A Database, specification and Theory Perspective (Data-Aware, Analysis)

Akbar Sanchouli *, Habib Piri** and Hossein Haghshenas***

*M.Sc. Graduate, Department of IT Management, University of Sistan and Baluchestan, Iran
**M.Sc. Graduate, Department of Accounting, Zahedan Branch, Islamic Azad University, Zahedan, Iran
***Department of Archeology, University of Sistan and Baluchestan, iran
(Corresponding author:akbar sanchouli)
akbarsabz@gmail.com

Abstract : While logical theories of information attitudes, such as knowledge, certainty and belief, have flourished in the past two decades, formalization of other facets of rational behavior have lagged behind significantly. In this work we survey the research of data-aware processes that has been carried out in the database theory community. We show that this community has indeed developed over the years a multi- faceted culture of merging data and processes. We argue that it is this community that should lay the foundations to solve, at least from the point of view of formal analysis, the dichotomy between data and processes still persisting in business process management. Will discuss one approach to tackling the notion within a logical framework, based on a database Perspective.

[Akbar Sanchouli, Habib Piri and Hossein Haghshenas. **A Database, specification and Theory Perspective (Data-Aware, Analysis).** *Rep Opinion* 2016;8(4):46-57]. ISSN 1553-9873 (print); ISSN 2375-7205 (online). http://www.sciencepub.net/report. 4. doi:10.7537/marsroj08041604.

Keywords: Database perspective, business, Data-Aware, information attitudes, database theory community

I. Introduction

On the contrary, data management experts consider data as the driver of the processes in an organization, and assume that guaranteeing data quality is sufficient to ensure proper consideration of business relevant data so as to impact process improvement efforts. A 2009 survey by Forrester1 [1], whose outcome is also reported in [2], has addressed the important question of which of the two aspects should be given priority from the point of view of IT management. Unsurprisingly, the role played by an individual within IT strongly affects the perception of the relative importance of processes and data within an organization: Professionals concerned with the management of business processes downplay the importance of data, and view it as subsidiary to the processes that manage them; as a consequence, they do not pay attention to the quality of data and on how the business processes can ensure that data assets can be maintained clean [3]. An immediate consequence of this dichotomy is that there is very limited collaboration and cost sharing between the teams on the one hand running the (master) data management (MDM) initiatives and on the other hand managing the business process [4]. This is also confirmed by Forrester's survey, where for 83% of the respondents there was no interaction, and only in 8% of the cases the master data management and business process modeling efforts were fully coordinated [5].

A further consequence is that there is little attention also on the side of tool vendors to address in their products the requirements coming from a combined treatment of processes and data. On the one hand, data management tool vendors cons ider processes only insofar as they affect the direct management of the data within the tools, but they do not pay attention to the processes that actually make use of the data [6]. On the other hand, Business process modeling suites do not allow for the connection of data to the processes [7]. Service oriented architectures (SOA), which make it possible to divide the functionality of large systems into component services, are advocated as a solution to the data-process dichotomy [8]. However, while favoring component reuse, they do not address the need of connecting the data to the organizational processes so as to facilitate their improvement, and in fact data continues to be "hidden" inside systems [9]. In addition to SOA, [10] identifies two key areas in which an explicit representation of data in process models is crucial. The first is the modeling of the core assets of an organization, due to the fact that the data stored in different IT systems is crucial for the execution of the business processes that create the value of the organization itself [11]. Hence, the business processes depend on such data, and in order to keep the organization operational, the former need access to the latter [12].

This dependency should be accounted for explicitly. The second is business process controlling, due to the fact that both the key performance indicators, and the specifically, we consider below the following lines of research:

1. database evolution and transactions;
2. temporal databusiness goals of the organization on which they

management;

3. active databases;

4. Work flowdepend, are defined in terms of data. To evaluate and formalisms and systems;

5.temporal integritycontrol these indicators, the activities the goals need to be identified, and contributing to this is done by constraints. For each of these areas we overview the main research objectives and achievements. Our aimconsidering the appropriate data objects on which these activities operate. In order to support this task, process models need to shift the emphasis from control flow to the data [13, 14]. It follows that there is a strong need to incorporate data modeling features in (business) process here is not to be comprehensive, but rather to highlight the mainstream directions relevant to the topic of this paper that have characterized the research in databases [16].modeling languages, and to enrich business process.

*A. Database Evolution and Transactions*analysis tools to deal with data [15].

This demands for the problem of evolution of data in a database by suitable modeling languages, methodologies and means of atomic operations and their combination systems supporting the integrated management of inside transactions has been considered from early on asprocesses and data, and, possibly above all, it calls for a more foundational approach, to provide a clear a key issue to investigate in databases [17]. Apart from the fundamental problems of concurrency control andsemantics for (data-aware) process consequently enable their analysis.

Models, and to serializability [18, 19, 20, 21] for early results), updates and transactions have been considered also of their interaction with (static) database constraints.

II. Dynamics In Database Theory

We overview here how the database theory community has been contributing to the analysis of data-aware processes [4]. We do so by first looking at some key lines of research that have considered the interaction of both static and dynamic aspects of data management.

Equivalence and optimization of relational transactions, consisting of linear sequences of insertions, deletions, and updates, using simple selection conditions based on individual attribute values for each tuple, is investigated in [22, 23].

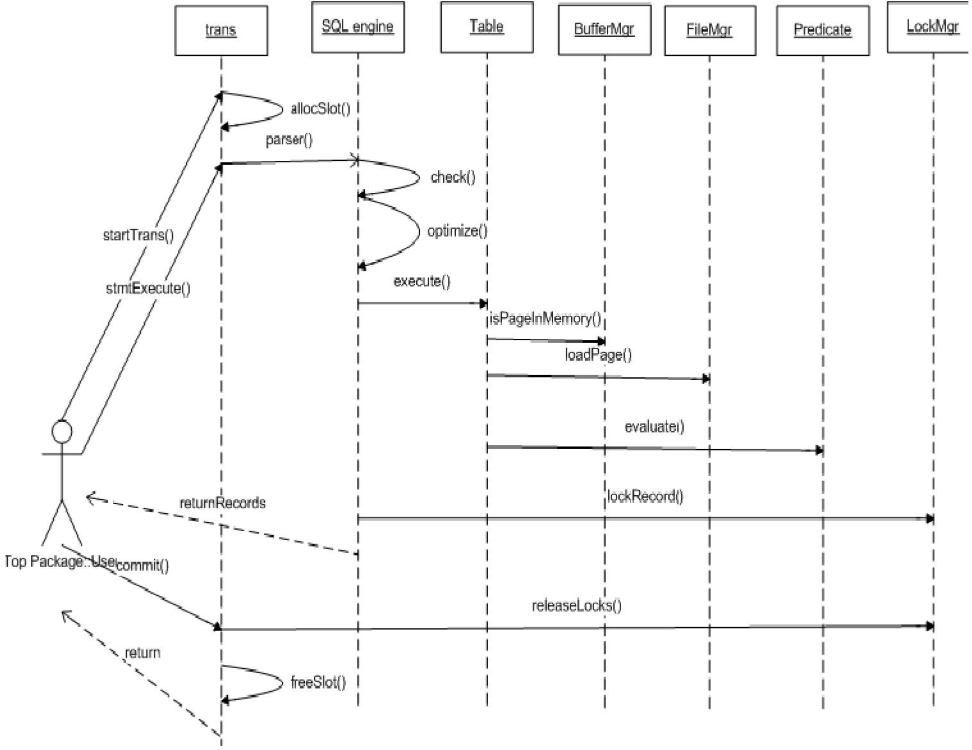

Fig. 1. A formal model (called dynamic relational model) for evolution over time of a database, seen as asequence of instances is presented in [24, 25]. The effects on evolution of dynamic constraints (specifically, dynamic functional dependencies), which relate one database instance to the next inthe sequence are studied. Specifically, the problem of inferring static constraints from knowledge about the evolution history of the database, as expressed by the dynamic constraints, is investigated [26]. The impact of dynamic constraints on the update of a specific form of views, in which each tuple represents an object with its properties, is considered in [27].

B. Temporal Data Management

A temporal database provides mechanisms to store data as it evolves, and to query its historical states using suitable extensions of standard query languages like SQL [28]. Research on temporal database originates from the observation that temporal data management can be very difficult if one uses conventional database systems [29]. The work on temporal databases and query languages goes back to [30], which provides a Description of syntax and semantics of temporal extension of Quel (a calculus-based query language for the Ingres system) that makes use of Allen's interval relations [31].

A temporal database model, in which each tuple is time stamped with a union of time intervals, is defined [32]. The notion of "weak relation" as the equivalence class of all time stamped relations for which the snapshots at each time point are equal, is introduced, and an algebra over such weak relations is defined and studied. Datalog extended with unary function symbols (successor), is studied in [33], and a mechanism is proposed to finitely represent infinite query answers via rules that may be returned together with explicit tuples. A framework for reasoning about infinite temporal information, based on generalized tuples with additional temporal attributes and constraints, is presented in [34]. Temporal attributes are defined by infinitely repeating points (of the form z (n) = c + kn) and constraints are conjunctions of linear equalities and inequalities on temporal attributes. Contrast this to constraint databases, where constraints are used to describe multiple databases, as opposed to a single database with infinite temporal information. The paper relates predicates definable by generalized relations with those definable in Presburger arithmetic. It studies the complexity of relational algebra on generalized relations, which return finite representations of possibly infinite answers [35]. Whereas positive existential queries are in PTIME (in data complexity), arbitrary queries (with negation) are NP-hard and in 2EXPTIME.

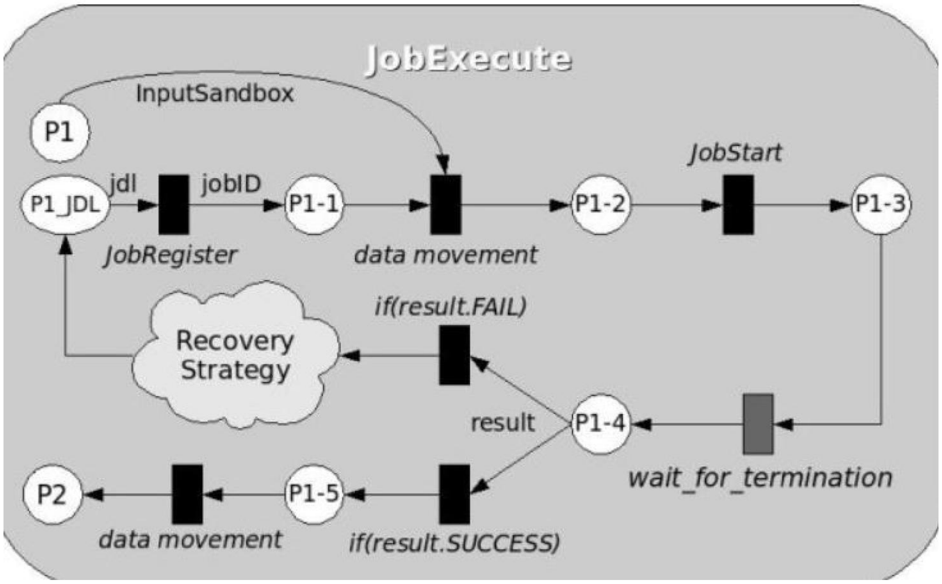

Fig. 2. The Job Register, Job Submit operations are provided by the WMS via its Web Service interface, this make it possible to express the Job Execute sub-workflow in terms of atomic operations (actually Web Service invocation) the engine can understand and execute. The execution of a job in the gLite middleware consists in a job Submit (which semantics can be also obtained by invoking -- in sequence -- job Register and a job Start) and the monitor of the job activity until done. Job monitoring can be done via the WMS using the Logging and Bookkeeping Service (LB) also available via a Web Service interface. The wait_for_termination task can be modelled using a sub-workflow which implement a specific monitoring strategy (*polling* or *notification-based*) [37].

C. Work Flow Formalisms and Systems

A further topic integrating static and dynamic aspects of data that has been addressed by the database community is that of systems to manage workflows. A workflow can be considered as a collection of activities designed in such a way that a group of (human or artificial) agents can carry out in a coordinated way a specific complex process. Workflow management systems provide a framework for capturing the interaction among the activities in a workflow [36]. Research in databases has contributed to this area by studying formalisms and systems that would support transactional aspects of workflows [38, 39]. Specifically, [40] considers a setting that deals both with task dependencies in a workflow, and dependencies between operations on the data by multiple interacting transactions. However, it does not consider the actual data and the changes performed on it, but only the order in which operations are executed. Another interesting perspective on workflows is the use of typical database functionalities (persistence, transactions, complex querying, provenance, etc.) to support the activities related to managing workflows and their execution [41, 42, 43, 44, 45]. The recent survey [46] contains an in-depth treatment of this aspect. The importance of data not only in the context of a single workflow, but to drive the integration between multiple, inter-organizational workflows, has been considered since the late nineties in the Vortex workflow management system [47].

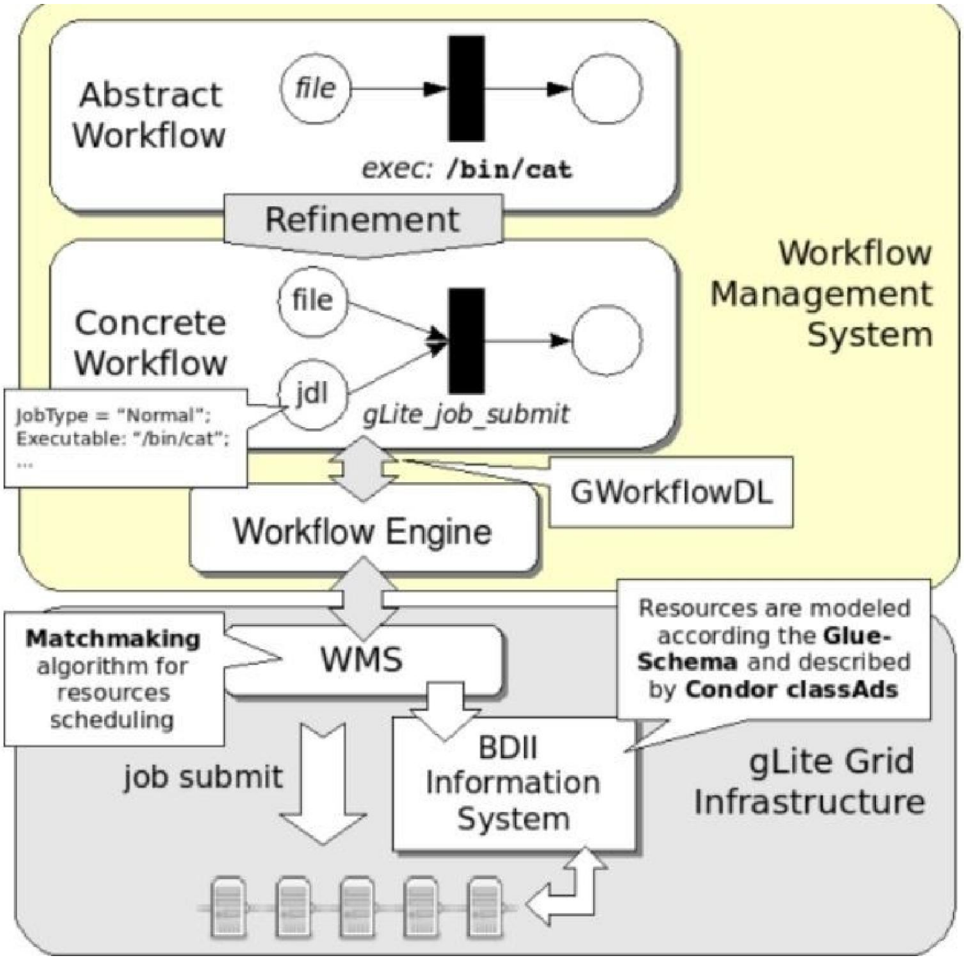

Fig 3. The core component of the system is the workflow engine which *execute* concrete workflows written in GWorkflow DL. The engine simply schedules the activities according to Petri Nets model and demand their execution to the WMS. However, the engine, by design, has no knowledge about the underying middleware and Grid operations, such as job submission, monitoring, etc... muse be expressed in terms of atomic operations the engine can execute. For example the submission of a job to the gLitemiddlware requires several web services to be invoked in sequence; sequence which can be easily represented using a workflow. As a conseguence, providing to the engine few *atomic* functionalities, such as web service interaction and local method execution make it possible to execute complex scientific processes simply by composing atomic operations as depicted below [49].In Vortex, data implicitly introduce additional dynamic (data-flow) constraints among activities belonging to the different interacting workflows. Verification of Vortex workflows has been studied by considering the control-flow component, but by considering the contribution of the data component only in terms of the induced data-flow constraints, without explicitly capturing the complex interplay between the two components [48].

D. Criteria for a Theory of Intention, and the Database Persective

When considering how to formalize intention nor any other complex natural notion it is useful to consider up front the yardsticks by which one would evaluate the theory. These in turn are dictated more deeply by the sort of relevance one seeks for the theory. The philosophical discourse relies strongly on particularly instructive test cases. The "morning star evening star" example [50] catalyzed discussion of cross-world identity in first-order modal logic (you may have different beliefs regarding the star seen in the morning from those regarding the star seen in the evening, even though, unbeknownst to you, they are in fact the same starVenus). Similarly, the example of believing that you will win the lottery and coincidentally later actually winning it served to disqualify the definition of knowledge as true belief, and another example argued against defining knowledge as *justified* true belief [51]. One type is psychological relevance. This characterizes much of the philosophical literature on intention, a few examples of which are [52, 53, 54, 55, 56]. Thus, for example, in [57] Bratman speaks of "the psychological economy of planning agency"; his goal, and that of most other philosophers writing on the topic, is to shed light on the human experience, in this case on "practical reasoning" of the kind performed by resource- constrained human beings. An alternative sort of relevance is social relevance, which typifies work in the social sciences. A clear case in connection with intention is [58], which studies the role of intention in the legal penal system. A rather different type of relevance, and the one I focus on in this article, is what might be called *artifactual relevance*. This has typified work in computer science, and in particular in artificial intelligence (AI). In this case there is a particular artifact (usually defined abstractly in mathematical terms), whose behavior is completely specified and thus in principle understood, but for which one seeks an intuitive high-level language to describe its behavior. A good example is the use of the notion of "knowledge" to reason about distributed systems [8]. The protocol governing the distributed system is well specified, but intuitively one tends to speak about what one processor does or does not know about the other processor at any given state of the system(including, recursively, the knowledge of the other processor), and the role of the mathematical theory of knowledge is to formalize this reasoning. The primary message of this article is that a similar art factual perspective can be useful in connection with intention. These different perspectives are not mutually exclusive, and in fact there is a healthy cross-pollination among them[59]. Thus for example the legal discussion in [60] is in direct dialog with the philosophical literature, and the Cohen and Levesque theory of intention [61]to which I will return lateris directly inspired by Bratman's theories, in particular [62].

E. The Belief Intention Database

This means that the database must represent both beliefs and intentions, and this in turn suggests a variety of consistences that must be preserved by the database:

1. Beliefs must be internally consistent.2. Intentions must be internally consistent. Adopting a somewhat restrictive view of action, we might say the following:(a) At most one action can be intended for any given time moment.(b) If two intended actions immediately follow one another, the earlier cannot have post conditions that are inconsistent with the preconditions of the latter. Condition 2(b) can actually be deduced from the following requirements. 3. Intentions must be consistent with beliefs. This means that: (a) If you intend to take an action you cannot believe that its preconditionsdo not hold.8 (b) If you intend to take an action, you believe that its postconditionshold. A few remarks on these requirements are in order. Requirement 1 is no different from the requirement in the belief change (AGM) theories. Requirement 2(b) is essentially Bratman's *consistency* requirement [3], instantiated to our setting. Requirement 3(a) is what is sometimes called *strong consistency*. A stronger version of this requirement would be that you believe that the preconditions of you intended action hold; this would be an instantiation of Bratman's *means-ends coherence* requirement [63]. But this does not seem useful from the database perspective, since only at the conclusion of planning and sometimes not even then are all these preconditions established. Making this stronger requirement will blur the distinction between the database and the planner it serves [64]. One could also question the asymmetry between pre and post conditions, and specifically, in connection with requirement, why one must believe that the post conditions of one's intentions.

From the philosophical perspective this indeed might be debatable or at least very unclear. From the database perspective, however, it is a good fit with how planners operate. Adopting an optimistic stance, they feel free to add intended actions so long as those are consistent with current beliefs, but once they do they continue acting based on the assumption that these actions will be taken, with all that follows from it. Since we are not considering actions whose effects are uncertain or dependent on the conditions that obtain when the action is taken, so long as action is planned the planner firmly believes whatever follows from it. Finally, these requirements relate the conditions on belief and on intention, but do not reduce the latter to the former. Arguments for and against the alternative, reductionist view (called "cognitivist" by Bratman), which does reduce intention to belief, are discussed, among other places, in [65, 66, 67, 68]. The main lesson from all this is that whereas in the philosophical approach there is much agonizing over what the *right* definition is, in the art factual(and in particular, database) approach the question is what a *useful* definition is[69]. One could imagine different intelligent databases, each providing different services to the planner, and each one would be governed by a different logic. The process of revision is made complex by these requirements. The revision of beliefs may trigger a revision of intentions and vice versa, potentially leading to a long cascade of changes [70].

Table 1: Revised relationships between terminology.

Viewpoint [1]	**[2]**	**[3]**	**[4] = [3] + commitment**	**[5]**
Psychology	Belief	Desire	Intention	Wherewithal ("how")
Design/Model	World Model	Goal	Commitment	Plan
Implementation	Knowledge Base	-	-	Running (or instantiated) Plan

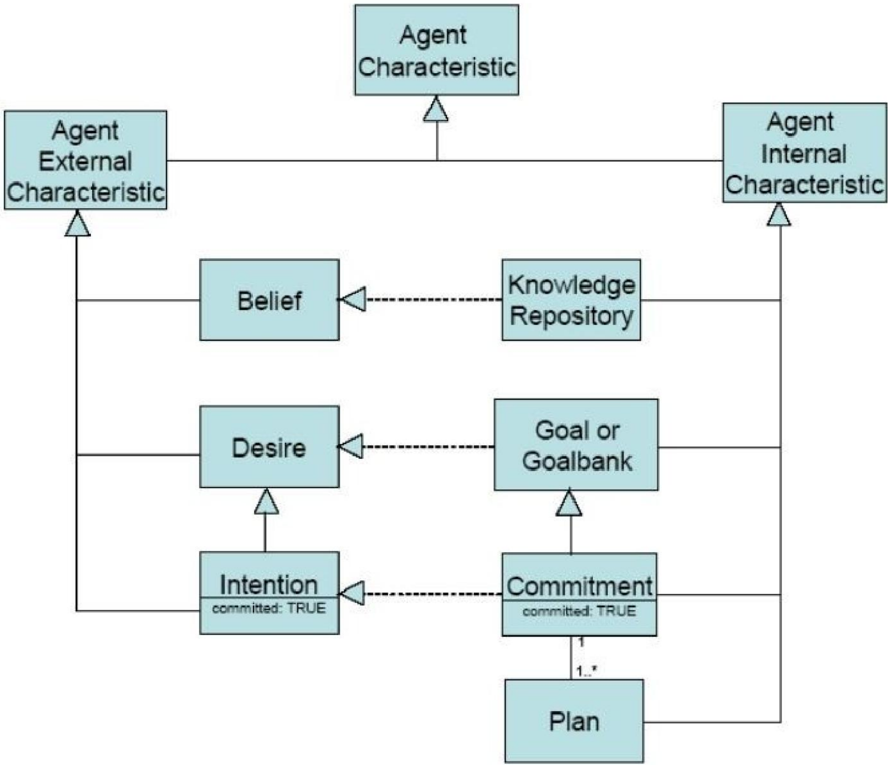

Fig. 4. Metamodel of concepts used in BDI architecture.

III. Business Process Analysis In Database Theory

Next turn to research that directly tackle data-aware process analysis, focusing on higher level processes than transactions, such as business processes [71]. Recall that the presence of data on the one hand makes the system dynamics infinite-state in general and, on the other hand, requires to go beyond propositional temporal logics[72]. Indeed, in order to properly query the state of the system by extracting data, one needs first-order quantification within and across the states of the system. Various approaches have been proposed, that differ in: The structure of the process component, as well as its interaction with the data component, and with the external environment. The kind of analysis problem that is considered; most works focus on verification of arbitrary temporal properties expressed in the adopted formalism; other approaches fix a set of specific problems they aim to solve. The considered temporal formalism, and consequently the kind of properties that can be expressed; such properties are typically formulated in a variant of first-order temporal logic [73].

A. Relational Transducers

One of the most significant approaches proposed by the database community to model high-level (business) processes is that of relational transducers, originally proposed to support forms of ecommerce[74, 75]. Relational transducers explicitly account fora dynamic component, reminiscent of active databases and transactional workflows, on top of full-fledged databases.

More specifically, a relational transducer is a tuple (S; -!), where: (i) S is the relational transducer schema, constituted by pair wise disjoint relational schemas for input, state, output, and fixed (external) database, and where the log is a further relational schema used to maintain the semantically meaningful portion of an input output exchange; (ii) is a state-update transition function mapping instances of input, state and fixed database to instances of the next state; (iii) ! is an output-update transition function mapping instances of input, state and fixed database to instances of the next output. The semantics of a relational transducer is based on linear time. In particular, it captures the evolution of state and output sequences, in response to a sequence of inputs representing the interaction with the external world, as stored in the log. The problems subject to analysis range from log validity (checking whether a log sequence can be generated with some input sequence), goal reachability, containment (testing whether every valid log of one transducer is also valid for another) and compatibility (checking whether two transducers have a common log) to the verification of specific first-order temporal properties with a past-time operator. These problems are in general undecidable. However, decidability and, in particular, a NEXPTIME upper bound for the verification problem, have been obtained in [76, 77] by requiring transducers to be semi-positive cumulative state (Spocus). In a Spocus transducer, the state accumulates all inputs received, and the outputs are defined by a non-recursive, semi positive set of catalog rules. In [78, 79], a generalization of Spocus transducers, called ASM transducers, has been studied. ASM transducers do not necessarily accumulate input, and their rule application is in general guarded by arbitrary first-order formulas. In this setting, the aforementioned problems are reconsidered, and verification is addressed for a variant of first-order LTL in which temporal operators can not appear in the scope of first-order quantifiers, except for outermost universal quantifiers. Even though verification is undecidable for general ASM transducers, two main restrictions that guarantee decidability are identified: (i) ASM transducers for which the fixed (external) database is explicitly known, and the set of values allowed in the input is restricted to those appearing in that database;(ii) ASM transducers which bound a-priori the maximum amount of input that can be received in one computation step. Complexity of verification for such restricted versions range between PSPACE complete to EXPSPACE- complete, depending on whether the maximumarity of the employed relations is bounded a-priori or not.

B. Artifact- Centric Systems

The artifact-centric approach to business process modeling, which began at IBM Research in the late

1990's and was first presented in [80], proposes business artifacts (or simply artifacts) to model key business-relevant entities. Artifacts are equipped with an information model, representing the data maintained by the artifact, and they evolve over time following a so-called lifecycle. Processes organize atomic tasks or available services that are of interest into a possibly complex workflow. The artifact-centric approach provides a simple and robust structure for business process development, which has been advocated as superior to the traditional activity-centric approach, especially when dealing with complex and large process models. While the traditional workflow approach does not lend itself to componentization in a natural way [81], the artifact-centric approach is claimed to enhance efficiency, especially when dealing with business processtrans formations to expand and/or streamline the process [82,83, 84]. Fundamental notions from the artifact-centric approach have also been deployed in commercial products underlying IBM's commercial service offerings [85].

The surveys [85, 60] overviewed the research results on the artifact-centric approach to business process specification, management, deployment and analysis, tracing the roadmap of research directions and challenges. As far as verification is concerned, this triggered several lines of research aiming at decidable techniques for verification over processes and data, to be reassessed and extended towards the artifact-centric setting. Seminal works on the analysis of artifact-centric systems is presented in [86, 85]. In [81], systems constituted by multiple interconnected artifacts are studied. The artifact information model contains the current state, and a tuple of attributes. Each attribute, in turn, may refer either to a primitive value or to some other artifact instance. Artifact lifecycles have a procedural flavor, based on finite state machines whose transitions either create a new artifact, or modify/eliminate an existing one. In this setting, first-order CTL with quantification across states is considered, showing decidability of verification for such formulas in the case of bounded domains, and in the case of unbounded domains, under the assumption that quantification only ranges over artifacts (and not values),and the number of artifacts is bounded. [42] tackles artifact systems that are similar, in spirit, to the ones of [81], but where lifecycles follow a more declarative style, based on business rules that activate services. Services are in turn described in terms of preconditions and non-deterministic effects related to the creation, manipulation and elimination of artifacts. Manipulation of attributes focuses only on whether these attributes are defined or undefined (so that values are abstracted away). A set of pre-defined reasoning tasks (successful path completion, existence of dead-end paths, attribute redundancy)is tackled, showing that all are undecidable in the general case, but become decidable if no new artifacts can be created, or by imposing various restrictions, such as monotonicity of services(each attribute is written at most once). In [70], the artifact model proposed in [42] is extended so as to include a static read-only database, and to handle a relational state in addition to attributes, whose values are not abstracted away and can be compared by service and property specifications according to a dense linear order. Runs can receive unbounded external in put from the infinite domain of values. As verification formalism, a variant of first order LTL is considered, where statements about individual artifact instances in the run may share variables that are outermost universally quantified. Decidability of verification is obtained by restricting such logic and the system specification to be guarded. The guarded restriction introduces a form of bounded quantification in the properties and formulas driving the system's evolution, which resembles input-boundedness [87, 88,89]. In particular, read-only and read-write database relations are accessed differently, querying the latter only by checking whether they contain a given tuple of constants. It is shown that this restriction is tight, and that integrity constraints cannot be added to the framework, since even a single functional dependency leads to undesirability of verification. Decidability comes with a PSPACE upper bound for fixedarity schemas, and EXPSPACE otherwise. [61,62, 63] extend this approach by forbidding read write relations, but this allows the extension of the decidability result to integrity constraints expressed as embedded dependencies with terminating chase, and to any decidable arithmetic. Another line of research building on the artifact-centric paradigmis [9, 3, 10, 11], which study the specification and verification of artifact- centric systems that rely on an active XML-based information model. Active XML [2] (AXML for short) extends XML by allowing parts of the document to be specified in an intentional way, by means of embedded calls to internal functions or external services. In the artifact-centric setting, AXML documents support the design of complex workflows, providing at the same time a description of the underlying data and of the sub- tasks (formally, internal functions) to be orchestrated by the workflow. In particular,the boundaries of decidability for the verification of systems basedon multiple, interacting AXML documents are delineated. Temporalproperties of runs are specified in a tree pattern-based temporallogic, called Tree-LTL, which exploits tree-like patterns to query the states of the system, and combines them through linear-time temporal operators to predicate about the evolution of a system run. Similarly to the logics considered in Section 4.2, in Tree-LTL variables are existentially quantified within a state, or universally quantified by means of an outermost quantifier. The systems considered for verification rely on guarded AXML (GAXML) documents, which control the initiation and completion of sub-tasks by means of Boolean combinations of tree-patterns. Decidability of verification is achieved by disallowing recursion in GAXML systems, which leads to bound the total number of sub-tasks invoked along a run. In this setting, the complexity of verification is shown to beco2-EXPTIME-complete.

In [4, 5], the problem of comparing different data-aware workflow specification frameworks based on AXML is tackled. It is argued that comparing workflow specification formalisms is intrinsically sically difficult because of the diversity of data models and control flow mechanisms, and the lack of a standard yardstick for expressiveness. For example, AXML workflows could employ automata, pre-and-post conditions, or declarative temporal logic formulas to express the dynamics of the system.

IV. Conclusions

In this work, we surveyed the research on foundations of data aware (business) processes that has been carried out in the database theory community. This community has developed rich techniques to deal with data and processes and among the various areas of computer science it is probably the one in the best position to lay the foundations of data-aware process analysis. Several challenges are ahead of us. In particular, the work done in the last years on verification of data-aware processes shows that the analysis techniques proposed are exponential in those data that change. So circumscribing what can be changed by a process appears to be a key issue to make verification practical. This is particularly relevant in the context of processes acting on web data like those that are the focus of. One could argue that this approach, while perhaps useful for some applications, does not shed light on core philosophical issues. I actually believe that the pragmatic approach forces one to confront issues that are otherwise glossed over. Obviously many of the design decisions made here make contact with notions that came up in philosophy: consistency of intentions, coherence of intentions, intention agglomeration. Of course, the very planning context is very consistent with the discussion of practical reason in philosophy. The difference is that here these notions take on a very precise meaning. To be sure this higher resolution comes at a price, since it ensures a mismatch with some elements of the human experience.

References

1. M. Abadi and Z. Manna.Temporal logic programming. *J. of Symbolic Computation*, 8(3): 277– 295, (1989).
2. Bratman, M. Intention, plans and practical reason. Cambridge: Harvard University Press.(1987).
3. S. Abiteboul, P. Bourhis, A. Galland, and B. Marinoiu. The AXML artifact model. *In Proc. of the 16th Int. Symp.On Temporal Representation and Reasoning (TIME 2009),* pages 11–17, (2009).
4. P. R. Cohen, and H. Levesque, Intention is choice with committment. Artificial Intelligence, 42(3), 213– 261. (1990).
5. J. de Kleer. An assumption-based TMS. *Artificial Intelligence,* 28(2), 127–162, (1986).
6. S. Abiteboul, L. Herr, and J. Van den Bussche. Temporalversus first-order logic to query temporal databases. *In Proc. of the 15th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'96),* pages 49–57, (1996).
7. S. Abiteboul and R. Hull.IFO: A formal semantic database model. *ACM Trans. on Database Systems,*12(4): 297–314, (1987).
8. S. Abiteboul, R. Hull, and V. Vianu. Foundations ofDatabases. Addison Wesley Publ. Co., (1995).
9. S. Abiteboul, L. Segoufin, and V. Vianu.Static analysis of Active XML systems. *In Proc. of the 27th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2008),* pages 221–230, (2008).
10. S. Abiteboul, L. Segoufin, and V. Vianu. Modeling and verifying Active XML artifacts. *Bull. of the IEEE Computer Society Technical Committee on Data Engineering,* 32(3): 10–15, (2009).
11. S. Abiteboul, L. Segoufin, and V. Vianu Static analysis of Active XML systems. *ACM Trans. on Database Systems,* 34(4): 23:1–23:44, (2009).
12. S. Abiteboul, P. Senellart, and V. Vianu.The ERC webdamon foundations of web data management. *In Proc. of the 21st Int. World Wide Web Conf. (WWW 2012*), pages 211–214, (2012).
13. S. Abiteboul and V. Vianu.Transactions in relational databases (preliminary report).In Proc. of the 10th Int. Conf. on Very Large Data Bases (VLDB'84), pages 46–56, (1984.
14. S. Abiteboul and V. Vianu.Transactions and integrity constraints.In Proc. of the 4th ACM SIGACT SIGMOD Symp.on Principles of Database Systems (PODS'85), pages 193–204, (1985).
15. S. Abiteboul and V. Vianu. Deciding properties of transactional schemas. In Proc. of the 5th ACM SIGACT SIGMOD Symp.on Principles of Database Systems (PODS'86), pages 235–239, (1986).
16. Khan, M.N., and Ghauri, S. "The WiMAX 802.16e Physical Layer Model" IEEE Explore.
17. IEEE 802.16-, "IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems" (2006).
18. Siaud, I., and Benko, J., "Text for full band diversity interleaving," [online] Doc#: IEEE 802.22- 07/0xx, available: http://www.ieee802.org/22/, (2007).
19. Ahmadi, M., Rohani, E., Naeeni, P.M., and Fakhraie, S.M. "Modeling and Performance Evaluation of IEEE 802.22 Physical Layer" *2nd International Conference on Future Computer and Communication* (2010).
20. Proakis, J.G., Digital Communications, 3rd ed., Vol. 2. McGraw-Hill: New York, (1995).
21. Aiken, J. Widom, and J. M. Hellerstein. Behavior of database production rules: Termination, confluence, and observable determinism. *In Proc. of the ACM SIGMOD Int. Conf. on Management of Data, pages* 59–68, (1992).
22. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM,* 26(11):832–843, (1983).
23. Awad, G. Decker, and M. Weske. Efficient compliance checking using BPMN-Q and temporal logic. I*n Proc. Of the 6th Int. Conference on Business Process Management (BPM 2008),* volume 5240 of Lecture Notes in Computer Science, pages 326–341. Springer, (2008).
24. Bagheri Hariri, B., Calvanese, D., De Giacomo, G., De Masellis, R. and Felli, P. Foundations of relational artifacts verification. In Proc. of the 9th Int. Conference on Business Process Management (BPM 2011), volume 6896 of Lecture Notes in Computer Science, pages 379–395. Springer, (2011).
25. B. Bagheri Hariri, D. Calvanese, G. De Giacomo, R. De Masellis, and P. Felli.Description logic Knowledge and Action Bases. *J. of Artificial Intelligence Research*, 46, (2013).
26. B. Bagheri Hariri, D. Calvanese, M. Montali, G. De Giacomo, R. De Masellis, and P. Felli. Verification of relational data-centric dynamic

systems with external services. In Proc. of the 32nd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2013), (2013).

27. C. Baier and J.-P. Katoen.Principles of Model Checking.MIT Press, (2008).
28. J. Bailey, L. Crnogorac, K. Ramamohanarao, and H. Søndergaard. Abstract interpretation of active rules and its use in termination analysis. In Proc. of the 6th Int. Conf. on Database Theory (ICDT'97), volume 1186 of Lecture Notes in Computer Science, pages 188–202. Springer, (1997).
29. J. Bailey, G. Dong, and K. Ramamohanarao. Structural issues in active rule systems. *In Proc. of the 6th Int. Conf. on Database Theory (ICDT'97), volume 1186 of Lecture Notes in Computer Science, pages 203–214. Springer, (1997).*
30. J. Bailey, G. Dong, and K. Ramamohanarao. Decidability and undecidability results for the termination problem of active database rules. *In Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems* (PODS'98), pages 264– 273, (1998).
31. Z. Bao, S. B. Davidson, and T. Milo. Labeling workflow views with fine-grained dependencies. *Proc. of the VLDB Endowment*, 5(11):1208–1219, (2012).
32. M. Baudinet. T Temporal logic programming is complete and expressive. *In Proc. of the 16th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL'89),* pages 267–280, (1989).
33. M. Baudinet. T On the expressiveness of temporal logic programming. *Information and Computation*, 117(2):157–180, (1995).
34. M. Baudinet, M. Niézette, and P. Wolper.On the representation of infinite temporal data and queries. *In Proc. of the 10th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'91),* pages 280–290, (1991).
35. C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying business processes. *In Proc. of the 32nd Int. Conf. on Very Large Data Bases (VLDB 2006),* pages 343–354, (2006).
36. C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying business processes with BP-QL. *Information Systems*, 33(6): 477–507, (2008).
37. F. Belardinelli, A. Lomuscio, and F. Patrizi. Verification of deployed artifact systems via data abstraction. In Proc. Of the 9th Int. Joint Conf. on Service Oriented Computing (ICSOC 2011), volume 7084 of Lecture Notes in Computer Science, pages 142–156. Springer, (2011).
38. F. Belardinelli, A. Lomuscio, and F. Patrizi.An abstraction technique for the verification of artifact- centric systems. *In Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning* (KR 2012), (2012).
39. M. Benedikt, T. Griffin, and L. Libkin. Verifiable properties of database transactions. In Proc. of the 15th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'96), pages 117– 127, (1996).
40. D. Berardi, D. Calvanese, G. De Giacomo, R. Hull, and M. Mecella..Automatic composition of transition- based semantic web services with messaging. *In Proc. of the 31stInt. Conf. on Very Large Data Bases (VLDB 2005),* pages 613–624, (2005).
41. K. Bhattacharya, N. S. Caswell, S. Kumaran, A. Nigam, and F. Y. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems Journal,* 46(4):703–721, (2007).
42. K. Bhattacharya, C. E. Gerede, R. Hull, R. Liu, and J. Su. Towards formal analysis of artifact-centric business process models. In Proc. of the 5th Int. Conference on Business Process Management (BPM 2007), volume 4714 of Lecture Notes in Computer Science, pages 288–234. Springer, (2007).
43. K. Bhattacharya, R. Guttman, K. Lyman, F. F. Heath, S. Kumaran, P. Nandi, F. Y. Wu, P. Athma, C. Freiberg, L. Johannsen, and A. Staudt. A model-driven approach to industrializing discovery processes in pharmaceutical research. *IBM Systems Journal,* 44(1):145–162, (2005).
44. M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Trans. on Computational Logic,* 12(4): 27, (2011).
45. M. Bojanczyk and T. Place. Toward model theory with data values. In Proc. of the 39th Int. Coll. on Automata, Languages and Programming (ICALP 2012), pages 116–127, (2012).
46. J. Bonner. Workflow, transactions, and datalog. *In Proc. of the 18th ACM SIGACT SIGMOD SIGART Symp. On Principles of Database Systems (PODS'99),* pages 294–305, (1999).
47. A. Bouajjani, P. Habermehl, Y. Jurski, and M. Sighireanu. Rewriting systems with data. *In Proc. of the 16th Int. Symp. on Fundamentals of Computation Theory (FCT 2007),* (2007).
48. Bouajjani, P. Habermehl, and R. Mayr. Automatic verification of recursive procedures with one integer parameter. *Theoretical Computer Science,* 295, (2003).

49. Y. Breitbart, A. Deacon, H.-J.Schek, A. P. Sheth, and G. Weikum. Merging application-centric and data- centric approaches to support transaction-oriented multi-system workflows. *SIGMOD Record,* 22(3): 23– 30, (1993).
50. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification of infinite structures. In Handbook of Process Algebra. Elsevier, (2001).
51. P. Cangialosi, G. De Giacomo, R. De Masellis, and R. Rosati. Conjunctive artifact-centric services. In Proc. Of the 8th Int. Joint Conf. on Service Oriented Computing (ICSOC 2010), volume 6470 of Lecture Notes in Computer Science, pages 318–333. Springer, (2010).
52. K. Chandra and D. Harel. Computable queries for relational data bases. *J. of Computer and System Sciences,* 21(2):156–178, (1980).
53. P. P. Chen. The Entity-Relationship model: Toward a unified view of data. *ACM Trans. on Database Systems,* 1(1): 9–36, Mar. (1976).
54. J. Chomicki. Temporal inegrity constraints in relational databases. *Bull. of the IEEE Computer Society Technical Committee on Data Engineering,* 17(2): 33–(1994).
55. J. Chomicki. Temporal query languages: a survey. In Proc. of the 1st Int. Conf. on Temporal Logic (ICTL'94), volume 827 of Lecture Notes in Computer Science, pages 506–534. Springer, (1994).
56. J. Chomicki. Efficient checking of temporal integrity constraints using bounded history encoding. *ACM Trans. On Database Systems*, 20(2): 149–186, (1995).
57. J. Chomicki and T. Imielinski. Temporal deductive databases and infinite objects. *In Proc. of the 7th ACMSIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'88),* pages 61–73, (1988).
58. J. Chomicki and D. Niwinski. On the feasibility of checking temporal integrity constraints. *In Proc. of the12th ACM SIGACT SIGMOD SIGART Symp. On Principles of Database Systems (PODS'93),* pages 202–213, (1993).
59. J. Chomicki and D. Niwinski. On the feasibility of checking temporal integrity constraints. *J. of Computer and System Sciences,* 51(3): 523–535, (1995).
60. D. Cohn and R. Hull. Business artifacts: A data-centric approach to modeling business operations and processes. *Bull. of the IEEE Computer Society Technical Committee on Data Engineering,* 32(3): 3–9, (2009).
61. E. Damaggio, A. Deutsch, R. Hull, and V. Vianu. Automatic verification of data-centric business processes. *In Proc. of the 9th Int. Conference on Business Process Management (BPM 2011),* volume 6896 of Lecture Notes in Computer Science, pages 30–16. Springer, (2011).
62. E. Damaggio, A. Deutsch, R. Hull, and V. Vianu. Artifact systems with data dependencies and arithmetic. *In Proc. of the 14thInt. Conf. on Database Theory (ICDT 2011),* pages 66–77, (2011).
63. E. Damaggio, A. Deutsch, R. Hull, and V. Vianu. Artifact systems with data dependencies and arithmetic. *ACM Trans. On Database Systems*, 37(3):22, (2012).
64. S. B. Davidson, T. Milo, and S. Roy. A propagation model for provenance views of public/private workflows. *In Proc. of the 16th Int. Conf. on Database Theory (ICDT 2013),* pages 165–176, (2013).
65. H. Davulcu, M. Kifer, C. R. Ramakrishnan, and I. V. Ramakrishnan.. Logic based modeling and analysis of workflows. *In Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems(PODS'98),* pages 25–33, (1998).
66. G. De Giacomo, R. De Masellis, and R. Rosati. Verification of conjunctive artifact-centric services. *Int. J. of Cooperative Information Systems,* 21(2):111–139, (2012).*Sanchouli, Piri and Haghshenas* 12
67. G. De Giacomo, Y. Lesperance, and F. Patrizi. Bounded situation calculus action theories and decidable verification. *In Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012),* pages 467–477, (2012).
68. S. Demri and R. Lazic.LTL with the Freeze quantifier and register automata. *In Proc. of the 21st IEEE Symp. on Logic in Computer Science (LICS2006),* pages 17–26, (1996).
69. D. Deutch and T. Milo. A quest for beauty and wealth (or, business processes for database researchers). *In Proc. of the 30th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2011),* pages 1–12, (2011).
70. Deutsch, R. Hull, F. Patrizi, and V. Vianu. Automatic verification of data-centric business processes. *In Proc. of the 12th Int. Conf. on Database Theory (ICDT 2009),* pages 252–267, (2009).
71. Deutsch, M. Marcus, L. Sui, V. Vianu, and D. Zhou.A verifier for interactive, data-driven web applications. *In Proc. of the ACM SIGMOD Int. Conf. on Management of Data,* pages 539–550, (2005).
72. Deutsch, L. Sui, and V. Vianu. Specification and verification of data-driven web services. *In Proc. of the 23rd ACM SIGACT SIGMOD SIGART*

Symp. on Principles of Database Systems (PODS 2004), pages 71–82, (2004).

73. A. Deutsch, L. Sui, and V. Vianu. Specification and verification of data-driven web applications. *J. of Computer and System Sciences,* 73(3): 442–474, (2007).
74. Deutsch, L. Sui, V. Vianu, and D. Zhou. A system for specification and verification of interactive, data-driven web applications. *In Proc. of the ACM SIGMOD Int. Conf. on Management of Data,* pages 772–774, (2006).
75. Deutsch, L. Sui, V. Vianu, and D. Zhou. Verification of communicating data-driven web services. *In Proc. of the 25th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2006),* pages 90–99, (2006).
76. J. Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica,* 34(2): 85–107, (1997).
77. J. Esparza. Decidability and complexity of Petri net problems – An introduction. In Lectures on Petri Nets I, Lecture Notes in Computer Science, pages 374– 428. Springer, (1998).[78] X. Fu, T. Bultan, R. Hull, and J. Su..Verification of Vortex workflows. In Proc. of the 7th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2001), volume 2031 of Lecture Notes in Computer Science, pages 143–157. Springer, (2001).
78. D. M. Gabbay, I. Hodkinson, and M. Reynolds. Temporal Logic: Mathematical Foundations and Computational Aspects, volume 28 of Oxford Logic Guides. Oxford University Press, (1994).
79. S. K. Gadia. Weak temporal relations. *In Proc. of the 5th ACM SIGACT SIGMOD Symp. on Principles of Database Systems (PODS'86),* pages 70–77, (1986).
80. C. E. Gerede and J. Su. Specification and verification of artifact behaviors in business process models. *In Proc. Of the 5th Int. Conf. on Service Oriented Computing (ICSOC 2007),* volume 4749 of Lecture Notes in Computer Science, pages 181–192. Springer, (2007).
81. M. H. Graham, N. D. Griffeth, and B. Smith-Thomas. Reliable scheduling of database transactions for unreliablesystems. *In Proc. of the 3rd ACM SIGACT SIGMOD Symp.* on Principles of Database Systems (PODS'84), pages 300–310, (198).
82. D. Harel. Recurring dominoes: Making the highly undecidable highly understandable. *Ann. of Discrete Mathematics,* 24: 51–72, (1985).
83. J. M. Hellerstein. The declarative imperative: Experiences and conjectures in distributed logics. SIGMOD Record, 39(1): 5–19, (2010).
84. R. Hull. Artifact-centric business process models: Brief survey of research results and challenges. In Proc. of the On the Move Confederated Int. Conf. (OTM 2008), volume 5332 of Lecture Notes in Computer Science, pages 1152–1163. Springer, (2008).
85. R. Hull, M. Benedikt, V. Christophides, and J. Su.E-services: a look behind the curtain. *In Proc. of the 22nd ACM SIGACT SIGMOD SIGART Symp.on Principles of Database Systems (PODS 2003),* pages 1– 14, (2003).
86. P. Mitra, G. K. Venayagamoorthy, Artificial Immune System Based DSTATCOM Control for an Electric Ship Power System, *39th IEEE Power Electronics Specialist Conference*, PESC 2008, June 15-19, pp. 718-723. (2008).
87. S. Kannan "Real and Reactive Power Coordination for a Unified Power Flow Controller", *IEEE Transactions on Power Systems*, Vol. 19, No. 3, Aug. pp. 1454-1461, (2004).
88. del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. and Harley, R. G.: Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power System', *IEEE Transactions on Evolutionary Computation*, April, 12, no. 2, pp. 171 -195, (2008).

4/6/2016