Towards Efficient Evolving Multi-Context Systems (Preliminary Report)

Ricardo Gonçalves and **Matthias Knorr** and **João Leite** [1]

Abstract. Managed Multi-Context Systems (mMCSs) provide a general framework for integrating knowledge represented in heterogeneous KR formalisms. Recently, evolving Multi-Context Systems (eMCSs) have been introduced as an extension of mMCSs that add the ability to both react to, and reason in the presence of commonly temporary dynamic observations, and evolve by incorporating new knowledge. However, the general complexity of such an expressive formalism may simply be too high in cases where huge amounts of information have to be processed within a limited short amount of time, or even instantaneously. In this paper, we investigate under which conditions eMCSs may scale in such situations and we show that such polynomial eMCSs can be applied in a practical use case.

1 Introduction

Multi-Context Systems (MCSs) were introduced in [7], building on the work in [16, 27], to address the need for a general framework that integrates knowledge bases expressed in heterogeneous KR formalisms. Intuitively, instead of designing a unifying language (see e.g., [17, 26], and [23] with its reasoner NoHR [22]) to which other languages could be translated, in an MCS the different formalisms and knowledge bases are considered as modules, and means are provided to model the flow of information between them (cf. [1, 21, 24] and references therein for further motivation on hybrid languages and their connection to MCSs).

More specifically, an MCS consists of a set of contexts, each of which is a knowledge base in some KR formalism, such that each context can access information from the other contexts using so-called bridge rules. Such non-monotonic bridge rules add its head to the context's knowledge base provided the queries (to other contexts) in the body are successful. Managed Multi-Context Systems (mMCSs) were introduced in [8] to provide an extension of MCSs by allowing operations, other than simple addition, to be expressed in the heads of bridge rules. This allows mMCSs to properly deal with the problem of consistency management within contexts.

One recent challenge for KR languages is to shift from static application scenarios which assume a one-shot computation, usually triggered by a user query, to open and dynamic scenarios where there is a need to react and evolve in the presence of incoming information. Examples include EVOLP [2], Reactive ASP [14, 13], C-SPARQL [5], Ontology Streams [25] and ETALIS [3], to name only a few.

Whereas mMCSs are quite general and flexible to address the problem of integration of different KR formalisms, they are essentially static in the sense that the contexts do not evolve to incorporate the changes in the dynamic scenarios. In such scenarios, new knowledge and information is dynamically produced, often from several different sources – for example a stream of raw data produced by some sensors, new ontological axioms written by some user, newly found exceptions to some general rule, etc.

To address this issue, two recent frameworks, evolving Multi-Context Systems (eMCSs) [19] and reactive Multi-Context Systems (rMCSs) [6, 12, 9] have been proposed sharing the broad motivation of designing general and flexible frameworks inheriting from mMCSs the ability to integrate and manage knowledge represented in heterogeneous KR formalisms, and at the same time be able to incorporate knowledge obtained from dynamic observations.

Whereas some differences set eMCSs and rMCSs apart (see related work in Sec. 6), the definition of eMCSs is presented in a more general way. That, however, means that, as shown in [19], the worst-case complexity is in general high, which may be problematic in dynamic scenarios where the overall system needs to evolve and react interactively. This is all the more true for huge amounts of data – for example raw sensor data is likely to be constantly produced in large quantities – and systems that are capable of processing and reasoning with such data are required.

At the same time, eMCSs inherit from MCSs the property that models, i.e., equilibria, may be non-minimal, which potentially admits that certain pieces of information are considered true based solely on self-justification. As argued in [7], minimality may not always be desired, which can in principle be solved by indicating for each context whether it requires minimality or not. Yet, avoiding self-justifications for those contexts where minimality is desired has not been considered in eMCSs.

In this paper, we tackle these problems and, in particular, consider under which conditions reasoning with evolving Multi-Context Systems can be done in polynomial time. For that purpose, we base our work on a number of notions studied in the context of MCSs that solve these problems in this case [7]. Namely, we adapt the notions of minimal and grounded equilibria to eMCSs, and subsequently a well-founded semantics, which indeed paves the way to the desired result.

The remainder of this paper is structured as follows. After introducing the main concepts regarding mMCSs in Sect. 2, in Sect. 3 we recall with more detail the framework of eMCSs already introducing adjustments to achieve polynomial reasoning. Then, in Sect. 4 we present an example use case, before we adapt and generalize notions from MCSs in Sect. 5 as outlined. We conclude in Sect. 6 with discussing related work and possible future directions.

---

[1] CENTRIA & Departamento de Informática, Faculdade Ciências e Tecnologia, Universidade Nova de Lisboa, email: rjrg@fct.unl.pt

2 Preliminaries: Managed Multi-Context Systems

Following [7], a multi-context system (MCS) consists of a collection of components, each of which contains knowledge represented in some *logic*, defined as a triple $L = \langle \mathbf{KB}, \mathbf{BS}, \mathbf{ACC} \rangle$ where $\mathbf{KB}$ is the set of well-formed knowledge bases of L, $\mathbf{BS}$ is the set of possible belief sets, and $\mathbf{ACC} : \mathbf{KB} \to 2^{\mathbf{BS}}$ is a function describing the semantics of L by assigning to each knowledge base a set of acceptable belief sets. We assume that each element of $\mathbf{KB}$ and $\mathbf{BS}$ is a set, and define $F = \{s : s \in kb \wedge kb \in \mathbf{KB}\}$.

In addition to the knowledge base in each component, *bridge rules* are used to interconnect the components, specifying what knowledge to assert in one component given certain beliefs held in the components of the MCS. Bridge rules in MCSs only allow adding information to the knowledge base of their corresponding context. In [8], an extension, called managed Multi-Context Systems (mMCSs), is introduced in order to allow other types of operations to be performed on a knowledge base. For that purpose, each context of an mMCS is associated with a *management base*, which is a set of operations that can be applied to the possible knowledge bases of that context. Given a management base OP and a logic L, let $OF = \{op(s) : op \in OP \wedge s \in F\}$ be the *set of operational formulas* that can be built from OP and F. Each context of an mMCS gives semantics to operations in its management base using a *management function* over a logic L and a management base OP, $mng : 2^{OF} \times \mathbf{KB} \to \mathbf{KB}$, i.e., $mng(op, kb)$ is the knowledge base that results from applying the operations in op to the knowledge base kb. Note that this is already a specific restriction in our case, as mng commonly returns a (non-empty) set of possible knowledge bases for mMCS (and eMCS). We also assume that $mng(\emptyset, kb) = kb$. Now, for a sequence of logics $L = \langle L_1, \dots, L_n \rangle$ and a management base OP_i, an L_i-*bridge rule* σ over L, $1 \leq i \leq n$, is of the form $H(\sigma) \leftarrow B(\sigma)$ where $H(\sigma) \in OF_i$ and $B(\sigma)$ is a set of *bridge literals* of the forms $(r : b)$ and $\mathbf{not}\,(r : b)$, $1 \leq r \leq n$, with b a belief formula of L_r.

A *managed Multi-Context System* (mMCS) is a sequence $M = \langle C_1, \dots, C_n \rangle$, where each C_i, $i \in \{1, \dots, n\}$, called a *managed context*, is defined as $C_i = \langle L_i, kb_i, br_i, OP_i, mng_i \rangle$ where $L_i = \langle \mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i \rangle$ is a logic, $kb_i \in \mathbf{KB}_i$, br_i is a set of L_i-bridge rules, OP_i is a management base, and mng_i is a management function over L_i and OP_i. Note that, for the sake of readability, we consider a slightly restricted version of mMCSs where $\mathbf{ACC}_i$ is still a function and not a set of functions as for logic suites [8].

For an mMCS $M = \langle C_1, \dots, C_n \rangle$, a *belief state of M* is a sequence $S = \langle S_1, \dots, S_n \rangle$ such that each S_i is an element of $\mathbf{BS}_i$. For a bridge literal $(r : b)$, $S \models (r : b)$ if $b \in S_r$ and $S \models \mathbf{not}\,(r : b)$ if $b \notin S_r$; for a set of bridge literals B, $S \models B$ if $S \models L$ for every $L \in B$. We say that a bridge rule σ of a context C_i is *applicable* given a belief state S of M if S satisfies $B(\sigma)$. We can then define $app_i(S)$, the set of heads of bridge rules of C_i which are applicable in S, by setting $app_i(S) = \{H(\sigma) : \sigma \in br_i \wedge S \models B(\sigma)\}$.

Equilibria are belief states that simultaneously assign an acceptable belief set to each context in the mMCS such that the applicable operational formulas in bridge rule heads are taken into account. Formally, a belief state $S = \langle S_1, \dots, S_n \rangle$ of an mMCS M is an *equilibrium* of M if, for every $1 \leq i \leq n$, $S_i \in \mathbf{ACC}_i(mng_i(app_i(S), kb_i))$.

3 Evolving Multi-Context Systems

In this section, we recall evolving Multi-Context Systems as introduced in [19] including some alterations that are in line with our intentions to achieve polynomial reasoning. As indicated in [19], we consider that some of the contexts in the MCS become so-called *observation contexts* whose knowledge bases will be constantly changing over time according to the observations made, similar, e.g., to streams of data from sensors.[2]

The changing observations then will also affect the other contexts by means of the bridge rules. As we will see, such effect can either be instantaneous and temporary, i.e., limited to the current time instant, similar to (static) mMCSs, where the body of a bridge rule is evaluated in a state that already includes the effects of the operation in its head, or persistent, but only affecting the next time instant. To achieve the latter, we extend the operational language with a unary meta-operation $next$ that can only be applied on top of operations.

Definition 1 *Given a management base OP and a logic L, we define eOF, the evolving operational language, as $eOF = OF \cup \{next(op(s)) : op(s) \in OF\}$.*

We can now define evolving Multi-Context Systems.

Definition 2 *An* evolving Multi-Context System (eMCS) *is a sequence $M_e = \langle C_1, \dots, C_n \rangle$, where each* evolving context C_i, $i \in \{1, \dots, n\}$ *is defined as $C_i = \langle L_i, kb_i, br_i, OP_i, mng_i \rangle$ where*

- $L_i = \langle \mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i \rangle$ *is a logic*
- $kb_i \in \mathbf{KB}_i$
- br_i *is a set of L_i-bridge rules s.t. $H(\sigma) \in eOF_i$*
- OP_i *is a management base*
- mng_i *is a management function over L_i and OP_i.*

As already outlined, evolving contexts can be divided into regular *reasoning contexts* and special *observation contexts* that are meant to process a stream of observations which ultimately enables the entire eMCS to react and evolve in the presence of incoming observations. To ease the reading and simplify notation, w.l.o.g., we assume that the first ℓ contexts, $0 \leq \ell \leq n$, in the sequence $\langle C_1, \dots, C_n \rangle$ are observation contexts, and, whenever necessary, such an eMCS can be explicitly represented by $\langle C_1^o, \dots, C_\ell^o, C_{\ell+1}, \dots, C_n \rangle$.

As for mMCSs, a *belief state for M_e* is a sequence $S = \langle S_1, \dots, S_n \rangle$ such that, for each $1 \leq i \leq n$, we have $S_i \in \mathbf{BS}_i$.

Recall that the heads of bridge rules in an eMCS are more expressive than in an mMCS, since they may be of two types: those that contain $next$ and those that do not. As already mentioned, the former are to be applied to the current knowledge base and not persist, whereas the latter are to be applied in the next time instant and persist. Therefore, we distinguish these two subsets.

Definition 3 *Let $M_e = \langle C_1, \dots, C_n \rangle$ be an eMCS and S a belief state for M_e. Then, for each $1 \leq i \leq n$, consider the following sets:*
- $app_i^{next}(S) = \{op(s) : next(op(s)) \in app_i(S)\}$
- $app_i^{now}(S) = \{op(s) : op(s) \in app_i(S)\}$

Note that if we want an effect to be instantaneous and persistent, then this can also be achieved using two bridge rules with identical body, one with and one without $next$ in the head.

Similar to equilibria in mMCS, the (static) equilibrium is defined to incorporate instantaneous effects based on $app_i^{now}(S)$ alone.

---

[2] For simplicity of presentation, we consider discrete steps in time here.

Definition 4 *Let* $M_e = \langle C_1, \ldots, C_n \rangle$ *be an eMCS. A belief state* $S = \langle S_1, \ldots, S_n \rangle$ *for* M_e *is a static equilibrium of* M_e *iff, for each* $1 \leq i \leq n$, *we have* $S_i \in \mathbf{ACC}_i(mng_i(app_i^{now}(S), kb_i))$.

Note the minor change due to mng now only returning one kb.

To be able to assign meaning to an eMCS evolving over time, we introduce evolving belief states, which are sequences of belief states, each referring to a subsequent time instant.

Definition 5 *Let* $M_e = \langle C_1, \ldots, C_n \rangle$ *be an eMCS. An* evolving belief state *of size s for* M_e *is a sequence* $S_e = \langle S^1, \ldots, S^s \rangle$ *where each* S^j, $1 \leq j \leq s$, *is a belief state for* M_e.

To enable an eMCS to react to incoming observations and evolve, an observation sequence defined in the following has to be processed. The idea is that the knowledge bases of the observation contexts C_i^o change according to that sequence.

Definition 6 *Let* $M_e = \langle C_1^o, \ldots, C_\ell^o, C_{\ell+1}, \ldots, C_n \rangle$ *be an eMCS. An* observation sequence *for* M_e *is a sequence* $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$, *such that, for each* $1 \leq j \leq m$, $\mathcal{O}^j = \langle o_1^j, \ldots, o_\ell^j \rangle$ *is an* instant observation *with* $o_i^j \in \mathbf{KB}_i$ *for each* $1 \leq i \leq \ell$.

To be able to update the knowledge bases in the evolving contexts, we need one further notation. Given an evolving context C_i and $k \in \mathbf{KB}_i$, we denote by $C_i[k]$ the evolving context in which kb_i is replaced by k, i.e., $C_i[k] = \langle L_i, k, br_i, OP_i, mng_i \rangle$.

We can now define that certain evolving belief states are evolving equilibria of an eMCS $M_e = \langle C_1^o, \ldots, C_\ell^o, C_{\ell+1}, \ldots, C_n \rangle$ given an observation sequence $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$ for M_e. The intuitive idea is that, given an evolving belief state $S_e = \langle S^1, \ldots, S^s \rangle$ for M_e, in order to check if S_e is an evolving equilibrium, we need to consider a sequence of eMCSs, $M^1, \ldots, M^s$ (each with ℓ observation contexts), representing a possible evolution of M_e according to the observations in Obs, such that S^j is a (static) equilibrium of M^j. The knowledge bases of the observation contexts in M^j are exactly their corresponding elements o_i^j in $\mathcal{O}^j$. For each of the other contexts C_i, $\ell + 1 \leq i \leq n$, its knowledge base in M^j is obtained from the one in M^{j-1} by applying the operations in $app_i^{next}(S^{j-1})$.

Definition 7 *Let* $M_e = \langle C_1^o, \ldots, C_\ell^o, C_{\ell+1}, \ldots, C_n \rangle$ *be an eMCS,* $S_e = \langle S^1, \ldots, S^s \rangle$ *an evolving belief state of size s for* M_e, *and* $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$ *an observation sequence for* M_e *such that* $m \geq s$. *Then,* S_e *is an* evolving equilibrium *of size s of* M_e *given Obs iff, for each* $1 \leq j \leq s$, S^j *is an equilibrium of* $M^j = \langle C_1^o[o_1^j], \ldots, C_\ell^o[o_\ell^j], C_{\ell+1}[k_{\ell+1}^j], \ldots, C_n[k_n^j] \rangle$ *where, for each* $\ell + 1 \leq i \leq n$, k_i^j *is defined inductively as follows:*

- $k_i^1 = kb_i$
- $k_i^{j+1} = mng_i(app_i^{next}(S^j), k_i^j)$

Note that $next$ in bridge rule heads of observation contexts are thus without any effect, in other words, observation contexts can indeed be understood as managed contexts whose knowledge base changes with each time instant.

The essential difference to [19] is that the k_i^{j+1} can be effectively computed (instead of picking one of several options), simply because mng always returns one knowledge base. The same applies in Def. 4.

As shown in [19], two consequences of the previous definitions are that any subsequence of an evolving equilibrium is also an evolving equilibrium, and mMCSs are a particular case of eMCSs.

4 Use Case Scenario

In this section, we illustrate eMCSs adapting a scenario on cargo shipment assessment taken from [32].

The customs service for any developed country assesses imported cargo for a variety of risk factors including terrorism, narcotics, food and consumer safety, pest infestation, tariff violations, and intellectual property rights.[3] Assessing this risk, even at a preliminary level, involves extensive knowledge about commodities, business entities, trade patterns, government policies and trade agreements. Some of this knowledge may be external to a given customs agency: for instance the broad classification of commodities according to the international Harmonized Tariff System (HTS), or international trade agreements. Other knowledge may be internal to a customs agency, such as lists of suspected violators or of importers who have a history of good compliance with regulations. While some of this knowledge is relatively stable, much of it changes rapidly. Changes are made not only at a specific level, such as knowledge about the expected arrival date of a shipment; but at a more general level as well. For instance, while the broad HTS code for tomatoes (0702) does not change, the full classification and tariffs for cherry tomatoes for import into the US changes seasonally.

Here, we consider an eMCS $M_e = \langle C_1^o, C_2^o, C_3, C_4 \rangle$ composed of two observation contexts C_1^o and C_2^o, and two reasoning contexts C_3 and C_4. The first observation context is used to capture the data of passing shipments, i.e., the country of their origination, the commodity they contain, their importers and producers. Thus, the knowledge base and belief set language of C_1^o is composed of all the ground atoms over ShpmtCommod/2, ShpmtDeclHTSCode/2, ShpmtImporter/2, ShpmtCountry/2, ShpmtProducer/2, and also GrapeTomato/1 and CherryTomato/1. The second observation context C_2^o serves to insert administrative information and data from other institutions. Its knowledge base and belief set language is composed of all the ground atoms over NewEUMember/1, Misfiling/1, and RandomInspection/1. Neither of the two observation contexts has any bridge rules.

The reasoning context C_3 is an ontological Description Logic (DL) context that contains a geographic classification, along with information about producers who are located in various countries. It also contains a classification of commodities based on their harmonized tariff information (HTS chapters, headings and codes, cf. http://www.usitc.gov/tata/hts). We refer to [11] and [8] for the standard definition of L_3; kb_3 is given as follows:

Commodity $\equiv (\exists\mathsf{HTSCode}.\top)$
EdibleVegetable $\equiv (\exists\mathsf{HTSChapter}. \{ \text{'07'} \})$
CherryTomato $\equiv (\exists\mathsf{HTSCode}. \{ \text{'07020020'} \})$
Tomato $\equiv (\exists\mathsf{HTSHeading}. \{ \text{'0702'} \})$
GrapeTomato $\equiv (\exists\mathsf{HTSCode}. \{ \text{'07020010'} \})$
CherryTomato $\sqsubseteq$ Tomato   CherryTomato $\sqcap$ GrapeTomato $\sqsubseteq \bot$
GrapeTomato $\sqsubseteq$ Tomato   Tomato $\sqsubseteq$ EdibleVegetable
EURegisteredProducer $\equiv (\exists\mathsf{RegisteredProducer.EUCountry})$
LowRiskEUCommodity $\equiv (\exists\mathsf{ExpeditableImporter}.\top)\sqcap$
$\qquad\qquad\qquad (\exists\mathsf{CommodCountry.EUCountry})$
EUCountry($portugal$)   RegisteredProducer($p_1, portugal$)
EUCountry($slovakia$)   RegisteredProducer($p_2, slovakia$)

OP_3 contains a single add operation to add factual knowledge. The bridge rules br_3 are given as follows:

---

[3] The system described here is not intended to reflect the policies of any country or agency.

$add(\mathsf{CherryTomato}(\mathbf{x})) \leftarrow (1\!:\!\mathsf{CherryTomato}(\mathbf{x}))$
$add(\mathsf{GrapeTomato}(\mathbf{x})) \leftarrow (1\!:\!\mathsf{GrapeTomato}(\mathbf{x}))$
$next(add(\mathsf{EUCountry}(\mathbf{x}))) \leftarrow (2\!:\!\mathsf{NewEUMember}(\mathbf{x}))$
$add(\mathsf{CommodCountry}(\mathbf{x},\mathbf{y})) \leftarrow (1\!:\!\mathsf{ShpmtCommod}(\mathbf{z},\mathbf{x})),$
$\qquad (1\!:\!\mathsf{ShpmtCountry}(\mathbf{z},\mathbf{y}))$
$add(\mathsf{ExpeditableImporter}(\mathbf{x},\mathbf{y})) \leftarrow (1\!:\!\mathsf{ShpmtCommod}(\mathbf{z},\mathbf{x})),$
$\qquad (1\!:\!\mathsf{ShpmtImporter}(\mathbf{z},\mathbf{y})), (4\!:\!\mathsf{AdmissibleImporter}(\mathbf{y})),$
$\qquad (4\!:\!\mathsf{ApprovedImporterOf}(\mathbf{y},\mathbf{x}))$

Note that kb_3 can indeed be expressed in the DL $\mathcal{EL}^{++}$ [4] for which standard reasoning tasks, such as subsumption, can be computed in PTIME.

Finally, C_4 is a logic programming (LP) indicating information about importers, and about whether to inspect a shipment either to check for compliance of tariff information or for food safety issues. For L_4 we consider that $\mathbf{KB}_i$ the set of normal logic programs over a signature Σ, $\mathbf{BS}_i$ is the set of atoms over Σ, and $\mathbf{ACC}_i(kb)$ returns returns a singleton set containing only the set of true atoms in the unique well-founded model. The latter is a bit unconventional, since this way undefinedness under the well-founded semantics [15] is merged with false information. However, as long as no loops over negation occur in the LP context (in combination with its bridge rules), undefinedness does not occur, and the obvious benefit of this choice is that computing the well-founded model is PTIME-data-complete [10]. We consider $OP_4 = OP_3$, and kb_4 and br_4 are given as follows:

$\mathsf{AdmissibleImporter}(\mathbf{x}) \leftarrow \sim\!\mathsf{SuspectedBadGuy}(\mathbf{x}).$
$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{RandomInspection}(\mathbf{x}).$
$\mathsf{FullInspection}(\mathbf{x}) \leftarrow \sim\!\mathsf{CompliantShpmt}(\mathbf{x}).$
$\mathsf{SuspectedBadGuy}(i_1).$

$next((\mathsf{SuspectedBadGuy}(\mathbf{x})) \leftarrow (2\!:\!\mathsf{Misfiling}(\mathbf{x}))$
$add(\mathsf{ApprovedImporterOf}(i_2,\mathbf{x})) \leftarrow (3\!:\!\mathsf{EdibleVegetable}(\mathbf{x}))$
$add(\mathsf{ApprovedImporterOf}(i_3,\mathbf{x})) \leftarrow (1\!:\!\mathsf{GrapeTomato}(\mathbf{x}))$
$add(\mathsf{CompliantShpmt}(\mathbf{x})) \leftarrow (1\!:\!\mathsf{ShpmtCommod}(\mathbf{x},\mathbf{y})),$
$\qquad (3\!:\!\mathsf{HTSCode}(\mathbf{y},\mathbf{z})), (1\!:\!\mathsf{ShpmtDeclHTSCode}(\mathbf{x},\mathbf{z}))$
$add(\mathsf{RandomInspection}(\mathbf{x})) \leftarrow (1\!:\!\mathsf{ShpmtCommod}(\mathbf{x},\mathbf{y})),$
$\qquad (2\!:\!\mathsf{Random}(\mathbf{y}))$
$add(\mathsf{PartialInspection}(\mathbf{x})) \leftarrow (1\!:\!\mathsf{ShpmtCommod}(\mathbf{x},\mathbf{y})),$
$\qquad \mathbf{not}\,(3\!:\!\mathsf{LowRiskEUCommodity}(\mathbf{y}))$
$add(\mathsf{FullInspection}(\mathbf{x})) \leftarrow (1\!:\!\mathsf{ShpmtCommod}(\mathbf{x},\mathbf{y})),$
$\qquad (3\!:\!\mathsf{Tomato}(\mathbf{y})), (1\!:\!\mathsf{ShpmtCountry}(\mathbf{x},slovakia))$

Now consider the observation sequence $Obs = \langle \mathcal{O}^1, \mathcal{O}^2, \mathcal{O}^3 \rangle$ where o_1^1 consists of the following atoms on s_1 (where s in s_1 stands for shipment, c for commodity, and i for importer):

$\mathsf{ShpmtCommod}(s_1, c_1)$	$\mathsf{ShpmtDeclHTSCode}(s_1, \text{'07020010'})$
$\mathsf{ShpmtImporter}(s_1, i_1)$	$\mathsf{CherryTomato}(c_1)$

o_1^2 of the following atoms on s_2:

$\mathsf{ShpmtCommod}(s_2, c_2)$	$\mathsf{ShpmtDeclHTSCode}(s_2, \text{'07020020'})$
$\mathsf{ShpmtImporter}(s_2, i_2)$	$\mathsf{ShpmtCountry}(s_2, portugal)$
$\mathsf{CherryTomato}(c_2)$	

and o_1^3 of the following atoms on s_3:

$\mathsf{ShpmtCommod}(s_3, c_3)$	$\mathsf{ShpmtDeclHTSCode}(s_3, \text{'07020010'})$
$\mathsf{ShpmtImporter}(s_3, i_3)$	$\mathsf{ShpmtCountry}(s_3, portugal)$
$\mathsf{GrapeTomato}(c_3)$	$\mathsf{ShpmtProducer}(s_3, p_1)$

while $o_2^1 = o_2^3 = \emptyset$ and $o_2^2 = \{\mathsf{Misfiling}(i_3)\}$. Then, an evolving equilibrium of size 3 of M_e given Obs is the sequence $S_e = \langle S^1, S^2, S^3 \rangle$ such that, for each $1 \leq j \leq 3$, $S^j = \langle S_1^j, S_2^j, S_3^j, S_4^j \rangle$. Since it is not feasible to present the entire S_e, we just highlight some interesting parts related to the evolution of the system. E.g., we have that $\mathsf{FullInspection}(s_1) \in S_4^1$ since the HTS code does not correspond to the cargo; no inspection on s_2 in S_4^2 since the shipment is compliant, c_2 is a EU commodity, and s_2 was not picked for random inspection; and $\mathsf{PartialInspection}(s_3) \in S_4^3$, even though s_3 comes from a EU country, because i_3 has been identified at time instant 2 for misfiling, which has become permanent info available at time 3.

5 Grounded Equilibria and Well-founded Semantics

Even if we only consider MCSs M, which are static and where an implicit mng always returns precisely one knowledge base, such that reasoning in all contexts can be done in PTIME, then deciding whether M has an equilibrium is in NP [7, 8]. The same result necessarily also holds for eMCSs, which can also be obtained from the considerations on eMCSs [19].

A number of special notions were studied in the context of MCSs that tackle this problem [7]. In fact, the notion of minimal equilibria was introduced with the aim of avoiding potential self-justifications. Then, grounded equilibria as a special case for so-called reducible MCSs were presented for which the existence of minimal equilibria can be effectively checked. Subsequently, a well-founded semantics for such reducible MCSs was defined under which an approximation of all grounded equilibria can be computed more efficiently. In the following, we transfer these notions from static MCSs in [7] to dynamic eMCSs and discuss under which (non-trivial) conditions they can actually be applied.

Given an eMCS $M_e = \langle C_1, \ldots, C_n \rangle$, we say that a static equilibrium $S = \langle S_1, \ldots, S_n \rangle$ is *minimal* if there is no equilibrium $S' = \langle S_1', \ldots, S_n' \rangle$ such that $S_i' \subseteq S_i$ for all i with $1 \leq i \leq n$ and $S_j' \subsetneq S_j$ for some j with $1 \leq j \leq n$.

This notion of minimality ensures the avoidance of self-justifications in evolving equilibria. The problem with this notion in terms of computation is that such minimization in general adds an additional level in the polynomial hierarchy. Therefore, we now formalize conditions under which minimal equilibria can be effectively checked. The idea is that the grounded equilibrium will be assigned to an eMCS M_e if all the logics of all its contexts can be reduced to special monotonic ones using a so-called reduction function. In the case where the logics of all contexts in M_e turn out to be monotonic, the minimal equilibrium will be unique.

Formally, a logic $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ is *monotonic* if

1. $\mathbf{ACC}(kb)$ is a singleton set for each $kb \in \mathbf{KB}$, and

2. $S \subseteq S'$ whenever $kb \subseteq kb'$, $\mathbf{ACC}(kb) = \{S\}$, and $\mathbf{ACC}(kb') = \{S'\}$.

Furthermore, $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ is *reducible* if for some $\mathbf{KB}^* \subseteq \mathbf{KB}$ and some reduction function $red : \mathbf{KB} \times \mathbf{BS} \to \mathbf{KB}^*$,

1. the restriction of L to $\mathbf{KB}^*$ is monotonic,

2. for each $kb \in \mathbf{KB}$, and all $S, S' \in \mathbf{BS}$:
   - $red(kb, S) = kb$ whenever $kb \in \mathbf{KB}^*$,
   - $red(kb, S) \subseteq red(kb, S')$ whenever $S' \subseteq S$,
   - $S \in \mathbf{ACC}(kb)$ iff $\mathbf{ACC}(red(kb, S)) = \{S\}$.

Then, an evolving context $C = (L, kb, br, OP, mng)$ is *reducible* if its logic L is reducible and, for all $op \in F_L^{OP}$ and all belief sets S, $red(mng(op, kb), S) = mng(op, red(kb, S))$.

An eMCS is *reducible* if all of its contexts are. Note that a context is reducible whenever its logic L is monotonic. In this case $\mathbf{KB}^*$ coincides with $\mathbf{KB}$ and red is the identity with respect to the first argument.

As pointed out in [7], reducibility is inspired by the reduct in (non-monotonic) answer set programming. The crucial and novel condition in our case is the one that essentially says that the reduction function red and the management function mng can be applied in an arbitrary order. This may restrict to some extent the sets of operations OP and mng, but in our use case scenario in Sect. 4, all contexts are indeed reducible.

A particular case of reducible eMCSs, definite eMCSs, does not require the reduction function and admits the polynomial computation of minimal evolving equilibria as we will see next. Namely, a reducible eMCS $M_e = \langle C_1, \ldots, C_n \rangle$ is *definite* if

1. none of the bridge rules in any context contains **not** ,
2. for all i and all $S \in \mathbf{BS}_i$, $kb_i = red_i(kb_i, S)$.

In a definite eMCS, bridge rules are monotonic, and knowledge bases are already in reduced form. Inference is thus monotonic and a unique minimal equilibrium exists. We take this equilibrium to be the grounded equilibrium. Let M_e be a definite eMCS. A belief state S of M_e is the *grounded equilibrium of M_e*, denoted by $\mathbf{GE}(M_e)$, if S is the unique minimal (static) equilibrium of M_e. This notion gives rise to evolving grounded equilibria.

Definition 8 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a definite eMCS, $S_e = \langle S^1, \ldots, S^s \rangle$ an evolving belief state of size s for M_e, and $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$ an observation sequence for M_e such that $m \geq s$. Then, S_e is the* evolving grounded equilibrium *of size s of M_e given Obs iff, for each $1 \leq j \leq s$, S^j is a grounded equilibrium of M^j defined as in Definition 7.*

Grounded equilibria for definite eMCSs can indeed be efficiently computed following [7]. The only additional requirement is that all operations $op \in OP$ are *monotonic*, i.e., for kb, we have that $kb \subseteq mng(op(s), kb)$. Note that this is indeed a further restriction and not covered by reducible eMCSs. Now, for $1 \leq i \leq n$, let $kb_i^0 = kb_i$ and define, for each successor ordinal $\alpha + 1$,

$$kb_i^{\alpha+1} = mng(app_i^{now}(E^\alpha), kb_i^\alpha),$$

where $E^\alpha = (E_1^\alpha, \ldots, E_n^\alpha)$ and $\mathbf{ACC}_i(kb_i^\alpha) = \{E_i^\alpha\}$. Furthermore, for each limit ordinal α, define $kb_i^\alpha = \bigcup_{\beta \leq \alpha} kb_i^\beta$, and let $kb_i^\infty = \bigcup_{\alpha > 0} kb_i^\alpha$. Then Proposition 1 [7] can be adapted:

Proposition 1 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a definite eMCS s.t. all OP_i are monotonic. A belief state $S = \langle S_1, \ldots, S_n \rangle$ is the grounded equilibrium of M_e iff $\mathbf{ACC}_i(kb_i^\infty) = \{S_i\}$, for $1 \leq i \leq n$.*

As pointed out in [7], for many logics, $kb_i^\infty = kb_i^\omega$ holds, i.e., the iteration stops after finitely many steps. This is indeed the case for the use case scenario in Sect. 4.

For evolving belief states S_e of size s and an observation sequence Obs for M_e, this proposition yields that the evolving grounded equilibrium for definite eMCSs can be obtained by simply applying this iteration s times.

Grounded equilibria for general eMCSs are defined based on a reduct which generalizes the Gelfond-Lifschitz reduct to the multi-context case:

Definition 9 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a reducible eMCS and $S = \langle S_1, \ldots, S_n \rangle$ a belief state of M_e. The S-reduct of M_e is defined as $M_e^S = \langle C_1^S, \ldots, C_n^S \rangle$ where, for each $C_i = \langle L_i, kb_i, br_i, OP_i, mng_i \rangle$, we define $C_i^S = (L_i, red_i(kb_i, S_i), br_i^S, OP_i, mng_i)$. Here, br_i^S results from br_i by deleting all*

1. *rules with $\mathbf{not}\,(r : p)$ in the body such that $S \models (r : p)$, and*

2. $\mathbf{not}$ *literals from the bodies of remaining rules.*

For each reducible eMCS M_e and each belief set S, the S-reduct of M_e is definite. We can thus check whether S is a grounded equilibrium in the usual manner:

Definition 10 *Let M_e be a reducible eMCS such that all OP_i are monotonic. A belief state S of M_e is a grounded equilibrium of M_e if S is the grounded equilibrium of M_e^S, that is $S = \mathbf{GE}(M_e^S)$.*

The following result generalizes Proposition 2 from [7].

Proposition 2 *Every grounded equilibrium of a reducible eMCS M_e such that all OP_i are monotonic is a minimal equilibrium of M_e.*

This can again be generalized to evolving grounded equilibria.

Definition 11 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OP_i are monotonic, $S_e = \langle S^1, \ldots, S^s \rangle$ an evolving belief state of size s for M_e, and $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$ an observation sequence for M_e such that $m \geq s$. Then, S_e is the* evolving grounded equilibrium *of size s of M_e given Obs iff, for each $1 \leq j \leq s$, S^j is the grounded equilibrium of $(M^j)^{S^j}$ with M^j defined as in Definition 7.*

This computation is still not polynomial, since, intuitively, we have to guess and check the (evolving) equilibrium, which is why the well-founded semantics for reducible eMCSs M_e is introduced following [7]. Its definition is based on the operator $\gamma_{M_e}(S) = \mathbf{GE}(M_e^S)$, provided $\mathbf{BS}_i$ for each logic L_i in all the contexts of M_e has a least element S^*. Such eMCSs are called *normal*.

The following result can be straightforwardly adopted from [7].

Proposition 3 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a reducible eMCS such that all OP_i are monotonic. Then γ_{M_e} is antimonotone.*

As usual, applying γ_{M_e} twice yields a monotonic operator. Hence, by the Knaster-Tarski theorem, $(\gamma_{M_e})^2$ has a least fixpoint which determines the well-founded semantics.

Definition 12 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OP_i are monotonic. The well-founded semantics of M_e, denoted $\mathbf{WFS}(M)$, is the least fixpoint of $(\gamma_{M_e})^2$.*

Starting with the least belief state $S^* = \langle S_1^*, \ldots, S_n^* \rangle$, this fixpoint can be iterated, and the following correspondence between $\mathbf{WFS}(M_e)$ and the grounded equilibria of M_e can be shown.

Proposition 4 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OP_i are monotonic, $\mathbf{WFS}(M_e) = \langle W_1, \ldots W_n \rangle$, and $S = \langle S_1, \ldots, S_n \rangle$ a grounded equilibrium of M_e. Then $W_i \subseteq S_i$ for $1 \leq i \leq n$.*

The well-founded semantics can thus be viewed as an approximation of the belief state representing what is accepted in all grounded

equilibria, even though $\mathbf{WFS}(M_e)$ may itself not necessarily be an equilibrium. Yet, if all $\mathbf{ACC}_i$ deterministically return one element of $\mathbf{BS}_i$ and the eMCS is acyclic (i.e., no cyclic dependencies over bridge rules exist between beliefs in the eMCS see [19]), then the grounded equilibrium is unique and identical to the well-founded semantics. This is indeed the case for the use case in Sect. 4.

As before, the well-founded semantics can be generalized to evolving belief states.

Definition 13 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OP_i are monotonic, and $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$ an observation sequence for M_e such that $m \geq s$. The* evolving well-founded semantics *of M_e, denoted $\mathbf{WFS}_e(M)$, is the evolving belief state $S_e = \langle S^1, \ldots, S^s \rangle$ of size s for M_e such that S^j is the well-founded semantics of M^j defined as in Definition 7.*

Finally, as intended, we can show that computing the evolving well-founded semantics of M_e can be done in polynomial time under the restrictions established so far. For analyzing the complexity in each time instant, we can utilize *output-projected* belief states [11]. The idea is to consider only those beliefs that appear in some bridge rule body. Formally, given an evolving context C_i within $M_e = \langle C_1, \ldots, C_n \rangle$, we can define OUT_i to be the set of all beliefs of C_i occurring in the body of some bridge rule in M_e. The *output-projection* of a belief state $S = \langle S_1, \ldots, S_n \rangle$ of M_e is the belief state $S' = \langle S'_1, \ldots, S'_n \rangle$, $S'_i = S_i \cap OUT_i$, for $1 \leq i \leq n$.

Following [11, 8], we can adapt the *context complexity* of C_i from [19] as the complexity of the following problem:

(CC) Decide, given $Op_i \subseteq OF_i$ and $S'_i \subseteq OUT_i$, if exist $kb'_i = mng_i(Op_i, kb_i)$ and $S_i \in \mathbf{ACC}_i(kb'_i)$ s.t. $S'_i = S_i \cap OUT_i$.

Problem (CC) can intuitively be divided into two subproblems: (MC) compute some $kb'_i = mng_i(Op_i, kb_i)$ and (EC) decide whether $S_i \in \mathbf{ACC}(kb'_i)$ exists s.t. $S'_i = S_i \cap OUT_i$. Here, (MC) is trivial for monotonic operations, so (EC) determines the complexity of (CC).

Theorem 1 *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OP_i are monotonic, $Obs = \langle \mathcal{O}^1, \ldots, \mathcal{O}^m \rangle$ an observation sequence for M_e, and (CC) is in PTIME for all C_i. Then, for $s \leq m$, computing $\mathbf{WFS}_e^s(M_e)$ is in PTIME.*

This, together with the observation that $\mathbf{WFS}_e(M_e)$ coincides with the unique grounded equilibrium, allows us to verify that computing the results in our use case scenario can be done in polynomial time.

6 Related and Future Work

In this paper we have studied how eMCSs can be revised in such a way that polynomial reasoning is possible, and we have discussed an example use case to which this result applies. We have also investigated the adaptation of notions concerning minimality of (evolving) equilibria, and we observe that the notion of reducible eMCSs is considerably restricted, but not to the same extent as the efficient computation of the well-founded semantics requires. An open question is whether a more refined computation eventually tailored to less restrictive operations than considered here can be used to achieve similar results.

As mentioned in the Introduction, eMCSs share the main ideas of reactive Multi-Context Systems sketched in [6, 12, 9] inasmuch as both aim at extending mMCSs to cope with dynamic observations. Three main differences distinguish them. First, whereas eMCSs rely

on a sequence of observations, each independent from the previous ones, rMCSs encode such sequences within the same observation contexts, with its elements being explicitly timestamped. This means that with rMCSs it is perhaps easier to write bridge rules that refer, e.g., to specific sequences of observations, which in eMCSs would require explicit timestamps and storing the observations in some context, although at the cost that rMCSs need to deal with explicit time which adds an additional overhead. Second, since in rMCSs the contexts resulting from the application of the management operations are the ones that are used in the subsequent state, difficulties may arise in separating non-persistent and persistent effects, for example, allowing an observation to override some fact in some context while the observation holds, but without changing the context itself – such separation is easily encodable in eMCSs given the two kinds of bridge rules, i.e., with or without operator $next$. Finally, bridge rules with $next$ allow for the specification of transitions based on the current state, such as the one encoded by the rule $next(add(p)) \leftarrow not\, p$, which do not seem possible in rMCSs. Overall, these differences indicate that an interesting future direction would be to merge both approaches, exploring a combination of explicitly timestamped observations with the expressiveness provided by operator $next$.

Another framework that aims at modeling the dynamics of knowledge is that of evolving logic programs EVOLP [2] focusing on updates of generalized logic programs. It is possible to show that EVOLP can be seen as a particular case of eMCSs, using the operator $next$ to capture the operator $assert$ of EVOLP. We leave the details for an extended version. Closely related to EVOLP, hence to eMCS, are the two frameworks of reactive ASP, one implemented as a solver *clingo* [14] and one described in [6]. The system *oclingo* extends an ASP solver for handling external modules provided at runtime by a controller. The output of these external modules can be seen as the observations of EVOLP. Unlike the observations in EVOLP, which can be rules, external modules in *oclingo* are restricted to produce atoms so the evolving capabilities are very restricted. On the other hand, *clingo* permits committing to a specific answer-set at each state, a feature that is not part of EVOLP, nor of eMCS. Reactive ASP as described in [6] can be seen as a more straightforward generalization of EVOLP where operations other than $assert$ for self-updating a program are permitted. Given the above mentioned embedding of EVOLP in eMCS, and the fact that eMCSs permit several (evolution) operations in the head of bridge rules, it is also not difficult to show that Reactive ASP as described in [6] can be captured by eMCSs.

Also, as already outlined in [20], an important non-trivial topic is the study of the notion of minimal change within an evolving equilibrium. Whereas minimal change may be desirable to obtain more coherent evolving equilibria, there are also arguments against adopting a one-size-fits-all approach embedded in the semantics. Different contexts, i.e., KR formalisms, may require different notions of minimal change, or even require to avoid it – e.g., suppose we want to represent some variable that can non-deterministically takes one of two values at each time instant: minimal change could force a constant value.

Another important issue open for future work is a more fine-grained characterization of updating bridge rules (and knowledge bases) as studied in [18] in light of the encountered difficulties when updating rules [28, 29, 31] and the combination of updates over various formalisms [29, 30].

Also interesting is to study how to perform AGM style belief revision at the (semantic) level of the equilibria, as in Wang et al [33], though different since knowledge is not incorporated in the contexts.

ACKNOWLEDGEMENTS

REFERENCES

[1] M. Alberti, A. S. Gomes, R. Gonçalves, J. Leite, and M. Slota, 'Normative systems represented as hybrid knowledge bases', in *CLIMA*, eds., J. Leite, P. Torroni, T. Ågotnes, G. Boella, and L. van der Torre, volume 6814 of *LNCS*, pp. 330–346. Springer, (2011).

[2] J. Alferes, A. Brogi, J. Leite, and L. Pereira, 'Evolving logic programs', in *JELIA*, eds., S. Flesca, S. Greco, N. Leone, and G. Ianni, volume 2424 of *LNCS*, pp. 50–61. Springer, (2002).

[3] D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic, 'Stream reasoning and complex event processing in ETALIS', *Semantic Web*, **3**(4), 397–407, (2012).

[4] Franz Baader, Sebastian Brandt, and Carsten Lutz, 'Pushing the el envelope', in *IJCAI*, eds., Leslie Pack Kaelbling and Alessandro Saffiotti, pp. 364–369. Professional Book Center, (2005).

[5] D. Barbieri, D. Braga, S. Ceri, E. Valle, and M. Grossniklaus, 'C-SPARQL: a continuous query language for RDF data streams', *Int. J. Semantic Computing*, **4**(1), 3–25, (2010).

[6] G. Brewka, 'Towards reactive multi-context systems', in *LPNMR*, eds., P. Cabalar and T. C. Son, volume 8148 of *LNCS*, pp. 1–10. Springer, (2013).

[7] G. Brewka and T. Eiter, 'Equilibria in heterogeneous nonmonotonic multi-context systems', in *AAAI*, pp. 385–390. AAAI Press, (2007).

[8] G. Brewka, T. Eiter, M. Fink, and A. Weinzierl, 'Managed multi-context systems', in *IJCAI*, ed., T. Walsh, pp. 786–791. IJCAI/AAAI, (2011).

[9] G. Brewka, S. Ellmauthaler, and J. Pührer, 'Multi-context systems for reactive reasoning in dynamic environments', in *ECAI*, eds., T. Schaub, G. Friedrich, and B. O'Sullivan. IOS Press, (2014). To appear.

[10] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov, 'Complexity and expressive power of logic programming', *ACM Comput. Surv.*, **33**(3), 374–425, (2001).

[11] T. Eiter, M. Fink, P. Schüller, and A. Weinzierl, 'Finding explanations of inconsistency in multi-context systems', in *KR*, eds., F. Lin, U. Sattler, and M. Truszczynski. AAAI Press, (2010).

[12] S. Ellmauthaler, 'Generalizing multi-context systems for reactive stream reasoning applications', in *ICCSW*, eds., A. V. Jones and N. Ng, volume 35 of *OASICS*, pp. 19–26. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, (2013).

[13] M. Gebser, T. Grote, R. Kaminski, P. Obermeier, O. Sabuncu, and T. Schaub, 'Stream reasoning with answer set programming: Preliminary report', in *KR*, eds., G Brewka, T. Eiter, and S. A. McIlraith. AAAI Press, (2012).

[14] M. Gebser, T. Grote, R. Kaminski, and T. Schaub, 'Reactive answer set programming', in *LPNMR*, eds., J. P. Delgrande and W. Faber, volume 6645 of *LNCS*, pp. 54–66. Springer, (2011).

[15] Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf, 'The well-founded semantics for general logic programs', *J. ACM*, **38**(3), 620–650, (1991).

[16] F. Giunchiglia and L. Serafini, 'Multilanguage hierarchical logics or: How we can do without modal logics', *Artif. Intell.*, **65**(1), 29–70, (1994).

[17] R. Gonçalves and J. Alferes, 'Parametrized logic programming', in *JELIA*, eds., T. Janhunen and I. Niemelä, volume 6341 of *LNCS*, pp. 182–194. Springer, (2010).

[18] R. Gonçalves, M. Knorr, and J. Leite, 'Evolving bridge rules in evolving multi-context systems', in *CLIMA XV*, eds., N. Bulling, L. van der Torre, S. Villata, W. Jamroga, and W. Vasconcelos, (2014). To appear.

[19] R. Gonçalves, M. Knorr, and J. Leite, 'Evolving multi-context systems', in *ECAI*, eds., T. Schaub, G. Friedrich, and B. O'Sullivan. IOS Press, (2014). To appear.

[20] R. Gonçalves, M. Knorr, and J. Leite, 'On minimal change in evolving multi-context systems (preliminary report)', in *ReactKnow 2014*, (2014). To appear.

[21] M. Homola, M. Knorr, J. Leite, and M. Slota, 'MKNF knowledge bases in multi-context systems', in *CLIMA*, eds., M. Fisher, L. van der Torre, M. Dastani, and G. Governatori, volume 7486 of *LNCS*, pp. 146–162. Springer, (2012).

[22] V. Ivanov, M. Knorr, and J. Leite, 'A query tool for $\mathcal{EL}$ with non-monotonic rules', in *ISWC*, eds., H. Alani, L. Kagal, A. Fokoue, P. T. Groth, C. Biemann, J. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz, volume 8218 of *LNCS*, pp. 216–231. Springer, (2013).

[23] M. Knorr, J. Alferes, and P. Hitzler, 'Local closed world reasoning with description logics under the well-founded semantics', *Artif. Intell.*, **175**(9-10), 1528–1554, (2011).

[24] M. Knorr, M. Slota, J. Leite, and M. Homola, 'What if no hybrid reasoner is available? Hybrid MKNF in multi-context systems', *J. Log. Comput.*, (2013).

[25] F. Lécué and J. Pan, 'Predicting knowledge in an ontology stream', in *IJCAI*, ed., F. Rossi. IJCAI/AAAI, (2013).

[26] B. Motik and R. Rosati, 'Reconciling description logics and rules', *J. ACM*, **57**(5), (2010).

[27] F. Roelofsen and L. Serafini, 'Minimal and absent information in contexts', in *IJCAI*, eds., L. Kaelbling and A. Saffiotti, pp. 558–563. Professional Book Center, (2005).

[28] M. Slota and J. Leite, 'On semantic update operators for answer-set programs', in *ECAI*, eds., H. Coelho, R. Studer, and M. Wooldridge, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pp. 957–962. IOS Press, (2010).

[29] M. Slota and J. Leite, 'Robust equivalence models for semantic updates of answer-set programs', in *KR*, eds., G. Brewka, T. Eiter, and S. A. McIlraith. AAAI Press, (2012).

[30] M. Slota and J. Leite, 'A unifying perspective on knowledge updates', in *JELIA*, eds., L. del Cerro, A. Herzig, and J. Mengin, volume 7519 of *LNCS*, pp. 372–384. Springer, (2012).

[31] M. Slota and J. Leite, 'The rise and fall of semantic rule updates based on SE-models', *TPLP*, (2014). To appear.

[32] Martin Slota, João Leite, and Terrance Swift, 'Splitting and updating hybrid knowledge bases', *TPLP*, **11**(4-5), 801–819, (2011).

[33] Y. Wang, Z. Zhuang, and K. Wang, 'Belief change in nonmonotonic multi-context systems', in *LPNMR*, eds., P. Cabalar and T. C. Son, volume 8148 of *LNCS*, pp. 543–555. Springer, (2013).