Data-Flow Modeling: A Survey of Issues and Approaches

Cristina-Claudia DOLEAN, Răzvan PETRUŞEL
Business Information Systems Department,
Faculty of Economics and Business Administration
Babeş-Bolyai University, Cluj-Napoca, Romania
cristina.dolean@econ.ubbcluj.ro, razvan.petrusel@econ.ubbcluj.ro

This paper presents a survey of previous research on modeling the data flow perspective of business processes. When it comes to modeling and analyzing business process models the current research focuses on control flow modeling (i.e. the activities of the process) and very little attention is paid to the data-flow perspective. But data is essential in a process. In order to execute a workflow, the tasks need data. Without data or without data available on time, the control flow cannot be executed. For some time, various researchers tried to investigate the data flow perspective of process models or to combine the control and data flow in one model. This paper surveys those approaches. We conclude that there is no model showing a clear data flow perspective focusing on how data changes during a process execution. The literature offers some similar approaches ranging from data modeling using elements from relational database domain, going through process model verification and ending with elements related to Web Services.
Keywords: *Data Flow, Process, Workflow*

1 Introduction

Nowadays, more and more information systems are process-centric. That is, the trend is to create and configure information systems focused on processes. But it wasn't like this all the time. The 1970s and 1980s were flooded by data-driven approaches [1]. This led to the development of data-centric information systems. Then the trend shifted and the era of user interface centric applications began. Process driven approaches appeared at the beginning of 1990s. Since then, the focus was on analyzing the control flow view of the process, to evaluate the order of tasks during a workflow execution, or to model and extract models from system logs. This type of analysis did not involve data or if it did, data was only reminded as being part of the control flow without being analyzed in detail. Considering this, we argue that data flow perspective is outside the aim of most workflow/process researchers and that they focus on analyzing the control-flow perspective [2], [3], [4], [5], [6], [7], [8]. Existing data mining techniques are too data-centric [4], while the research made in the process mining field emphasizes the control-flow perspective. This paper tries to argue that a new approach that balances between those extremes is needed. This is because running a process (i.e. executing the control flow) requires tasks to be enabled or disabled and this is done at the data level. If data is missing or is not available when is needed, the entire execution of the workflow ends. Some data is available at the beginning of the workflow, but there is also data which is generated during the execution of the workflow, after a specific task is executed. Therefore, we are dealing with input and output data elements. Each activity is characterized by a set of input data elements, respectively a set of output data elements.

Real data from enterprises is represented in an abstract way and stored in databases (see Figure 1). Entity Relationship Diagram (ERD) offers an abstract view of data in order to depict a database. An ERD is created based on data and uses abstract concepts. On the other hand, the analysis of system log data using process mining techniques and methods enables a process model to be automatically extracted. Based on the process model, modeling experts may improve the model with the data flow perspective. Important sources of knowledge for an expert are the ERD concepts and elements from the software design documentation. Once the data flow is modeled, this needs to be verified if it is conformant with the analyzed data.

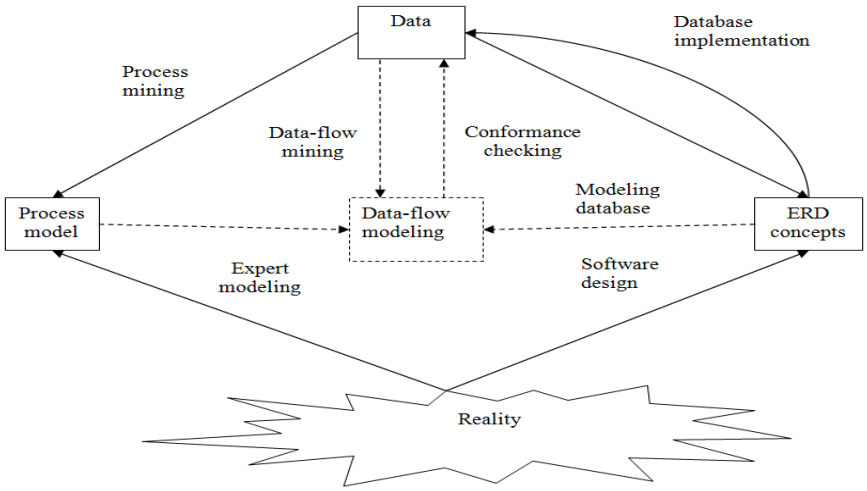

Fig. 1. Improving a process model with the Data flow perspective

2 Related Work

The literature offers different ways to model the dataflow perspective of a process, but: a) none of them provides a step by step view over the data transformation during the workflow execution or b) not all approaches refer on discovering data model from event logs. This section reviews all previous research that approached the problem of modeling data flows and, more specifically, the process data perspective.

The most research done in business process modeling area focused on the control flow perspective. It analyses the sequence of the tasks: which task must be executed and when. A task represents an abstract logical unit of work which may be executed for many cases, for example "send invoice" [9]. Tasks are executed by resources with certain roles and organized in groups. The resources are actors able to execute activities and they can be human beings or not. The case dimension is formed by the process instances (a series of tasks). The work item resulted from control-flow dimension and the case dimension represents a concrete piece of work which may be executed for a certain case. If at these two we add resources we get a complete piece of work being executed, the execution of a task for a specific case. Work items and activities are task instances.

Figure 2 shows the interest shown on data analysis during time. Data analysis started with the study of Data Structured Diagrams in 1969. Not long after this, the concept of Entity Relationship Diagram (ERD) was defined. Then the process-aware systems made their presence felt, new approaches focused on data have been developed. In order to analyze the data-flow of a process, researchers tried to provide a standalone data-flow view or they combined the control-flow with data-flow. In this context, data validation methods [10] are provided and some data-flow errors [11] that can occur in the data-flow are identified.

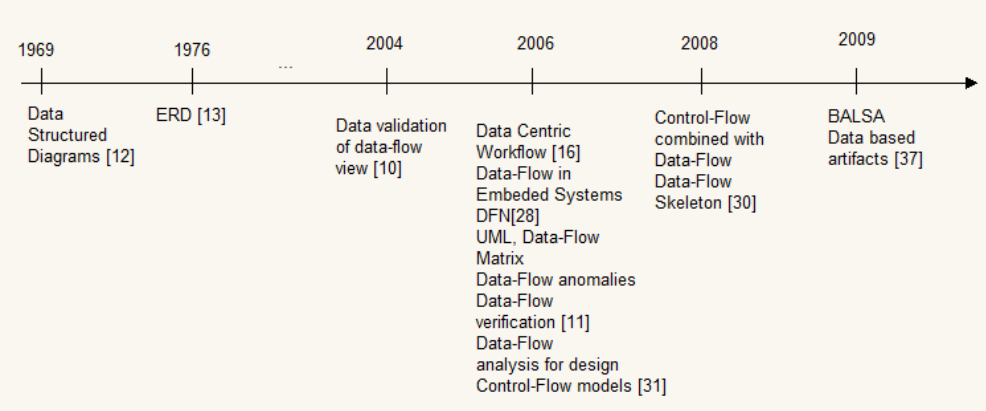

Fig. 2. Data analysis evolution

2.1 Data Modeling

Data modeling refers to analyze data-oriented structures. First of all, in relational databases area, the model widely known and used for depicting the data elements and their interaction is the Entity Relationship (ER) diagram [12], [13]. Such a model shows entities, their attributes and the relationships between entities. The ER diagrams are the standard in modeling relational databases. We assume that the enterprise software uses data stored in relational databases. But, it is obvious that creating an ER diagram from a multiple-trace log and using it in connection to process models is unfeasible.

An UML state diagram is depicted as a directed graph where the nodes represent the states and the hyperarcs represent the state transitions. In this sense, UML integrated the UML state machine defined by Harel [14]. With respect to object-oriented design, a data element can be seen as an object. Therefore it can be assumed that it has some states (a UML State Diagram [15] can be used to model the life-cycle of the data element), properties and relationships with other objects (an UML Class/Object Diagram may be used). Again, using those diagrams in the workflow context is unfeasible.

2.2 Data Usage Analysis

Another technique is based on **data usage analysis**. The Data Flow Diagram (DFD) is a popular model since it was first proposed, in the 70s. There are many objections to using it in the workflow context. A similar approach, based on DFD is presented in [16]. The authors propose a solution for Data Centric Workflow (modified DFD) based on the existing standards of Web Service Resource Framework (WS-RF) specification. The aim is to identify the initial and final data, but also intermediate data from a process (data flow modeling) using web services and metadata. First step is to identify data definitions with respect to the application. Web Services offer a clear separation of input and output data.

WS-RF contains four specifications: WS-Resource Properties, WS-Resource Lifetime, WS-Service Group and WS-Base Faults [16]. WS-Resource Properties refers to how WS-Resources are depicted by XML documents. WS-Resource Lifetime defines methods for destroy WS-Resources. WS-Service Group depicts the way how collections of Web Services can be represented and managed. WS-Base Faults defines a standard exception reporting format.

Every Web Service having its own data model can be seen as a workflow: the XML schema includes the information needed to extract the workflow. The resources are shared between web services. Basically, the data model was encapsulated as a WS-Resource. The Data-Centric workflow model focuses on data types and data flow without emphasize the characteristics of each service separately. The drawback of this approach is that it does not provide a visualization of the resulted model.

2.3 Product-Based Workflow Design

Product-based workflow design offers a new view in the context of modeling data. Van der Aalst [1] showed that the Bill-of-Material [17] can be used to generate a workflow process definition. The concept of Bill-of-Material depicts how a product is manufactured, the raw materials needed for getting the end product and it has a tree-like structure. In [17] the definition of Bill-Of-Material has been extended with options and choices. Therefore, the Bill-of-Material of an insurance policy has been depicted. The elements of the insurance policy as Bill-of-Material (see Figure 2) are: customer data, insurance data, insurance policy, medical report, historical data, personal data, standard rates, custom rates and risk data. The end product (insurance policy) is represented as the root element of the tree and the other elements are represented as leafs. This approach offers the data-centric view of the process while the Petri net description provides the control-flow view of the process.

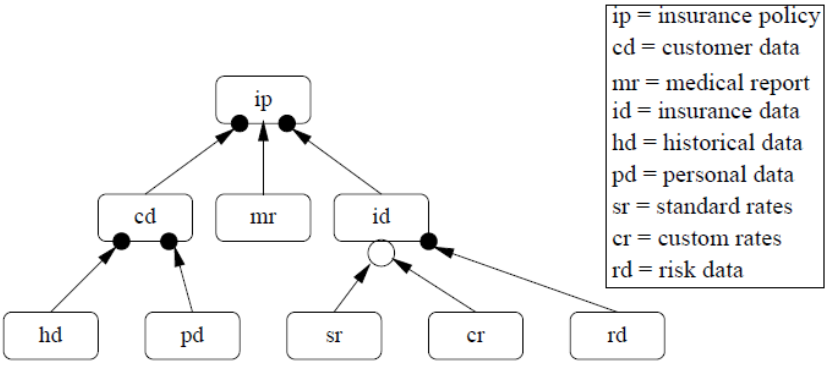

Fig. 3. The bill-of-materials of an insurance policy [1]

The idea of this approach is to model some information product (e.g. a decision to grant a mortgage or not) as a physical product composed of different parts (e.g. a car composed of chassis, wheels and seats) [1], [18]. In a Product Data Model (PDM), the parts of the final product, which is the root of the model, are the various data items that are used in order to arrive to it. Such a model has its own syntax and semantics. This approach wasn't integrated with the activities of the process and there are no efforts towards mining it from logs (it was assumed it can be created by experts).

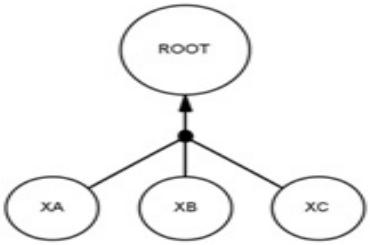

Fig. 4. Product Data Model Structure

The data-centric approach using the concept of Bill-of-Material has been developed in [18] by introducing the Product Data Model (PDM, see Figure 4) concept which put the basis of the Product-Based Workflow Design (PDWD [19]) methodology.

2.4 Data Verification

A lot of research has been done in control-flow verification area and less attention has been given to the data-flow analysis. Thus there are several detection methods and techniques designed to discover control-flow errors in workflow designs [3],[20], [21] and implemented in tools (Woflan [22],[23],[24]), while the data-flow verification assumes a detailed analysis of data dependencies [25].

Control flow verification refers to deadlocks, livelocks, soundness, interminable looping, and synchronization. A workflow process is sound if it fulfills the following conditions: option to complete, proper completion and no dead tasks [26]. The literature shows a clear separation between control and data flow analysis. Modeling data dependencies in an inappropriate way may cause errors in the control flow. [27] proposes a data-centric approach in order to find deadlocks from a business process. They consider the business process being modeled as a Petri Net. The reachability graph of the Petri Net helps to detect the deadlocks in the control flow. One of the causes of deadlocks is using inappropriate guards. Adding a guard or replacing a guard by a more restrictive one may remove deadlocks. But this approach does not offer a proper model of data involved in a workflow execution; it only provides another method to verify deadlocks from a workflow using the data from the workflow.

The data-flow perspective was almost never considered in the research done in this dimension of process analysis. In [10] there were identified a series of potential data validation problems that can occur in data-flow view of a process: redundant data, lost data, missing data, mismatched data, inconsistent data, misdirected data and insufficient data. Moreover three data flow implementation models were defined: explicit data flow, implicit data flow through control flow and implicit data flow through process data store. In the first model, the data flow transitions are defined as part of the workflow model. Basically the data flow transitions model the data changes from one activity to another. The second model calls the control flow to cross data from one activity to

another. In the last model all the inputs and outputs of the activities are recorded in the process data store.

2.5 Control Flow Combined With Data Flow

Another idea was to integrate **data-flow into process models**. The idea behind this approach is to define an extension for a workflow model so that data-flow can be shown at the same time with the activities control-flow. A formal fundament was proposed first in connection with embedded-systems [28] and then applied to workflows in [29]. The proposal is based on Petri Net formalism but instead of just two elements, uses a structure with three building blocks: storage, reactive and transformational units. The authors propose an extended version of Dual Workflow Nets (combination of control flow, data flow and interactions between control flow and data flow). The upside of this approach is that it is a sound formal workflow modeling technique that models control and data (and the interaction of the two), and there is also some ground work on model verification. The downside is that the model is complex even for small running-examples and that understanding it requires very specialized knowledge (it is hard to follow and understand by persons familiar with the Petri-Net notation and extremely difficult to understand by business persons). Also, compared to our proposal, there is no indication about how this model can be created (it is assumed that it exists or can be, somehow, designed by analysts).
Less formal approaches, that aim to integrate control-flow with data, use extended Petri Nets [25], UML Activity Diagram [11] or a dedicated data model [10]. In first two papers, the data view is simply an annotation of each activity in the control-flow model with the data that is read, written or deleted by that particular activity. The data dependencies are not modeled at semantic or syntactic level.
[30] proposes an approach called Data-Flow Skeleton Filled with Activities (DFSFA). Basically, the workflow process is derived from the data-flow skeleton and then is filled with activities. First, a data-flow dependency tree is generated based on the data dependency. Next step is to generate the data-flow skeleton and then fills it with activities. The main difference is that we take into consideration event logs generated by a decision-aware information system and automatically produce the data dependency by classifying the data elements into input and/or output data items, while in [30] and in [31], the data dependency is pulled out by analyzing the semantics. In other words, the input and output data are known for the process designer from the beginning. [30] also propose logical operators (rules) between data: and-rule, or-rule, and combined law, or combined law. "And combined law" refers to: $P = (P0 \wedge P1) \vee (P0 \wedge P2) \Rightarrow P = P0 \vee (P1 \wedge P2)$, meanwhile „or combined law" refers to $P = (P0 \vee P1) \wedge (P0 \vee P2) \Rightarrow P = P0 \wedge (P1 \vee P2)$.
A similar approach to [30] was adopted using metagraphs and document-driven workflows [32], [33], [34]: the control-flow is derived from the dataflow. Basically using the document-driven approach, the execution of the workflow is ordered by input documents.
Metagraphs represent extensions of directed graphs and hypergraphs. Initially they were used to model the interaction between Decision Support Systems' (DSSs) elements: stored data, decision models and expert knowledge. A metagraph has three types of elements [34]: atomic data items (for example the loan amount or the monthly debt), invertices and outvertices of edges (documents – set of information elements, e.g. loan application) and tasks (workflow tasks, e.g. contracting a loan). We observe that such graphs use elements from document-driven workflows approach.
The processes can be modeled as conditional metagraphs, while workflows (because they are instantiation of a process for a set of specific data) can be represented as (unconditional) metagraphs. Every workflow can be modeled in different ways using several modeling tools. Usually the workflow perspectives were approached individually. Metagraphs want to integrate the three dimensions of workflows in a single model.
In Fig. **5** an investment process is represented as a conditional metagraph. There are 2 tasks (represented by edges) characterizing the workflow from e_1,e_2. Each task is executes by a specific resource. The informational data and some assumptions are represented as ellipsoidal shapes. There are four information elements: INV (identifier of the production facility proposed for investment), CAP (capacity of the proposed production facility), UTIL (utilization rate of the proposed production facility), REV (revenues resulting from the proposed production facility) and EXP (expenses resulting from the proposed production facility). Task e_1 is performed by operations analyst and contains information about the proposed production facility, while task e_2 is

performed by financial analyst team. The operation analyst examines the information stored in e_1 in order to determine the resulting production or service delivery capacity. On the other hand, the financial analysts determine the revenues and the expenses given the capacity and utilization of the proposed production facility.

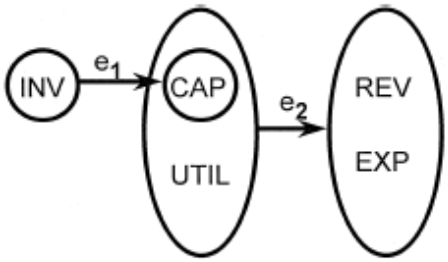

Fig. 5. Metagraph example [34]

The invertex of task e_1 is {INV} and the outvertex of e_1 is {CAP}. They can be associated with input and output data elements of a task. The simplest form of connectivity is represented by a path, e.g. the simple path connecting INV to REV is $<e_1,e_2>$. If the activities are not in sequence we speak about metapayhs. [34] also treat metagraphs types: conditional metagraph, dual metagraph or pseudodual metagraph. Dual metagraph is similar to Data Flow Diagram.

The organization of metagraph elements can be associated to Database Management System (DBMS). A metagraph dictionary contains the depiction of the information elements and its aim is to determine the documents which are invertices, respectively outvertices for each task (edge).

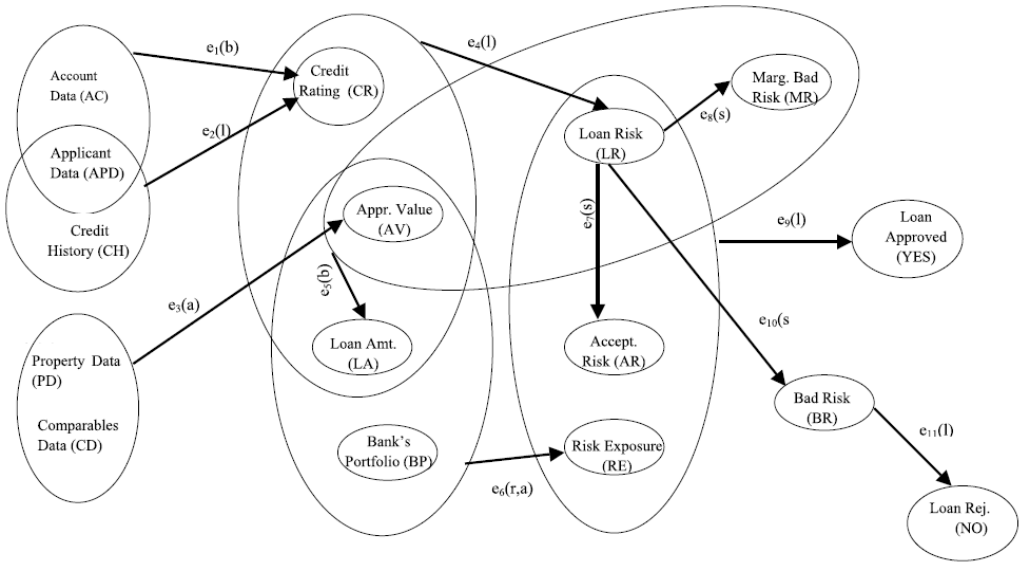

Fig. 6. Loan process represented as a metagraph [34]

Metagraphs are also used to model the information from DSSs: stored data, decision models and expert knowledge [34]. Concerning the stored data, the elements are represented by data attributes, the edges – data relations and the invertices, respectively outvertices – key and content attributes of the relations. The second type of information modeled refers to decision models. In this case, the elements represent the variables, the edges are the models and the invertices, respectively outvertices refer to input, respectively output variables. Finally, with respect to expert knowledge, the propositions refer to elements, the rules are the edges and the invertices, respectively outvertices represent the antecedents and consequents of the rules. Thus, metagraphs are used in order to integrate the different types of information from DSSs.

The visualization provided using metagraphs combines data elements and the tasks of workflows. Therefore it does not present a pure data-centric approach, but the real shortcoming appears when we are dealing with complex metagraph: they are difficult to be read, respectively to be analyzed (see Figure 6).

2.6 UML Activity Diagrams

In order to model a workflow using **UML Activity diagrams**, the model resulted must have a start task, respectively an end one. During the

workflow life cycle, a series of tasks and conditions are executed. Moreover, a bold black line refers to AND-Split, respectively AND-Join operators. On the other hand, the conditions are represented by diamonds; here we speak about XOR-Split, respectively XOR-Join operators. But workflows represented as UML Activity show only the control-flow perspective of the modeled process. UML Activity diagrams don't model complex (OR) splits and joins. A more complete approach is provided in [29], where a data-flow matrix and various relations between data and activities are defined. Still, there is no data model provided. In all the approaches a-priori constructed models are assumed. In none of the papers in this category there are no validations against real company data.

2.7 Low-Event Logs

Mining low-level logs to create workflow models approach aims to use low-level data in order to extract some workflow model. However, existing research [36] aimed to cluster traces so that process mining algorithms can produce better results. This goal led to an algorithm that relies mostly on the notion of log proximity. ERP systems generates low-level events. These events are different from the event generated by workflow systems because they are focusing on data. The mining of activities may provide the link between low-level event and high-level events. Thus, activity mining helps to discover the tacit processes existing on ERP systems. The methodology proposed in [36] emphasis the need of a process instance ID for each event. It does not only apply on event logs generated by ERP systems, but also on other systems which are not process-aware.

Clustering methods and techniques are use especially in data mining field and because we are looking for a data perspective of the process / workflow, these can be successfully applied. The aim is to find recurring patterns in the event log. In a similar way, data patterns can be sought in the event log and then aggregated.

2.8 Data-Flow Verification

Data-flow verification approach tries to improve process model verification by integrating checks on control and data flows. This is the actual goal of the [10], [11], [25] papers. In those papers, the authors define some errors like missing data, inconsistent data, redundant or conflicting data, and then check the control-flow activities that use this data. In [25] is argued that the control flow perspective has gained a lot of attention during time, whilst the data flow perspective was almost neglected. That is the reason why, the authors introduced the term of Workflow Data (WFD) nets. Basically a WFD is a formal business workflow with data which can be read, written or destroyed. There are some steps to convert a workflow net to a workflow net with data. First, the guards must be transformed into places (one for each value of guard), then in order to capture the parallel execution, the transitions have to be split (into start, respectively into end) and finally encode the assumption that writing to a data element can change its depending guards in any way.

A series of anti-patters concerning data flow are introduced in [25]: missing data, strongly redundant data, weakly redundant data, strongly lost data, weakly lost data, inconsistent data, never destroyed data, twice destroyed and not deleted on time.

Some of these are also reminded as data flow errors in [31]: missing data error, inconsistent data error and redundant data error. Actually dataflow anti-patterns represent dataflow errors which may slow, or worse, stop the execution of a case belonging to a workflow. Missing data anti-pattern refers to the data availability when a task is ready to be executed. There are situations when not all the data is available when is needed in order to execute an activity, for example a client of a certain hotel wants to do the check-out activity, but in the system is missing the number of nights he spent in the hotel. Thus, the check-out activity cannot be finalized until the number of nights is filled. Redundant data is classified by in [25] two categories: strong redundant data and weakly redundant data. [31] does not split this into two different approaches, associating the redundant data term with strong redundant data. If a task creates a data which is never read or it is destroyed before being read during the workflow execution it is considered strongly redundant. On the other hand, a data is weakly redundant if it is written and never read after. Inconsistency refers to the parallel execution of two or more tasks which are using (writing or destroying) the same data.

[31] proposes an algorithm to discover the dataflow errors mentioned above. In order to find data flow errors, the GTforDF (Graph Traversal for DataFlow) algorithm provide an approach that traverses the workflow to be analyzed in order to find all case instances. For every task of a case are defined the input and output sets. In

order to find dataflow errors an assumption is made: the concerned workflow must accomplish the control-flow verification methods [26]. A test is made to verify if each output data element is produced during the analyzed workflow case execution. If exist elements which were not produced during the analyzed workflow case execution, the workflow case execution ends. Then the AND-Joins and XOR-Joins operators are analyzed. Analyze of AND-Joins help to find inconsistent data and lost data, while XOR-joins support finding redundant data. In workflows containing no loops, the redundant data are identified at the end of the execution. Missing data is checked after execution of each task of the workflow instance.

2.9 Other Workflow Research Areas That Deal With Data

In most workflow modeling and simulation tools (e.g. YAWL, Protos, BPM|one, Staffware, IBM WebSphere, etc.) data needs to be defined for each activity. Typically, this is done by a matrix that shows the activities, the data elements and the operations on the data elements that are performed for each activity (i.e. read, write, update or delete). Clearly, this representation shows no overview to an expert and is not executable.

2.10 Artifact-Centric Process Modeling

This alternative approach tackles the problem where multiple cases of the same process overlap and synchronize. For example, in reality there are many situations where multiple orders are dispatched using just one invoice. Classic process mining approach assumes that each case is isolated. Artifact-centric modeling relies on the use of Proclets [36]. Even if Proclets solve the many to many modeling problem, this approach is tailored for dealing with activities and not with data.

[37] depicts a methodology (BALSA – Business Artifacts with Lifecycle, Service and Associations) based on data, rather than control flow for business process and workflows that focuses on artifacts, highlighting the relevant data, the lifecycle and the associations. The format of artifacts is depicted using the Entity Relationship data model. An artifact-centric workflow model has the following elements: business artifact information model, business artifact macro-level lifecycle, services (tasks), and the association of services to business artifacts.

The result is an artifact-centric workflow model that merges the lifecycles of key business entities and the information model. It can be used for business process design, but it does not have a graphical visualization, being more declarative. The methodology has three levels: Business Operation Model (BOM), the conceptual workflow and the operational workflow. An artifact can be monitored during the workflow execution and it uses attributes to store data needed for the workflow execution. The first step is to identify the business artifacts (business entities involved in the workflow). Next, the data involved in each artifact is needed to be identified. The link between artifacts and services need to be identified as well. These will be depicted in a declarative way. Here we speak about the business operations modeling. Last step is to provide a procedural specification of the declarative model – conceptual workflow design. This can be mapped into a physical implementation.

The artifact data refers to data produced or received during the business process execution. The dataflow can be found out before knowing the control-flow. In [31] is proposed an approach for deriving activity relations from a data-flow model. In [31] are defined the basic dataflow concepts like: data dependency, conditional routing constraint, activity dependency. These notions are valid in the context of our research; therefore we will take a closer look at them.

Data Dependency: Activity v_i depends on a set of input data I_{vi} to produce a set of output data O_{vi}, which is referred to as the data dependency for v_i and is denoted as $\lambda_{vi}(I_{vi},O_{vi})$.

The data dependency is classified in three categories: mandatory ($\lambda^{m}_{vi}(I^{m}_{vi},O_{vi})$), conditional ($\lambda^{c}_{vi}(I^{c}_{vi},O_{vi})$) and execution ($\lambda^{e}_{vi}(I^{e}_{vi},O_{vi})$) data dependency.

Conditional Routing Constraint: A conditional routing constraint c specifies that when a condition clause $f(D)$ is evaluated to be true, a set of activities V will be executed, denoted as $c=f(D):Execute(V)$, where D is a set of data items and $f(D)$ is a logic expression on D.

Activity Dependency: Given two business activities v_i and v_j, vi is dependent on v_j, denoted as $v_j \Rightarrow v_i$, if there exists a data item d such that $d \in Ov_j$, $d \in I_{vi}$, and $d \notin E$, where O_{vj} is the output data set of v_j, I_{vi} is the input data set of v_i and $I_{vi}= I^{m}_{vi} \cup I^{c}_{vi} \cup I^{e}_{vi}$, and E is the set of data provided by some external resources at various steps in the workflow. Activity dependency follows the transitive law, i.e., if $v_x \Rightarrow v_i$ and $v_j \Rightarrow v_x$, then $v_j \Rightarrow v_i$.

If there is no activity dependency between two activities v_i and v_j, we denote the non-dependency between the two activities as $v_i \infty v_j$. Further, if $d \in I^m_{vi}$ and $d \in O_{vj}$, v_i has a mandatory dependency on v_j, denoted as $v_j \Rightarrow_m v_i$. If $d \in I^c_{vi}$ and $d \in O_{vj}$, v_i has a conditional dependency on v_j, denoted as $v_j \Rightarrow_c v_i$. If $d \in I^e_{vi}$ and $d \in O_{vj}$, v_i has an execution dependency on v_j, denoted as $v_j \Rightarrow_e v_i$.

3 Proposal

The literature does not provide a concrete solution in order to create a data model linked to a certain process. Our aim is to offer a data-flow model of a (business) process, taking into account the data changes during the process execution. The main assumption in process mining is that there are differences between the desired model (the one created by managers) and the actual process performed by employees. The field is now mature and mining a process model is possible using several existing techniques and software. However, everything is based on an activity log. But sometimes the right format is unavailable [38]. What is always available is data stored in the relational database of enterprise software. We aim to build a model that is either stand-alone and shows a data-centric process model or is a complement to the process model and shows its data perspective upgraded with executable semantics.

The control flow perspective of the process of getting the approval to go in an international or national mobility is represented in Fig. 7. This means that the order of activities is analyzed, but it does not provide a clear transformation of data form one activity to another or from first activity of the model until the last activity is executed.

Here, the aim is to find a complementary model to the control-flow view of this specific process. The challenge is to depict in a graphical way the existing data from an event log and the dependencies between them.

First of all, the document must be filled with information about the date of departure, the date of arrival, the amount for the international or national mobility, the holder sign and the city for the international or national mobility (*Prepare document* and *Generate initial document* activities). Next, the holder needs a series of signatures: from Project Manager of the project that will support the financing, from the Head of Department where the holder belongs and the Dean of the faculty where the holder belongs (activities *Sign Project Manager*, *Sign Head of Department*, *Sign Dean*).

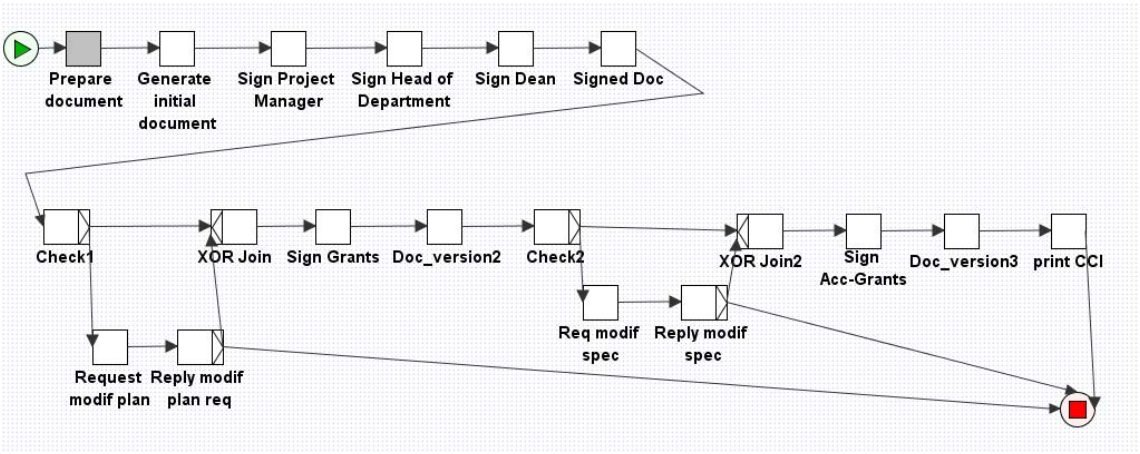

Fig. 7. Control-flow perspective of the process

Once the holder gets all the signatures (*Signed Doc* activity), the document must be transferred to *Grants Department*. Primarily, a first check (*Check1* activity) occurs in order to find out if the mobility was budgeted in the Research Plan of the project. If the answer is a positive one, the *Grants Department* approves the mobility (*Sign Grants* task), a second version of the document is issued (*Dec_version2* activity) and a second verification (*Check2* activity) takes place to see if there are sufficient funds. This activity is performed by qualified employees from *Accounting-Grants Department*. In case the activity was not budgeted in the research plan, a change request of the Research Plan must be submitted to *Grants Department* (*Request modified plan activity*) in order to be approved. On the one hand, if the modification request is

approved, the document is sent to *Accounting-Grants Department* for signature (*Sign Acc-Grants* activity). Otherwise, the activity of mobility is rejected. Further, if *Accounting-Grants* department considers that there are not sufficient funds for the mobility, a request of specifications change must be filled (*Req modif spec* activity). If the request is denied, the mobility request is rejected. On the other hand, if the request of specifications change receives a positive reply, the *Accounting-Grants Department* signs the mobility document (*Sign Acc-Grants* activity). This version of the document is sent to the Center for International Cooperation where is printed (*print CCI* activity).

In order to represent the data perspective of the workflow we propose an extended version of Product Data Model. A PDM can be manually generated based on interviews, questionnaires or by using 'report while doing' approaches, but we offer an automatically methodology for discovering PDMs from event logs.

The workflow of a process contains a series of tasks. Based on the workflow settings and on the condition assigned to each task, a certain task may be activated or not during the workflow execution. In order to create the data model of the process, usually, each event from the event log corresponds to an operation from the PDM. The data elements produced, respectively consumed after the execution of a task are the data model elements.

For a better understanding, we will consider the event logs generated after two executions of the workflow. These are depicted below. All workflow executions contain the document signed (produced by *Signed Doc* task) because before the concerned task there are not XOR Split, OR Split or AND Split tasks. Thus, the frequency of the operation having *Initial Doc* element as data output increases in the aggregated Product Data Model (see Figure 7, e.g. the initial document element has frequency two because it appears in the both cases of the event log).

In order to activate the execution of *Sign Grants* task there are two possible ways to do this, depending on *Check1* task result. Thus, there are different data elements used (produced or consumed) in order to execute a certain task. For example, for the first case, we assume that the mobility was budgeted in the Research Plan (*prev_act* data element from Check1 task is true). Thus, the document immediately receives the Grants Department signature. The second version of the document (*Doc_version2*) is obtained based on the *Signed Doc* and *Grants sign* data elements.

For the second case, we assume that the mobility was not budgeted in the Research Plan, therefore, a change request plan (*Request modif Plan* task) must be approved (*modif_plan_approve* data element from *Reply modif plan req* task) before authorized staff from *Grants Department* signs the mobility document. Thus, the second version of the document (*Doc_version2*) is obtained based on the *Signed Doc*, *Grants sign* and *Modif Plan Doc* data elements.

In this case, the frequency of the operations having *Doc V2* element as data output remains 1 for each operation in the aggregated Product Data Model (see Figure 7, the operations which generate *Doc V2* data element, each have frequency 1 because one of them is executed for the first case and the other one for the second case). The aggregated PDM represents the data model resulted from the entire event log generated after the workflow execution.

In both cases considered, after the second version of the document is produced (*Doc V2*), the Accounting Grants approves the mobility and the final document (*Doc V3*) is printed at CCI.

As we mentioned before, there are activities which are composed of different inputs; while the output element is the same (e.g. getting the second version of the document after the authorized employees verified if the activity was budgeted in the Research Plan). As depicted above, the control-flow perspective depicts two ways of executing this activity: if the activity was budgeted in the Research Plan, the authorized person signs the document; otherwise a request of changing the Research Plan must be submitted to the Grants Department. If the request is approved, Grants Department signs the document. Therefore, on the one hand the initial document signed by the authorized persons from holder's faculty and by the authorized employee from Grants Department lead to the second version of the document. On the other hand, the second version of the document is obtained having elements from the first case and adding the request of modifying Research Plan approved by Grants Department. The aggregation is understandable: there are two operations whose result is the same, the second version of the document (see Figure 7).

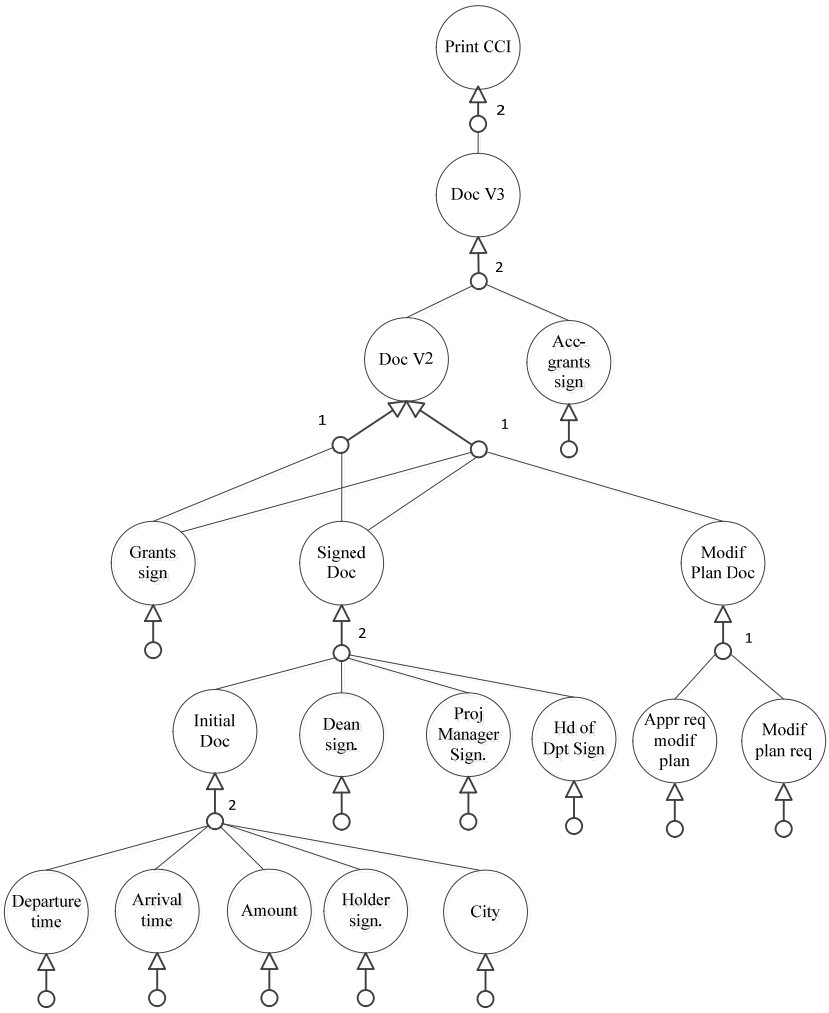

Fig. 8. Data Flow perspective of the process

The depiction of the workflow shows that the second version of the document can be produced in two ways: a) based on the signed document and Grants signature and b) based on the signed document, Grants signature and the changed Research Plan. In order to be approved, the last document needs a request of changing the Research Plan and its approval.

4 Conclusions

The paper reviews the approaches aimed at creating and exploiting a data flow view of a process. Some researchers tried to give either a "pure" data flow of the process or a combined data flow, mixing the data with the activities. But none of them offered a clear and understandable methodology or model of the process data perspective, focused on the data changes during the process execution. The closest to this goal are the approaches which look for data errors or data anti-patterns that can be encountered in a workflow.

Given the literature review in the paper we can conclude that not all the research done in order to model the data of a process use event logs as a starting point. Instead, many approaches assume that such a model is available or can be constructed manually by an expert.

The methodologies proposed so far don't use the event logs as source for mining the data model perspective. ERDs and DFD offer an abstract depiction of data stored in a database. The Data Centric Workflow combines DFD with Web Services and metadata, but it doesn't offer a visualization of the model. Metagraphs combines data flow, control flow and resource allocation, but don't provide an understandable model for complex processes. It was assumed that the PDM is created by experts and there is no automatically method in order to get it. Thus, the

literature doesn't give a methodology to automatically create the data perspective of a workflow.

Our approach discovers DDM(s) from event logs. It is based on several algorithms which offer a different data visualization of the workflow. This issue will be dealt with in future papers. We propose a different model in order to present the data perspective of a business process which should be intuitive, clear and more understandable than the other approaches presented before. The new data model visualization is based on the data consumed, respectively produced by each task (activity). The execution of a task is linked to an operation from the dataflow model. The data consumed by a task's execution is the set of data inputs of the operation, while the data resulted after the task's execution are the output data elements of the operation. In this way, the model depicts the data changes from one task to another. It also shows which operation must be executed in order to arrive to a certain data element (which data elements are needed to execute a specific operation). More details on the syntax, on the semantics of the model as well as its validation will be available in future papers. The main point of this paper was to show there isn't such a model available. Therefore, our approach is new and innovative.

Acknowledgement

This work was supported by CNCSIS-UEFISCSU, project number PN II – RU - TE 52/2010 code 292/2010.

This work was possible with the financial support of the Sectoral Operational Programme for Human Resources Development 2007-2013, co-financed by the European Social Fund, under the project number POSDRU/107/1.5/S/76841 with the title „Modern Doctoral Studies: Internationalization and Interdisciplinarity".

References

[1] W.M.P. van der Aalst, "On the Automatic Generation of Workflow Processes based on Product Structures", *Computers in Industry*, 39:97–111, 1999

[2] W.M.P. van der Aalst, "Woflan: a Petri-net-based workflow analyzer", *Syst. Anal. Model. Simul. 35, 3*, 345-357, 1999

[3] W.M.P. van der Aalst, "Workflow verification: Finding control-flow errors using petri-net based techniques", In *W.M. P. van der Aalst, J. Desel, and A. Oberweis, editors, Business Process Management: Models, Techniques, and Empirical Studies,* pages 161–183. Springer-Verlag, Berlin, 2000

[4] W.M.P. van der Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes", *Springer Verlag*, 2011 (ISBN 978-3-642-19344-6)

[5] W.M.P. van der Aalst, A.J.M.M. Weijters, Maruster L., "Workflow Mining: Discovering Process Models from Event Logs", *IEEE Transactions on Knowledge and Data Engineering,* 16(9):1128-1142, 2004

[6] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining Process Models from Workow Logs", In *I. Ramos G. Alonso H.-J. Schek, F. Saltor*, editor, Advances in Database Technology - EDBT'98: Sixth International Conference on Extending Database Technology, volume 1377 of Lecture Notes in Computer Science, pages 469-483, 1998

[7] J.E. Cook, A.L. Wolf, "Discovering Models of Software Processes from Event-Based Data", *ACM Transactions on Software Engineering and Methodology*, 7(3):215-249, 1998

[8] A.K.A. de Medeiros, B.F. van Dongen, W.M.P. van der Aalst, A.J.M.M. Weijters, „Process Mining: Extending the α-algorithm to Mine Short Loops", *BETA Working Paper Series*, WP 113, Eindhoven University of Technology, Eindhoven, 2004

[9] W. M. P. van der Aalst, M. Weske, and D. Grűnbauer, „Case handling: a new paradigm for business process support" *Data and Knowledge Engineering, 53(2):*129-162, 2005

[10] S.W. Sadiq, M.E. Orlowska, W.Sadiq, C. Foulger, „Data Flow and Validation in Workflow Modelling", *In Fifteenth Australasian Database Conference (ADC)*, Dunedin, New Zealand, volume 27 of CRPIT, pages 207-214, Australian Computer Society, 2004

[11] S.X. Sun, J.L. Zhao, J.F. Nunamaker, O.R. Liu Sheng, "Formulating the Data Flow Perspective for Business Process Management", *Information Systems Research*, 17(4), pages 374-391, 2006

[12] C.W. Bachman, "Data Structure Diagrams", in: *DataBase: A Quarterly*

Newsletter of SIGBDP, vol. 1, no. 2, Summer 1969

[13] P.P.-S. Chen, "The entity-relationship model toward a unified view of data", *ACM Transaction Database System* 1, 1 pages 9-36, 1976

[14] D. Harel, "Statecharts: A visual formalism for complex systems", Science of Computer Programming , Volume 8 Issue 3, pages 231 -274, 1987

[15] OMG. 2005. Unified Modeling Language: Superstructure, Version 2.0, formal/2005-07-04

[16] A. Akram, J. Kewley, R. Allan, "A Data Centric approach for Workflows", *Enterprise Distributed Object Computing Conference Workshops*, EDOCW '06. 10th IEEE International, 2006

[17] J.A. Orlicky, "Structuring the bill of materials for mrp", *Production and Inventory Management*, pages 19–42, 1972

[18] I. Vanderfeesten, "Product-Based Design and Support of Workflow Processes", *Eindhoven University of Technology*, Eindhoven, 2009

[19] I.T.P. Vanderfeesten, H.A. Reijers, W.M.P. van der Aalst, "Product-based Workflow Support", *J. Information Systems 36*, 517-535, 2011

[20] W.M.P. van der Aalst and K.M. van Hee, „Workfow Management: Models, Methods", and *Systems. MIT press*, Cambridge, MA, 2004

[21] W. Sadiq, M.E. Orlowska, „Analyzing Process Models using Graph Reduction Techniques", *Information Systems*, 25(2):117-134, 2000

[22] E. Verbeek , W. M. P. van der Aalst, "Woflan 2.0: a Petri-net-based workflow diagnosis tool", *In Proceedings of the 21st international conference on Application and theory of petri nets (ICATPN'00)*, Mogens Nielsen and Dan Simpson (Eds.), Springer-Verlag, Berlin, Heidelberg, pages 475-484, 2000

[23] H.M.W. Verbeek, T. Basten, W.M.P. van der Aalst, "Diagnosing Workflow Processes using Woflan", *Computing Science Report 99/02*, Eindhoven University of Technology, Eindhoven, The Netherlands, 1999

[24] W.M.P. van der Aalst, "Woflan: a Petri-net-based workflow analyzer", *Syst. Anal. Model. Simul.,* 35, 3, 345-357, 1999

[25] N. Trcka, W.M.P. van der Aalst, and N. Sidorova, "Data-Flow Anti-Patterns: Discovering Data-Flow Errors in Workflows", In P. van Eck, J. Gordijn, , and R. Wieringa, editors, *Advanced Information Systems Engineering, Proceedings of the 21st International Conference on Advanced Information Systems Engineering (CAiSE'09), volume 5565 of Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pages 425-439, 2009

[26] A. ter Hofstede, W. van der Aalst, M. Adams, N. Russell, "Modern Business Process Automation: YAWL and its Support Environment", ISBN: 978-3-642-03120-5, *Springer*, 2010

[27] C. Wagner, „A Data-Centric Approach to Deadlock Elimination in Business Processes", In *Daniel Eichhorn, Agnes Koschmider, and Huayu Zhang*, editors, Proceedings of the 3rd Central-European Workshop on Services and their Composition, ZEUS 2011, Karlsruhe, Germany, February 21-22, 2011, volume 705 of CEUR Workshop Proceedings, pages 104-111, 2011

[28] M. Varea, B.M. Al-Hashimi, L.A. Corte´s, P. Eles, Z. Peng, "Dual Flow Nets: Modeling the Control/Data-Flow Relation in Embedded System", *ACM Transactions on Embedded Computing Systems, 5(1)*, pages 54–81, 2006

[29] S. Fan, W. Dou, J. Chen, "Dual Workflow Nets: Mixed Control/Data-Flow Representation for Workflow Modeling and Verification", *K.C. Chang et al. (Eds.)*: APWeb/WAIM 2007 Ws, LNCS 4537, Springer-Verlag, Berlin, pages 433–444, 2007

[30] N. Du, Y. Liang, L. Zhao, "Data-flow skeleton filled with activities driven workflow design", in *Won Kim & Hyung-Jin Choi*, ed., 'ICUIMC' , ACM, pages 570-574, 2008

[31] S. X. Sun, J.L. Zhao, „Activity Relations: A Dataflow Approach to Workflow Design", proceedings of International Conference on Information Systems 2006, Milwaukee, Wisconsin, USA, Paper 44, 2006

[32] J. Wang, A. Kumar, „A framework for document-driven workflow systems", In *Proceedings of the 3rd International Conference on Business Process Management. Lecture Notes in Computer*

Science, vol. 3649, Springer Verlag, pages 285–301, 2005

[33] A. Basu, R.W. Blanning, "Metagraphs and Their Applications", *Integrated Series in Information Systems*, Senes, Springer, 2007

[34] A. Basu, R.W. Blanning, „Metagraphs in workflow support systems", *Decision Support Systems 25*, pages 199– 208, 1999

[35] C.W. Gűenther, W.M.P. van der Aalst, "Mining Activity Clusters from Low-Level Event Logs", *BETA Working Paper Series*, WP 165, Eindhoven University of Technology, Eindhoven, 2006

[36] D. Fahland, M. De Leoni, B. van Dongen, W.M.P. van der Aalst, "Behavioral Conformance of Artifact-Centric Process Models", In *A. Abramowicz et al. (eds)*, BIS 2011, LNBIP 87, Springer-Verlag, Berlin pages 37-49, 2011

[37] K. Bhattacharya, R. Hull, J. Su., „A Data-Centric Design Methodology for Business Processes", In J. Cardoso, & W. van der Aalst (Eds.), *Handbook of Research on Business Process Modeling,* Hershey, PA: Information Science Reference, pages 503-531, 2009

[38] W. van der Aalst, A. Adriansyah, A.K. Alves de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, et al., "Process Mining Manifesto" In Lecture Notes in Business Information Processing, 99, Berlin, Germany; New York, NY, USA: Springer, pages 169–194, 2012

Cristina-Claudia DOLEAN has graduated the Faculty of Economics and Business Administration, Babeş Bolyai University, Cluj-Napoca in 2008. She holds a bachelor degree in Business Informatics and a master degree in E-Business. She is currently a PhD student in the field of Business Informatics. Her current research interest include Process mining and Workflow Management.

Răzvan PETRUŞEL holds a Ph.D. in Cybernetics, Statistics and Business Informatics starting 2008. He started in 2003 as a full-time Ph.D. student at the Business Information Systems Department, Economical Sciences and Business Administration Faculty, in Babeş-Bolyai University of Cluj-Napoca. In 2007 he became an assistant professor and since 2009 he holds the current position as lecturer. His research is focused on DSS Specification, Modeling and Analysis; Process Mining; Workflow Management; and Decision Mining and Analysis.