

# VERIFICATION OF DATA-AWARE BUSINESS PROCESSES IN THE PRESENCE OF ONTOLOGIES

Ario Santoso

`santoso@inf.unibz.it`

Data and processes are golden ingredients for any information system. As usual, data are simply facts that might be used for a specific purpose, while a (business) process is a sequence of actions/activities that are performed in order to achieve a certain (business) goal, and that might also manipulate data during its execution. Within an information system, data are also considered as the elements that characterize the *static* aspect of the system, while processes characterize the *dynamic* aspect of the system. Due to the importance of data, they are even often considered as the driver of an organization. In fact, typically many prominent and critical (business-related) decisions within an organization are made based on the data. On the other hand, processes are also vital for any competitive business. They differentiate between good and outstanding business performance. Hence, it is inevitable that data and processes are notable aspects within information systems that influence the performance of organizations.

Although data and processes are fundamentally two different entities, they are tightly connected. However, traditional system modeling approaches model data and processes separately. When it comes to process modeling, people often abstract away the data, and when modeling the data, people often think about the processes only afterwards [44, 43, 30]. This situation might be unsatisfactory. As witnessed by [43, 30, 44, 40, 34, 27], there is evidence of the need to treat both data and processes as first class citizens when building a system. They may even be considered as “*two sides of the same coin*” [43]. Thus, focusing on data and processes separately while designing the system might be insufficient. In fact, considering both data and processes together while designing the system could promote us into a better unified holistic view of the system. Furthermore, it could help us in avoiding various problems of the traditional system modeling approaches that consider these two aspects independently (e.g., the system is inadequately covering some process scenarios [43]).

Along with the need of focusing on both data and processes simultaneously, the *artifact-centric business process* paradigm [42, 34, 27] emerges as a promising approach that combines both static and dynamic aspects while designing a system. It provides a rich and robust model for devising business processes in which data and processes are first class citizens. This initiative was initially pioneered at IBM research<sup>1</sup> [42]. Since then, extensive studies have been accomplished in this area and numerous fruitful outcomes have been achieved (e.g., [12, 2, 8, 9, 23, 32, 31]). Moreover, the artifact-centric paradigm has been successfully applied in various settings (cf. [11, 13, 24]). This line of research is often also called *data-aware (business) processes*.

Orthogonal to processes and data, *ontologies* allow us to have a formal conceptualization of the structural/intensional knowledge about the domain of interest. In particular, what do we mean by knowledge is the universal statements about data. Such statements describe the structure of the domain as well as enable us to infer/derive some implicit information from the explicit one. Typically, ontologies are formalized in logic-based languages (e.g., First Order Logic (FOL), or Description Logic (DL)). As an example, consider a *customer order processing* scenario within a company. In FOL-based ontologies, we can encode domain knowledge saying that “*each assembled order is an order*” as a first order sentence/axiom as follows:  $\forall x. \text{AssembledOrder}(x) \rightarrow \text{Order}(x)$ . Besides enabling us to conveniently structure the domain knowledge, a crucial advantage of ontologies is that they allow us to reason about the domain. For instance, in our example, whenever we know that something is an assembled order, we can infer that it is also an order. Since fundamentally ontology captures the structural knowledge of the domain of interest, we often also consider it as the structural knowledge component of a system.

Looking at ontologies and the artifact-centric approach, there are some researches on data-aware processes framework that take into account ontologies (e.g., [33, 3]). Besides allowing us to focus simultaneously on data and processes, the proposed framework enables us to incorporate the domain knowledge inside the designed system and leads us to a semantically-rich system.

---

<sup>1</sup> International Business Machine (IBM) Corp. - <https://www.ibm.com/>

When it comes to the need of ensuring the correctness of the developed system, there are various techniques that are usually applied such as (software/system) testing, peer review, simulation and formal verification. The choice of the method is typically based on the complexity of the system as well as the required degree of safety. Each of those techniques has its own advantages and disadvantages. For instance, testing might be easier to do than formal verification, but is in general less reliable. As stated by the famous computer scientist E. Dijkstra, “*Testing can only show the presence of errors, but not their absence*”. In fact, as reported in the survey of artifact-centric business processes models [34], formal verification for artifact-centric systems is an important research direction aimed at establishing sophisticated techniques to analyze the correctness of data-aware business processes systems. Model checking [7] is a widely studied and successful formal verification technique, see, e.g., [25] for notable success stories. However, the interactions between data and processes typically makes the problem more difficult since it makes the system in general become infinite states. Thereby typical model checking techniques for finite state systems are inapplicable.

In this thesis, motivated by various works on data, processes and ontologies, we focus on the formal verification of several variants of data-aware business processes systems that are enriched with ontologies. It is noteworthy to remark that this line of research opens up various fascinating connections among diverse research areas such as Databases, Formal Verification, Model Checking, Business Process Management, Knowledge Representation, and specifically Description Logics, and Reasoning About Actions.

## 1 Research Challenges

Many data-aware processes systems that have been studied so far consider a simple formalism for specifying the progression mechanism. For instance, [6, 4, 3] only consider condition-action rules to specify when and how an atomic action can be executed. Although this approach is quite expressive, one might desire a better control in specifying the desired order of actions (e.g., to choose one action or another based on the result of a condition checked over the current state, or to specify that a certain sequence of actions should be executed as long as a specified condition holds). Thus, a more sophisticated formalism is required to specify the system dynamics at a higher-level of abstraction.

Concerning inconsistency management, the majority of approaches dealing with verification in data-aware processes systems assume a rather simple treatment. In particular, they simply reject inconsistent system states that are produced by the effects of action executions (see, e.g., [29, 6, 33, 3]). In general, this mechanism is not satisfactory, since the inconsistency may affect just a small portion of the entire data, and thus should be treated in a more careful way. This is in line with what is done in numerous researches that specifically deal with inconsistencies (cf. [36, 14, 10, 20]).

Many works on data-aware processes system that incorporate structural domain knowledge typically assume that such knowledge remains fixed along the system evolution (e.g., [22, 41, 33]), i.e., that it is independent from the actual system state. However, this assumption might be too restrictive, since specific knowledge might hold or be applicable only in specific, *context-dependent* circumstances. Ideally, one should be able to form statements that are known to be true in certain cases, but not necessarily in all.

As witnessed by numerous works on data-aware processes (see e.g., [29, 8, 6, 33, 41]) the verification problem in this setting is in general difficult (more precisely, undecidable without suitable restrictions) since the number of systems states is in general infinite. Thus, off the shelf model checking technique for finite state system cannot be used directly. The situation becomes even more challenging when we also need to deal with inconsistencies and/or take into account the presence of contextual information.

In some formalisms of data-aware processes, the information model typically relies on relatively simple structures, such as tuples of typed-attributes (e.g. [31, 32, 35, 28]). This situation might cause an abstraction gap between the high-level conceptual view that business stakeholders have, and the low-level representation of information. In addition, the data layer within the system might be complicated and difficult to interact with. In this light, there is a need to have a high level conceptual view over the system evolution.

In this thesis, we aim at addressing all the issues mentioned above, by proposing novel extensions of existing models for data-aware processes systems, and by studying how these extensions affect the problem of formal

verification of expressive temporal properties. In the remaining part of the chapter, we discuss in detail the original contributions that we have provided along these lines.

## 2 Contributions

As a first broad contribution of this thesis, we introduce and study several variants and extensions of the formalism of Knowledge and Action Bases (KABs) [33], that is a formal framework which allows one to capture the manipulation of a DL Knowledge Base over time. The dynamic aspect of KABs is characterized by condition-action rules that, together with the data manipulated during the system evolution, determine the possible sequences of actions that can be executed over the KB. Specifically, the extensions we introduce are the following:

1. A formal framework, namely Golog-KABs (GKABs), for specifying *semantically-rich data-aware business processes systems* that is obtained by leveraging on the current state of the art data-aware processes systems equipped with ontologies.
2. Several variants of *inconsistency-aware Golog-KAB*, which extend GKABs by incorporating various inconsistency handling mechanisms that had been proposed in the literatures.
3. An extended version of GKABs, namely *Context-Sensitive Golog-KABs* (CSGKABs), which takes into account contextual information during the evolution.
4. Several variants of *inconsistency-aware context-sensitive Golog-KAB*, which are obtained from CSGKABs by incorporating various inconsistency management mechanisms.
5. An extension of GKABs, called *Alternating GKABs*, that separates the sources of non-determinism within a single step of evolution and allows for a more fine-grained analysis on the system evolution, while also employing sophisticated inconsistency handling mechanism and taking into account contextual information.
6. A novel framework, called *Semantically-Enhanced Data-Aware Processes* (SEDAPs), which enables us to have a high-level conceptual view over the evolution of a data-aware processes system by utilizing ontologies.

We observe that this thesis establishes two different approaches in devising a semantically-rich data-aware business processes system. One, based on GKABs and their variants, in which we have a KB that evolves under the effect of actions, requires us to specify the system from scratch. The other one, namely SEDAPs, enables us to enhance existing data-aware processes systems towards a semantically-rich system by connecting an ontology via mappings to a traditional relational data layer that evolves under action execution.

Within all of the settings above, we tackle the problem of verification of temporal properties over the system executions. This task is more challenging than in the basic setting of KABs, on which we build, since we need to deal with inconsistency in a more sophisticated manner and consider the contextual information. In the following sub-sections, we provide more details on each of these contributions.

### 2.1 Golog-KABs (GKABs)

Here we devise a formal framework for specifying *semantically-rich data-aware business processes systems* by leveraging on the current state of the art data-aware processes system equipped with ontologies [33, 5, 41, 22]. Specifically, we build on the Knowledge and Action Bases (KABs) framework that was initially proposed in [33]. Fundamentally, KABs provide a semantically rich representation of a domain in the form of a KB expressed in the lightweight DL *DL-Lite* [16], while also simultaneously taking into account the dynamic aspects of the modeled system. As usual, the *DL-Lite* KB is constituted by a *TBox* that captures the intensional knowledge about the domain and an *ABox* that keeps the data (extensional parts). The execution semantics of a KAB is given in terms of a (possibly infinite) transition system, in which each state is labeled by a DL KB and each transition represents the manipulation of the ABox by an action. Concerning action specification, rather than following the original KABs [33, 5], in which at each action execution the state is reconstructed from scratch, we adopt the action formalism in [41], in which one specifies only the facts to add and those to delete from the current state. Similar to KABs, an action execution might issue external

service calls that might inject fresh values (constants) into the system. Roughly speaking, the calls to external services can be used to model the interaction with external systems/entities as well as user input. As for the execution semantics w.r.t. service calls, instead of following [33, 5], we use the service call evaluation semantics as in [6], which is considered to be less abstract, more natural, and closer to reality. I.e., we evaluate the service calls in the sense that we substitute each service call with a concrete value when constructing the transition system. Since we use KABs that are slightly different from their original version in [33, 5], here we also show that the verification of  $\mu\mathcal{L}_A^{\text{EQL}}$  properties over KABs can be reduced to the corresponding verification of  $\mu\mathcal{L}_A$  over DCDSs [6], where  $\mu\mathcal{L}_A^{\text{EQL}}$  and  $\mu\mathcal{L}_A$  are variants of first order  $\mu$ -calculus [15] (one of the most powerful temporal logics, which subsumes LTL, PSL, and CTL\* [26]). The difference between  $\mu\mathcal{L}_A$  and  $\mu\mathcal{L}_A^{\text{EQL}}$  formulas is in the atomic parts of the formulas. The former consider Domain Independent First Order Logics queries [1] as the atomic components of the formulas while the latter consider Domain Independent EQL-Lite (UCQ) [17] queries. The reduction also preserves *run-boundedness*, which is a restriction that guarantees the decidability of DCDSs verification. Thus, exploiting the results on verification of run-bounded DCDSs, it follows that the verification of run-bounded KABs is decidable and can be reduced to standard finite state model checking.

In this thesis, we enrich KABs with a high-level, compact action language inspired by a well-known action programming language in the area of Artificial Intelligence (AI), namely Golog [39]. We call the resulting formalism *Golog-KABs* (GKABs). Thus, instead of using simple condition-action rules as in KABs, the progression mechanism in GKABs is specified using Golog programs. This allows modelers to conveniently specify the processes at a high-level of abstraction and represent the dynamic aspects of the systems much more compactly. Roughly speaking, the Golog program characterizes the evolution of a GKAB by determining the possible orders of action executions that evolve the KB over time.

To elegantly accommodate various ways of updating the ABox, we introduce a parametric execution semantics of GKABs. Technically, we adopt Levesque’s functional approach, i.e., we assume that a GKAB provides two operations:

- ASK, to answer queries over the current KB;
- TELL, to update the KB (ABox) through an atomic action.

In this work, the ASK operator corresponds to the *certain answers* computation. The TELL operation is parameterized by *filter relations*, which are used to refine the way in which an ABox is updated, based on a set of facts to be added and deleted (that are specified by the action).

In this light, filter relations provide an abstract mechanism to accommodate in the execution semantics several inconsistency management approaches based on the well-known notion of repair [37, 38, 19]. Basically, we can obtain various execution semantics for GKABs, including *inconsistency-aware semantics*, by simply defining different kinds of filter relation. For instance, we define *GKABs with standard execution semantics*, briefly *S-GKABs*, by defining a filter relation  $f_S$  that updates an ABox based on the facts to be added and deleted, and does nothing w.r.t. inconsistency (i.e., updates that lead to an inconsistent state are simply rejected).

Concerning the verification of  $\mu\mathcal{L}_A^{\text{EQL}}$  properties over S-GKABs, we have shown that we can reduce this problem to verification of KABs and vice versa. To encode KABs into S-GKABs, we simulate the standard execution semantics using a Golog program that runs forever to non-deterministically pick an executable action with parameters, or stops if no action is executable. For the opposite direction, the key idea is to inductively interpret a Golog program as a structure consisting of nested processes, suitably composed through the Golog operators. We mark the starting and ending point of each Golog subprogram, and use accessory facts in the ABox to track states corresponding to subprograms. Each subprogram is then inductively translated into a set of actions and condition-action rules, encoding its entrance and termination conditions.

## 2.2 Inconsistency-Aware GKABs

We introduce GKABs with inconsistency-aware semantics by exploiting the filter relations (i.e., we introduce various kind of filter relations and plug them in into GKABs). By incorporating inconsistency-aware semantics,

we allow each action that leads to an inconsistent state and then we repair the inconsistency. Technically, we introduce filter relations B-filter  $f_B$ , C-filter  $f_C$ , and B-evol filter  $f_E$ , where

- $f_B$  incorporates the ABox Repair (AR) semantics in [37]. Here we call such approach *bold-repair* (*b-repair*), where a b-repair of an ABox  $A$  w.r.t. TBox  $T$  is a maximal (w.r.t. set containment) subset of  $A$  that is consistent with  $T$ .
- $f_C$  incorporates the Intersection ABox Repair (IAR) semantics in [37]. Here we call such approach *certain-repair* (*c-repair*), where a c-repair of an ABox  $A$  is an ABox that is obtained by intersecting all b-repairs of  $A$  w.r.t.  $T$ .
- $f_E$  updates the ABox using the bold semantics of KB evolution [19]. In this approach, if an inconsistency arises due to an update, newly introduced assertions are preferred to those already present in the current ABox.

We call the GKABs adopting the execution semantics obtained by employing those filter relations *B-GKABs*, *C-GKABs*, and *E-GKABs*, respectively. We group them under the umbrella of *inconsistency-aware GKABs* (*I-GKABs*).

With respect to verification of  $\mu\mathcal{L}_A^{\text{EQL}}$  properties over the various types of GKABs introduced so far, we have proved the results summarized in Figure 1, where an arrow indicates that we can reduce verification in (G)KABs in the source to verification in (G)KABs in the target. Furthermore, the semantic property of

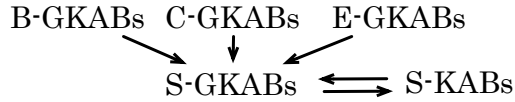


Figure 1: Reductions from I-GKABs (i.e., B-GKABs, C-GKABs, and E-GKABs) to KABs

*run-boundedness* (which guarantees the decidability of KAB verification) [6] is preserved by all our reductions. Thus, it follows that verification of  $\mu\mathcal{L}_A^{\text{EQL}}$  properties over run-bounded S-GKABs and I-GKABs is decidable, and reducible to standard  $\mu$ -calculus finite-state model checking. For all reductions from I-GKABs to S-GKABs, our general strategy is to show that S-GKABs are sufficiently expressive to incorporate the repair-based approaches, so that an action executed under certain inconsistency-aware semantics can be compiled into a Golog program that applies the action with the standard semantics, and then explicitly handles the inconsistency, if needed.

### 2.3 Context-Sensitive GKABs

As the next contributions, we extend GKABs towards Context-Sensitive GKABs (CSGKABs), which allow us to incorporate contextual information within the system. The context might change during the system evolution and influences the system execution in several ways such as: (i) determining relevant TBox assertions at each state (i.e., TBox changes along the system execution depending on the context), and (ii) influencing the decision about action executability. As a consequence of the TBox changes, essentially context also indirectly affects the results of query answering over the KB.

Concerning execution semantics, it is worth mentioning that we lift GKABs into CSGKABs by also retaining their parametric execution semantics. Therefore, we can easily define various ways of updating the ABox in CSGKABs by simply “shaping” the filter relation, which is a great basis for integrating various inconsistency management mechanisms into CSGKABs.

Regarding verification, to specify the properties to be verified, we consider a context-sensitive temporal logic  $\mu\mathcal{L}_{\text{CTX}}$ , which extends  $\mu\mathcal{L}_A^{\text{EQL}}$  with the possibility of having also “context expressions” as an atomic part of the formula. It follows that, using  $\mu\mathcal{L}_{\text{CTX}}$  we can also say something about contextual information inside the properties that we want to verify. In this thesis, we study the verification of *CSGKABs with standard execution semantics*, briefly *S-CSGKABs*, that are obtained by using the standard filter relation. To cope with the problem of verifying  $\mu\mathcal{L}_{\text{CTX}}$  over S-CSGKABs, we reduce the problem to the corresponding  $\mu\mathcal{L}_A^{\text{EQL}}$  verification problem over S-GKABs.



## 2.4 Inconsistency-Aware Context-Sensitive GKABs

We also study the combination of CSGKABs and various inconsistency management mechanisms (as in I-GKABs), which led us to the formalization of Inconsistency-aware Context-sensitive GKABs. In particular, similar to the way of obtaining I-GKABs, we employ three filter relations that incorporate the b-repair, c-repair, and bold-evolution computations. We call CSGKABs adopting the execution semantics obtained by injecting those filter relations B-CSGKABs, C-CSGKABs, and E-CSGKABs, respectively. We group them under the umbrella of Inconsistency-aware Context-sensitive GKABs (I-CSGKABs).

For the verification of  $\mu\mathcal{L}_{\text{CTX}}$  over I-CSGKABs, we show that the verification of  $\mu\mathcal{L}_{\text{CTX}}$  over B-CSGKABs, C-CSGKABs, and E-CSGKABs can be reduced to the corresponding verification of  $\mu\mathcal{L}_A^{\text{EQL}}$  over S-GKABs. Furthermore, all our reductions also preserve run-boundedness. It follows that the verification of run-bounded S-CSGKABs, B-CSGKABs, C-CSGKABs, and E-CSGKABs are decidable and reducible to the standard finite state model checking.

## 2.5 Alternating GKABs

As a deeper study on GKABs, we introduce AGKABs, which separate sources of non-determinism during the computation of successor states. Those sources of non-determinism are: (i) the choice of grounded actions, (ii) the choice of service call results, (iii) the choice among all possible new contexts, and (iv) the choice of repaired ABoxes when there are several possible repairs (which is the case for b-repairs). In I-CSGKABs, we encapsulate the computation of all of those sources of non-determinism in a single transition (i.e., roughly speaking, in a single transition, non-determinism can be caused by those four sources). In AGKABs, we separate them such that each state only has one possible source of non-determinism (one of those four sources).

Thanks to the separation of the sources of non-determinism, we are capable to do a more fine-grained analysis over the system evolution. In particular, we can verify temporal properties that quantify over each source of non-determinism. For instance, we can check a property like “*no matter which action is executed, there exists a service call result in which no matter how the context is changing, there exists a repair that leads us into a certain state that satisfy a certain property*”.

Concerning verification, we introduce  $\mu\mathcal{L}_{\text{CTX}}^{\text{Alt}}$ , which is a fragment of  $\mu\mathcal{L}_{\text{CTX}}$  where we always use the modal operators in groups of 4 (e.g.,  $\langle \rightarrow \rangle [ \rightarrow ] [ \rightarrow ] \langle \rightarrow \rangle \Phi$ ) in order to quantify separately over each source of non-determinism. Similar to I-CSGKABs, we employ three filter relations that incorporate the b-repair, c-repair, and bold-evolution computations, obtaining respectively B-AGKABs, C-AGKABs, and E-AGKABs. To tackle the problem of  $\mu\mathcal{L}_{\text{CTX}}^{\text{Alt}}$  verification over B-AGKABs, C-AGKABs, and E-AGKABs, we prove again that those problems are reducible to the verification of  $\mu\mathcal{L}_A^{\text{EQL}}$  over S-GKABs. Also in this case, our reductions preserve run-boundedness, allowing us again to reduce verification to standard finite state model checking.

## 2.6 Semantically-Enhanced Data-Aware Processes

As a further contribution, we devise a novel framework that enables us to enhance the existing data-aware business processes system into a semantically-rich data-aware processes system. In particular we propose *Semantically-Enhanced Data-Aware Processes* (SEDAPs) which are inspired by the research on Ontology-Based Data Access (OBDA) [18], where an ontology is used to provide a conceptual view over (existing) data repositories, to which the ontology is connected by means of mappings. Roughly speaking, SEDAPs can be considered as an extension of DCDSs [6] where the data layer is constituted by an OBDA system instead of simply a relational database. Through the presence of the ontology, a SEDAP provides a unified, high-level conceptual view of the system, reflecting the relevant concepts and relations of the domain of interest and abstracting away how processes and data are concretely realized and stored at the implementation level. This, in turn, is the basis for different important reasoning tasks such as verification of conceptual temporal properties, regulating how new processes can be injected into the system, synthesizing new processes starting from high level conceptual requirements, and reasoning under implicit and incomplete information.

Basically a SEDAP is constituted by three components: (i) an *OBDA system*, which keeps all the data of interest and provides a conceptual view over it in terms of a *DL-Lite<sub>A</sub> TBox*; (ii) a *process component* as in DCDSs, which characterizes the evolution (dynamic aspect) of the system; and (iii) an *initial database instance*. Conceptually, a SEDAP separates the system into two layers, the *relational layer* and the *semantic layer*. The relational layer captures the database evolution (manipulation) done by the process execution, while the semantic layer exploits the ontology for providing a conceptual view of the system evolution. This enables us to (i) *understand* the evolving system through the semantic layer, and (ii) *govern* the evolution of the system at the semantic layer by rejecting those process actions that, currently executed at the relational layer, would lead to new system states that violate some constraint of the ontology. Formally, the semantics of SEDAPs is defined in terms of two transition systems: a *Relational Layer Transition System* (RTS) and a *Semantic Layer Transition System* (STS). The RTS is the same as the transition system of a classical DCDS, which captures the evolution of the system at the relational layer, tracking how the database is evolved by the process component. On the other hand, the STS is a “virtualization” of the RTS in the semantic layer and provides a conceptual view of the system evolution. In particular, the STS maintains the structure of the RTS unaltered, reflecting that the process component is executed over the relational layer, but it associates to each state the set of concept and role assertions obtained from the application of the mappings starting from the corresponding database instance.

Within SEDAP, we address the problem of verifying *conceptual temporal properties* that are specified at the semantic layer. Roughly speaking, to tackle the verification problem, we bring down the conceptual temporal property from the semantic layer into the relational layer, by adopting the concept of “rewriting” and “unfolding” in OBDA, and then exploit the decidability results of temporal property verification in DCDS. I.e., we show that the verification of SEDAPs can be reduced to the verification of DCDSs.

Going beyond theoretical results only, we have instantiated the concept of SEDAPs into a working tool called OBGSM, in which we use the standard Guard-Stage-Milestone (GSM) model [35, 28] to represent the system in the relational layer. OBGSM provides a functionality to translate the temporal property specified at the semantic layer into the temporal property over the relational layer by applying the “rewriting” and “unfolding” technique. It exploits two already existing tools to provide its functionalities: (i) -ONTOP-<sup>2</sup>, a JAVA-based framework for OBDA, and (ii) the *GSMC model checker*, developed within the EU FP7 Project ACSI<sup>3</sup>, to verify GSM-based artifact-centric systems against temporal/dynamic properties [9]. OBGSM also becomes a part of EU FP7 Project ACSI deliverable (see [21]), and additionally, we also show how OBGSM can be used in one of the practical use cases of the EU FP7 Project ACSI.

## 2.7 Putting it all together

In addition to all reductions above, we also show that the verification of S-GKABs can be reduced to the corresponding verification of B-GKAB, C-GKAB, E-GKAB, S-CSGKABs, B-CSGKABs, C-CSGKABs, E-CSGKABs, B-AGKABs, C-AGKABs, and E-AGKABs. Thus, summing it up, we have enriched the state of the art data-aware business processes systems equipped with ontologies so that they can accommodate various prominent scenarios and all proposed extensions have no negative impact on computational complexity. All our reductions are visually summarized in Figure 2, where an arrow indicates that we can reduce verification in the formalism at the source of the arrow to verification in the formalism at the destination of the arrow. Some of the core results in this thesis have been published as detailed below:

- D. Calvanese, M. Montali and A. Santoso. *Verification of generalized inconsistency-aware knowledge and action bases*. In Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI), pages 2847-2853, AAAI Press, 2015.
- D. Calvanese and İ. İ. Ceylan and M. Montali and A. Santoso. *Verification of context-sensitive knowledge and action bases*. In Proc. of the 14th European Conference on Logics in Artificial Intelligence (JELIA), vol. 8761 of LNCS, pages 514-528. Springer, 2014.
- B. Bagheri Hariri, D. Calvanese, M. Montali, A. Santoso, and D. Solomakhin. *Verification of semantically-enhanced artifact systems*. In Proc. of the 11th Int. Conf. on Service Oriented Computing (ICSOC), volume 8274 of LNCS, pages 600-607. Springer, 2013.

<sup>2</sup> <http://ontop.inf.unibz.it/>

<sup>3</sup> “Artifact-Centric Service Interoperation”, see <http://www.acsi-project.eu/>

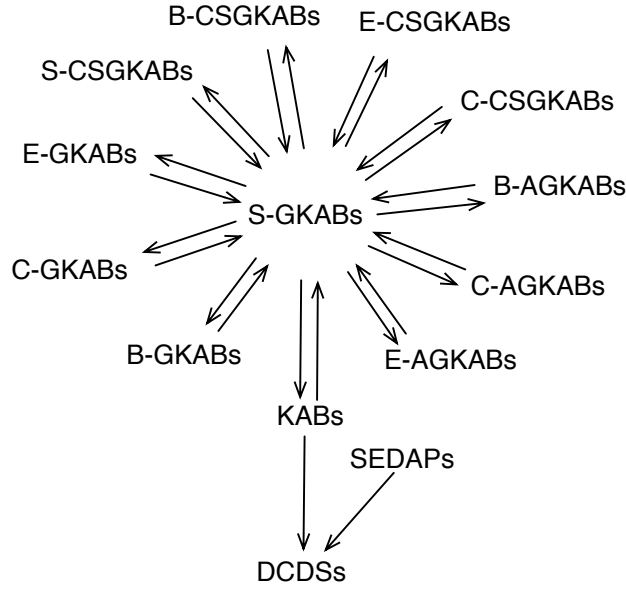


Figure 2: Summary of the reductions developed in this thesis. The meaning of an arrow from formalisms  $A$  to formalism  $B$  is that verification of  $A$  is reducible to verification of  $B$

- D. Calvanese, E. Kharlamov, M. Montali, A. Santoso, and D. Zheleznyakov. *Verification of inconsistency-aware knowledge and action bases*. In Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI), pages 810-816. AAAI Press, 2013.
- D. Calvanese, G. De Giacomo, D. Lembo, M. Montali, and A. Santoso. *Ontology-based governance of data-aware processes*. In Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR), volume 7497 of LNCS, pages 25-41. Springer, 2012.
- A. Santoso. *When data, knowledge and processes meet together*. In Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR), vol. 7497 of LNCS, pages 291-296. Springer, 2012.
- D. Calvanese, M. Montali and A. Santoso. *Inconsistency management in generalized knowledge and action bases*. In Proc. of the 28th Int. Workshop on Description Logic (DL), vol. 1350, 2015.
- D. Calvanese, İ. İ. Ceylan, M. Montali and A. Santoso. *Adding context to knowledge and action bases*. In Workshop Notes of the 6th International Workshop on Acquisition, Representation and Reasoning about Context with Logic (ARCOE-Logic 2014), volume arXiv:1412.7965 of CoRR Technical Reports, pages 25-36. arXiv.org e-Print archive, 2014. Available at <http://arxiv.org/abs/1412.7965>.
- D. Calvanese, E. Kharlamov, M. Montali, A. Santoso, and D. Zheleznyakov. *Verification of inconsistency-aware knowledge and action bases*. In Proc. of the 26th Int. Workshop on Description Logics (DL), vol. 1014 of CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/>, pages 107-119, 2013.
- D. Calvanese and A. Santoso. *Best service synthesis in the Weighted Roman Model*. In Proc. of the 4th Central-European Workshop on Services and their Composition (ZEUS), volume 847 of CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/>, 2012.
- D. Calvanese, G. De Giacomo, D. Lembo, M. Montali, and A. Santoso. *Semantically-governed data-aware processes*. In Proc. of the 1st Int. Workshop on Knowledge-intensive Business Processes (KiBP), vol. 861 of CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/>, pages 21-32, 2012.
- D. Calvanese, M. Montali and A. Santoso. *Verification of generalized inconsistency-aware knowledge and action bases (extended version)*. CoRR Technical Report arXiv:1504.08108, arXiv.org e-Print archive, 2015. Available at <http://arxiv.org/abs/1504.08108>.
- B. Bagheri Hariri, D. Calvanese, M. Montali, A. Santoso, and D. Solomakhin. *Verification of semantically-enhanced artifact systems (extended version)*. CoRR Technical Report abs/1308.6292, arXiv.org e-Print archive, 2013. Available at <http://arxiv.org/abs/1308.6292>.
- D. Calvanese, B. Bagheri Hariri, R. De Masellis, D. Lembo, M. Montali, A. Santoso, D. Solomakhin, and S. Tessaris. *Techniques and tools for KAB, to manage action linkage with the Artifact Layer - Iteration 2*. Deliverable ACSI-D2.4.2, ACSI Consortium, May 2013.
- D. Calvanese, E. Kharlamov, M. Montali, A. Santoso, and D. Zheleznyakov. *Verification of inconsistency-aware knowledge and action bases (extended version)*. CoRR Technical Report arXiv:1304.6442, arXiv.org e-Print archive, 2013. Available at <http://arxiv.org/abs/1304.6442>.
- D. Calvanese, G. De Giacomo, D. Lembo, M. Montali, M. Ruzzi and A. Santoso. *Techniques and Tools for KAB, to Manage Data Linkage with the Artifact Layer*. Deliverable ACSI-D2.3, ACSI Consortium, May 2012.



# BIBLIOGRAPHY

---

- [1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995. (Cited on page 4.)
- [2] Serge Abiteboul, Pierre Bourhis, Alban Galland, and Bogdan Marinoiu. The AXML artifact model. In *Proc. of the 16th Int. Symp. on Temporal Representation and Reasoning (TIME)*, pages 11–17, 2009. (Cited on page 1.)
- [3] Babak Bagheri Hariri, Diego Calvanese, Giuseppe De Giacomo, and Riccardo De Masellis. Verification of conjunctive-query based semantic artifacts. In *Proc. of the 24th Int. Workshop on Description Logic (DL)*, volume 745 of *CEUR Electronic Workshop Proceedings*, <http://ceur-ws.org/>, 2011. (Cited on pages 1 and 2.)
- [4] Babak Bagheri Hariri, Diego Calvanese, Giuseppe De Giacomo, Riccardo De Masellis, and Paolo Felli. Foundations of relational artifacts verification. In *Proc. of the 9th Int. Conf. on Business Process Management (BPM)*, volume 6896 of *Lecture Notes in Computer Science*, pages 379–395. Springer, 2011. (Cited on page 2.)
- [5] Babak Bagheri Hariri, Diego Calvanese, Giuseppe De Giacomo, Riccardo De Masellis, Marco Montali, and Paolo Felli. Verification of description logic Knowledge and Action Bases. In *Proc. of the 20th Eur. Conf. on Artificial Intelligence (ECAI)*, pages 103–108, 2012. (Cited on pages 3 and 4.)
- [6] Babak Bagheri Hariri, Diego Calvanese, Giuseppe De Giacomo, Alin Deutsch, and Marco Montali. Verification of relational data-centric dynamic systems with external services. In *Proc. of the 32nd ACM SIGACT SIGMOD SIGAI Symp. on Principles of Database Systems (PODS)*, pages 163–174, 2013. (Cited on pages 2, 4, 5, and 6.)
- [7] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008. (Cited on page 2.)
- [8] Francesco Belardinelli, Alessio Lomuscio, and Fabio 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)*, pages 319–328, 2012. (Cited on pages 1 and 2.)
- [9] Francesco Belardinelli, Alessio Lomuscio, and Fabio Patrizi. Verification of GSM-based artifact-centric systems through finite abstraction. In *Proc. of the 10th Int. Joint Conf. on Service Oriented Computing (ICSOC)*, volume 7636 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2012. doi: 10.1007/978-3-642-34321-6\_2. (Cited on pages 1 and 7.)
- [10] Leopoldo E. Bertossi. Consistent query answering in databases. *SIGMOD Record*, 35(2):68–76, 2006. (Cited on page 2.)
- [11] K. Bhattacharya, N. S. Caswell, S. Kumaran, A. Nigam, and F. Y. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems J.*, 46(4):703–721, 2007. (Cited on page 1.)
- [12] K. Bhattacharya, C. Gerede, R. Hull, R. Liu, and J. Su. Towards formal analysis of artifact-centric business process models. In *Proc. of the 5th Int. Conf. on Business Process Management (BPM)*, volume 4714 of *Lecture Notes in Computer Science*, pages 288–234. Springer, 2007. (Cited on page 1.)
- [13] Kamal Bhattacharya, Robert Guttman, Kelly Lyman, Fenno F. Heath, Santhosh Kumaran, Prabir Nandi, Frederick Y. Wu, Prasanna Athma, Christoph Freiberg, Lars Johannsen, and Andreas Staudt. A model-driven approach to industrializing discovery processes in pharmaceutical research. *IBM Systems J.*, 44(1):145–162, 2005. (Cited on page 1.)
- [14] Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI)*, pages 705–711, 2012. (Cited on page 2.)
- [15] Julian Bradfield and Colin Stirling. Modal mu-calculi. In *Handbook of Modal Logic*, volume 3, pages 721–756. Elsevier, 2007. (Cited on page 4.)
- [16] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007. (Cited on page 3.)
- [17] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. EQL-Lite: Effective first-order query processing in description logics. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 274–279, 2007. (Cited on page 4.)
- [18] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodríguez-Muro, and Riccardo Rosati. Ontologies and databases: The *DL-Lite* approach. In Sergio Tessaris and Enrico Franconi, editors, *Reasoning Web. Semantic Technologies for Informations Systems – 5th Int. Summer School Tutorial Lectures (RW)*, volume 5689 of *Lecture Notes in Computer Science*, pages 255–356. Springer, 2009. (Cited on page 6.)
- [19] Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Dmitriy Zheleznyakov. Evolution of *DL-Lite* knowledge bases. In *Proc. of the 9th Int. Semantic Web Conf. (ISWC)*, volume 6496 of *Lecture Notes in Computer Science*, pages 112–128. Springer, 2010. (Cited on pages 4 and 5.)
- [20] Diego Calvanese, Evgeny Kharlamov, Marco Montali, and Dmitriy Zheleznyakov. Inconsistency tolerance in OWL 2 QL knowledge and action bases - statement of interest. In *Proc. of the 9th Int. Workshop on OWL: Experiences and Directions (OWLED)*, volume 849 of *CEUR Electronic Workshop Proceedings*, <http://ceur-ws.org/>, 2012. (Cited on page 2.)

- [21] Diego Calvanese, Babak Bagheri Hariri, Riccardo De Masellis, Domenico Lembo, Marco Montali, Ario Santoso, Dimitry Solomakhin, and Sergio Tessaris. Techniques and tools for KAB, to manage action linkage with the Artifact Layer – Iteration 2. Deliverable ACSI-D2.4.2, ACSI Consortium, May 2013. (Cited on page 7.)
- [22] Diego Calvanese, Giuseppe De Giacomo, Marco Montali, and Fabio Patrizi. Verification and synthesis in description logic based dynamic systems. In *Proc. of the 7th Int. Conf. on Web Reasoning and Rule Systems (RR)*, volume 7994 of *Lecture Notes in Computer Science*, pages 50–64. Springer, 2013. (Cited on pages 2 and 3.)
- [23] Piero Cangialosi, Giuseppe De Giacomo, Riccardo De Masellis, and Riccardo Rosati. Conjunctive artifact-centric services. In *Proc. of the 8th Int. Joint Conf. on Service Oriented Computing (ICSOC)*, volume 6470 of *Lecture Notes in Computer Science*, pages 318–333. Springer, 2010. (Cited on page 1.)
- [24] Tian Chao, David Cohn, Adrian Flatgard, Sandy Hahn, Mark Linehan, Prabir Nandi, Anil Nigam, Florian Pinel, John Vergo, and Frederick y Wu. Artifact-based transformation of IBM Global Financing, a case study. In *Proc. of the 7th Int. Conf. on Business Process Management (BPM)*, *Lecture Notes in Computer Science*. Springer, 2009. (Cited on page 1.)
- [25] Edmund M. Clarke and Jeannette M. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys*, 28(4):626–643, 1996. (Cited on page 2.)
- [26] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. The MIT Press, Cambridge, MA, USA, 1999. (Cited on page 4.)
- [27] David Cohn and Richard 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. (Cited on page 1.)
- [28] Elio Damaggio, Richard Hull, and Roman Vaculín. On the equivalence of incremental and fixpoint semantics for business artifacts with Guard-Stage-Milestone lifecycles. *Information Systems*, 38(4):561–584, 2013. (Cited on pages 2 and 7.)
- [29] Alin Deutsch, Richard Hull, Fabio Patrizi, and Victor Vianu. Automatic verification of data-centric business processes. In *Proc. of the 12th Int. Conf. on Database Theory (ICDT)*, pages 252–267, 2009. (Cited on page 2.)
- [30] Marlon Dumas. Integrated data and process management: Finally? In *Proc. of the 1st Int. Workshop on Knowledge-intensive Business Processes (KiBP)*, volume 861 of *CEUR Electronic Workshop Proceedings*, <http://ceur-ws.org/>, pages 21–32, 2012. (Cited on page 1.)
- [31] 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. (Cited on pages 1 and 2.)
- [32] C. E. Gerede, K. Bhattacharya, and J. Su. Static analysis of business artifact-centric operational models. In *Proc. of the 5th Int. Conf. on Service Oriented Computing (ICSOC)*, 2007. (Cited on pages 1 and 2.)
- [33] Babak Bagheri Hariri, Diego Calvanese, Marco Montali, Giuseppe De Giacomo, Riccardo De Masellis, and Paolo Felli. Description logic Knowledge and Action Bases. *J. of Artificial Intelligence Research*, 46:651–686, 2013. ISSN 1076-9757. doi: 10.1613/jair.3826. (Cited on pages 1, 2, 3, and 4.)
- [34] Richard Hull. Artifact-centric business process models: Brief survey of research results and challenges. In *Proc. of the 7th Int. Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, volume 5332 of *Lecture Notes in Computer Science*, pages 1152–1163. Springer, 2008. doi: 10.1007/978-3-540-88873-4\_17. (Cited on pages 1 and 2.)
- [35] Richard Hull, Elio Damaggio, Riccardo De Masellis, Fabiana Fournier, Manmohan Gupta, Fenno Terry Heath, III, Stacy Hobson, Mark Linehan, Sridhar Maradugu, Anil Nigam, Piwadee Noi Sukaviriya, and Roman Vaculin. Business artifacts with Guard-Stage-Milestone lifecycles: Managing artifact interactions with conditions and events. In *Proc. of the 5th ACM Int. Conf. on Distributed Event-Based Systems (DEBS 2011)*, pages 51–62, 2011. (Cited on pages 2 and 7.)
- [36] Domenico Lembo and Marco Ruzzi. Consistent query answering over description logic ontologies. In *Proc. of the 1st Int. Conf. on Web Reasoning and Rule Systems (RR)*, 2007. (Cited on page 2.)
- [37] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of the 4th Int. Conf. on Web Reasoning and Rule Systems (RR)*, pages 103–117, 2010. (Cited on pages 4 and 5.)
- [38] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Query rewriting for inconsistent *DL-Lite* ontologies. In *Proc. of the 5th Int. Conf. on Web Reasoning and Rule Systems (RR)*, 2011. (Cited on page 4.)
- [39] H. J. Levesque, R. Reiter, Y. Lesperance, F. Lin, and R. Scherl. GOLOG: A logic programming language for dynamic domains. *J. of Logic Programming*, 31:59–84, 1997. (Cited on page 4.)
- [40] Andreas Meyer, Sergey Smirnov, and Mathias Weske. *Data in business processes*. Universitätsverlag Potsdam, 2011. (Cited on page 1.)
- [41] Marco Montali, Diego Calvanese, and Giuseppe De Giacomo. Verification of data-aware commitment-based multiagent systems. In *Proc. of the 13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 157–164, 2014. ISBN 978-1-4503-2738-1. (Cited on pages 2 and 3.)
- [42] A. Nigam and N. S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems J.*, 42(3):428–445, 2003. (Cited on page 1.)
- [43] Manfred Reichert. Process and data: Two sides of the same coin? In *Proc. of the On the Move Confederated Int. Conf. (OTM 2012)*, volume 7565 of *Lecture Notes in Computer Science*, pages 2–19. Springer, 2012. doi: 10.1007/978-3-642-33606-5\_2. (Cited on page 1.)
- [44] Clay Richardson. Warning: Don’t assume your business processes use master data. In *Proc. of the 8th Int. Conf. on Business Process Management (BPM)*, volume 6336 of *Lecture Notes in Computer Science*, pages 11–12. Springer, 2010. (Cited on page 1.)