ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1680

Multi-Context Systems in Time

Stefania Costantini and **Andrea Formisano** [1]

Abstract. In this paper we consider how to enhance flexibility and generality in Multi-Context Systems (MCS) by considering that contexts can evolve over time, that bridge-rule application can be proactive (according to a context's specific choice), and not instantaneous but requiring an execution mechanism. We introduce bridge-rule patterns to make bridge-rules parametric w.r.t. the involved contexts.

1 BRIDGE RULES AND MCS

Multi-Context Systems (MCSs) have been proposed in Artificial Intelligence and Knowledge Representation to model information exchange among heterogeneous sources. MCSs are defined so as to drop the assumption of making such sources in some sense homogeneous: rather, the approach deals explicitly with their different representation languages and semantics. Heterogeneous "contexts" (also called "sources", or "modules") interact through special inter-context *bridge rules*. The reason why MCSs are particularly interesting is that they aim at modeling in a formal way real applications requiring access to sources distributed, for instance, on the web. In view of such practical applications it is important to notice that, being logic-based, contexts may encompass logical agents, to which MCSs have in fact already been extended (cf. [2, 3]).

We refer the reader to [1, 5], and the references therein, for the formal definition of basic notions and properties concerning MCSs and *managed MCSs* (for short, mMCSs), such as context, bridge rule, belief state, management function, equilibrium, etc.

2 PROPOSED EXTENSIONS

2.1 Update Operators and Timed Equilibria

Let us assume a discrete, linear model of time. States $t_1, t_2, \ldots$ can be seen as time instants (or 'time points') in abstract terms. Moreover, we assume that each context is subject, at each time point, to a (possibly empty) finite update. Thus, for mMCS $M = (C_1, \ldots, C_n)$ let $\Pi_T = \langle \Pi_T^1, \ldots \Pi_T^n \rangle$ be a tuple composed of the finite updates performed to each context at time T, where, for $1 \leq i \leq n$, Π_T^i is the update to context C_i (possibly including the set Ops of sensor inputs of [1]). Let $\Pi = \Pi_1, \Pi_2, \ldots$ be a sequence of such updates performed at time instants $t_1, t_2, \ldots$ Let us assume that each context copes with updates in its own particular way, so let $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_n\}$ be the tuple composed of the *update operators* $\mathcal{U}_i$s that modules C_is employ for incorporating the new information. Let, morever, the *update base* $uops_i$ be a set of update operations which are admitted on context C_i. Then we have: $\mathcal{U}_i : 2^{uops_i} \times KB_i \rightarrow 2^{KB_i} \setminus \{\emptyset\}$, where KB_i is the set of knowledge bases pertaining to C_i. Consequently, we allow contexts' knowledge bases and belief states to evolve in time.

Definition 1 *Given context* $C_i = (c_i; L_i; kb_i; br_i; OP_i; mng_i)$ *as originally defined in mMCS, we define the corresponding* timed *context w.r.t. belief state* S *as follows:*
$C_i^0 = (c_i; L_i; kb_i^0; br_i; OP_i; mng_i; uops_i; \mathcal{U}_i)$
$C_i^{T+1} = (c_i; L_i; kb_i^{T+1}; br_i; OP_i; mng_i; uops_i; \mathcal{U}_i)$,
where $kb_i^0 = kb_i$ *and* $kb_i^{T+1} = mng_i(app(S^T), \mathcal{U}_i(\Pi_T^i, kb_i^T))$, *with* Π_T^i *being the update performed on* C_i^T, *and* $app(\cdot)$ *determines the set of bridge rules that are applicable in a belief state.*
Timed context C_i^T *will also be referred to as "context* C_i *at time* T*".*

Definition 2 *Let* $M = (C_1^T, \ldots, C_n^T)$ *be an mMCS at time* T*. A* timed *belief state at time* T *is a tuple* $S^T = (S_1^T, \ldots, S_n^T)$ *where each* S_i^T *is a possible set of consequences of* C_i^T*'s knowledge base.*

The timed belief state S^0 will possibly be an equilibrium, according to original mMCS definition. Later on, however, transition from a timed belief state to the next one, and consequently the definition of an equilibrium, is determined both by the update operators and by the application of bridge rules. Therefore:

Definition 3 *A timed belief state of mMCS* M *at time* $T+1$ *is a timed equilibrium iff, for* $1 \leq i \leq n$ *it holds that* $S_i^{T+1} \in ACC_i(kb_i^{T+1})$*, where* $ACC_i(\cdot)$ *is a function which computes the semantics of* C_i*.*

The meaning is that a timed equilibrium is now a data state which encompasses bridge rules applicability on the updated contexts' knowledge bases. As seen in Def. 1, bridge-rule applicability is checked on belief state S^T, but bridge rules are applied (and their results incorporated by the management function) on the knowledge base resulting from the update. The enhancement w.r.t. [1] is that the management function now resumes its original role concerning bridge rules, while the update operator copes with updates. So, we relax the limitation that each rule involving an update should be considered to be a bridge rule, and that updates should consist of (the combination and elaboration of) simple atoms occurring in bridge bodies. Our approach in fact allows update operators to consider and incorporate any piece of knowledge. Moreover, we make time explicit thus showing the timed evolution of contexts and equilibria.

2.2 Bridge Rule Grounding and Activation

In the original definition of mMCSs, bridge rules are by definition ground and their application is reactive. However, according to a context's own logic, other patterns of application might in principle be defined. In particular, we admit non-ground bridge rules that might be grounded over all terms that can be built over the signature of every context's underlying logic. This because mMCSs admit

[1] DISIM – Università di L'Aquila, Italy, email: stefania.costantini@univaq.it and DMI–Università di Perugia, Italy, email: formis@dmi.unipg.it. Research partially supporded by GNCS and FCRPG.2016.0105.021 project.

"value invention", i.e., via the results of bridge-rules application, beliefs (and their arguments) are propagated among contexts; so, via a bridge rule a context may obtain a result involving constants and terms previously not occurring therein.

Definition 4 *Let r be a non-ground bridge rule occurring in context C_i of a given mMCS M with (timed) belief state S^T. A ground instance ρ of r w.r.t. S^T is obtained by substituting every variable occurring in r via ground terms occurring in S^T.*

As concerns proactive activation, let, for each context C_i, $H(T,i) = \{h | \text{there exists rule } r \in br_i \text{ with head } h \text{ at time } T\}$, and let $tr_i^T : KB_i \rightarrow 2^{H(T,i)}$ be a *timed triggering function*, specifying which rules are triggered at each time T, by performing some reasoning over the present knowledge base in KB_i of C_i. Then, a bridge rule r of context C_i is *triggered at time T* iff $r \in tr_i^T(kb_i^T)$. Note that, a (ground instance of) a bridge rule can be triggered at a time T', but can become applicable at some later time T. Thus, any bridge rule which has been triggered in actual terms remains in predicate for applicability, which occurs whenever its body should be entailed by some future data state. The definition of timed equilibrium remains unchanged, save for modified bridge-rule applicability. However, the added expressivity is remarkable, as with our solution a context is not just the passive recipient of new information, but rather can reason about which bridge rules to potentially apply at each stage. Moreover, in practical applications, the grounding of literals in bridge rules can be computed at run-time, when the rule is actually applied.

2.3 Bridge-Rules Patterns

In [4] we have proposed, for logical agents, *bridge-rule patterns* which are enhanced bridge rules of the form

$$s \leftarrow (\mathcal{C}_1{:}p_1), \ldots, (\mathcal{C}_j{:}p_j), not\,(\mathcal{C}_{j+1}{:}p_{j+1}), \ldots, not\,(\mathcal{C}_m{:}p_m).$$

where each $\mathcal{C}_d$ can be either a constant (i.e., a context name, as in usual bridge rules) or a *context designator* (which is a term built up from fresh function symbol and constants. Intuitively, a context designator indicates a specific kind of context, and can be substituted by a context name. New bridge rules can thus be obtained by replacing, in a bridge-rule pattern, context designators via actual context names. So, contexts will now evolve also in the sense that they may increase their set of bridge rules by exploiting bridge-rule patterns. Note that the context names to substitute to a context designator are established by suitable reasoning performed according to context's knowledge base. All other previously-introduced notions (equilibria, bridge-rule triggering and applicability, etc.) remain unchanged. Notice that bridge-rule patterns instantiation corresponds to specializing rules with respect to the context which is deemed more suitable for acquiring some specific information at a certain stage of a context's operation. This evaluation is performed via specific predicates (cf. [5]) so as to take several factors into account, among which, for instance, trust and preferences. Also, this enhancement goes in the direction of *dynamic* mMCS, where contexts can either join or leave the system during its operation, while rule applicability may depend upon the presence in the system of suitable contexts.

3 COMPLEXITY ISSUES

In general, the property that we may wish to check is whether a specific belief of our interest will eventually occur at some stage in one (or all) timed equilibria of a given mMCS. The formal definition is the following.

Definition 5 *The problem $Q^\exists$ (respectively $Q^\forall$), consists in deciding whether, for a given mMCS M under a sequence $\Pi = \Pi_1, \Pi_2, \ldots$ of updates performed at time instants $t_1, t_2, \ldots, t$, and for a context C_i of M and a belief b_i for C_i, it holds that $b_i \in S_i^{t'}$ for some (respectively for all) timed equilibria $S^{t'}$ at time $t' \leq t$.*

The system's *context complexity* (see [6]) depends upon the logics of the contexts composing M. Then, in general, the problems $Q^\exists$ and $Q^\forall$ are undecidable for infinite update sequences, because contexts' logics can in general simulate a Turing Machine and such problems reduce to the halting problem. Complexity results can however be obtained under some restrictions. For instance, if we assume that all contexts C_i's knowledge bases and belief states are finite at any stage, all update operators $\mathcal{U}_i$, management functions and triggering functions are computable in polynomial time, and that the set of bridge-rule patterns is empty and all bridge rules are ground, it is easy to show that for finite update sequences the context complexity determines the complexity of $Q^\exists$ and, complementarily, that of $Q^\forall$.

Reconsidering bridge rules patterns, and assuming that substitutions for each context designator can be completed in polynomial time, given a set of bridge-rule patterns of size $\hat{c}$, the size of the set of its valid instances is in general single exponential in $\hat{c}$. The same holds in principle for the grounding of (non-ground) bridge rules.

4 CONCLUDING REMARKS

In this paper (see also [5]) we have discussed and extended MCSs, which are a general and powerful framework for modeling systems composed by several heterogeneous and possibly distributed sources. We have in particular considered that such systems may evolve in time in consequence of updates of several kinds, and that bridge rules get actually instantiated at run-time, and that their actual execution in a distributed setting cannot be instantaneous, but rather may involve some delay. We introduced bridge-rule patterns to make bridge rules parametrical w.r.t. the queried contexts.

We intend to employ the proposed features in the implementation, that will start in the near future, of a smart Cyber-Physical System for e-health, which will be developed in cooperation with medical oncology doctors, and will be tested and applied in the monitoring of oncology patients with co-morbidities. Participation to this project will allow us to perform also practical experiments to assess the performance of this kind of systems, and to identify possible limitations and/or further aspects that can be subject to improvements.

REFERENCES

[1] G. Brewka, S. Ellmauthaler, and J. Pührer, 'Multi-context systems for reactive reasoning in dynamic environments', in *Proc. of ECAI 2014*, ed., Torsten Schaub. IJCAI/AAAI, (2014).

[2] S. Costantini, 'Knowledge acquisition via non-monotonic reasoning in distributed heterogeneous environments', in *Proc. of LPNMR-13*, eds., M.Truszczyński, G.Ianni, and F.Calimeri, volume 9345 of *LNCS*. Springer, (2015).

[3] S. Costantini and G. De Gasperis, 'Exchanging data and ontological definitions in multi-agent-contexts systems', in *RuleML Challenge track*, eds., A.Paschke, P.Fodor, A.Giurca, and T.Kliegr, CEUR, (2015).

[4] S. Costantini and A. Formisano, 'Augmenting agent computational environments with quantitative reasoning modules and customizable bridge rules', in *Proc. of AAMAS Workshop, EMAS 2016, Proceedings*, eds., M.Baldoni J.P.Müller, I.Nunes, and R.Zalila-Wenkstern, (2016).

[5] S. Costantini and A. Formisano. Multi-context systems in time, 2016. Full version. `www.dmi.unipg.it/formis/MCSIT.pdf`.

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