Towards Context-Aware Workflow Management for Ubiquitous Computing♦

Feilong Tang[1,2], Minyi Guo[1,2], Mianxiong Dong[2], Minglu Li[1], Hu Guan[1]
[1]*Department of Computer Science and Engineering*
Shanghai Jiao Tong University, Shanghai 200240, China
[2]*School of Computer Science and Engineering*
The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan
E-mail: tang-fl@cs.sjtu.edu.cn

Abstract

Ubiquitous computing is a user-centric distributed computing paradigm, allowing users to access to their preferred services even while moving around. To make such a vision a reality, context-aware workflow management is one of key issues because the context of ubiquitous applications is highly varying. In this paper, we propose a context model for intelligent campus navigation applications, then present a context-adaptive workflow management algorithm which can dynamically adjust workflow execution policies in terms of current context information. Moreover, we model the workflow management algorithm in Petri nets. Our context model and workflow management algorithm may be easily extended to other ubiquitous applications with a little change on context structure.

1. Introduction

Ubiquitous computing is a new paradigm of distributed systems with the goal for users to access information anytime anywhere while moving around. Owing to the high mobility of users, ubiquitous softwares run in an extremely dynamic and heterogeneous environment, where many kinds of network nodes and communication protocols co-exit, and network topology and bandwidth are frequently varying. As a result, context awareness and context-driven self-adaptation is an important characteristic distinguishing from traditional distributed systems. Ubiquitous softwares have to be aware of varying context for dynamically adjusting execution policies and supporting seamless roaming to achieve the user-centric vision. Along with this direction there have been many active application-driven researches [1-5].

Context-adaptive workflow management is an important enabling technology to achieve the user-centric vision. So far, however, not many results have been produced in context-aware ubiquitous workflow management. Workflow management in the ubiquitous environment should consider the characteristics of both static and dynamic topologies, wired and wireless subnets, mobile and disconnected users, etc. Neither of existing workflow management proposals sufficiently addresses the combined and new requirements for ubiquitous applications.

Ubiquitous computing was spurred from application requirements[16]. Context-aware navigation service is one of significant ubiquitous applications. In this paper, we target on context-awareness-based workflow management for ubiquitous campus navigation. Let there be many libraries, restaurants and bus stops, each of them in different locations. Alice plans to firstly visit a library to borrow some books, then have a lunch in a restaurant, and lastly take a bus to go home. Fig. 1 illustrates such a tour, where A_1, A_2 and A_3 are three sub-activities in Alice's campus tour. Alice is not familiar with the campus environment and just holds a PDA (personal digital assistant) with GPS (global positioning system) functions. To make Alice's campus tour convenient and efficient, the ubiquitous campus should provide an intelligent navigation for him in a context-aware way. When Alice wants to take a bus, for example, a campus navigation system should

♦ This work is supported by the National High Technology Research and Development Program (863 Program) of China (Grant Nos. 2006AA01Z172 and 2006AA01Z199), the National Natural Science Foundation of China (NSFC) (Grant Nos. 60773089, 60533040, and 60725208), Natural Science Foundation of Shanghai Municipality of China (Grant No. 05ZR14081), and Shanghai Pujiang Program (Grant No. 07pj14049).

978-0-7695-3287-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICESS.2008.83

find the shortest bus stop away from his current location and send navigation information (e.g., routing path) to Alice's PDA. Furthermore, the navigation system should automatically transmit requests and responses via qualified connections that have stable connection and sufficient communication bandwidth.

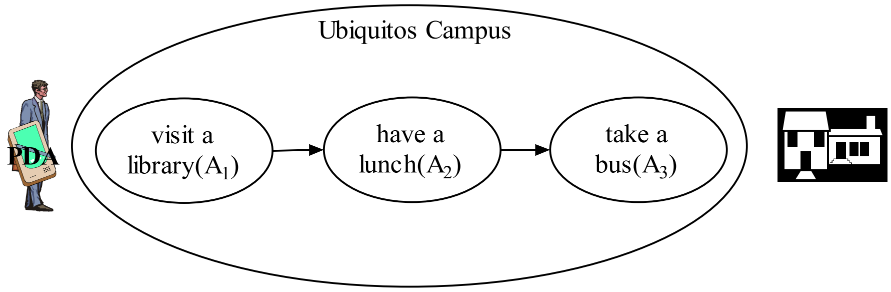


Fig.1. A campus tour.

The above scenario is highly context-dependent, especially in the following ways, which is different from traditional workflow applications:

- *location-dependent*. When a user hope to have a lunch, for example, the navigation system should find a restaurant closest to the user's current location.
- *network-dependent*. The navigation system has to select a qualified network to transmit the request and response for the user, based on current network topology and bandwidth.
- *process-transparent*. The details of a workflow execution process is invisible to the user, including qualified network selection, expected target finding etc.

In this paper, we propose a context-aware workflow management service characteristic to navigation applications in ubiquitous campuses. We model the context in the campus navigation, present a context-aware workflow management algorithm (CAWM). Finally, we model the CAWM through Petri nets.

2. Related work

In this Section, we carefully review existing proposals of workflow management for traditional distributed systems as well as ubiquitous environments.

2.1. Traditional workflow management

Workflow management technology has been extensively researched for traditional distributed systems. DISCOBOLE [6] is a service architecture for interconnecting workflow processes. This model supports dynamic heterogeneous workflow process interconnection for inter-enterprise cooperations, where several heterogeneous workflow management systems coexist. HowU [7] is an information service model, where a service is regarded as the basic unit of information publication. The HowU service is able to guarantee service existence in wide area and service information validity in organization area. McRunjob [8] is a grid workflow manager used to manage the generation of large number of production processing jobs in high energy physics. It converts core metadata into jobs submitted in a variety of environments. In 2002, Krishnan et al. [9] presented a workflow framework for Grid services(GSFL) to manage workflows within the Open Grid Services Architecture (OGSA) framework. The GSFL focuses on the definition of an XML-based workflow language, but also includes a description of an implementation of a workflow engine. An important feature distinguishing from Web services workflow models is the GFSL proposes a Notification Model to provide the solution to the problem of the workflow engine mediating at every step of an activity.

Nichols et al.[10] proposed a model for autonomic workflow management in Grids. The model integrated dynamic fault tolerance mechanism selection into Grid-based workflow management systems, providing awareness and resilience to failures for automatic Grid computing. ChinaGrid Support Platform (CGSP) [11] is a grid middleware developed for the ChinaGrid project, one of the world's largest implementations of Grid computing, fund supported by the Ministry of Education of China with more than 500M RMB (about 60M USD dollars). Workflow Engine is one of main components of CGSP, providing Grid applications with workflow execution, control, monitoring as well as workflow load balancing. Task net [12] is transactional workflow model based on colored Petri net. It defined a workflow specification language that can express task state dependency relationships and enable users to express both the transaction- and application-oriented requirements of complex business rules. However, these existing workflow systems have not yet included functions enough to support context-aware services for ubiquitous computing [13].

2.2. Workflow management for ubiquitous computing

There also have been reports [13-15] on workflow management for ubiquitous computing. Cho et al. [13] proposed a context-aware workflow service system that uses contexts as conditions of service execution, and dynamically derives service transition according to a user's situation information generated from real environments. When a change of a user's situation information happens, the system can dynamically reconstruct a context-aware service workflow. In [14], Hsieh presented a workflow model to describe the medical service processes in healthcare systems, with the goal of automatic resource allocation. Elcsarva [15] is a workflow-centric context-aware collaboration framework, focusing the collaboration in meeting the trends of ubiquitous computing for process automation. In order to support heterogeneous and dynamic collaboration environment, Elcsarva maintains clients' context profile, and periodically uses a "pull" mechanism to survey each client's profile to maintain and update the whole collaboration context environment.

Distinguishing from above proposals, this paper investigates how to automatically manage workflows based on varying context in *ubiquitous environments* where many kinds of networks (wired, WiFi, WiMax, Bluetooth etc.) co-exist and users keep highly mobile.

3. Context model for campus navigation

Context awareness enables systems to automatically configure and adjust to provide better services for users. A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context. The challenge for such systems lies in the complexity of representing and processing contextual data.

Context is the representation of the information that is relevant to the individuals and devices within the activity space[17]. A context adaptive system typically enables the user to maintain a certain application (in different forms) while roaming between different wireless access technologies, locations, devices and even simultaneously executing everyday tasks. Ubiquitous campus navigation systems focus on providing navigation information (e.g., text, photo) of targets for users with mobile devices (e.g., PDA) based on current location of the users and status of networks and devices. So, we consider the following context dimensions for campus navigation applications (see Table 1).

- Person: profile, preferences and requirements of a user,
- Location: longitude and latitude (or relative position) of a user,
- Network: connection and performance of networks, and
- Mobile device: computing and storage capacity of devices

Table 1. Context of ubiquitous campus

Entity	*Attribute*	*Value*
Person	Name	Name description
	Sex	Male or female
	Age	Value of age
	Identity	Teacher, student, guest etc.
	Requirement	Requirement description
	Preference	Behavior preference
Network	Connectivity	Connected, disconnected
	Bandwidth	High, medium, low
	Cost	Expensive, cheap, free
	Stability	Good, medium, bad
Device	Available_ Battery	Full, half, low
	Available_ Data	Available, unavailable
	Computing_ Capacity	High, medium, low
	Available_ Memory	Full, half, low
	Available_ Cache	Full, half, low
Location	Longitude	Value of longitude
	Latitude	Value of latitude

We model context information using *entity*, *attribute* and *relationship* among entities. Note that some attributes (e.g., preference) can not be directly abstracted from entity profiles and data mining-like technologies have to be used for this purpose. The graphic context model for ubiquitous campus navigation is shown in Fig.2, where we only illustrate attributes of the entity *Person*. Attributes of other entities can be easily complemented in the same way according to Table 1. An entity is fully captured by means of its attributes. For example, a user with the name Alice can be described just like this:

Person.Name="Alice",

Person.Identity="student",

Person.Age="20",

Person.Sex="male",

Person.Requirement="he needs to visit a library, a restaurant and a bus stop", and

Person.Preference="he likes computer books".

On the other hand, there are relationships between two entities. For example, the relationship "Person $\frac{m}{n}$ Location" means that these two entities have many to many relationships, i.e., a person may lie at many locations while a location can be visited by many persons.

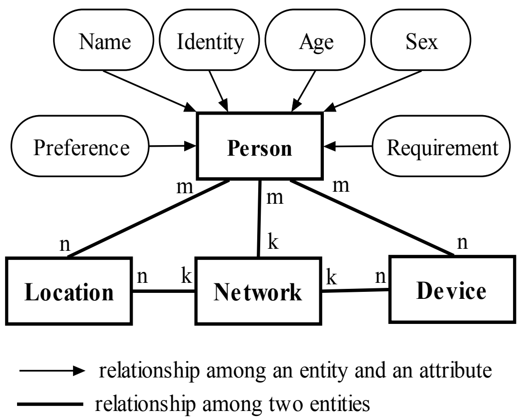


Fig.2. Context model for ubiquitous campus navigation.

4. Context-aware workflow management

For a context-aware ubiquitous navigation application, system would make the navigation functionality available for different availability of networks, locations as well as adapting the user interaction operability on interesting targets while keeping in mind the overall applicability depending on the user preferences. In this section, we define Petri net model for a context-aware workflow, and then present an algorithm for context-aware workflow management.

4.1. Context-aware workflow based on Petri nets

A Petri net is an effective modeling tool applicable to many systems, especially to distributed and parallel systems. It can model systems' events, conditions and relationships. The occurrence of these events may change the state of the system, causing some of the previous conditions to cease holding and other conditions to begin to hold [18-20]. In the following, we model a context-adaptive ubiquitous workflow in Petri nets.

Definition 1. a context-adaptive workflow (CAW) is a 7-tuple CAW=(P,T,I,O,IT,CO,M_0), where

$P=\{p_1,p_2,\ldots,p_n\}$ is a finite set of places,

$T=\{t_1,t_2,\ldots,t_n\}$ is a finite set of transitions such that $P\cap T=\varnothing$ and $T\cap P=\varnothing$,

$I\subseteq P\times T$ is a finite set of input arcs from p_i to t_j ($1\leqslant i\leqslant m, 1\leqslant j\leqslant n$),

$O\subseteq T\times P$ is a finite set of input arcs from t_j to p_i ($1\leqslant i\leqslant m, 1\leqslant j\leqslant n$),

$IT=\{IT_1, IT_2,\ldots,IT_k\}$ is a finite set of user's interesting targets,

CO is a finite set of context information, and

M_0 is the initial marking.

4.2. Context-aware workflow management algorithm

The goal of campus navigation is to provide expected navigation information based on current context. Let a user be interested in visiting n targets (e.g., library, restaurant) in a campus tour. We describe the set of a user's interesting targets (IT) as $IT=\{IT_i \mid IT_i$ is one of expected targets, $1\leq i\leq n\}$. IT is an ordered set with the dependence $IT_i \prec IT_{i+1}$, i.e., IT_{i+1} is not executed until IT_i finishes. C_i is a set of candidates corresponding to an interesting target IT_i such that $C_i=\{C_{i,j} \mid C_{i,j}$ is a candidate of IT_i, $1\leq j\leq m\}$. C_i is also a ordered set such that $C_{i,j}$ is closer to a user's *current location* than $C_{i,j+1}$. Note that *current location* is the newest location where the user lies when last sub-activity finishes. At the beginning of a campus navigation workflow, for example, current location means the place where Alice initiates a request while after Alice arrives at a library, current location refers to that library. Moreover, we assume a request can be sent to servers via K networks such that the set of networks $N=\{N_k \mid N_k$ is a network between the request node and the server, $1\leq k\leq K\}$.

```
Context-Aware Workflow Management Algorithm
Input: a set of a user's expected targets (IT);
Output: navigation information of IT;
{
   i=1;
   while (IT ≠ ∅ ) do
      C_i= candidates corresponding to IT_i;
      j=1;
      suc=true;
      while (C_i ≠ ∅ ) and (suc) do
         k=1;
         repeat
            η =bandwidth of network N_k;
            if ( η is low) k++;
         until (k>K) or ( η is high or medium);
         send request for C_i,j to server via network N_k;
         wait for navigation information of IT_i,j;
         if (the user does not satisfy) {
            C_i=C_i-{C_i,j};
            j++;  }
         else
            suc=false;
      endwhile
      IT=IT-{IT_i};
      i++;
   endwhile
}
```

Fig.3. Context-aware workflow management algorithm for campus navigation (CAWM).

Our context-aware workflow management algorithm (CAWM) adopts an event-context-action (ECoA) mechanism. The algorithm is illustrated in Fig.3, which is driven by events and can dynamically adjust workflow execution policies based on current context information. Attributes of each entity can have values shown in Table 1 such that $\forall\ \eta$=Network.Bandwidth, $\exists\ \eta \in$ {high, medium, low}. The algorithm requires that the bandwidth of a network for transmitting workflow request and result have to be high or medium. Currently, we only consider main context dimensions such as the user location, network bandwidth. More dimensions of context will be incorporated in our future work.

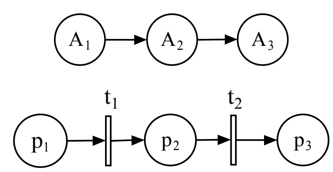
5. Petri nets of the algorithm CAWM

5.1. Graphic Petri nets of workflow fragments

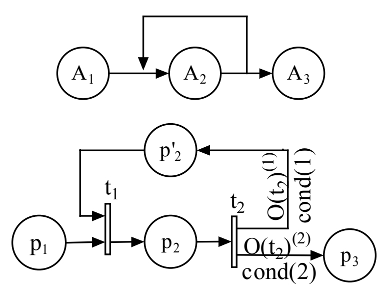
Before illustrating graphic Petri nets of the CAWM, we introduce various workflow execution structures for our context-adaptive workflow as shown in Fig.4, where A_i denotes a sub-activity in a workflow, and p_i and t_i/$t_i(j)$ mean corresponding system states and actions respectively. Each sub-figure in Fig.4 denotes a workflow fragments and the corresponding Petri net. To express the selective execution, we introduce a *selective transition* concept. Let the set of output arcs of a transition t be O(t) such that $O(t)=\{O(t)^{(i)} \mid O(t)^{(i)}$ is one of output arcs, $1 \le i \le n\}$, where n is the number of t's output arcs.

Definition 2. A transition t is *selective* if (1) any $O(t)^{(i)} \in O(t)$ associates with a condition $O(t)^{(i)}.cond(i)$, and (2) when a firing occurs, only $O(t)^{(i)}$ with $O(t)^{(i)}.cond(i)=true$ $(1 \le i \le n)$ is fired.

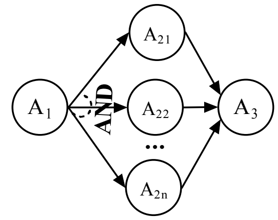
Selective transaction concept extends the modeling ability of Petri nets by introducing firing conditions for each output arcs. Let a selective transition t_2 in Fig.4(2) denote express transmitting data via a network N_k, and corresponding conditions be network performance. When t_2 is fired, only $O(t)^{(2)}$ is actually fired if the N_k is qualified, i.e., $O(t)^{(2)}.cond(2)=true$. Otherwise, $O(t)^{(1)}$ is fired when the N_k is not qualified.

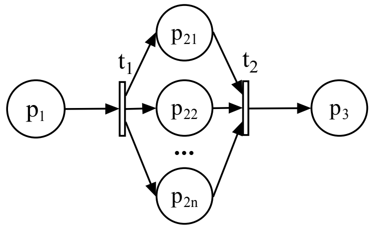


(1) sequential execution: sub-activities A_1,A_2,A_3 are executed one by one

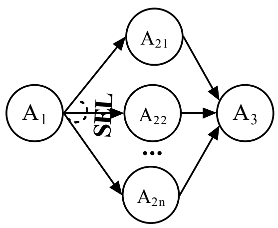


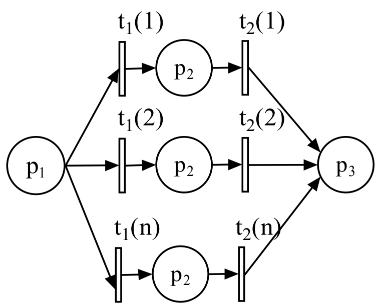
(2) iterative execution: A_2 is repeatedly executed



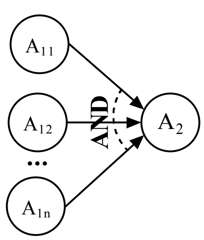


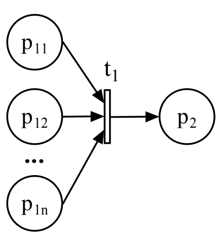
(3) AND-Split: all successors $A_{21}, A_{22}, \ldots A_{2n}$ are executed



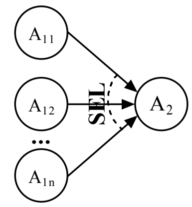


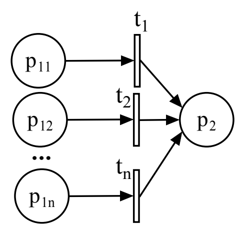
(4) Selective(SEL)-Split: one of successors $A_{21}, A_{22}, \ldots A_{2n}$ is executed



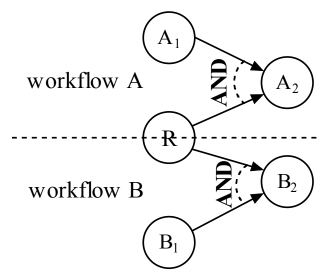


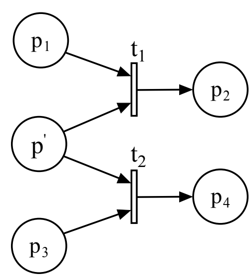
(5) AND-Join: A_2 is executed if and only if all predecessors $A_{11}, A_{12}, \ldots A_{1n}$ finish





(6) Selective(SEL)-Join: A_2 is executed if one of predecessors $A_{11}, A_{12}, \ldots A_{1n}$ finishes





(7) Exclusive resource: workflows A and B exclusively use the resource R

Fig. 4. Workflow execution structures and corresponding Petri nets.

5.2. Petri nets of CAWM

Fig.5 depicts the graphic Petri net of the CAWM, simplified PN-CAWM, where places refer to states of sub-activities and networks while transitions means the execution of corresponding actions (see Table 2). Three selective transitions t_3, t_5 and t_6 denote respectively inspecting network status, checking the navigation information for a target IT_i and judging whether a workflow finishes, i.e.:

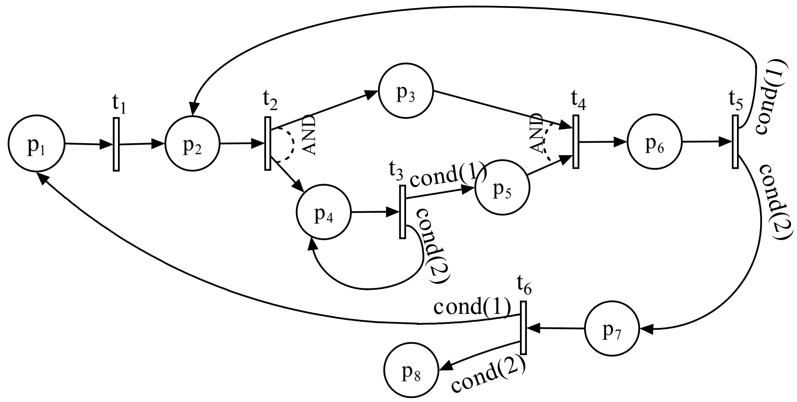


Fig. 5. Graphic Petri net of the CAWM (PN-CAWM).

$O(t_3)$.cond(1): network N_k has good performance especially enough bandwidth,

$O(t_3)$.cond(2): N_k does not has enough bandwidth,

$O(t_5)$.cond(1): a user does not accept navigation information of the candidate $C_{i,j}$,

$O(t_5)$.cond(2): the user accepts navigation information of the candidate $C_{i,j}$,

$O(t_6)$.cond(1): a workflow does not finish, and

$O(t_6)$.cond(2): the workflow finishes.

According to Fig.5, we have the CAWN's Petri net PN-CAWN=(P,T,I,O,IT,CO,M_0), where

$P=\{p_1,p_2,\ldots,p_8\}$,

$T=\{t_1,t_2,\ldots,t_6\}$,

$I=\{I(t_1), I(t_2), I(t_3), I(t_4), I(t_5), I(t_6)\}$, where $I(t_1)=\{p_1\}$, $I(t_2)=\{p_2\}$, $I(t_3)=\{p_4\}$, $I(t_4)=\{p_3,p_5\}$, $I(t_5)=\{p_6\}$,$I(t_6)=\{p_7\}$

$O=\{O(t_1), O(t_2), O(t_3), O(t_4), O(t_5), O(t_6)\}$, where $O(t_1)=\{p_2\}$, $O(t_1)=\{p_2\}$, $O(t_2)=\{p_3, p_4\}$, $O(t_3)=\{p_5|$ cond(1), $p_4|$cond(2)$\}$, $O(t_4)=\{p_6\}$, $O(t_5)=\{p_2|$cond(1), $p_7|$cond(2)$\}$, $O(t_6)=\{p_1|$cond(1), $p_8|$cond(2)$\}$. Note that for a selective transaction t, only qualified output arcs are actually fired when t is fired.

IT is the set of interesting visiting targets in a workflow, for example, IT={library, restaurant, bus stop} in Alice's tour,

CO is the set of context during the execution of a workflow, and

$M_0=\{1,0,0,0,0,0,0,0\}$.

Table 2. Places and transitions of PN-CAWM

p_i/t_i	*Description*	*Meaning*
p_1	Active(IT_i)	sub-activity IT_i is initiated
p_2	Executing(IT_i)	IT_i is executing
p_3	Found($C_{i,j}$);	Candidate $C_{i,j}$ is found
p_4	Active(N_k);	Network N_k is active
p_5	Qualified(N_k);	N_k is qualified
p_6	Finished(IT_i);	A user gets navigation info. Of IT_i
p_7	Satisfied(IT_i);	A user satisfies the navigation info.
p_8	End	A workflow finishes
t_1	IT_i is initicated	
t_2	CAWM finds $C_{i,j}$ for IT_i	
t_3	CAWM inspects whether N_k is qualified	
t_4	CAWM requests navigation info. of $C_{i,j}$ via N_k	
t_5	whether a user satisfies navigation information for IT_i or not	
t_6	CAWM checks whether a workflow finishes	

6. Conclusions and future work

We have presented a context model and a context-aware workflow management algorithm for ubiquitous campus navigation. Our algorithm can adjust workflow execution behavior in terms of current context information. On the other hand, we also modeled the workflow management algorithm through Petri nets.

Based on the proposed workflow management algorithm, we developed preliminarily a ubiquitous campus navigation system, which integrates various heterogeneous networks (wired and wireless networks, sensor network, self-organized network, Bluetooth etc.). The system can provide the campus navigation information for users with the graphical and textual way based on user's current locations, network states and device capacity.

We are going to investigate how to mine and reason users' behavior model and further predict the users' actions based on the users' profiles and history data. Based on this, we will set up a context knowledge based user-centric intelligent campus.

7. References

[1] T.G.Kanter. HotTown, enabling context-aware and extensible mobile interactive spaces. *IEEE Wireless Communications* Oct. 2002; 9(5): 18 - 27.

[2] A.Ranganathan, R.H.Campbell, A.Ravi, A.Mahajan. ConChat: a context-aware chat program. *IEEE Pervasive Computing* 2002; 1(3): 51 - 57.

[3] D.Bottazzi, R. Montanari, A.Toninelli. Context-Aware Middleware for Anytime, Anywhere Social Networks. *Intelligent Systems* 2007; 22(5): 23 - 32.

[4] K.Rehman, F.Stajano, G.Coulouris. An Architecture for Interactive Context-Aware Applications. *IEEE Pervasive Computing* 2007 6(1): 73 - 80.

[5] Z.W. Yu, X.S. Zhou, D.Q. Zhang et al. Supporting Context-Aware Media Recommendations for Smart Phones. *IEEE Pervasive Computing* 2006 5(3): 68 - 75.

[6] K.Baïna, K. Benali, C. Godart. DISCOBOLE: A service architecture for interconnecting workflow processes. *Computers in Industry* 2006; 57: 768-777.

[7] D.Zou, H.Jin, F.Mao et al. Service organization and information consistency for service Grid. *Int. J. Web and Grid Services* 2005; 1(1): 69-86

[8] G.E.Graham, D.E.Bertram. Mcrunjob: A high energy physics workflow planner for grid computing. *High Energy and Nuclear Physics* 2003; 3

[9] S.Krishnan, P.Wagstrom, G.V. Laszewski. GSFL: A Workflow Framework for Grid Services. *Available from: http://www-unix.globus.org/cog/papers/gsfl-paper.pdf.* 2002.

[10] J.Nichols, H.Demirkan, M. Goul. Autonomic workflow execution in the grid. *IEEE Transactions on Systems, Man and Cybernetics, Part C* 2006; 36(3): 353-364.

[11] Y.Wu, S.Wu, H.Yu et al. CGSP: An Extensible and Reconfigurable Grid Framework. *Proceedings of the Sixth International Workshop on Advanced Parallel Processing Technologies (APPT 2005)*, 2005, pp. 292-300.

[12] I.Choi, C.Park, C.Lee. Task net: Transactional workflow model based on colored Petri net. *European Journal of Operational Research* 2002; 136: 383-402.

[13] Y.Y.Cho, J.S. Choi, J.Y.Choi. A Context-Aware Workflow System for a Smart Home. *Proceedings of International Conference on Convergence Information Technology*, Nov. 2007, pp.95 - 100.

[14] F.S. Hsieh. Context-aware Workflow Driven Resource Allocation for e-Healthcare. *Proceedings of 9th International Conference on e-Health Networking, Application and Services*, June 2007, pp.34-39.

[15] J.Yu, Y.V.R. Reddy, S.Sentil et al. A workflow-centric, context-aware collaboration framework *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design*, May 2005, pp.514-519.

[16] F.L. Tang, M.Y. Guo, M.L. Li. An Adaptive Context-Aware Transaction Model for Mobile and Ubiquitous Computing. *to appear in Computing and Informatics,* 2008.

[17] W. Mark. Turning pervasive computing into mediated spaces. *IBM Systems Journal* 1999; 38(4): 672-692.

[18] F.L. Tang, M.L. Li, Joshua Z.X. Huang et al. Real-time transaction processing for autonomic Grid applications. *Engineering Applications of Artificial Intelligence,* 2004, 17(7): 799-807.

[19] J.L.Peterson. Petri nets. *Computing Surveys,* 1977, Vol. 9, pp. 223-252.

[20] T.Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 1989, pp.541-580.