Scientific Research

Reasoning about Context Information in Cloud Computing Environments

Norihiro Kamide[1], Yishui Zhu[1,2]

[1]Faculty of Information Technology and Business, Cyber University, Japan; [2]Graduate School of Human Sciences, Waseda University, Tokyo, Japan.
Email: drnkamide08@kpd.biglobe.ne.jp, syuisui@gmail.com

ABSTRACT

The notion of context provides flexibility and adaptation to cloud computing services. Location, time identity and activity of users are examples of primary context types. The motivation of this paper is to formalize reasoning about context information in cloud computing environments. To formalize such context-aware reasoning, the logic LCM of context-mixture is introduced based on a Gentzen-type sequent calculus for an extended resource-sensitive logic. LCM has a specific inference rule called the context-mixture rule, which can naturally represent a mechanism for merging formulas with context information. Moreover, LCM has a specific modal operator called the sequence modal operator, which can suitably represent context information. The cut-elimination and embedding theorems for LCM are proved, and a fragment of LCM is shown to be decidable. These theoretical results are intended to provide a logical justification of context-aware cloud computing service models such as a flowable service model.

Keywords: Context Information; Context-Mixture Rule; Sequent Calculus; Resource-Sensitive Reasoning; Context-Aware Reasoning

1. Introduction

1.1. Contexts in Cloud Computing Environments

The motivation of this paper is to formalize reasoning about context information in cloud computing environments. To formalize such context-aware reasoning, the logic LCM of context-mixture is introduced as a Gentzen-type sequent calculus based on linear logic [1,2], which is known to be a useful resource-sensitive logic. LCM has a specific inference rule called the context-mixture rule and a specific modal operator called the sequence modal operator [3,4]. The cut-elimination and embedding theorems for LCM are proved as the main results of this paper. A fragment of LCM is also shown to be decidable. These theoretical results are intended to provide a concrete logical justification of context-aware cloud computing service models such as a flowable service model [5,6].

The definitions of cloud computing, including on-demand, pay-by-use, virtualized and dynamically-scalable, imply the characteristics of cloud computing environments [7,8]. Cloud-related issues have been discussed and studied based on the notion of contexts which include *location*, *time*, *identity* and *activity* of users. The use of *context* is known to be very important in cloud and ubiquitous computing.

There is a widely accepted definition of context [9]:

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between the user and application, including the user and application themselves.

Location, time, identity and activity are primary context types for characterizing the situation of a particular entity. Contexts can be classified into three categories [6]: *nature context*, *human context* and *culture context*. In the present paper, nature context is especially considered. Nature context includes when (time context) and where (location or space context) information.

1.2. Flowable Services

Context provides flexibility and adaptation to services. A *flowable service*, which is a new notion of context-aware cloud computing services, is a logical stream that organizes and provides circumjacent services in such a way that they are perceived by individuals as those naturally embedded in their surrounding environments [5,6,10-14]. A flow of service is a metaphor for a subconsciously controlled navigation that guides the user through ful-

fillment of a flowable service process that fits the user's context and situation and runs smoothly with unbroken continuity in an unobtrusive and supportive way. Flowable services can be useful to context-aware cloud computing applications such as *Cloud Campus* which is the e-learning environment of Cyber University in Japan.

The original intention of the *flowable service model* is to apply *resources* in open cloud environments [5,6,10-14]. The model uses intensifying *context information* to adjust services flow to be more usable. The model shares resources or services fairly and to utmost extent. To formalize reasoning about the context-aware flowable service model in open cloud computing environments, we need an appropriate logic that can represent the following three items:

1) Context-mixture rule;
2) Resource-sensitive reasoning;
3) Context information.

1.3. Context-Mixture Rule

In this paper, the logic LCM of context-mixture, which can represent the above three items (context-mixture rule, resource-sensitive reasoning and context information), is introduced as a Gentzen-type sequent calculus based on linear logic. LCM has a specific inference rule called the *context-mixture rule*, which can naturally represent a mechanism for merging formulas with context information.

Merging formulas with context information, which represents an interaction between different context information, is required for suitable representation of context-aware flowable services, since to handle various kinds of context information is an important issue for flowable services. We call here such a merging mechanism *context-mixture*.

The notion of context-mixture is also important for representing *deployment models* in cloud computing environments [8]. Deployment models are classified as private cloud, community cloud, public cloud and hybrid cloud. The cloud infrastructure of hybrid cloud is a "mixture" (or composition) of two or more distinct cloud infrastructures (private, community or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

The context-mixture rule of LCM is of the form:

$$\frac{\Gamma_1, \Delta \Rightarrow \gamma \quad \Gamma_2, \Delta \Rightarrow \gamma}{\Gamma_1, \Gamma_2, \Delta \Rightarrow \gamma} \, (\text{mixture})$$

where the multisets Γ_1 and Γ_2 of formulas with context information are mixed by this rule.

The rule (mixture) was introduce in [15], and was called the *mingle rule*. The name "mingle" was from the originnal version [16] of the mingle rule:

$$\frac{\Gamma_1 \Rightarrow \Delta_1 \quad \Gamma_2 \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2} \, (\text{mingle})$$

This original rule and the corresponding Hilbert-style axiom scheme $\alpha \rightarrow (\alpha \rightarrow \alpha)$ have been studied in formalizing "relevant" human reasoning [16,17], grammatical reasoning [18] and reasoning about communication-merge in process algebras [15].

As presented in [15], the context-mixture rule has been used for representing communication-merge in process algebras. This is also justified for the present study, since there is the slogan "context-as-process" presented in [19]: We must consider "context as a part of process of interacting with an ever-changing environment that is composed of reconfigurable, migratory, distributed and multiscale resources" because it is "not simply the state of a predefined environment with a fixed set of interaction resources".

1.4. Resource-Sensitive Reasoning

The logic LCM of context-mixture is obtained from *linear logic* [1,2] by adding the context-mixture rule (mixture) and a sequence modal operator, which represents a sequence of symbols. By the sequence modal operator in LCM, we can appropriately express "context information" in "resource-sensitive reasoning".

The notion of "resources", encompassing concepts such as processor time, memory, cost of components and energy requirements, is fundamental to computational systems. This notion is also very important for handling efficient resource management in cloud computing environments [20]. Examples of resources in cloud computing environments include storage, processing, memory, and network bandwidth [8].

Linear logic can elegantly represent the notion of "resources" [1]. In linear logic, the concept of "resource consumption" can be represented by using the linear implication connective $\rightarrow$ and the fusion connective, and the concept of "reusable resource" can be represented by using the linear exponential operator $!$. A typical example formula is:

$$\text{coin} * \text{coin} \rightarrow \text{coffee} * (!\text{water})$$

This example means "if we spend two coins, then we can have a cup of coffee and as much of water as we like" when the price of coffee is two coins and water is free. It is to be noted that this example cannot be expressed using classical logic, since the formula $\text{coin} \wedge \text{coin}$ (two coins) in classical logic is logically equivalent to coin (one coin), *i.e.*, classical logic has no resource-awareness.

1.5. Context Information

In order to discuss certain real and practical examples, the resource descriptions in linear logic should be more

fine-grained and expressive and capable of conveying context information. For example, the following expressions may be necessary for some practical situations:

$$[\text{teashop}\ ;\ \text{john}](\text{coin} * \text{coin} * \text{coin} \rightarrow$$
$$[1\text{min}\ ;\ 1\text{min}]\text{coffee} * [1\text{min}]\text{water}),$$

$$[\text{cafeteria}\ ;\ \text{john}]$$
$$(\text{coin} * \text{coin} \rightarrow [1\text{min}]\text{coffee})$$

These examples respectively mean "in a teashop, if John spends three coins, then he can have a cup of coffee after two minutes and a cup of water after one minute," and "in a cafeteria, if John expends two coins, then he can have a cup of coffee after one minute." In these examples, the expressions $[\text{teashop}\ ;\ \text{John}]$, $[\text{cafeteria}\ ;\ \text{John}]$, $[1\min\ ;\ 1\min]$ and $[1\min]$, which are regarded as "context information", can naturally be represented by the sequence modal operator in LCM.

As presented in [4], the following expressions are available in a subsystem of LCM:

$$[\text{client}\ ;\ \text{server}\ ;\ t]$$
$$(\text{incorrect User ID} * \text{password}) \rightarrow [\text{server}\ ;\ \text{client}]\text{error},$$

$$[\text{server}\ ;\ \text{client}]$$
$$(\text{error} * \text{error} * (!\text{error})) \rightarrow [\text{server}\ ;\ \text{client}]\text{reject}$$

which respectively mean:
"if a client sends an incorrect user ID and a correct password to login to a server at the t -th login attempt, then the server returns an error message to the client."
"if a server returns the error messages more than twice to a client, then the server returns the password reject message to the client."

Note that the error messages are expressed as a "resource" by using the connectives $*$ and $!$, and the "information" on servers, clients, and login-attempts is expressed by the sequence modal operator.

1.6. Informational Interpretation

The reason underlying the use of the notion of "sequences" in the sequence modal operator is explained below. The notion of "sequences" is fundamental to practical reasoning in computer science because it can appropriately represent "data sequences", "program-execution sequences", "action sequences", "time sequences" etc. The notion of sequences is thus useful to represent the notions of "information", "attributes", "trees", "or-ders", "preferences" and "ontologies". To represent "context information" by sequences is especially suitable because a sequence structure gives a *monoid* $\langle M, ;, \varnothing \rangle$ with *informational interpretation* [21]:

1) M is a set of pieces of (ordered or prioritized) in-

formation (*i.e.*, a set of sequences);

2) ; is a binary operator (on M) that combines two pieces of information (*i.e.*, a concatenation operator on sequences);

3) $\varnothing$ is the empty piece of information (*i.e.*, the empty sequence).

Based upon the informational interpretation, a formula of the form $[b_1 ; b_2 ; \cdots ; b_n]\alpha$ intuitively means that "α is true based on a sequence $b_1 ; b_2 ; \cdots ; b_n$ of (ordered or prioritized) information pieces." Further, a formula of the form $[\varnothing]\alpha$, which coincides with α, intuitively means that "α is true without any information (*i.e.*, it is an eternal truth in the sense of classical logic)."

2. The Logic LCM of Context-Mixture

Prior to the precise discussion, the language of the proposed logic is introduced below. *Formulas* are constructed from propositional variables, 1 (multiplicative truth constant), $\top$ (additive truth constant), $\bot$ (additive falsity constant), $\rightarrow$ (implication), $\wedge$ (conjunction), $*$ (fusion), $\vee$ (disjunction), ! (exponential), and $[b]$ (sequence modal operator) where b is a sequence. *Sequences* are constructed from atomic sequences, $\varnothing$ (empty sequence) and; (composition). Lower-case letters $b, c, \cdots$ are used for sequences, lower-case letters $p, q, \cdots$ are used for propositional variables, Greek lower-case letters $\alpha, \beta, \cdots$ are used for formulas, and Greek capital letters $\Gamma, \Delta, \cdots$ are used for finite (possibly empty) multisets of formulas. For any $\# \in \{!, [b]\}$, an expression $\#\Gamma$ is used to denote the multiset $\{\#\gamma \mid \gamma \in \Gamma\}$. The symbol $\equiv$ is used to denote the equality of sequences (or multisets) of symbols. An expression $[\varnothing]\alpha$ means α, and expressions $[\varnothing ; b]\alpha$ and $[b ; \varnothing]\alpha$ mean $[b]\alpha$. A *sequent* is an expression of the form $\Gamma \Rightarrow \gamma$ where γ is nonempty. It is assumed that the terminological conventions regarding sequents (e.g., antecedent and succedent) are the usual ones. If a sequent S is provable in a sequent calculus L, then such a fact is denoted as $L \vdash S$ or $\vdash S$. The parentheses for $*$ is omitted since $*$ is associative, *i.e.*,
$\vdash \alpha * (\beta * \gamma) \Rightarrow (\alpha * \beta) * \gamma$ and
$\vdash (\alpha * \beta) * \gamma \Rightarrow \alpha * (\beta * \gamma)$ for any formulas α, β and γ. A rule R of inference is said to be *admissible* in a sequent calculus L if the following condition is satisfied: for any instance

$$\frac{S_1 \cdots S_n}{S}$$

of R, if $L \vdash S_i$ for all i, then $L \vdash S$.

Definition 2.1. *Formulas and sequences are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, respectively:*

$$\alpha ::=$$
$$p\,|\,1\,|\,\top\,|\,\bot\,|\ \alpha \to \alpha\,|\ \alpha \wedge \alpha\,|\ \alpha * \alpha\,|\ \alpha \vee \alpha\,|\,!\,\alpha\,|\,[b]\,\alpha$$
$$b ::= e\,|\,\varnothing\,|\,b\,;\,b$$

The set of sequences (including the empty sequence) is denoted as SE. An expression $\left[\hat{d}\right]$ is used to represent $[d_0][d_1]\cdots[d_i]$ with $i \in \omega$ and $d_0 \equiv \varnothing$, i.e., $\left[\hat{d}\right]$ can be the empty sequence. Also, an expression $\hat{d}$ is used to represent $d_0; d_1; \cdots; d_i$ with $i \in \omega$ and $d_0 \equiv \varnothing$.

The logic LCM of context-mixture is then introduced below.

Definition 2.2. *The initial sequents of* LCM *are of the form: for any propositional variable p,*

$$\left[\hat{d}\right]p \Rightarrow \left[\hat{d}\right]p \qquad \Rightarrow \left[\hat{d}\right]\mathbf{1}$$
$$\Gamma \Rightarrow \left[\hat{d}\right]\top \qquad \left[\hat{d}\right]\bot, \Gamma \Rightarrow \gamma$$

The cut rule of LCM is of the form:

$$\frac{\Gamma \Rightarrow \alpha \quad \alpha, \Delta \Rightarrow \gamma}{\Gamma, \Delta \Rightarrow \gamma}\,(\text{cut})$$

The context-mixture rule of LCM is of the form:

$$\frac{\Gamma_1, \Delta \Rightarrow \gamma \quad \Gamma_2, \Delta \Rightarrow \gamma}{\Gamma_1, \Gamma_2, \Delta \Rightarrow \gamma}\,(\text{mixture})$$

The sequence rules of LCM are of the form:

$$\frac{\left[\hat{d}\right][b][c]\alpha, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right][b;c]\alpha, \Gamma \Rightarrow \gamma}\,(;\text{left})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right][b][c]\alpha}{\Gamma \Rightarrow \left[\hat{d}\right][b;c]\alpha}\,(;\text{right})$$

The logical inference rules of LCM are of the form:

$$\frac{\Gamma \Rightarrow \gamma}{\left[\hat{d}\right]\mathbf{1}, \Gamma \Rightarrow \gamma}\,(\mathbf{1}\text{we})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right]\alpha \quad \left[\hat{d}\right]\beta, \Delta \Rightarrow \gamma}{\left[\hat{d}\right](\alpha \to \beta), \Gamma, \Delta \Rightarrow \gamma}\,(\to \text{left})$$

$$\frac{\left[\hat{d}\right]\alpha, \Gamma \Rightarrow \left[\hat{d}\right]\beta}{\Gamma \Rightarrow \left[\hat{d}\right](\alpha \to \beta)}\,(\to \text{right})$$

$$\frac{\left[\hat{d}\right]\alpha, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right](\alpha \wedge \beta), \Gamma \Rightarrow \gamma}\,(\wedge \text{left1})$$

$$\frac{\left[\hat{d}\right]\beta, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right](\alpha \wedge \beta), \Gamma \Rightarrow \gamma}\,(\wedge \text{left2})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right]\alpha \quad \Gamma \Rightarrow \left[\hat{d}\right]\beta}{\Gamma \Rightarrow \left[\hat{d}\right](\alpha \wedge \beta)}\,(\wedge \text{right})$$

$$\frac{\left[\hat{d}\right]\alpha, \left[\hat{d}\right]\beta, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right](\alpha * \beta), \Gamma \Rightarrow \gamma}\,(*\text{left})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right]\alpha \quad \Delta \Rightarrow \left[\hat{d}\right]\beta}{\Gamma, \Delta \Rightarrow \left[\hat{d}\right](\alpha * \beta)}\,(*\text{right})$$

$$\frac{\left[\hat{d}\right]\alpha, \Gamma \Rightarrow \gamma \quad \left[\hat{d}\right]\beta, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right](\alpha \vee \beta), \Gamma \Rightarrow \gamma}\,(\vee \text{left})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right]\alpha}{\Gamma \Rightarrow \left[\hat{d}\right](\alpha \vee \beta)}\,(\vee \text{right1})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right]\beta}{\Gamma \Rightarrow \left[\hat{d}\right](\alpha \vee \beta)}\,(\vee \text{right2})$$

$$\frac{\left[\hat{d}\right]\alpha, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right]!\alpha, \Gamma \Rightarrow \gamma}\,(!\text{left})$$

$$\frac{\left[\hat{d_1}\right]!\gamma_1, \cdots, \left[\hat{d_k}\right]!\gamma_k \Rightarrow \left[\hat{d}\right]\alpha}{\left[\hat{d_1}\right]!\gamma_1, \cdots, \left[\hat{d_k}\right]!\gamma_k \Rightarrow \left[\hat{d}\right]!\alpha}\,(!\text{right})$$

$$\frac{\left[\hat{d}\right]!\alpha, \left[\hat{d}\right]!\alpha, \Gamma \Rightarrow \gamma}{\left[\hat{d}\right]!\alpha, \Gamma \Rightarrow \gamma}\,(!\text{co})$$

$$\frac{\Gamma \Rightarrow \gamma}{\left[\hat{d}\right]!\alpha, \Gamma \Rightarrow \gamma}\,(!\text{we})$$

It is remarked that Girard's intuitionistic linear logic ILL is a subsystem of LCM: It is obtained from LCM by deleting (mixture) and the sequence modal operators.

The sequents of the form $\left[\hat{d}\right]\alpha \Rightarrow \left[\hat{d}\right]\alpha$ for any formula α are provable in cut-free LCM. This fact is shown by induction on α.

The (possibly empty) multiset expression Δ in (mixture) is needed to show the cut-elimination theorem for an extended linear logic with (mixture) [15].

Proposition 2.3. *The following rules are admissible in cut-free* LCM.

$$\frac{\Gamma \Rightarrow \gamma}{[b]\Gamma \Rightarrow [b]\gamma}\,([b]\text{regu})$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right](\alpha \to \beta)}{\left[\hat{d}\right]\alpha, \Gamma \Rightarrow \left[\hat{d}\right]\beta}\,(\to \text{right}^{-1})$$

$$\frac{\left[\hat{d}\right](\alpha * \beta),\Gamma \Rightarrow \gamma}{\left[\hat{d}\right]\alpha,\left[\hat{d}\right]\beta,\Gamma \Rightarrow \gamma}\left(*\text{left}^{-1}\right)$$

$$\frac{\left[\hat{d}\right][b;c]\alpha,\Gamma \Rightarrow \gamma}{\left[\hat{d}\right][b][c]\alpha,\Gamma \Rightarrow \gamma}\left(;\text{left}^{-1}\right)$$

$$\frac{\Gamma \Rightarrow \left[\hat{d}\right][b;c]\alpha}{\Gamma \Rightarrow \left[\hat{d}\right][b][c]\alpha}\left(;\text{right}^{-1}\right)$$

Proof. Straightforward. Here, we show only for the rule ($[b]$ regu) by induction on the proofs P of $\Gamma \Rightarrow \gamma$ in cut-free LCM. We distinguish the cases according to the last inference of P. We show only the following cases.

Case ($\Rightarrow \left[\hat{d}\right]\mathbf{1}$): The last inference of P is of the

form: $\Rightarrow \left[\hat{d}\right]\mathbf{1}$. In this case, $\Rightarrow [b]\left[\hat{d}\right]\mathbf{1}$ is also an

initial sequent.

Case ($\rightarrow$ left): The last inference of P is of the form:

$$\frac{\Pi \Rightarrow \left[\hat{d}\right]\alpha \quad \left[\hat{d}\right]\beta,\Delta \Rightarrow \gamma}{\left[\hat{d}\right](\alpha \rightarrow \beta),\Pi,\Delta \Rightarrow \gamma}\left(\rightarrow\text{left}\right)$$

By induction hypothesis, we obtain:

$$\text{LCM}-(\text{cut})\vdash [b]\Pi \Rightarrow [b]\left[\hat{d}\right]\alpha$$

and

$$\text{LCM}-(\text{cut})\vdash [b]\left[\hat{d}\right]\beta,[b]\Delta \Rightarrow [b]\gamma.$$

We then obtain the required fact:

$$\frac{[b]\Pi \Rightarrow [b]\left[\hat{d}\right]\alpha \quad [b]\left[\hat{d}\right]\beta,[b]\Delta \Rightarrow [b]\gamma}{[b]\left[\hat{d}\right](\alpha \rightarrow \beta),[b]\Pi,[b]\Delta \Rightarrow [b]\gamma}\left(\rightarrow\text{left}\right) \quad \blacksquare$$

An expression $\alpha \Leftrightarrow \beta$ means two sequents $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \alpha$.

Proposition 2.4. *The following sequents are provable in cut-free* LCM: *for any formulas* α,β *and any* $b,c \in \text{SE}$,

1) $[b]\# \Leftrightarrow \#$ where $\# \in \{\mathbf{1},\bot,\top\}$;

2) $[b](\alpha \circ \beta) \Leftrightarrow [b]\alpha \circ [b]\beta$ where $\circ \in \{\rightarrow,\wedge,*,\vee\}$;

3) $[b]!\alpha \Leftrightarrow ![b]\alpha$;

4) $[b;c]\alpha \Leftrightarrow [b][c]\alpha$.

3. Some Results on LCM

Definition 3.1. LM *is obtained from* LCM *by deleting* {(;left), (;right)} *and all the expressions as* $\left[\hat{d}\right]$ *appearing in the initial sequents and the logical inference rules. The names of the logical inference rules of* LM *are denoted by labeling* "$\star$" *in superscript position, e.g.,*

($\rightarrow \text{left}^\star$).

The logic LM is equivalent to a logic introduced in [15]. Indeed, LM is equivalent to the $\mathbf{0}$-free fragment of MILLm [15] where $\mathbf{0}$ is the multiplicative falsity constant. As shown in [15], the cut-elimination theorem holds for LM. This fact will be used to show the cut-elimination theorem for LCM. The fact that the !-free fragment LM$^-$ of LM is decidable will also be used to show the decidability of the !-free fragment LCM$^-$ of LCM.

Definition 3.2. *We fix a countable set* Ψ *of propositional variables, and define the sets* $\Psi_{\hat{d}} := \left\{p_{\hat{d}} \mid p \in \Psi\right\}$

$\left(\hat{d} \in \text{SE}\right)$ *of propositional variables where* $p_{\varnothing} := p$,

i.e., $\Psi_{\varnothing} := \Psi$. *The language (or the set of formulas)* $\mathcal{L}_s$ *of* LCM *is obtained from* Ψ , $\mathbf{1},\top,\bot,\rightarrow,\wedge,*,\vee,!$ *and* $[b]$. *The language (or the set of formulas)* $\mathcal{L}$ *of* LM *is obtained from* $\bigcup_{\hat{d} \in \text{SE}}\Psi_{\hat{d}}$, $\mathbf{1},\top,\bot,\rightarrow,\wedge,*,\vee$ *and* !.

A mapping f from L_s to L is defined by:

1) for any $p \in \Psi$, $f\left(\left[\hat{d}\right]p\right) := p_{\hat{d}} \in \Psi_{\hat{d}}$;

2) $f\left(\left[\hat{d}\right]\#\right) := \#$ where $\# \in \{1,\top,\bot\}$;

3) $f\left(\left[\hat{d}\right](\alpha\#\beta)\right) := f\left(\left[\hat{d}\right]\alpha\right)\#f\left(\left[\hat{d}\right]\beta\right)$ where

$\# \in \{\rightarrow,\wedge,*,\vee\}$;

4) $f\left(\left[\hat{d}\right](!\alpha)\right) := !f\left(\left[\hat{d}\right]\alpha\right)$;

5) $f\left(\left[\hat{d}\right][b;c]\alpha\right) := f\left(\left[\hat{d}\right][b][c]\alpha\right)$.

Let Γ be a set of formulas in $\mathcal{L}_s$. Then, an expression $f(\Gamma)$ means the result of replacing every occurrence of a formula α in Γ by an occurrence of $f(\alpha)$.

Theorem 3.3. (Embedding) *Let* Γ *be a multiset of formulas in* $\mathcal{L}_s$, γ *be a formula in* $\mathcal{L}_s$, *and f be the mapping defined in Definition* 3.2. *Then:*

$$\text{LCM} \vdash \Gamma \Rightarrow \gamma \text{ iff } \text{LM} \vdash f(\Gamma) \Rightarrow f(\gamma).$$

Proof. • ($\Rightarrow$): By induction on the proofs P of $\Gamma \Rightarrow \gamma$ in LCM. We distinguish the cases according to the last inference of P. We show some cases.

Case $\left(\left[\hat{d}\right]p \Rightarrow \left[\hat{d}\right]p\right)$: The last inference of P is of

the form: $\left[\hat{d}\right]p \Rightarrow \left[\hat{d}\right]p$. Since $f\left(\left[\hat{d}\right]p\right) = p_{\hat{d}}$ by the

definition of f, we obtain the required fact

$$\text{LM} \vdash f\left(\left[\hat{d}\right]p\right) \Rightarrow f\left(\left[\hat{d}\right]p\right) .$$

Case (mixture): The last inference of P is of the form:

$$\frac{\Gamma_1,\Delta \Rightarrow \gamma \quad \Gamma_2,\Delta \Rightarrow \gamma}{\Gamma_1,\Gamma_2,\Delta \Rightarrow \gamma}\left(\text{mixture}\right)$$

By induction hypothesis, we have
$\mathrm{LM} \vdash f(\Gamma_1), f(\Delta) \Rightarrow f(\gamma)$ and
$\mathrm{LM} \vdash f(\Gamma_2), f(\Delta) \Rightarrow f(\gamma)$. Then, we obtain the required fact:

$$\frac{f(\Gamma_1), f(\Delta) \Rightarrow f(\gamma) \quad f(\Gamma_2), f(\Delta) \Rightarrow f(\gamma)}{f(\Gamma_1), f(\Gamma_2), f(\Delta) \Rightarrow f(\gamma)}$$

$(\text{mixture}^\star)$

Case ($\vee$ right1): The last inference of P is of the form:

$$\frac{\Gamma \Rightarrow [\hat{d}]\alpha}{\Gamma \Rightarrow [\hat{d}](\alpha \vee \beta)} (\vee\text{right1})$$

By induction hypothesis, we have
$\mathrm{LM} \vdash f(\Gamma) \Rightarrow f([\hat{d}]\alpha)$. Then, we obtain the required

fact:

$$\frac{f(\Gamma) \Rightarrow f([\hat{d}]\alpha)}{f(\Gamma) \Rightarrow f([\hat{d}]\alpha) \vee f([\hat{d}]\beta)} (\vee\text{right1}^\star)$$

where $f([\hat{d}]\alpha) \vee f([\hat{d}]\beta)$ coincides with
$f([\hat{d}](\alpha \vee \beta))$ by the definition of f.

Case (; left): The last inference of P is of the form:

$$\frac{[\hat{d}][b][c]\alpha, \Gamma \Rightarrow \gamma}{[\hat{d}][b;c]\alpha, \Gamma \Rightarrow \gamma} (;\text{left})$$

By induction hypothesis, we have
$\mathrm{LM} \vdash f([\hat{d}][b][c]\alpha), f(\Gamma) \Rightarrow f(\gamma)$. Then, we obtain

the required fact, since $f([\hat{d}][b][c]\alpha)$ coincides with

$f([\hat{d}][b;c]\alpha)$ by the definition of f.

$\bullet (\Leftarrow)$: By induction on the proofs Q of $f(\Gamma) \Rightarrow f(\gamma)$ in LM. We distinguish the cases according to the last inference of Q. We show some cases.

Case $(\to \text{left}^\star)$: The last inference of Q is of the form:

$$\frac{f(\Gamma) \Rightarrow f([\hat{d}]\alpha) \quad f([\hat{d}]\beta), f(\Sigma) \Rightarrow f(\gamma)}{f([\hat{d}](\alpha \to \beta)), f(\Gamma), f(\Sigma) \Rightarrow f(\gamma)}$$

$(\to\text{left}^\star)$

where $f([\hat{d}](\alpha \to \beta))$ coincides with

$f([\hat{d}]\alpha) \to f([\hat{d}]\beta)$ by the definition of f. By induc-

tion hypothesis, we have $\mathrm{LCM} \vdash \Gamma \Rightarrow [\hat{d}]\alpha$ and

$\mathrm{LCM} \vdash [\hat{d}]\beta, \Sigma \Rightarrow \gamma$. Then, we obtain the required fact:

$$\frac{\Gamma \Rightarrow [\hat{d}]\alpha \quad [\hat{d}]\beta, \Sigma \Rightarrow \gamma}{[\hat{d}](\alpha \to \beta), \Gamma, \Sigma \Rightarrow \gamma} (\to\text{left})$$

Case (cut): The last inference of Q is of the form:

$$\frac{f(\Gamma) \Rightarrow \beta \quad \beta, f(\Sigma) \Rightarrow f(\gamma)}{f(\Gamma), f(\Sigma) \Rightarrow f(\gamma)} (\text{cut})$$

Since β is in $\mathcal{L}$, we can obtain $\beta = f(\beta)$ by induction on β. Then, by induction hypothesis, we have $\mathrm{LCM} \vdash \Gamma \Rightarrow \beta$ and $\mathrm{LCM} \vdash \beta, \Sigma \Rightarrow \gamma$. We then obtain the required fact $\mathrm{LCM} \vdash \Gamma, \Sigma \Rightarrow \gamma$ by using (cut) in LCM. ■

Theorem 3.4. (Cut-elimination) *The rule* (cut) *is admissible in cut-free* LCM.

Proof. We have the following modified statements of Theorem 3.3:

1) if $\mathrm{LCM} \vdash \Gamma \Rightarrow \gamma$, then $\mathrm{LM} \vdash f(\Gamma) \Rightarrow f(\gamma)$;

2) if $\mathrm{LM} - (\text{cut}) \vdash f(\Gamma) \Rightarrow f(\gamma)$, then

$\mathrm{LM} - (\text{cut}) \vdash \Gamma \Rightarrow \gamma$.

To show the second statement, we do not need to prove the case for (cut) as in Theorem 3.3.

We now prove the cut-elimination theorem for LCM as follows. Suppose $\mathrm{LCM} \vdash \Gamma \Rightarrow \gamma$. Then, we have $\mathrm{LM} \vdash f(\Gamma) \Rightarrow f(\gamma)$ by the modified statement 1) of Theorem 3.3, and hence $\mathrm{LM} - (\text{cut}) \vdash f(\Gamma) \Rightarrow f(\gamma)$ by the cut-elimination theorem for LM. By the modified statement 2) of Theorem 3.3, we obtain $\mathrm{LCM} - (\text{cut}) \vdash \Gamma \Rightarrow \gamma$. ■

Corollary 3.5. (Consistency) LCM *is consistent, i.e., the empty sequent* $\Rightarrow$ *is not provable in cut-free* LCM.

In the following, we show that the !-free fragment LCM^- of LCM is decidable. Before to show the decidability of LCM^-, we mention that LCM is undecidable. ILL is known to be undecidable. The proof of the undecidability of ILL is carried out by encoding Minsky machine. LCM can encode Minsky machine in the same way as in ILL, since LCM is an extension of ILL.

Definition 3.6. LCM^- *is obtained from* LCM *by deleting* {(!left), (!right), (!co), (!we)}, *i.e.,* LCM^- *is the* !*-free fragment of* LCM.

Definition 3.7. LM^- *is obtained from* LCM^- *by deleting* {(;left), (;right)} *and all the expressions as* $[\hat{d}]$ *appearing in the initial sequents and the logical inference rules.*

Theorem 3.8. (Decidability) LCM^- *is decidable.*

Proof. The provability of LCM^- can be transformed into that of LM^- by the restriction of Theorem 3.3. Since LM^- is decidable, LCM^- is also decidable. ■

4. Conclusions

In this paper, the logic LCM of context-mixture, which

can suitably express context information in cloud computing environments, was introduced. The cut-elimination and embedding theorems for LCM were proved, and the !-free fragment of LCM was shown to be decidable. LCM is based on an extended resource-sensitive (intuitionistic linear) logic with both the context-mixture rule (mixture) and the sequence modal operator [*b*]. The rule (mixture) of LCM can suitably represent a mechanism for merging formulas with context information, and the operator [*b*] of LCM can represent context information. A concrete logical foundation of reasoning about context information in cloud computing environments was thus obtained in this paper. Some technical remarks on LCM and some related works on context-aware modeling are addressed in the rest of this paper.

It is remarked that the sequence modal operator in LCM can be adapted to a wide range of non-classical logics. An extended intuitionistic linear logic with the sequence modal operator but without the context-mixture rule was shown to be useful for describing secure password authentication protocols [4]. An extended *full computation-tree logic* with the sequence modal operator was shown to be applicable to certain ontological descriptions [3]. An extended *linear-time temporal logic* with the sequence modal operator was shown to be useful for specifying some time-dependent secure authentication systems [22,23]. The sequence modal operator may be applicable to other useful non-classical logics, e.g., some extended linear logics [24,25].

The present paper was intended to provide a logical justification of context-aware cloud computing service models (such as a flowable service model) in cloud computing environments. We now give a survey of such context-aware model approaches. Context is used to challenge various issues in cloud and ubiquitous environments. Many context models have been proposed and developed: A key-value model, a markup model, an object-oriented model, and an ontology-based model (see [26] for a survey). Since location is one of the most typical context information, the location context involves special models: Geometric models, symbolic models and hybrid models. Wohltorf *et al*. [27] introduced a context-awareness module which combines three sub-modules: The location-based service module, the personalization module and the device and network independence module. This work introduced an agent-based service-ware framework to assist service providers in developing innovative services. Gu *et al*. [28] proposed an ontology-based context model which is based on the OWL inside the SOCAM (Service-Oriented Context-Aware Middleware) architecture. Coutaz *et al*. [19] proposed a conceptual framework for context-aware computing, including ontological and architectural foundations. In this method,

the context is modeled as a directed state graph, where the nodes denote contexts and the edges denote the conditions for changes in contexts. Macedo *et al*. [29] developed a distributed information repository for automatic context-aware MANETs in order to adapt the multimedia context-rich application into service computing. Feug *et al*. [30] conceived a shared situation awareness model that supplies each user with agents for focused and customized decision support according to the user's context.

REFERENCES

[1] J.-Y. Girard, "Linear Logic," *Theoretical Computer Science*, Vol. 50, No. 1, 1987, pp. 1-102. doi:10.1016/0304-3975(87)90045-4

[2] A. S. Troelstra, "Lectures on Linear Logic," CSLI Lecture Notes, CSLI, Stanford, 1992.

[3] N. Kamide and K. Kaneiwa, "Extended Full Computation-Tree Logic with Sequence Modal Operator: Representing Hierarchical Tree Structures," *Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence* (AI'09), *Lecture Notes in Artificial Intelligence* 5866, Melbourne, 1-4 December 2009, pp. 485-494.

[4] N. Kamide and K. Kaneiwa, "Resource-Sensitive Reasoning with Sequential Information," *Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence* (AI'10), *Lecture Notes in Artificial Intelligence* 6464, Adelaide, 7-10 December 2010, pp. 22-31.

[5] Y. Zhu, R. Y. Shtykh and Q. Jin, "Provision of Flowable Services in Cloud Computing Environments," *Proceedings of the 5th International Conference on Future Information Technology* (FutureTech'2010), Busan, 21-23 May 2010, pp. 1-6.

[6] Y. Zhu, R. Y. Shtykh and Q. Jin, "A Human-Centric Framework for Context-Aware Flowable Services in Cloud Computing Environments," *Information Sciences*, 2012, in Press.

[7] I. Foster, Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," *Grid Computing Environments Workshop* (GCE'08), Austin, 16 November 2008, pp. 1-10. doi:10.1109/GCE.2008.4738445

[8] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[9] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles, "Towards Better Understanding of Context and Context-Awareness," *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing* (HUC'99), *Lecture Notes in Computer Science* 1707, Karlsruhe, 27-29 September 1999, pp. 304-307.

[10] R. Y. Shtykh, Y. Zhu and Q. Jin, "A Context-Aware Framework for Flowable Services," *Proceedings of the 3rd International Conference on Multimedia and Ubiquitous Engineering* (MUE'09), Qingdao, 4-6 June 2009, pp.

251- 256.

[11] Y. Zhu, R. Y. Shtykh, Q. Jin and J. Ma, "A Flowable Service Model for Seamless Integration of Services," *Proceedings of the* 4*th International Conference on Advances in Computer Science and Engineering* (ACSE'2009), Phuket, 16-18 March 2009, pp. 199-204.

[12] Y. Zhu, R. Y. Shtykh and Q. Jin, "Harnessing User Contexts to Enable Flowable Services Model," *Proceedings of the* 3*rd International Conference on Human-Centric Computing* (HumanCom-10), Cebu, 11-13 August, 2010, pp. 1-6.

[13] Y. Zhu and Q. Jin, "An Adaptively Emerging Mechanism for Context-Aware Service Selections Regulated by Feedback Distributions," *Human-Centric Computing and Information Sciences*, 2012, in Press.

[14] Y. Zhu and Q. Jin, "An Adaptively Emerging Mechanism for Selection of Ambient Services," *Proceedings of the* 2012 *FTRA International Conference on Advanced IT, Engineering and Management* (FTRA AIM'2012), Seoul, 6-8 February 2012, pp. 157-158.

[15] N. Kamide, "Linear Logics with Communication-Merge," *Journal of Logic and Computation*, Vol. 15, No. 1, 2006, pp. 3-20. doi:10.1093/logcom/exh029

[16] M. Ohnishi and K. Matsumoto, "A System for Strict Implication," *Annals of the Japan Association for Philosophy of Science*, Vol. 2, No. 4, 1964, pp. 183-188.

[17] A. R. Anderson and N. D. Belnap Jr., "Entailment: The Logic of Relevance and Necessity, Volume 1," Princeton University Press, Princeton, 1975.

[18] N. Kamide, "Substructural Logic with Mingle," *Journal of Logic*, *Language and Information*, Vol. 11, No. 2, 2002, pp. 227-249. doi:10.1023/A:1017586008091

[19] J. Coutaz, J. L. Crowley, S. Dobson and D. Garlan, "Context Is Key," *Communications of the ACM*, Vol. 48, No. 3, 2005, pp. 49-53. doi:10.1145/1047671.1047703

[20] A. J. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon and W. Carithers, "Efficient Resource Management for Cloud Computing Environments," *Proceedings of Green Computing Conference*, Chicago, 15-18 August 2010, pp. 357-364.

doi:10.1109/GREENCOMP.2010.5598294

[21] H. Wansing, "The Logic of Information Structures," *Lecture Notes in Artificial Intelligence* 681, 1993, pp. 1-163.

[22] N. Kamide, "A Proof System for Temporal Reasoning with Sequential Information," *Proceedings of the* 20*th Brazilian Symposium on Artificial Intelligence* (SBIA 2010), *Lecture Notes in Artificial Intelligence* 6404, São Bernardo do Campo, 23-28 October 2010, pp. 283-292.

[23] K. Kaneiwa and N. Kamide, "Sequence-Indexed Linear-Time Temporal Logic: Proof System and Application," *Applied Artificial Intelligence*, Vol. 24, No. 10, 2010, pp. 896-913. doi:10.1080/08839514.2010.514231

[24] N. Kamide, "Combining Soft Linear Logic and Spatio-Temporal Operators," *Journal of Logic and Computation*, Vol. 14, No. 5, 2004, pp. 625-650. doi:10.1093/logcom/14.5.625

[25] N. Kamide, "Linear and Affine Logics with Temporal, Spatial and Epistemic Operators," *Theoretical Computer Science*, Vol. 353, No. 1-3, 2006, pp. 165-207.

[26] C. Hoareau and I. Satoh, "Modeling and Processing Information for Context-Aware Computing: A Survey," *New Generation Computing*, Vol. 27, No. 3, 2009, pp. 177-196. doi:10.1007/s00354-009-0060-5

[27] J. Wohltorf, R. Cissee and A. Rieger, "BerlinTainment: An Agent-Based Context-Aware Environment Planning System," *IEEE Communications Magazine*, Vol. 43, No. 6, 2005, pp. 102-109. doi:10.1109/MCOM.2005.1452837

[28] T. Gu, H. K. Pung and D. Q. Zhang, "A Service-Oriented Middleware for Building Context-Aware Services," *Journal of Network and Computer Applications*, Vol. 28, No. 1, 2005, pp. 1-18. doi:10.1016/j.jnca.2004.06.002

[29] D. Macedo, A. Dos Santos, J. M. S. Nogueira and G. Pujolle, "A Distributed Information Repository for Autonomic Context-Aware MANETs," *IEEE Transactions on Network and Service Management*, Vol. 6, No. 1, 2009, pp. 45-55. doi:10.1109/TNSM.2009.090304

[30] Y. Feng, T. Teng and A. Tan, "Modeling Situation Awareness for Context-Aware Decision Support," *Expert System with Applications*, Vol. 36, No. 1, 2009, pp. 455-463. doi:10.1016/j.eswa.2007.09.061