



## CSP-Based Formulation of Multi-Agent Communication for a First-Order Agent Theory

メタデータ	言語: eng 出版者: 公開日: 2010-04-02 キーワード (Ja): キーワード (En): 作成者: Izumi, Noriaki, Takamatsu, Shinobu, Kise, Koichi, Fukunaga, Kunio メールアドレス: 所属:
URL	<a href="https://doi.org/10.24729/00008304">https://doi.org/10.24729/00008304</a>

## CSP-Based Formulation of Multi-Agent Communication for a First-Order Agent Theory

Noriaki IZUMI\*, Shinobu TAKAMATSU\*\*, Koichi KISE\*\*\* and Kunio FUKUNAGA\*\*\*

(Received June 28, 1996)

For the integration of work on multi-agent systems, we propose a communication model of multi-agent systems which unifies CSP (Communicating Sequential Processes)-based formulation of the agents' communication and a first-order logical formalization based on meta logics. This model captures various concepts of multi-agent systems: the concepts of the agents' environment such as time, actions, conditional and causality, the concepts of mental attitudes such as knowledge, plans, goals and intentions, and the concepts of agents' interaction such as speech acts, negotiations and collaboration. In this paper, modal logical formulas of those concepts are rebuilt into a first-order theory with three layers, as an executable specification for a proof system on first-order logic. This first-order theory can be taken as a agent oriented programming language which gives a general framework for integration of agent theories as specifications, agent architectures as problem solving systems, and agent languages for programming.

### 1. Introduction

A multi-agent system is a distributed computing system composed of a number of intelligent agents. It solves complex problems in a cooperative way even if an agent cannot solely solve.

Much work has been done on multi-agent systems in various fields such as theories, architectures and languages<sup>10)</sup>. Most theories for representing and reasoning about the properties of agents are based on modal logics with possible worlds semantics<sup>2),6),8)</sup>. These theories allow us to describe complex behavior of agents and mental attitudes. Modal logics, however, capture static properties in an abstract sense. Because of less efficiency of proof mechanism, it is difficult to model multi-agent systems where agents interact in their dynamic environments. For the construction of agent systems that satisfy the properties specified in agent theories, various agent architectures have been proposed. Although these systems can be applied to solve practical problems, they have no clear correspondence with the logic used to express the specification<sup>11)</sup>. As noted above, there is a gap

between agent theories and architectures, that is, static natures and dynamic systems. To bridge this gap, a lot of agent languages including some architectures have been developed. Those languages allow us to program multi-agent systems in terms of the concepts in agent theories. There is, however, no unified model that clarifies a correspondence of agent theories, architectures and languages.

For the integration of work on agent theories, architectures and languages, this paper proposes a model of multi-agent systems unifying a formulation of communication based on CSP<sup>5)</sup> and a first-order formalization of multi-modal logics<sup>9)</sup>.

In this paper, modal logics of agents' environment and mental attitudes are rebuilt into a first-order theory with three layers as executable specifications on the bases of meta logics. Furthermore, we propose a description method on agents' interaction, such as speech acts, negotiation and collaboration, based on the semantics of the calculus in CSP. This formalization gives a general framework used to analyze, specify, design, and implement multi-agent systems.

### 2. Formalization of environment models

#### 2.1 Object-level descriptions

Agents' environment consists of various concepts such as states, actions, state transitions, time and

\* Graduate Student, Department of Computer and Systems Sciences, College of Engineering.

\*\* Department of Information Systems Engineering, Faculty of Engineering, Osaka Sangyo University.

\*\*\* Department of Computer and Systems Sciences, College of Engineering.

Table 1 The formation rules of modal logical formulas

---


$$\begin{aligned}
\langle \text{term} \rangle &::= \langle \text{individual constant} \rangle \mid \langle \text{individual variable} \rangle \mid \langle \text{function form} \rangle \\
\langle \text{function form} \rangle &::= \langle \text{n-ary function symbol} \rangle (\langle \text{term}_1 \rangle, \langle \text{term}_2 \rangle, \dots, \langle \text{term}_n \rangle) \\
\langle \text{state expression} \rangle &::= \langle \text{n-ary state predicate symbol} \rangle (\langle \text{term}_1 \rangle, \langle \text{term}_2 \rangle, \dots, \langle \text{term}_n \rangle) \mid \dots \\
\langle \text{action formula} \rangle &::= \langle \text{n-ary action predicate symbol} \rangle (\langle \text{term}_1 \rangle, \langle \text{term}_2 \rangle, \dots, \langle \text{term}_n \rangle) \mid \dots \\
\langle \text{formula} \rangle &::= \langle \text{state expression} \rangle \mid \neg \langle \text{formula} \rangle \mid \langle \text{formula} \rangle \& \langle \text{formula} \rangle \mid \\
&\quad \langle \text{formula} \rangle \vee \langle \text{formula} \rangle \mid \langle \text{formula} \rangle \rightarrow \langle \text{formula} \rangle \mid \dots \\
&\quad \text{FUTsome}(\langle \text{formula} \rangle) \mid \text{UNTIL}(\langle \text{formula} \rangle, \langle \text{formula} \rangle) \mid \dots \\
&\quad \text{OCCUR}(\langle \text{action formula} \rangle) \mid \text{CAUSE}(\langle \text{action formula} \rangle, \langle \text{formula} \rangle) \mid \dots \\
&\quad \text{COND2}(\langle \text{formula} \rangle, \langle \text{action formula} \rangle, \langle \text{formula} \rangle) \mid \dots
\end{aligned}$$


---

Table 2 The transformation rules on  $T('A', s)$ 


---

(a) basic expressions

$$\begin{aligned}
T('p(t_1, t_2, \dots, t_n)', s) &\equiv p(t'_1, t'_2, \dots, t'_n, s) \\
t'_i &= t_i: t_i \text{ is a individual constant or a individual variable} \\
t'_i &= f(\dots, s): t_i \text{ is a function expression } f(\dots) \\
T('A \& B', s) &\equiv T('A', s) \& T('B', s) \\
T('A \vee B', s) &\equiv T('A', s) \vee T('B', s) \\
T('A \rightarrow B', s) &\equiv T('A', s) \rightarrow T('B', s) \\
T('¬A', s) &\equiv ¬T('A', s) \\
T('∀xA', s) &\equiv ∀xT('A', s) \\
T('∃xA', s) &\equiv ∃xT('A', s) \\
&\dots
\end{aligned}$$

(b) temporal expressions

$$\begin{aligned}
T('FUTall(A)', s_0) &\equiv \forall s_1 \{ \leq(s_0, s_1) \rightarrow T('A', s_1) \} \\
T('FUTsome(A)', s_0) &\equiv \exists s_1 \{ \leq(s_0, s_1) \& T('A', s_1) \} \\
T('UNTIL(A, B)', s_0) &\equiv \exists s_1 \{ \leq(s_0, s_1) \& T('B', s_1) \& \forall s_2 \{ \leq(s_0, s_2) \& <(s_2, s_1) \rightarrow T('A', s_2) \} \} \\
T('BEFORE(A, B)', s_0) &\equiv \forall s_1 \{ <(s_0, s_1) \& T('B', s_1) \rightarrow \exists s_2 \{ <(s_0, s_2) \& <(s_2, s_1) \& T('A', s_2) \} \} \\
&\dots
\end{aligned}$$

(c) action expressions

$$\begin{aligned}
T('OCCUR(act)', s_0) &\equiv \exists s_1 T('OCCUR(act)', s_0, s_1) \\
T('OCCUR(act)', s_0, s_1) &\equiv E(act, s_0, s_1), \text{ act is a basic action} \\
T('OCCUR(IF p THEN act_1 ELSE act_2)', s_0, s_1) &\equiv \{ T('p', s_0) \& T('OCCUR(act_1)', s_0, s_1) \} \\
&\quad \vee \{ \neg T('p', s_0) \& T('OCCUR(act_2)', s_0, s_1) \} \\
T('OCCUR(act_1 THEN act_2)', s_0, s_2) &\equiv \exists s_1 \{ T('OCCUR(act_1)', s_0, s_1) \& T('OCCUR(act_2)', s_1, s_2) \} \\
&\dots
\end{aligned}$$

(d) causal and conditional expressions

$$\begin{aligned}
T('CAUSE(act, A)', s_0) &\equiv \exists s_1 \{ T('OCCUR(act)', s_0, s_1) \& T('A', s_1) \} \\
T('COND1(act, A)', s_0) &\equiv \forall s_1 \{ T('OCCUR(act)', s_0, s_1) \rightarrow T('A', s_1) \} \\
T('COND2(A, act, B)', s_0) &\equiv \forall s_1 \{ T('A', s_0) \& T('OCCUR(act)', s_0, s_1) \rightarrow T('B', s_1) \} \\
&\dots
\end{aligned}$$


---

causal relations.

To express those concepts, we employ a multi-modal predicate logic (called 'modal logic' for short) as a description language, which extends first-order logic

by adding modal operators for time, actions, conditional and causality. Formation rules of modal logical formulas are given in Table 1. In this paper, a level of the description regarding an environment of

Table 3 The definitions of mental actions

---

$T('OCCUR^M(m-act)', s_1, \Psi_1)$
$\equiv \exists \Psi_2 \exists s_2 T(\Psi_1, s_1, 'OCCUR^M(m-act)', s_2, \Psi_2)$
$T(\Psi_1, s_1, 'OCCUR^M(assume(A))', s_2, \Psi_2)$
$\equiv R(\Psi_1, s_1, 'A', s_2, \Psi_2)$
$T(\Psi_1, s_1, 'OCCUR^M(verify(A))', s_2, \Psi_2)$
$\equiv T('A', s_1, \Psi_1) \& R(\Psi_1, s_1, 'A', s_2, \Psi_2)$
$T(\Psi_1, s_1, 'OCCUR^M(execute(act))', s_3, \Psi_3)$
$\equiv R_{sense}(\Psi_1, s_1, 'Precond(act)', s_2, \Psi_2)$
$\& R(\Psi_2, s_2, 'OCCUR^M(act))', s_3, \Psi_3)$
$T(\Psi_1, s_1, 'OCCUR^M(execute(act;acts))', s_3, \Psi_3)$
$\equiv T(\Psi_1, s_1, 'OCCUR^M(execute(act))', s_2, \Psi_2)$
$\& T(\Psi_2, s_2, 'OCCUR^M(execute(acts))', s_3, \Psi_3)$
$T(\Psi_1, s_1, 'OCCUR^M(plan(goal, act-sq))', s_2, \Psi_2)$
$\equiv T(\Psi_1, s_1, 'OCCUR^M(verify(COND1(act-sq, goal)))', s_2, \Psi_2)$
...

---

agents is called the object-level.

To integrate concepts expressed by modal logical formulas, we formalize the structure of agents' environment based on first-order logic. We introduce the first-order predicate:  $T('A', s)$ , which represents that a modal logical formula  $A$  is true at a state  $s$ . Table 2 lists the transformation rules of  $T('A', s)$  on basic, temporal, action, causal and conditional expressions.

## 2.2 Reasoning system based on first-order logic

In this paper, we take agent's knowledge of his environment as a set of first-order logical formulas called a situation-description. We set a truth-value of logical formulas based on the provability in a situation-description and a constraint-description given by a set of rules and meta-level descriptions.

We represent that a modal logical formula  $A$  is true at a state  $s$  in a situation-description  $\Psi$  and a constraint-description  $\Upsilon$ , by the following provability relation:

$$\Upsilon \cup \Psi \vdash T('A', s) \quad (1)$$

This relation means that  $T('A', s)$  is derivable from  $\Psi$  and  $\Upsilon$  by the first-order logical inference of resolution, paramodulation and default inference<sup>4)</sup>. The default inference is used for the reasoning about a continuation of a state independent of action occurrences, and is implemented as an inference procedure.

## 2.3 Dynamic interpretation of logical formulas

To interpret dynamically the meaning of a logical formula, we propose a mechanism which revises a situation-description responding to a state transition caused by an occurrence of an action. For example, a procedure *Revise*, which revises a situation-description

$\Psi_1$  into  $\Psi_2$  for the occurrence of an action  $act$  between states  $s_1$  and  $s_2$ , is defined as follows:

$$\begin{aligned} & Revise(\Upsilon, \Psi_1, s_2, OCCUR(act), s_2, \Psi_2) \\ & \equiv \Psi_2 = (\Psi_1 \cup \Psi' \cup \Psi'') \end{aligned} \quad (2)$$

where

$$\begin{aligned} \Psi' = & \{T('OCCUR(act)', s_1, s_2)\} \cup \\ & \{T('a', s_2) \mid \Upsilon \cup \Psi_1 \vdash T('b', s_1) \text{ and} \\ & \text{Action rule:} \\ & T('b', s_1) \& T('OCCUR(act)', s_1, s_2) \\ & \rightarrow T('a', s_2) \in \Upsilon \cup \Psi_1\} \\ \Psi'' = & \{T('g', s_2) \mid \\ & T('g', s_1) \in \Psi_1 \text{ and} \\ & T('g', s_2) \text{ is derivable from } \Psi_1 \cup \Psi' \\ & \text{by the default inference procedure}\} \end{aligned}$$

## 3. Formalization of an agent model

### 3.1 Knowledge-level descriptions

We consider a situation-description  $\Psi$  as an individual constant of a meta-level. A situation-description  $\Psi$ , which shows an environment of an agent, can be taken to be agent's knowledge of his environment in his cognition process. On the basis of reflection<sup>1)</sup>, we introduce meta-level expressions of first-order predicates  $T$  and  $R$  which correspond to the procedures of proofs and revisions of situation-descriptions respectively, as follows:

$$\Upsilon \vdash T('A', s, \Psi) \Leftrightarrow \Upsilon \cup \Psi \vdash T('A', s) \quad (3)$$

$$\begin{aligned} \Upsilon \vdash R(\Psi_1, s_1, 'A', s_2, \Psi_2) \\ \Leftrightarrow Revise(\Upsilon, \Psi_1, s_1, A, s_2, \Psi_2) \end{aligned} \quad (4)$$

The formula of the predicate  $T$  is an expression of

Table 4 The definitions of the knowledge level revision

---


$$\begin{aligned}
& M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(assume(A)), s_2, \langle \Phi_2, \Psi_2 \rangle) \\
& \equiv M-T('R(\Psi_1, s_1, 'A', s_2, \Psi_2)', \Phi_1) \\
& \quad \& \Phi_2 = \Phi_1 \cup \{R(\Psi_1, s_1, 'A', s_2, \Psi_2)\} \\
& M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(verify(A)), s_2, \langle \Phi_2, \Psi_2 \rangle) \\
& \equiv M-T('T('A', s_1, \Psi_1)', \Phi_1) \\
& \quad \& M-T('R(\Psi_1, s_1, 'A', s_2, \Psi_2)', \Phi_1) \\
& \quad \& \Phi_2 = \Phi_1 \cup \{T('A', s_1, \Psi_1), R(\Psi_1, s_1, 'A', s_2, \Psi_2)\} \\
& M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(execute(act)), s_3, \langle \Phi_3, \Psi_3 \rangle) \\
& \equiv M-T('R_{sense}(\Psi_1, s_1, 'Precond(act)', s_2, \Psi_2)', \Phi_1) \\
& \quad \& M-T('R(\Psi_2, s_2, 'OCCUR(act)', s_3, \Psi_3)', \Phi_1) \\
& \quad \& \Phi_3 = \Phi_1 \cup \{R_{sense}(\Psi_1, s_1, 'Precond(act)', s_2, \Psi_2), \\
& \quad \quad R(\Psi_2, s_2, 'OCCUR(act)', s_3, \Psi_3)\} \\
& M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(plan(goal, act-sq)), s_2, \langle \Phi_2, \Psi_2 \rangle) \\
& \equiv M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(verify(COND1(act-sq, goal))), s_2, \langle \Phi_2, \Psi_2 \rangle) \\
& \dots
\end{aligned}$$


---

Table 5 The definitions of the intention-level revision

---


$$\begin{aligned}
& MM-R(\langle \Omega_1, \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(achieve(A)), s_2, \langle \Omega_2, \Phi_2, \Psi_2 \rangle) \\
& \equiv MM-R(\langle \Omega_1, \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(accomplish(A)), s_2, \langle \Omega_2, \Phi_2, \Psi_2 \rangle) \\
& \quad \vee MM-R(\langle \Omega_1, \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(request(set-goal(A))), s_2, \langle \Omega_2, \Phi_2, \Psi_2 \rangle) \\
& MM-R(\langle \Omega_1, \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(accomplish(A)), s_3, \langle \Omega_3, \Phi_3, \Psi_3 \rangle) \\
& \equiv \{\neg MM-T(' \exists act INTEND(act, s_1, \langle \Phi_1, \Psi_1 \rangle', \Omega_1) \\
& \quad \& MM-T('M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(plan(A, act-sq)), s_2, \langle \Phi_2, \Psi_2 \rangle)', \Omega_1) \\
& \quad \& \Omega_2 = \Omega_1 \cup \{INTEND(act-sq, s_1, \langle \Phi_2, \Psi_2 \rangle), \\
& \quad \quad M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(plan(A, act-sq)), s_2, \langle \Phi_2, \Psi_2 \rangle)\} \\
& \quad \& MM-R(\langle \Omega_2, \Phi_2, \Psi_2 \rangle, s_2, OCCUR^M(accomplish(A)), s_3, \langle \Omega_3, \Phi_3, \Psi_3 \rangle) \\
& \vee MM-T(' \exists act INTEND(act-sq, s_1, \langle \Phi_1, \Psi_1 \rangle', \Omega_1) \\
& \quad \& MM-T('M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(execute(first(act-sq))), s_2, \langle \Phi_2, \Psi_2 \rangle)', \Omega_1) \\
& \quad \& \Omega_2 = \Omega_1 \cup \{M-R(\langle \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(execute(first(act-sq))), s_2, \langle \Phi_2, \Psi_2 \rangle), \\
& \quad \quad INTEND(tail(act-sq, s_2, \langle \Phi_2, \Psi_2 \rangle)\} \\
& \quad \& MM-R(\langle \Omega_2, \Phi_2, \Psi_2 \rangle, s_2, OCCUR^M(accomplish(A)), s_3, \langle \Omega_3, \Phi_3, \Psi_3 \rangle)\} \\
& \vee \{MM-R(\langle \Omega_1, \Phi_1, \Psi_1 \rangle, s_1, OCCUR^M(verify(A)), s_3, \langle \Omega_3, \Phi_3, \Psi_3 \rangle)\} \\
& \dots
\end{aligned}$$


---

agent's mental state, and the formula of the predicate  $R$  is an expression of an agent's mental state transition. In this paper, a level of expressions using the predicates  $T$  and  $R$  is called a knowledge-level.

An occurrence of agents' mental action  $m-act$ , which causes a change of his mental state, can be expressed by  $T$  ( $OCCUR^M(m-act)$ ,  $s, \Psi$ ) defined in Table 3. In the definition of  $execute$ ,  $R_{sense}(\Psi_1, s_1, 'Precond(act)', s_2, \Psi_2)$  represents that an agent senses and makes sure of "Precond (act)" at a state  $s_1$  in a situation-description  $\Psi_1$ , and then, revises the state and the situation-description into  $s_2$  and  $\Psi_2$ .

Furthermore, to interpret dynamically the meanings of knowledge-level formulas, we define revision procedures in knowledge-level, which satisfy the descriptions as specifications on mental actions using the predicates  $T$  and  $R$  in Table 3. For this purpose, a

situation-description on the knowledge-level  $\Phi$  is given, which consists of knowledge-level formulas of proofs and revisions on the object-level. We introduce an expression on the proof procedure for knowledge-level descriptions on the basis of reflection. A first-order predicate  $M-T$  is defined as follows:

$$\mathbf{T}' \vdash M-T(' \psi', \Phi) \Leftrightarrow \mathbf{T}' \cup \Phi \vdash \psi \quad (5)$$

where  $\psi$  is a knowledge-level expression and  $\mathbf{T}'$  is a constraint-description  $\mathbf{T}$  without  $\Phi$ .

A procedure of revising knowledge-level descriptions responding to an occurrence of a mental action  $m-act$  is represented using the following first-order formula:

$$M-R(\langle \Phi, \Psi \rangle, s, OCCUR^M(m-act), s', \langle \Phi', \Psi' \rangle) \quad (6)$$

Table 4 lists the definitions of the knowledge-level revision  $M-R$  using the meta-level predicate  $M-T$ .

Table 6 The syntax of abstract-level descriptions

---

• self action label	
$\langle SAL \rangle ::= assume \mid verify \mid accomplish \mid \dots$	
• communication action label	
$\langle CAL \rangle ::= inform \mid notify \mid set-goal \mid \dots$	
• basic abstract action expression	
$\langle BAAE \rangle ::=$	
$\langle SAL \rangle \downarrow \langle formula \rangle$	... self action
$\mid \langle CAL \rangle ? \langle formula \rangle$	... requested action
$\mid \langle CAL \rangle ! \langle formula \rangle$	... request action
• abstract action expression	
$\langle AAE \rangle ::=$	
$\mid \langle BAAE \rangle \dots$	... basic action
$\mid STOP \dots$	... nil action
$\mid \langle BAAE \rangle \Rightarrow \langle AAE \rangle$	... action prefix
$\mid \langle AAE \rangle [] \langle AAE \rangle$	... nondeterministic
$\mid \langle AAE \rangle \parallel \langle AAE \rangle$	... communication
$\mid \langle AAE \rangle \setminus \langle BAAE \rangle$	... concealment
$\mid \mu X. \langle AAE \rangle$	... recursion
$\mid \dots$	

---

### 3.2 Intention-level descriptions

In the same way as introducing the knowledge level, an intention-level, which is a meta-level of the knowledge-level, can also be formalized constructively using the expressions of  $M-T$  and  $M-R$ . We introduce an intention-level description  $\Omega$  which consists of meta-level expressions of proofs and revisions in the knowledge-level using the predicates  $M-T$  and  $M-R$ . A first-order predicate  $MM-T$  which corresponds to a proof procedure for intention-level descriptions is defined as follows:

$$\mathbb{T}'' \vdash MM-T(\phi, \Omega) \Leftrightarrow \mathbb{T}'' \cup \Omega \vdash \psi \quad (7)$$

where  $\mathbb{T}''$  is a constraint-description  $\mathbb{T}'$  without  $\Omega$ , and  $\psi$  is an intention-level expression. An intention-level expression is a formula of  $M-T$ ,  $M-R$  or a first-order formula:  $INTEND(act, s, \langle \Phi, \Psi \rangle)$  which means that an agent intends to  $act$  at the state  $s$  in the situation  $\langle \Phi, \Psi \rangle$ . We define a meta-meta-level expression  $MM-R$  of the revision procedure on the intention-level. For example, an action *achieve* by which an agent autonomously plans his action and executes it until he achieves his own goal or he requests of others to achieve it is formalized in Table 5.

## 4. Communication models between agents

It is conceivable that communication between agents is based on some protocols defined for types of messages. We formalize models of communication between agents using CSP that is to specify descriptions of communication protocols.

### 4.1 Abstract-level descriptions based on CSP

Communication between agents is conceivable as a higher concept than concepts on the intention-level because agents decide their goals and intentions through agents' interaction like speech acts. In our approach, we describe mental actions of the intention-level by CSP. This description is called an abstract-level description. The syntax of expressions for the abstract-level description is given in Table 6, and inference rules are given in Table 7. In this paper, we aim at formalizing communication just between two agents. It is considered that the abstract-level description of an agent is independent of those of the other agent.

### 4.2 Example

Speech acts can be expressed in an abstract-level description. For example, a series of actions  $M-act_a$ , which shows "if the agent  $a$  is requested  $X$ , he attempts the achievement of  $X$  and informs the reali-

Table 7 The Inference rules on abstract-level descriptions

self action	
$\frac{MM-R(\langle \Omega, \Phi, \Psi \rangle, s, OCCUR^M(\alpha(A)), s', \langle \Omega', \Phi', \Psi' \rangle)}{\alpha \downarrow A \Rightarrow E\{\alpha \downarrow A\}E}$	
transmit action	
$\frac{MM-R(\langle \Omega, \Phi, \Psi \rangle, s, OCCUR^M(request(\alpha(A))), s', \langle \Omega', \Phi', \Psi' \rangle)}{\alpha \downarrow A \Rightarrow E\{\alpha \downarrow A\}E}$	
receive action	
$\frac{MM-R(\langle \Omega, \Phi, \Psi \rangle, s, OCCUR^M(requested(\alpha(A))), s', \langle \Omega', \Phi', \Psi' \rangle)}{\alpha \uparrow A \Rightarrow E\{\alpha \uparrow A\}E}$	
SUM	$\frac{E_j\{\alpha\}E'_j}{\sum_{j \in I} E_j\{\alpha\}E'_j} (j \in I)$
COM <sub>1</sub>	$\frac{E\{\alpha\}E'}{E \parallel F\{\alpha\}E' \parallel F}$
COM <sub>2</sub>	$\frac{F\{\alpha\}F'}{E \parallel F\{\alpha\}E' \parallel F'}$
COM <sub>3</sub>	$\frac{E\{\alpha \downarrow A\}E' \quad F\{\alpha \uparrow X\}F'}{E \parallel F\{comm(\alpha, A)\}E' \parallel F'}$
...	

zation of  $X$ ", is expressed in the following manner:

$$M-act_a = set-goal?X \Rightarrow accomplish \downarrow X \Rightarrow notify! X \quad (8)$$

We express a series of actions  $M-act_b$ , which is a repetition that the agent  $b$  requests  $Y$  or he adds  $Z$  to his knowledge if he is told  $Z$  is true.

$$M-act_b = \mu X. (set-goal!Y \Rightarrow X \\ \parallel notify?Z \Rightarrow assume \downarrow Z \Rightarrow X) \quad (9)$$

In this example, the agent  $b$  wants to achieve  $G$ , in other words, he is going to prove a revision on the intention-level corresponding to the mental action *achive* ( $G$ ).

If abstract-level descriptions of the agent  $a$  and the agent  $b$  are given by " $M-act_a \parallel M-act_b$ ", the agent  $b$  requests the agent  $a$  for the achievement and the agent  $a$  reports the agent  $b$  its success.

## 5. Conclusion

For the integration of various work on agent theories, architectures and languages, it is important to model multi-agent systems in a unified way. From this viewpoint, we have proposed a model of multi-agent systems based on CSP and meta logics. In this paper, descriptions by multi-modal predicate logics are rebuilt into a theory of first-order logic based on meta logics. Those descriptions can be regarded as executable specifications for a proof system of first-order logic. This formalization allows us to program multi-agent systems in terms of logics on agents' environments and their mental conditions. First-order logic gives a general framework for analyzing, specifying, designing, and implementation of multi-agent systems. Formulation of agents' interactions based on the calculation system of CSP gives a communication

model of multi-agent systems which enables us both to describe specifications of speech acts and to define agents' behavior. This model can be taken as to be a model of an operating system for an agent language.

It would also be important to model interactions among many agents, although communication between just two agents is formalized in this paper. We take knowledge regarding each agent's environment as to be fundamentally independent. But in the case that the proposed method is applied to the real world, it becomes required to formalize common knowledge and its revision. These should be addressed in future work.

## References

- 1) Attardi G., Simi M., "Reflection about Reflection", KR & R-91, 1991, pp. 22-31.
- 2) Cohen, P. R. and Levesque, H. J., "Intention is choice with commitment", Artificial Intelligence, vol. 42, pp. 213-261, 1990.
- 3) Davies N., "A First Order Logic of Truth, Knowledge and Belief", LNAI-478, 1990, pp. 170-179.
- 4) Ginsberg, M. L. and Smith, D. E., "Reasoning about Action I, A Possible Worlds Approach", Artificial Intelligence, vol. 35, 1988, pp. 165-195.
- 5) Hoare C. A. R., Communicating Sequential Processes, Prentice-Hall, 1985.
- 6) Rao, A. S., Georgeff, M. P., "Modeling rational agents within a BDI-architecture", Principles of Knowledge Representation and Reasoning (KR-91), pp. 473-484, 1991.
- 7) Shoham Y., "Agent-oriented programming", Artificial Intelligence Vol. 60, 1993, pp. 51-92.
- 8) Singh M. P., "Multiagent Systems, A Theoretical Framework for Intentions, Know-How, and Communications", LNAI-799 Springer-Verlag, 1994.
- 9) Takamatsu, S., Izumi, N., Kise, K. and Fukunaga, K., "First-order Logic Based Situational and Dynamic Interpretation of Natural Language Descriptions in Hardware Design Specifications", Journal of Japanese Society for Artificial Intelligence, vol. 10, 1995, pp. 720-730, in *Japanese*.
- 10) Wooldridge M., Jennings N. R. (Eds.), Intelligent Agents, Proceedings of ECAI-94 Workshop on Agent Theories,

- Architectures, and Languages, LNAI-890, Springer-Verlag, 1995.
- 11) Wooldridge M., "This is My World, The logic of an agent-oriented testbed for DAI", Intelligent Agents, Proceedings of the 1994 Workshop on Agent Theories, Architectures, and Languages, pp. 160-174, 1994.