Performing Workflows in Pervasive Environments Based on Context Specifications

[1]Xiping Liu and [1,2]Jianxin Chen

1Institute of Computer Technology, College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China, 210003
liuxp@njupt.edu.cn
2Gradiant ETSI Telecomunicación, Lagoas Marcosende s/n, Vigo Pontevedra, España, 36310.jianxinchen06@gmail.com

Abstract

The workflow performance consists of the performance of activities and transitions between activities. Along with the fast development of varied computing devices, activities in workflows and transitions between activities could be performed in pervasive ways, which causes that the workflow performance need to migrate from traditional computing environments to pervasive environments. To perform workflows in pervasive environments needs to take account of the context information which affects both the performance way of activities and the transition control between activities. Accordingly, a framework is described in this paper based on definitions of activity-context and transition-context to performing workflows in pervasive environments. After that, a context specifying method, in which activity-context and transition-context could be represented through simple or composite context, is provided to append necessary context on workflow definitions. Furthermore, corresponding performance principles are proposed, including capturing relevant actual context value according to given specifications, mapping captured context information with corresponding specifications based on presented context mapping algorithm, and making appropriate response according to predefined reaction rules relevant to matched context specifications. Moreover, we also illustrate our work through a case study about the instant meeting on call.

Keywords*: workflow performance, activity-context, transition-context, context mapping*

1. Introduction

Workflow technology, which was originally emerged for the automation of business processes, is usually deployed in desktop computing environments [1]. Along with the fast development of varied computing devices, activities in workflows and transitions between activities could be performed in pervasive ways [2]. Therefore, as an important technology to coordinate the control and the data relationships among activities in a process, workflows need to migrate from traditional computing environments to pervasive environments.

One of key issues in the migration is to adapt to the change of performance ways. The change of performance ways includes two aspects. One is that a single activity could be performed at different places rather than comparatively fixed locations, or performed by different types of devices except for traditional desktop computers, etc. The other is that a transition between activities could be triggered by kinds of environment factors, i.e., transition conditions would consider certain environment factors except for traditional factors relevant to the task logic.

Sequentially, to perform workflow in pervasive environments should take two kinds of context into account, i.e. activity-context and transition-context, to adapt to the change of performance ways. The activity-context is used to determine the way in which an activity in a workflow is performed. The transition-context is used to control whether transition conditions are satisfied and what the next activity should be. During the performance of workflows, the workflow engine should capture both the activity-context and the transition-context, and make correct responses to the corresponding context.

More and more researchers have worked on the context-aware workflow performance. Montagut et al. [3-4] propose an adaptive transactional protocol to support the execution of workflow in pervasive setting. Joohyun Han et al. [5-7] propose a language uWDL to specify the context information on transition constraints to support the performance of adaptive service in workflows. Jinqiao Yu et al. [8-9] propose a generic framework EkSarva to support people's generic collaboration in heterogeneous computing environments. However, few existing researches put efforts on both kinds of context which affects the workflow performance in pervasive environments. The dynamic partner assignment presented by Montagut [3-4] focuses on the partial activity-context; uWDL presented by Joohyun Han [5-7] mainly handles the transition-context; and Jinqiao Yu [8-9] pays little attention to the context relevant to transitions between activities.

Differing from those researches, this paper proposes a dual-context based workflow performance framework applied in pervasive environments, which considers both activity-context and transition-context. The rest of this paper is organized as follows. Section 2 presents a motivating example to illustrate a simple scenario of the workflow performance in pervasive environments. Section 3 proposes a framework with two main parts, one is relevant to the modeling phase and the other is relevant to the performing phase. The details about the two parts are described respectively in section 4 and section 5. Consequently, section 6 discusses a case study derived from the motivating example based on above work. Finally, we conclude this paper at section 7.

2. A motivating example

A motivating example about an instant meeting on call is described in this section, which will be used throughout the paper to illustrate an application scenario of our framework.

Suppose there is a project handled collaboratively by several departments of a company. During the project development, it is high possible that the project manager need to convoke employees relevant to the project (called as members) to discuss some important urgent issues at any moment. Such a process could be defined as a simple workflow showed in Figure 1. Three members are supposed to be involved in this project while the case with more members could be deductive in the similar way. Once the manager decides to call an urgent meeting, he or she will enable the workflow performance, which begins from the activity of sending the meeting calls to each member. As long as responses from all members are received, the workflow engine will control the meeting activity to be performed after it detects that the situation of each member and the manager is ready for the meeting. After everyone quit from the meeting, the workflow performance is finished.

In the traditional case that people work in a fixed local place, the performance of such a workflow is simple. As long as the manager says a word in the office, everyone will gather in a meeting room and commence the meeting. However, the performance in pervasive environments is comparatively complex. As everyone could work in a pervasive way, a simple activity could be performed in varied patterns. Besides, the detection on transition conditions would depend on the real situation in pervasive environments. All of these bring great challenges to the control and management on the workflow performance. For example, imagine Tom is on a business trip at another city and Amy is nearby the company when the sending activity is being performed. The workflow engine needs to check the online contact way of them and arrange the meeting call is sent through certain instant messaging tools, or mobile phones, or other real-time contact ways. Moreover, Tom maybe attends the meeting through the video transferred on Internet. Amy maybe joins the meeting through PDA at first and moves to the company at the same time, and then presents herself in the meeting room. Therefore, considering each member's different situation, the workflow engine needs to detect in different ways whether everyone is ready for the meeting.

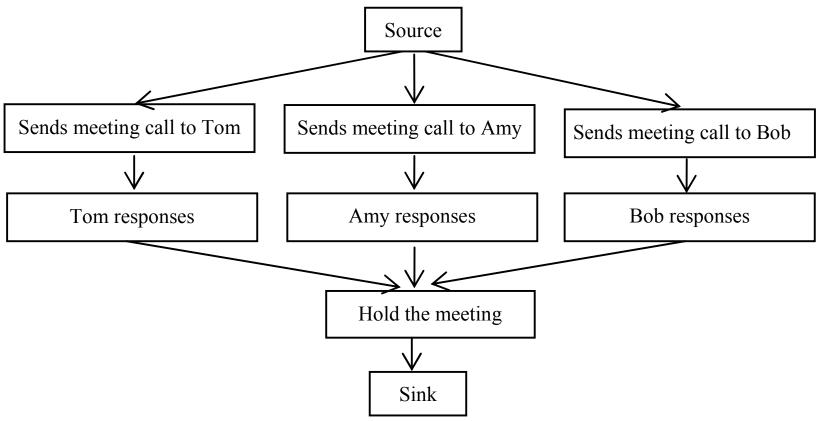


Figure 1. The workflow definition about the instant meeting on call

Through this simple workflow, we can see kinds of context information need to be sensed and responded respectively. It often involves much more complex context information to be handled during the workflow performance in pervasive environments. However, such context information could be basically classified into activity-context and transition-context. With these considerations, this paper proposes a dual-context based workflow performance framework in pervasive environments, which is introduced in detail at following sections.

3. The dual-context based workflow performance framework

The motivating example given in section 2 illustrates two generic kinds of context relevant to the workflow performance in pervasive environments to some extent. The details are presented at following definitions.

Definition 1 (Activity-context). The activity-context means the context information used to determine the way in which an activity in the workflow is performed, i.e., how an activity performs, who or which device is relevant to the activity, where the activity performs, and etc.

Definition 2 (Transition-context). The transition- context means the context information used to control the progress of the workflow performance, i.e., what the next activity is and when it performs.

In the motivating example, the online contact information of each member is an example of activity-context, which determines the varied ways of sending the meeting call. Besides, the ready state information related to all members is an example of transition-context. Only when the workflow engine detect everyone is ready for the meeting, the workflow would run forward to the "meeting" activity. Sometimes, the two kinds of context maybe influence each other. E.g., the transition-context relevant to "hold the meeting" activity would imply the activity-context under which members attend the meeting.

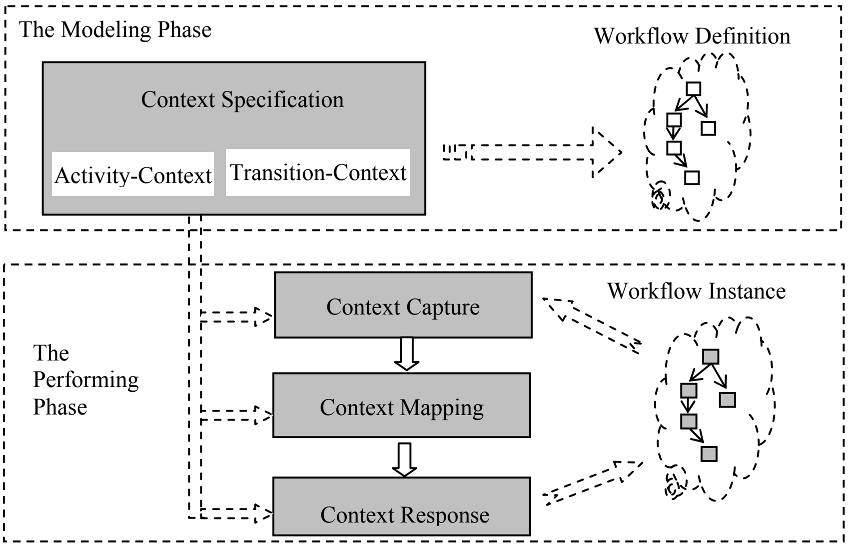


Figure 2. The dual-context based workflow performing framework

Both of activity-context and transition-context need to be considered for the workflow performance in pervasive environments. Figure 2 shows our proposed dual-context based workflow performance framework. Firstly, both kinds of context relevant to the workflow performance should be specified at the workflow modeling phase, also with reaction rules corresponding to certain context, as showed in the context specification module of Figure 2. Secondly, during the performance, the context capture module should proactively capture the actual context information relevant to the current activity or transition according to the specifications. Thirdly, the actual context generated during the performance should be compared with predefined context specifications through the context mapping module. Lastly, the context response module would make proper actions based on the matched context and corresponding rules. The more detailed discussion about each component will be presented at section 4 and section 5.

4. The dual-context specifying method

To enable the automatic performance adapted to varied situations in pervasive environments, relevant context information must be pre-specified exactly and clearly. The presentation of the relevant context is concerned with the workflow definition model. There are lots of workflow definition methods, such as Petri Nets based ones [10], directed graph based ones [11-12], UML based ones [13], etc. As this paper does not focus on the modeling of workflow definition, we just choose one of those methods, i.e. directed graph, to represent workflows. Note that whatever the workflow definition method is, our dual-context specifications could be appended on the workflow definition without too much extra effort.

As mentioned before, both activity-context and transition-context need to be specified based on workflow definitions. In essence, the representation method of both kinds of context is similar, while the main difference between them is how the context is used to control and mange the performance of workflows.

Context relevant to the workflow performance in pervasive environments usually includes kinds of information, such as location, time, device equipment, temperature, kinds of application-specified one, and so on. Besides, context often concerns with certain entity such as a person. Therefore, we represent a simple context through a triple with three elements, i.e., the entity, the attribute, and the value. The simple context could be connected through logic operations, such as AND, OR, NOT, to compose more complicated context. Following definitions give more exact descriptions.

Definition 3 (Ontology library). It provides the information of basic elements in the simple context and is denoted as $\mathcal{O}=<\mathcal{E}, \mathcal{A}, \mathcal{V}>$, where $\mathcal{E}=\{e_i \mid e_i$ represents the certain entity about which certain information need to be sensed during the workflow performance$\}$, $\mathcal{A}=\{ a_j \mid a_j$ represents the certain information type about certain $e_i\}$, $\mathcal{V}=\{ v_k \mid v_k$ represents the certain value of certain $a_j\}$.

Definition 4 (Simple context). It represents a certain situation in pervasive environments through a single triple. The set of simple context relevant to a workflow is denoted as $\mathcal{S}im\mathcal{C}=\{sim\_c_p=<e_i, a_j, v_k> \mid e_i \in \mathcal{E}, a_j \in \mathcal{A}, v_k \in \mathcal{V}\}$.

Definition 5 (Composite context). It represents a complex situation through a logic expression with one or more simple context triples. The set of composite context relevant to a workflow is denoted as $\mathcal{C}om\mathcal{C}=\{com\_c_q ::= NOT <com\_c_q> \mid <com\_c_q> AND <com\_c_q> \mid <com\_c_q> OR <com\_c_q> \mid sim\_c_p$, where $sim\_c_p \in \mathcal{S}im\mathcal{C}\}$. Note that a single sim_c_p is supposed to be simple context rather than composite context in the following sections.

Definition 6 (Activity-context Set and Transition-context Set). Action-context set represents the set of activity-context relevant to a workflow and is denoted as $\mathcal{A}ct\mathcal{C}=\{act\_\mathcal{C}_i \mid act\_\mathcal{C}_i$ includes all activity-context relevant to the *i*th activity in a workflow and $act\_\mathcal{C}_i \subseteq \mathcal{S}im\mathcal{C} \cup \mathcal{C}om\mathcal{C}\}$. Transition-context set represents the set of transition-context relevant to a workflow and is denoted as $\mathcal{T}ra\mathcal{C}=\{tra\_\mathcal{C}_i \mid tra\_\mathcal{C}_i$ includes all transition-context relevant to the *i*th transition in a workflow and $tra\_\mathcal{C}_i \subseteq \mathcal{S}im\mathcal{C} \cup \mathcal{C}om\mathcal{C}\}$.

To perform a workflow in pervasive environment, the workflow definition should be appended with specifications on the relevant context. Both activity-context and transition-context could be specified through the form of simple context or composite context. Each presented simple context is supposed to be reasonable, that is, e_i, a_j and v_k in a simple context triple is supposed to be correct in semantic logic. Besides identifying a certain situation through simple context or composite context, corresponding response actions should also be provided, which could be represented as reaction rules defined as follows.

Definition 7 (Reaction rule). It provides a context-based performance rule of a workflow in pervasive environments and the set of reaction rules is denoted as $\mathcal{R}=\{<c_i, action_j \mid c_i \in \mathcal{A}ct\mathcal{C} \cup \mathcal{T}ra\mathcal{C}$ and represents certain context. $action_j$ represents the concrete action description corresponding to certain context, including the participant, the allocation, the performing pattern of an activity, or the triggering of a transition, and so on$\}$.

Figure 3 illustrates the context specification for a workflow definition segment, where the number in the node means the *i*th activity in a workflow and the number besides the edge means the *j*th transition. Note that an activity or a transition might concerns with multiple rules. As rules with same type actions might be redundant or inconsistent, it would be necessary to predefine a preference order of those rules. The ontology library presents detailed information about numbered entities, attributions, and values for references. The real content of the ontology library is application-specific and could be constructed based on lots of researches relevant to ontology technology.

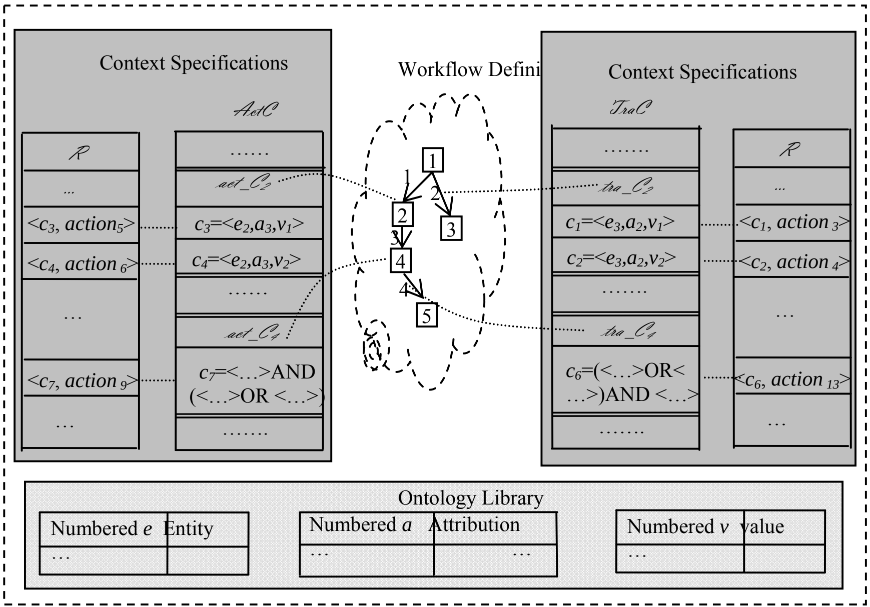


Figure 3. The dual-context specification example of a workflow definition segment

5. The dual-context based performance principles

With detailed dual-context specifications, the workflow engine could automatically control and manage the performance of workflows according to predefined workflow model and the real performance situation. After the workflow performance is commenced, the context capture module would check the activity-context specification on the first activity to determine what kind of context information should be captured proactively. The captured context would be transferred to the context mapping module to find out matched context specifications. Then the context response module would make proper actions according to reaction rules in which the matched context specifications locate. When the performance of an activity is finished, similar operations would happen to the transition-context appending on the edge adjacent to that activity. Such a dual-context based performance behavior would continue until the last activity of the workflow is completed. The more details are presented as follows.

5.1. Capture relevant context based on specifications

As both activity-context and transition-context could be simple context or composite context, we discuss the context capture mechanism without considering whether it is activity-context or transition-context.

To capture relevant context is a key step of the context based performance. Here, the relevant context is determined by both the progress of workflow instances and predefined context specifications, as showed in the framework in Figure 2. More specifically, when the workflow is performed at certain point of an activity or a transition, the workflow engine would check corresponding context specifications on the activity or the transition, i.e. certain act_C_i or tra_C_i, to decide which context need to be captured. Note that not every activity or transition has relevant context specifications. If certain act_C_i or tra_C_i, is Φ , the corresponding reaction rules could be formalized as <*null*, *null*>. At this time, the workflow engine just does what it usually does and pay little attention on the context aware and response procedure during the performance of this activity or transition.

As far as a simple context is concerned, e_i and a_j in certain simple context triple provide the relevant object and the type of the context information, which are determinate and unnecessary to be captured during the performance. However, what this kind of context about this object looks like is indeterminate, and this is the value v_k that the context capture module should find out. As far as the composite context is concerned, it is a logic combination of multiple simple context triples. Therefore, the content to be captured during the performance is the values of all simple context triples in the composite context expression. In other words, to a context specification (either simple context or composite context), to capture relevant context information means to capture the value v_k in one or more simple context triples $<e_i, a_j, v_k>$. This process could be implemented with the assistances of wireless sensor network technology, mobile devices technology, and so on, while the details are not the emphasis of this paper.

5.2. Map captured context with specifications

After the actual context information is captured from the performance scene, it would be sent to the context mapping module to search matched simple context or composite context specifications. The captured actual context information is associated with certain activity or transition, which limits the context specifications to be compared. Moreover, e_i and a_j in certain triple of actual context information is determinate. Therefore, the comparison operation mainly focuses on values in actual context information and ones in limitative specifications.

As mentioned in section 5.1, the actual context information is represented as a single simple context triple or the conjunction of multiple triples. There might be multiple matched context specifications (either simple context or composite context) corresponding to the actual context information. All of these matched ones could be found out based on the algorithm described in Figure 4. If the matched context specifications are found, they would be transferred to the context response module to make proper actions according to the corresponding predefined reaction rules.

If no context specifications are matched with captured actual context information, i.e., is Φ after running the context mapping algorithm, the engine would periodically repeat the procedure of catching and mapping until matched context specifications are found or given time is over. In case that matched context specifications can not be found in the given time, the engine could present the abnormal performance information and organize the corresponding decision making.

```
Algorithm: Context mapping.
Input: a set of captured context triples relevant to the ith activity or transition, denoted as T={t1, …, tm}.
Output: matched context specifications in act_Ci or tra_Ci //the set of matched context is denoted as MC
//find matched simple context
for each simple context triple sim_ci in act_Ci or tra_Ci
  do for each tj in T
        do if sim_ci=tj
           then add sim_ci in MC
              break
//find matched composite context
for each composite context com_ci in act_Ci or tra_Ci
  do matched←FALSE
     matched←MAP(com_ci)
     if matched=TRUE
       then add com_ci in MC
MAP(com_ci)
 if com_ci is a single simple context triple
  then for each tj in T
         do if com_ci=tj
               then return TRUE
        return FALSE
  else represent com_ci as a binary tree
     switch the root
       case 'NOT': return not MAP(left-subtree);
       case 'AND': return MAP(lef- subtree) and MAP(right-subtree)
             case 'OR': return map(left-subtree)or MAP(right-subtree)
```

Figure 4. Context mapping algorithm description

5.3. Make Proper Actions According to Matched Context

As mentioned before, both activity-context and transition-context could be simple context or composite context, which makes the capture and mapping method similar. However, the response actions corresponding to activity-context and transition-context are different to some extent.

As far as activity-context is concerned, it affects the way in which an activity in a workflow is performed. Multiple reaction rules might be provided to specify different performance actions corresponding to different context situations. The context response module should arrange participants of a certain activity to perform the activity under the directions of those matched specifications. E.g., for the activity "hold the meeting", varied locations of all members could be represented by several simple context triples, which specify whether a member should attend the meeting in which meeting room or through the way of video conference. Then, actions corresponding to varied location context would be accordingly deployed based on reaction rules, such as, automatically open devices in certain meeting room or connect service provider of video conference.

As far as transition-context is concerned, it affects the progress of the workflow performance. After the performance of an activity is completed, the next activity could be started only if the matched transition-context satisfies certain specifications. Sometimes, there are transitions without special transition conditions. Such transitions would happen naturally without considerations on context capture and handling.

Sometimes, there might be more than one matched context specifications for an activity or a transition, which means that more than one reaction rule will be applied. At this time, if actions in those rules corresponding to matched context specifications are unrelated with each other, i.e. they are not the same type, then all of them could be carried out. For example the reaction rule which determines the person to perform certain activity and the reaction rule which determine the place to perform certain activity could be all applied. However, if those actions are same type, then they might be redundant or even conflict. At this time, the context response module could deal with the situation according to the preference order predefined at the modeling phase.

6. A Case Study

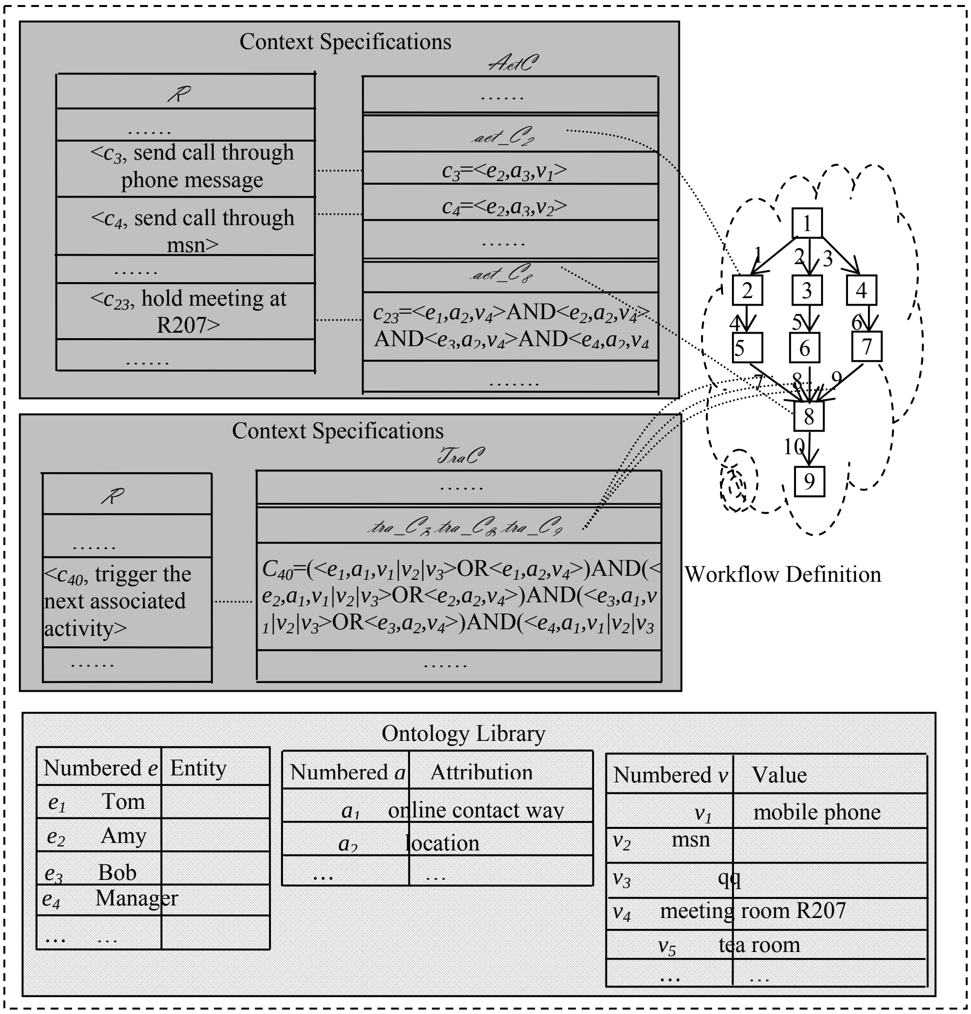


Figure 5. The dual-context specifications on the workflow definition about instant meeting on call

In this section, a case study based on the motivating example would be introduced to illustrate our workflow performance framework and principles. As Figure 5 showing, partial specification examples on both activity-context and transition-context are appended on the workflow definition shown in section2, where activities and transitions are represented by number correspondingly.

According to context specifications showed in Figure 5, the context capture module would capture proactively the online contact way of Tom when performing the activity "send meeting call to Tom". Then the context mapping module applies the proposed algorithm in section 5.2 to compare the captured actual context information with predefined specifications to find out the matched ones. As Tom might enable not only one contact way, it is possible to get more than one matched specifications. In this case, the meeting call would be sent through the preference way decided by predefined order. Moreover, the workflow would proceed forward the activity "hold the meeting" only when the context information of the real performance scene meets the requirement of the transition-context specification. Such transition-context is a composite context expression consisting of four simple context triples, where each one implies the ready situation of a project member. We use the triple $<e_i, a_1, v_1|v_2|v_3>$ to represent e_i is ready for meeting through the video conference, where three values represent three possible means to attend the meeting. If no video conference mean is available for e_i, he/she must be located at the meeting room, otherwise, e_i is not ready for the meeting and the transition could not happen. In this way, this workflow could be orderly performed in pervasive environments based on those two kinds of context which provide necessary information to manage and control the performance way of an activity and the triggering of a transition.

7. Conclusion and Future Work

To perform workflow in pervasive environments, both the performance way of activities and the triggering of transitions between activities would be affected by the real context information. Therefore, context relevant to both activity and transition need to be taken into account seriously. However, most current related work only focuses on one kind of context and lacks a comparatively general consideration on the context relevant to workflows in pervasive environment.

Based on definitions of activity-context and transition-context, this paper proposes a dual-context based workflow performance framework applied in pervasive environments. The framework consists of two main parts, one is for the workflow modeling phase, and the other is for the performing phase. At the modeling phase, context specifications appended on the workflow definition could be represented through our provided method. At the performing phase, the context capture module captures relevant actual context information according to predefined specifications. After that, the context mapping module compares the actual context with specifications to find matched ones through our proposed algorithm. Finally, the context response module takes proper actions based on matched specifications and reaction rules. Furthermore, a case study about the instant meeting on call is introduced to demonstrate our work.

In the future, we are going to further research the validation and rationalization of reaction rules, such as find possible conflict actions and propose corresponding solutions. Moreover, how to capture the value of certain attribution of certain entity is another research issue.

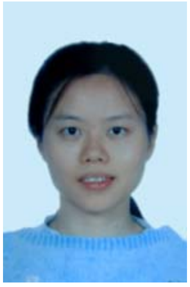
Acknowledgments

This paper is partially published in proceedings of UNESST09 and partly supported by National High Technology Research and Development Program of China under Grant No. 2009AA12Z219, Foundation of Jiangsu Educational Committee under Grant No. 08KJD520024, and Foundation of NJUPT under Grant No. NY207138.

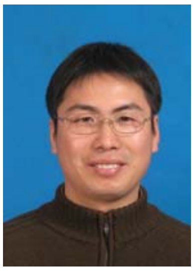
References

[1] David Hollingsworth, The Workflow Reference Model, Technical report [WFMC-TC-1003], Issue 1.1, Workflow Management Coalition, 1995.
[2] Dipanjan Chakraborty, Hui Lei, "Pervasive Enablement of Business Processes", Proc. Of 2nd IEEE Annual Conference on Pervasive Computing and Communications, 2004, pp. 87-97.
[3] Montagut F, Molva, R, "Enabling Pervasive Execution of Workflows", Proc. of Collaborative Computing: Networking, Applications and Work sharing, 2005, pp. 1-10.
[4] Montagut F, Molva R, "The Pervasive Workflow: A Decentralized Workflow System Supporting Long-Running Transactions", IEEE Transactions on Systems, Man, and Cybernetic-Part C: Applications and Reviews, Vol. 38, No. 3, 2008, pp. 319-332.
[5] Han J, Cho YY, Choi J, "Context-aware workflow language based on Web services for ubiquitous computing", O. Gervasi et al. (eds.): ICCSA. Lecture Notes in Computer Science, Vol. 3481, 2005, pp. 1008-1017.
[6] Cho YY, Han J, Choi J, Yoo CW, "A uWDL handler for context-aware workflow services in ubiquitous computing environments", T. Enokido et al. (eds.): EUC Workshop. Lecture Notes in Computer Science, Vol. 3823, 2005, pp.131-140.
[7] Han J, Cho YY, Kim EH, Choi J, "A Ubiquitous Workflow Service Framework", M. Garvrilova et al. (eds.): ICCSA. Lecture Notes in Computer Science, Vol. 3983, 2006, pp. 30-39.
[8] Yu J, Reddy YVR, Selliah S, Bharadwaj V, Reddy S, Kankanahalli S, "The Design of a Workflow-Centric, Context-Aware Framework to Support Heterogeneous Computing Environments in Collaboration", Y. Luo (ed.): CDVE. Lecture Notes in Computer Science, Vol. 3675, 2005, pp. 22-29.
[9] Yu JQ, Reddy YVR, Bharadwaj V, Reddy S, Kankanahalli S, "Workflow-Centric Distributed Collaboration in Heterogeneous Computing Environments", W. Shen et al. (eds.): CSCWD. Lecture Notes in Computer Science, Vol. 3865, 2006, pp. 504-515.
[10] W.M.P. Van der Aalst, "The application of Petri Nets to workflow management", J. Circuits Syst. Comput., Vol.8, No.1, 1998, pp. 21-66.
[11] Liu XP, Dou WC, Chen JJ, Fan SK, Cheung SC, Cai SJ, "On Design, Verification, and Dynamic Modification of the Problem-Based Scientific Workflow Model", Simulation Modelling Practice and Theory, Vol. 15, No. 9, 2007, pp. 1068-1088.
[12] Shazia W. Sadiq, Maria E. Orlowska, Wasim Sadiq, "Specification and Validation of Process Constraints for Flexible Workflows", Information Systems, Vol. 30, No. 5, 2005, pp. 349-378.
[13] Ricardo M. Bastos, Duncan Dubugras A. Ruiz, "Extending UML activity diagram for workflow modeling in production systems", Proc. of the 35th Hawaii International Conference on System Sciences, 2002, pp. 3786-3795.

Authors



Dr. Liu received her Ph. D. degree from in computer application from Nanjing University in 2007. She has worked in the institute of computer technology, and college of computer at Nanjing University of Posts and Telecommunications since 2007. Her research interests include workflow technology, cooperative computing, and pervasive computing.



Dr. Chen received his Ph. D. degree in communication and information system from Shanghai Jiaotong University (China) in 2007. Then He works as a lecturer in the college of Computer at Nangjing University of Posts and Telecommunications. From the May of 2008 to August of 2009, Chen did postdoctoral research in IPP Hurray Research Group, Polytechnic Institute of Porto (Portugal). After that, Jianxin Chen works as a researcher in Gradiant ETSI Telecommunication (Vigo, Spain). His research interests include pervasive sensing, embedded system, and communication algorithm design.