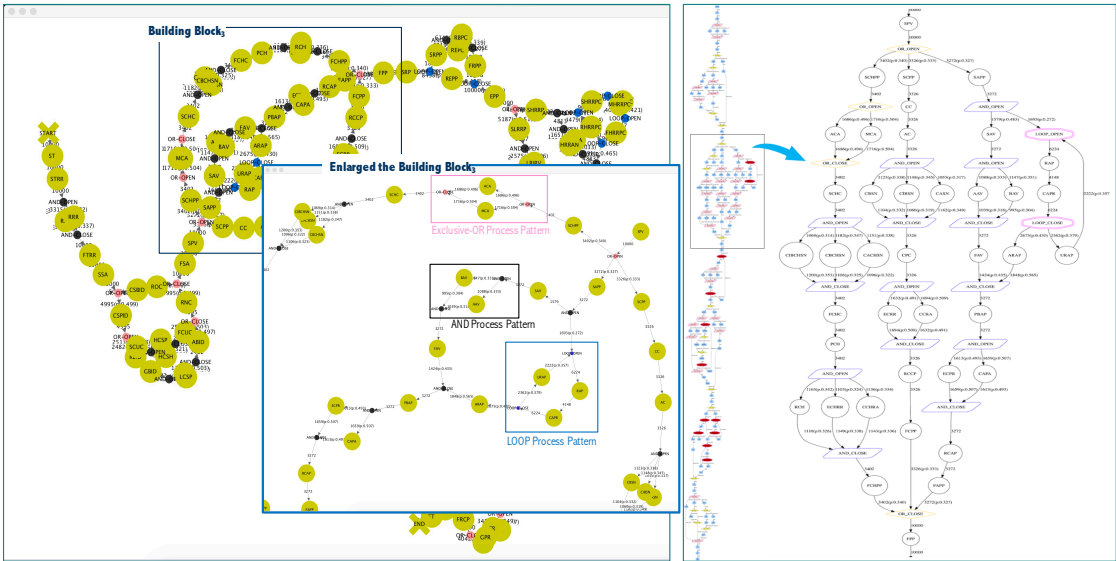
Figure 10. The Rediscovered SICN Process Model from the Experimental Dataset-1
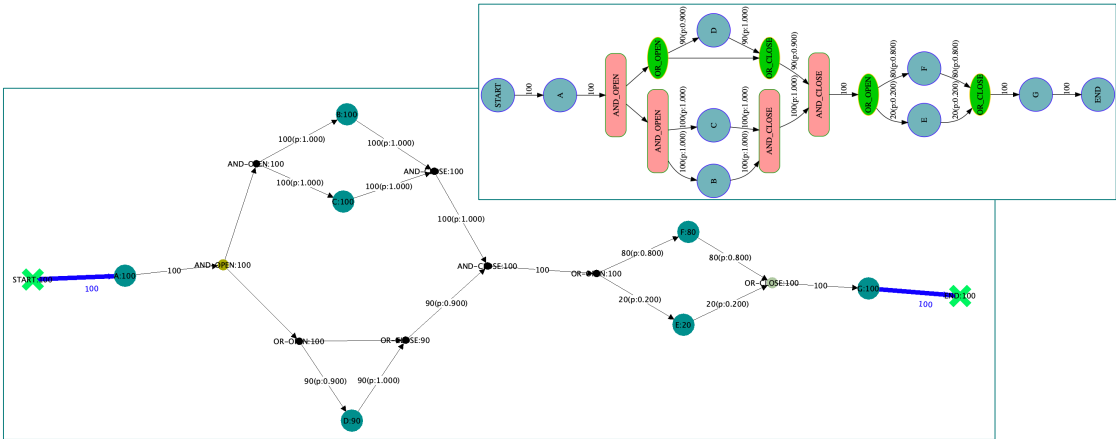
Figure 11. The Rediscovered SICN Process Model with Activities' Occurrences from the Experimental Dataset-2

505 As you can see in Figure 10, the experimental result of the process mining subsystem is the rediscovered
506 SICN process model containing 113 activities with all the primitive process patterns and all the activities'
507 execution occurrences. That is, the lefthand side of the figure is a captured screen representing the SICN process
508 model rediscovered from the experimental dataset-1 of 10000-all-nonoise-150MB.xes that contains 10,000
509 process instance event traces and 678,864 activity event logs under the control of enacting 113 business activities

510 of the large bank transaction process model. The enlarged process block indicates all the primitive process
511 patterns by the labeled boxes, in particular. And moreover, the process mining subsystem supports the printout
512 form of the rediscovered SICN process model as shown in the righthand side of the figure. Secondly, another
513 experiment of the process mining subsystem was carried out on the experimental dataset-2 and it was able to
514 mine the rediscovered SICN process model of the Loan Application Example (Configuration 1) as shown in
515 Figure 11. The captured-screen and its printout form in the figure show the rediscovered SICN process model
516 containing 9 activities with their execution occurrences. The experimental dataset-2 is built under the control of
517 9 business activities of the loan application example process model with enacting 100 process instance event
518 traces and storing 590 event logs (the average number of events per trace is 5.9 event logs).

519   Through these SICN process model rediscovery experiments, the process mining functionality of the
520 integrated functional architecture and its implemented system ought to be functionally validated as you have
521 seen. In other words, the system is able to rediscover all the structural control-flow primitive patterns, such as
522 conjunctive process pattern of AND-split and AND-join, disjunctive process pattern of OR-split and OR-join, and
523 iterative process patterns of LOOP-split and LOOP-join, from both of the experimental dataset-1 and dataset-2.
524 Also, it is able to mine the number of execution occurrences on each of the 113 business activities from the
525 experimental dataset-1, and the number of execution occurrences on each of the 7 business activities from the
526 experimental dataset-2, as well.

Figure 12. The Quantitative Analysis of Structural Constructs and Business Activities in the Rediscovered SICN Process Model from the Experimental Dataset-1

6.2. Experimental Validation I: The Process Mining with Structural Analysis
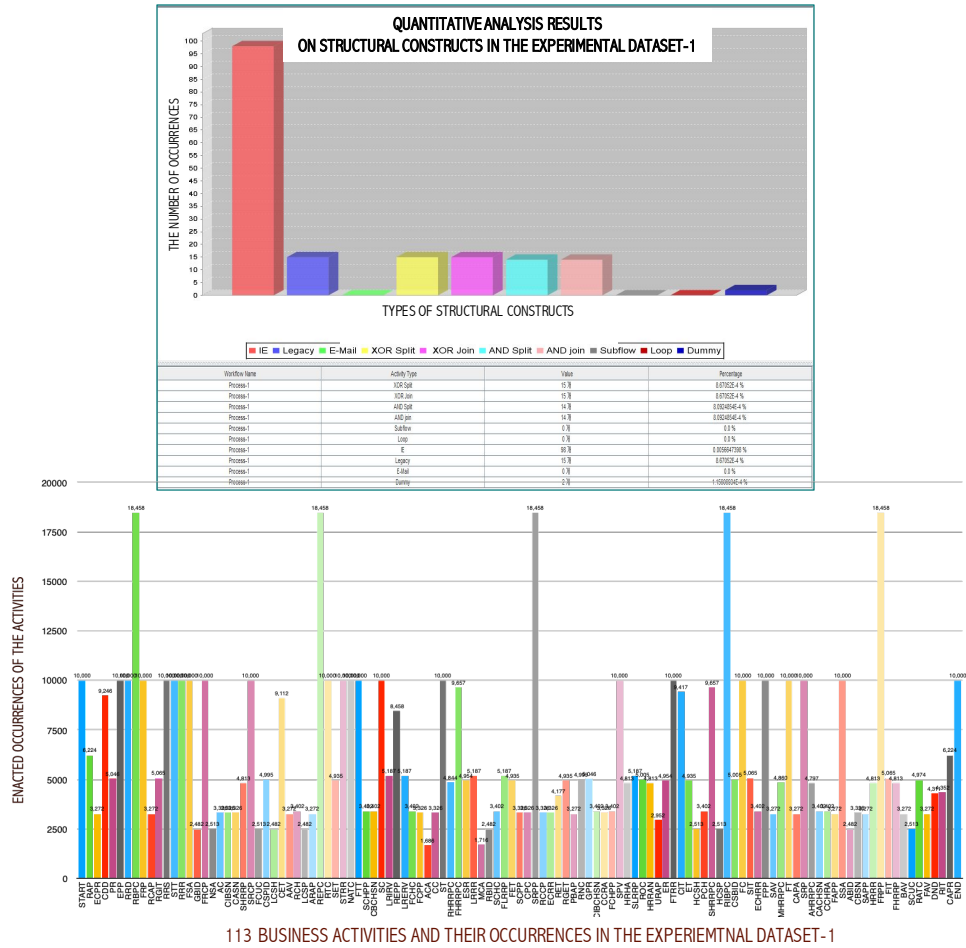
This section carries out an experimental validation study on the experimental dataset-1 of the large bank transaction process. Through this experimental validation study, it is proved that the implemented process mining subsystem based upon the ρ-Algorithm is able to completely rediscover a SICN process model and to analyze the structural aspect of the model with structural analysis statistics.

As you can see, Figure 12 shows the quantitative analysis results in terms of the number of presents on each of the structural construct types, such as conjunctive process pattern of AND-split and AND-join, disjunctive process pattern of OR-split and OR-join, iterative process patterns of LOOP-split and LOOP-join, subprocess, and others, and the number of execution occurrences on each of the 113 business activities on the experimental dataset-1. Note that these bar-chart graphs are based on the structural analysis function of the process analyzing subsystem and simply visualized by the commercialized data analysis tool.
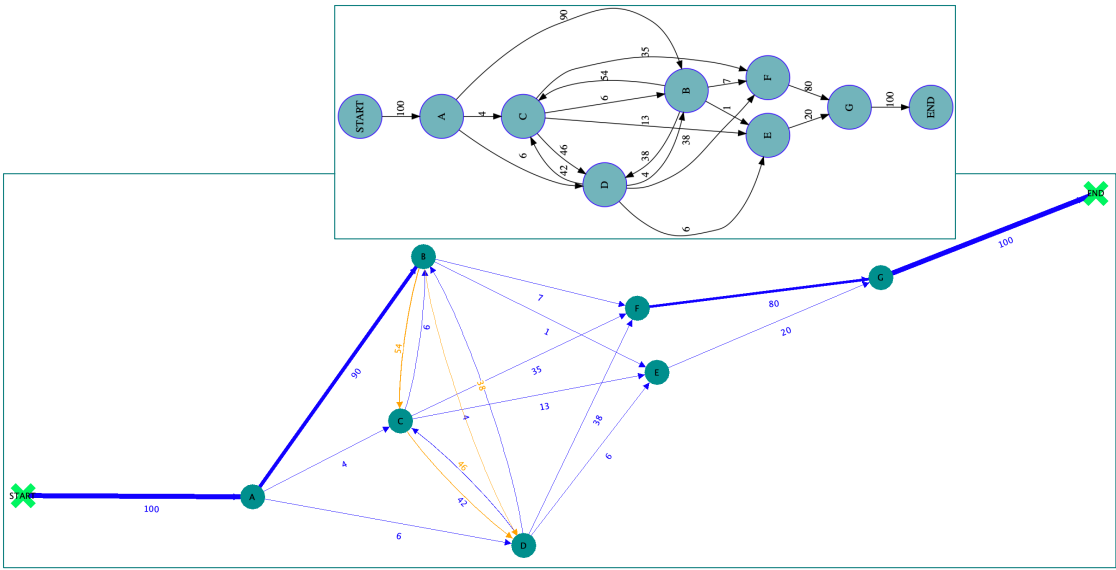
Figure 13. The Behavioral Sequence Net Rediscovered from the Experimental Dataset-2

Table 2. The Behavioral Sequence (BS) Analysis Result on the Experimental Dataset-2

BS Species ID	No. of BSs	The BS IDs in the Species	No. of Events
BS-1	1	92	8
BS-2	12	44,45,46,47,48,49,50,51,52,53,54,55	8
BS-3	26	56,57,58,59,60,61,62,63,64,65,66,67, 68,69,70,71,72,73,74,75,76,77,78,79, 80,81	8
BS-4	2	97,98	8
BS-5	4	93,94,95,96	8
BS-6	8	82,83,84,85,86,87,88,89	7
BS-7	38	6,7,8,9,10,11,12,13,14,15,16,17,18, 19,20,21,22,23,24,25,26,27,28,29,30, 31,32,33,34,35,36,37,38,39,40,41,42,43	8
BS-8	6	0,1,2,3,4,5	8
BS-9	1	99	7
BS-10	1	90	7
BS-11	1	91	8

6.3. Experimental Validation II: The Process Mining with Behavioral Analysis

This section carries out another experimental validation study with and the experimental dataset-2 of the loan application example process model for evaluating the behavioral analysis functionality devised in the previous

section. The behavioral analysis aspects of the implemented process analyzing subsystem are based upon the formal concept of the behavioral sequence nets that can be generated by the Algorithm 2 and the interactive and visual process instance tracing mechanism, as well. First, Figure 13 shows the behavioral sequence net rediscovered from the experimental dataset-2 with a captured-screen and a printout form. As you can see in the figure, the process analyzing subsystem is able to rediscover every control-transition with its occurrences. Table 2 is the summary of the behavioral sequence analysis results. As you can see in the table, there are 11 behavioral sequence species rediscovered from the dataset; the behavioral sequence having the maximum number of behavioral sequences is the BS species ID, BS-7, holding 38 behavioral sequences; the average number of events per behavioral sequence is 7.73.
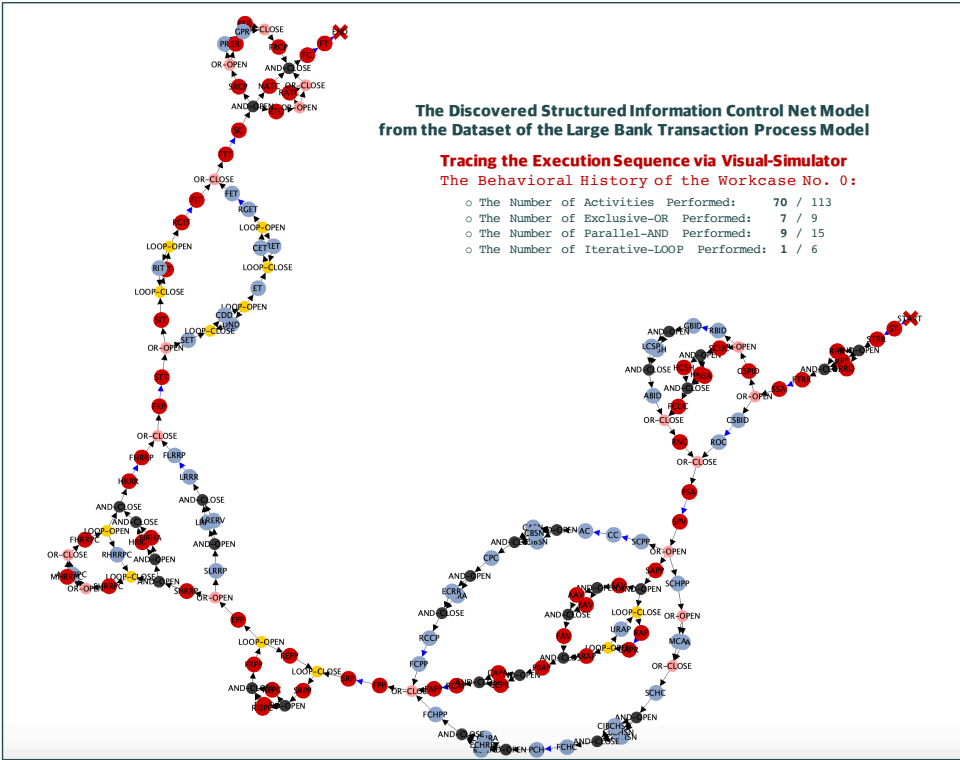
Figure 14. The Interactive and Visual Tracing of Behavioral Sequences on the Rediscovered SICN Process Model from the Experimental Dataset-1

Based upon the Algorithm 2 and the formal concept of the behavioral sequence nets, the interactive and visual tracing functionality of behavioral sequences is implemented in the process analyzing subsystem. Therefore, the second experimental validation of the behavioral analysis aspect carries out on the experimental dataset-1. Figure 14 is a captured screen of the process analyzing subsystem, on which the rediscovered SICN process model is displayed and the user is able to visually trace a behavioral sequence in an interactive fashion. In the figure, those red-colored circles represent the business activities that are already traversed during interactively running the behavioral sequence tracing session. Additionally, it supports displaying the structural statistics on the behavioral sequence traced; the behavior sequence (*i.e.* process instance ID, No. 0) involves 70 business activities out of 113 business activities; the numbers of disjunctive exclusive-OR, conjunctive parallel-AND and repetitive iterative-LOOP constructs involved are 7, 9 and 1, respectively.

7. CONCLUSION

This paper tries to build a conceptual initiation for realizing the seamless integration of the process mining and the process analyzing functionalities. Actually, this conceptual initiation came up with a couple of the experimental validation studies dealing with a very large-scale process mining challenge. As shown in Figure 15, the rediscovered SICN process model obtained from the experimental study was made up of 624 business

activities. In the experimental validation studies, it ought to be almost impossible to fulfill any further meaningful analytical work and study with the rediscovered SICN process model except the process mining work itself. This is the reason why the functional integration of the process mining and the process analyzing is issued, and this paper ought to be one of the reasonable solutions for the challenging topic in the process mining literature, even though this conceptual approach and its implemented framework won't be a complete solution for the challenging issue.

Figure 15. The Rediscovered SICN Process Model with Very Large-Scale and Massively Parallel Business Activities

This paper has proposed the integrated functional architecture for seamlessly providing the process mining functionality and the process analyzing functionality. In order to identify the structural attributes and the behavioral process patterns in a very large-scale process model, the paper devised a series of approaches that formally extract those attributes and patterns from the structured information control nets as well as from the XML process definition language process models. Finally, the paper implemented the integrated functional architecture as being composed of a process mining subsystem and process analyzing subsystem, which are named as the process mining with structural analyzer and the process mining with behavioral analyzer. Finally, the paper fulfilled a couple of experimental studies with two datasets obtained from the 4TU.Centre for Research Data and validated the feasibility of the integrated functional architecture and its implemented system. In conclusion, the issues of the process mining and analytics methodologies and systems are rapidly growing and coping with a wide diversity of application domains. So, the literature needs various, advanced, and specialized process analytical techniques and simulation methodologies being conjunction with the process mining technology, which are used for giving highly valuable feed-backs to the redesign and reengineering activities for the existing process models and packages. We strongly believe that this work might be one of those impeccable attempts and pioneering contributions for improving and advancing the process mining and analytics technology.

As a future work, the paper has a plan to expand the proposed integrated functionality with much more sophisticated features with the BPMN-related functionality with performing a series of experimental validation studies by using the expanded framework. Through the experimental validation studies, it ought to be expected to completely analyze the very large-scale and massively parallel process model, introduced in Figure 15, in terms

of automatically analyzing the structural correctness and the behavioral properness coping with the theoretical approach with SICN as well as the practical approach with XPDL and BPMN. Additionally, as a more concrete future work, it is necessary for the proposed functional architecture and system to be enhanced so as for such advanced features and functions to be safely appended as the plug-in type of software components. One of the advanced features ought to be the process-aware organizational social network discovery functionality [31] that has been issued, fulfilled, produced and implemented by the author's research group, thus far.

REFERENCES

1.   Kim, K.; Ellis, C.A. Workflow Mining: Definitions, Techniques, and Future Directions. In *Workflow Handbook*; Fischer, L., Ed.; Future Strategies, Incorporated, 2006; pp. 213–228.

2.   Wang, Y.; Zacharewicz, G.; Traore, M.K.; Chen, D. An integrative approach to simulation model discovery: Combining system theory, process mining and fuzzy logic. *Journal of Intelligent & Fuzzy Systems* **2018**, *34*, 477–490.

3.   Ellis, C.A.; Kim, K.; Rembert, A.; Wainer, J. Beyond Workflow Mining. *Lecture Notes in Computer Science* **2006**, *4102*, 49–64.

4.   Yun, J.; Jin, M.; Pham, D.; Kim, K.P. Conversion of an ICN-Based Workflow Process Model to a Data-Sequence-based Workflow Model. Proceedings of the 13th Asia Pacific International Conference on Information Science and Technology, 2018, pp. 33–35.

5.   Deelman, E.; Singh, G.; Su, M.H.; Blythe, J.; Gil, Y.; Kesselman, C.; Mehta, G.; Vahi, K.; Berriman, G.B.; Good, J.; Laity, A.; Jacob, J.C. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming* **2005**, *13*, 219–237.

6.   Liu, R.; Kumar, A. An Analysis and Taxonomy of Unstructured Workflows. *Lecture Notes in Computer Science* **2005**, *3649*, 268–284.

7.   van Hee andO. Oanea, K.; Post, R. Yasper: a tool for workflow modeling and analysis. Proceedings of the 6th International Conference on Application of Concurrency to System Design, 2006, pp. 279–282.

8.   Kim, K.P. An XPDL-Based Workflow Control-Structure and Data-Sequence Analyzer. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS* **2019**, *13*, 1702–1721.

9.   Kim, K. An Enterprise Workflow Grid/P2P Architecture for Massively Parallel and Very Large Scale Workflow Systems. *Lecture Notes in Computer Science* **2006**, *3842*, 472–476.

10.   XML Process Definition Language (XPDL) Document Number WFMC-TC-1025: Version 2.1 Document Status – Working. In *WfMC Standard Specification - XPDL*; Committee, W.T., Ed.; Workflow Management Coalition (WfMC), 2005.

11.   Business Process Model and Notation – Version 2.0. In *OMG Document Number Formal/2011-01-03*; OMG, Object Management Group, 2011.

12.   Dumas, M.; Rosa, M.L.; Mendling, J.; Reijers, H.A. *Fundamentals of Business Process Management*; Springer, 2012.

13.   Kim, K.; Ellis, C.A. σ–Algorithm: Structured Workflow Process Mining Through Amalgamating Temporal Workcases. *Lecture Notes in Computer Science* **2007**, *4426*, 119–130.

14.   Kim, K.; Ellis, C.A. *Section II / Chapter VII. An ICN-based Workflow Model and Its Advances, Handbook of Research on BP Modeling*; IGI Gloabal, ISR, 2009.

15.   Zacharewicz, G.; Frydman, C.; Giambiasi, N. G-DEVS/HLA environment for distributed simulations of workflows. *Simulation* **2008**, *84*, 197–213.

16.   Ellis, C.A. Information Control Nets: A Mathematical Model of Information Flow. Proceedings of ACM International Conference on Simulation, Modeling and Measurement of Computer Systems, 1979, pp. 225–240.

17.   BPI Challenge 2012, 2013, 2014, 2015, 2016, 2017, 2018. In *https://data.4tu.nl/repository/collection:event-logs-real*; 4TU.Centre for Research Data, 2018.

18.   de Vreede, G.J.; Verbraeck, A.; van Eijck, D. Integrating the Conceptualization and Simulation of Business Processes: A Modelling Method and an Arena Template. *Simulation* **2003**, *79*, 43–55.

19.   Jansen-Vullers, M.; Netjes, M. Business Process Simulation - a Tool Survey. Proceedings of the the Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, 2006.

20. Ramos-Merino, M.; Alvarez-Sabucedo, L.M.; Santos-Gago, J.M.; de Arriba-Perez, F. A Pattern Based Method for Simplifying a BPMN Process Model. *APPLIED SCIENCES* **2019**, *9*, 2322–2350.

21. Erdogan, T.G.; Tarhan, A. A Goal-Driven Evaluation Method Based On Process Mining for Healthcare Processes. *APPLIED SCIENCES* **2018**, *8*, 894–916.

22. Arriagada-Benitez, M.; Sepulveda, M.; Munoz-Gama, J.; Buijs, J.C.A.M. Strategies to Automatically Derive a Process Model from a Configurable Process Model Based on Event Data. *APPLIED SCIENCES* **2017**, *7*, 1023–1051.

23. Wu, Q.; He, Z.; Wang, H.; Wen, L.; Yu, T. A Business Process Analysis Methodology Based on Process Mining for Complaint Handling Service Processes. *APPLIED SCIENCES* **2019**, *9*, 3313–3334.

24. Wynn, M.; Dumas, M.; Fidge, C.J. Business Process Simulation for Operational Decision Support. Proceedings of the 3rd International Workshop on Business Process Intelligence (BPI 07) in conjunction with Business Process Management Conference, 2007.

25. Verbeek, H.M.; Basten, T.; van der Aalst, W.M.P. Diagnosing Workflow Processes Using Woflan. *The Computer Journal* **2001**, pp. 246–279.

26. Kim, K.; Yeon, M.; Jeong, B.; Kim, K. A Conceptual Approach for Discovering Proportions of Disjunctive Routing Patterns in a Business Process Model. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS* **2017**, *11*, 1148–1161.

27. Kim, K.; Lee, Y.K.; Ahn, H.; Kim, K.P. An Experimental Mining and Analytics for Discovering Proportional Process Patterns from Workflow Enactment Event Logs. Proceedings of the International Conference on Big Data Technology and Applications, 2018, pp. 61–70.

28. Brandes, U.; Eiglsperger, M.; Herman, I.; Himsolt, M.; Marshall, M. GraphML Progress Report: Structural Layer Proposal. *Lecture Notes in Computer Science, The Proceedings of the International Symposium on Graph Drawing (GD2001)* **2002**, *2265*, 501–512.

29. Kim, K.; Lee, Y.K.; Ahn, H.; Kim, K.P. An Experimental Mining and Analytics for Discovering Proportional Process Patterns from Workflow Enactment Event Logs. *Wireless Networks* **2018**, *online*, 1–8.

30. IEEE. IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams; 2016.

31. Ahn, H.; Pham, D.L.; Kim, K.P. An Experimental Analytics on Discovering Work Transference Networks from Workflow Enactment Event Logs. *APPLIED SCIENCES* **2019**, *9*, 2368–2391. doi:https://doi.org/10.3390/app9112368.