Opus Framework: Research and Evaluation

Nahla Haddar, Mohamed Tmar, and Faiez Gargouri

***Abstract*—Existing Workflow Management Systems (WfMSs) provide promising perspectives to support business processes. However, numerous processes are still defined with hard-coded process logic. Accordingly, the resulting workflow applications are both complex to design and costly to maintain. The main reason of this problem is that many processes are data-driven; i.e., evolution of process instances depends on values of data elements. Therefore, process models have to be sufficiently integrated with the underlying data structure, in order to allow processes to manage the control flow and data. *Opus* system project offers a comprehensive association between data and processes, in order to provide not only a generic processes support, but also generic activity details become possible. This paper presents typical properties of existing data-driven WfMSs, which we collected after an exhaustive literature study, and it elaborates to what degree these systems are able to satisfy the problems addressed by our *Opus* project.**

***Index Terms*—Data-driven workflow, process modeling, process analysis and verification, data-driven execution of processes.**

I. Introduction

Workflows based on structured processes are generally centered on the activities performed during their execution. However, capturing business logic in activity-centric process models leads to a contradiction between how the process can be modeled and the preferred work practice [1]. In particular, there are many processes that are heavily related to knowledge (knowledge-intensive) and dynamism, and their instances progress is directly depending on the values of the available data. In this type of processes data are manipulated as objects each of which is an abstraction of a real entity of a specific domain (e.g., orders and bills in a business process, applications and interviews in a human resource process, etc.). Each object is represented by a set of attributes (e.g., the attribute describing the status of a bill payment, the name of an applicant in an application object, etc.), and plays a fundamental role in the deployment of the process (e.g., after the assessing of an application, the applicant is accepted for making an interview or not). Therefore, it is not sufficient to model processes only in terms of atomic activities to allow process models to be compliant with data objects. The identification of this need has guided the interests of researchers to data-driven processes.

Unlike activity-centric approaches, data-driven approaches allow a particular step in a process that is not directly depending on the completion of the previous steps, but rather changes in some attribute values related to an object.

Manuscript received August 2, 2014; revised April 22, 2015. This work was supported in part by the Laboratory of Multimedia Information Systems & Advanced Computing (MIRACL) of Sfax University, Tunisia.

Nahla Haddar is with the Faculty of Economic Science and Management, Sfax 3018, Tunisia (e-mail: nhaddar@ymail.com).

Mohamed Tmar and Faiez Gargouri are with the Higher Institute of Computer Science and Multimedia, Sfax 3021, Tunisia (e-mail: mohamed.tmar, faiez.gargourig@isimsf.rnu.tn).

Most of the approaches resulting from researches in the data-driven workflow management, such as [2]–[4], are inspired by the Petri nets (P-nets) formalism. However, most of them often provide default activities (called atomic or black box activities, e.g., Web services or executable artifacts) which are running non-specific tasks to individual needs of an organization [3], [5]. As a result, the resulting processes reflect only a macroscopic view on the real work, and there is a discrepancy between the way these processes can be defined and the preferred work practice [6]. Because of these limitations, most of business organizations use professional applications (e.g., Enterprise Resource Planning systems) instead of WfMSs. However, the resulting applications are both complex to design and expensive to deploy, in addition, the simplest changes in processes require hard code adaptation and expensive tests effort. Thus, our main challenge in *Opus* project was to allow process designers to model activities at the lowest level of granularity, taking into account their impact on the data transformation. So, this position paper discusses key challenges for *Opus* process management system in which processes, data objects and users are well integrated, in order to provide a data-driven execution of processes that are costumed to the management rules of an organization.

The remaining of the paper is organized as follows: we first motivate the problem addressed in our research in Section II. In Section III, we summarize the characteristic properties of existing data-driven WfMSs. Section IV, then, elucidates our *Opus* framework. Section V describes related researches along well defined evaluation schemas. Section VI concludes.

II. Problem Statement

Many approaches [2], [4], [7], [8] have been proposed to allow a comprehensive integration between data and process. However, the poor semantics of these approaches influence the ability of workflow engines to manipulate data during the process execution. Indeed, WfMSs, based on these approaches, focus on the control flow perspective, i.e., they include generic functions for assigning tasks to actors (i.e., creating worklists for each actor), notifying actors on the most urgent tasks, ensuring the data routing between actors (e.g., presenting data as forms), or mapping data so that they can be used by invoked applications. However, the detail of activities execution (i.e., tasks / elementary actions of activities) and data, in turn, are out of the control of existing WfMSs, which implies the use of specific programming and

DOI: 10.7763/IJIMT.2015.V6.606

the spending of more time and efforts as the modeling of the corresponding processes. Moreover, most of WfMSs often provide atomic activities which are running non-specific tasks to individual needs of an organization [3], [5], [9]; i.e., data are managed by the invoked applications themselves. Thus, the operational perspective of a process is limited to the invocation of applications and the mapping of their input/output data.

To address this limitation, we need a process modeling approach which enables the workflow designer to customize data-driven processes under the management rules required by an organization. This modeling approach must meet basic requirements, defined as follows:

1) Data objects are located above the definition of the underlying control flow [6], i.e., the definition of activities and routing rules must depend on data which ensures a data-driven execution of processes.
2) The definition of actors performing the work must be around the data manipulated by their activities. In fact, each actor needs data to perform his activities which produce other data needed by other actors.
3) The modeled processes must be analyzed to verify their correctness before their deployment. In fact, a process model that contains errors can lead to the accumulation of work, loss of time, production of unskilled services, etc. We are going to demonstrate in Section V that despite the abundance of the formal verification techniques, those defined for the data-driven processes are still incomplete.

We assume that a comprehensive integration between process and data promotes the efficacy and the productivity of the resulting workflows. This means, not only a generic process support, but also generic activity details become possible without needing for professional programmers for the missing process logic.

III. Emerged Data-Driven Workflow Systems

Nowadays, the emerged workflow systems are conscious by the need of providing a comprehensive integration between process and data. In this section, we are going to detail the features and the drawbacks of the most popular WfMSs that suitably support data, in order to deduce typical criteria allowing us to evaluate *Opus* system relatively to these systems.

PHILharmonicFlows [3], [4], [10], [11] manages data as inter-connected object types. Each type of object has a set of inter-related attributes. Basically, the behavior of an object, defined in terms of states and transitions, determines in what order and by whom the attributes of the object must be written, and what are the valid parameters. Thus, for each type of object a set of states must be defined. Each state implies the modification of certain values of specific attributes to an object type. Thus, the implementation of activities depends on the behavior of the processed object instances and their inter-relations.

Moreover, *PHILharmonicFlows* provides a set of consistency rules which assume that the process can be completed successfully if it does not contain deadlocks, and in cases where the deadlock occurs during execution, the system guides the underlying user to its recovery. In addition, the system allows for the automatic generation of forms from objects, and then during the execution of a particular activity, users can manipulate the forms corresponding to the same object instance simultaneously. In this context, a mechanism for controlling the simultaneous access to data is applied. Nevertheless, activities with advanced features, such as complex calculations, are classified by the system as black boxes that must be implemented by a programmer or executed by Web services. In this case, a data mapping is performed between the workflow data objects and the parameters of the invoked Web services.

Case Handling Paradigm WfMSs (such as *FLOWer* [2], *Staffware* [12] and *COSA Activity Manager* [13]): the Case Handling Pradigm (*CaseHP*) [2] is centered on the concept of *Case*. The *Case* is the manufactured *product*, e.g., the evaluation of a job application. Each *Case* is a collection of data objects and activities. Each activity has a life cycle based on finite state machines, and is interconnected with other activities by a directed *acyclic* graph; however, this is not a fundamental limitation. It is possible to model structured loops [2]. Besides, WfMSs based on the *CaseHP* can handle two types of activities: The first type presents the activities interacting with the Information System (IS), i.e., relational databases or XML documents, to import / save the data that are manipulated by authorized users by means of forms; the second type presents activities as Web services. Moreover, as the *CaseHP* is originally based on Business Artifacts (BAs) [14], the method proposed in [15] to verify the termination (i.e., the process is deadlocks and livelocks free) property of BAs models is adopted by the *CaseHP* WfMSs.

The main drawback of the *CaseHP* process models is that the definition of a *Case* can be associated with multiple roles involved in its processing, however, the approach does not define rules for managing competitive access to data, i.e., users can modify the data in the same *Case* simultaneously, without applying data access control, which affects the data consistency [8], [10].

ArtiFact system [8], [16] is developed to provide a graphical modeling and execution environment for Guard-Stage-Milestone (GSM) models. The GSM meta-model is recently proposed to specify the lifecycle of artifacts, by a more declarative way, with an operational semantics based on Event-Condition-Action rules.

Concurrency control mechanisms are not provided. But data consistency is ensured. Indeed, the approach assumes that two activities that write to the same data attribute can be executed in parallel, and values written by the first activity can be replaced when the second activity is finished. The schema of artifacts are saved in modeling step as XML schemas and their instances can be entered by users or imported at run-time from a relational database. The management of the artifacts life cycles is ensured by the invocation of Web services (WSDL and REST services), and the routing conditions (guards) must be written by users in JEXL or OCL languages. Accordingly, like most other WfMSs, *ArtiFact* is intended to support the management of activities, but not the details of their execution [8]. Finally, to verify the correctness of *ArtiFact* process models, two formal verification techniques have been proposed by [17], [18] to verify, respectively, their termination and boundedness properties.

Yet Another Workflow Language (YAWL) [7], [19], [20] extends the colored P-nets to provide a comprehensive support to present all the control flow patterns defined in [21], [22]. Furthermore, it suitably supports data by treating them through XML standards [23]. In fact, *YAWL* defines data types as XML schemas. Besides, at run-time, when a process requires data from an external environment, either a Web form is generated to enter data by the authorized user, or a Web service is invoked to provide the required data. In addition to the activities presented as Web services, *YAWL* can coordinate activities that read, write, apply a calculation expression, or compare data from XML documents.

Process designers using *YAWL* must have the necessary knowledge to use XML standards; because data types are defined by XML schemas, the calculation and the comparison expressions are defined by XQuery language, and the conditional and parallel routings are specified as XPath expressions. Nevertheless, the latest version of *YAWL* [24] simplifies the definition of data types by introducing a new language based on Pascal syntax, allowing for more easily defining data types that will be automatically transformed to/from XML schemas [25]. This version also enable the automatic generation of XQuery codes for querying XML data. To verify the correctness of *YAWL* process models, Wynn et al. [19], [20] propose a technique to verify the weak soundness property for *YAWL* models that contain OR-Join blocks or cancellation regions. This property relaxes the termination property, assuming that a process may, in some cases, not produce the desired results (e.g., due to the cancellation of its execution). For the other *YAWL* models the soundness property is decidable [20].

IV. *OPUS* WORKFLOW MANAGEMENT FRAMEWORK

Opus framework [1] is implemented according to the approach described in [26]–[29]. In this section, we are going to present an overview of *Opus* system features and drawbacks.

A. Opus Approach

Opus approach assumes that the decomposition of a process is related to the organizational criterion of an organization, so that an activity is performed by procedural roles. Each role defines a list of tasks related to the production of particular data. It can be human or system. Accordingly, a role is defined as a sub-process belonging to an overall workflow process. The data handled by the workflow can be entered by a role, imported from the IS (relational database or XML documents), or produced by other tasks. In particular, each role work is presented as a P-net in which its places are relational entities, called *Data Structures* (DSs), dedicated to support structured tokens (i.e., data *tuples*), and its transitions are *data operations* some of which are inspired from the relational algebra [26], [27].

The use of the relational algebra is due to its theoretical basis allowing for the automation of the data management. The other proposed operations have various utilities, such as writing the values of an attribute, performing complex calculations, presenting conditional and parallel routing, presenting iterative routing, invoking Web services, mapping of heterogeneous data, and integrating the process with the IS. Thanks to these operations our approach can present the details of most activities manipulating data, which leads to create workflow applications that really reflect the preferred work practice.

In order to maintain the data consistency during parallel writing in the same data element, our framework ensures that when one of the parallel operations wrote in an attribute of a DS, the other operations are blocked until the termination of the first one. Furthermore, we assured the modeling of a well-formed process through the definition of a method to verify a relaxed soundness property [28]. Particularly, our verification method ensures that there are no livelocks, deadlocks, or dead tasks in process models that contain many initial and final states.

Finally, in [29], we extended our meta-model by adding a multidimensional aspect to the relational data handled by our workflow models. Indeed, the extension allows generating multidimensional data models (OLAP hypercubes) from the DS instances. This extension allows to deduce statistical analysis data, and to calculate Key Performance Indicators relative to the improvement or deterioration in the performance of an essential activity for the success of a business, e.g., the average of monthly sales.

B. Opus System

Opus system consists of a number of components including a modeling editor, a workflow engine, a matching tool, a verification module, and a reporting tool.

Modeling and verification of processes: the modeling editor is equipped with a set of graphical interfaces to model workflow processes. Using this editor, the designer is not intended to know the formal aspect of the data operations. In fact, the editor provides all the necessary assistants to handle the designer during the modeling step. Besides, in case of changes in a role sub-process, the designer can easily update its sub-process model without damaging the other sub-processes. He can also use the verification tool to analyze his designed process models and verify their well-formedness [28] before their deployment.

Data-driven execution of processes: due to the formal definitions of the data operations, *Opus* engine can follow up the data flow routing, simulates the processing of operations, and automatically instantiates the resulting DSs of each operation. It is also equipped with the *IS Integration Module*. The latter uses the matching tool to map the data imported, saved, or updated in the IS.

The main drawback of the engine, is that it instantiates data to end-users using a simple JAVA tool, i.e., the *JTable* component that presents data as tuples and allows the authorized users to manipulate a DS attribute values. Thus, the implementation of the engine must be completed to ensure the generation of forms from the handled DS instances, which allows users to manipulate data through forms.

[1] All the features of *Opus* project, described in this section, are illustrated by the example of the *Order management process* provided in the following link: https://sites.google.com/site/wfmsopus/workflow-management-demo

Reporting tool [2]: the tool allows modeling complex business reports from the DS instances, as easily as possible, by providing drag and drop tools to organize the report objects. Data in the report are visualized as multidimensional tables or statistical graphs, and generated from the current DS instance, according to the dimensions and the fact defined graphically by business analysts. The main drawback of this tool is that the analyzed data source is limited to the DS instances generated by the workflow instances, so it must be improved to support more data sources (relational tables, XML documents, Excel tabular, etc.).

V. Related Work

On the one hand, *Opus* system provides comprehensive tools to manage data-driven workflows; on the other hand, the system offers an intelligent reporting tool. In this section, we illustrate related work in both data-driven workflow management and business intelligent fields in order to evaluate *Opus* system.

A. Modeling, Verifying, and Executing Data-Driven Workflows

Many meta-models and notations have been proposed to integrate data and processes. Techniques such as P-nets formalism [30] (with its various extensions such as hierarchy [31], color [32], objects [33], etc.), State Diagrams (SDs) [34], Event-driven Process Chains (EPCs) [35], UML Activity Diagrams (UML-ADs) [36], and Business Process Modeling Notation (BPMN) [37] are all considered useful to capture data over control flow. However, some techniques, such as P-nets, SDs, and BPMN, focus on a single perspective and require to be used in conjunction with one or more complementary formalism to provide a complete description of processes. That may cause possible erroneous interpretations when developing process models [6].

Indeed, the SDs describe the reactions of an object in response to events [38]. These diagrams have been extended in the latest version of UML [36] by the protocol state machines. The latter specify the sequences of operations that can be invoked on an object to define its life cycle. Despite this extension, the SDs are still focusing on the operational perspective, and should be used with other UML diagrams (*use case diagram* and *class diagram*) in order to introduce other aspects of processes. Furthermore, until now there is no standard that allows the mapping of SDs to executable workflow prototypes. Nonetheless, the BPMN specification includes a description of how a BPMN model can be converted to Business Process Execution Language (BPEL) [37]. However, this mapping focuses on the conversion of the control flow and neglects data, and therefore, it fails to specify the input value of an activity instance [39]. In addition, BPEL can only coordinate activities presented as Web services. Thus, the operational perspective is limited to the invocation of Web services, without being able to specify custom requirements associated to the process tasks.

The conceptual modeling languages (such as BPMN, EPCs, UML-ADs) are informal languages whose their semantics are not well defined and do not allow verifying process models [6]. Indeed, this verification is concerned with the determination, in advance, if a process model has certain desirable behavior. By performing this verification from modeling step, it is possible to identify potential problems, and if it is the case, the model can be modified before being deployed. The conceptual modeling languages perform syntactic basic controls, but they allow the modeling of processes with deadlocks and other anomalies [6]. Besides, the poor semantic of these languages influences the ability of workflow engines to automatically generate the operational behavior of processes from their graphical models, or to integrate the workflow with the existing IS [6]. Nevertheless, formal languages (such as P-nets and process algebra formalisms that include: Algebra of Communicating Processes [40], Communicating Sequential Processes [41], Calculus of Communicating Systems [42], etc.) are characterized by their rigorous semantics having a theoretical and mathematical basis for representing processes at a low level of abstraction.

Thanks to these features, these languages have the ability of analysis and automatic interpretation of the code running the workflow process. Compared to other formal languages, the P-nets formalism plays an important role in the process management area. Indeed, the P-nets formalism is the first formalism treating the competition in processes based on the notion of tokens. This is very important because in processes, many tasks can be triggered in parallel, and must be treated simultaneously. Therefore, most of notations and WfMSs adopt the P-nets semantics. Indeed, several attempts to formalize conceptual languages have been proposed, in which most of them are a mapping to P-nets or one of its extensions (e.g., BPMN [43], UML-ADs [44], [45], and EPCs [46]). However, these formalizations cover only some parts of processes, namely, data abstraction, expression of the OR-Join blocks, etc. Therefore, it is impossible to produce a correct semantics for all cases of process instances. Furthermore, due to its formal semantics despite its graphical nature, the P-nets formalism is widely used by most of the approaches that result from research in the data-driven workflow management area [2]–[4].

To discover the drawbacks of the WfMSs implemented according to these approaches, we performed a detailed investigation by an extensive literature study. We find that the integration of data with processes differs from a WfMS to another. Some WfMSs (e.g., *PHILharmonicFlows*) present data as objects and integrate them in the control flow by exploiting their life cycles presented as finite state machines. Some other systems (e.g., *ArtiFact*, *FLOWer*, *Staffware*, and *COSA*) extend the OLC approach by the use of BAs, to combine the informational and functional aspects of a process in a global unit (i.e., the artifact). And finally, we studied *YAWL* as the most emerged WfMS that extends the P-nets formalism by abstracting from data and presenting them as tokens. By detailing, in Section III, the features and the drawbacks of each of these systems, we extract a set of evaluation criteria that allows us to evaluate our system relatively to them.

[2] An example illustrating the functioning of this tool is provided via the following link: https://sites.google.com/site/wfmsopus/download

Evaluation Criteria

Systems	C1: parallel execution while keeping data consistency	C2: generating forms from data objects	C3: allowing users to make entries	C4: executing the details of activities	C5: interacting the workflow with the IS	C6: used by novice users	C7: data-driven execution of processes	C8: Process models verification
PHILharmonicFlows	+	+	+	-	-	±*a	+	Deadlocks
CaseHP WfMSs	-	+	+	-	+	+	+	Termination
ArtiFact	+	+	+	-	+*b	±*a,c	+	Termination, Boundness
YAWL	+	+	+	±*d	+	±*c	-*e	Soundness, Weak-soundness
Opus	+	-*	+	+	+	±*g	+	Well-formedness

Fig. 1. Evaluation of *Opus* system relatively to the existing data-driven WfMSs.

In summary, our systems analysis has revealed the following evaluation criteria:

- *C1*: the system supports the parallel execution of tasks manipulating the same data instance, while keeping data consistency;
- *C2*: the system presents data to end-users as forms;
- *C3*: The *system* allows users to make entries that will be assigned to the attributes of suitable data;
- *C4*: the *system* is able to execute the details of certain activities manipulating data (e.g., complex calculations, data transformation, etc.);
- *C5*: the system can directly interact the data handled by the workflow with those of the existing IS, without the intermediary of invoked applications or Web services;
- *C6*: the *system* can be used by novice users in programming languages or XML standards.
- *C7*: the *system* ensures a data-driven execution of processes, i.e., the definition of activities and routing rules depend on data;
- *C8*: the *system* is able to verify certain properties in its process models.

The signs in Fig. 1 denote the following meanings:
(+) supported criterion; (-) unsupported criterion;
(±) criterion partially supported; (a) black boxes activities may require an implementation by professional programmers [4]; (b) only for a relational databases; (c) conditional routing must be defined at a specific language (XPath expression for *YAWL* [24], OCL or JEXL languages for *ArtiFact* [8]); (d) only for complex calculations or comparison; (e) execution directed by control flow and data [47]; (f) *Opus* system presents data as relational table or as business analytic reports in PDF format[3]; (g) The designer must had the necessary competence to use the relational algebra.

[3] See https://sites.google.com/site/wfmsopus/home/reporting-tool

B. Reporting Tools and Generation of Multidimensional Data Models

Some authors, like [48]–[51], have used the *Entity Relationship (ER)* model to propose multidimensional data models. They present concepts of facts, measures, dimensions and their hierarchy of attributes. Other authors [52]–[57] have studied the same concepts relatively to the notion of the cube, or more generally, the hypercube. The two types of approaches provide high-level operations on the resulting data models, such as *Slice* (i.e., it displays a slice of a cube) and *Dice* (i.e., it displays distinct measures of a dimension) which are studied by [49], [50], [54]–[57], *Drill-down* [49], [50], [54]–[57] (i.e., it allows to navigate from less detailed data to more detailed data by stepping down a concept hierarchy for a dimension or introducing additional dimensions), Roll-up [49], [50], [52], [54]–[57] (i.e., it aggregates data by climbing up a concept hierarchy for a dimension), etc.

The main disadvantages of the studied approaches are:

- Some approaches [48], [50], [51], [53] do not support the definition of aggregation functions for Key Performance Indicators.
- Some authors [54], [56] did not provide graphical *reporting* tools to support their data models, however, the data model *Oracle Essbase* [57] has been implemented in the *Oracle Business Intelligence Foundation Suite (OracleBI)* system [58], and *SQL Server* [49] has been implemented in the *Microsoft Power BI (PowerBI)* [59].
- *OracleBI* simplifies the definition of KPIs by a graphical querying interface, however, for [49], [52], [55] the *definition* of KPIs are based on complex querying languages.
- The data models of some works [48]–[51], [57] are very close to the *star schema* [60]. The latter is presented by a central fact table and a graph per dimension to represent its attributes hierarchy. The links between the fact table and its dimensions are insured by means of the primary keys of each dimension table presented as foreign keys in the fact table. So, to generate a multidimensional data model, a large number of join operations is performed, leading to increase the time of querying processing and to influence the performance of the resulting reporting tool.

In summary our multidimensional data models evaluation is based on the following criteria:

- *C1:* the resulting *multidimensional* data models are different from the star schema;
- *C2:* the model *supports* the definition of KPIs;
- *C3:* the model *provides* high level operations (e.g., Slice, Dice, etc.);
- *C4:* the *model* is implemented in a graphical reporting tool;
- *C5:* the *developed* tool enables the graphical definition of KPIs;

We can deduce from Fig. 2 that unlike the proposed approaches in [48], [50], [51], [53], *Opus* approach allows for the definition of KPIs by applying aggregation functions on measures of the same hierarchical level, or those of the hierarchical level having the lowest granularity. Besides, relatively to approaches proposed in [49], [55], *Opus* provides a graphical tool to allow users to easily define all the measures of a hypercube. Finally, unlike the approaches in [48]–[51], [57], *Opus* multidimensional meta-model is quite different from the star schema, which decreases the time of the generation of multidimensional data models.

Approaches	Evaluation Criteria C1	C2	C3	C4	C5
Rizzi [51], Malinowski and Zimnyi [48]				×	
Ravat *et al.* [50]			×	×	
Microsoft SQL Server [49]		×	×	×	
Oracle Essbase 11.1.2.3 [57]		×	×	×	×
Cabibbo and Torlone [52]	×	×	×	×	
Tsois *et al.* [53]	×			×	
Kamble [54]	×	×	×		
Pardillo *et al.* [55]	×	×	×	×	
Ciferri *et al.* [56]	×	×	×		
Opus Reporting tool [29]	×	×		×	×

Fig. 2. Evaluation criteria supported by related work in business reporting solutions.

Indeed, *Opus* generates a hypercube from a single instance of DS that is already created during its underlying process execution. So, this generation does not require join operations that consume a lot in term of execution time. However, *Opus* multidimensional meta-model does not provide high level operations (e.g., slice, dice, etc.) to facilitate analysis. In addition, the model is limited to a single DS which limits the analyzed data sources to the DS instances produced by workflow instances or to relational tables or XML documents that, in turn, are transformed by the system to DSs.

VI. Conclusion

In *Opus* project we aim to develop generic concepts, methods and tools for ensuring a real data-driven execution of processes. In this paper, we report on our overall problem statement and the solution we apply. We also conducted extended literature study on the emerged data-driven WfMSs. Based on a detailed comparison of these systems we elicit the benefits of using our framework, and also, its missing features that must be completed. Currently, we are developing the missing module of forms generations. In future work, we elaborate more detailed issues in the context of our framework and apply it to real world processes for evaluation purpose (i.e., health care, human resource management, etc.).

References

[1] W. Sadiq, K. Schulz, M. Orlowska, and S. Sadiq, "When workflows will not deliver — The case of contradicting work practice," *International Proceedings of Business Information Systems*, 2005, pp. 69–84.

[2] W. V. D. Aalst, M. Weske, and D. Grunbauer, "Case handling: A new paradigm for business process support," *Data & Knowl. Eng.*, vol. 53, no. 2, pp. 129–162, 2005.

[3] V. Kunzle and M. Reichert, "Philharmonic flows: Towards a framework for object-aware process management," *Softw. Maint. and Evol.: Res. and Pract.*, vol. 23, no. 4, pp. 205–244, 2011.

[4] C. M. Chiao, V. Kunzle, and M. Reichert, "Integrated modeling of process- and data-centric software systems with Philharmonicflows," *Workshop on Communicating Business Process and Software Models, IEEE Computer Society*, 2013, pp. 1–10.

[5] N. Cerezo, J. Montagnat, and M. Blay-Fornarino, "Computer-assisted scientific workflow design," *Grid Comput.*, vol. 11, no. 3, pp. 585–612, 2013.

[6] W. V. D. Aalst, "Business process management: A comprehensive survey," *ISRN Softw. Eng.*, vol. 2013, 2013.

[7] W. V. D. Aalst and A. H. M. ter Hofstede, "Yawl: Yet another workflow language," *Inf. Syst.*, vol. 30, no. 4, pp. 245–275, 2005.

[8] E. Damaggio, R. Hull, and R. Vaculin, "On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles," *Inf. Syst.*, vol. 38, no. 4, pp. 561–584, 2013.

[9] B. Plale, E. C. Withana, C. Herath, K. Chandrasekar, and Y. Luo, "Effectiveness of hybrid workflow systems for computational science," *International Proceedings of Computational Science, ser. Procedia Computer Science*, vol. 9. Elsevier, 2012, pp. 508–517.

[10] C. M. Chiao, V. Kunzle, and M. Reichert, "Enhancing the case handling paradigm to support object-aware processes," *International Proceedings of Symposium on Data-driven Process Discovery and Analysis*, CEUR-WS.org, 2013, pp. 89–103.

[11] C. M. Chiao, V. Kunzle, K. Andrews, and M. Reichert, "A tool for supporting object-aware processes," *International Proceedings of Enterprise Distributed Object Computing Conference - Workshops and Demonstrations*, IEEE Computer Society Press, September 2014, pp. 410–413.

[12] *TIBCO Software Inc.*, "Tibco IProcess suite," 2009, version 11.0.

[13] *BPS-Solutions & COSA BV*, "Cosa BPM 5.7 product description," 2009.

[14] A. Nigam and N. Caswell, "Business artifacts: An approach to opera-tional specification," *IBM Syst.*, vol. 42, no. 3, pp. 428–445, 2003.

[15] A. Deutsch, R. Hull, F. Patrizi, and V. Vianu, "Automatic verification of data-centric business processes," *International Proceedings of Database Theory*, 2009, pp. 252–267.

[16] F. Heath, D. Boaz, M. Gupta, R. Vaculin, Y. Sun, R. Hull, and L. Limonad, "Barcelona: A design and runtime environment for declarative artifact-centric BPM," *International Proceedings of Service-Oriented Computing*. Springer, 2013, pp. 705–709.

[17] P. Gonzalez, A. Griesmayer, and A. Lomuscio, "Verifying gsm-based business artifacts," *International Proceedings of Web Services*, IEEE Computer Society, 2012, pp. 25–32.

[18] D. Solomakhin, M. Montali, S. Tessaris, and R. de Masellis, "Verification of artifact-centric systems: Decidability and modeling issues," *International Proceedings of Service-Oriented Computing*, 2013.

[19] M. Wynn, "Semantics, verification, and implementation of workflows with cancellation regions and or-joins," Ph.D. dissertation, Queensland Univ. of Technology, Brisbane, Australie, 2006.

[20] M. Wynn, W. V. D. Aalst, H. Verbeek, A. T. Hofstede, and D. Edmond, "Business process verification – Finally a reality!" *Bus. Process Manag.*, vol. 15, no. 1, pp. 74–92, 2009.

[21] B. Kiepuszewski, "Expressiveness and suitability of languages for control flow modeling in workflows," Ph.D. dissertation, Queensland Univ. of Technology, Brisbane, Australia, 2002.

[22] W. V. D. Aalst, A. T. Hofstede, B. Kiepuszewski, and A. Barros, "Workflow patterns," *Distrib. & Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003.

[23] C. Ouyang. (2005). How to manipulate data in yawl. Document version 0.4. [Online]. Available: http://yawlfoundation.org/yawldocs/YAWLDataManual-beta7.pdf

[24] YAWL Foundation. (2014). Yawl editor 3.0 (beta). [Online]. Available: http://www.yawlfoundation.org/pages/resources/yawl3.html

[25] M. Adams, "Editor 3.0: Redesigning the yawl user interface," *International Proceedings of YAWL Symposium*, Ceurws, 2013, pp. 56–61.

[26] N. Haddar, M. Tmar, and F. Gargouri, "A data-driven workflow based on structured tokens Petri net," *International Proceedings of Software Engineering Advances*, 2012, pp. 154–160.

[27] N. Haddar, M. Tmar, and F. Gargouri, "Implementation of a data-driven workflow management system," *International Proceedings of Computational Science and Engineering*, IEEE Computer Society, 2012, pp. 111–118.
[28] N. Haddar, M. Tmar, and F. Gargouri, "A framework for data-driven workflow management: modeling, verification and execution," *International Proceedings of Database and Expert Systems Applications*, ser. LNCS, vol. 8055. Springer, 2013, pp. 239–253.
[29] N. Haddar, M. Tmar, and F. Gargouri, "Generating analytic data models around workflow applications," *International Proceedings of Italian Symposium on Advanced Database Systems*, 2014, pp. 139–151.
[30] C. Petri, "Communications with automata," Ph.D. dissertation, Institut fur¨ instrumentelle Mathematik, Bonn, 1962.
[31] Y. Choo, "A structural petri net approach to concurrency," California Institute of Technology, Technical Report, 1982.
[32] K. Jensen, "Coloured petri nets," *Lecture Note in Compt. Sci.*, vol. 254, pp. 248–299, 1986.
[33] C. Sibertin-Blanc, "High level petri nets with data structure," presented at Work-shop on Application and Theory of Petri Nets, 1985.
[34] D. Harel, "Statecharts: A visual formalism for complex systems," *Sci. of Comput. Program.*, vol. 8, no. 3, pp. 231–274, 1987.
[35] G. Keller and T. Teufel, *Sap R/3 Process Oriented Implementation*, 1st ed., Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.
[36] OMG, "Omg unified modeling languageTM (OMG UML), superstructure version 2.4.1," Object Management Group, Technical Report formal/2011-08-06, 2011.
[37] OMG. (2011). Business process model and notation (BPMN) (version 2.0). Object Management Group, Tech. Rep. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/
[38] P. Roques, *UML 2 par la Pratique*, 5th ed., EYROLLES, 2006.
[39] J. Lenhard and G. Wirtz, "Detecting portability issues in model-driven bpel mappings," *International Proceedings of Software Engineering and Knowledge Engineering*, Boston, Massachusetts, USA, 2013.
[40] J. Bergstra and J. Klop, "Fixed point semantics in process algebra," Mathematical Centre, Amsterdam, Ulm, Germany, Technical Report, 1982.
[41] C. Hoare, "Communicating sequential processes," *Commun. of the ACM*, vol. 21, no. 8, pp. 666–677, 1978.
[42] R. Milner, *A Calculus of Communicating Systems*. Secaucus, NJ, USA: Springer New York, Inc., 1982.
[43] P. van Gorp and R. Dijkman, "A visual token-based formalization of BPMN 2.0 based on in-place transformations," *Inf. and Softw. Technol.*, vol. 55, no. 2, pp. 365–394, 2013.
[44] T. S. Staines, "Intuitive mapping of UML 2 Activity diagrams into fundamental modeling concept petri net diagrams and colored petri nets," in *International Proceedings of Engineering of Computer Based Systems*. IEEE Computer Society, 2008, pp. 191–200.
[45] E. Syriani and H. Ergin, "Operational semantics of uml activity diagram: An application in project management," in *International Proceedings of Model-Driven Requirements Engineering Workshop*, IEEE Computer Society, 2012, pp. 1–8.
[46] E. Kindler, "On the semantics of EPCs: Resolving the vicious circle," *Data & Knowl. Eng*., vol. 56, no. 1, pp. 23-40, 2006.
[47] A. Meyer, S. Smirnov, and M. Weske, "Data in business processes," des Hasso-Plattner-Instituts fur Softwaresystemtechnik an der Universitat Potsdam, Munchen, Technifcal Report, 2011.
[48] E. Malinowski and E. Zimnyi, Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications (*Data-Centric Systems and Applications*). Springer, 2008.
[49] Microsoft SQL Server, "Sql server in-memory OLTP internals overview for CTP1," 2013.
[50] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh, "Algebraic and graphic languages for olap manipulations," in *Strategic Advancements in Utilizing Data Mining and Warehousing Technologies: New Concepts and Developments*, IGI Global, 2008, pp. 60–90.
[51] S. Rizzi, "Conceptual modeling solutions for the data warehouse," in *International Proceedings of Database Technologies: Concepts, Methodologies, Tools, and Applications*, 2007, pp. 86–104.
[52] L. Cabibbo and R. Torlone, "A logical approach to multidimensional databases," in *International Proceedings of Advances in Database Technology*, ser. LCNS, vol. 1377. Springer, 1998, pp. 253–269.
[53] A. Tsois, N. Karayannidis, and T. Sellis, "Mac: Conceptual data modeling for OLAP," in *International Proceedings of Design and Management of Data Warehouses*, 2001, pp. 5–11.
[54] A. Kamble, "A conceptual model for multidimensional data," in *International Proceedings of Asia-Pacific conference on Conceptual Modeling - Volume 79*, 2008, pp. 29–38.
[55] J. Pardillo, J.-N. Mazon, and J. Trujillo, "Extending OCL for OLAP querying on conceptual multidimensional models of data warehouses," *Inf. Sci.*, vol. 180, no. 5, pp. 584–601, 2010.
[56] C. Ciferri, R. Ciferri, L. Gmez, M. Schneider, A. Vaisman, and E. Zimnyi, "Cube algebra: A generic user-centric model and query language for OLAP cubes," *Data Warehous. and Min.*, vol. 9, no. 2, pp. 39–65, 2013.
[57] G. Schwartzberg, "Oracle Essbase 11.1.2.3," 2014.
[58] Oracle Corporation. (2013). Oracle business intelligence foundation suite. [Online]. Available: http://www.oracle.com/technetwork/middleware/bi/bi-foundation-suite-wp-215243.pdf
[59] Microsoft Corporation. (2014). Power BI for office 365 faq. [Online]. Available: http://office.microsoft.com/en-us/excel-help/power-bi-for-office-365-faq-HA104219793.aspx
[60] R. Kimball, *Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. New York, NY, USA: John Wiley & Sons, Inc., 1996.

Nahla Haddar has received her master degree in new technology and computer system dedicated from the National School of Engineers of Sfax in January 2011. She is working towards his Ph.D. at the Faculty of Economic and Management of Sfax, and she is a researcher member at Multimedia Information Systems & Advanced Computing Laboratory (MIRACL) of Sfax University. Her research interests include modeling, verification and automation of business process.

Mohamed Tmar has received his Ph.D. degree in computer science from Paul Sabatier University, Toulouse III in July 2002. He is currently an assistant professor at the Higher Institute of Computer Science and Multimedia of Sfax. His research interests are in many areas of information retrieval, including retrieval models, representation, web search, query processing, medical retrieval, and XML retrieval, and business process management. He is also a member at MIRACL and he has served on numerous program committees.

Faiez Gargouri has received his Ph.D. degree in computer science from Renes Descartes, Paris 5, since 1995. Then, received his university accreditation degree in IT Management from University of Tunis. He is currently a professor at the Higher Institute of Computer Science and Multimedia of Sfax. His research interests are in many areas of information retrieval, including ontology, artificial intelligence, knowledge management, business process management, and systems engineering. He is also a member at MIRACL and he has served on numerous program committees.