An XPDL-Based Workflow Control-Structure and Data-Sequence Analyzer

Kwanghoon Pio Kim[1*]
[1] Division of Computer Science and Engineering, KYONGGI UNIVERSITY
Suwon, 16227, South Korea
[e-mail: kwang@kgu.ac.kr]
*Corresponding author: Kwanghoon Pio Kim

Abstract

A workflow process (or business process) management system helps to define, execute, monitor and manage workflow models deployed on a workflow-supported enterprise, and the system is compartmentalized into a modeling subsystem and an enacting subsystem, in general. The modeling subsystem's functionality is to discover and analyze workflow models via a theoretical modeling methodology like ICN, to graphically define them via a graphical representation notation like BPMN, and to systematically deploy those graphically defined models onto the enacting subsystem by transforming into their textual models represented by a standardized workflow process definition language like XPDL. Before deploying those defined workflow models, 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 corresponding workflow models. In this paper, we are particularly interested in verifying very large-scale and massively parallel workflow models, and so we need a sophisticated analyzer to automatically analyze those specialized and complex styles of workflow models. One of the sophisticated analyzers devised in this paper is able to analyze not only the structural complexity but also the data-sequence complexity, especially. The structural complexity is based upon combinational usages of those control-structure constructs such as subprocesses, exclusive-OR, parallel-AND and iterative-LOOP primitives with preserving matched pairing and proper nesting properties, whereas the data-sequence complexity is based upon combinational usages of those relevant data repositories such as data definition sequences and data use sequences. Through the devised and implemented analyzer in this paper, we are able eventually to achieve the systematic verifications of the syntactical correctness as well as the effective validation of the structural properness on those complicate and large-scale styles of workflow models. As an experimental study, we apply the implemented analyzer to an exemplary large-scale and massively parallel workflow process model, the Large Bank Transaction Workflow Process Model, and show the structural

complexity analysis results via a series of operational screens captured from the implemented analyzer.

1. Introduction

A workflow process management system (WPMS) is defined as a system that fully automates the definition, analysis, deployment, execution, and monitoring and controlling of work procedures in a process-aware enterprise. One of the essential components of WPMS is the modeling subsystem (Buildtime) that is supported by a graphical and formal methodology of workflow process models [1]. The conventional modeling subsystem is equipped with a series of functional components supporting from modeling a workflow process with graphical notations to deploying it onto the enacting subsystem (Runtime). In this paper, however, we try to detach the verification functionality from the Buildtime subsystem and make up a standalone tool that is exclusively charged for verification and structural analysis [2] of the workflow process models defined by the conventional Buildtime subsystem. As a consequence, we are able to accomplish the enterprise-wide goal which is minimize the loss of effectiveness and the depreciation of efficiency in controlling the workflow processes, so as for the workflow process designer to inspect the syntactical correctness as well as the structural performance the corresponding workflow process models prior to deploying them. This inspection ought to be much more effective if the corresponding workflow process models are especially characterized with very large-scale and massively parallel structures [3].

In this paper, we so focus on the structural complexity and verification on the control-structures [2] and data definition-use sequences [4] of very large-scale and massively parallel workflow process models, in particular. We assume that these models are textually represented in the standardized XML-based process definition language (XPDL) [5][6], and that it is also formally represented in a form of the information control nets [1]. In order to efficiently and effectively verify the structural correctness of control and data perspectives on these very large-scale and massively parallel workflow process models, we need to develop a control and data structure analysis tool supporting the structural verification and generating the analytical statistics, and we named the tool as an XPDL-based workflow control-structure and data-sequence analyzer. The structural components of a workflow 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. Based upon these structural components, the analyzer will be developed in this paper. In other words, we apply the structural verification functionality to the associative components, and on the other hand we apply the analytical statistics functionality to the entity components. Additionally, we carry out an experimental analysis on an exemplary large-scale and massively parallel workflow process model by using the developed analyzer. The model used in the experiment is the Large Bank Transaction Workflow Process Model released to the public by the 4TU.Centre for

Research Data [7] and the model is composed of 8 subprocesses and 113 activities with a large number of combinational control-structures such as exclusive-OR, parallel-AND and iterative-LOOP primitives.

In terms of organizing the paper, we are going to describe the functional scope of the workflow control-structure and data-sequence analyzer in the next section. In the consecutive sections, we specify the formal and graphical structural characteristics of the very large-scale and massively parallel workflow process models, describe the functional details and design artifacts of the architectural components of the analyzer, and carry out an experimental analysis for an exemplary model of the very large-scale and massively parallel workflow process model. Finally, we present a series of screens captured from the experimental deployment on the verifier and finalize the paper with surveying the related works and describing the implications.

2. Related Works

In this paper, we try to describe the detailed design and implementation of the XPDL-based control-structure and data-sequence analyzer and its functional application results as well as its experimental analytics especially for the large bank transaction process model characterized as the very large scale and massively parallel workflow process models. In the workflow technology research and development literature, there have been many reported and published workflow modeling and analysis tools and systems, until now. In recent, together with hot-issuing the workflow intelligence and mining knowledge, analyzing the control-structural aspect and the data definition and use sequence aspect of workflow process models (especially very large scale and massively parallel workflow process models) is becoming also much more meaningful functionality. In other words, the analyzing functionality proposed in this paper is an essential part as the prerequisite activities not only for the workflow process mining and knowledge discovery [8][9][10][11][14][19][28] but also for the workflow process simulating and knowledge estimation [15][16][18][23][24][25][26][27].

In particular, before describing the details, we perform the investigation on the state-of-the-art in the *workflow process simulating and knowledge estimation* issue. First of all, the authors of [15] defined the concept of the structured workflow process model and its properties, which are the properties that can be checked up by the analyzer proposed in this paper. They described a taxonomy that serves as a framework for analyzing unstructured workflows. The taxonomy organizes unstructured workflows in terms of two considerations: improper nesting and mismatched split-join pairs. This taxonomy characterizes situations that are well-behaved and others that are not, and that are well-behaved unstructured workflows having equivalent structured mappings or not. In [16], the authors proposed a functional mechanism for analyzing XPDL-based workflow process models and its implementation architecture, which is based upon the functional architecture of the analyzer proposed in this paper. [18] gave a definite intuition for the analyzer of this paper, in which the authors described a template that was built in the simulation language Arena, which is able to decrease the gap between the conceptualization activities and the translation into a simulation workflow process model. The necessity of the analyzer is given by [23], where they insisted that an important part of the evaluation of designed and redesigned business and workflow processes is business and workflow process analysis and simulation. The authors discussed a number of analysis and simulation tools that are relevant for the BPM and workflow field, evaluated their

applicability for business and workflow process analysis and simulations, and formulated recommendations for further research.

Theoretically, the analyzer to be proposed in this paper is based upon the theory of the information control net modeling methodology, whereas [26] presented a tool of workflow process analyzer based upon the theory of Petri net modeling methodology, which is named as Yasper as a tool for modeling, analyzing and simulating workflow systems, based on Petri nets. And other research outcomes like [24], [25] and [27] provided us the essential intuitions as well as the functional scopes and definitions for specifying the very useful requirements of the analyzer to be proposed in this paper.

3. Functional Scope of the Analyzer

Basically, there are two research topics of workflow process analytics that are mainly covered by the workflow process intelligence [8][9][10][11][12]. One is to support the mining and analytics activities to discover a variety of process-centered knowledges [13][14][15] from the audit trails and logs stored in Runtime, and the other is to support the workflow process simulation [16][17][18] to measure a series of process-centered performances on the workflow process repository defined in Buildtime. As a matter of the latter, we have a research plan to develop a sophisticated workflow process analyzer that is able to support the structural analytics functionality [12][13] as well as the simulative performance analytics functionality [14][15]. In particular, the structural analytics functionality is dealt in this paper. The overall functional scope of the analyzer to be developed is illustrated in **Fig. 1**. The colored components, like structural statistics and report generator, XPDL structural analyzer, and models and analytics visualizer, in the left-hand side of the figure are those essential functionalities to be dealt in the paper, while the gray-colored components, like XPDL simulation analyzer and simulation analytics and report generator, in the right-hand side will be covered in the future work of the paper. Especially, the XPDL structural analyzer is able to support the analytical functions related with the relevant data definition and use sequences in a workflow process model. Every activity in a workflow process model has to have a special association of relevant-data group that can be used for input and output repositories of a corresponding activity. Conclusively, the scope and goal of the paper is to develop a sophisticated workflow process analyzer named as the XPDL-based control-structure and data-sequence analyzer.
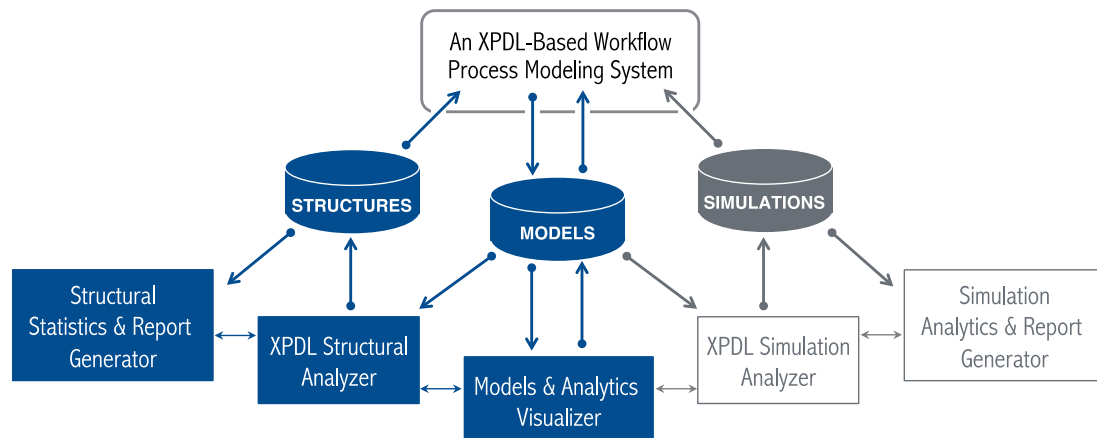

Fig. 1. The Functional Compoinents and the Scope of the Analyzer

4. Structural Attributes in the Analyzer

In this section, we define the theoretical model of workflow procedures and its structural attributes as the input and the subject being analyzed by the analyzer proposed in the paper. A workflow process model can be formally represented by the information control net methodology [1][17] abbreviated as ICN, and the model is eventually represented by the standardized format of XPDL [5][6][20] standing for XML process definition language. We assume that all the algorithms to be designed for the analyzer are formalized in the mathematical notation of the information control net methodology. And the algorithmic programs to be implemented for the analyzer are based on the standardized format of XPDL. In this section, we define the structural attributes of workflow control-flow and data-flow from the point of theoretical view (ICN) and the point of practical view (XPDL) as well.

4.1 Structural Attributes in ICN

The information control net methodology (ICN) is based upon a theoretical graph model to formally represent the model of workflow procedures, and it is known as the most well-fitted methodology to describe and analyze information flows by capturing temporal transitions and associations among the essential entities, such as activities, roles, actors, applications and repositories, within a workflow procedure. The ICN has also been used within actual as well as hypothetical automated offices to yield a comprehensive description of business-activities, to test the underlying office description for certain flaws and inconsistencies, to quantify certain aspects of office information flows, and to suggest the possible office restructuring permutations.
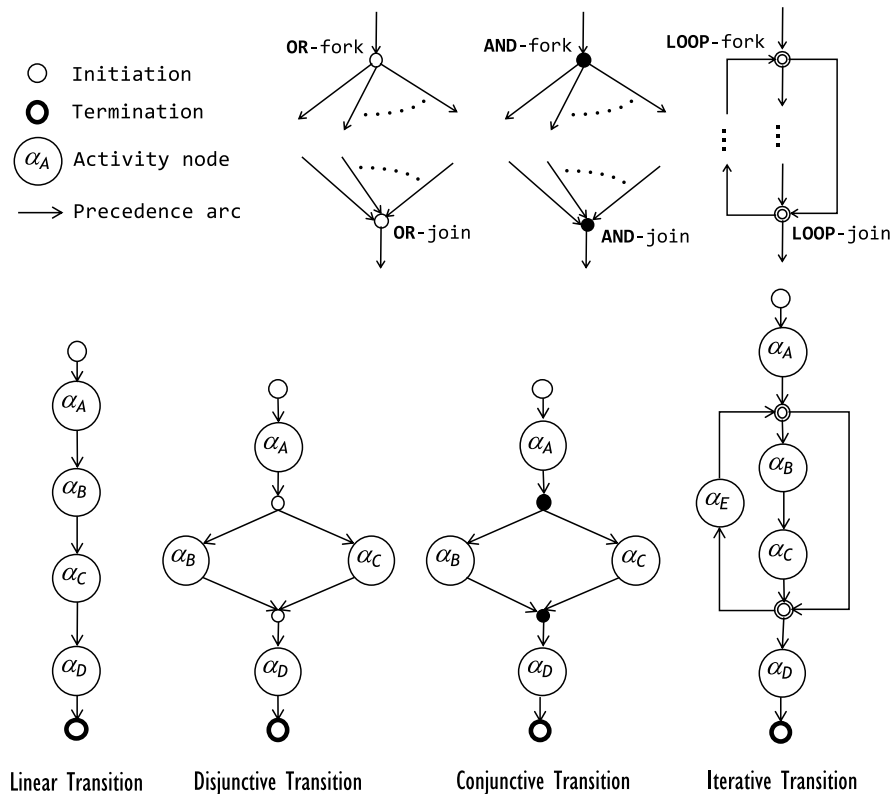

Fig. 2. Four Types of Control-Flow Transitions in ICN

In this subsection, we focus on the structural attributes in the model of the information control nets, firstly. The structural complexity of an information control net model is determined by the control-flow attributes that are built through the combinations of four types of the structural transition primitives, such as sequential, disjunctive, conjunctive and iterative transition primitives, as shown in **Fig. 2**. The control-structure of a workflow process model is built by a set of activities connected by temporal orderings (control-flow) called gateway-activity control-transitions. Activities can be related to each other by combining linear (sequential) transition type, disjunctive (exclusive-OR: after activity α_A, do either activity α_B or α_C, alternatively) transition type with predicates attached, conjunctive (parallel-AND: after activity α_A, do activities α_B and α_C concurrently) transition type, and iterative (LOOP: after activity α_A, entering the loop-body of activities α_B and α_C with repeating activity α_E) transition type. Naturally, there are three types of workflow activities: compound activity type characterized by the concept of subprocesses, elementary activity type concretized by the basic unit of office work with invoked applications, and gateway activity type classified into three transitions of control-flow such as disjunctive, conjunctive and iterative with their nested and matched pairs of splits and joins, as shown in **Fig. 2**. Based upon these basic concepts and their structural components, we can define an information control net model of workflow procedures and its structural attributes as the following [Definition 1]:

[Definition 1] Structural Attributes in the Information Control Net. A basic structure of an information control net model of workflow procedures is formally defined through 8-tuple $=(\delta, \gamma, \chi, \varepsilon, \pi, \kappa, \mathbf{I}, \mathbf{O})$ over a set $\mathbf{A}$ of activities (including a set of group activities), a set $\mathbf{T}$ of transition conditions, a set $\mathbf{R}$ of repositories, a set $\mathbf{G}$ of invoked application programs, a set $\mathbf{P}$ of roles, and a set $\mathbf{C}$ of actors or performers (including a set of actor groups), 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, $\delta_o : \mathbf{A} \rightarrow P(\alpha \in \mathbf{A})$ is a multi-valued mapping function of an activity to its set of (immediate) successors, and $\delta_i : \mathbf{A} \rightarrow P(\alpha \in \mathbf{A})$ is a multi-valued mapping function of an activity to its set of (immediate) predecessors;
- $\gamma = \gamma_i \cup \gamma_o$: *Data-Flow Structural Attributes*
  where, $\gamma_o : \mathbf{R} \rightarrow P(\alpha \in \mathbf{A})$ is a multi-valued mapping function of an activity to its set of output data repositories, and $\gamma_i : \mathbf{R} \rightarrow P(\alpha \in \mathbf{A})$ is a multi-valued mapping function of an activity to its set of input data repositories;
- $\chi = \chi_a \cup \chi_p$: *Invoked Application Associative Attributes*
  where, $\chi_p : \mathbf{G} \rightarrow P(\alpha \in \mathbf{A})$ is a single-valued mapping function of an activity to its invoked application program, and $\chi_a : \mathbf{A} \rightarrow P(\tau \in \mathbf{G})$ is a multi-valued mapping function of an invoked application program to its set of associated activities;
- $\varepsilon = \varepsilon_a \cup \varepsilon_p$: *Role Associative Attributes*
  where, $\varepsilon_p : \mathbf{P} \rightarrow P(\alpha \in \mathbf{A})$ is a single-valued mapping function of an activity to one of the roles, and $\varepsilon_a : \mathbf{A} \rightarrow P(\eta \in \mathbf{P})$ is a multi-valued mapping function of a role to its sets of associated activities;

- $\pi = \pi_p \cup \pi_c$: *Actor (Performer) Associative Attributes*
  where, $\pi_c : \mathbf{C} \to \mathrm{P}(\eta \in \mathbf{P})$ is a multi-valued mapping function of a role to its sets of associated actors, and $\pi_p : \mathbf{P} \to \mathrm{P}(o \in \mathbf{C})$ is a multi-valued mapping function of an actor to its sets of associated roles;

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

Table 1. Structural Attributes in XPDL

Package	WorkflowProcesses	Activities	Transitions	Implementation	DataFields	Participant
- Id - Name - Description - Extended Attributes	- Id - Name - Description - Extended Attributes	- Id - Name - Description - Extended Attributes	- Id - Name - Description - Extended Attributes	- Id - Name - Description - Extended Attributes	- Id - Name - Description - Extended Attributes	- Id - Name - Description - Extended Attributes
- XPDL version - Source Vendor ID - Creation Date - Version - Author - Country Key - Publication Status - Conformance Class - Priority Unit	- Creation Date - Version - Author - Codepage - Country Key - Publication Status - Priority - Limit - Valid From Date - Valid To Date	- Automation Mode - Split - Join - Priority - Limit - Start Mode - Finish Mode - Deadline			- Data Type	- Participant Type
- Responsible - External Package	- Parameters - Responsible	- Performer - Tool - Subflow - Activityset - Actual Parameter	- Condition - From - To	- Actual Parameters	- Initial Value	
- Documentation - Icon	- Documentation - Icon	- Documentation - Icon				
- Cost Unit	- Duration Unit - Duration - Waiting Time - Working Time	- Cost - Duration - Waiting Time - Working Time				

4.2 Structural Attributes in XPDL

In this subsection, we describe the structural attributes in the XML process definition language (XPDL) [22][29][30][31][32], the specifications of which was released by the international standardization organization of workflow management coalition. The XPDL version 1.0 [5] is the XML-formatted version of the workflow process definition language (WPDL) [20]. Recently, the workflow management coalition released the new specifications of XPDL reflecting the OMG's standardized graphical notation of the workflow process modeling notation (BPMN) [21] as the XPDL version 2.0 [6]. In this paper, we consider the XPDL version 1.0 as the textual format of the workflow process model, because the XPDL version 2.0 is extended from the version 1.0 only for the artifacts of supplementary BPMN notation [21], such as pool, lane, annotation, event and so on, which are directly related with the workflow process model, itself. In particular, the control-flow structural attributes can be formed by the FROM and TO properties in the transition attribute of the XPDL standard format. Conclusively, the structural attributes of the XPDL version 1.0 are summarized in **Table 1**. Also, the data-flow structural attributes can be formed by the actual parameter property with modes of IN and OUT. Because of the page limitation, we simply introduce the structural attributes and their properties in this paper.

4.3 Data Definition and Use Sequences

In this subsection, we describe the concept of data-sequences in workflow procedures. There are two types of data-sequences; One is a type of data *definition* sequences, and the other is a type of data *use* sequences. The data-sequence concept is concretized from analyzing the data-flow associations [1] between the activities and the relevant data in a workflow model of information control nets. In other words, the data *definition* sequence type implies the sequence of WRITE-operations executed by a group of the activities that are associated with their relevant data in the data-flow associations of an information control net model, whereas the data *use* sequence type defines the sequence of READ-operations executed by a group of the activities that are associated with their relevant data in the data-flow associations of the information control net model. Note that the data-flow associations are discovered from the data-flow structural attributes ($\gamma = \gamma_i \cup \gamma_o$) defined in a corresponding information control net model. From the data-flow associations and their corresponding information control net model, we can discover a series of the data *definition* sequences and data *use* sequences. There are also four patterns of data-sequences in a workflow model of information control nets, in general. The following are the formal notations for formally representing these patterns and their associated activity occurrences in the relevant data *definition* and *use* sequences:

- Linear data-sequence: ➔
- Disjunctive data-sequence: ⌄
- Conjunctive data-sequence: ⌃
- Iterative data-sequence: ( )$^+$

Fig. 3 is to illustrate the data-flow structural attributes between the relevant data and the activities in two exemplary models of information control nets. Also, **Fig. 4** shows a sample of the data-flow structural attributes definitions [22] in XPDL. In **Fig. 3**, three relevant data types, r_1, r_2 and r_3, are in the ICN model of the left-hand side, and two relevant data types, r_1 and r_2, are in the ICN model of the right-hand side. And **Fig. 4** represents two actual parameters, `orderInfo` and `orderNumber`, and their involvements in an activity, `Enter Order (ID: 32)`. Theoretically, the data-flow structural attributes in an information control net model can be formalized by the concepts of data *definition* and data *use* operations, which is so-called data-flow associations. The following are the data-flow associations and the data-sequence associations discovered from the exemplary ICN models:
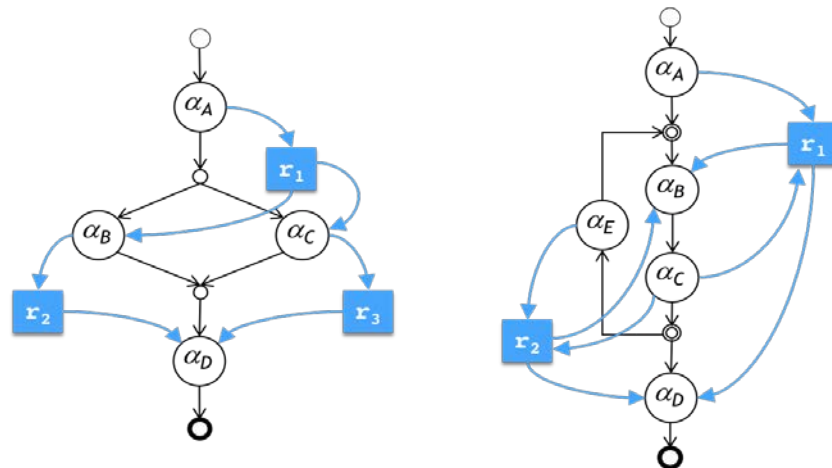

Fig. 3. Relevent Data Definitions and Uses in ICNs

(1) Data-Flow Associations:

- Definitions and Uses of relevant data in the ICN model (left-hand)

$$\text{Definition}(r_1) = \{\ a_A\ \};\ \ \text{Use}(r_1) = \{\ a_B, a_D\ \};$$
$$\text{Definition}(r_2) = \{\ a_B\ \};\ \ \text{Use}(r_2) = \{\ a_D\ \};$$
$$\text{Definition}(r_3) = \{\ a_C\ \};\ \ \text{Use}(r_3) = \{\ a_D\ \};$$

- Definitions and Uses of relevant data in the ICN model (right-hand)

$$\text{Definition}(r_1) = \{\ a_A, a_C\ \};\qquad \text{Use}(r_1) = \{\ a_B, a_D\ \};$$
$$\text{Definition}(r_2) = \{\ a_C, a_E\ \};\ \text{Use}(r_2) = \{\ a_B, a_D\ \};$$

(2) Data-Sequences Associations:

- Data Definition/Use Sequences in the ICN model (left-hand)

$$\text{Def-sequence}(r_1) = (\ a_A\ );\ \text{Use-sequence}(r_1) = (\ a_B \blacktriangledown a_C\ )\ ;$$
$$\text{Def-sequence}(r_2) = (\ a_B\ );\ \text{Use-sequence}(r_1) = (\ a_D\ )\ ;$$
$$\text{Def-sequence}(r_3) = (\ a_C\ );\ \text{Use-sequence}(r_1) = (\ a_D\ )\ ;$$

- Data Definition/Use Sequences in the ICN model (right-hand)

$$\text{Def-sequence}(r_1) = (\ a_A \blacktriangleright (a_C)^+\ );\ \text{Use-sequence}(r_1) = (\ (a_B)^+ \blacktriangleright a_D\ );$$
$$\text{Def-sequence}(r_2) = (\ a_C \blacktriangleright a_E\ )^+;\ \text{Use-sequence}(r_1) = (\ (a_B)^+ \blacktriangleright a_D\ );$$

```
<WorkflowProcesses>
    <WorkflowProcess Id="1" Name="EOrder" AccessLevel="PUBLIC">
    <ProcessHeader/>
    <DataFields>
        <DataField Id="orderNumber" IsArray="FALSE">
            <DataType>
                    <BasicType Type="INTEGER"/>
            </DataType>
            <InitialValue>0</InitialValue>
            <Length>0</Length>
        </DataField>
    </DataFields>
    ...
    <Activities>
        <Activity Id="32" Name="Enter Order">
        <Implementation>
        <Tool Id="enterOrder" Type="APPLICATION">
           <ActualParameters>
               <ActualParameter>orderInfo</ActualParameter>
               <ActualParameter>orderNumber</ActualParameter>
           </ActualParameters>
        </Tool>
        </Implementation>
        ...
        </Activity>
        ...
    </Activities>
    ...
    </WorkflowProcess>
    ...
</WorkflowProcesses>
```

Fig. 4. A Sample of the Data-Flow Structural Attributes Definition in XPDL

5. Functional Architecture and Implementation of the Analyzer

In this section, we describe the functional architecture of the analyzer and its design and implementation details. The essential part of the analyzer is on the functional architecture and its theoretical algorithms, but on the other hand the detailed description of the design and implementation is based on a series of computer-screens captured from the operational study carrying out the structural analysis on the exemplary model of workflow procedure, the Automatic Teller Machine Malfunctional Error Handling Workflow Process Model, which was deployed on a company providing the workflow-supported maintenance services for Banks' automatic teller machines.

5.1 Functional Architecture

The functional architecture of the analyzer is depicted in **Fig. 5**. There are five functional components with a database connection agent: Dashboard Manager, XPDL Control-structure Analyzer, XPDL Schema Parser and Verifier, XPDL Data-Sequence Analyzer, Analysis Report Generator and Visualizer. The database schema for the analyzer consists of MODELS database and ANALYSIS RESULTS database. The MODELS database stores the XPDL-based workflow process models built by the workflow process modeling tool, while the ANALYSIS RESULTS database preserves the analyzed results including the statistical data of the structural attributes in the corresponding workflow process models.
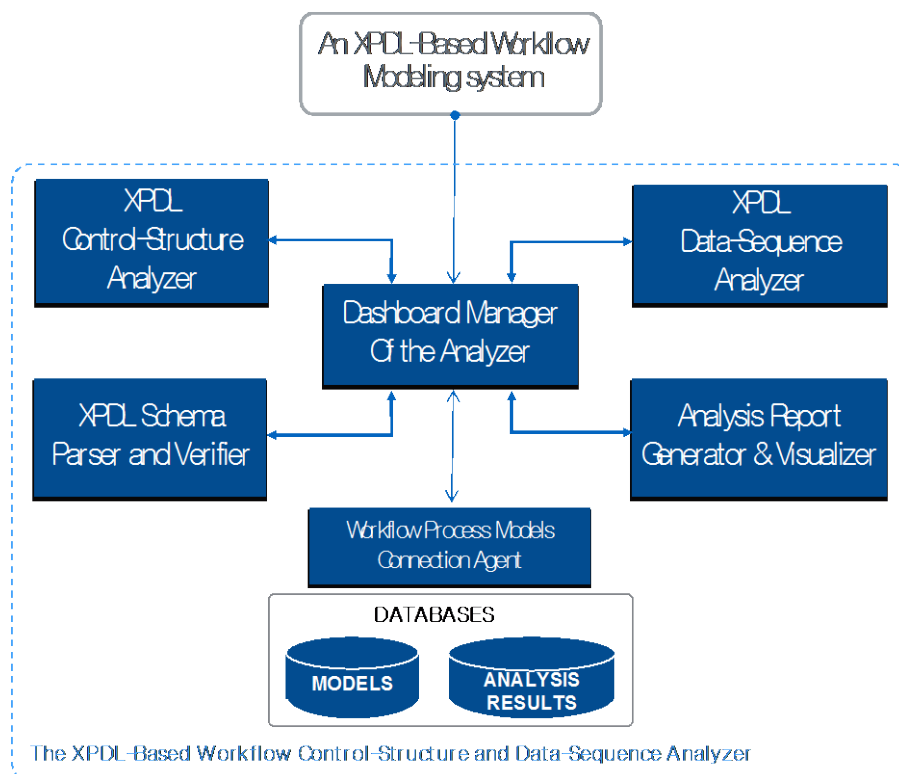

Fig. 5. The Functional Architecture of the Analyzer

The *Dashboard Manager* is in charge of the overall control of the analyzer including user management, session & access control, and XPDL file management. Assume that the XPDL-based workflow process modeling system, the theoretical background of which is the information control net methodology, is able to support the Export-to-XPDL functionality that is able to transform the graphical representation of an ICN-based workflow process model into a textual representation of its corresponding XPDL-based workflow process model. Continuously, the manager is also able to store the XPDL-based workflow process model onto the MODELS database schema by opening the corresponding XML-formatted XPDL file. Since then, all the functions and operations, such as verifying, analyzing, reporting and visualizing operations, to be applied into the XPDL-based workflow process model are controlled and managed *via* this manager.

The *XPDL Schema Parser and Verifier* is able to parse an XPDL-based workflow process model and check up its syntactical correctness according to the XML format criteria. After completing and passing the verification phase, the *XPDL Control-Structure Analyzer* performs the following verifiable analysis functions:

- *Control-Structure rule verification*: Checking up whether the corresponding model is keeping the rules of proper-nesting and matched-pairing in building its gateway-type activities.

- *Association rule verification*: Checking up whether the corresponding model is keeping the correct association rules in building activity-to-role associations, activity-to-program associations, activity-to-data associations, and role-to-actor associations.

The *XPDL Data-Sequence Analyzer* is able to generate a series of relevant data definition and use sequences for all the relevant data used in an XPDL-based workflow process model. The data-sequence analyzer works in the principle of the data *definition* sequences and data *use* sequences performed by their associated activities of an information control net model. As described in the previous section, a series of data definition sequence associations and data use sequence associations can be generated from both the data-flow associations and the relevant data definition and use attributes in a corresponding information control net model, in theory. Based upon the theoretical basis, the data sequence analyzer concretizes the data update sequences and the data reference sequences on the specific relevant data set associated with a corresponding workflow process model. It ought to be so useful knowledge for a workflow management engine to recover the enactment of instances from their erroneous situations.

The *Analysis Report Generator and Visualizer* is to generate the analytical statistics of each structural components in an XPDL-based workflow process model and visualize the analytical results in a variety of graphical forms. This generator is able to produce two-level analytical statistics. One is the process-level analytical statistics, and the other is the package-level analytical statistics. Note that a workflow process package comprises a group of workflow process models, and the XPDL schema is formatted from a pair of the package-level tags like `<package> ... </package>`. The following are the analytical statistics to be analyzed and produced by the generator:

(1) Process-level Analytical Statistics
- Species of structural patterns and their usage ratios
- The number of participants (actors or performers) and their participation ratios
- The number of roles and their involvement ratios
- The number of invoked applications and their usage ratios
- The number of relevant data types and their usage ratios
- The number of subprocesses and their usage ratios

(2) Package-level Analytical Statistics
- The number of workflow process models in a corresponding workflow process package
- The number of activities in each model and their usage ratios
- The number of roles in each model and their involvement ratios
- The number of invoked applications in each model and their usage ratios
- The number of relevant data types in each model and their usage ratios
- The number of subprocesses in each model and their usage ratios
- The usage ratio of each model as subprocesses

5.2 Design and Implementation

Based upon the structural attributes and the functional architecture, we designed implemented the XPDL-based control-structures and data-sequence analyzer that is able to verify the syntactical correctness of those XPDL-based workflow process models, analyze their structural statistics and relevant data definition and use sequences, and visualize their analytical outcomes and structural statistics. The implementation and operational computing environment are characterized as follows:

- Operating system: Windows XP Pro Version 2002 Service Pack 3
- Implementation programming language: Java Development Toolkit v6.0
- workflow process definition language: XPDL 1.0
- Libraries: JGraph, JFreeChart, etc.
- Development toolkit: JBuilder 2006

As stated in the previous section, the implemented analyzer supports to produce the two-level structural analytics such as the package-level analytical statistics and the process-level analytical statistics. In the package-level, it is able to support the activity-type analytics, component-type analytics, component-type's usage ratio analytics, and the subprocess usage ratio analytics. It also supports the activity-related structural attributes analysis and the associative attributes analysis such as activity-to-program, activity-to-role, activity-to-data, and role-to-performer associations. **Fig. 6**, **Fig. 7**, and **Fig. 8** are a series of the operational screens captured from the implemented analyzer.
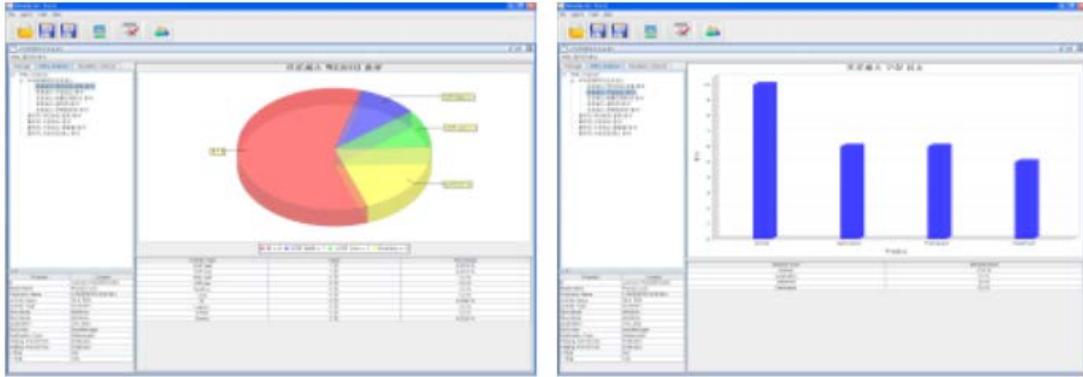
Fig. 6. Activity-types and Structural Attributes Statistics in the Automatic Teller Machine Malfunctional Error Handling Process Model

Fig. 6 shows two operational screens displaying a pie-chart with the number of activity-types and a bar-chart with the number of structural attributes, respectively, for the imaginary workflow process model, the Automatic Teller Machine malfunctional error handling workflow process model. **Fig. 7** also shows two operational screens displaying the analyzed bar-charts for the invoked applications' usage ratios and the numbers of the structural components for each workflow process model, respectively. Additionally, **Fig. 8** shows the bar-charts of the activity-types and the structural components' usage ratios built in each workflow process model through two operational screens.
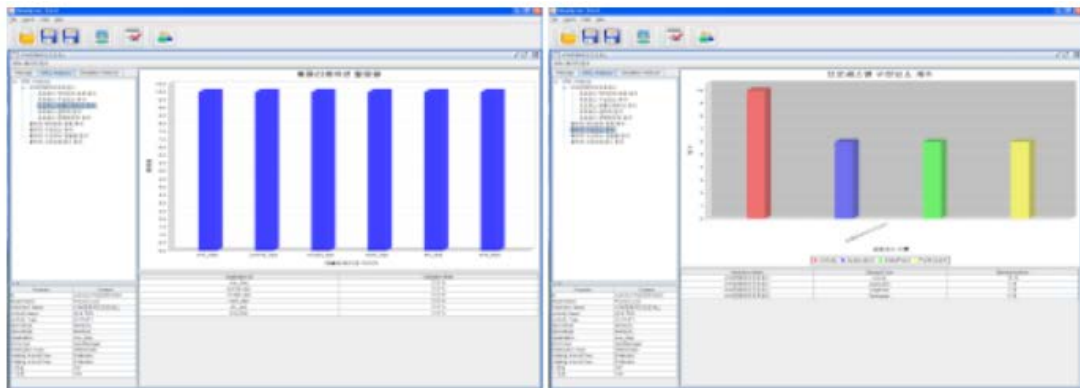

Fig. 7. Applications' Usage Ratios and Structural Components' Statistics in the Automatic Teller Machine Malfunctional Error Handling Process Model
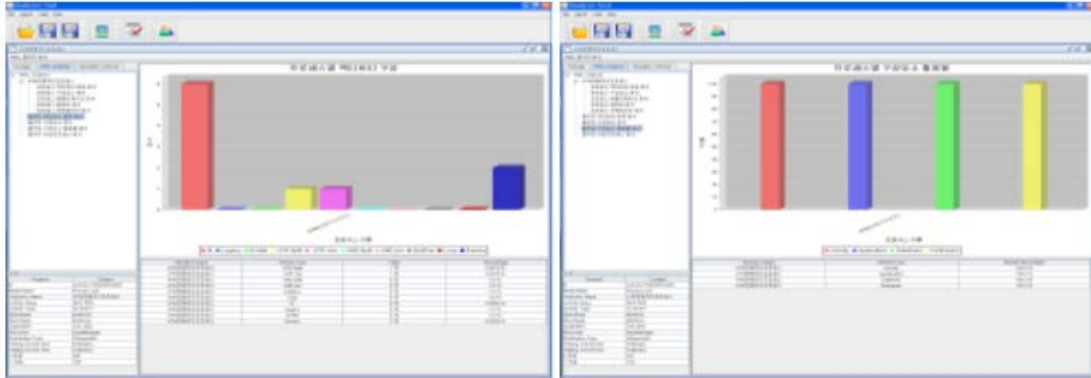
Fig. 8. Activity-types and Structural Components' Usage Ratios in the Automatic Teller Machine Malfunctional Error Handling Process Model

6. An Experiment of the Analyzer

The crucial advantage of the analyzer ought to be on the cases of analyzing such very large scale and massively parallel workflow process models. In this section, we carry out an experimental analysis based on a typical workflow process model of large-scale and massively parallel structural components. The model used in the experiment is the Large Bank Transaction Workflow Process Model released to the public by the 4TU.Centre for Research Data [7]. We would strongly believe that this model ought to be very large scale and massively parallel in terms of its structural characteristics, because this workflow process model is composed of 8 subprocesses and 113 activities with a large number of combinational control-structures such as exclusive-OR, parallel-AND and iterative-LOOP primitives.

6.1 An Experimental Model

For the purpose of the experiment, first of all it is very important to explore and acquire a very large scale and massively parallel workflow process model. Speaking the conclusion first, we found out one of those models from the BPI Challenges' datasets in 4TU.Centre for Research Data [7]. In other words, we fulfilled a workflow process mining experiment on the dataset of the 2018 BPI Challenge and discovered the Large Bank Transaction Process Model from the workflow event log dataset as followings:

The Large Bank Transaction Process Model: This is a synthetic event log, it was published by Universitat Polit`ecnica de Catalunya. It described the bank transfer structure from the open and register transaction step to notify and close transaction step. There are 113 activities and 8 subprocesses in this workflow process model. Fig. 9 shows an information control net model of the Large Bank Transaction Process Model that is systematically discovered from the dataset by the workflow process mining system developed by the author's research group.
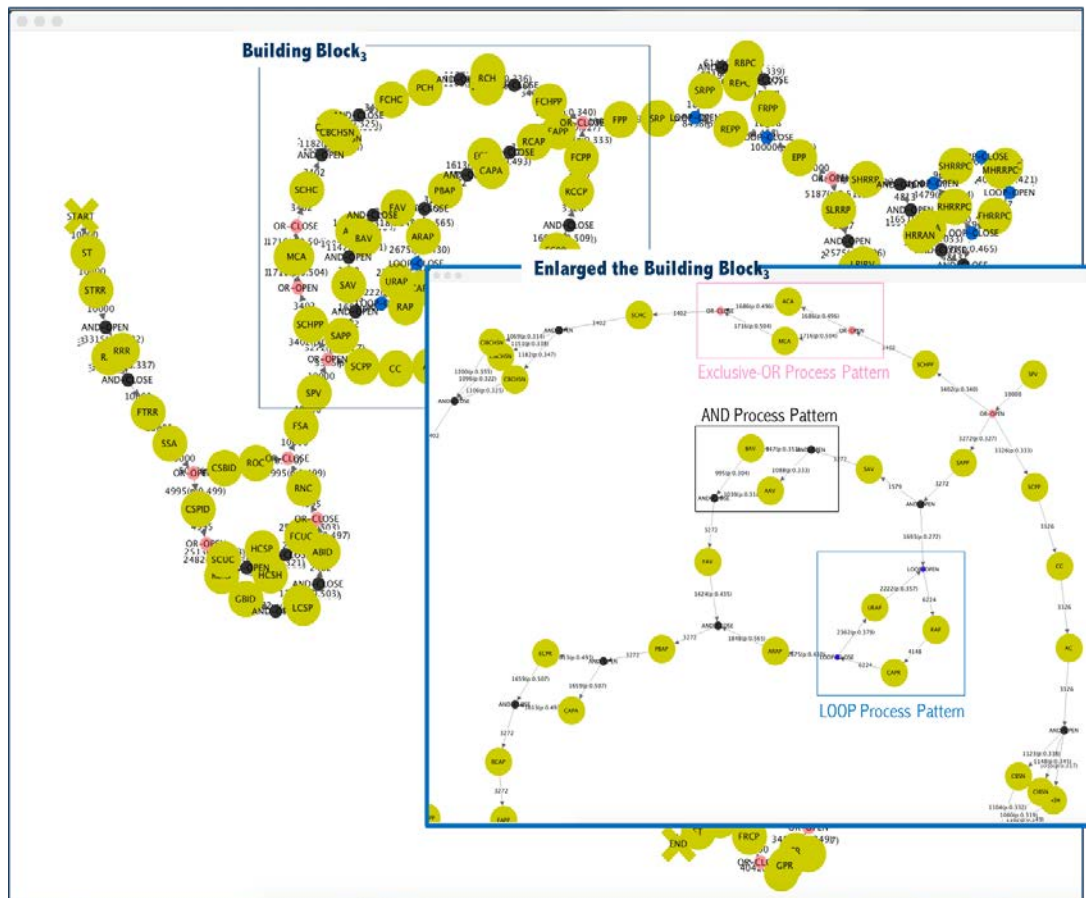
Fig. 9. A Discovered Information Control Net Model of the Large Bank Transaction Process Model

As we can see in an enlarged view of the building block of the figure, the information control net model contains all the different patterns of control-structures such as linear, exclusive-OR, parallel-AND and iterative-LOOP process patterns. Through the experiment by using the implemented analyzer, we are able to automatically analyze its control-structures and their statistical analytics. Before carrying out the experiment, it is necessary to prepare an XPDL-formatted workflow process model of the large bank transaction process model. We performed a modeling work for defining an ICN-based workflow process model based on the large bank transaction process model for the discovered ICN model of **Fig. 9**. The result of the modeling work is shown in **Fig. 10** that is screen-captured from an ICN-based workflow process modeling system and that is also rearranged with three screen-cuts (I, II and III) for presenting a large number of activities all together in a single figure. From this modeling work, we were able to prepare an XPDL-based workflow process model for analyzing the large bank transaction process model by the implemented analyzer. Besides the control-structural aspect, we supplemented the model with the other structural components, like invoked applications, relevant data set, roles and performers, so as for the modeling work to be completed as a result of the sound workflow process model.
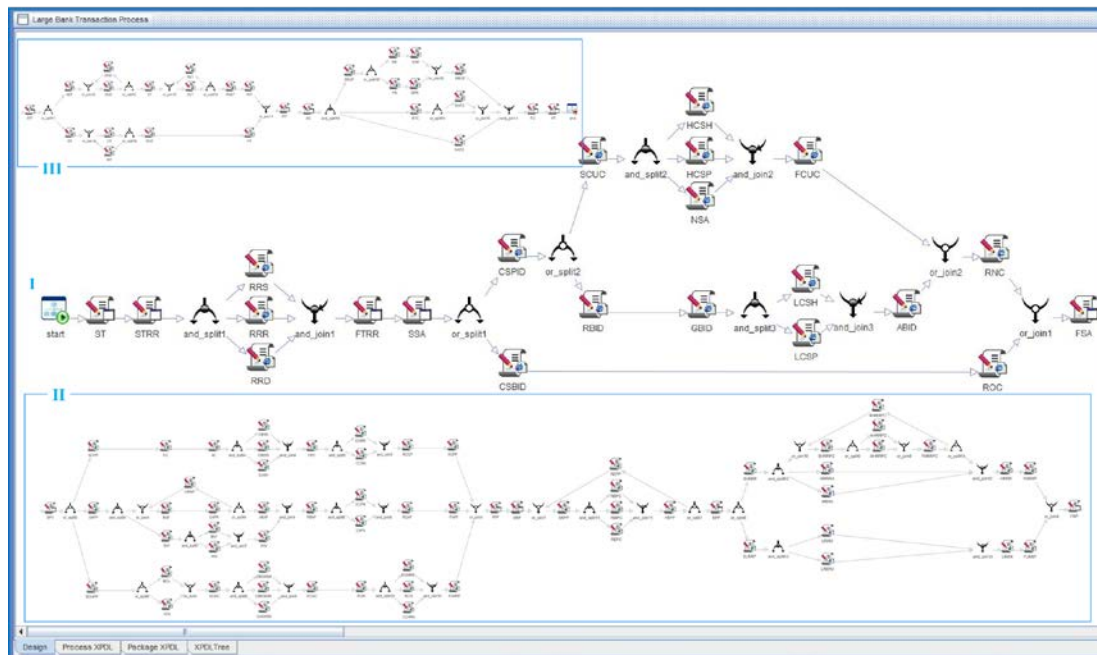
Fig. 10. The ICN-Based Large Bank Transaction Process Model Defined by an ICN-Based Workflow Modeling System

6.2 An Experimental Analysis on the Model

Based upon the XPDL-based workflow process model prepared in the previous section, we deployed the XPDL-formatted large back transaction process model onto the implemented analyzer. As stated in the previous, the crucial advantage of the analyzer implemented in this paper is to easily and effectively analyze a sort of very large scale and massively parallel workflow process models in terms of their control-structures and data-sequences. In this experiment, we were able to fulfill only the analysis of the control-structural aspect of the very large scale and massively parallel workflow process model, because only the control-structural aspect of the large bank transaction process model was discovered from the mining activities of the BPI challenge datasets. Nevertheless, it ought to be worthy enough for the analyzer to be used for analyzing those very large scale and massively parallel workflow process models.

Fig. 11 shows the analyzed results and their visualization that were obtained from a series of analysis works based upon the XPDL-based large bank transaction process model. The four visualized graphs generated from the implemented analyzer present the control-structural characteristics of the analyzed model as followings:

- Screen 1: The level of modeling completeness of the structural components (`Activities – 100%, Involed applications – 100% and Relevant data set – 100%`) used in the analyzed model
- Screen 2: The number of each of the structural components (`Activity – 173 including event and gateway activities, Invoked application – 18 application types, Relevent data set – 7 data fields`) used in the analyzed model

- Screen 3: The number of subprocesses used in the analyzed model, which is `zero`. Note that the reason is because the modeling system used does not support the subprocess definition functionality.
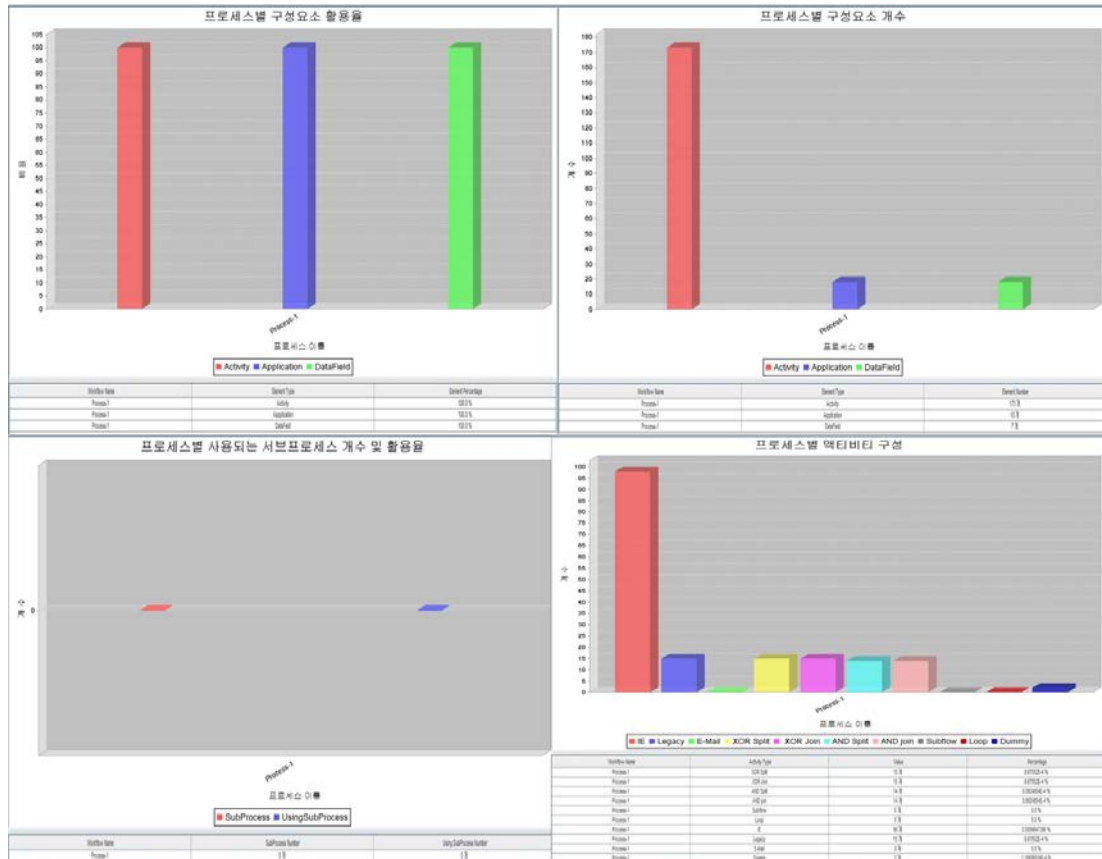

Fig. 11. The Analyzed Results and Their Visualization of the Large Bank Transaction Process Model

- Screen 4: The number of each activity used in the analyzed model as followings:
  - o Gateway activity type
    - `exclusive-OR split – 15 (including LOOP splits)`
    - `exclusive-OR join – 15 (including LOOP join)`
    - `parallel-AND split – 14`
    - `parallel-AND split – 14`
    - `interaitve-LOOP – 0`[1]
  - o Application activity type
    - `subprocess – 0`
    - `Activity – 98`
    - `Legacy Application type activity – 15`

---

[1] The reason is because the XPDL does not provide the LOOP gateway activity type through explicit makeups, and so the LOOP gateway activity types are defined via exclusive-OR gateway types.

- ▪ `E-Mail type activity – 0`
  - o Event activity type
    - ▪ `Event (start and end) activity - 2`

7. Conclusion

In this paper, we have described the functional and architectural details for designing and implementing an XPDL-based control-structure and data-sequence analyzer that is used for verifying the XPDL-based workflow process models. In order to precisely identify the structural attributes and data definition and use sequences in a workflow process model as the subjects to be analyzed, we formally extracted them from the information control nets as well as from the XML process definition language. Additionally, we devised the functional architecture for the analyzer, and described the specifications of the functional components, as well. Finally, we designed and implemented the XPDL-based analyzer based upon the functional architecture and verified the implemented analyzer by showing a series of operational screens captured from the operational example of the imaginary workflow process model. For the sake of the usefulness of the analyzer, we carried out an experimental analysis on a specific example of the very large scale and massively parallel workflow process models, which is the Large Bank Transaction Process Model discovered from the workflow log dataset provided for the BPI challenge contest. In conclusion, the issues of the workflow process modeling 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 workflow process analytical techniques and simulation methodologies that are used for finally giving feed-backs to the redesign and reengineering phase of the existing workflow 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 workflow process analytics and simulation technology. As future works to be extended from the paper, we need to conceive a feasible approach to systematically connect to those workflow process mining systems that automatically provide a certain model of very large-scale workflow processes so as for the implemented analyzer of this paper to analyze its structural patterns and data-sequences.

Acknowledgements

References

[1] Kim, K., Ellis, C. A., "Section II / Chapter VII. An ICN-based Workflow Model and Its Advances," *Handbook of Research on BP Modeling*, IGI Global, ISR, pp. 142-172, 2009. Article (CrossRef Link)

[2] K. Kim, C. Lee, B. Jeong and K. P. Kim, "An XPDL-based Structural Analyzer for Verifying Business Process Models," in *Proc. of the 12th Asia Pacific International Conference on Information Science and Technology*, pp. 137–144, 2017.

[3] K. Kim, "An Enterprise Workflow Grid/P2P Architecture for Massively Parallel and Very Large-scale Workflow Systems," *Lecture Notes in Computer Science*, Vol. 3842, pp. 472–476, 2006. Article (CrossRef Link)

[4] Jaeyoung Yun, M. Jin, D. Pham and K. P. Kim, "Conversion of an ICN-Based Workflow Process Model to a Data-Sequence-based Workflow Model," in *Proc. of the 13th Asia Pacific International Conference on Information Science and Technology*, pp. 33-35, 2018.

[5] Workflow Management Coalition Specification Document, "XML Process Definition Language (XPDL)," *Document Number WFMC-TC-1025: Version 1.14 Document Status – Final*, October 3, 2005.

[6] Workflow Management Coalition Specification Document, "XML Process Definition Language (XPDL)," *Document Number WFMC-TC-1025: Version 2.1 Document Status – Working*, December 17, 2005.

[7] 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.

[8] Won, J., "A Framework: Organizational Network Discovery on Workflows," *Ph.D. Dissertation, Department of Computer Science*, KYONGGI UNIVERSITY, 2008. Article (CrossRef Link)

[9] Skerlavaj, M., Dimovski, V., Desouza, K. C., "Patterns and Structures of Intra-Organizational Learning Networks within a Knowledge-Intensive Organization," *Journal of Information Technology*, Vol. 25, No. 2, pp. 189-204, 2010. Article (CrossRef Link)

[10] Aalst, W. M. P., et al., "Discovering Social Networks from Event Logs," *COMPUTER SUPPORTED COOPERATIVE WORK*, Vol. 14, No. 6, pp. 549-593, 2005. Article (CrossRef Link)

[11] Song, J., et al., "A Framework: Workflow-based Social Network Discovery and Analysis," in *Proc. of the International Workshop on Workflow Management in Service and Cloud Computing*, Hongkong, China, pp. 421-426, 2010.

[12] Chuang, S., Liao, C., Lin, S., "Determinants of Knowledge Management with Information Technology Support Impact on Firm Performance," *Information Technology and Management*, Vol. 14, Iss. 3, pp. 217-230, 2013. Article (CrossRef Link)

[13] Poelmans, S., Reijers, H. A., Recker, J., "Investigating the Success of Operational Business Process Management Systems," *Information Technology and Management*, Vol. 14, Iss. 4, pp. 295-314, 2013. Article (CrossRef Link)

[14] Kyoungsook Kim, Moonsuk Yeon, Byeongsoo Jeong, Kwanghoon Kim, "A Conceptual Approach for Discovering Proportions of Disjunctive Routing Patterns in a Business Process Model," *KSII Transactions on Internet and Information Systems*, Vol. 11, No. 2, pp. 1148-1161, 2017. Article (CrossRef Link)

[15] R. Liu and A. Kumar, "An Analysis and Taxonomy of Unstructured Workflows," *Lecture Notes in Computer Science*, Vol. 3649, pp. 268–284, 2005. Article (CrossRef Link)

[16] Minhyuck Jin and K. P. Kim, "An XPDL-Based Structural Workflow Process Simulator," in *Proc. of the 13th Asia Pacific International Conference on Information Science and Technology*, pp. 217-221, 2018.

[17] Clarence A. Ellis and Gary J. Nutt, "Office Information Systems and Computer Science," *Computing Surveys*, Vol. 12, No. 1, March 1980. Article (CrossRef Link)

[18] G.J. de Vreede, A. Verbraeck, and D.T.T. van Eijck. "Integrating the Conceptualization and Simulation of Business Processes: A Modelling Method and an Arena Template," *SIMULATION*, Vol. 79(1), pp. 43–55, 2003. DOI: Article (CrossRef Link)

[19] Kwang-Hoon Kim, "Sigma-Algorithm: Structured Workflow Process Mining through Amalgamating Temporal Workcases," *Lecture Notes in Artificial Intelligence*, Vol. 4426, pp. 119-130, 2007. Article (CrossRef Link)

[20] Michael zur Muehlen, Jorg Becker, "Workflow Process Definition Language - Development and Direction of a Meta-Language for Workflow Processes," *Proceedings of the 1st KnowTechForum*, 1999.

[21] Object Management Group, "Business Process Model and Notation – Version 2.0," *OMG Document Number formal/2011-01-03*, 2011.

[22] Workflow Management Coalition Specification Document, "Workflow Process Definition Interface -- XML Process Definition Language," *Document Number WFMC-TC-1025, Document Status –1.0 Final Draft*, October 2002.

[23] M. Jansen-Vullers and M. Netjes, "Business Process Simulation - a Tool Survey," in *Proc. of the Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus, Denmark, Information Systems IE&IS, University of Aarhus, pp. 77-96, 2006. Article (CrossRefLink)

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

[25] M. Westergaard, "BRITNeY Suite : Experimental Test-bed for New Features for CPN Tools," *Lecture Notes in Computer Science*, Vol. 4024, pp. 431-440, 2006. Article (CrossRef Link)

[26] van Hee, K., Oanea, O., Post, R., "Yasper: a tool for workflow modeling and analysis," in *Proc. of the 6th International Conference on Application of Concurrency to System Design*, pp. 279-282, 2006. Article (CrossRef Link)

[27] H. M.W. Verbeek, T. Basten, and W. M. P. van der Aalst, "Diagnosing Workflow Processes Using Woflan," *The Computer Journal*, pp. 246–279, 2001.

[28] Kyoung-Sook Kim, et al, "An Experimental Mining and Analytics for Discovering Proportional Process Patterns from Workflow Enactment Event Logs," in *Proc. of the International Conference on Big Data Technology and Applications*, pp. 61-70, Exeter, Great Britain, September 4-5, 2018. Article (CrossRef Link)

[29] Workflow Management Coalition Specification Document, "Workflow Management Application Programming Interface (Interface 2&3) Specification," *Version 2.0e, Document Number: WFMC TC-1009*, July 1998.

[30] Workflow Management Coalition Specification Document, "Workflow Standard - Interoperability Abstract Specification," Version 1.0, Document Number: WFMC-TC-1012, October 1996.

[31] Workflow Management Coalition Specification Document, "Workflow Management Coalition Audit Data Specification," Version 1.1, Document Number: WFMC-TC-1015, September 1998.

[32] Workflow Management Coalition Specification Document, "The Workflow Reference Model," Version 1.1, November 1994.

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