Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Decidability of Model Checking Non-Uniform Artifact-Centric Quantified Interpreted Systems

Francesco Belardinelli
Laboratoires IBISC
Université d'Evry
belardinelli@ibisc.fr

Alessio Lomuscio
Department of Computing
Imperial College London
a.lomuscio@imperial.ac.uk

Abstract

Artifact-Centric Systems are a novel paradigm in service-oriented computing. In the present contribution we show that model checking bounded, non-uniform artifact-centric systems is undecidable. We provide a partial model checking procedure for artifact-centric systems against the universal fragment of a first-order version of the logic CTL. We obtain this result by introducing a counterpart semantics and developing an abstraction methodology operating on these structures. This enables us to generate finite abstractions of infinite artifact-centric systems, hence perform verification on abstract models.

1 Introduction

Formalisms and techniques based on Multi-Agent Systems (MAS) have made a significant contribution in a number of application areas, including web-services [Singh and Huhns, 2005; Baldoni *et al.*, 2012]. This has taken the form of MAS-based architectures [Denti and Omicini, 1999; Omicini *et al.*, 2008; Winikoff, 2009], monitoring methodologies [Lomuscio *et al.*, 2011; Modgil *et al.*, 2009], formal verification [Lomuscio *et al.*, 2012], and many others.

Artifact-centric (AC) systems are a novel paradigm in web-services [Hull, 2008; Hull *et al.*, 2011]. In the artifact approach, now used prominently in business-process modelling and case-based management, structured data and processes live together with equal importance in the service model and the interface descriptions. Structured data pose a considerable challenge from a verification perspective. They entail dealing with infinite state-spaces, which in turn require specification languages supporting quantification, such as first-order temporal logics.

Existing literature on the verification of AC systems [Hariri *et al.*, 2012; Belardinelli *et al.*, 2012b; Deutsch *et al.*, 2009; Gerede and Su, 2007] makes it clear that the model checking problem is typically undecidable, irrespectively of the underlying formalisation. However, the state-of-the-art points to conditions that can be imposed on the semantics of AC systems to obtain decidability. In particular, a condition referred to as "uniformity" is shown to guarantee decidability of systems with bounded runs [Belardinelli *et al.*, 2012a; 2012b], even when considering infinite-state systems. The same condition is satisfied by definition by the systems introduced in [Hariri *et al.*, 2012], thus guaranteeing the decidability of the model checking problem for bounded runs. In both cases decidability is shown by means of finite abstractions for infinite models.

A natural open question in the area is therefore whether uniformity is a necessary condition for deciding the model checking problem for AC systems. In this paper we show that this is not the case. Specifically, we show that a sound, albeit incomplete, finite abstraction technique can be given for bounded, non-uniform systems with respect to formulas in the universal fragment of the first-order computation tree logic FO-CTL. As we show, finite abstractions of infinite AC systems can be obtained by defining a suitable counterpart semantics [Brauner and Ghilardi, 2007; Corsi, 2001]. While our results for the universal fragment are positive, we also conclusively show that conditions such as uniformity are required for deciding the model checking problem against the full logic FO-CTL. To this end, we prove that model checking bounded, but non-uniform, AC systems against full FO-CTL is in general undecidable.

The rest of the paper is organised as follows. In Section 2 we fix our notation and give the syntax of the temporal specification language used as well as a semantics for AC systems in terms of artifact-centric quantified interpreted system (AC-QIS). Section 3 explores the model checking problem for bounded, non-uniform AC-QIS, thereby showing this to be undecidable in general. A counterpart semantics for AC systems and related simulation concepts are introduced in Section 4. Section 5 contains the main result of the paper concerning the derivation of finite abstractions for non-uniform, bounded AC-QIS against the universal fragment of FO-CTL. This also contains the decidability result. We conclude in Section 6 by discussing the results and pointing to future work.

2 Preliminaries

In this section we define artifact-centric quantified interpreted systems (AC-QIS), introduce a first-order version of the computation tree logic CTL, and present the model checking problem within this framework. Our notion of AC-QIS is based on [Belardinelli *et al.*, 2011; 2012a], but here we abstract from the agents and their actions. We firstly introduce

the basic terminology on databases that will be used throughout the paper as it appears in [Belardinelli *et al.*, 2012a].

Definition 1 (Database schema and instance) *A* database schema *is a finite set* $\mathcal{D} = \{P_1/q_1, \ldots, P_n/q_n\}$ *of* predicate symbols P_i *with arity* $q_i \in \mathbb{N}$.

Given a (possibly infinite) interpretation domain U*, a* $\mathcal{D}$-instance *over* U *is a mapping* D *associating each predicate symbol* P_i *to a* finite q_i*-ary relation on* U*, i.e.,* $D(P_i) \subseteq U^{q_i}$.

The set $\mathcal{D}(U)$ contains all $\mathcal{D}$-instances on the domain U. The *active domain* $ad(D)$ of a $\mathcal{D}$-instance D is the *finite* set of all individuals occurring in some predicate interpretation $D(P_i)$. The *primed version* of a database schema $\mathcal{D}$ as above is the schema $\mathcal{D}' = \{P'_1/q_1, \ldots, P'_n/q_n\}$. Finally, the disjoint union $D \oplus D'$ of two $\mathcal{D}$-instances D and D' is the $(\mathcal{D} \cup \mathcal{D}')$-instance s.t. *(i)* $D \oplus D'(P_i) = D(P_i)$; and *(ii)* $D \oplus D'(P'_i) = D'(P_i)$. We can now introduce the notion of AC-QIS.

Definition 2 (AC-QIS) *An* artifact-centric quantified interpreted system *is a tuple* $\mathcal{P} = \langle \mathcal{D}, U, \mathcal{S}, D_0, \rightarrow \rangle$*, where*

- $\mathcal{D}$ *is the* database schema
- U *is the* interpretation domain
- $D_0 \in \mathcal{D}(U)$ *is the* initial state
- $\rightarrow \subseteq \mathcal{D}(U) \times \mathcal{D}(U)$ *is the* transition relation
- $\mathcal{S} \subseteq \mathcal{D}(U)$ *is the set of* states reachable from D_0 *by* $\rightarrow$

Intuitively, an AC-QIS represents the evolution of a database instance D_0 according to the transition relation $\rightarrow$. A *run* r from $D \in \mathcal{S}$ is an infinite sequence $D^0 \rightarrow D^1 \rightarrow \cdots$, for $D^0 = D$. For $n \in \mathbb{N}$, $r^n = D^n$. A state D' is *reachable from* D if there exists a run r from $r^0 = D$ s.t. $r^i = D'$ for some $i \geq 0$. In what follows we assume that the transition relation $\rightarrow$ is serial, while $\mathcal{S}$ is the set of states reachable from D_0. Notice that $\mathcal{S}$ can be infinite. Hence, the AC-QIS are infinite-state systems in general.

AC-QIS can be seen as a generalisation of the Artifact-Centric Multi-Agent Systems (AC-MAS) in [Belardinelli *et al.*, 2012a], as the former abstract from the agents and their actions, which determine the transition relation for AC-MAS. AC-QIS are first-order temporal structures that can be used to interpret a quantified version of CTL.

Definition 3 (FO-CTL) *Given a set Var of* individual variables *and a set Con* $\subseteq U$ *of* individual constants*, the FO-CTL formulas* φ *on the predicate symbols in a database schema* $\mathcal{D}$ *are defined as follows:*

$$\varphi ::= P_i(\vec{t}) \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \forall x\varphi \mid AX\varphi \mid A\varphi U\varphi \mid E\varphi U\varphi$$

where $P_i \in \mathcal{D}$ *and* $\vec{t}$ *is a* q_i*-tuple of* terms*, i.e., elements in Var* $\cup$ *Con.*

The language FO-CTL is a first-order extension of CTL, where the only function symbols are constants. Free and bound variables are defined as standard, as well as the formulas $EX\varphi$, $AF\varphi$, $AG\varphi$, $EF\varphi$, and $EG\varphi$. The sublanguage FO-ACTL is the restriction of FO-CTL to the universal modalities AX and AU defined as follows:

$$\varphi ::= P_i(\vec{t}) \mid \neg P_i(\vec{t}) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \forall x\varphi \mid \exists x\varphi \mid AX\varphi \mid A\varphi U\varphi$$

An *assignment* is a function $\sigma : Var \mapsto U$. We denote by σ^x_u the assignment s.t. *(i)* $\sigma^x_u(x) = u$; and *(ii)* $\sigma^x_u(x') = \sigma(x')$ for $x' \neq x$. Also, we assume a Herbrand interpretation of constants, that is, $\sigma(c) = c$ for all $c \in Con$.

Definition 4 (Semantics of FO-CTL) *We define whether an AC-QIS* $\mathcal{P}$ satisfies *a formula* φ *in a* $\mathcal{D}$*-instance* D *according to assignment* σ *as follows:*

$(\mathcal{P}^\sigma, D) \models P_i(\vec{t})$ *iff* $\langle \sigma(t_1), \ldots, \sigma(t_{q_i}) \rangle \in D(P_i)$
$(\mathcal{P}^\sigma, D) \models \neg\varphi$ *iff* $(\mathcal{P}^\sigma, D) \not\models \varphi$
$(\mathcal{P}^\sigma, D) \models \varphi \rightarrow \varphi'$ *iff* $(\mathcal{P}^\sigma, D) \not\models \varphi$ *or* $(\mathcal{P}^\sigma, D) \models \varphi'$
$(\mathcal{P}^\sigma, D) \models \forall x\varphi$ *iff for all* $u \in ad(D)$, $(\mathcal{P}^{\sigma^x_u}, D) \models \varphi$
$(\mathcal{P}^\sigma, D) \models AX\varphi$ *iff for all* r*, if* $r^0 = D$ *then* $(\mathcal{P}^\sigma, r^1) \models \varphi$
$(\mathcal{P}^\sigma, D) \models A\varphi\mathcal{U}\varphi'$ *iff for all* r*, if* $r^0 = D$ *then there is* $k \geq 0$ *s.t.* $(\mathcal{P}^\sigma, r^k) \models \varphi'$*, and for all* j*,* $0 \leq j < k$ *implies* $(\mathcal{P}^\sigma, r^j) \models \varphi$
$(\mathcal{P}^\sigma, D) \models E\varphi\mathcal{U}\varphi'$ *iff for some* r*,* $r^0 = D$ *and there is* $k \geq 0$ *s.t.* $(\mathcal{P}^\sigma, r^k) \models \varphi'$*, and for all* j*,* $0 \leq j < k$ *implies* $(\mathcal{P}^\sigma, r^k) \models \varphi$

A formula φ *is* true *at* D*, or* $(\mathcal{P}, D) \models \varphi$*, if* $(\mathcal{P}^\sigma, D) \models \varphi$ *for all* σ*;* φ *is* true *in* $\mathcal{P}$*, or* $\mathcal{P} \models \varphi$*, if* $(\mathcal{P}, D_0) \models \varphi$.

Notice that we adopt an *active domain* semantics, where quantifiers range over the active domain $ad(D)$ of D.

Finally, we present the model checking problem for AC-QIS with respect to the specification language FO-CTL.

Definition 5 (Model Checking Problem) *Given an AC-QIS* $\mathcal{P}$ *and an FO-CTL formula* φ*, determine whether there is an assignment* σ_0 *such that* $(\mathcal{P}^{\sigma_0}, D_0) \models \varphi$.

In general, the model checking problem for AC-QIS is undecidable. In [Belardinelli *et al.*, 2012a] the same problem is proved to be decidable for *bounded* and *uniform* AC-MAS, a subclass of AC-QIS. We introduce both notions in relation to AC-QIS. To do so, we define a notion of isomorphism between $\mathcal{D}$-instances: the $\mathcal{D}$-instances D and D' are *isomorphic*, or $D \simeq D'$, iff there exists a bijection $\iota : ad(D) \cup Con \mapsto ad(D') \cup Con$ s.t. *(i)* ι is the identity on *Con*; and *(ii)* for every $P_i \in \mathcal{D}$, for every $\vec{u} \in U^{q_i}$, $\vec{u} \in D(P_i)$ iff $\iota(\vec{u}) \in D'(P_i)$.

Definition 6 (Boundedness and Uniformity) *An AC-QIS* $\mathcal{P}$ *is* bounded *iff there is* $b \in \mathbb{N}$ *s.t. for all* $D \in \mathcal{S}$*,* $|ad(D)| \leq b$.

An AC-QIS $\mathcal{P}$ *is* uniform *iff for every* $D, D', D'' \in \mathcal{S}$*,* $D''' \in \mathcal{D}(U)$*, if* $D \rightarrow D'$ *and* $D \oplus D' \simeq D'' \oplus D'''$*, then* $D'' \rightarrow D'''$.

We remark that boundedness places a restriction on the number of elements appearing in the active domain of each $\mathcal{D}$-instance, not on the number of $\mathcal{D}$-instances in $\mathcal{S}$, which might as well be infinite. Uniformity can intuitively be seen as a fullness condition on AC-QIS: all pairs of $\mathcal{D}$-instances isomorphic to a pair of temporally related $\mathcal{D}$-instances are also temporally related. In [Belardinelli *et al.*, 2012a] both features were shown to entail a decidable model checking problem through abstraction.

3 Undecidability for Bounded AC-QIS

In this section we show that in general the model checking problem for bounded AC-QIS is undecidable. We assume standard definitions of Turing machines and reductions [Papadimitriou, 1994]. Let $\mathcal{T} = \langle \mathcal{Q}, \Sigma, q_0, q_h, \delta \rangle$ be a deterministic Turing machine, where $\mathcal{Q}$ is the finite set of states, Σ

is the finite alphabet, $q_0 \in \mathcal{Q}$ is the initial state, $q_h \in \mathcal{Q}$ is the halting state, and δ is the transition function. We assume δ to be a function $\mathcal{Q} \times \Sigma \to \mathcal{Q} \times \Sigma \times \{L, R\}$ with $\delta(q, c) = (q', c', d)$ representing a transition from state q to state q', with symbols c and c' being read and written respectively, and head direction *L*eft or *R*ight respectively. W.l.o.g. $\mathcal{T}$ uses only the righthand half of the tape.

Given the Turing machine $\mathcal{T}$ with input string $in = in_0, \ldots, in_k \in \Sigma^*$, we define the AC-QIS $\mathcal{P}_{\mathcal{T},in} = \langle \mathcal{D}, U, \mathcal{S}, D_0, \to \rangle$ as follows. Let $\mathcal{D}$ be the database schema $\{C/2, Q/1, S/1, H/0\}$; while $U = \mathbb{N} \cup \mathcal{Q} \cup \Sigma$. Further, let $\mathcal{D}_{n,m} \subset \mathcal{D}(U)$ be the set of $\mathcal{D}$-instances $D_{n,m}$ containing exactly one pair (n, m) in the interpretation of the predicate symbol C, for $n, m \in \mathbb{N}$, that is, $D(C) = \{(n, m)\}$. Intuitively, $D(C) = \{(n, m)\}$ indicates that D represents the n-th cell in the m-th step of the computation of $\mathcal{T}$ on input in; while Q and S encode the state and symbol in the cell respectively. Finally, H is a boolean variable expressing whether the reading/writing head is pointing to the cell.

The initial state D_0 is defined, by using infix notation for relations, as $D_{0,0} = \{C(0,0), Q(q_0), S(in_0), \top\}$. The transition relation $\to$ for $\mathcal{P}_{\mathcal{T},in}$ is given as follows: *(i)* $D_{0,0} \to D_{1,0}$ where $D_{1,0} = \{C(1,0), Q(q_0), S(in_1), \bot\}$; *(ii)* for $1 \leq n < k$, $D_{n,0} \to D_{n+1,0}$ if $D_{n+1,0} = \{C(n+1, 0), Q(q_0), S(in_{n+1}), \bot\}$; *(iii)* for $n \geq k$, $D_{n,0} \to D_{n+1,0}$ if $D_{n+1,0} = \{C(n+1, 0), Q(q_0), \emptyset, \bot\}$.

Moreover, for $n, m \in \mathbb{N}$, $d \in \{L, R\}$, $D_{n,m} \to D_{n,m+1}$ iff:

- $\delta(q, c) = (q', c', d)$, $D_{n,m}(Q) = \{q\}$, $D_{n,m}(S) = \{c\}$, $D_{n,m}(H) = \top$ and $D_{n,m+1}(Q) = \{q'\}$, $D_{n,m+1}(S) = \{c'\}$, $D_{n,m}(H) = \bot$; or
- $\delta(q, c) = (q', c', L)$, $D_{n+1,m}(Q) = \{q\}$, $D_{n+1,m}(S) = \{c\}$, $D_{n+1,m}(H) = \top$, $D_{n,m}(Q) = \{q''\}$, $D_{n,m}(S) = \{c''\}$, $D_{n,m}(H) = \bot$ and $D_{n,m+1}(Q) = \{q'\}$, $D_{n,m+1}(S) = \{c''\}$, $D_{n,m+1}(H) = \top$; or
- $\delta(q, c) = (q', c', R)$, $D_{n-1,m}(Q) = \{q\}$, $D_{n-1,m}(S) = \{c\}$, $D_{n-1,m}(H) = \top$, $D_{n,m}(Q) = \{q''\}$, $D_{n,m}(S) = \{c''\}$, $D_{n,m}(H) = \bot$ and $D_{n,m+1}(Q) = \{q'\}$, $D_{n,m+1}(S) = \{c''\}$, $D_{n,m+1}(H) = \top$; or
- $D_{n-1,m}(H) = D_{n+1,m}(H) = \bot$, $D_{n,m}(Q) = \{q''\}$, $D_{n,m}(S) = \{c''\}$, $D_{n,m}(H) = \bot$ and $D_{n,m+1}(Q) = \{q''\}$, $D_{n,m+1}(S) = \{c''\}$, $D_{n,m+1}(H) = \bot$.

As standard $\mathcal{S}$ in $\mathcal{P}_{\mathcal{T},in}$ is the set of all $\mathcal{D}$-instances reachable from D_0. Notice that by definition of the transition relation, $\to$ mimicks the transition function δ. The $\mathcal{D}$ instances $D_{0,0}, \ldots, D_{k,0}$ contain the input as the interpretation of the predicate symbol S; at the beginning of the computation the head is on the initial cell $D_{0,0}$. The $\mathcal{D}$ instances $D_{0,m}, D_{1,m}, \ldots$ mimick the m-th step of the computation of $\mathcal{T}$ on in. Further, for each $m \in \mathbb{N}$, there exists exactly one $n \in \mathbb{N}$ s.t. $D_{n,m}(H) = \top$. This reflects the fact that the Turing machine is deterministic.

By the definition of D_0 and $\to$ it can readily be seen that the AC-QIS $\mathcal{P}_{\mathcal{T},in}$ is bounded. In fact, for all $D \in \mathcal{S}$, $|ad(D)| \leq 4$. Thus, $\mathcal{P}_{\mathcal{T},in}$ is indeed a bounded AC-QIS. We can now prove the following result on the AC-QIS $\mathcal{P}_{\mathcal{T},in}$.

Lemma 1 *$\mathcal{P}_{\mathcal{T},in} \models EF\ Q(q_h)$ iff the Turing machine $\mathcal{T}$ halts on input in.*

Proof (sketch). The proof relies on the fact that $\to$ represents faithfully the transition function δ. $\Leftarrow$ If $\mathcal{T}$ halts on input in, then there exists a step in the computation of $\mathcal{T}$ on input in, say the m-th, where $\mathcal{T}$ enters state q_h while reading the n-th cell, say. By the definition of $\mathcal{P}_{\mathcal{T},in}$ this means that for $D_{n,m} \in \mathcal{S}$, $D_{n,m}(Q) = \{q_h\}$. Hence, $\mathcal{P}_{\mathcal{T},in} \models EF\ Q(q_h)$. $\Rightarrow$ If $\mathcal{P}_{\mathcal{T},in} \models EF\ Q(q_h)$ then there exists $D_{n,m} \in \mathcal{S}$ s.t. $D_{n,m}(Q) = \{q_h\}$. Since the transition relation $\to$ mimicks the transition function δ, for each $k \in \mathbb{N}$, the sequence $D_{0,k}, D_{1,k}, \ldots$ represents the k-th step in the computation of $\mathcal{T}$ on input in. In particular, $D_{n,m}(Q) = \{q_h\}$ implies that $\mathcal{T}$ reaches the halting state at the m-th step, while reading the n-th cell. Thus, the Turing machine $\mathcal{T}$ halts on input in. □

As a consequence of the undecidability of the halting problem, we obtain the main result of this section.

Theorem 2 *Model checking bounded AC-QIS against FO-CTL specifications is undecidable.*

Notice that the AC-QIS $\mathcal{P}_{\mathcal{T},in}$ obtained from the Turing machine $\mathcal{T}$ and input in is not uniform. To see this, suppose that $\delta(q, c) = (q', c', d)$, $D_{n,m} = \langle \{(n, m)\}, \{q\}, \{c\}, \top \rangle$ and $D_{n,m+1} = \langle \{(n, m+1)\}, \{q'\}, \{c'\}, \bot \rangle$. By definition of the transition relation $\to$ we have that $D_{n,m} \to D_{n,m+1}$. Moreover, for $D_{n',m'} = \langle \{(n', m')\}, \{q''\}, \{c''\}, \top \rangle$ and $D_{n',m''} = \langle \{(n', m'')\}, \{q'''\}, \{c'''\}, \bot \rangle$ we have that $D_{n,m} \oplus D_{n,m+1} \simeq D_{n',m'} \oplus D_{n',m''}$. However, it is not the case that $D_{n',m'} \to D_{n',m''}$ whenever $m'' \neq m' + 1$. As a result, the AC-QIS $\mathcal{P}_{\mathcal{T},in}$ is not uniform in general; therefore the technique developed in [Belardinelli *et al.*, 2012a; Hariri *et al.*, 2012] cannot be applied here.

In the following section we show that, notwithstanding this undecidability result, a decidable, although incomplete, model checking procedure can be given if the specifications are restricted to the universal fragment of FO-CTL.

4 Counterpart Semantics

In this section we introduce a semantics for FO-CTL based on counterpart models. Then, we define a notion of simulation for these systems and show that the simulation relation preserves the interpretation of formulas in the universal fragment of FO-CTL. We firstly introduce counterpart models as a generalisation of AC-QIS.

Definition 7 (c-model) *A* counterpart model *is a tuple $\mathcal{M} = \langle S, s_0, \to, U, C, I \rangle$ s.t.* (i) *S is a non-empty* set of states*;* (ii) *s_0 is the* initial state*;* (iii) *$\to$ is a serial binary* transition relation *on S; and* (iv) *for $s, s' \in S$,*

- *$U(s)$ is a non-empty* set of individuals*;*
- *$C_{s,s'}$ is a serial* counterpart relation *on $U(s) \times U(s')$;*
- *I is a first-order interpretation, i.e.,* (i) *if P^n is an n-ary predicate symbol and $s \in S$, then $I(P^n, s)$ is an n-ary relation on $U(s)$; and* (ii) *if $c \in$ Con, then $I(c, s) \in U(s)$.*

Similarly to AC-QIS, the active domain *ad*(s) of a state s is defined as the set of all individuals occurring in some predicate interpretation $I(P^n, s)$. Counterpart models can be thought of as generalisations of AC-QIS, as an AC-QIS can

be seen as a c-model on a finite language, where the counterpart relation is the identity, and constants are interpreted as themselves:

Remark 1 *Let $\mathcal{M} = \langle S, s_0, \to, U, C, I\rangle$ be a c-model and $\mathcal{D}$ a database schema containing some predicate symbols interpreted by I. If for all $s, s' \in S$,* (i) *the counterpart relation $C_{s,s'}$ is the identity; and* (ii) *$I(c, s) = c$, then $\mathcal{M}_{\mathcal{D}} = \langle \mathcal{D}, \bigcup_{s\in S} U(s), S, s_0, \to\rangle$ is an AC-QIS.*

On the other hand, any AC-QIS $\mathcal{P}$ can be seen as a c-model:

Remark 2 *Let $\mathcal{P} = \langle \mathcal{D}, U, \mathcal{S}, D_0, \to\rangle$ be an AC-QIS and consider $\mathcal{M}_{\mathcal{P}} = \langle \mathcal{S}, D_0, \to, U', C, I\rangle$, where for all $D, D' \in \mathcal{S}$* (i) *$U'(D) = U$;* (ii) *$C_{D,D'}$ is the identity relation;* (iii) *$I(c, D) = c$; and* (iv) *$I(P^n, D) = D(P^n)$. Then $\mathcal{M}_{\mathcal{P}}$ is a c-model.*

The notion of *run* is defined as for AC-QIS. The relation C^* is the transitive closure of C, i.e., $C^*_{s,s'}(a, a')$ iff there is a sequence $s^0 \to \ldots \to s^k$ s.t. $s^0 = s$, $s^k = s'$, and there are $a^0, \ldots, a^k$ s.t. $a^0 = a$, $a^k = a'$, and $C_{s^i,s^{i+1}}(a^i, a^{i+1})$ for $i < k$.

To define satisfaction for FO-CTL formulas in counterpart models, we consider typed languages and finitary assignments. This is standard when working in counterpart semantics [Brauner and Ghilardi, 2007; Corsi, 2001]. Specifically, every variable $x_j \in$ *Var* is a term of type n, or n-term, for $n \geq j$; while every constant $c \in$ *Con* is an n-term.

Definition 8 (FO-CTL$_T$) *The typed language FO-CTL$_T$ contains all n-formulas $\phi : n$, for $n \in \mathbb{N}$, defined as follows:*

- *if P^m is an m-ary predicate symbol and $\vec{t}$ is an m-tuple of n-terms, then $P^m(\vec{t})$ is an (atomic) n-formula;*
- *if ψ, ψ' are n-formulas, then $\neg\psi$, $\psi \to \psi'$ are n-formulas;*
- *if ψ, ψ' are m-formulas and $\vec{t}$ is an m-tuple of n-terms, then $(AX\psi)(\vec{t})$, $(A\psi U\psi')(\vec{t})$, $(E\psi U\psi')(\vec{t})$ are n-formulas;*
- *if ψ is an $(n+1)$-formula, then $\forall x_{n+1}\psi$ is an n-formula;*

The other logical operators are defined as standard. The formula $AX\phi : n$ is a shorthand for $(AX\phi)(x_1, \ldots, x_n) : n$; similarly for $A\phi U\phi' : n$ and $E\phi U\phi' : n$. In what follows we consider also the sublanguage FO-ACTL$_T$, which is obtained by restricting FO-CTL$_T$ to the universal modalities AX and AU. Also, FO$_T$ is the non-modal fragment of FO-CTL$_T$.

The meaning of a typed formula $\phi : n$ at a state s can intuitively be understood as a subset of $U(s)^n$, i.e., the set of n-tuples satisfying $\phi : n$ at s. Therefore, the definition of satisfaction is given by means of finitary assignments, where an n-assignment in s is an n-tuple $\vec{a}$ of elements in $U(s)$. Let t be an n-term, the valuation $\vec{a}(t)$ for the n-assignment $\vec{a}$ is equal to a_j if $t = x_j$; otherwise, $\vec{a}(t) = a_n$ whenever $t = c$.

Definition 9 (Semantics of FO-CTL$_T$) *The satisfaction relation $\models$ for a state $s \in \mathcal{M}$, a typed formula $\phi : n$ and an n-assignment $\vec{a}$ is inductively defined as follows:*

$(\mathcal{M}^{\vec{a}}, s) \models P^m(\vec{t})$ *iff* $\langle \vec{a}(t_1), \ldots, \vec{a}(t_m)\rangle \in I(P^m, s)$

$(\mathcal{M}^{\vec{a}}, s) \models \neg\psi$ *iff* $(\mathcal{M}^{\vec{a}}, s) \not\models \psi$

$(\mathcal{M}^{\vec{a}}, s) \models \psi \to \psi'$ *iff* $(\mathcal{M}^{\vec{a}}, s) \not\models \psi$ *or* $(\mathcal{M}^{\vec{a}}, s) \models \psi'$

$(\mathcal{M}^{\vec{a}}, s) \models (AX\psi)(\vec{t})$ *iff for all r, $\vec{b} \in U(r^1)$, if $r^0 = s$ and $C_{s,r^1}(\vec{a}(t_i), b_i)$ then $(\mathcal{M}^{\vec{b}}, r^1) \models \psi$*

$(\mathcal{M}^{\vec{a}}, s) \models (A\varphi U\varphi')(\vec{t})$ *iff for all r, if $r^0 = s$ then there are $k \geq 0$, $\vec{b} \in U(r^k)$ s.t. $C^*_{s,r^k}(\vec{a}(t_i), b_i)$ and $(\mathcal{M}^{\vec{b}}, r^k) \models \varphi'$, and for all j, $\vec{c} \in U(r^j)$, if $0 \leq j < k$ and $C^*_{s,r^j}(\vec{a}(t_i), c_i)$ then $(\mathcal{M}^{\vec{c}}, r^j) \models \varphi$*

$(\mathcal{M}^{\vec{a}}, s) \models (E\varphi U\varphi')(\vec{t})$ *iff there is r s.t. $r^0 = s$, and $k \geq 0$, $\vec{b} \in U(r^k)$ s.t. $C^*_{s,r^k}(\vec{a}(t_i), b_i)$ and $(\mathcal{M}^{\vec{b}}, r^k) \models \varphi'$, and for all j, $\vec{c} \in U(r^j)$, if $0 \leq j < k$ and $C^*_{s,r^j}(\vec{a}(t_i), c_i)$ then $(\mathcal{M}^{\vec{c}}, r^j) \models \varphi$*

$(\mathcal{M}^{\vec{a}}, s) \models \forall x_{n+1}\psi$ *iff for all $a^* \in ad(s)$, $(\mathcal{M}^{\vec{a}\cdot a^*}, s) \models \psi$*

where $\vec{a} \cdot a^*$ is the $(n+1)$-assignment $\langle a_1, \ldots, a_n, a^*\rangle$.

An n-formula ϕ is true at a state s, *or $(\mathcal{M}, s) \models \phi$, iff it is satisfied by every n-assignment; ϕ is* true on a c-model $\mathcal{M}$, *or $\mathcal{M} \models \phi$, iff $(\mathcal{M}, s_0) \models \phi$.*

Notice that by considering c-models where the counterpart relation is the identity and constants are interpreted as themselves, we have that the relation of satisfaction in Def. 9 reduces to the notion in Def. 4. Thus, c-models can really be seen as a generalisation of AC-QIS. We state this result formally. Firstly, we define a translation π_n, for $n \in \mathbb{N}$, from FO-CTL to FO-CTL$_T$. Given an FO-CTL formula ϕ and n greater than or equal to the maximum k such that x_k occurs in ϕ, the n-formula $\pi_n(\phi)$ in FO-CTL$_T$ is inductively defined as follows:

$$
\begin{array}{lcl}
\pi_n(P^m(\vec{t})) & := & P^m(\vec{t}) \\
\pi_n(\neg\psi) & := & \neg\pi_n(\psi) \\
\pi_n(\psi \to \psi') & := & \pi_n(\psi) \to \pi_n(\psi') \\
\pi_n(AX\psi) & := & AX\pi_n(\psi) \\
\pi_n(A\psi U\psi') & := & A\pi_n(\psi)U\pi_n(\psi') \\
\pi_n(\forall x_i\psi) & := & \forall x_{n+1}(\pi_n(\psi)[x_i/x_{n+1}])
\end{array}
$$

Notice that π_n simply renames bound variables in ϕ. We can now state the following result on the relation between an AC-QIS $\mathcal{P}$ and the corresponding c-models $\mathcal{M}_{\mathcal{P}}$.

Lemma 3 *Let $\mathcal{P}$ be an AC-QIS, $\phi[\vec{x}] \in$ FO-CTL with free variables $\vec{x}$, and $\sigma(\vec{x}) = \vec{a}$. We have that*

$$(\mathcal{P}^\sigma, D) \models \phi[\vec{x}] \quad \textit{iff} \quad (\mathcal{M}^{\vec{a}}_{\mathcal{P}}, D) \models \pi_n(\phi[\vec{x}])$$

Proof (sketch). By induction on the length of ϕ. □

This result allows us to model check an AC-QIS by verifying the corresponding c-model.

We now present a notion of simulation for counterpart models and we show that it preserves the satisfaction of formulas in FO-ACTL$_T$. Firstly, we introduce some preliminary definitions. In what follows we consider the c-models $\mathcal{M} = \langle S, s_0, \to, U, C, I\rangle$ and $\mathcal{M}' = \langle S', s'_0, \to', U', C', I'\rangle$, with $s \in S$ and $s' \in S'$.

Definition 10 *A state s'* simulates *s, or $s \preceq s'$, iff there exists a surjective mapping $\iota : U(s) \to U'(s')$ s.t.* (i) *for every constant c, $I(c, s) = \iota(I(c, s'))$;* (ii) *for every P_i, $\vec{u} \in U(s)^{q_i}$, $\vec{u} \in I(P_i, s)$ iff $\iota(\vec{u}) \in I(P_i, s')$.*

Any function ι as above is a *witness* for $s \preceq s'$. We write $s \overset{\iota}{\preceq} s'$ to state this explicitly.

Definition 11 *Let $\vec{a} \in U(s)^n$ and $\vec{a}' \in U'(s')^n$ be n-assignments, $(s', \vec{a}')$* simulates *$(s, \vec{a})$, or $(s, \vec{a}) \preceq (s', \vec{a}')$, iff for some witness ι,* (i) *$s \overset{\iota}{\preceq} s'$; and* (ii) *$\iota(\vec{a}) = \vec{a}'$.*

We overload the symbol $\preceq$ for representing the relations in Def. 10 and 11; the difference will be clear from the context. Notice that $\preceq$ is a transitive relation. Also, simulations preserve the interpretation of FO-formulas:

Lemma 4 *If $(s, \vec{a}) \preceq (s', \vec{a}')$, then for each n-formula ϕ in FO$_T$,*

$$(\mathcal{M}^{\vec{a}}, s) \models \phi \quad \textit{iff} \quad (\mathcal{M}'^{\vec{a}'}, s') \models \phi$$

Proof (sketch). By induction on the length and type of ϕ. □

We now introduce the notion of simulation on c-models, which is then used to extend the result in Lem. 4 to FO-ACTL$_T$.

Definition 12 (Simulation) *A c-model $\mathcal{M}'$* simulates *$\mathcal{M}$, or $\mathcal{M} \preceq \mathcal{M}'$, iff* (i) *for every $\vec{a}'_0 \in U'(s'_0)^n$ there exists $\vec{a}_0 \in U(s_0)^n$ s.t. $(s_0, \vec{a}_0) \preceq (s'_0, \vec{a}'_0)$; and* (ii) *if $(s, \vec{a}) \preceq (s', \vec{a}')$ then for every t, $\vec{b} \in U(t)^n$, if $s \to t$ and $C_{s,t}(\vec{a}, \vec{b})$, then there are $t' \in S'$, $\vec{b}' \in U(t')^n$ s.t. $s' \to t'$, $C_{s',t'}(\vec{a}', \vec{b}')$, and $(t, \vec{b}) \preceq (t', \vec{b}')$.*

We use the same symbol $\preceq$ to express a simulation between c-models and states; the difference will be clear from the context. Also notice that, since, according to Remark 2, AC-QIS are a specific subclass of c-models, the definition of simulation for the latter applies also to the former. Further, the simulation relation on c-models preserves the satisfaction of formulas in FO-ACTL$_T$.

Theorem 5 *Suppose that $\mathcal{M} \preceq \mathcal{M}'$ and $(s, \vec{a}) \preceq (s', \vec{a}')$, then for each n-formula ϕ in FO-ACTL$_T$, we have*

$$(\mathcal{M}'^{\vec{a}'}, s') \models \phi \quad \Rightarrow \quad (\mathcal{M}^{\vec{a}}, s) \models \phi$$

Proof (sketch). By induction on the length and type of ϕ. The base case and the inductive cases for propositional connectives and quantifiers are proved as in Lem. 4. For $\phi = (AX\psi)(\vec{t})$, suppose that $(\mathcal{M}^{\vec{a}}, s) \not\models \phi$. Then, there exists a run r, $\vec{b} \in U(r^1)$ s.t. $r^0 = s$, $C_{s,r^1}(\vec{a}(t_i), b_i)$ and $(\mathcal{M}^{\vec{b}}, r^1) \not\models \psi$. By definition of simulation there are $t' \in S'$, $\vec{b}' \in U'(t')^n$ s.t. $s' \to' t'$, $C'_{s',t'}(\vec{a}'(t_i), b'_i)$, and $(r^1, \vec{b}) \preceq (t', \vec{b}')$. Notice that by seriality $s' \to' t'$ can be extended to an infinite run r' s.t. $r'^0 = s'$ and $r'^1 = t'$. Further, by induction hypothesis we have that $(\mathcal{M}'^{\vec{b}'}, r'^1) \not\models \psi$. Thus, $(\mathcal{M}'^{\vec{a}'}, s') \not\models \phi$. The case for $\phi = (A\psi U\psi')(\vec{t})$ is proved similarly. □

From Thm. 5 we immediately obtain the following result.

Corollary 6 *If $\mathcal{M} \preceq \mathcal{M}'$, then for every n-formula $\phi \in$ FO-ACTL$_T$, $\vec{a}'_0 \in U'(s'_0)^n$, there exists $\vec{a}_0 \in U(s_0)^n$ s.t.*

$$(\mathcal{M}'^{\vec{a}'_0}, s'_0) \models \phi \quad \Rightarrow \quad (\mathcal{M}^{\vec{a}_0}, s_0) \models \phi$$

Proof (sketch). By the definition of simulation for every $\vec{a}'_0 \in U'(s'_0)^n$ there exists $\vec{a}_0 \in U(s_0)^n$ s.t. $(s_0, \vec{a}_0) \preceq (s'_0, \vec{a}'_0)$. Moreover, by Thm. 5 we obtain that $(\mathcal{M}'^{\vec{a}'_0}, s'_0) \models \phi$ implies $(\mathcal{M}^{\vec{a}_0}, s_0) \models \phi$. □

By Lem. 3 and Cor. 6 we can solve the model checking problem for an AC-QIS $\mathcal{P}$ and an FO-ACTL formula ϕ by considering a c-model $\mathcal{M}'$ that is similar to $\mathcal{M}_\mathcal{P}$. By Cor. 6, if $(\mathcal{M}'^{\vec{a}'_0}, s'_0) \models \pi_n(\phi)$ for some n-assignment $\vec{a}'_0$, then there exists $\vec{a}_0 \in U(s_0)$ s.t. $(\mathcal{M}_\mathcal{P}^{\vec{a}_0}, s_0) \models \pi_n(\phi)$. Moreover, by Lem. 3, if $(\mathcal{M}_\mathcal{P}^{\vec{a}_0}, s_0) \models \pi_n(\phi)$ then $(\mathcal{P}^\sigma, s_0) \models \phi$ for some assignment σ that agrees with $\vec{a}_0$ on the free variables in ϕ. Thus, a positive solution to the model checking problem for the abstract c-model $\mathcal{M}'$ implies a positive solution also for the concrete AC-QIS $\mathcal{P}$. In the following section we analyse the conditions under which the abstract c-model $\mathcal{M}'$ is finite.

5 Finite Abstractions

In this section we introduce the *abstraction* of a c-model $\mathcal{M}$ and show that it is similar to $\mathcal{M}$. Further, we identify the conditions under which such abstraction is finite, thus allowing the verification of infinite-state AC-QIS by the results in Section 4.

Firstly, we define $[s]$ as the equivalence class of s according to the symmetric closure $\approx$ of the relation $\preceq$. Further, for $s \in [t]$, $[s, a]_{[t]}$ is the equivalence class of (s, a) according to the symmetric closure $\approx$ of $\preceq$. Notice that we overload the symbol $\approx$ as we did with $\preceq$; the distinction will be clear from the context. Also, $\approx$ is not to be confused with the isomorphism relation $\simeq$.

Definition 13 (Abstraction) *Given a c-model $\mathcal{M} = \langle S, s_0, \to, U, C, I\rangle$, the* abstraction *of $\mathcal{M}$ is a tuple $\mathcal{M}' = \langle S', s'_0, \to', U', C', I'\rangle$ s.t.* (i) *$S' = \{[s] \mid s \in S\}$;* (ii) *$s'_0 = [s_0]$; and for every $[s], [s'] \in S'$,*

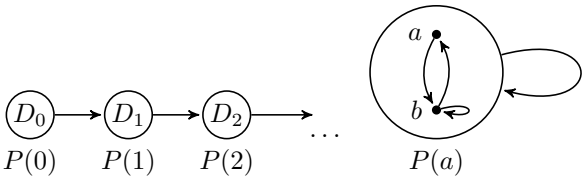
- *$[s] \to' [s']$ iff there are $u \in [s]$, $v \in [s']$ s.t. $u \to v$*
- *$U([s]) = \{[s, a]_{[s]} \mid a \in U(s)\}$*
- *$C'_{[s],[s']} = \{([s, a]_{[s]}, [s', b]_{[s']}) \mid$ there are $(u, a') \in [s, a]_{[s]}$, $(v, b') \in [s', b]_{[s']}$ and $C_{u,v}(a', b')\}$*
- *$I'(c, s) = [s, I(c, s)]_{[s]}$ and $[s, \vec{a}]_{[s]} \in I'(P, [s])$ iff $\vec{a} \in I(P, s)$.*

It can be shown that the abstraction of a c-model $\mathcal{M}$ is also a c-model. In particular, the interpretation I' is well-defined as it is independent from the specific representative: if $(s', a') \in [s, \vec{a}]_{[s]}$, then $(s', a') \approx (s, a)$. In particular, $\vec{a} \in I(P, s)$ iff $\vec{a}' \in I(P, s')$.

Given a c-model $\mathcal{M}$ and its abstraction $\mathcal{M}'$, we define a mapping $f : \mathcal{M} \to \mathcal{M}'$ s.t. $f(s, \vec{a}) = ([s], [s, \vec{a}]_{[s]})$. We now prove that the mapping f defines a simulation relation.

Theorem 7 *Let $\mathcal{M}$ be a c-model and $\mathcal{M}'$ its abstraction. The relation $F((s, \vec{a}), f(s, \vec{a}))$ is a simulation between $\mathcal{M}$ and $\mathcal{M}'$.*

Proof (sketch). Firstly notice that for every $[s_0, \vec{a}_0]_{[s_0]} \in U'(s'_0)^n$ there exists $\vec{a}_0 \in U(s_0)^n$ and $F((s_0, \vec{a}_0), f(s_0, \vec{a}_0))$. To we show that $(s, \vec{a}) \preceq f(s, \vec{a})$, we define a mapping $\iota : U(s) \to U'([s])$ s.t. $\iota(a) = [s, a]_{[s]}$ for $a \in U(s)$. It is clear that $s \overset{\iota}{\preceq} [s]$ by the definition of I'. Thus, $(s, \vec{a}) \preceq f(s, \vec{a})$. Finally, suppose that for $t \in S$ and $\vec{b} \in U(t)^n$, we have that $s \to t$ and $C_{s,t}(\vec{a}, \vec{b})$. By the definition of $\mathcal{M}'$ we clearly see that $[s] \to [t]$. Also, for every $a \in \vec{a}$,



(a) The AC-QIS $\mathcal{P}$ (b) The c-model $\mathcal{M}'$

$b \in \vec{b}$, $(s,a) \in [s,a]_{[s]}$, $(t,b) \in [t,b]_{[t]}$ and $C_{s,t}(a,b)$. Hence, $C_{[s],[t]}([s,\vec{a}]_{[s]}, [t,\vec{b}]_{[t]})$, and $F((t,\vec{b}), f(t,\vec{b}))$. □

By Cor. 6 and Thm. 7 we obtain the following result.

Corollary 8 *Let $\mathcal{M}$ be a c-model and $\mathcal{M}'$ its abstraction . For every n-formula ϕ in FO-ACTL$_T$, for every $\vec{a}_0 \in U(s_0)$,*

$$(\mathcal{M}'^{[s_0,\vec{a}_0]_{[s_0]}}, [s_0]) \models \phi \quad \Rightarrow \quad (\mathcal{M}^{\vec{a}_0}, s_0) \models \phi$$

Now we investigate the problem of determining sufficient conditions for the abstraction of a c-model $\mathcal{M}$ to be finite. This will allow the verification of an infinite-state AC-QIS $\mathcal{P}$ by model checking the finite abstraction of $\mathcal{M}_\mathcal{P}$.

Theorem 9 *If $\mathcal{P}$ is a bounded AC-QIS, then the abstraction $\mathcal{M}'$ of the c-model $\mathcal{M}_\mathcal{P}$ is finite.*

Proof (sketch). Notice that if $\mathcal{P}$ is bounded, then there exists $b \in \mathbb{N}$ s.t. for every $D \in \mathcal{S}$, $|ad(D)| \leq b$. This means that there is only a finite number of equivalence classes $[D]$ according to relation $\approx$. Hence, S' is finite. Further, since the active domain of each D is finite, there is only a finite number of equivalence classes $[D,a]_{[D]}$, for $a \in U$. Thus, each $U([D])$ is also finite. □

As a consequence of Cor. 8 and Thm. 9, to verify an FO-ACTL formula ϕ on a bounded AC-QIS $\mathcal{P}$, we can model check $\pi_n(\phi) \in$ FO-ACTL$_T$ on the finite abstraction of $\mathcal{M}_\mathcal{P}$. We illustrate the procedure above by an example. Consider the AC-QIS $\mathcal{P}$ in Fig. (a), where *(i)* $\mathcal{D} = \{P/1\}$; *(ii)* $U = \mathbb{N}$; *(iii)* $S = \{D_n \mid n \in \mathbb{N}$ and $D_n(P) = \{n\}\}$; *(iv)* D_0 is the initial state; *(v)* for all $n \in \mathbb{N}$, $D_n \to D_{n+1}$. Firstly, notice that the AC-QIS $\mathcal{P}$ is not uniform, as $D_1 \oplus D_2 \simeq D_1 \oplus D_3$ and $D_1 \to D_2$, but not $D_1 \to D_3$. Thus, the techniques developed in [Belardinelli *et al.*, 2012a] cannot be applied. Consider also the following specifications in FO-ACTL: $\chi = \forall x AG(P(x) \to AX\neg P(x))$ and $\theta = \forall x AG(P(x) \to AXAG\neg P(x))$. Since $\mathcal{P}$ is infinite-state, χ and θ cannot be model checked directly on $\mathcal{P}$.

To define the abstraction $\mathcal{M}'$ of $\mathcal{P}$ we observe that for every $i, j \in \mathbb{N}$, $D_j \preceq D_i$ by the witness ι that maps j into i, and $\mathbb{N} \setminus \{j\}$ into $\mathbb{N} \setminus \{i\}$. Hence, $S' = \{S\}$ and $S \to' S$. Further, for every $i, j \in \mathbb{N}$, $(D_j, j) \preceq (D_i, i)$; while for $i' \neq i, j' \neq j$, $(D_j, j') \not\preceq (D_i, i')$. Hence, we obtain two equivalence classes $a = \{(D_n, n) \mid n \in \mathbb{N}\}$ and $b = \{(D_n, n') \mid n \neq n'\}$. Then the counterpart relation is defined as $C_{S,S} = \{(a,b),(b,b),(b,a)\}$, corresponding respectively to the transitions $(D_n, n) \to (D_{n+1}, n)$, $(D_n, n') \to (D_{n+1}, n')$ for $n' \neq n$, $n' \neq n+1$, and $(D_{n-1}, n) \to (D_n, n)$. Finally, we have $I(P, S) = \{a\}$. The abstract c-model $\mathcal{M}'$ is illustrated in Fig. (b). We can now model check the typed versions of χ and θ, namely $\chi_T = \forall x_1 AG(P(x_1) \to AX\neg P(x_1))$ and $\theta_T = \forall x_1 AG(P(x_1) \to AXAG\neg P(x_1))$, on $\mathcal{M}'$. Clearly, $(\mathcal{M}', S) \models \chi_T$, while $(\mathcal{M}', S) \not\models \theta_T$. Hence, by Lem. 3 and Cor. 8 we can derive that $(\mathcal{P}, D_0) \models \chi$; while nothing can be said about θ in $\mathcal{P}$.

We conclude by remarking that in general the abstraction of an AC-QIS as defined in Def. 13 is not an AC-QIS, but a c-model. This motivates the introduction of c-models.

6 Conclusions and Further Work

As remarked earlier, a considerable amount of work has focused on the development of verification methodologies for artifact-centric systems [Belardinelli *et al.*, 2012a; 2012b; Deutsch *et al.*, 2009; Gerede and Su, 2007; Hariri *et al.*, 2012]. The-state-of-the-art in the area involves constructing finite abstractions of the artifact-system's infinite state space. So far work in the literature has shown that this process can be conducted whenever the artifact system is bounded and "uniform". In this paper we showed that there are noteworthy cases where this condition can be relaxed. Specifically, we demonstrated that in the case of universal specifications, finite abstractions can be constructed for bounded but non-uniform systems. This leads us to the following considerations.

From the theoretical point of view, it is of interest to ascertain whether a necessary condition exists for finite abstractions. Our current results show that uniformity is not necessary for partial decision procedures, but that at the same time, bounded systems do not in general admit finite abstractions. Still, none of these results are "tight" and it is an open problem whether the existence of finite abstractions can be characterised in terms of system conditions.

From an application standpoint, while uniformity covers a wide range of systems, the corresponding condition on the underlying databases, namely *genericity* [Abiteboul *et al.*, 1995], is satisfied only by a restricted number of systems. It follows that a large class of data-aware systems, such as those implemented by GSM [Heath *et al.*, 2011], are not "amenable" (in the sense of [Belardinelli *et al.*, 2012b]), hence may generate non-uniform models. Therefore, only the techniques introduced in this paper may be used to model check them. It would be of interest to implement the methodology of this paper into a toolkit for the practical verification of artifact-centric systems. The model checker GSMC [Gonzalez *et al.*, 2012] already targets the verification of artifact-centric systems, but it presently offers no support for data and deals with finite state spaces only. We leave both points above for further work.

Acknowledgements

This research was partly funded by the EC FP7 under grant n. 257593 (ACSI), and by the EPSRC under grant EP/I00520X.

References

[Abiteboul *et al.*, 1995] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[Baldoni *et al.*, 2012] Matteo Baldoni, Cristina Baroglio, Elisa Marengo, Viviana Patti, and Claudio Schifanella.

Flexible choreography-driven service selection. *Intelligenza Artificiale*, 6(1):97–115, 2012.

[Belardinelli *et al.*, 2011] Francesco Belardinelli, Alessio Lomuscio, and Fabio Patrizi. Verification of Deployed Artifact Systems via Data Abstraction. In *Proc. of the 9th International Conference on Service-Oriented Computing (ICSOC'11)*, pages 142–156, 2011.

[Belardinelli *et al.*, 2012a] Francesco Belardinelli, Alessio Lomuscio, and Fabio Patrizi. An Abstraction Technique for the Verification of Artifact-Centric Systems. In *Proc. of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, pages 319 – 328, 2012.

[Belardinelli *et al.*, 2012b] Francesco Belardinelli, Alessio Lomuscio, and Fabio Patrizi. Verification of GSM-based artifact-centric systems through finite abstraction. In *Proc. of the 10th International Conference on Service-Oriented Computing (ICSOC'12)*, pages 17–31, 2012.

[Brauner and Ghilardi, 2007] T. Brauner and S. Ghilardi. First-order modal logic. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, pages 549–620. Elsevier, 2007.

[Corsi, 2001] G. Corsi. *Counterparts and possible worlds. A study on quantified modal logic*, volume 21. CLUEB, Bologna, 2001.

[Denti and Omicini, 1999] Enrico Denti and Andrea Omicini. An architecture for tuple-based coordination of multi-agent systems. *Softw., Pract. Exper.*, 29(12):1103–1121, 1999.

[Deutsch *et al.*, 2009] A. Deutsch, R. Hull, F. Patrizi, and V. Vianu. Automatic Verification of Data-Centric Business Processes. In *Proc. of ICDT*, 2009.

[Gerede and Su, 2007] Cagdas E. Gerede and Jianwen Su. Specification and Verification of Artifact Behaviors in Business Process Models. In *Proc. of the 5th International Conference on Service-Oriented Computing (ICSOC'07)*, pages 181–192, 2007.

[Gonzalez *et al.*, 2012] Pavel Gonzalez, Andreas Griesmayer, and Alessio Lomuscio. Verifying GSM-Based Business Artifacts. In *Proc. of the 19th IEEE International Conference on Web Services (ICWS'12)*, pages 25–32, 2012.

[Hariri *et al.*, 2012] Babak Bagheri Hariri, Diego Calvanese, Giuseppe De Giacomo, Alin Deutsch, and Marco Montali. Verification of Relational Data-Centric Dynamic Systems with External Services. *CoRR*, abs/1203.0024, 2012.

[Heath *et al.*, 2011] Fenno Terry Heath, Rick Hull, and Roman Vaculín. Barcelona: A Design and Runtime Environment for Modeling and Execution of Artifact-centric Business Processes (demo). In *Proc. of the 9th International Conference on Business Process Management Demo Track (BPM'11)*, 2011.

[Hull *et al.*, 2011] Richard Hull, Elio Damaggio, Riccardo De Masellis, Fabiana Fournier, Manmohan Gupta, Fenno Terry Heath, III, Stacy Hobson, Mark Linehan, Sridhar Maradugu, Anil Nigam, Piwadee Noi Sukaviriya, and Roman Vaculin. Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, DEBS '11, pages 51–62, New York, NY, USA, 2011. ACM.

[Hull, 2008] R. Hull. Artifact-centric business process models: Brief survey of research results and challenges. In R. Meersman and Z. Tari, editors, *OTM Conferences (2)*, volume 5332 of *Lecture Notes in Computer Science*, pages 1152–1163. Springer, 2008.

[Lomuscio *et al.*, 2011] Alessio Lomuscio, Wojciech Penczek, Monika Solanki, and Maciej Szreter. Runtime monitoring of contract regulated web services. *Fundam. Inform.*, 111(3):339–355, 2011.

[Lomuscio *et al.*, 2012] Alessio Lomuscio, Hongyang Qu, and Monika Solanki. Towards Verifying Contract Regulated Service Composition. *Autonomous Agents and Multi-Agent Systems*, 24(3):345–373, 2012.

[Modgil *et al.*, 2009] Sanjay Modgil, Noura Faci, Felipe Rech Meneguzzi, Nir Oren, Simon Miles, and Michael Luck. A framework for monitoring agent-based normative systems. In Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman, editors, *AAMAS (1)*, pages 153–160. IFAAMAS, 2009.

[Omicini *et al.*, 2008] Andrea Omicini, Alessandro Ricci, and Mirko Viroli. Artifacts in the a&a meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, 2008.

[Papadimitriou, 1994] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[Singh and Huhns, 2005] Munindar P. Singh and Michael N. Huhns. *Service-Oriented Computing: Semantics, Processes, Agents*. Wiley, 2005.

[Winikoff, 2009] Michael Winikoff. Future directions for agent-based software engineering. *International Journal of Agent-Oriented Software Engineering*, 3(4):402–410, 2009.