A Survey of Mobile Boundary Process Calculus Theory and Applications: Postprint

Authors: Lin Rongde, Jiang Hua, Huang Jianxin

Date: 2019-01-28T00:00:00+00:00

Abstract

Mobile Ambient Calculus employs the core concept of "ambient" to express computational places with boundaries, and provides capabilities such as ambient mobility, authentication, and authorization to characterize the essence of mobile computing from the most fundamental level, thereby establishing itself as an important research branch in the formal theory and application of mobile computing systems. This paper provides a comprehensive overview of the theoretical and applied research and development of Mobile Ambient Calculus, systematically compiles and analyzes methodologies for extended semantics and algebraic properties, spatial logic and model checking algorithms, as well as the current application status of mobile ambients in computational system modeling, and prospects future research directions in this domain.

Full Text

Preamble

Vol. 37 No. 4
Application Research of Computers
ChinaXiv Cooperative Journal

Overview on Theory and Applications of Ambients Calculus

Lin Rongde[1], Jiang Hua[2]†, Huang Jianxing[1]
(1. Fujian Province University Key Laboratory of Computational Science, School of Mathematical Science, Huaqiao University, Quanzhou, Fujian 362000, China;
2. Key Laboratory of Granular Computing, Minnan Normal University, Zhangzhou, Fujian 363000, China)

Abstract: Ambient Calculus introduces the central notion of "ambients" to describe computation areas with boundaries, and provides capabilities for mobility,

authentication, and authorization to capture the fundamental characteristics of mobile computation systems, making it an important research area in formal-modeling theory and applications of mobile computation systems. This paper reviews the main theoretical and applied research works and their development in ambient-based calculus, analyzes extended semantics and algebra behavior analyzing methods, spatial logics and model checking algorithms, and applications of formal-modeling in computation systems, and discusses future research directions in these areas.

Key words: ambient calculus; bisimulation congruences; context barb congruences; ambient logic; model checking

0 Introduction

Mobile Ambients (MA) [1] is a process calculus model for mobile computation proposed by Cardelli and Gordon. Inspired by the -calculus, it introduces the special notion of "ambients"to characterize computation spaces with boundaries, such as Web pages, components, and mobile devices. Ambients are composed in a tree-like structure, with each ambient containing communication processes, mobile processes, and child ambients. An ambient can move as a whole under the control of mobile processes. Through the concepts of ambients, mobility, and mobility-related authorization, MA calculus can capture the essential features of distributed and mobile computation at a fundamental level.

The syntax of MA calculus is defined as follows:

In the above definition, M represents capabilities: it can be empty (ϵ), a name variable, a name, the capability to enter an ambient (in n), the capability to exit an ambient (out n), the capability to open an ambient (open n), or a sequential composition of capabilities ($M.M'$). P represents processes: 0 denotes the null process; $M.P$ represents executing capability M and then continuing as process P; $n[P]$ denotes an ambient named n with internal process P; $P|Q$ represents the parallel composition of processes P and Q; $(x).P$ is an asynchronous output action that releases capability x and then continues as process P; $(x)P$ represents inputting a capability and binding it to the formal parameter in process P; $(\nu n)P$ restricts name n for use within process P; and $!P$ denotes the infinite replication of process P.

The semantics of mobile ambients calculus consists of two components: structural congruence relation ($\equiv$) and reduction relation ($\rightarrow$), defined by the rules in Table 1 and Table 2 . Structural congruence represents processes that are syntactically different but semantically equivalent, while reduction relation represents possible execution results of processes.

In Table 2 , the reduction rule (Red In) indicates that the enter capability in n can lead its enclosing ambient m to migrate into another sibling ambient n and become a child ambient. The rule (Red Out) shows that the exit capability out n

can lead its enclosing ambient m to leave its parent ambient n and become a sibling. The rule (Red Open) demonstrates that the open capability open n can remove the boundary of ambient n and expose its internal process P. The rule (Red Comm) represents local communication within an ambient: the output capability process $\langle M \rangle$ releases capability M, which is captured by the input action (x) and bound to the formal parameter x in process P. Rules (Red Amb), (Red Par), and (Red Res) indicate that reductions can occur inside ambients, in parallel environments, and within private name spaces, respectively. Rule (Red $\equiv$) shows that reduction results are consistent for processes satisfying structural congruence relations.

From the above syntax and semantics definitions of MA calculus, we can see that names serve as ambient names rather than channel names as in the -calculus. Each ambient explicitly indicates its position in the entire tree structure of the system, allowing every process expression to induce an abstract tree-like configuration. This directly captures concepts such as locality, parent/child resources, boundaries, and system topology in mobile computation systems. System configurations can dynamically change through the execution of mobility capabilities within ambients, local communication, and dynamic binding of resources. In particular, the dynamic binding of names through process communication enables implicit reconfiguration of mobile computation systems, transparently updating the state of receiving parties. Due to these advantages, MA calculus has become an exemplary model for describing the distribution of processes and resources, mobility between locations, and related security control issues in mobile computation systems.

Over the past decade, research on extended ambient calculus models and applications centered around the "ambient" concept has gradually become an active area in formal modeling theory and applications of mobile computation systems, primarily manifested in: (a) research on extended semantics and algebraic theory of mobile ambients; (b) research on spatial logic and model checking algorithms for mobile ambients; and (c) research on modeling applications of mobile ambients in computation systems.

---

1 Extended Semantics and Algebraic Theory of Mobile Ambients

The ambient concept and its interactive calculus proposed in mobile ambients reflect the essence of mobile computation systems. Numerous research efforts have extended the interactive semantics of mobile ambients from different perspectives and employed various forms of bisimulation, context barb congruence, and other algebraic tools to analyze properties of the calculus models.

1.1 Extensions of Interactive Semantics for Mobile Ambients

In MA calculus, mobility interaction primitives between ambients are unilaterally participated, which causes strong interference issues syntactically [2] that cannot be identified and eliminated through type systems, resulting in relatively weak algebraic theory and difficulty in determining process equivalence. Many researchers have extended MA calculus from different aspects, with representative models including Safe Ambients [2,3], Boxed Ambients [4], and Fair Ambients [5].

1) Safe Ambients Calculus
Levi and Sangiorgi first identified the strong interference problem in MA calculus in their work and proposed Safe Ambients calculus, which retains the basic framework of MA while controlling strong interference. Addressing the unilateral participation of primitive actions in MA, they introduced corresponding co-action primitives for each primitive. For example, reduction rules (Red In) and (Red Open) were modified as shown in equations (2) and (3).

By introducing co-action primitives, the strong interference phenomena in MA can be eliminated through explicit coordination. For instance, in MA, the process in $n.P|n[Q]$ might exhibit two interfering reductions as shown in equation (4). In Safe Ambients calculus, the reduction direction can be explicitly determined through co-actions, eliminating interference as shown in equations (5) and (6).

Extended ambient calculi based on Safe Ambients, such as those with authenticated interaction, enhance the security of ambient mobility operations while strengthening the algebraic properties of these extended models. However, communication between ambients in these extensions still requires opening ambients, which has limitations for certain security operations. Consequently, various scholars proposed extended models with cross-boundary communication mechanisms, primarily represented by Boxed Ambients and Fair Ambients.

2) Boxed Ambients and Fair Ambients
Boxed Ambients calculus is an extension of MA focusing on cross-boundary communication mechanisms. It sets communication directions for input and output actions to clarify communication partners. For example, the rule for ambient n outputting to external processes is shown in equation (7).

(Output $\langle M\rangle^{\uparrow}$)
Equation (7)' s communication mechanism provides fine-grained ambient communication primitives. Since communication directions are explicitly defined, it specifies the receiver of output resources, providing basic operations for authentication and authorization behaviors in mobile computation system communication security. However, Boxed Ambients calculus may lead to unbounded growth in the number of ambients in mobile computation systems due to the removal of the Open primitive.

Fair Ambients calculus (FA calculus) is a relatively complete calculus model

with secure mobility and communication interaction capabilities among ambient-centered extensions. FA calculus proposes three principles for ambient interaction: all interactions are participated by ambients as subjects; participating ambients must authenticate each other' s identities; and participating ambients must be mutually authorized. These principles meet the basic requirements for secure operations in network environments. The main reduction semantics are shown in Table 3 .

In Table 3, rule (R2) means: the co-action out a is contained in parallel ambient Q, so the primary action in m means "needs to exit upon invitation from external ambient n" , while $\overline{\text{in}}m$ means "desires the arrival of a child ambient from n" .

The communication mechanism (R4) also adopts mutual authentication and authorization similar to (R1) to establish cross-ambient communication. With cross-ambient communication mechanisms, the Open primitive in FA calculus no longer needs to assist communication but only completes the merging of two ambients, using mutual authentication and authorization for interaction like (R1).

Compared with other extended calculi, FA calculus possesses the secure mobility mechanisms of Safe Ambients and the cross-boundary secure communication mechanisms of Boxed Ambients. To some extent, FA calculus has become a "fair resource transaction" calculus: exchanging market information through communication interactions, using In and Out primitives to find transaction partners, and finally completing transactions through resource merging via the Open primitive.

3) Other Extended Ambient Calculi

Research extending ambient calculus based on location (region/boundary) and cross-boundary interactions according to different application backgrounds mainly includes context-aware ambients [6], multi-type boundary interaction membrane calculi [7], temporal and probabilistic extensions of interaction behaviors [8,9], and studies on semantics and expressiveness of ambient calculi [10].

Context-aware ambients calculus (CCA) is an extended ambient calculus composed of ambients and their contextual environments, placing ambients and their environments on equal footing: it defines not only interaction semantics between ambients but also explicit interaction semantics between ambients and environments. Subsequent work [11] studied a type system to guarantee confidentiality properties during CCA process execution; work [12] investigated a type system to ensure correctness of communication patterns/process abstractions and context environments in CCA. Meanwhile, work [13] proposed a context-sensitive extended ambient calculus with global communication similar to CCS, and studied its labeled reduction semantics and observational bisimulation equivalence; similar work includes space-aware ambients [14].

Membrane calculi simulate biological cell membrane behavior, primarily represented by Brane calculus [7] and Membranes calculus [15,16]. Membrane

calculi model computational behaviors such as endocytosis, exocytosis, and mitosis occurring on cell membrane surfaces. Subsequent research [17] established hierarchical membrane spaces and constructed richer membrane mobility semantics. Additionally, literature [18] proposed a static analysis method for Brane calculus (membrane calculus) based on abstract interpretation techniques, and studied polynomial-time algorithms for obtaining global hierarchical structure and relationship information constructed by membrane processes; literature [19] proposed a timed membrane calculus, presented various spatiotemporal equivalence properties involved in membrane process mobility during lifetime, and characterized location observational equivalence properties using ambient logic.

Regarding temporal and probabilistic extensions of ambient calculus interaction behaviors, literature [8] proposed probabilistic ambient calculus, which describes that process interaction actions may produce multiple residual processes with associated probabilities after execution, and presented observational probabilistic bisimulation to characterize behavioral equivalence between processes, as well as adding probabilistic temporal formulas to ambient logic formulas. In research on temporal constraints for ambient calculus interaction behaviors, literature [9] proposed that ambients exist as resources with constraints such as duration, capacity, and access radius, along with corresponding timeout handling processes; interaction actions also have duration constraints and timeout handling processes. Furthermore, literature [20] proposed a timed-constrained extended ambient calculus to describe runtime temporal and spatial states of mobile agents; it enables formal description of time-constrained actions, action durations, and spatial requirements, and subsequent work [21] proposed a spatiotemporal context model based on actions for reasoning about time-constrained and space-constrained actions.

1.2 Behavioral Equivalence Theory Methods for Ambient Calculus

In process calculi based on algebraic semantics (such as CSP, -calculus, etc.), various equivalence relations between processes are important means for analyzing process behaviors. The structural congruence relation between processes in mobile ambients only expresses static and finest-grained process equivalence, while higher abstraction levels are sometimes needed to discuss behavioral equivalence. For example, in MA calculus, $(\nu n)n[P]$ represents an ambient named n with internal process P. If name n does not appear in internal process P, then according to reduction rules in Table 2, it cannot interact with any other process: it cannot exit from ambient n, nor can other processes be aware of its existence and interact with it. Therefore, it can be considered behaviorally equivalent to the null process 0 under certain behavioral semantics. How to reasonably define behavioral semantics equivalence between processes and how to establish inference rules for such equivalence? The following discusses methods using labeled bisimulation, context barb congruence, and other algebraic theory methods to analyze and infer behavioral properties of mobile ambients processes.

1) Labeled Bisimulation

Labeled bisimulation characterizes equivalence of interaction behaviors between processes, which is an equivalence relation based on labeled transition semantics. It can generally be defined as shown in equation (8) for the labeled bisimulation relation between processes P and Q.

$P \approx Q$ if and only if for any label a: if $P \xrightarrow{a} P'$ then $Q \xrightarrow{a} Q'$ and $P' \approx Q'$, as shown in equation (8). Here, $P \xrightarrow{a} P'$ represents process P transitioning to process P' through label a; $Q \xrightarrow{a} Q'$ represents process Q transitioning to Q' through zero or more internal transitions, label a transition, and then zero or more internal transitions. Meanwhile, according to different granularities of behavioral equivalence requirements, the relation can be defined as strong bisimulation [22,23] or Early bisimulation [22], etc.

Labeled transition semantics is consistent with reduction semantics and expresses the possibility of a single process interacting with the external environment. In -calculus, all process communication interactions occur in a parallel space, where communication actions can be directly defined as labels to construct a commitment relation, and then a labeled transition system consistent with reduction semantics can be constructed based on this commitment relation. However, in ambient-centered calculi, ambients form a hierarchical space, and execution of interaction actions between ambients leads to reorganization of the hierarchical space. Methods based on commitment relations would result in labeled transition outcomes that are not only processes but also what are called concretions (containing both the participating part and remaining part of the process) [2]. Another approach is to construct a hardening relation $\rhd$ (hardening) that first hardens processes into concretions, and then constructs a labeled transition system using the hardening relation and reduction semantics [24], ensuring that labeled transition results are also processes.

Figure 1 [Figure 1: see original paper] shows the construction of a labeled transition rule consistent with reduction semantics (Red In) in Table 2 for MA calculus. In Figure 1, hardening rules (Harden Cap), (Harden Par), and (Harden Amb) can derive the hardening relation: in $n.P \rhd$ in $n.P'$. Meanwhile, from (Harden Amb) we get the hardening relation: $m[\text{in } n.P] \rhd m[\text{in } n.P']$, and from (Harden Cap) and (Harden Par) we obtain $m[\text{in } n.P]|n[Q] \xrightarrow{\tau} m[n[P|Q]]$.

According to the labeled transition rule (Trans Cap), we obtain the labeled transition: in $n.P \xrightarrow{\text{in } n} P$.

Finally, according to (Trans In), we obtain the labeled transition consistent with reduction rule (Red In) in Table 2, as shown in equation (9).

Constructing a reasonable labeled transition system is a necessary challenge for various ambient-centered extended calculi to establish bisimulation relations. By defining specific labels, richer behavioral bisimulation relations can be expressed (e.g., in literature [23], extending labels from actions to processes enables

a stronger bisimulation relation). The hardening relation divides processes into directly participating parts and parts existing as the environment, establishing a more intuitive labeled transition system. In membrane calculus [7] (an extension of MA calculus), the hardening relation is utilized to achieve fine-grained description of process interaction processes, directly defining the participating parts as membranes to express more complex interaction behaviors in membrane calculus (details can be found in literature [7]).

2) Context Barb Congruence

Given two processes, context barb congruence means that placing them in any context yields equivalent observation results (i.e., the two processes cannot be distinguished based on observation results). Observation results of processes (barbs) refer to whether a given process can exhibit specified results, such as specified ambients [24] or potential interaction actions in ambients [5,25]. For example, equations (10) and (11) define ways to specify that a process can exhibit an ambient named n and can exhibit a potential interaction action in n in some ambient.

$\{P \downarrow n\}$ means there exists P' such that $P \equiv (\nu\tilde{p})(n[P']|P'')$ with $n \notin \tilde{p}$, as defined in equation (10).
$\{P \downarrow \text{in } n\}$ means there exists P', P'' such that $P \equiv (\nu\tilde{p})(m[n[P']|P'']|P''')$ with $n \notin \tilde{p}$, as defined in equation (11).

Correspondingly, we can define $\{P \Downarrow a\}$ as a weak observation result: indicating that process P can exhibit observation result a after zero or more reductions.

In different ambient extended calculi, the definition methods for observation results can be determined according to the reduction semantics of the extended calculus and the construction requirements of observational equivalence. For example, literature [2] defines $\{P \downarrow n\}$ to indicate that process P has an ambient n that can interact (be entered or opened by other ambients). Context barb congruence defines and constructs corresponding contexts to analyze behavioral equivalence properties between processes; it ensures consistency with structural congruence semantics while establishing behavioral equivalence relations between processes at more abstract levels.

Determining or proving context barb congruence relations between processes requires solving two main problems:

a) Contexts may have multiple holes, and name capture may occur when processes fill context holes (i.e., free names in processes become bound), leading to complexity in equivalence determination processes.

b) Context observation results $\{C[P] \Downarrow a\}$ require examining zero or more reductions of $C[P]$, so context equivalence determination needs fine-grained analysis of context and process interaction processes.

By establishing a simplified context (Harness) and then deriving context lemmas [24] combined with structural congruence and reduction rules, it can be further

shown that simplified context barb congruence and general context barb congruence are equivalent [24]. Simplified contexts have only one hole and stipulate that bound names must be renamed before process filling to avoid name capture complexity. Simplified contexts can be defined as shown in equation (13).

The context lemma demonstrates that context barb congruence maintains equivalence relations under any name substitution within the name space of given processes, as stated in Lemma 1.

Lemma 1 (Context Lemma). For given processes P and Q, $P \approx Q$ if and only if for any name substitution σ in the domain of $fv(P) \cup fv(Q)$, any simplified context $C\{\}$ and observation result a, we have: $\{C[P]\sigma \Downarrow a\} \Leftrightarrow \{C[Q]\sigma \Downarrow a\}$.

As can be seen, context barb congruence under any name substitution in Lemma 1 already includes scenarios where name capture may occur when processes fill holes in general contexts. Furthermore, the equivalence between simplified context barb congruence and general context barb congruence can be proved by induction based on the structural congruence and reduction semantics of the given ambient calculus or extended ambient calculus, as seen in literature [26].

Similar to the labeled bisimulation method, literature [24] introduces a hardening relation for contexts and constructs a labeled transition system consistent with reduction semantics based on observation result sets, requiring examination of only three cases in Lemma 2 to derive observation results of reductions.

Lemma 2 (Activity Lemma) [24].
$C[P] \to R$ if and only if one of (a)~(c) holds:
(a) Reduction caused by internal interaction of the filling process: $P \to P'$ and $R \equiv C[P']$.
(b) Reduction caused by internal interaction of the context: $C\{\} \to C'\{\}$ and $R \equiv C'[P]$.
(c) Reduction caused by interaction between context and filling process: $C\{\} \xrightarrow{a} C'\{\}$ and $P \xrightarrow{\overline{a}} P'$ and $R \equiv C'[P']$.

Thus, through the context lemma and activity lemma, simplified context barb congruence can effectively express general context barb congruence.

3) Other Related Research
Both labeled bisimulation and context barb congruence methods obtain equivalence of potential behaviors between processes through fine-grained analysis of interaction processes based on structural congruence and reduction semantics of ambient calculus or extended ambient calculus. However, the former has a relatively weaker abstraction level than the latter. The former mainly analyzes potential behaviors from the process' s own structural composition and possible internal reduction actions, while the latter also observes potential behaviors when processes are placed in any context, including scenarios where processes may interact with contexts. However, determining context barb congruence means placing processes in any necessary contexts for obser-

vation, making it quite difficult to determine context barb congruence between arbitrary processes, except for certain specific process types given in literature [25,26]. To address this issue, literature [5] proposed behavioral equivalence relations based on observation results under weaker conditions: such as weak observation bisimulation based on labeled transitions and observation results [5], and context weak observation equivalence relations established in contexts only for specific types of observation results (not all observation results) [5], etc.

---

2 Spatial Logic and Model Checking for Mobile Ambients

2.1 Spatial Logic for Mobile Ambients

During modeling of mobile computation systems, it may be necessary to analyze model properties such as process spatial distribution structure, mobility liveness, and security. Using modal logic to describe these properties is an important method, and model checking techniques can be employed to automatically verify these logical properties.

Compared with traditional concurrent systems, the spatial structure of mobile computation systems consists of many relatively independent and distributed subsystems. Traditional modal logics such as LTL, CTL* temporal logics, or HML logic cannot express spatial structural properties between multiple subsystems in mobile computation systems. Ambient Logic [27] (AL logic) has become an exemplary spatial logic because it can well characterize spatial structures of mobile or distributed systems. The basic syntax of AL logic formulas is shown in equation (14), where η represents a name or name variable:

$$A ::= 0 \mid A|B \mid A@\eta \mid A \rhd B \mid \diamond A \mid \exists x.A \mid \neg A \mid A \vee B \mid \forall x.A \mid \diamond A$$

In the logic formulas defined by equation (14) syntax, 0 represents spatial null; $A|B$ represents spatial conjunction property; $A@\eta$ represents spatial location property; $A \rhd B$ represents spatial implication property; $\diamond A$ represents temporal property at some moment; $\diamond A$ represents spatial location property; the remaining formulas are traditional first-order logic formulas.

The satisfaction relation $P \vDash A$ indicates that process P satisfies closed formula A. The satisfaction relation of AL logic formulas relative to MA calculus processes is defined by Table 4 . For the modal operator $A@\eta$ representing a location in space, we need to define the relation $\rhd$ as shown in equation (15), where $\rhd^*$ represents the reflexive transitive closure of $\rhd$.

From Table 4, we can see that the satisfaction relation of spatial formulas in AL logic strictly depends on the structural congruence relation $\equiv$ of MA calculus. Thus, with the three basic spatial formulas 0, $A|B$, and $A@\eta$, MA processes can be directly and finely characterized. Meanwhile, through the weak temporal modality $\diamond$, reduction properties of MA processes can be characterized, especially introducing the weak spatial location modality $\diamond$ to achieve observation of

a certain spatial location. AL formulas have the expressive power to characterize properties of ambient calculus processes at any time and location.

$\forall P.P \models \top$ holds for any MA process. According to the satisfaction relation definition, this formula describes a testing observation: if any process P satisfies A, then $P|Q$ will satisfy A for any Q. Therefore, although AL logic does not use modal operators like HML logic that directly observe potential interaction behaviors of processes, it can indirectly test and observe action behavioral properties through modal operator $\triangleright$.

Regarding ambient logic with fixed-point operators, Lin Huimin proposed predicate -calculus for mobile ambients [28], whose fixed-point logic formulas can characterize infinite computational behaviors of processes. This logic modifies the weak temporal modality $\diamond$ to a Next modality X representing the next moment, and expresses $\diamond A$ through fixed-point formulas and Next modality X, while also adding revealing name modality formulas reveal $x.A$ and fresh name quantifier formulas fresh $x.A$.

It is worth noting that predicate -calculus for mobile ambients can characterize the weak spatial modality $\diamond$ in AL logic as shown in equation (16). In equation (16), predicate variable X does not occur freely in formula A, and the strong spatial modality $\Box$ is defined as $\Box A \equiv \neg \diamond \neg A$.

On the other hand, the decidability of spatial logic in model checking is crucial as it concerns whether the logic system can be practically used. The weak temporal modality $\diamond$ and spatial conjunction modality $|$ in AL logic significantly affect the expressive power of AL logic formulas. The sublogic of AL without weak temporal modality $\diamond$ is called AL static sublogic, which has very limited expressive power. Literature [29,30] studied decidability issues of various AL logic formula sets and different fragments of MA calculus models, with main conclusions:

a) Model checking is undecidable on MA calculus fragments containing replication operation $!P$, or on AL logic formula sets containing spatial conjunction modality $|$.

b) Model checking is decidable on MA calculus fragments without replication operation $!P$, and on AL logic formula sets without spatial conjunction modality $|$.

From these conclusions, besides the replication operation $!P$ in MA calculus processes that may cause infinite reduction behaviors and unbounded state expansion leading to undecidable model checking, the modal operator $|$ is the key cause of undecidability in AL logic, because spatial conjunction modality $|$ leads to infinite testing of processes in infinite process sets in MA. However, decidable AL logic formula sets suffer greatly reduced expressive power without spatial conjunction modality $|$, and cannot indirectly observe potential action interaction behaviors of processes.

2.2 Model Checking Algorithms for Mobile Ambients

The state explosion problem in model checking arises because detection algorithms need to search all states generated by alternating execution of concurrent reduction actions, requiring massive state information storage. Model checking for mobile ambients involves verification of both temporal and spatial properties simultaneously, especially when detecting fixed-point formulas in predicate -calculus ambient logic, which requires examining all states generated by process composition alternation, leading to even greater state space expansion.

On the other hand, traditional symbolic model checking, partial order reduction methods, and other model checking algorithms based on temporal logics LTL-x, CTL*-x, or behavioral modal logics HML, -calculus are difficult to directly apply in ambient calculus model checking. This is because the reduction process in ambient calculus model checking may cause alternating changes in system temporal and spatial properties. Therefore, equivalence of concurrent action execution paths in ambient calculus involves not only reduction partial order properties of processes themselves but also whether spatial properties in the current state satisfy partial order properties [31].

Currently, research on model checking methods for mobile ambients under different application requirements includes: literature [27] discussed model checking for finite ambient calculus without replication process $!P$ and finite ambient calculus where replication process $!P$ does not contain ambients; literature [29] studied complexity of model checking algorithms for ambient calculus and ambient logic; literature [32] proposed finite-control ambient calculus and provided model checking algorithms for it; literature [28,33] first presented model checking algorithms for finite-control ambient calculus and ambient logic with fixed-point operators.

Regarding efficient algorithms for mobile ambients model checking, building upon Lin Huimin' s predicate ambient logic model checking algorithm based on -calculus [33], Jiang Hua et al. [34,35] proposed a global model checking algorithm with time complexity exponential in the alternation depth d of fixed-point operators in ambient logic formulas and space complexity linear in d. In subsequent work [36], they studied two partial order relations satisfied by intermediate results in the computation process of predicate ambient logic model checking based on -calculus, proposing an efficient local model checking algorithm with time complexity exponential in $d/2$ and space complexity linear in the scale of computation node sets.

Other related research on ambient calculus model checking algorithms includes: literature [31] studied partial order reduction equivalence properties when concurrent operators, ambient operators, and temporal operators appear alternately in ambient logic; literature [26] also studied sufficiency issues of some single-threaded processes satisfying partial order properties in ambient calculus; and literature [37] proposed a method to encode mobile ambient processes in action temporal logic for logical behavior verification.

3 Modeling Applications of Mobile Ambients in Computation Systems

The mobile computation model centered on "ambients" can characterize essential features of distributed and mobile computation from a fundamental level using concepts of ambients, mobility, and security. In recent years, mobile ambients calculus has also been applied in modeling and analysis of cloud computing service interaction protocols and management systems, model-driven software engineering, and as mobile computation programming or system modeling languages.

Cloud computing is a virtualized computing resource service system that provides dynamic and scalable resources through interconnected networks. Main research on mobile ambients calculus modeling applications in cloud computing includes: literature [38] proposed virtually timed ambients calculus based on mobile ambients, enabling modeling of nested virtual machine systems with different computing capabilities under cloud computing frameworks, and proposed analysis methods for cloud computing models such as weak time bisimulation equivalence and reduction observation congruence; literature [39] used a variant of mobile ambients—membrane calculus—for formal modeling and analysis of Dryad-based cloud computing programming models; through objects in membrane variants and interactions between membranes to reflect data type changes during execution, serving as an auxiliary tool for program correctness verification; literature [40] proposed an abstract state machine based on safe ambients with passwords for modeling and analysis of cloud computing service models and access control technologies; literature [41] studied using ambient calculus and ambient logic to model role-based user authorization security policies in multi-domain mobile network systems, and used model checking technology to verify authorization security policies. Additionally, literature [42,43] studied applications of extended ambient calculi in network information security transmission protocols; literature [44] proposed an IoT operation and service interaction protocol model based on safe ambients, and used ambient logic for analysis.

In model-driven software engineering applications, representative work includes Ali et al. [45,46] introducing ambients as architectural elements in traditional software architecture description languages (ADLs) and aspect-oriented software architecture to intuitively describe locations, interactions, and cross-boundary behaviors of mobile objects. Ambient-PRISMA [46] is a .NET platform-based model-driven software engineering project implementing functions from software requirement definition, software architecture design, to automatic code generation. In this project, ambients as architectural elements express component/connector locations and their compositions, while also expressing behavioral constraints of component mobility functions in aspect-oriented architecture. In subsequent research of this project, literature [47] used channel ambient calculus [48] to construct a formal verification

framework for component/connector mobility behaviors and constraints in Ambient-PRISMA; literature [49] studied issues faced by interactions between devices, environments, and service components in runtime systems of mobile application systems, and proposed a solution based on ambient calculus software architecture models for runtime systems; literature [50] studied methods for characterizing location and mobility requirements of adaptive mobile resources in aspect-oriented software architecture, and proposed algorithms for dynamic mobile configuration of mobile processes based on service location and resource location requirement information. The above series of work established an exemplary framework for formal analysis and verification of mobile software architecture models in model-driven software engineering.

Regarding mobile computation programming or system modeling languages centered on "ambients," main work includes: literature [48] demonstratively proposed channel ambient systems, using extended ambient calculus models as core programming languages to construct runtime systems where mobile program objects move and interact across network nodes. Literature [31] studied using ambient calculus to express conditional selection semantics and loop control issues, proposing an extended ambient calculus model that uses a normalization operation to handle semantics of conditional selection structure processes and parameterized recursive processes, simplifying complexity of spatial logic semantics during model checking. Literature [51] proposed a mobile collaborative resource aggregation model, using mobile ambients processes to autonomously encapsulate resources, and employed logical reasoning processes of MA to perform logical reasoning calculations on some ambient operations and problems in the model.

---

4 Future Research Directions

The proposal of mobile ambients calculus theory represents significant progress in formal modeling theory and applications of mobile computation systems, with rich research achievements obtained over the past decade. However, as a formal model oriented toward practical modeling needs, many key issues remain to be further explored and innovated.

In algebraic theory research of mobile ambients, current behavioral equivalence determination methods based on the "bisimulation and two lemmas" framework (the various bisimulations and context barb congruences described in Section 1.2) have high computational complexity [5,24,52-54], making it still very difficult to use them for inferring complex properties of mobile computation systems. Therefore, future research can explore innovative approaches from two aspects:

a) Under the existing theoretical framework, study efficient procedures or steps to determine whether given processes have context equivalence, or weaker relations such as weak observation bisimulation, context weak ob-

servation equivalence, or labeled bisimulation, and other behavioral equivalences with higher abstraction levels than structural congruence. Type systems are static analysis tools that can infer whether processes share common structural features, and carefully constructed type systems can identify whether processes have specific potential behaviors [12]. Therefore, type inference technology can be combined to classify ambient calculus processes and apply heuristic procedures or steps for different categories to reduce computational complexity of behavioral equivalence determination.

b) Logic-based inference systems have relative maturity and sufficient expressive power [54], so a new equivalence theoretical framework for mobile ambients process behaviors can be explored: mapping mobile ambients processes to description and verification frameworks based on temporal logic and spatial logic, and verifying equivalence of mobile ambients behavioral properties through equivalence inference in temporal and spatial logic frameworks. Research methods in literature [37,54] provide some inspiration for the above ideas. However, behavioral properties characterized by mobile ambients calculus have interleaved temporal and spatial characteristics, making it a challenging task to research equivalence theoretical frameworks based on temporal logic and spatial logic with rich expressive power.

In spatial logic and model checking research of mobile ambients, decidable AL logic formula sets suffer greatly reduced expressive power without spatial conjunction modality |, and cannot indirectly observe potential action interaction behaviors of processes. Therefore, exploring ambient logic that can observe process behaviors according to application backgrounds [55,56], especially mobile ambients behavioral observation logic satisfying higher abstraction levels based on ambient observational equivalence theory, represents an important future research direction.

In model checking research for mobile ambients, efficient core algorithms for mobile ambients model checking oriented toward computation system modeling and verification [36] and practical mobile ambients model checking tools should become main research work: constructing efficient and practical ambient calculus model checking software can promote in-depth application of mobile ambients calculus in formal modeling of mobile computation systems, and will also promote further research and development of efficient ambient calculus model checking algorithms and software.

In modeling applications of mobile ambients in computation systems, research on quantifiable real-time interaction behavioral properties needed for practical modeling and verification, and probabilistic interaction behaviors for quantitative analysis of system uncertainty, are important for further promoting mobile ambients calculus in modeling and analysis of interaction protocols in actual computation systems. Meanwhile, cyber-physical systems (CPS) exhibit typical characteristics of multi-level autonomy and interaction as characterized

by mobile ambients calculus, such as loose coupling between computational entities, strong autonomy, and spatiotemporal dynamic evolution [44,57,58]. Therefore, work on mobile ambients calculus and model checking technology in modeling and verification of CPS interaction protocols also has important practical significance.

References

[17] Aman B, Ciobanu G. Describing the immune system using enhanced mobile membranes [J]. Electronic Notes in Theoretical Computer Science, 2008, 194 (3): 5-18.

[18] Bodei C, Brodo L, Gori R, et al. A static analysis for brane calculi providing global occurrence counting information [J]. Theoretical Computer Science, 2017, 696 (10): 11-51.

[19] Aman B, Ciobanu G. Behavioural observations of cell movements with timing aspects [J]. Nano Communication Networks, 2015, 6 (3): 73-87.

[20] Boukharrou R, Ilie J-M, Saidouni D E, et al. Spatio-temporal planning for mobile ambient agents [J]. Procedia Computer Science, 2015, 56 (1): 123-132.

[21] Boukharrou R, Ilie J M, Saidouni D E, et al. Contextual time reasoning for mobile ambient agents [J]. International Journal of Wireless and Mobile Computing, 2016, 10 (3): 250-260.

[1] Cardelli L, Gordon A D. Mobile ambients [J]. Theoretical Computer Science, 2000, 240 (1): 177-213.

[22] Merro M, Hennessy M. Bisimulation congruences in safe ambients [J]. Symposium on Principles of Programming Languages, 2002, 37 (1): 71-80.

[2] Levi F, Sangiorgi D. Controlling interference in ambients [C]// Proc of Conference Record of the Annual ACM Symposium on Principles of Programming Languages. New York: ACM Press, 2000: 352-364.

[23] Jiang Hua, Tan Xinxing. Bisimulations in the boxed safe ambients with password [C]// Proc of International Conference on Information Technology: New Generations. [S. l.]: IEEE Computer Society, 2009: 1523-1528.

[3] Bugliesi M, Castagna G. Secure safe ambients [J]. Symposium on Principles of Programming Languages, 2001, 36 (3): 222-235.

[24] Gordon A D, Cardelli L. Equational properties of mobile ambients [J]. Mathematical Structures in Computer Science, 2003, 13 (3): 371-408.

[4] Bugliesi M, Castagna G, Crafa S, et al. Boxed ambients [C]// Proc of International Symposium on Theoretical Aspects of Computer Software. Berlin: Springer-Verlag, 2001: 38-63.

[25] Vigliotti M G, Phillips I. Barbs and congruences for safe mobile ambients [J]. Electronic Notes in Theoretical Computer Science, 2002, 66 (3): 37-51.

[5] Fu Yuxi. Fair ambients [J]. Acta Informatica, 2007, 43 (8): 535-594.

[26] Guan Xudong, Yang Yiling, You Jinyuan. Further control on the grave interference in mobile ambients [J]. Journal of Software, 2002, 13 (5): 1018-1023.

[6] Siewe F, Zedan H, Cau A, et al. The calculus of context-aware ambients [J]. Journal of Computer and System Sciences, 2011, 77 (4): 597-620.

[27] Cardelli L, Gordon A D. Anytime, anywhere: modal logics for mobile ambients [C]// Proc of Conference Record of the Annual ACM Symposium on Principles of Programming Languages. New York: ACM Press, 2000: 365-377.

[7] Cardelli L. Brane calculi [C]// Proc of International Conference on Computational Methods in Systems Biology. Berlin: Springer-Verlag, 2004: 257-278.

[28] Lin Huimin. Predicate -calculus for mobile ambients [J]. Journal of Computer Science and Technology, 2005, 20 (1): 95-104.

[8] Kwiatkowska M Z, Norman G, Parker D, et al. Probabilistic mobile ambients [J]. Theoretical Computer Science, 2009, 410 (12): 1113-1138.

[29] Charatonik W, Dalzilio S, Gordon A D, et al. The complexity of model checking mobile ambients [J]. Foundations of Software Science and Computation Structure, 2001: 152-167.

[9] Ciobanu G. Interaction in time and space [J]. Electronic Notes in Theoretical Computer Science, 2008, 203 (3): 5-18.

[30] Yan Feng, Chen Taolue, Han Tingting, et al. A definition framework of spatial logic and decidability [J]. Computer Science, 2006, 33 (6): 7-10.

[10] Long Huan. On the semantics and expressiveness of ambient calculi [D]. Shanghai: Shanghai Jiao Tong University, 2009.

[31] Lin Rongde. Research on key issues of mobile ambients and model checking applications [D]. Guangzhou: South China University of Technology, 2010.

[11] Siewe F. A privacy type system for context-aware mobile ambients [J]. Procedia Computer Science, 2015, 52 (1): 98-105.

[32] Charatonik W, Gordon A D, Talbot J M. Finite-control mobile ambients [C]// Proc of European Symposium on Programming Languages and Systems. Berlin: Springer-Verlag, 2002: 295-313.

[12] Pasqualin D P, Vizzotto J K, Piveta E K, et al. Typed context awareness ambient calculus for pervasive applications [J]. Formal Aspects of Computing, 2015, 27 (5): 885-916.

[33] Lin Huimin. A predicate spatial logic for mobile processes [J]. Science in China Serise F: Information Science, 2004, 47 (3): 394-408.

[13] Gul N. A Calculus of mobility and communication for ubiquitous computing [C]// Proc of International Workshop on Automated Specification and Verification of Web Systems. [S. l.]: Open Publishing Association, 2015: 6-22.

[34] Jiang Hua. Efficient global model-checking for propositional -calculus [J]. Journal of Computer Research and Development, 2010, 47 (8): 1424-1433.

[14] Barbanera F, Bugliesi M, Dezaniciancaglini M, et al. Space-aware ambients and processes [J]. Theoretical Computer Science, 2007, 373 (1): 41-69.

[35] Jiang Hua, Li Xiang. Model checking for mobile ambients [J]. Journal of Computer Research and Development, 2009, 46 (10): 1750-1757.

[15] Ciobanu G, Aman B. On the relationship between membranes and ambients [J]. BioSystems, 2008, 91 (3): 515-530.

[36] Jiang Hua. Model checking for first-order predicate ambient logic based on -calculus with partial orders [J]. Chinese Journal of Computers, 2016, 39 (12): 2547-2561.

[16] Paun G. Membrane computing and brane calculi (some personal notes) [J]. Electronic Notes in Theoretical Computer Science, 2007, 171 (2): 5-8.

[37] Aman B, Ciobanu G. Expressing mobile ambients in temporal logic of actions [C]// Proc of Romanian Academy Series a-Mathematics Physics Technical Sciences Information Science. Bucharest: Editura ACAD Romane, 2014, 15 (1): 95-104.

[38] Johnsen E B, Steffen M, Stumpf J B. Virtually timed ambients: a calculus of nested virtualization [J]. Journal of Logical and Algebraic Methods in Programming, 2018, 94 (1): 109-127.

[39] Liu Lei, Liu Feng, Ren Junqi, et al. A formal description method of dryad using membrane calculus [J]. Journal of Harbin Engineering University, 2016, 37 (11): 1539-1545.

[40] Lu Chen. A cloud computing model based on extended ambient ASM [D]. Nanning: Guangxi University for Nationalities, 2016.

[41] Unal D, Caglayan M U. XFPM-RBAC: XML-based specification language for security policies in multidomain mobile networks [J]. Security and Communication Networks, 2013, 6 (12): 1420-1444.

[42] Jiang Hua, Li Xiang. Boxed Safe Ambients with password and simulation of e-mail system [C]// Proc of IEEE International Symposium on Information Technologies and Applications in Education. [S. l.]: IEEE Computer Society, 2007: 337-342.

[43] Jiang Hua, Tan Xinxing, Li Xiang. Boxed safe ambients with password and application on the Internet [C]// Proc of IEEE International Conference on Networking, Sensing and Control. [S. l.]: IEEE Computer Society, 2008: 472-477.

[44] Cong Xinyu, Yu Huiqun. Modeling and verification of mobility in cyber physical systems based on mobile safe ambients [J]. Journal of East China University of Science and Technology: Natural Science Edition, 2015, 41 (3): 391-395.

[45] Ali N, Millan C, Ramos I, et al. Developing mobile ambients using an aspect-oriented software architectural model [C]// Proc of International Conference on Move to Meaningful Internet Systems. Berlin: Springer-Verlag, 2006: 1633-1649.

[46] Ali N, Ramos I, Solis C, et al. Ambient-PRISMA: ambients in mobile aspect-oriented software architecture [J]. Journal of Systems and Software, 2010, 83 (6): 937-958.

[47] Ali N, Tuosto E. Architectural models of ambient-PRISMA in channel ambient calculus [C]// Proc of IEEE Software Engineering Workshop. [S. l.]: IEEE Computer Society, 2012: 1-10.

[48] Phillips A, Yoshida N, Eisenbach S, et al. A distributed abstract machine for boxed ambient calculi [J]. Lecture Notes in Computer Science, 2004, 2986: 155-170.

[49] Ali N, Solis C. Mobile architectures at runtime: research challenges [C]// Proc of International Conference on Mobile Software Engineering and Systems. New York: ACM Press, 2014: 41-44.

[50] Ali N, Solis C. Self-adaptation to mobile resources in service oriented architecture [C]// Proc of IEEE International Conference on Mobile Services. [S. l.]: IEEE Computer Society, 2015: 407-414.

[51] Hao Yanmei. Research on aggregation model of mobile collaboration [D]. Shijiazhuang: Hebei University of Economics & Business, 2014.

[52] Busi N, Zavattaro G. Deciding reachability in mobile ambients [C]// Proc of the 14th European Symposium on Programming. London: Springer-Verlag, 2005: 248-262.

[53] Maffeis S, Phillips I. On the computational strength of pure ambient calculi [J]. Theoretical Computer Science, 2005, 330 (3): 501-551.

[54] Wei Jun, Feng Yulin. Analysis of formal models and methods on mobile computing [J]. Journal of Computer Research and Development, 2000, 37 (2): 129-139.

[55] Lin Rongde, Xi Jianqing, Guo Yubin. Dormancy and spatial logic of mobile ambients [J]. Computer Science, 2009, 36 (3): 173-178.

[56] Chen Jiang, Lin Rongde. Spatial logic with behavioral observations for ambient calculus [J]. Journal of Nanjing Normal University: Engineering and Technology Edition, 2011, 11 (4): 70-76.

[57] Peng Chaoyu. Research on security of model checking-based cyber-physical systems [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2015.

[58] Lanotte R, Merro M. A semantic theory of the Internet of things [J]. Information and Computation, 2018, 259: 72-101.

Note: Figure translations are in progress. See original paper for figures.