Framework of Performance Evaluation for Mobile Process Based on Mobile Ambient *

Taolue Chen† Tingting Han Jian Lu
State Key Laboratory of Novel Software Technology,
Nanjing University, Nanjing, Jiangsu, P.R.China 210093

Abstract

Formal methodology for distributed and concurrent system, especially computation system with mobility, is increasingly important both in the theory and the practice. Based on the ***Calculus of Mobile Ambient*** *(MA), a widely studied formal mechanism for mobile computation, this paper focuses on the quantitative analysis of mobile computation system and provides a framework of performance evaluation for it. In details, this paper introduces an enhanced labelled transition system as the system model, and assigns rate to each label through so-called cost function. Based on it, the labelled transition system can be mapped to Continuous Time Markov Chains and thus performance evaluation can be carried out by standard numerical techniques and tools. In some sense, the main work of this paper can be regarded as integrating behavior and performance analysis in a compositional formal framework, which provides some basis of unified formal methodology for the development of mobile system.*

Keywords*: Formal methodology, Software quantity, Process algebra, Mobile ambient, Performance evaluation*

1 Introduction

Formal analysis for distributed and concurrent system is one of the key problems in theoretical computer science. At the same time, it is also very important in practice.

Calculus of Mobile Ambient ([1], MA in short), introduced by Cardelli and Gordon is a formalism which can describe two common aspects of mobility, that is mobile computing and mobile computation ([1]) within a single framework. It is regarded as a suitable candidate for formal model of novel distributed computing paradigms, such as mobile computing and Internet computing and has been one of the hottest topics in the area of mobile process calculi after π-calculus ([3]).

In the traditional research for process calculi, a lot of attention has been paid to the behavior analysis (quality analysis) of systems, such as the liveness and justice of system, and synchronization or communication related problems. However, in practice, the performance of system (quantity analysis) is also very important. Performance evaluation means to describe, to analysis, and to optimize the dynamic, time-dependent behavior of systems ([2]). In our point of view, the performance evaluation should be integrated into the design process from the very beginning, i.e. the behavior specification and performance specification should be integrated into a unified framework, which is in need of support from formal methodology.

To the author's knowledge, there are no reports on performance analysis in the framework of MA. The main work of this paper is to provide a framework of performance analysis for mobile process while leaving the syntax of the calculus unchanged. In details, we introduce an enhanced labelled transition system, and base on it, we convert it into *Continue Time Markov Chain* (CTMC in short) through the notion of cost functions. The performance analysis is achieved through some standard numeral techniques based on CTMC. The advantage of keeping the syntax of calculi unchanged lies in that we can inherit lots of research on behavior analysis for MA, as integrating behavior and performance analysis in a compositional formal framework.

The rest of this paper is organized as follows: Section 2 introduces some necessary background knowledge, including the syntax and reduction semantics of MA, and some knowledge for probability and Markov chain; Section 3 introduces the enhanced labelled transition system; in Section 4, the cost functions are in-

*Supported by NNSFC (60273034, 60233010), 863 Program (2001AA113110, 2002AA116010), 973 Program of China (2002CB312002), JSFC (BK2002203, BK2002409)

†Email: ctl@ics.nju.edu.cn

troduced and based on it, the Markov chain is derived, which can be used in performance analysis; Section 5 concludes the paper and discusses related work.

2 Background

In this section, we introduce some basic knowledge used in this paper, and mainly focus on the syntax and semantics of MA.

2.1 Syntax and Semantics of MA

2.1.1 Syntax

We assume $\mathcal{N}$ is a countable infinite name set and is ranged over by m, n; $I = \{\tau_0, \tau_1, \tau_2 \dots\}$ is a countable infinite internal action set and ranged by τ_i. In general, we assume $\mathcal{N} \cap I = \emptyset$. Π is MA process set ranged by P, Q and is recursively defined as follows:

$$P, Q ::= 0 | (\nu n)P | P|Q | M.P | M[P] | \langle M \rangle.P | (x).P | A(\tilde{U})$$

where, M is element of capability sets (denoted by CAP) and is recursively defined as:

$$M, N ::= \varepsilon \mid x \mid n \mid in\langle n \rangle \mid out\langle n \rangle \mid open\langle n \rangle \mid \tau_i \mid M.N$$

where, $n \in \mathcal{N}$ and $\tau_i \in I$.

It is noted that in order to fulfill the need of performance analysis, we make some small modifications for MA, that is, we extend the original MA by introducing the notion of internal action, which is used to represent some application related system behaviors while has nothing to do with mobility or communication. Moreover, in this paper, we use the synchronization communication mechanism, which is something different from the original design of MA. The construction of $A(\tilde{U})$ which plays the similar role of replication operator in [1], is standard in process algebra. Due to space restriction, we omit the intuitional interpretation for other operators and refer the reader to [1] for more details.

As in common process calculi, $(\nu n)P$ introduces the distinction of bound names and free names. In general, we use $fn(P)$ to denote the set of free names appearing in process P. For capability M, every name appearing in it is free name. We define $n(P) = fn(P) \cup bn(P)$, where $bn(P)$ denotes the bound name of P. A process P is called closed if $fn(P) = \emptyset$. In general, we denote P is a process expression by $P : \Pi$.

Due to space restriction, we omit the reduction semantics of MA and some background knowledge on probability theory and Markov chain, since they are standard and can be found in literature easily. We refer the reader to [1][4] or the full text of this paper [6] for more details.

3 Enhanced Labelled Transition System

In this section, the enhanced labelled transition system is introduced. We first give some intuitional ideas and then provide the formal definitions.

3.1 Basic Idea

In the research for MA, there are two common methods for presenting labelled transition system, one is the method based on 'commitment' and 'outcome', the other is the method based on 'hardening' relation. Unfortunately, these two kinds of methods can not satisfy our requirements. The former makes system have too many labels and the results of transition may be 'concrete' besides process, which makes it difficult to describe the system states; the latter uses so-called hardening relation, which is a kind of syntax conversion for process expression in essential. However, we think that the structure character of process expression usually embodies the interaction between the process and the context, which has a tight relation on performance. From this point of view, we want to keep as much structure information of process expression as possible, that is, use fewer structure congruence rules, and obtain a 'pure' labelled transition system similar to CCS or π-calculus. Besides that, we want to record more information on process and the application of inference rules. To arrive at this goal, we first introduce some addition actions as well as the prefix operators introduced by MA itself, and explain their intuitional senses as follows:

(i) $enter\langle m, n \rangle$. It means that the ambient named m moves into ambient named n.

(ii) $exit\langle m, n \rangle$. It means the ambient named m move out ambient named n.

(iii) n. The reason for introducing this action is due to technological consideration, while in practice, it can be regard as the safe checking carried by static ambient for the mobile ambient.

(iv) $?x$. It means one process want to receive some messages.

(v) $!M$. It means one process want to send out some messages.

In order to record the application of inference rules, we use the so-called 'tag method' as [5], i.e. attach some tags, e.g. $||_0, ||_1, (\nu n), \tilde{m}$ with actions. In intuition, these tags are mainly used to record the context

of action execution, where $||_0, ||_1$ have something to do with parallel operator |, which are used to record which concurrent process or ambient actually executes the corresponding action; (νn) has something to do with restriction operator, which indicates what names can not be used again; $\tilde{m}$ relates to recursive defined process body, which is a real number vector, and every dimension records the size of actual parameter for recursive defined process.

3.2 Labelled Transition System

In this section, we will give the formal description for labelled transition system.

Definition 1 *(Enhanced Label)*

(i) The basic labels in labelled transition system are given by BNF as follows:

$$\lambda ::= in\langle n\rangle | out\langle n\rangle | open\langle n\rangle | \tau_i$$
$$\alpha ::= \tau_i | open\langle n\rangle | enter\langle m,n\rangle | exit\langle m,n\rangle | n | ?x | !M$$

where, $m, n \in \mathcal{N}$.

(ii) Let $L = \{||_0, ||_1\}$, $\chi \in L^*$, $o \in O = \{(\nu n), \tilde{m}\}$, where for any $\tilde{m}$, we have $\forall m_i \in \tilde{m}, m_i \in R^+$, $\vartheta \in (L \cup O)^*$. Enhanced label set Θ, which is ranged by θ, is defined by BNF as follows:

$$\theta ::= \vartheta\alpha | \vartheta\langle ||_0\vartheta_0\alpha_0, ||_1\vartheta_1\alpha_1\rangle | n[\langle ||_0\vartheta_0\alpha_0, ||_1\vartheta_1\alpha_1\rangle]$$

where, α_0, α_1 is defined as follows:

- *$\alpha_0 = enter\langle m,n\rangle$, $\alpha_1 = n$ or in converse;*
- *$\alpha_0 = exit\langle m,n\rangle$, $\alpha_1 = n$ or in converse;*
- *$\alpha_0 = open\langle n\rangle$, $\alpha_1 = n$ or in converse;*
- *$\alpha_0 = ?x$, $\alpha_1 = !M$ or in converse;*

*(iii) **Label conversion function** is recursively defined as follows:*

$$\begin{cases} l(\vartheta\alpha) = \alpha \\ l(\vartheta\tau_i) = \tau \\ l(\vartheta\langle ||_0\vartheta_0\alpha_0, ||_1\vartheta_1\alpha_1\rangle) = \tau \\ l(n[\langle ||_0\vartheta_0\alpha_0, ||_1\vartheta_1\alpha_1\rangle]) = \tau \end{cases}$$

Definition 2 *The rule for label transition is defined as follows:*

$$(Act)\quad \frac{-}{\lambda.P \xrightarrow{\lambda} P} \qquad (Out)\ \frac{-}{\langle M\rangle.P \xrightarrow{!M} P}$$

$$(In)\quad \frac{-}{(x)P \xrightarrow{?x} P} \qquad (Amb)\ \frac{-}{n[P] \xrightarrow{n} P}$$

$$(Com0)\quad \frac{P \xrightarrow{!M} P' \quad Q \xrightarrow{?x} Q'}{P|Q \xrightarrow{\langle ||_0 !M, ||_1 ?x\rangle} P'|Q'\{M/x\}}$$

$$(Com1)\quad \frac{P \xrightarrow{?x} P' \quad Q \xrightarrow{!M} Q'}{P|Q \xrightarrow{\langle ||_0 ?x, ||_1 !M\rangle} P'\{M/x\}|Q'}$$

$$(Par1\_0)\quad \frac{P \xrightarrow{\lambda} P' \quad \lambda \neq open\langle n\rangle}{P|Q \xrightarrow{||_0\lambda} P'|Q} \qquad (Par1\_1)\ \frac{P \xrightarrow{\lambda} P' \quad \lambda \neq open\langle n\rangle}{Q|P \xrightarrow{||_1\lambda} Q|P'}$$

$$(Par2\_0)\quad \frac{P \xrightarrow{\theta} P' \quad l(\theta) = \tau}{P|Q \xrightarrow{||_0\theta} P'|Q} \qquad (Par2\_1)\ \frac{P \xrightarrow{\theta} P' \quad l(\theta) = \tau}{Q|P \xrightarrow{||_1\theta} Q|P'}$$

$$(Enter)\quad \frac{P \xrightarrow{in\langle n\rangle} P'}{m[P] \xrightarrow{enter\langle m,n\rangle} m[P']} \qquad (Exit)\ \frac{P \xrightarrow{out\langle n\rangle} P'}{m[P] \xrightarrow{exit\langle m,n\rangle} m[P']}$$

$$(Mob\_in0)\quad \frac{P \xrightarrow{enter\langle m,n\rangle} P' \quad Q \xrightarrow{n} Q'}{P|Q \xrightarrow{\langle ||_0 enter\langle m,n\rangle, ||_1 n\rangle} n[P'|Q']}$$

$$(Mob\_in1)\quad \frac{P \xrightarrow{n} P', Q \xrightarrow{enter\langle m,n\rangle} Q'}{P|Q \xrightarrow{\langle ||_0 n, ||_1 enter\langle m,n\rangle\rangle} n[P'|Q']}$$

$$(Mob\_out0)\quad \frac{P \xrightarrow{exit\langle m,n\rangle} P'}{n[P|Q] \xrightarrow{\langle ||_0 exit\langle m,n\rangle, ||_1 n\rangle} P'|n[Q]}$$

$$(Mob\_out1)\quad \frac{P \xrightarrow{exit\langle m,n\rangle} P'}{n[Q|P] \xrightarrow{\langle ||_0 n, ||_1 exit\langle m,n\rangle\rangle} P'|n[Q]}$$

$$(Mob\_open0)\quad \frac{P \xrightarrow{open\langle n\rangle} P' \quad Q \xrightarrow{n} Q'}{P|Q \xrightarrow{\langle ||_0 open\langle n\rangle, ||_1 n\rangle} P'|Q'}$$

$$(Mob\_open1)\quad \frac{P \xrightarrow{n} P' \quad Q \xrightarrow{open\langle n\rangle} Q'}{P|Q \xrightarrow{\langle ||_0 n, ||_1 open\langle n\rangle\rangle} P'|Q'}$$

$$(Res)\quad \frac{P \xrightarrow{\theta} P' \quad m \notin n(l(\theta))}{(\nu m)P \xrightarrow{(\nu m)\theta} (\nu m)P'} \qquad (Tau)\ \frac{P \xrightarrow{\theta} P' \quad l(\theta) = \tau}{m[P] \xrightarrow{m[\theta]} m[P']}$$

$$(Ide)\quad \frac{P\{\tilde{K}/\tilde{U}\} \xrightarrow{\theta} P'}{A(\tilde{K}) \xrightarrow{\tilde{m}\theta} P'} \quad A(\tilde{U}) = P$$

$$(Struc)\quad \frac{P \equiv_R P' \quad P' \xrightarrow{\theta} P''}{P \xrightarrow{\theta} P''}$$

where, we admit only partially structural congruence $\equiv_R$ defined as follows:

$$(\nu n)(\nu m)P \equiv_R (\nu m)(\nu n)P$$
$$(\nu n)(P|Q) \equiv_R P|(\nu n)Q \quad if\ n \notin fn(P)$$
$$(\nu n)m[P] \equiv_R m[(\nu n)P] \quad if\ n \neq m$$

Comparing with standard labelled transition system, the enhanced label transition system introduced in this paper mainly adds some tags. By label conversion function defined in Definition 1, it is not difficult to convert the enhanced label to common one, so the common label transition semantics can be recovered. The following theorem states that the label transition semantics defined in Definition 2 coincides with standard reduction semantics, which can be proved through standard proof techniques. Due to space restriction, we omit the detailed proof.

Theorem 1 *For any process P and Q, $P \to Q$ if and only if $P \xrightarrow{\theta} \equiv Q$, where $l(\theta) = \tau$.*

In the below, we give the formal definition for the labelled transition system, which is important in the rest of the paper.

Definition 3 *For any process P, the enhanced label transition system is defined as following triple $\langle d(P), \xrightarrow{\theta}, P\rangle$, where $\xrightarrow{\theta} = \{Q \xrightarrow{\theta} R | Q, R \in d(P)\}$.*

4 Framework for Performance Analysis

In this section, we will show how to carry out performance analysis using the enhanced labelled transition system defined in Section 3. The basic idea is inspired by [5], that is, to define cost function corresponding to transition label θ, and derive the Markov chain. Note that in this section, we only deal with finite transition system, i.e. the state space generated by process is finite, which means that the process expression is image-finite. For most actual systems, especially reactive systems, this requirement can be easily satisfied.

4.1 Cost function

In this section, we first give some primitive ideas for cost function. Considering any specification of a mobile system, it can be described as a process expression P in MA, then the behavior of the system is totally determined by the labelled transition system $\langle d(P), \xrightarrow{\theta}_P, P\rangle$ which is derived from P. For any two states P_i, P_j, where $P_i \xrightarrow{\theta} P_j$, i.e. process P_i accomplishes action θ and then arrives at P_j, it is noted that P_i may have many kinds of transition, that is, in formal, $P_i \xrightarrow{\theta_i} P_{jk} (1 \leq k \leq n)$, every transition should satisfy corresponding probability distribution and the actual behavior of P_i is random. Obviously, the probability of transition has something to do with the rate of the action. As in the tradition research for performance analysis, the probability distribution considered in this paper is the exponential distribution, because one one hand, the exponential distribution is a good approximation of actual instance, on the other hand, exponential distribution has memoryless property ([5]), which makes it easy to derive Markov chain from the labelled transition system and continue the analysis. Based on the above ideas, the cost function introduced for action θ, in intuition, has a tight relation to the expectation time of the action which is regarded as the parameter of exponential distribution indeed. We argue that two factors play the important roles in the definition of the cost, one is the action itself, that is $\alpha = l(\theta)$, since different actions, such as $enter\langle m, n\rangle, exit\langle m, n\rangle$, should have different cost; the other is the context that the process executing, which is represented in the linked tag ϑ formally. The definition for cost is just according to the two factors. Note that in this paper, the cost of action is regarded as the parameter of exponential distribution, by the definition for the expectation of exponential distribution ([5]), cost is in inverse proportion to the time of actions, which is important to the comprehension of cost function. In the below, the formal definitions are given.

Definition 4 *Function $\$_\alpha$: $I \cup \{enter\langle m, n\rangle, exit\langle m, n\rangle, open\langle n\rangle, ?x, !M, n\} \rightarrow R^+$ is defined as follows:*

$$\begin{cases} \$_\alpha(\tau_i) = \lambda_i \\ \$_\alpha(enter\langle m, n\rangle) = f_{enter}(size(m), bw(m, n)) \\ \$_\alpha(exit\langle m, n\rangle) = f_{exit}(size(m), bw(m, n)) \\ \$_\alpha(open\langle n\rangle) = f_{open}(n) \\ \$_\alpha(?x) = f_{in}(size(x)) \\ \$_\alpha(!M) = f_{out}(size(M)) \\ \$_\alpha(n) = f_{check}(size(m), bw(m, n)) \end{cases}$$

Definition 4 is the definition for the cost of some basic action, we give their intuitional explanations respectively. For the application related internal action τ_i, its cost is given by λ_i. And f_{enter}, f_{exit} define the cost of a ambient (say m) moving in or out some other static ambient (say n). In practice, the mobility of ambient is implemented in the form of channel stream, in this sense, the impact factors are consisted of the bandwidth of channel (represented by $bw(m, n)$) and the bytes which must be sent (represented by $size(m)$). f_{open} defines the cost a process opening a ambient (say n), which is represented by safe checking in practice. f_{in}, f_{out} define the cost of communication, that is, the cost of receiving and sending a message respectively. In MA, the communication is carried out locally, so the cost has relation to the size of communication data and the instance of local data link, the former is represented by $size(x), size(M)$, while the latter has something to do with the context, whose information is recorded in the tag, and its effect on the cost will be represented in the full definition of cost function. f_{check} defines that when a ambient moves in or out a static ambient (say n), the cost of checking done by n, in practice, a typical example is sandbox mechanism for applet in Java; the concrete value of function has something to do with the implementation.

Definition 5 *Function $\$_\vartheta : L \cup O \rightarrow (0, 1]$ is defined as follows:*

$$\begin{cases} \$_\vartheta(||_i) = f_{||}(np),\ i = 0, 1 \\ \$_\vartheta((\nu n)) = f_\nu(|n(P)|) \\ \$_\vartheta(\tilde{m}) = f_{()}(\tilde{m}, np) \end{cases}$$

The definition of $\$_\vartheta$ is similar to the counter part of [5], parallel operator is evaluated according to the

number of processors (np) available, restriction operator tells whether a name can be used, in implementation, its resolution needs a search in a table of names, so the cost of restriction depends on the number of names of process ($|n(P)|$). The definition of $f_{()}$ has something to do with const invocation $A(\tilde{U})$, obviously the cost depends on the size and the number of its actual parameters, as well as on the number of processors available.

In fact, Definition 4 and Definition 5 give the definition of cost for $\vartheta\alpha$ in enhanced label sets Θ introduced in Definition 1. In the below, we give the definition of cost function for the remainder label θ (s.t. $l(\theta) = \tau, \theta \neq \tau_i$). First, we deal with parallel tag part ($||$) in the label. In intuition, the sequence of $||$ represents the distribution location of processor where the concurrent processes run. In implementation, an allocation table is used to record the physical locations of process, whose effect can be defined by a function $f_{<>} : L^* \times L^* \rightarrow (0, 1]$, where $\chi \in L^*$ represents the $||$ sequence of interaction processes or ambients. Therefore, we should extract the parallel tags from $\vartheta(\vartheta \in (L \cup O)^*)$. So, an auxiliary function is defined as follows:

Definition 6 $[.] : (L \cup O)^* \rightarrow L^*$

$$\begin{cases} [\varepsilon] = \varepsilon \\ [||_i\vartheta] = ||_i[\vartheta] & i{=}0,1 \\ [o\vartheta] = [\vartheta] & o \in O \end{cases}$$

Definition 7 *Function* $\$: \Theta \rightarrow R^+$ *is defined as follows:*

$$\begin{cases} \$(\alpha) = \$_\alpha(\alpha) \\ \$(o\theta) = \$_\vartheta(o) \times \$(\theta)\ o \in O \\ \$(||_i\theta) = \$_\vartheta(||_i) \times \$(\theta)\ i = 0, 1 \\ \$(\langle ||_0\vartheta_0\alpha_0, ||_1\vartheta_1\alpha_1\rangle) \\ \quad = f_{<>}([||_0\vartheta_0], [||_1\vartheta_1]) \times \oplus(\$(||_0\vartheta_0\alpha_0), \$(||_1\vartheta_1\alpha_1)) \\ \$(n[\langle ||_0\vartheta_0\alpha_0, ||_1\vartheta_1\alpha_1\rangle]) \\ \quad = c(n) \times f_{<>}([||_0\vartheta_0], [||_1\vartheta_1]) \times \oplus(\$(||_0\vartheta_0\alpha_0), \$(||_1\vartheta_1\alpha_1)) \end{cases}$$

where

$$\oplus \quad (\$(||_0\vartheta_0\alpha_0, \$(||_1\vartheta_1\alpha_1)) = \begin{cases} \frac{\$(||_0\vartheta_0\alpha_0) \times \$(||_1\vartheta_1\alpha_1)}{\$(||_0\vartheta_0\alpha_0) + \$(||_1\vartheta_1\alpha_1)} & \alpha_0 = ?x, \alpha_1 = !M \\ min\{\$(||_0\vartheta_0\alpha_0), \$(||_1\vartheta_1\alpha_1)\} & o.w. \end{cases}$$

Now, we explain the intuition for this definition. As stated above, for an label, say θ, we can obtain two kinds of information, one is the basic action given by $l(\theta)$, the other is the additional tags. The former can be evaluated by the cost function introduced by Definition 4, while the latter in fact represents that the context slow down the speed of actions, whose affection is given by Definition 5. In Definition 7, we synthesize both of the two factors. For the label θ which satisfies $l(\theta) = \tau, \theta \neq \tau_i$, we need some special process. In fact, what we care about are the two interaction partners α_0, α_1. There are mainly two cases: one is communication action pattern. Since communication is synchronous and handshaking, we take the minimum of the costs of the operations performed by the participants independently to make communications reflect the speed of the slower partner. (Recall that we take the cost as the parameter of exponential distribution, so the cost is in inverse proportion to the time of action). The other is mobility action pattern. In implementation, mobility is accomplished in two steps: first the mobile part is transmitted in the form of byte stream, then when it "enter" the accepted part, it is checked by some safety mechanism, so we take sum of the costs of the operations performed by the participants independently, which is the essential idea for definition of operator $\oplus$. Function $c(n)$ is to reflect the affection of runtime context. In practice, an ambient may represent the runtime context, different context has different affection to the interaction, e.g. the communication. In the definition of $\$_\alpha$, it is difficult to reflect the factor. Now, to overcome this deficiency, we introduce *c(n)* as rate function to represent the character of context n, such as the channel bandwidth of local communication and then the difference of the context can be embodied. The other part of function $ is self-interpreted.

4.2 Derivation of CTMC

In this section, based on the cost function introduced in above section, we give the method for deriving the CTMC. We mainly utilize the memoryless property ([5]) of exponential distribution and homogeneous premise of state transmission. Note that in fact, a Markov chain is decided by the set of states and Q Matrix, and the former can be easily derived from labelled transition system, therefore, it is enough to give the definition of Q Matrix.

Definition 8 *For any process P, with its label transition system* $\langle d(P), \xrightarrow{\theta}, P\rangle$*, assume that* $n = |d(P)|$ *is the cardinal of d(P). For any* $P_i, P_j \in d(P)$*, the Q Matrix is defined as* $Q = (q_{ij})_{n\times n}$*, where*

$$q_{ij} = \begin{cases} q(P_i, P_j) = \sum_{P_i \xrightarrow{\theta} P_j} \$(\theta_k) & \text{if } i \neq j \\ -\sum_{j=1, j\neq i}^{n} q_{ij} & o.w. \end{cases}$$

Based on it, now we give the formal description of the Markov chain derived from P.

Definition 9 *For any process P with its labelled transition system* $\langle d(P), \xrightarrow{\theta} \rangle$, *the* ***Continued Time Markov Chain*** *derived from P is defined as transition system* $CTMC(P) = \langle d(P), \xrightarrow{r}_M, P \rangle$, *where* $r \in R^+$, *and the relation* $\rightarrow_M$ *satisfy* $P_i \xrightarrow{\theta_k} P_j$ *if and only if* $P_i \xrightarrow{q_{ij}}_M P_j$, *where* q_{ij} *is given in Definition 8.*

In the below, we will prove that the *CTMC(P)* given in Definition 9 is indeed a *Continued Time Markov Chain.*

Theorem 2 *For any finite state process (image-finite process) P, let X(t) be the corresponding stochastic process, then X(t) is a* ***Continued Time Markov Chain*** *with state space d(P), and coincides with* ***CTMC(P)*** *defined in Definition 16. In further, the sojourn time in any state* $X(t_i) = P_j$ *satisfies exponentially distribution with the parameter* $\lambda = \sum \$(\theta)$, *where* $\theta \in \{\theta | \exists P_k, P_i \xrightarrow{\theta} P_k\}$.

We omit the proof and refer the reader to [6].

Now, Following [5], we can measure the performance of a process P by associating a so-called reward structure with it. Since in our framework, the CTMC is regard as an underlying performance model for distributed concurrent system, the reward structure is simply a function that associates a value with any derivative of P. As in [5], we assume ρ_θ as a given transition reward associated with any transition θ, in accordance with the measures of interest (utilization, throughput, etc), then the reward of a state P is $\rho_P = \sum_{P \xrightarrow{\theta} Q} \rho_\theta$. Based on it, the total reward of P, which combines stationary distribution and rewards, can be defined as $R(P) = \sum_{R \in d(P)} \rho_R \times \Pi(R)$. We can use some standard tool (e.g. Mathematica) to carry out performance evaluation for some systems. In the full version of this paper, we invite an illuminating example to demonstrate the application of our method. Some simulation results are also reported in our technical report ([6]). However, due to space restriction, here we have to omit the detailed presentation and refer the interested reader to [6] for more details.

5 Conclusion and Future Work

The main work of this paper is to provide a framework of performance evaluation for mobile process in the context of calculus of Mobile Ambient. We first introduce a new enhanced labelled transition system for MA, Then, we assign each label a cost by introducing *cost function*, which represents the probability information of the action execution. In analysis, this work can be accomplished mechanically when analyzer provides corresponding parameter for the system. Based on the above work, the CTMC for the system can be easily derived which makes it easy to use some standard tools to obtain the performance results. It is worth pointing out that the advantages of our framework are twofold: (1) the performance task can be accomplished mechanically or semi-mechanically; (2) since we don't change the syntax of specification language(MA), and the standard semantics can be easily recovered from enhanced semantics by label conversion function given in Definition 1. In this sense, we integrate the functional and performance analysis into a compositional formal framework, which provides some basis of unifying formal methodology for the development of mobile system.

The framework introduced in this paper is preliminary. There are many directions remained as future work. First, the probability distribution used in this paper is exponential distribution, we want to introduce other kind of distribution which can model actual case more precisely; at the same time, we want to refine the definition of cost function introduced in Section 3; and more, it is very interesting to design and implement a compositional tool which can support behavior and performance analysis in context of MA, then apply our theoretical results to practice.

References

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

[2] H.Hermanns, U.Herzog, J.Katoen. Process algebra for performance evaluation. Theoretical computer science, 274(2002), pp.43-87, 2002.

[3] R.Milner, J.Parrow, D.Walker. A Calculus of Mobile Process, part I/II. Journal of Information and Computation, 100:1-77, Sept.1992.

[4] R.Nelson. Probability, Stochastic Processes, and Queueing Theory. 1995.

[5] C.Nottegar, C.Priami, P.Degano. Performance evaluation of mobile process via abstract machines, IEEE transactions on software engineering, vol.27, no.10, pp.857-889, 2001.

[6] T.Chen, T.Han, J.Lu. Framework of Performance Evaluation for Mobile Process Based on Mobile Ambient, Technical Report of State Key Laboratory for Novel Software Technology, Nanjing Univ., Nanjing, P.R.China, 2003.