THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 10, Number 1/2009, pp. 000–000

COORDINATED MOBILE AGENTS

Gabriel CIOBANU

Romanian Academy, Institute of Computer Science and A.I.Cuza University of Iaşi
Blvd. Carol I no.8, 700505 Iaşi, Romania
Email: gabriel@iit.tuiasi.ro

Coordination of mobile agents is realized by extending a formalism for mobility with timers. The formalism has explicit notions of location and mobility. Timers define timeouts for various resources, making them available only for a certain period of time. We define shortly the syntax of the model and its operational semantics, and use an example to illustrate how the model is working.

1 INTRODUCTION

Coordination is increasingly relevant to complex concurrent and distributed systems, open systems with mobile entities, and dynamically re-configurable evolving systems. These systems accept uncertainty and are difficult to be described. These aspects lead to difficult control schemes where decision-making are distributed widely and unexpectedly. Among the several challenges related to these systems, we can mention the mechanisms for delivery of adequate responses to time-critical demands, coordination methods that for real-time systems, a decentralized control take into account the timed resources.

In this paper we use ambient calculus for describing distributed and mobile agents. In contrast with other formalisms for mobile processes such as the π-calculus [8] whose computational model is based on the notion of communication, the ambient calculus is based on the notion of movement by using an explicit notion of location. An ambient is both a location and the unit of movement. Ambients mobility is controlled by the capabilities *in, out*, and *open*. Inside an ambient there are processes which may exchange messages. Several variants [2, 9, 7] have been proposed by adding and/or removing features of the original calculus [3].

We define the coordinated mobile agents by starting from mobile ambients and defining communication actions, capabilities and ambients as temporal resources. A timer Δt of each temporal resource makes the resource available only for a determined period of time t.

We use the following notions and conventions:

• domain (location) is a space of local computational activity;

• distributed system is an open collection of domains, each capable of hosting a specific execution context (operating systems, rules and resources), and mobile agents which can migrate between domains;

• open systems: a compositional description (process algebra);

• we do not assume a global clock; we use only the local clocks and the relative time of interaction at a specific location.

The aim is to combine the local execution and the migration changing the execution context, using a non-flat (hierarchical) representation of the distributed network. A non-monotonic behaviour is expressed by using timers which trigger a timeout recovery processes when they expire.

We consider an example which can motive and illustrate the model. We assume a student finishing the university activities and moving out of the university building. The student is looking for an available vehicle in order to move to a given location. In front of the university there are a tram stop, a bus stop, and certain cabs. The student has the possibility to use any of the three types of vehicles (tram, bus, and cab) to reach the target location. The scenario used in this paper involves a tram, a bus and two cabs. The coordinated mobile agents try to provide a faithful syntactic description, and a flexible dynamic semantics for such an example. Coordination is more complex when the components are moving. Therefore we define the coordinated agents

as an extension with timers of the mobile ambients. Timers define timeouts for various resources, making them available only for a determined period of time. Time represents a primary notion; however, since space is dual to time, we think to a model where the notions of location (space) can be explicitly described. Coordination across time is associated with the ability to specify relative starting and ending points of the resource availability. Coordination across space relates to the ability to identify a specific location in which a required resource exists.

2 MOBILE AMBIENTS

Mobile ambients were introduced in 1998 by L. Cardelli and A.Gordon with main purpose to explicitly modelling mobility of agents involved in distributed computation. An ambient is a boundary place where a computation happens (laptop, web-page, virtual address space, etc.). An ambient can be nested within other ambient, and can be moved as whole. An ambient has a name, a collection of local capabilities which control the ambient, and a collection of subambients.

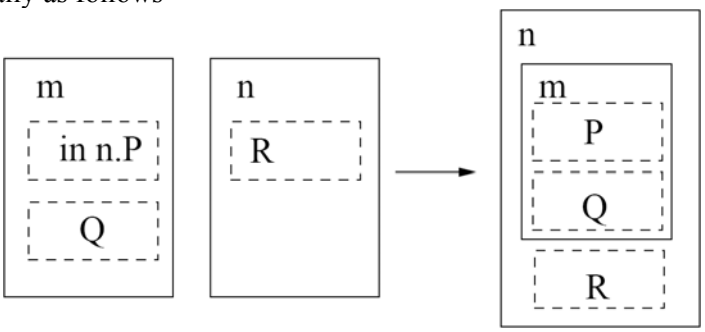
2.1 Formal Syntax

n		names			
P,Q	::=	agents	M	::=	capabilities
(νn) P		restriction		in n	can enter n
0		inactivity		out n	can leave n
P | Q		composition		open n	can open n
*P		replication			
n[P]		ambient			
M.P		action			

Entry capability $in\ n.P$ instructs the surrounding ambient to enter a sibling ambient n; if n does not exist, it blocks. The reduction rule of this capability is given by

$$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$$

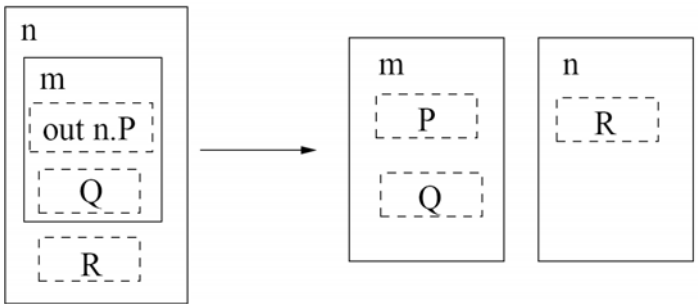
and it is represented graphically as follows



Exit capability $out\ n.P$ instructs the surrounding ambient to exit its parent ambient n; if n does not exist, it blocks. The reduction rule of this capability is given by

$$m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$$

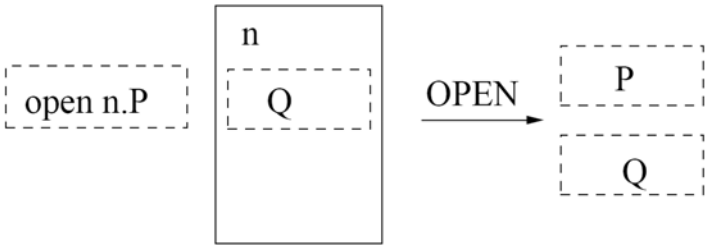
and it is represented graphically as follows



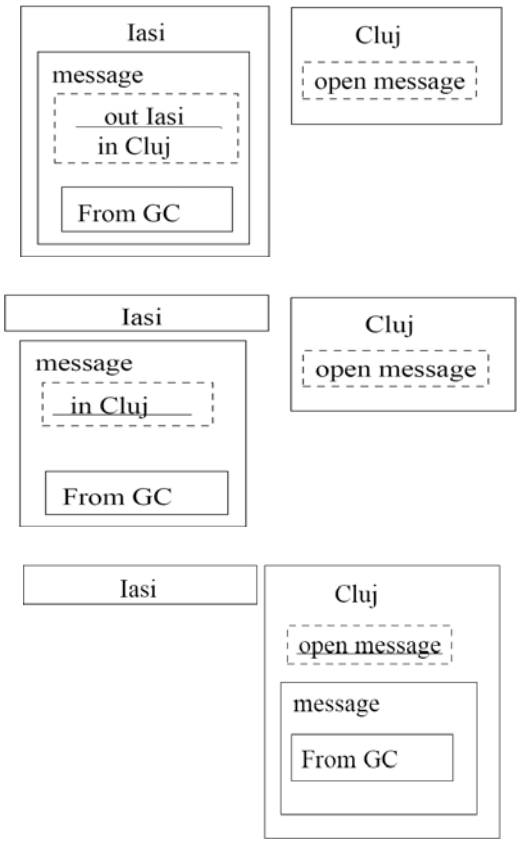
Open capability $open\ n.P$ dissolves the ambient n at the same level as the surrounding ambient; if n does not exist, it blocks. The reduction rule of this capability is given by

$$open\ n.P \mid n[Q] \rightarrow P \mid Q$$

and it is represented graphically as follows



The following sequence of pictures describes how the mobile ambients are working:



3 COORDINATED MOBILE AGENTS

We extend the mobile ambients to $(n^{\Delta t}_{(l,h,d)}[P],Q)$, where

- n denotes an ambient name;
- a *timer* Δt makes the ambient available for a determined period of time t;
- a *capacity* l represents the number of the available resources;
- a *weight* h represents the number of resources consumed by ambient n when it moves into another ambient;
- d defines a *domain* where the ambients can use the local computation resources to plan its movements and navigation (local knowledge);
- P and Q are *processes*, where Q represents a *safety process*.

According to a local clock, Δt is decreasing its value (it expires when $t = 0$); if nothing happens in t units of time, the ambient $n^{\Delta t}_{(l,h,d)}$ is dissolved, process P is reduced to **0**, and safety process Q is executed.

3.1 Syntax

The syntax of the coordinated mobile agents is as follows:

n,m	names		P,Q::=	processes
x,y	variables			
			0	inactivity
M ::=	capabilities		$M^{\Delta t}.(P,Q)$	movement
in n	can enter n		$(n^{\Delta t}_{(l,h,d)}[P],Q)$	ambient
out n	can exit n		P | Q	composition
open n	can open n		$(\nu n)P$	restriction
go y	migration to y		$!\langle m\rangle^{\Delta t}.(P,Q)$	output action
			$?(x)^{\Delta t}.(P,Q)$	input action
			*P	replication
			P + Q	choice

A migration $go\ y$ changes the domain d to a domain y. A capability $open^{\Delta t}n.(P,Q)$ evolves to P whenever in Δt the process becomes sibling to an ambient n; otherwise evolves to Q. An output action $!\langle m\rangle^{\Delta t}.(P,Q)$ evolves to P whenever in Δt the process becomes sibling to a process which is intending to capture the name m; otherwise evolves to Q.

The passage of time is described by a discrete time-stepping function $\phi_\Delta : \mathcal{P} \to \mathcal{P}$.

$$\phi_\Delta(P) = \begin{cases} M^{\Delta(t-1)}.(R,Q) & if\ P = M^{\Delta t}.(R,Q), t > 0 \\ Q & if\ P = M^{\Delta t}.(R,Q), t = 0 \\ \phi_\Delta(R) \mid \phi_\Delta(Q) & if\ P = R|Q \\ (\nu n)\phi_\Delta(R) & if\ P = (\nu n)R \\ (n^{\Delta(t-1)}_{(l,h,d)}[\phi_\Delta(R)],Q) & if\ P = (n^{\Delta t}_{(l,h,d)}[R],Q), t > 0 \\ Q & if\ P = (n^{\Delta t}_{(l,h,d)}[R],Q), t = 0 \\ P & if\ P = *R or P = 0 \\ \phi_\Delta(R) + \phi_\Delta(Q) & if\ P = R + Q \end{cases}$$

ϕ_Δ affects the ambients and their capabilities which are not consumed. The consumed capabilities disappear together with their timers.

3.2 Semantics

The operational semantics is given by a reduction relation defined by the following reduction rules:

(R-GTProgress)	$\dfrac{P \not\rightarrow}{P \rightarrow \varphi_\Delta(P)}$
(R-In)	$\dfrac{h' \le l''}{\left(n^{\Delta t'}_{(l',h',d)}\left[in^{\Delta t} m.(P,P') \mid Q\right], S'\right) \mid \left(m^{\Delta t''}_{(l'',h'',d)}\left[R\right], S''\right) \rightarrow \left(m^{\Delta t''}_{(l''-h',h'',d)}\left[\left(n^{\Delta t'}_{(l',h',d)}\left[\psi(P) \mid Q\right], S'\right) \mid R\right], S''\right)}$
(R-Out)	$\dfrac{h'' \le l}{\left(k^{\Delta t}_{(l,h,d)}\left[\left(m^{\Delta t'}_{(l',h',d)}\left[\left(n^{\Delta t''}_{(l'',h'',d)}\left[out^{\Delta t} m.(P,P') \mid Q\right], S''\right) \mid R\right], S'\right)\right], S\right) \rightarrow \left(k^{\Delta t}_{(l-h'',h,d)}\left[\left(n^{\Delta t''}_{(l'',h'',d)}\left[\psi(P) \mid Q\right], S''\right) \mid \left(m^{\Delta t'}_{(l'+h'',h',d)}\left[R\right], S'\right)\right], S\right)}$
(R-Open)	$\dfrac{-}{\left(m^{\Delta t'}_{(l',h',d)}\left[open^{\Delta t} n.(P,P') \mid \left(n^{\Delta t''}_{(l'',h'',d)}\left[Q\right], S''\right)\right], S'\right) \rightarrow \left(m^{\Delta t'}_{(l''+l'',h',d)}\left[\psi(P) \mid Q\right], S'\right)}$
(R-Go)	$\dfrac{t' = 0}{\left(n^{\Delta t}_{(l,h,d)}\left[go^{\Delta t'} d'.P\right], Q\right) \rightarrow \left(n^{\Delta t}_{(l,h,d')}\left[P\right], Q\right)}$
(R-Com)	$\dfrac{-}{!\langle m \rangle^{\Delta t}.(P,Q) \mid ?(x)^{\Delta t'}.(P',Q') \rightarrow P \mid P'\{m/x\}}$
(R-Amb)	$\dfrac{P \rightarrow Q}{\left(n^{\Delta t}_{(l,h,d)}\left[P\right], R\right) \rightarrow \left(n^{\Delta t}_{(l,h,d)}\left[\psi(Q)\right], R\right)}$
(R-LPar)	$\dfrac{P \rightarrow Q}{R \mid P \rightarrow R \mid Q}$
(R-RPar)	$\dfrac{P \rightarrow Q}{P \mid R \rightarrow Q \mid R}$
(R-Par)	$\dfrac{P \rightarrow Q, P' \rightarrow Q'}{P \mid P' \rightarrow Q \mid Q'}$
(R-Res)	$\dfrac{P \rightarrow Q}{(\nu n)P \rightarrow (\nu n)Q}$
(R-Struct)	$\dfrac{P' \equiv_p P, P \rightarrow Q, Q \equiv_p Q'}{P' \rightarrow Q'}$

Ambients can interact using these rules only if they are on the same domain. To preserve the timers of P, we define $\psi : \mathcal{P} \rightarrow \mathcal{P}$ by $\phi_\Delta(\psi(P)) = P$.

We denote by $\not\rightarrow$ the fact that rules (R-In), (R-Out), (R-Open) and (R-Com) cannot be applied.

If one process evolves by one of (R-In), (R-Out), (R-Open) and (R-Amb), while another one does not perform any reduction, then one of (R-LPar), (R-RPar) should be applied. If more than one process evolve in parallel by applying either (R-In), (R-Out), (R-Open) or (R-Amb), then (R-Par) should be applied. When rules (R-In), (R-Out), (R-Open) and (R-Com) cannot be applied anymore, then (R-GTProgress) is applied.

4 INTERACTION OF COORDINATED MOBILE AGENTS

Various agents are in a distributed system where we point out two domains: univ and campus. In fact, we can imagine a student located at univ and looking to move to campus. It is possible to use either a tram, a bus, or a cab.

$$student_{(1,1,univ)}[\ in^{\Delta t_1} tram.P_1 + in^{\Delta t_2} bus.P_2 + in^{\Delta t_3} cab.$$

$$at[out^{\Delta t 4} student!\langle campus\rangle^{\Delta t_5}.(P_3, Q_3)\ ]\ ]\ldots$$

The bus and the tram are moving according to predetermined schedules:

$$tram_{(l,h,d\_init)}[\ go^{\Delta t_1} s_1.go^{\Delta t_2} s_2 \ldots go^{\Delta t_n} s_n\ ]\ldots$$

$$bus_{(l,h,d\_init)}[\ go^{\Delta t_1} p_1.go^{\Delta t_2} p_2 \ldots go^{\Delta t_n} p_n\ ]\ldots$$

Our scenario involves also two cabs (a hired cab and an available cab).

$$cab_{(4,1,univ)}[\ *(open^{\Delta t_1} at.?(y)^{\Delta t_2}.(P,P').go^{\Delta t_3}\ y.(Q,Q'))\ ]\ldots$$

Interaction between the agents student and tram is described by

$$[student_{(1,1,univ)}[\ in^{\Delta t_{12}} tram\ ]\ |$$

$$|\ tram_{(l\_tram,h\_tram,univ)}[\ go^{\Delta t_{k+1}} s_{k+1} \ldots go^{\Delta t_j} campus \ldots\ ]\ ]$$

After t_k units of time, the tram activates its capability $go^{\Delta t_k} s_k$ and moves out of the domain $univ$.

Interaction between the agents student and bus is described by

$$[student_{(1,1,univ)}[\ in^{\Delta t_{10}} bus\ ]\ |\ bus_{(0,h\_bus,univ)}[\ go^{\Delta t_{i+1}} s_{i+1} \ldots go^{\Delta t_j} campus \ldots\ ]\ ]$$

If the bus has free resources, then $student$ can use bus. However the bus have not free resources (its capacity is 0).

Interaction between the agents student and cabs is described by

$$student_{(1,1,univ)}[\ in^{\Delta t_{14}} cab.\ldots\ ]\ |$$

$$cab_{(4,1,univ)}[\ *(open^{\Delta t_5} at.?(y)^{\Delta t_6}.(P,P').go^{\Delta t_7}\ y.(Q,Q'))\ ]\ |\quad cab_{(0,1,univ)}[\ldots]$$

One cab is not available (its capacity is 0). The other one is available with capacity 4; once hired, its capacity becomes 0. Interaction between the students student and cabs is described by the following evolution steps:

$$univ^{\infty}[student_{(1,1,univ)}[\ in^{\Delta t_{14}} cab.\ldots\ ]\ |$$

$$cab_{(4,1,univ)}[\ *(open^{\Delta t_5} at.?(y)^{\Delta t_6}.(P,P').go^{\Delta t_7}\ y.(Q,Q'))\ ]\ ]\ |\ campus^{\infty}[\ ]$$

$$\rightarrow univ^{\infty}[cab_{(4,1,univ)}[\ *(open^{\Delta t_5} at.?(y)^{\Delta t_6}.(P,P').go^{\Delta t_7}\ y.(Q,Q'))\ |$$

$$student_{(1,1,univ)}[at^{\Delta t'}[out^{\Delta t_3} student.!\langle campus\rangle^{\Delta t_4}.(P_3,Q_3)\ ]]]]\ |\ campus^{\infty}[\ ]$$

$$\equiv univ^{\infty}[cab_{(4,1,univ)}[\ open^{\Delta t_5} at.?(y)^{\Delta t_6}.(P,P').\ldots.\ |\ *(open^{\Delta t_5} at.\ldots)\ |$$

$$student_{(1,1,univ)}[at^{\Delta t'}[out^{\Delta t_3} student.!\langle campus\rangle^{\Delta t_4}.(P_3,Q_3)\ ]]]]\ |\ campus^{\infty}[\ ]$$

$$\rightarrow univ^{\infty}[cab_{(4,1,univ)}[\ open^{\Delta t_5} at.?(y)^{\Delta t_6}.(P,P').\ldots.\ |\ *(open^{\Delta t_5} at.\ldots)\ |$$

$$at^{\Delta t'}[!\langle campus\rangle^{\Delta t_4}.(P_3,Q_3)\ ]\ |\ student_{(1,1,univ)}[\ ]]]\ |\ campus^{\infty}[\ ]$$

$$\rightarrow univ^{\infty}[cab_{(4,1,univ)}[\ ?(y)^{\Delta t_6}.(P,P').go^{\Delta t_7}\ y.(Q,Q')\ |\ *(open^{\Delta t_5} at.\ldots)\ |$$

$$!\langle campus\rangle^{\Delta t_4}.(P_3, Q_3) \mid student_{(1,1,univ)}[\,]]] \mid campus^{\infty}[\,]$$
$$\rightarrow univ^{\infty}[cab_{(4,1,univ)}[\ go^{\Delta t_7}\ campus.(Q, Q') \mid *(open^{\Delta t_5} at\ldots.) \mid$$
$$student_{(1,1,univ)}[\,]]] \mid campus^{\infty}[\,]$$
$$\rightarrow univ^{\infty}[\,] \mid campus^{\infty}[cab_{(4,1,univ)}[*(open^{\Delta t_5} at\ldots.) \mid student_{(1,1,univ)}[\,]]]$$

5 CONCLUSION

Coordination of mobile agents is given by using explicit notions of location and mobility, as well as some quantitative notions including time and capacity. We extend mobile ambients for controlling mobility in open distributed systems by means of timers, bounded capacities and a simple migration primitive. The evolution of agents is illustrated by an example involving explicit locations, mobility, and non-monotonic behaviour.

The formalism could be used for real application involving mobile agents. A related formalism presented in [1] is used to describe network protocols.

REFERENCES

1. AMAN, B., CIOBANU, G., *Timed Mobile Ambients for Network Protocols*, Lecture Notes in Computer Science vol.**5048**, Springer, pp.234--250, 2008.
2. BUGLIESI, M., CASTAGNA, G., CRAFA, S., *Boxed Ambients*, Lecture Notes in Computer Science vol.**2215**, Springer, pp.38--63, 2001.
3. CARDELLI, L., GORDON, A., *Mobile Ambients*, Lecture Notes in Computer Science vol.**1378**, Springer, pp.140--155, 1998.
4. CIOBANU G., *Time and Space Coordination of Mobile Agents*, Advances in Intelligent and Distributed Computing, Springer, pp.9--24, 2007.
5. CIOBANU, G., PRISACARIU, C., *Timers for Distributed Systems*, Electronic Notes in Theoretical Computer Science vol.**164**, pp.81--99, 2006.
6. HENNESSY, M., *A Distributed pi-calculus*, Cambridge University Press, 2007.
7. LEVI, F., SANGIORGI, D., *Controlling Interference in Ambients*, Principles of Programming Languages, ACM Press, pp.352--364, 2000.
8. MILNER, R., Communicating and Mobile Systems: the π -calculus, Cambridge University Press, 1999.
9. TELLER, D., ZIMMER P., HIRSCHKOFF, D., *Using Ambients to Control Resources*, Lecture Notes in Computer Science vol.**2421**, Springer, pp.288-303, 2002.

Received July 21, 2008