On Efficient Evolving Multi-Context Systems

Matthias Knorr, Ricardo Gonçalves, and João Leite

CENTRIA & Departamento de Informática, Faculdade Ciências e Tecnologia,
Universidade Nova de Lisboa

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

In this paper, we investigate conditions under which reasoning within evolving Multi-Context Systems [1] can be performed in polynomial time.

Evolving Multi-Context Systems [1] are a recent extension of Multi-Context Systems (MCSs) [2] and Managed Multi-Context Systems (mMCSs) [3] to combine the ability to integrate and manage knowledge represented in heterogeneous KR formalisms, inherited from MCSs and mMCSs, with the ability to incorporate knowledge obtained from dynamic observations.

Building on the work in [4,5], Multi-Context Systems [2] were introduced 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., [6,7], and [8] with its reasoner NoHR [9]) 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. [10,11,12] 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 their heads to the context's knowledge base provided the queries (to other contexts) in their bodies are successful. Managed Multi-Context Systems were introduced in [3] to provide an extension of MCSs by allowing operations, other than simple addition, to be expressed in the heads of bridge rules, thus allowing to properly deal with the problem of consistency management within contexts.

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 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. The dynamic requirements imposed by these scenarios – where there is a need to react and evolve in the presence of incoming information, as opposed to the classical static scenarios which assume a one-shot computation usually triggered by a user query – have been recently guiding research in KR languages, resulting in systems such as EVOLP [13], Reactive ASP [14,15], C-SPARQL [16], Ontology Streams [17] and ETALIS [18].

To address the requirements posed by such dynamic scenarios, evolving Multi-Context Systems (eMCSs) [1] were recently introduced with 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.

Just like an mMCS, an eMCS is composed of a collection of components, each of which contains knowledge represented in some logic, interconnected by bridge rules which can specify different ways to share knowledge. Some contexts of an eMCS called observation contexts, are reserved for dynamic incoming observations, changing at each state according to what is observed. Additionally, eMCSs' bridge rules have additional expressiveness to allow the specification of how contexts should react and evolve. The semantics of the resulting system is based on the so-called *evolving equilibrium* which extends the notion of *equilibrium* from mMCSs to the dynamic setting.

Whereas the semantics based on the *evolving equilibrium* adequately addresses the semantic issues of incorporating dynamic incoming observations in the contexts, other results in [1] show that worst-case complexity is in general high. The source for the high complexity can be traced to the high complexity of the KR formalism used by each of the contexts, although the bridge rules also contribute, up to a certain level, to such overall complexity. This can be a major problem in 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.

In this paper, illustrated by an example inspired by a real-world use-case, we investigate conditions under which reasoning with eMCSs can be done in polynomial time, along two distinct ways: by characterizing a subclass of eMCSs whose semantics can be computed efficiently; and by defining a well-founded semantics which can be computed efficiently for the general class of reducible eMCSs (i.e., eMCSs where the contexts are not the source of high complexity). While addressing these complexity problems, we also tackle an additional issue with eMCSs, inherited from MCSs and discussed in [2], namely that equilibria may be non-minimal, which potentially admits that certain pieces of information are considered true based solely on self-justification. Whereas such issue can in principle be solved by indicating for each context whether it requires minimality or not [2], avoiding self-justifications for those contexts where minimality is desired has not been considered in eMCSs. Thus, the main contributions of the paper are:

– we study the notion of minimality under which self-justifications in eMCSs can be avoided and introduce reducible eMCSs for which so-called evolving grounded equilibria can be computed without additional minimality checks;
– we introduce definite eMCSs, a subclass of eMCSs for which evolving grounded equilibria can be computed in polynomial time;
– we introduce a well-founded semantics for the more general class of reducible eMCSs, which can be computed in polynomial time;
– we illustrate eMCSs with a use-case inspired by a real-world scenario on cargo shipment assessment taken from [19].

2 Evolving Multi-Context Systems

In this section, we revisit evolving Multi-Context Systems as introduced in [1], already including one simplification, which will be pointed out appropriately.

An evolving multi-context system (eMCS) 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, $\mathbf{BS}$ the set of possible belief sets, and $\mathbf{ACC} : \mathbf{KB} \rightarrow 2^{\mathbf{BS}}$ 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 operations to perform on its knowledge base given certain beliefs held in the components of the eMCS. For that purpose, each component of an eMCS is associated with a *management base*, which is a set of operations that can be applied to the possible knowledge bases of that component. 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* over OP and L. Each component of an eMCS gives semantics to operations in its management base using a *management function* over a logic L and a management base OP, $mng : 2^{OF} \times \mathbf{KB} \rightarrow \mathbf{KB}$, i.e., $mng(op, kb)$ is the knowledge base that results from applying the operations in *op* to the knowledge base kb. We also assume that $mng(\emptyset, kb) = kb$. Note that ensuring that mng is deterministic is the restriction on eMCSs in [1] we introduce here for the sake of efficiency.

As indicated in [1], some of the contexts in an eMCS are 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.[1] 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 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, the set of operational formulas is extended with a unary meta-operation $next$ that can only be applied on top of operations. Given a management base OP and a logic L, eOF, the evolving operational language, is defined as $eOF = OF \cup \{next(op(s)) : op(s) \in OF\}$.

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

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 $H(\sigma) \in eOF_i$ 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. Then evolving Multi-Context Systems are defined as follows.

Definition 1. *An* 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,* OP_i *is a management base, and* 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 M_e can be explicitly represented by $\langle C_1^o, \dots, C_\ell^o, C_{\ell+1}, \dots, C_n \rangle$. Then, 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 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 2. *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:*

- $app_i^{next}(S) = \{op(s) : next(op(s)) \in app_i(S)\}$
- $app_i^{now}(S) = \{op(s) : op(s) \in app_i(S)\}$

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.

The (static) equilibrium is defined to incorporate instantaneous effects based on $app_i^{now}(S)$ alone.

Definition 3. *Let* $M_e = \langle C_1, \dots, C_n \rangle$ *be an eMCS. A belief state* $S = \langle S_1, \dots, S_n \rangle$ *for* M_e *is a static* equilibrium of M_e *iff we have* $S_i \in \mathbf{ACC}_i(mng_i(app_i^{now}(S), kb_i))$ *for each* $1 \leq i \leq n$*.*

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

Definition 4. *Let* $M_e = \langle C_1, \dots, C_n \rangle$ *be an eMCS. An* evolving belief state *of size* s *for* M_e *is a sequence* $S_e = \langle S^1, \dots, S^s \rangle$ *s.t. all* S^j*,* $1 \leq j \leq s$*, are belief states 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 5. *Let* $M_e = \langle C_1^o, \dots, C_\ell^o, C_{\ell+1}, \dots, C_n \rangle$ *be an eMCS. An* observation sequence *for* M_e *is a sequence* $Obs = \langle \mathcal{O}^1, \dots, \mathcal{O}^m \rangle$*, such that, for each* $1 \leq j \leq m$*,* $\mathcal{O}^j = \langle o_1^j, \dots, o_\ell^j \rangle$ *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 6. *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 the eMCS M^j obtained as $M^j = \langle C_1^o[o_1^j], \ldots, C_\ell^o[o_\ell^j], C_{\ell+1}[k_{\ell+1}^j], \ldots, C_n[k_n^j] \rangle$ where, for each $\ell + 1 \leq i \leq n$, k_i^j is defined inductively as follows:*

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

Note that $next$ in bridge rule heads of observation contexts are thus without any effect.

3 Use Case Scenario

We illustrate eMCSs adapting a scenario on cargo shipment assessment taken from [19].

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.[2] 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 change 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,

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

the commodity they contain, their importers and producers. The knowledge base and belief set language of C_1^o is composed of all the ground atoms over $\mathsf{ShpmtCommod}/2$, $\mathsf{ShpmtDeclHTSCode}/2$, $\mathsf{ShpmtImporter}/2$, $\mathsf{ShpmtProducer}/2$, $\mathsf{ShpmtCountry}/2$, and also $\mathsf{GrapeTomato}/1$ and $\mathsf{CherryTomato}/1$. 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 ground atoms over $\mathsf{NewEUMember}/1$, $\mathsf{Misfiling}/1$, and $\mathsf{RandomInspection}/1$. Both observation contexts have no bridge rules.

The reasoning context C_3 is an ontological Description Logic (DL) [20] context for which the standard definition of L_3 can be found, e.g., in [21] and [3]. This context contains a geographic classification, along with information about producers who are located in various countries, and a classification of commodities based on their harmonized tariff information (HTS chapters, headings and codes, cf. `http://www.usitc.gov/tata/hts`). Then kb_3 is given as follows:

$$\begin{array}{ll}
\mathsf{Commodity} \equiv (\exists \mathsf{HTSCode}.\top) & \mathsf{EdibleVegetable} \equiv (\exists \mathsf{HTSChapter}.\{\text{`07'}\}) \\
\mathsf{Tomato} \sqsubseteq \mathsf{EdibleVegetable} & \mathsf{Tomato} \equiv (\exists \mathsf{HTSHeading}.\{\text{`0702'}\}) \\
\mathsf{CherryTomato} \sqsubseteq \mathsf{Tomato} & \mathsf{CherryTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{`07020020'}\}) \\
\mathsf{GrapeTomato} \sqsubseteq \mathsf{Tomato} & \mathsf{GrapeTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{`07020010'}\}) \\
\mathsf{CherryTomato} \sqcap \mathsf{GrapeTomato} \sqsubseteq \bot & \\
\multicolumn{2}{l}{\mathsf{EURegisteredProducer} \equiv (\exists \mathsf{RegisteredProducer}.\mathsf{EUCountry})} \\
\multicolumn{2}{l}{\mathsf{LowRiskEUCommodity} \equiv (\exists \mathsf{ExpeditImporter}.\top) \sqcap (\exists \mathsf{CommodCountry}.\mathsf{EUCountry})} \\
\mathsf{EUCountry}(portugal) & \mathsf{RegisteredProducer}(p_1, portugal) \\
\mathsf{EUCountry}(slovakia) & \mathsf{RegisteredProducer}(p_2, slovakia)
\end{array}$$

OP_3 contains operations to add (add) and remove (rm) factual knowledge. The bridge rules br_3 are given as follows:

$$\begin{array}{l}
add(\mathsf{CherryTomato}(\mathbf{x})) \leftarrow (1{:}\mathsf{CherryTomato}(\mathbf{x})) \\
add(\mathsf{GrapeTomato}(\mathbf{x})) \leftarrow (1{:}\mathsf{GrapeTomato}(\mathbf{x})) \\
next(add(\mathsf{EUCountry}(\mathbf{x}))) \leftarrow (2{:}\mathsf{NewEUMember}(\mathbf{x})) \\
next(rm(\mathsf{EUCountry}(\mathbf{x}))) \leftarrow (2{:}\mathsf{LeftEU}(\mathbf{x})) \\
add(\mathsf{CommodCountry}(\mathbf{x},\mathbf{y})) \leftarrow (1{:}\mathsf{ShpmtCommod}(\mathbf{z},\mathbf{x})), (1{:}\mathsf{ShpmtCountry}(\mathbf{z},\mathbf{y})) \\
add(\mathsf{ExpeditImporter}(\mathbf{x},\mathbf{y})) \leftarrow (1{:}\mathsf{ShpmtCommod}(\mathbf{z},\mathbf{x})), (1{:}\mathsf{ShpmtImporter}(\mathbf{z},\mathbf{y})), \\
\qquad\qquad (4{:}\mathsf{AdmissibleImporter}(\mathbf{y})), (4{:}\mathsf{ApprovedImporterOf}(\mathbf{y},\mathbf{x}))
\end{array}$$

Note that kb_3 can indeed be expressed in the DL $\mathcal{EL}^{++}$ [22] 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 $\mathbf{KB}_i$ is 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 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 [23] 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 [24]. We consider $OP_4 = OP_3$, and kb_4 and br_4:

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

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

To illustrate how this eMCS can evolve, 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 stands for shipment, c for commodity, and i for importer):

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

o_1^2 of the following atoms on s_2:

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

and o_1^3 of the following atoms on s_3:

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

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.

Besides illustrating eMCSs, what this use case scenario also shows is that efficient reasoning is a very important matter in the sense that the decision whether to (partially) inspect a shipment or not should be essentially instantaneous once the sensor data is received. To this end, in the next section, we focus on conditions under which reasoning with eMCSs becomes efficient w.r.t. the worst-case complexity. At the same time, basically orthogonal to that, we can also speed-up computation utilizing only a restricted part of the sequence of evolving MCSs for computing evolving equilibria (or even only one for which the static equilibrium suffices). This is matched accordingly in the next section, as notions are introduced that also cover the static case.

4 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 still in NP [2,3]. The same result necessarily also holds for eMCSs, which can also be obtained from the considerations on eMCSs [1].

A number of notions were studied in the context of MCSs that tackle this problem [2]. In fact, minimal equilibria were 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 [2] to dynamic eMCSs and discuss under which (non-trivial) conditions they can actually be applied.

First, given an eMCS $M_e = \langle C_1, \ldots, C_n \rangle$, 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 avoiding self-justifications in evolving equilibria. The problem with it 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 $S' \subseteq S$, and $\mathbf{ACC}(red(kb, S)) = \{ S \}$ iff $S \in \mathbf{ACC}(kb)$.

An evolving context $C = (L, kb, br, OP, mng)$ is *reducible* if its logic L is reducible and, for all $op \in OP$ 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 [2], reducibility is inspired by the reduct in (non-monotonic) answer set programming. The novel condition in our case is the one saying that the reduction function red and the management function mng have to be applicable 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. 3, 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 $\mathbf{not}$, and
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, which we call 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 7. *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 s.t. $m \geq s$. Then, S_e is the* evolving grounded equilibrium *of size s of M_e given Obs iff, for each $1 \leq j \leq s$, $S^j = \mathbf{GE}(M^j)$ with M^j defined as in Def. 6.*

Grounded equilibria for definite eMCSs can indeed be efficiently computed. The only additional requirement is that OF is *monotonic* for each context, i.e., for any kb and each $op(s) \in OF$ with $H(\sigma) = op(s)$ for some $\sigma \in br$, we have that $kb \subseteq mng(op(s), kb)$. Note that this is in fact a restriction not covered by reducible eMCSs, yet it does not affect operations occurring under $next$. 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 we have:

Proposition 1. *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a definite eMCS s.t. all OF_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 [2], 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. 3.

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.

This same iteration cannot be applied to arbitrary reducible eMCSs right away. Instead, grounded equilibria for general reducible eMCSs are defined based on a reduct which generalizes the Gelfond-Lifschitz reduct to the multi-context case:

Definition 8. *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 $M_e^S = \langle C_1^S, \ldots, C_n^S \rangle$ s.t., 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 rules with $\mathbf{not}\,(r:p)$ in the body such that $S \models (r:p)$, and all $\mathbf{not}$ 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, and it can be shown that grounded equilibria of reducible eMCSs are minimal.

Definition 9. *Let M_e be a reducible eMCS such that all OF_i are monotonic. A belief state S of M_e is a* grounded equilibrium *of M_e if $S = \mathbf{GE}(M_e^S)$.*

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

This can again be generalized to evolving grounded equilibria.

Definition 10. *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a reducible eMCS such that all OF_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 Def. 6.*

For reducible eMCSs in general, this computation is not polynomial, since, intuitively, we have to guess and check the (evolving) equilibrium, which is why we introduce the well-founded semantics for reducible eMCSs M_e following the ideas in [2]. 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*.

It can be shown that γ_{M_e} is antitonic which means that 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 11. *Let M_e be a normal, reducible eMCS s.t. all OF_i are monotonic. The well-founded semantics of M_e, denoted $\mathbf{WFS}(M_e)$, 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, establishing the relation between $\mathbf{WFS}(M_e)$ and the grounded equilibria of M_e.

Proposition 3. *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OF_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 [1]), then the grounded equilibrium is unique and identical to the well-founded semantics. This is indeed the case for the use case in Sect. 3.

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

Definition 12. *Let $M_e = \langle C_1, \ldots, C_n \rangle$ be a normal, reducible eMCS such that all OF_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 of size s*, denoted $\mathbf{WFS}_e^s(M_e)$, 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 6.*

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 [21]. 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 [21,3], we can adapt the *context complexity* of C_i from [1] as follows:

(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 OF_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 $\mathbf{WFS}_e^s(M_e)$ coincides with the unique evolving grounded equilibrium of size s, allows us to verify that computing the results in our use case scenario can be done in polynomial time.

5 Related and Future Work

In this paper, we have investigated how to obtain efficient eMCSs. On the one hand, we have considered notions of minimality that avoid self-support of beliefs in eMCSs, on the other hand, we have studied how to revise eMCSs such that polynomial reasoning becomes possible. We have also discussed an example use case to which these results successfully apply.

Closely related to eMCSs is the framework of reactive Multi-Context Systems (rMCSs) [25,26,27] since both aim at extending mMCSs to cope with dynamic observations. The main difference between and eMCSs and rMCSs is that eMCSs have the meta operator $next$ that allows for a clear separation between persistent and non-persistent effects, and also the specification of transitions based on the current state, while rMCSs utilize explicit time stamps thus making it easier to write bridge rules that refer to specific sequences of observations. In general, rMCSs have also an associated high complexity, which is again problematic in dynamic scenarios where the overall system needs to evolve and react interactively. An interesting question is whether the techniques presented here for eMCSs could be adapted and applied to rMCSs to make polynomial reasoning possible.

Another framework closely related to eMCSs is that of evolving logic programs EVOLP [13] which deals with updates of generalized logic programs, and the two frameworks of reactive ASP, one implemented as a solver *clingo* [14] and one described in [25]. Whereas EVOLP employs an update predicate that is similar in spirit to the $next$ predicate of eMCSs, it does not deal with distributed heterogeneous knowledge, neither do both versions of Reactive ASP.

An important issue open for future work is a more fine-grained characterization of updating bridge rules (and knowledge bases) as studied in [28] in light of the encountered difficulties when updating rules [29,30,31] and the combination of updates over various formalisms [30,32].

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.

Acknowledgments We would like to thank the referees for their comments. Matthias Knorr and João Leite were partially supported by FCT under project "ERRO – Efficient Reasoning with Rules and Ontologies" (PTDC/EIA-CCO/121823/2010). Ricardo Gonçalves was supported by FCT grant SFRH/BPD/47245/2008 and Matthias Knorr was also partially supported by FCT grant SFRH/BPD/86970/2012.

References

1. Gonçalves, R., Knorr, M., Leite, J.: Evolving multi-context systems. In Schaub, T., Friedrich, G., O'Sullivan, B., eds.: Procs. ECAI, IOS Press (2014) 375–380
2. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: Procs. AAAI, AAAI Press (2007) 385–390
3. Brewka, G., Eiter, T., Fink, M., Weinzierl, A.: Managed multi-context systems. In Walsh, T., ed.: Procs. IJCAI, IJCAI/AAAI (2011) 786–791
4. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics or: How we can do without modal logics. Artif. Intell. **65**(1) (1994) 29–70
5. Roelofsen, F., Serafini, L.: Minimal and absent information in contexts. In Kaelbling, L.P., Saffiotti, A., eds.: Procs. IJCAI, Professional Book Center (2005) 558–563
6. Gonçalves, R., Alferes, J.: Parametrized logic programming. In Janhunen, T., Niemelä, I., eds.: Procs. JELIA. Volume 6341 of LNCS., Springer (2010) 182–194
7. Motik, B., Rosati, R.: Reconciling description logics and rules. J. ACM **57**(5) (2010)
8. Knorr, M., Alferes, J., Hitzler, P.: Local closed world reasoning with description logics under the well-founded semantics. Artif. Intell. **175**(9-10) (2011) 1528–1554
9. Ivanov, V., Knorr, M., Leite, J.: A query tool for $\mathcal{EL}$ with non-monotonic rules. In Alani, H., Kagal, L., Fokoue, A., Groth, P.T., Biemann, C., Parreira, J., Aroyo, L., Noy, N.F., Welty, C., Janowicz, K., eds.: Procs. ISWC. Volume 8218 of LNCS., Springer (2013) 216–231
10. Alberti, M., Gomes, A.S., Gonçalves, R., Leite, J., Slota, M.: Normative systems represented as hybrid knowledge bases. In Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L., eds.: Procs. CLIMA XII. Volume 6814 of LNCS., Springer (2011) 330–346

11. Homola, M., Knorr, M., Leite, J., Slota, M.: MKNF knowledge bases in multi-context systems. In Fisher, M., van der Torre, L., Dastani, M., Governatori, G., eds.: Procs. CLIMA XIII. Volume 7486 of LNCS., Springer (2012) 146–162
12. Knorr, M., Slota, M., Leite, J., Homola, M.: What if no hybrid reasoner is available? Hybrid MKNF in multi-context systems. J. Log. Comput. (2013)
13. Alferes, J., Brogi, A., Leite, J., Pereira, L.: Evolving logic programs. In Flesca, S., Greco, S., Leone, N., Ianni, G., eds.: Procs. JELIA. Volume 2424 of LNCS., Springer (2002) 50–61
14. Gebser, M., Grote, T., Kaminski, R., Schaub, T.: Reactive answer set programming. In Delgrande, J.P., Faber, W., eds.: LPNMR. Volume 6645 of LNCS., Springer (2011) 54–66
15. Gebser, M., Grote, T., Kaminski, R., Obermeier, P., Sabuncu, O., Schaub, T.: Stream reasoning with answer set programming: Preliminary report. In Brewka, G., Eiter, T., McIlraith, S.A., eds.: Procs. KR, AAAI Press (2012)
16. Barbieri, D., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: C-SPARQL: a continuous query language for RDF data streams. Int. J. Semantic Computing **4**(1) (2010) 3–25
17. Lécué, F., Pan, J.: Predicting knowledge in an ontology stream. In Rossi, F., ed.: Procs. IJCAI, IJCAI/AAAI (2013)
18. Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Stream reasoning and complex event processing in ETALIS. Semantic Web **3**(4) (2012) 397–407
19. Slota, M., Leite, J., Swift, T.: Splitting and updating hybrid knowledge bases. TPLP **11**(4-5) (2011) 801–819
20. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. 2nd edn. Cambridge University Press (2007)
21. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding explanations of inconsistency in multi-context systems. In Lin, F., Sattler, U., Truszczynski, M., eds.: KR, AAAI Press (2010)
22. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In Kaelbling, L.P., Saffiotti, A., eds.: IJCAI, Professional Book Center (2005) 364–369
23. Gelder, A.V., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. J. ACM **38**(3) (1991) 620–650
24. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Comput. Surv. **33**(3) (2001) 374–425
25. Brewka, G.: Towards reactive multi-context systems. In Cabalar, P., Son, T.C., eds.: Procs. LPNMR. Volume 8148 of LNCS., Springer (2013) 1–10
26. Ellmauthaler, S.: Generalizing multi-context systems for reactive stream reasoning applications. In: ICCSW. Volume 35 of OASICS., Schloss Dagstuhl, Germany (2013) 19–26
27. Brewka, G., Ellmauthaler, S., Pührer, J.: Multi-context systems for reactive reasoning in dynamic environments. In Schaub, T., Friedrich, G., O'Sullivan, B., eds.: Procs. ECAI, IOS Press (2014) 159–164
28. Gonçalves, R., Knorr, M., Leite, J.: Evolving bridge rules in evolving multi-context systems. In Bulling, N., van der Torre, L.W.N., Villata, S., Jamroga, W., Vasconcelos, W., eds.: Procs. CLIMA XV. Volume 8624 of LNCS., Springer (2014) 52–69
29. Slota, M., Leite, J.: On semantic update operators for answer-set programs. In Coelho, H., Studer, R., Wooldridge, M., eds.: Procs. ECAI, IOS Press (2010) 957–962
30. Slota, M., Leite, J.: Robust equivalence models for semantic updates of answer-set programs. In Brewka, G., Eiter, T., McIlraith, S.A., eds.: Procs. KR, AAAI Press (2012)
31. Slota, M., Leite, J.: The rise and fall of semantic rule updates based on SE-models. TPLP (2014) To appear.
32. Slota, M., Leite, J.: A unifying perspective on knowledge updates. In del Cerro, L., Herzig, A., Mengin, J., eds.: Procs. JELIA. Volume 7519 of LNCS., Springer (2012) 372–384
33. Wang, Y., Zhuang, Z., Wang, K.: Belief change in nonmonotonic multi-context systems. In Cabalar, P., Son, T.C., eds.: LPNMR. Volume 8148 of LNCS., Springer (2013) 543–555