ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-81

Model Checking Auctions as Artifact Systems: Decidability via Finite Abstraction

Francesco Belardinelli [1]

Abstract. The formal verification of auctions has recently received considerable attention in the AI and logic community. We tackle this problem by adopting methodologies and techniques originally developed for Artifact Systems, a novel paradigm in Service Oriented Computing. Specifically, we introduce a typed version of artifact-centric multi-agent systems (AC-MAS), a multi-agent setting for Artifact Systems, and consider the model checking problem against typed first-order temporal epistemic specifications. Notably, this formal framework is expressive enough to capture a relevant class of auctions: parallel English (ascending bid) auctions. We prove decidability of the model checking problem for AC-MAS via finite abstraction. In particular, we put forward a methodology to formally verify interesting properties of auctions.

1 Introduction

The formal verification of game structures is a topic of growing interest in the AI and logic community [2, 16, 25]. In particular, the verification of auctions has received considerable attention recently [3, 18, 26]. Indeed, it is hard to overestimate the relevance of auctions and auction-based mechanisms in a wide range of distributed systems [23, 19]. However, with some notable exceptions, most of the research on this topic has focus on the design of auctioning mechanisms, while the model checking problem has only partially been addressed.

In this paper we tackle the issues pertaining to model checking auctions by adopting methodologies and techniques originally developed for Artifact Systems, a novel paradigm for business processes [21]. Artifact Systems (AS) are best described in terms of interacting modules, or *artifacts*, which typically consist of a *data model*, accounting for the relational structure of data, and a *lifecycle*, describing the evolution of the system over time. To keep the verification task tractable, most contributions disregard the data content of artifacts as well as the agents implementing the services. Still, in Artifact Systems and auctions alike it is crucial to reason about the actions agents can perform, the knowledge they possess, as well as the states they can jointly reach. Hence, the formal verification of both AS and auctions can benefit from techniques developed in the area of *reasoning about knowledge* [15, 22].

Taking inspiration from the works above, this paper aims at providing a twofold contribution. Firstly, we put forward an agent-based abstraction techniques to model check Artifact Systems. Secondly, we apply this methodology to the formal verification of auctions. In this paper we focus on parallel English auctions and model these as artifact-centric multi-agent systems (AC-MAS) [6, 7], a multi-agent setting for Artifact Systems. Then, we tackle the model checking problem against specifications written in a first-order temporal epistemic logic. Notably, the specification language includes predicates whose interpretation might be infinite (e.g. total orders on rational numbers). This modelling choice calls for novel abstraction techniques with respect to the state-of-the-art. Specifically, the notion of *uniformity*, which has proved to be sufficient to obtain finite abstractions [6, 12] has to be recast to account for this more complex setting. Finally, we describe an abstraction techniques for AC-MAS, and prove that a specification is satisfied by a concrete, infinite-state AC-MAS iff it is satisfied by its finite abstraction. In particular, this result applies to parallel English auctions.

Related Work. To our knowledge [3, 18, 26] are among the first contributions to consider the formal verification of auctions. In [18] the problem of model checking strategy-proofness of Vickrey auctions is investigated; while [26] propose a formal approach to check for shilling behaviours in auctions. Overall, [3] is the contribution most closely related to the present work in spirit, as the authors also analyse the verification of agent-based English auctions, but a key difference is that their models abstract from the data content of auctions. On the more general subject of Artifact Systems verification, in [11, 13] this problem is investigated in relation to first-order linear-time specifications; while [20] considers data-centric dynamic systems. In both cases the specification language is synctactically restricted, while no such restriction is here considered. Other works considering these features are [5, 6, 7], upon which this paper builds. However, the task of formally verifying parallel English auctions calls for novel abstraction techniques with respect to the cited references.

Scheme of the Paper. In Sections 2-4 we present parallel English auctions and AC-MAS, a multi-agent framework for Artifact Systems. Also, we introduce the typed first-order temporal epistemic logic tFO-CTLK and state the corresponding model checking problem. In Section 5 we show that AC-MAS are expressive enough to model parallel English auctions. Sections 6 and 7 contain the main theoretical results of the paper: in Section 6 we define a notion of bisimulation for AC-MAS and in Section 7 we state sufficient conditions for the model checking problem to be decidable via finite abstraction. The technique is then applied to the formal verification of parallel English auctions. For reasons of space all proofs are omitted. However, an extended version of this paper with further details and selected proofs is available as [8].

2 Auctions

Hereafter we focus on parallel English (ascending bid) auctions [14]. This kind of auction is of particular interest in the present context, as it is common to a number of distributed scenarios, including popular

[1] Laboratoire IBISC, Université d'Evry, France, email: belardinelli@ibisc.fr

auctioning websites. In parallel English auctions we typically have a single auctioneer A and a finite number of bidders $B_1, \ldots, B_\ell$. The auctioneer puts on sale a finite number of items, starting from a *base price* that is public to all bidders. For sake of presentation, we consider the bidding process as structured in discrete rounds. At each round, the bidder can either choose to bid for a specific item or to skip the round. At the end of the bidding process, each item is assigned to the bidder with the highest offer. We assume that our bidders are rational and each of them has an intrinsic value for each item being auctioned: she is willing to buy the item for a price up to her *true value*, but not for any higher price. Also, each bidder keeps this information private from other bidders and the auctioneer.

We are interested in verifying auctions against properties concerning the evolution of the bidding process and the knowledge acquired by bidders. For instance, we might want to check that *(i)* the base price for each item is indeed known to all agents, and not only this but that the base price is actually common knowledge. Also, we might want to express that *(ii)* the true value of each bidders for each item is indeed unknown to the actioneer and the other bidders, and it remains so throughout the bidding process. Other specifications of interest might be liveness properties such as *(iii)* the bidders are always able to make a higher bid, unless they have already hit their true value.

We remark that model checking such properties is extremely complex. Indeed, prices are usually represented by real or rational numbers; hence auctions typically belong to the realm of infinite-state systems. In what follows we provide a formal model for auctions and prove that we can model check properties such as *(i)*-*(iii)* above by considering finite abstractions of concrete infinite-state auctions.

3 Artifact-centric Multi-agent Systems

We now fix the basic notation for databases used hereafter [1]. In what follows we assume a finite number of types $T_1, \ldots, T_k$.

Definition 1 (Db schema and instance) *A* (typed) database schema *is a finite set* $\mathcal{D} = \{P_1/a_1, \ldots, P_n/a_n, Q_1/b_1, \ldots, Q_m/b_m\}$ *of* typed relation symbols R *with arity* $c \in \mathbb{N}$ *and type* $T_{k_1}, \ldots, T_{k_c}$.

Given a countable interpretation domain U_h *for each type* T_h*, a* $\mathcal{D}$-instance *over* $U_1, \ldots, U_k$ *is a mapping* D *associating in a type-consistent way* (i) *each relation symbol* $P \in \mathcal{D}$ *with a* finite *a-ary relation* $D(P)$ *over* $U_{k_1} \times \ldots \times U_{k_a}$*, and* (ii) *each relation symbol* $Q \in \mathcal{D}$ *with a (*possibly infinite*) b-ary relation* $D(Q)$ *over* $U_{k_1} \times \ldots \times U_{k_b}$.

In Def. 1 we depart from the standard notion of db instance as the interpretation $D(Q)$ of a symbol $Q \in \mathcal{D}$ can be *infinite* in principle. Intuitively, the symbols Q are used to model background information on the interpretation domains, e.g. the total order $<$ on the set $\mathbb{Q}$ of rational numbers. The set of all $\mathcal{D}$-instances over $U_1, \ldots, U_k$ is denoted as $\mathcal{D}(\vec{U})$. The *active domain* $adom(D) = \langle adom_1(D), \ldots, adom_k(D)\rangle$ of a db instance D is a tuple where each $adom_h(D)$ is the set of all individuals in U_h occurring in some relation $D(P)$. Since $\mathcal{D}$ and each $D(P)$ are finite, so is each $adom_h(D)$. Notice that the relations $D(Q)$ do not contribute to the definition of the active domain. Finally, with an abuse of notation we write $f : U_h \to U'_h$ to express that f is a function s.t. for each type T_h, $f(u) \in U'_h$ if $u \in U_h$. We now introduce the disjoint union $\oplus$ of db instances. Let the *primed version* of the db schema $\mathcal{D}$ above be the db schema $\mathcal{D}' = \{P'_1/a_1, \ldots, P'_n/a_n, Q'_1/b_1, \ldots, Q'_m/b_m\}$.

Definition 2 (Disjoint union $\oplus$) *Given* $\mathcal{D}$*-instances* D *and* D', $D \oplus D'$ *is the* $(\mathcal{D} \cup \mathcal{D}')$*-instance s.t. for every relation symbol* R, $D \oplus D'(R) = D(R)$ *and* $D \oplus D'(R') = D'(R)$.

Intuitively, the operator $\oplus$ will be used to describe the transition of a system from a current state D to the successor state D'.

We now introduce a notion of *agent* inspired to multi-agent systems [6, 15].

Definition 3 (Agent) *Given a countable interpretation domain* U_h *for each type* T_h*, an* agent *is a tuple* $A = \langle \mathcal{D}, Act, Pr\rangle$ *s.t.* (i) $\mathcal{D}$ *is the* local database schema*;* (ii) *Act is a finite set of* (typed) actions $\alpha(\vec{T})$*, where the tuple* $\vec{T}$ *of types are the* formal parameters *of* α*; and* (ii) $Pr : \mathcal{D}(\vec{U}) \mapsto 2^{Act(\vec{U})}$ *is the* local protocol function*, where* $Act(\vec{U})$ *is the set of* ground actions $\alpha(\vec{u})$*, for* $\alpha(\vec{T}) \in Act$ *and* $\vec{u} \in \vec{U}^{|\vec{T}|}$ *a tuple of* (type-consistent) ground parameters.

As standard in multi-agent systems, each agent A performs the actions in Act according to the protocol function Pr. Moreover, we assume that A is in some local state $D \in \mathcal{D}(\vec{U})$, that is, the information she possesses is structured as a database.

As agents interact, we consider their composition.

Definition 4 (AC-MAS) *Given a countable interpretation domain* U_h *for each type* T_h *and a set* $Ag = \{A_0, \ldots, A_\ell\}$ *of agents* $A_i = \langle \mathcal{D}_i, Act_i, Pr_i\rangle$*, an* artifact-centric multi-agent system *is a tuple* $\mathcal{P} = \langle Ag, s_0, \tau\rangle$ *s.t.* (i) $s_0 \in \mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U})$ *is the* initial global state*; and* (ii) $\tau : \mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U}) \times Act(\vec{U}) \mapsto 2^{\mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U})}$ *is the* global transition function*, where* $Act(\vec{U}) = Act_0(\vec{U}) \times \ldots \times Act_\ell(\vec{U})$ *is the set of* joint (ground) actions*, and the transition* $\tau(s, \langle \alpha_0(\vec{u}_0), \ldots, \alpha_\ell(\vec{u}_\ell)\rangle)$ *is defined iff* $\alpha_i(\vec{u}_i) \in Pr_i(D_i)$ *for all* $i \leq \ell$.

AC-MAS are rich enough to formalize the framework of Artifact Systems, as shown in [5, 7] for instance. We now introduce some basic terminology. We denote a joint (ground) action as $\alpha(\vec{u})$ for $\alpha = \langle \alpha_0(\vec{T}_0), \ldots, \alpha_\ell(\vec{T}_\ell)\rangle$ and $\vec{u} = \langle \vec{u}_0, \ldots, \vec{u}_\ell\rangle$, and define the *transition relation* $s \to s'$ on global states iff $s \xrightarrow{\alpha(\vec{u})} s'$, i.e., $s' \in \tau(s, \alpha(\vec{u}))$ for some $\alpha(\vec{u}) \in Act(\vec{U})$. An *s-run* r is an infinite sequence $s^0 \to s^1 \to \cdots$, with $s^0 = s$. For $n \in \mathbb{N}$, we set $r(n) = s^n$. A state s' is *reachable from* s if there is an s-run r s.t. $r(i) = s'$ for some $i \geq 0$. Finally, we introduce $\mathcal{S}$ as the set of global states reachable from the initial state s_0. The following class of AC-MAS will feature prominently in the paper.

Definition 5 (Rigidity) *An AC-MAS* $\mathcal{P}$ *is* rigid *iff for every* $Q \in \mathcal{D}$, $s, s' \in \mathcal{S}$*, and* $D_i \in s$, $D_j \in s'$*, we have* $D_j(Q) = D_i(Q)$.

In rigid AC-MAS the symbols $Q \in \mathcal{D}$ have the same interpretation in all global states and for all agents, consistently with the intuition that these represent persistent properties of the interpretation domains known to all agents. We refer to this relation as $\mathcal{P}(Q)$. Further, two global states $s = \langle D_0, \ldots, D_\ell\rangle$ and $s' = \langle D'_0, \ldots, D'_\ell\rangle$ in $\mathcal{S}$ are *indistinguishable* for agent A_i, or $s \sim_i s'$, if $D_i = D'_i$ [15]. Finally, for technical purposes we refer to the *global* db schema $\mathcal{D} = \bigcup_{A_i \in Ag} \mathcal{D}_i$ of an AC-MAS. Then, each state s is associated with the $\mathcal{D}$-instance $D_s \in \mathcal{D}(\vec{U})$ s.t. $D_s(R) = \bigcup_{A_i \in Ag} D_i(R)$. Also, we write *adom*$(s)$ for *adom*(D_s). Notice that for every state s, the associated D_s is unique, whereas the converse is not true in general. Furthermore, we lift the disjoint union operator $\oplus$ to global states so that $s \oplus s'$ is defined as $\langle D_0 \oplus D'_0, \ldots, D_\ell \oplus D'_\ell\rangle$.

4 The Typed Logic tFO-CTLK

We now consider the specification language for AC-MAS. For each type T_h let *Var*$_h$ (resp. *Con*$_h$) be a countable set of *(typed) variables* (resp. *constants*). A *(typed) term* is any element $t \in Var_h \cup Con_h$.

Definition 6 (tFO-CTLK) *The typed first-order CTLK formulas φ over a db schema $\mathcal{D}$ are defined by the following BNF:*

$$\varphi ::= t = t' \mid R(\vec{t}) \mid \neg\varphi \mid \varphi \to \varphi \mid \forall x\varphi \mid AX\varphi \mid A\varphi U\varphi \mid E\varphi U\varphi \mid K_i\varphi \mid C\varphi$$

where $R \in \mathcal{D}$, $\vec{t}$ is a type-consistent tuple of terms, t, t' are terms of the same type, $x \in Var_h$, and $i \leq \ell$.

We introduce the abbreviations $\exists$, $\wedge$, $\vee$, $\neq$, and define free and bound variables as standard. For a formula φ, $var_h(\varphi)$ (resp. $fr_h(\varphi)$) and $con_h(\varphi)$ denotes the set of its variables (resp. free variables and constants) of type T_h. A *sentence* is a formula with no free variables. We use the standard abbreviations $EX\varphi$, $AF\varphi$, $AG\varphi$, $EF\varphi$, and $EG\varphi$. By Def. 6 free variables can occur within the scope of modal operators; this is a major feature of the present framework in comparison with, for instance, [9, 17].

We now assign a meaning to tFO-CTLK formulas by using AC-MAS. Given countable interpretation domains U_h s.t. $Con_h \subseteq U_h$, a *(type-consistent) assignment* is a function $\sigma : Var_h \mapsto U_h$. Also, we denote by σ^x_u the assignment s.t. *(i)* $\sigma^x_u(x) = u \in U_h$; and *(ii)* $\sigma^x_u(x') = \sigma(x')$ for every $x' \in Var_h$ different from x. For convenience, we extend assignments to constants so that $\sigma(t) = t$ whenever $t \in Con_h$,

Definition 7 (Satisfaction) *We define whether an AC-MAS $\mathcal{P}$* satisfies *a tFO-CTLK formula φ in a state $s \in \mathcal{S}$ for assignment σ, or $(\mathcal{P}, s, \sigma) \models \varphi$, as follows (the clauses for propositional connectives are straightforward and thus omitted):*

$(\mathcal{P}, s, \sigma) \models R(\vec{t})$	*iff*	$\langle\sigma(t_1), \ldots, \sigma(t_c)\rangle \in D_s(R)$
$(\mathcal{P}, s, \sigma) \models t = t'$	*iff*	$\sigma(t) = \sigma(t')$
$(\mathcal{P}, s, \sigma) \models \forall x\varphi$	*iff*	*for all* $u \in adom_h(s)$, $(\mathcal{P}, s, \sigma^x_u) \models \varphi$
$(\mathcal{P}, s, \sigma) \models AX\varphi$	*iff*	*for all runs* r, *if* $r(0) = s$ *then* $(\mathcal{P}, r(1), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models A\varphi U\varphi'$	*iff*	*for all runs* r, *if* $r(0) = s$ *then there is* $k \geq 0$ *s.t.* $(\mathcal{P}, r(k), \sigma) \models \varphi'$, *and for all* j, $0 \leq j < k$ *implies* $(\mathcal{P}, r(j), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models E\varphi U\varphi'$	*iff*	*for some run* r, $r(0) = s$ *and there is* $k \geq 0$ *s.t.* $(\mathcal{P}, r(k), \sigma) \models \varphi'$, *and for all* j, $0 \leq j < k$ *implies* $(\mathcal{P}, r(j), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models K_i\varphi$	*iff*	*for all* s', $s \sim_i s'$ *implies* $(\mathcal{P}, s', \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models C\varphi$	*iff*	*for all* s', $s \sim s'$ *implies* $(\mathcal{P}, s', \sigma) \models \varphi$

where $\sim$ is the transitive closure of $\bigcup_{A_i \in Ag} \sim_i$.

A formula φ is *true* in s, or $(\mathcal{P}, s) \models \varphi$, if $(\mathcal{P}, s, \sigma) \models \varphi$ for all σ; while φ is *true* in $\mathcal{P}$, or $\mathcal{P} \models \varphi$, if $(\mathcal{P}, s_0) \models \varphi$.

Notice that we adopt an *active-domain* semantics, that is, in each state s quantified variables range only over the active domain of s, which is finite. Nonetheless, by the unconstrained alternation of free variables and modal operators, we can refer to these "active" individuals in successive states, where they might no longer be active.

The key concern of this paper is to investigate the model checking problem for AC-MAS against tFO-CTLK specifications defined as follows.

Definition 8 (Model Checking Problem) *Model checking an AC-MAS $\mathcal{P}$ against a tFO-CTLK formula φ amounts to finding an assignment σ_0 such that $(\mathcal{P}, s_0, \sigma_0) \models \varphi$.*

If all U_h are finite, the model checking problem is decidable, as $\mathcal{P}$ is a finite-state system. However, this is not the case in general, as the following result shows.

Theorem 1 *The model checking problem for AC-MAS w.r.t. tFO-CTLK is undecidable.*

In Section 6 and 7 we develop an abstraction technique to tackle this issue. But first we introduce an auction scenario to illustrate the formal machinery.

5 Auctions as AC-MAS

In this section we apply the formal framework of AC-MAS developed in Section 3 to model the parallel English auctions in Section 2. The relatively small size of the data model in auction AC-MAS will allow us to outline in Section 7 the verification procedure for tFO-CTLK specifications. We consider a single auctioneer A and a finite number of bidders $B_1, \ldots, B_\ell$. The domains of interpretation include a finite set $Items$ of items, as well as the set $\mathbb{Q}$ of rational numbers to represent values for base prices, true values and bids. We use the same names to denote interpretation domains and types. We start by defining the auctioneer as an agent according to Def. 3.

Definition 9 (Auctioneer) *The auctioneer $A = \langle\mathcal{D}_A, Act_A, Pr_A\rangle$ is defined as*

- $\mathcal{D}_A = \{Base/2, \{Bid_i/2\}_{i\leq\ell}, Status/2, </2\}$ *where $Base(it, bp)$ represents the base price $bp \in \mathbb{Q}$ for item $it \in Items$, each $Bid_i(it, bd)$ represents the bid $bd \in \mathbb{Q}$ of bidder B_i for item it, $Status(it, st)$ keeps track of the status of items; status st has two possible values:* active *if item it is actively traded, or* term *if the bidding phase for it has terminated. Finally, $<$ is the standard "strictly less" symbol on $\mathbb{Q}$.*
- $Act_A = \{init_A(it, bp), time\_out(it), skip_A\}$.
- $init_A(it, bp) \in Pr_A(D)$ *if item it does not appear in any tuple in $D(Status)$; $time\_out(it) \in Pr_A(D)$ if $(it,$ active$) \in D(Status)$; while the action $skip_A$ is always enabled.*

Intuitively, the auctioneer non-deterministically chooses to put some item it up for auctioning by performing action $init_A(it, bp)$. The base price bp is then registered in $Base$. She keeps track of bidder B_i's offers in Bid_i and non-deterministically stops the bidding phase for a specific item it by action $time\_out(it)$. At that point, the item is withdrawn and can no longer be put on sale.

Further, each bidder B_i can be represented as the following agent.

Definition 10 (Bidder) *Each bidder $B_i = \langle\mathcal{D}_i, Act_i, Pr_i\rangle$ is defined as*

- $\mathcal{D}_i = \{TValue_i/2, Base/2, \{Bid_i/2\}_{i\leq\ell}, Status/2, </2\}$ *where $TValue_i(it, tv)$ represents the true value $tv \in \mathbb{Q}$ of item it for bidder B_i, while Base, Bid_i, Status and $<$ are defined as for the auctioneer.*
- $Act_i = \{init_i(it, tv), bid_i(it, bd), skip_i\}$.
- $init_i(it, tv) \in Pr_i(D)$ *if $(it,$ active$) \in D(Status)$ and item it does not appear in $D(TValue_i)$; $bid_i(it, bd) \in Pr_i(D)$ whenever item it appears in $D(TValue_i)$, the highest bid bd_j in some Bid_j $(j \neq i)$ for item it is strictly less than the true value tv for bidder B_i, $(it,$ active$) \in D(Status)$, and $bd_j < bd \leq tv$. The action $skip_i$ is always enabled.*

By Def. 10 it is apparent that each bidder can bid only for actively traded items, whenever bids have not exceeded her true value. Notice that symbols $Base$, Bid_i, $Status$ and $<$ are shared by all agents. However, each relation can be modified by at most one agent ($Base$ and $Status$ by the auctioneer; Bid_i by bidder B_i). Hence, the consistency of db instances is preserved. Also, the information contained in $TValue_i$ is private to each agent B_i.

We can now define English auctions as AC-MAS.

Definition 11 (Auction AC-MAS) *Given the set $Ag = \{A, B_1, \ldots, B_\ell\}$ of agents on sets Items, $\mathbb{Q}$, and $\{$active, terms$\}$, the* auction AC-MAS *is a tuple $\mathcal{A} = \langle Ag, s_0, \tau\rangle$ where*

- $s_0 = \langle D_A, D_1, \ldots, D_\ell \rangle$ *is the global state where for all* $j \in \{A, 1, \ldots, \ell\}$, $D_j(<)$ *is the "strictly less" relation on* $\mathbb{Q}$, *while all other relations are empty;*
- τ *is the global transition function s.t.* $s \xrightarrow{\alpha(\vec{u})} s'$ *iff*
  - $\alpha_A = init_A(it, bp)$ *and* s' *modifies* s *by adding tuples* (it, bp) *and* $(it, \textsf{active})$ *to relations* $D'_A(Base)$ *and* $D'_A(Status)$ *resp.;*
  - $\alpha_i = init_i(it, tv)$ *and* s' *modifies* s *by adding tuple* (it, tv) *to relation* $D'_i(TValue_i)$ *for bidder* B_i*;*
  - $\alpha_i = bid_i(it, bd')$ *and* s' *modifies* s *by replacing any tuple* (it, bd) *in* $D_j(Bid_i)$ *with* (it, bd')*, for* $j \in \{1, \ldots, \ell\}$*;*
  - $\alpha_A = time\_out(it)$ *and* $(it, \textsf{active}) \notin D'_j(Status)$ *and* $(it, \textsf{term}) \in D'_j(Status)$*, for* $j \in \{A, 1, \ldots, \ell\}$*;*
  - $\alpha_A = skip_A$ *or* $\alpha_i = skip_i$ *for some* $i \leq \ell$*, and* $D'_i = D_i$*.*

Notice that the auction AC-MAS $\mathcal{A}$ in Def. 11 respects the intuitions on the progress of an auction for multiple items in parallel. Since bidders can bid any value in $\mathbb{Q}$ (up to tv), there can be an infinite number of bids in principle, so the AC-MAS $\mathcal{A}$ is really an infinite-state system. Of course, in our presentation we made a number of conceptual abstractions. However, we maintain that the present formalisation satisfies an idealised notion of parallel English auction, as it has been successfully analysed in game theory [14]. Finally, notice that the interpretation of symbol $<$ is rigid. Since $<$ is the only symbol with an infinite interpretation, the auction AC-MAS $\mathcal{A}$ is indeed *rigid*.

While $\mathcal{A}$ intuitively fulfils the informal description of an auction, we need to develop formal verification methods to check this fact. Hence, we turn to considering properties of interest that can be expressed in the specification language tFO-CTLK. First, one feature of the auction we might want to check is that for each item $it \in Items$ there is exactly one base price bp registered in the relation $Base$, while bidders associate *at most* one true value tv to each item it (possibly none). This can be expressed as

$$AG\, \forall it (\exists! bp\ Base(it, bs) \land \exists^{\leq 1} tv\ TValue_i(it, tv))$$

where the quantifiers $\exists!$ and $\exists^{\leq 1}$ are defined as standard in first-order logic with identity.

In tFO-CTLK we can also express what agents know about the information content of the auction. For instance, specification *(i)* in Section 2 requires that the base prices of items remain common knowledge throughout the auction:

$$AG\, \forall it\, \exists bp\ C\ Base(it, bp)$$

On the contrary, according to *(ii)* the true value of items for each bidder B_i is secret to all other bidders and to the auctioneer:

$$AG\, \forall it\, \neg \exists tv\ \bigvee_{j \neq i \lor j = A} K_j\ TValue_i(it, tv)$$

Further, we can express properties on the progress of the auctioning process. As an example, for each bidder B_i each bid is less or equal to her true value:

$$AG\, \forall it, bd, tv((Bid_i(it, bd) \land TValue_i(it, tv)) \rightarrow bd \leq tv)$$

Also, specification *(iii)* states that each bidder B_i can raise her bid unless she has already hit her true value:

$$AG\ \forall it, bd(Bid_i(it, bd) \rightarrow$$
$$\rightarrow (TValue_i(it, bd) \lor EF\, \exists bd'(bd' > bd \land Bid_i(it, bd'))))$$

We refer to [8] for further examples of specifications in tFO-CTLK.

In the next sections we develop the theory that will allow us to model check specifications as above on a particular class of artifact-centric multi-agent systems that includes the auction AC-MAS $\mathcal{A}$.

6 Bisimulation

In this section we introduce a notion of bisimulation for AC-MAS. Similar notions have already appeared in the literature [6, 7]. However, in this paper we consider typed languages and, most importantly, relations $Q \in \mathcal{D}$ with a possibly infinite interpretation. This extended framework has an impact notably on the key concept of *uniformity*. In the rest of the section we let $\mathcal{P} = \langle Ag, s_0, \tau \rangle$ and $\mathcal{P}' = \langle Ag', s'_0, \tau' \rangle$ be AC-MAS and assume that $s = \langle D_0, \ldots, D_\ell \rangle \in \mathcal{S}$ and $s' = \langle D'_0, \ldots, D'_\ell \rangle \in \mathcal{S}'$. Also, each $\mathcal{C}_h$ is a finite set of constants of type T_h.

Definition 12 (Isomorphism) *The db instances* $D, D' \in \mathcal{D}(\vec{U})$ *are* isomorphic*, or* $D \simeq D'$*, iff there exists a type-consistent bijection* $\iota : adom_h(D) \cup \mathcal{C}_h \mapsto adom_h(D') \cup \mathcal{C}_h$ *s.t.* (i) *ι is the identity on each $\mathcal{C}_h$; and* (ii) *for every* $R \in \mathcal{D}$ *and* $\vec{u} \in (Dom(\iota))^c$, $\vec{u} \in D(R)$ *iff* $\iota(\vec{u}) \in D'(R)$*. When this is the case,* ι *is a* witness *for* $D \simeq D'$*.*

The global states s and s' are isomorphic*, or* $s \simeq s'$*, iff there exists a type-consistent bijection* $\iota : adom_h(s) \cup \mathcal{C}_h \mapsto adom_h(s') \cup \mathcal{C}_h$ *s.t. for every* $A_i \in Ag$*,* ι *is a witness for* $D_i \simeq D'_i$*. Any function* ι *as above is a* witness *for* $s \simeq s'$*.*

Isomorphisms preserve the interpretation of individual constants as well as of relation symbols $P \in \mathcal{D}$. As to symbols $Q \in \mathcal{D}$, the witness ι preserves the interpretation only for individuals in the active domain. Clearly, $\simeq$ is an equivalence relation.

Observe that isomorphisms are such w.r.t. specific sets $\mathcal{C}_h$ of constants. Hereafter we assume the various $\mathcal{C}_h$ to be fixed. While isomorphic states share a common relational structure, they do not necessarily satisfy the same first-order formulas, as satisfaction depends also on values assigned to free variables. To account for this, we have to recast the notion of *equivalent assignments* in [6].

Definition 13 (Equivalent assignments) *Given isomorphic states s, s' and sets of variables* $V_h \subseteq Var_h$ *for each type* T_h*, the assignments* $\sigma : Var_h \mapsto U_h$ *and* $\sigma' : Var_h \mapsto U'_h$ *are* equivalent for all V_h w.r.t. s and s' *iff there exists a bijection* $\gamma : adom_h(s) \cup \mathcal{C}_h \cup \sigma(V_h) \mapsto adom_h(s') \cup \mathcal{C}_h \cup \sigma'(V_h)$ *s.t.* (i) *the restriction* $\gamma|_{adom_h(s) \cup \mathcal{C}_h}$ *is a witness for* $s \simeq s'$*;* (ii) $\sigma'|_{V_h} = \gamma \cdot \sigma|_{V_h}$*; and* (iii) *for every* $\vec{u} \in (Dom(\gamma))^b$ *and* $A_i \in Ag$*,* $\vec{u} \in D_i(Q)$ *iff* $\gamma(\vec{u}) \in D'_i(Q)$*.*

Intuitively, equivalent assignments preserve the (in)equalities of the variables in each V_h as well as the interpretation of symbols $Q \in \mathcal{D}$. Two assignments are said to be *equivalent for a tFO-CTLK formula* φ if they are equivalent for all $fr_h(\varphi)$.

Plain bisimulations are known to preserve satisfaction in a propositional modal setting [10]. We now investigate the conditions under which this applies to AC-MAS as well, beginning with a notion of simulation. Throughout the rest of the paper we assume w.l.o.g. that $con_h(\varphi) \subseteq \mathcal{C}_h$ for every type T_h.

Definition 14 (Simulation) *A relation* S *on* $\mathcal{S} \times \mathcal{S}'$ *is a* simulation *if* $\langle s, s' \rangle \in S$ *implies: 1.* $s \simeq s'$*;*

2. *for* $t \in \mathcal{S}$*, if* $s \rightarrow t$ *then there is* $t' \in \mathcal{S}'$ *s.t.* $s' \rightarrow t'$*,* $s \oplus t \simeq s' \oplus t'$ *and* $\langle t, t' \rangle \in S$*;*
3. *for* $A_i \in Ag$*,* $t \in \mathcal{S}$*, if* $s \sim_i t$ *then there is* $t' \in \mathcal{S}'$ *s.t.* $t \sim_i t'$*,* $s \oplus t \simeq s' \oplus t'$ *and* $\langle t, t' \rangle \in S$*.*

Two states s and s' are *similar* iff $\langle s, s' \rangle \in S$ for some simulation S. Simulations can naturally be extended to bisimulations.

Definition 15 (Bisimulation) *A relation* B *on* $\mathcal{S} \times \mathcal{S}'$ *is a* bisimulation *iff both* B *and* $B^{-1} = \{\langle s', s \rangle \mid \langle s, s' \rangle \in B\}$ *are simulations.*

Two states s and s' are *bisimilar* iff $\langle s, s' \rangle \in B$ for some bisimulation B. Also, $\mathcal{P}$ and $\mathcal{P}'$ are *bisimilar*, or $\mathcal{P} \approx \mathcal{P}'$, iff so are their initial states s_0 and s'_0. By Lemma 2 in [8] bisimilar, hence isomorphic, states preserve (typed) first-order formulas. However, this is no longer the case for the full tFO-CTLK language. We refer to [4] for an example of this fact. To overcome this difficulty we introduce a novel notion of *uniformity*.

Definition 16 (Uniformity) *An AC-MAS $\mathcal{P}$ is* uniform *iff for every $s, t, s' \in \mathcal{S}$, $t' \in \mathcal{D}(\vec{U})$,*

1. *if $s \xrightarrow{\alpha(\vec{u})} t$ and $s \oplus t \simeq s' \oplus t'$ for some witness ι, then for every type-consistent constant-preserving extension ι' of ι to $\vec{u}$, we have that $s' \xrightarrow{\alpha(\iota'(\vec{u}))} t'$;*
2. *if $s \sim_i t$ and $s \oplus t \simeq s' \oplus t'$, then $s' \sim_i t'$.*

Further, if $\mathcal{P}$ is rigid, then (i) *the set $\mathcal{D}(\vec{U})$ above is restricted to db instances t' agreeing on the interpretation $\mathcal{P}(Q)$ of symbols $Q \in \mathcal{D}$;* (ii) *for all $\vec{u} \in U^*$, there exist $\vec{v}$, $\vec{v}'$ in U^* s.t. $(\vec{v}, \vec{u}) \in \mathcal{P}(Q)$ and $(\vec{u}, \vec{v}') \in \mathcal{P}(Q)$; and* (iii) *for all $\vec{u} \in \mathcal{P}(Q)$, for all $i < b - 1$, there exists v s.t. $(u_0, \ldots, u_i, v, u_{i+1}, \ldots, u_{b-2}) \in \mathcal{P}(Q)$ and $(u_1, \ldots, u_i, v, u_{i+1}, \ldots, u_{b-1}) \in \mathcal{P}(Q)$ (with an abuse of notation we assume that for $u_{i+1} = u_{b-1}$ or $u_i = u_0$ the tuple ends or begins with v).*

Intuitively, conditions (1) and (2) in Def. 16 say that if state t is reached by executing the ground action $\alpha(\vec{u})$ in s, and v is uniformly replaced with v' in s, $\vec{u}$ and t, thus obtaining, say, s', $\vec{u}'$ and t', then t' can be reached by executing $\alpha(\vec{u}')$ in s'. Further, the condition on rigid AC-MAS is aimed at obtaining the same uniform transitions while keeping fixed the interpretation of symbols $Q \in \mathcal{D}$.

In particular, we have the following result.

Proposition 17 *The auction AC-MAS $\mathcal{A}$ is indeed uniform.*

As a result, the auction AC-MAS $\mathcal{A}$ is both rigid and uniform.

We now state the main contribution of this section, which lifts the result in [6] to AC-MAS with types and predicates with an infinite interpretation. Hereafter $\sup_{s \in \mathcal{S}} \{|adom_h(s)|\} = \infty$ whenever an AC-MAS $\mathcal{P}$ is unbounded, i.e., there are no $b_h \in \mathbb{N}$ s.t. $|adom_h(s)| \leq b_h$ for all $s \in \mathcal{S}$.

Theorem 2 *Consider bisimilar and uniform AC-MAS $\mathcal{P}$ and $\mathcal{P}'$, and a tFO-CTLK formula φ. If for every T_h,*

1. *$|U'_h| \geq 2 \sup_{s \in \mathcal{S}} \{|adom_h(s)|\} + |\mathcal{C}_h| + |var_h(\varphi)|$*
2. *$|U_h| \geq 2 \sup_{s' \in \mathcal{S}'} \{|adom_h(s')|\} + |\mathcal{C}_h| + |var_h(\varphi)|$*

then $\mathcal{P} \models \varphi$ iff $\mathcal{P}' \models \varphi$

A proof of Theorem 2 is provided in [8]. Intuitively, if each AC-MAS has enough elements to simulate the transitions in the other system, then they satisfy the same formulas. By this result, if in addition each $\{|adom_h(s)| \mid s \in \mathcal{S}\}$ is bounded, and therefore all $\sup_{s \in \mathcal{S}} \{|adom_h(s)|\}$ are finite, then an infinite and uniform AC-MAS $\mathcal{P}$ can in principle be verified by model checking a finite bisimilar system $\mathcal{P}'$, whose interpretation domains satisfy condition (1) in Theorem 2. In the next section we introduce a class of infinite and uniform AC-MAS that admits finite abstractions.

7 Finite Abstraction

In this section we state sufficient conditions to reduce the model checking problem for an infinite AC-MAS to the verification of a finite system. The main result is given as Theorem 3, which guarantees that for bounded and rigid AC-MAS uniformity is sufficient to obtain bisimilar finite abstractions that preserve tFO-CTLK formulas. In the following we assume for technical reasons and w.l.o.g. that any AC-MAS $\mathcal{P}$ is such that $adom_h(s_0) \subseteq \mathcal{C}_h$ (as each $adom_h(s_0)$ is finite). Also, $N_h = \sum_{A_i \in Ag} \max_{\{\alpha(\vec{x}) \in Act_i, \vec{x} \in Var_h\}} \{|\vec{x}|\}$.

Definition 18 (Bounded AC-MAS) *An AC-MAS $\mathcal{P}$ is b_h-*bounded*, for $b_h \in \mathbb{N}$, iff for all $s \in \mathcal{S}$, $|adom_h(s)| \leq b_h$.*

Thus, an AC-MAS is b_h-bounded if no active domain of its reachable state space contains more than b_h distinct elements of type T_h. An AC-MAS $\mathcal{P}$ is *bounded* if for every type T_h, $\mathcal{P}$ is b_h-bounded for some $b_h \in \mathbb{N}$. Observe that bounded AC-MAS may still contain infinitely many states. So, bounded AC-MAS are infinite-state systems in general, whose model checking problem cannot be tackled by standard techniques for finite-state systems.

We now introduce abstractions in a modular manner by first defining abstract agents.

Definition 19 (Abstract agent) *Let $A = \langle \mathcal{D}, Act, Pr \rangle$ be an agent defined on a countable interpretation domain U_h for each type T_h. Given a countable set U'_h of individuals for each T_h, the* abstract agent *A' is a tuple $\langle \mathcal{D}', Act', Pr' \rangle$ on $U'_1, \ldots, U'_k$ s.t.* (i) *$\mathcal{D}' = \mathcal{D}$;* (ii) *$Act' = Act$; and* (iii) *$Pr'$ is the smallest function s.t. if $\alpha(\vec{u}) \in Pr(D)$, $D' \in \mathcal{D}'(\vec{U}')$ and $D' \simeq D$ for some witness ι, then $\alpha(\vec{u'}) \in Pr'(D')$, where $\vec{u'} = \iota'(\vec{u})$ for some type-consistent constant-preserving bijection ι' extending ι to $\vec{u}$.*

Given a set Ag of agents, let Ag' be the set of the corresponding abstract agents.

We remark that A', as defined in Def. 19, is indeed an agent according to Def. 3. We now present the notion of abstraction.

Definition 20 (Abstraction) *Let $\mathcal{P} = \langle Ag, s_0, \tau \rangle$ be an AC-MAS, and Ag' the set of abstract agents as in Def. 19. The AC-MAS $\mathcal{P}' = \langle Ag', s'_0, \tau' \rangle$ is an* abstraction *of $\mathcal{P}$ iff* (i) *$s'_0 \simeq s_0$, and* (ii) *τ' is the smallest function s.t. if $s \xrightarrow{\alpha(\vec{u})} t$, $s', t' \in \mathcal{D}'(\vec{U})$ and $s \oplus t \simeq s' \oplus t'$ for some witness ι, then $s' \xrightarrow{\alpha(\iota'(\vec{u}))} t'$ for some type-consistent constant-preserving bijection ι' extending ι to $\vec{u}$.*

Notice that $\mathcal{P}'$ is indeed an AC-MAS as it satisfies the relevant conditions on protocols and transitions in Def. 4. Also, by varying each U'_h we can obtain different abstractions. Moreover, the abstraction of a rigid AC-MAS is not itself rigid in general. The last point is key in the definition of finite abstractions.

We immediately state the main technical result of the paper, while referring to [8] for details and full proofs.

Theorem 3 *Consider a bounded, uniform and rigid AC-MAS $\mathcal{P}$ over infinite interpretation domains U_h, a tFO-CTLK formula φ, and interpretation domains U'_h s.t. $\mathcal{C}_h \subseteq U'_h$. If for every type T_h, $|U'_h| \geq 2b_h + |\mathcal{C}_h| + \max\{|var_h(\varphi)|, N_h\}$, then there exists an abstraction $\mathcal{P}'$ of $\mathcal{P}$ over $U'_1, \ldots, U'_k$ s.t. $\mathcal{P} \models \varphi$ iff $\mathcal{P}' \models \varphi$*

We remark that the U'_h in Theorem 3 might as well be finite. So, by using a sufficient number of abstract values in U'_h, we can in principle reduce the model checking problem for infinite-state AC-MAS to the verification of a finite abstraction.

Corollary 4 *Given a bounded, uniform and rigid AC-MAS $\mathcal{P}$ over infinite domains $\vec{U}$, and a tFO-CTLK formula φ, there exists a finite abstract AC-MAS $\mathcal{P}'$ s.t. φ is satisfied by $\mathcal{P}$ iff $\mathcal{P}'$ satisfies φ.*

Notice that the assumption of rigidity is essential to obtain finite abstractions. To conclude this section we briefly outline how to derive a finite abstraction of the auction AC-MAS $\mathcal{A}$ in Section 5.

7.1 Abstract Auction

We observe that the auction AC-MAS $\mathcal{A}$ is indeed bounded, uniform and rigid. We showed above that $\mathcal{A}$ is uniform and rigid. As to boundedness, notice that the only infinite interpretation domain in $\mathcal{A}$ is the set $\mathbb{Q}$ of rational numbers. By definition of $\mathcal{A}$, for each global state s, there can be at most $|Items|(2|Ag|-1)$ distinct rational numbers in the active domain of s: $|Items|$ elements to represent base prices, $|Items|(|Ag|-1)$ elements to represent true values, and $|Items|(|Ag|-1)$ elements for bids. Further, consider the specifications appearing in Section 5 to be verified. No constant appears in these formulas and the active domain of the initial state s_0 is empty, therefore so is the set $\mathcal{C}_{\mathbb{Q}}$ of constants for rational numbers. Finally, 13 variables of type $\mathbb{Q}$ appear in our specifications, and this number exceeds $N_{\mathbb{Q}}$. As a consequence, we consider a finite abstract domain $U'_{\mathbb{Q}}$ of cardinality greater of equal to $2|Items|(2|Ag|-1)+13$, as required in Theorem 3.

We now describe briefly the abstract agents A' and $B'_1, \ldots, B'_\ell$ for the concrete auctioneer A and bidders $B_1, \ldots, B_\ell$. By Def. 19 the abstract bd schema $\mathcal{D}'$ and action types in Act' are the same as $\mathcal{D}$ and Act. As to the protocol functions, now these take values not in $\mathbb{Q}$ but $U'_{\mathbb{Q}}$. As an example, consider the clause for action $bid_i(it, bd)$ in Def. 10: $bid_i(it, bd) \in Pr_i(D)$ whenever the item it appears in $D(TValue_i)$, the highest bid bd_j in some Bid_j $(j \neq i)$ for item it is strictly less than the true value tv for bidder B_i, $bd_j < bd \leq tv$, and $(it, \textsf{active}) \in D(Status)$. Now, the condition on protocols in Def. 19 requires that for $D' \in \mathcal{D}'(\vec{U})$, $bid_i(it, bd') \in Pr'_i(D'_i)$ whenever $D' \simeq D$ for some witness ι. In particular, this means that $bd' \in U'_{\mathbb{Q}}$ is an abstract value that has not yet been used to represent any bid in D'. By assumption $|U'_{\mathbb{Q}}| \geq 2|Items|(2|Ag|-1)+13$ on the cardinality of $U'_{\mathbb{Q}}$ in Theorem 3 it is always possible to find such an element.

Finally, given the set $Ag' = \{A', B'_1, \ldots, B'_\ell\}$ of abstract agents on $Items$, $\{\textsf{active}, \textsf{term}\}$ and $U'_{\mathbb{Q}}$, we briefly illustrate the abstract auction AC-MAS $\mathcal{A}' = \langle Ag', s'_0, \tau' \rangle$ where

- $s'_0 = s_0|_{adom(s_0)}$ is the initial state;
- τ' is the global transition function that mimicks τ. For instance, if $\alpha_i = bid_i(it, bd')$, then $s' \xrightarrow{\alpha(\vec{u})} t'$ whenever t' is the db instance that modifies s' by replacing any pair (it, bd) in $D'_j(Bid_i)$ with (it, bd'), where $bd' \in U'_{\mathbb{Q}}$ has been found as detailed above.

Moreover, by Def. 20 and the definition of isomorphism, we have that bd' is strictly greater than the highest bid bd_j in some Bid_j in t' for item it, but less than the true value tv for bidder B_i. This information defines the interpretation $D_{t'}(<)$ of symbol $<$ in t'. Indeed, the interpretation of $<$ in $\mathcal{A}'$ is not rigid, thus allowing for the reuse of values. In our example, the old value bd would no longer appear in the active domain of t'. Hence it might be used again in the next transition if needed.

Reasoning as above we can generate the whole state space $\mathcal{S}'$ of the abstract auction AC-MAS $\mathcal{A}'$, as $Items$, $\{\textsf{active}, \textsf{term}\}$ and $U'_{\mathbb{Q}}$ are finite. Then, we can model check our tFO-CTLK formulas on this finite abstraction. Indeed, both the concrete AC-MAS $\mathcal{A}$ and the interpretation domain $U'_{\mathbb{Q}}$ satisfy the hypotheses of Theorem 3. Thus, by this result it is guarantee that the specifications are satisfied in the abstraction $\mathcal{A}'$ iff they are satisfied in the concrete AC-MAS $\mathcal{A}$.

8 Conclusions and Future Work

In this paper we advanced the state-of-the-art on the verification of auctions by model checking. First, we extended the framework of AC-MAS [6, 7] to support both typed languages and relations with an infinite interpretation. We argued that both features are essential to formalise parallel English (ascending bid) auctions as AC-MAS. Most importantly, we provided a novel abstraction technique within this enhanced setting. As a result, we are now able to model check a significant class of infinite-state AC-MAS, including parallel English auctions, against sophisticated specifications in tFO-CTLK.

In future work we aim at applying the theoretical results above to concrete use cases. Indeed, one relevant issue concerns the boundedness check for AC-MAS. In this respect, it would be of interest to find sufficient conditions ensuring boundedness, similarly to those discussed in [20]. Further, general constructive techniques to build abstractions from concrete AC-MAS are also essential for deployment in business processes.

REFERENCES

[1] S. Abiteboul et al. *Foundations of Databases*, Addison-Wesley, 1995.

[2] T. Ågotnes et al. 'Verifiable equilibria in boolean games', In Rossi [24].

[3] A. Badica and C. Badica, 'Specification and verification of an agent-based auction service', in *Information Systems Development*, 239–248, (2010).

[4] F. Belardinelli and A. Lomuscio, 'Decidability of model checking non-uniform artifact-centric quantified interpreted systems', In Rossi [24].

[5] F. Belardinelli et al. 'Verification of Deployed Artifact Systems via Data Abstraction', in *Proc. of ICSOC'11*, pp. 142–156, (2011).

[6] F. Belardinelli et al. 'An Abstraction Technique for the Verification of Artifact-Centric Systems', in *Proc. of KR'12*, pp. 319 – 328, (2012).

[7] F. Belardinelli et al. 'Verification of GSM-Based Artifact-Centric Systems through Finite Abstraction', in *Proc. of ICSOC'12*, pp. 17–31, (2012).

[8] F. Belardinelli, 'Model checking auctions as artifact systems via finite abstraction', Technical report, Université d'Evry, (2014).

[9] K. Bhattacharya et al. 'Towards Formal Analysis of Artifact-Centric Business Process Models', in *Proc. of BPM'07*, pp. 288–304, (2007).

[10] P. Blackburn et al. *Modal Logic*, Cambridge University Press, 2001.

[11] E. Damaggio et al. 'Artifact Systems with Data Dependencies and Arithmetic', *ACM Transactions on Database Systems*, **37**(3), 22:1–22:36, (2012).

[12] G. De Giacomo et al. 'Bounded Situation Calculus Action Theories and Decidable Verification', in *Proc. of KR'12*, pp. 467–477, (2012).

[13] A. Deutsch et al. 'Automatic Verification of Data-centric Business Processes', in *Proc. of ICDT'09*, pp. 252–267, (2009).

[14] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010.

[15] R. Fagin et al. *Reasoning About Knowledge*, The MIT Press, 1995.

[16] D. Fischer et al. 'Model Checking Games for the Quantitative mu-Calculus', *Theory Comput. Syst.*, **47**(3), 696–719, (2010).

[17] C. Gerede, J. Su, 'Specification and Verification of Artifact Behaviors in Business Process Models', in *Proc. of ICSOC'07*, pp. 181–192, (2007).

[18] E. Tadjouddine et al. 'Abstractions for model-checking game-theoretic properties of auctions', in *Proc. of AAMAS '08*, pp. 1613–1616, (2008).

[19] N. Haque et al. 'Resource allocation in communication networks using market-based agents', *Knowledge-Based Systems*, **18**(4-5), 163–170, (2005).

[20] B. Hariri et al. 'Verification of relational data-centric dynamic systems with external services', in *Proc. of PODS*, pp. 163–174. ACM, (2013).

[21] R. Hull, 'Artifact-Centric Business Process Models', in *Proc. of OTM'08*, pp. 1152–1163, (2008).

[22] A. Lomuscio et al. 'MCMAS: A Model Checker for the Verification of Multi-Agent Systems', in *Proc. of CAV'09*, pp. 682–688, (2009).

[23] D. Reeves et al. 'Exploring bidding strategies for market-based scheduling', *Decision Support Systems*, **39**(1), 67–85, (2005).

[24] F. Rossi, ed. *Proc. of IJCAI 2013, Beijing, August 3-9, 2013*.

[25] N. Troquard et al. 'Model checking strategic equilibria', in *Proc. of MoChArt*, pp. 166–188, (2008).

[26] H. Xu et al. 'Real-time model checking for shill detection in live online auctions', in *Software Engineering Research and Practice*, pp. 134–140, (2009).