An Architecture for Reliable Mobile Workflow in a Grid Environment

Bill Karakostas
Centre for HCI Design, School of Informatics
City University
London, UK
billk@soi.city.ac.uk

George Fakas
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
g.fakas@mmu.ac.uk

Abstract— **Mobile peer to peer (P2P) computing is becoming a major revolution in computing owing to advances in computing power, network connectivity and storage capacity of mobile devices. Mobile workflow is, however, characterized by the unreliability of mobile peers who can drop out of the network due to sudden loss of connectivity, drained battery and so on, thus making problematic the guaranteed completion of Workflow Processes in accordance with deadlines. Existing research in P2P Workflow Management has not considered the problem of guaranteeing workflow process completion in an unreliable mobile peer environment. This paper proposes a P2P Workflow Management Architecture for the distributed definition and execution of workflows in a grid environment of mobile peers. The architecture is based on the use of proxies, i.e. fixed grid nodes with which mobile peers register and which act as workflow coordinators. Proxies employ statistical reliability measures of mobile peers in order to calculate the redundancy in the number of peers required to participate in a workflow. The performance of the proposed architecture is illustrated in a journalism scenario.**

Keywords- Grid Computing, Proxy, Mobile P2P Computing, Mobile Workflow Management, Dependability.

I. INTRODUCTION

Grid computing facilitates the sharing of computer resources and services by direct exchange between systems. Users can take advantage of existing desktop computing power and networking connectivity, allowing clients to wield their collective power in order to benefit the a wider community. Peer to Peer (P2P) computing is related to grid computing but is characterized by less centralization of resources.

P2P is increasingly becoming an important technique in grid computing. Workflow Management, the automation of business, scientific, etc processes, can benefit from the P2P paradigm in many ways. For instance, better reliability can be achieved because the execution of workflows does no longer depend on a single centralized server (i.e. even when some mobile peers are down the workflow execution may still continue). Reliability can also be improved by employing redundancy methods. P2P computing can also facilitate remote access to resources and services in workflows. Finally, scalability can be achieved because it is relatively easy to add any number of workflow mobile peers where needed.

A P2P Workflow Management Architecture is ideal for systems where dependability, reliability and remote sharing of resources are critical requirements. This paper addresses the problem of reliability of workflows that are executed by mobile peers, by utilising mobile peer redundancy techniques. In such settings, mobile peers are not necessarily dependable as they may leave or join the network at any time. Redundancy techniques can be applied to P2P or Grid environments such as SETI@home where the number of potentially available mobile peers is very large.

The types of grids that we consider in this paper comprise nodes that have fixed network addresses and are relatively stable and offer services such as registration, authentication, registry and workflow management. Mobile grid nodes (that we call *peers* to emphasize their status in the grid) do not have fixed network addresses, but they participate by connecting to and registering with fixed nodes called *proxies*.

This type of grids could represent for example communities of journalists, where a core group of publishers, editors are supported by a large number of freelancers, or even 'citizen journalists' that use mobile devices to participate in workflows to produce some publication such as a news bulletin.

Previous research in this area (e.g. [1] [2] [3] [4]) has focused on definition and execution of distributed workflows. These approaches, however, do not address adequately dependability or reliability issues.

The contributions of this paper to research in grid and mobile workflow management are as follows:

- Use of proxies [15] to allow mobile peers to participate in grid workflows and deal with their inherent reliability problems (such as frequent and unpredictable disconnections, network migration etc)
- Use of mobile peer redundancy policies to improve workflow process dependability i.e. ability to execute according to deadlines. The proposed system is based on redundant work allocation to mobile peers, so as to increase overall system dependability to meet deadlines. It employs a probabilistic approach in order to estimate the required minimum redundancy on the basis of the probabilities of mobile peers' availability.

The rest of this paper is organized as follows. Section 2 contains a detailed discussion of research issues and existing work pertaining to P2P Workflow Management Architectures. Section 3 describes the proposed architecture, whereas section 4 presents a method for estimating the required peer redundancy. In section 5, a case study and experimental validations of the proposed architecture are presented. Finally, section 6, summarizes the research results and suggests future work.

II. RELATED WORK

This section discusses research issues related to P2P Workflow Management and compares existing approaches.

A. *Research Issues in Distributed and P2P Workflow*

In pure P2P workflows, the definition of the workflow process must be shared among the mobile peers so that each mobile peer can act autonomously towards the execution of the workflow. Service orchestration languages, such as BPEL4WS, WSFL, XLANG, although they support some degree of P2P interaction among partners, do not facilitate the definition of completely decentralized workflows.

Mobile peer unavailability is one of the key characteristics of P2P computing, as mobile peers may not always be available and often leave the network unexpectedly. The workflow management system therefore must be able to detect mobile peers' unavailability and act accordingly.

P2P communities are dynamic, in the sense that mobile peers can join or leave at any time. Workflow mobile peers therefore must be able to advertise their services and also discover other mobile peers and their services dynamically. P2P systems such as Gnutella, P-Grid, Chord and others have proposed solutions to this problem.

Workflow scheduling policies is an active research issue, because P2P environments are mutable, as mobile peers may join or leave, or even dynamically add, remove or alter their provided services. P2P Workflow Management Architectures therefore, are characterized by mobile peer unreliability and they require mechanisms that will facilitate the timely execution of workflow activities and process.

This means that the set of mobile peers to which work can be allocated is dynamic, and that the real time discovery of such peers is very important. Existing work in the area proposes a solution that clusters the available mobile peers on the basis of their capabilities (forming virtual communities) to complete the current workflow activity and their workload [1] [2]. In [8], a multi-attribute peer selection policy is proposed. However, these policies do not consider any temporarily unavailable (previously discovered) mobile peers that are capable of executing the work. In contrast, this paper argues that the opportunity to allocate work to off-line users should not be discounted. Our approach deals with the problem of temporarily unavailable peers by using proxies that know which mobile peers are online and which are temporarily unavailable.

B. *Existing P2P, Distributed and Grid Workflow Management Systems*

Several P2P, Web Service orchestration and Grid Workflow Management Systems have been proposed. Table I summarizes those that have addressed workflow reliability and dependability issues.

TABLE I. APPROACHES TO RELIABLE P2P WORKFLOW

WORKFLOW MANAGEMENT SYSTEM	RELIABILITY/DEPENDABILITY TECHNIQUE
Exotica/FMQM [5] [6]	Persistent Messaging
METEOR [10] [11]	Dynamic Workflow Definition
SpiderNet [13]	Proactive recovery on mobile peers failures
[7]	Data Replication

Several approaches to increase the overall reliability of the system have been proposed. *SpiderNet* [13] is a failure resilient P2P service composition framework that provides proactive failure recovery (by maintaining backup composition scenarios) to overcome dynamic changes such as mobile peers departures. *Self-Serv* [8] [9] facilitates a P2P Web Services composition orchestration model through statecharts, whereby the responsibility of coordinating the execution of services is distributed across several software components called coordinators.

METEOR [12] increases reliability by allowing the specification of dynamic changes to running instances of Workflow Definitions (which facilitates adaptation and evolution) and also by allowing the handling of exceptions. The approach described in [5] enhances fault tolerance by using persistent messaging, exception handling and data replication.

The work described in [7] improves data availability in unreliable P2P data sharing systems by employing a dynamic replication model based on Bernoulli trials. Their method of computing replicas of a file is based on the probability of mobile peer's availability and is similar to our approach for computing redundancy. However none of the above work addresses P2P workflow reliability caused by mobile peers unavailability due to network disconnections, failure etc.

III. A MOBILE WORKFLOW MANAGEMENT ARCHITECTURE

This section describes the proposed architecture. Mobile workflow peers (MWPs) are mobile devices that can connect to grids and participate in workflows. Grid Proxies (GPs) are fixed grid nodes with sufficient resources to allow mobile peers to connect to and participate in workflows on the grid. GPs act on behalf of MWPs in workflow processes.

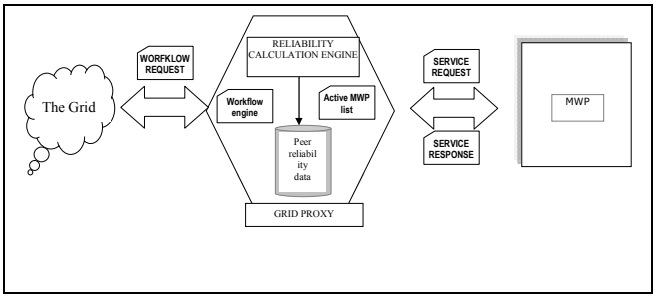

Figure 1: Architecture of the mobile grid workflow system

Because MWPs may be part of different networks they cannot communicate directly with each other, or with other grid nodes; instead they communicate only with the GPs. Each MWP can be in one of the following states:

- Connected (i.e. on line) i.e. in a state where it can receive and execute work;
- Unavailable (i.e. off line or otherwise busy) i.e. in a state where it is not available to receive and execute new work.

A GP keeps track of connected MWPs using a 'heartbeat' protocol that involves a periodic sending of a message to a MWP and waiting for a response. If no response is received within a timeout period, the GP considers the MWP unavailable. For each MWP, the GP maintains an index of its reliability, based on the number of disconnections over a certain period. A GP also calculates as the average reliability of the registered MWPs

GPs advertise to the grid the services of MWPs that have been registered with them. A GP receives an invitation to participate in a grid workflow by providing a service. The GP can accept the invitation to execute a workflow activity by delegating its execution to a registered MWP. The MWP remains transparent as far as the workflow process and the other grid participants are concerned, as all the interaction of the workflow process is with the GP (Figure 1).

However the GP relies on its connected MWPs to actually perform the activity. GP cannot guarantee at the time of the service request, that a suitable, registered MWP will be online. Instead, a GP must perform a reliability calculation to determine the degree of redundancy required to meet the workflow deadlines. This means that the GP will delegate the execution of the activity to multiple MWPs to ensure that one of those will be available to provide the service with an acceptable probability. This method is described in the following section.

IV. A PEER REDUNDANCY CALCULATION METHOD

In this section, we show how we use redundancy of MWPs in order to improve availability, and therefore facilitate the timely execution of workflows.

Let a process P consist of a sequence of activities $A_1,..,A_n$, where activity A_i starts at time t_{si} and needs to be completed at time t_{ei}. Let also assume that MWPs are not always available to complete A_i. Then, in order to ensure the completion of task A_i at time t_{ei} with probability π_i, the GP needs to employ a redundancy policy and assign task A_i to multiple MWPs.

Next, we describe how we can calculate the minimum required redundancy r_i of MWPs in order to have a Workflow Activity A_i completed by a required deadline d_i with some specified required probability π_i. More precisely, for an Activity A_i, if, $P_{i1}..P_{iM}$ is the set of MWP capable of executing the Activity, and MWPs $P_{i1}..P_{iM}$ are available with probability $q_1...q_M$; then r_i is the minimum redundancy number of MWPs required for the completion of the activity with probability π_i. We use Bernoulli trials to calculate r_i.

Let
$\mathbf{q_j}$= P(P_{ij} is available to complete the Activity A_i),
$\mathbf{p_j}$= P(P_{ij} is unavailable to complete the Activity A_i)=1- q_j,

$\boldsymbol{\rho}=\prod_{i=1}^{r} p_i$ = P(all MWP allocated to complete Activity A_i are unavailable), where r is the number of MWP (out of $P_{i1}..P_{iM}$) that were allocated to compete A_i ($r \leq M$) and
$\mathbf{P(n)}$= the probability that the Activity is completed at time n then,
$\mathbf{P(n)}= (1- \rho)\ \rho^{n-1}$

Now, let us assume that
$\mathbf{P_{Ai}(n)}$= the probability that the Activity is completed by time n, therefore,

$$P_{Ai}(n)=\sum_{i=1}^{n} P(i)=\sum_{i=1}^{n} (1- \rho)\ \rho^{i-1}= (1- \rho)(1-\rho^n)/(1- \rho)= 1-\rho^n,$$

therefore,

$$P_{Ai}(n)=1-(\prod_{i=1}^{r} p_i)^n$$

Now, if we assume that $n= d_i$ and for simplicity that
$p_i= p_j= p\ \forall i,j$ then $\rho= p^r$ and therefore
$P_{Ai}(d_i)= 1-p^{r \cdot di}$,
since we require $P_{Ai}(d_i) \geq \pi_i$ then,
$1-p^{r \cdot di} \geq \pi_i \Rightarrow p^{r \cdot di} \geq 1-\pi_i \Rightarrow d_i \cdot r \cdot \log(p) \geq \log(1-\pi_i) \Rightarrow$

$$r \geq \frac{\log(1-\pi_i)}{d_i \cdot \log(p)}$$

therefore, after rounding to the next higher integer

$$r_i=\left\lceil \frac{\log(1-\pi_i)}{d_i \cdot \log(p)} \right\rceil \qquad (1)$$

Table II shows the values of r_i for values {0.1...0.8} for p, {1…10} for d_i and {0.9, 0.7} for π_i. However, the calculation of r_i is based on the assumption that there is a big population of MWPs capable of executing A_i. This is often true in big Grid or Web P2P communities (e.g. SETI@home); however there is a possibility that $r_i > M$ (where $M=|P_{i1}.. P_{iM}|$). In this case at least, we know the

achievable d_i with M MWPs so we can design our Workflow Processes accordingly. For instance, if p=0.8, d_i=3 and π_i=0.9 then r_i=4; however if M=3 then we know that this deadline cannot be met. With M=3 the best deadline we can meet is 4 since r_i=3 for p=0.8, d_i=4 and π_i=0.9.

TABLE II. $\mathbf{R_i}$ FOR π_i =0.7, 0.9

		p							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
d_i	1	1, 1	1, 2	1, 2	2, 3	2, 4	3, 5	4, 7	6, 11
	2	1, 1	1, 1	1, 1	1, 2	1, 2	2, 3	2, 4	3, 6
	3	1, 1	1, 1	1, 1	1, 1	1, 2	1, 2	2, 3	2, 4
	4	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2	1, 2	2, 3
	5	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2	2, 3
	6	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2	1, 2
	7	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2
	8	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2
	9	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2
	10	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 2

V. EXPERIMENTAL VALIDATION AND CASE STUDY

In this section, we present an experimental validation method for the proposed architecture, using a case study from the domain of mobile journalism. Newspaper and magazine publishers typically use the services of journalism (either normal employees or freelancers) for the coverage of various events (such as a sports event, a natural disaster and so on). Such journalists are by necessity mobile as they need to be present where the event takes place. Unpredictable situations (for example disruptions in telecommunications, congestion of the networks, or failure of their mobile devices) mean that such journalists present the characteristics of mobile peers. Because of the unpredictable availability of mobile journalists, a redundancy policy must be employed, i.e. several mobile journalists may be allocated the same task.

When a news editor commissions a new article about an event, he/she will request the assistance of mobile journalists that will provide input such as photographs, video or commentary to be used in the article or broadcast. As there are usually strict deadlines for the production of such articles or broadcasts, an editor needs to ensure that all material will be made available before the deadline. An editorial workflow will therefore consist of a number of activities $A_1,\ldots,A_n$. Some of these activities need to be performed by mobile journalists (MWP) (e.g. $P_{i1,\ldots,}P_{iM}$). Let's assume that these MWPs are unavailable to complete the Activity with probabilities $p_1,..,p_M$. In the following steps, for the sake of uniformity, we use a common p for all MWPs (i.e. $p_j = p$ for each j). We use a discrete time approach and we calculate the dependability of the workflow by failing each MWP with a probability p at each time unit and then by calculating the required redundancy in order to meet deadlines. We measure the dependability of the system on the basis of two criteria:

a) the **Execution Time *ET*** of Workflow Process Instances i.e. by measuring the delays caused to the execution time of a Workflow Activity because of MWP failures;
b) the ***Reliability Index R,*** i.e. by measuring the average percentage of Activity Instances which have met their deadlines timeouts.

Experiment Parameters.

The following parameters and terms are used in the experiments:

d_i: the deadline of Workflow Activity A_i;

od: the deadline of a Workflow Process, (if we assume that Processes consists of sequential Activities then $od=\sum_{i=1}^{N} di$);

π_i: the probability that A_i will be completed by deadline d_i;

p: the probability of a MWP unavailability. For each experiment we use a common ***p*** for all MWP participating in the experiment (e.g. 10%, 20%, 30%);

***M*:** the number of MWPs that will participate in a Workflow Process (i.e. $|P_1.. P_M|$) or Activity (i.e. $|P_{i1}.. P_{iM}|$)

N: the number of Workflow Activities comprising a Workflow Process Definition (i.e. $|A_1.. A_N|$)

We define the following parameters

Def 1: r: Degree of Workflow Process Redundancy is the ratio of the total number of MWPs, M participating in a Workflow Process per Workflow Activities N as defined in the Process Definition (WPD), i.e. ***r =M/N.***

Def 2. r_i: Degree of Workflow Activity Redundancy of an Activity A_i is the number of MWP, such as M MWPs (P_{i1}, ...P_{iM}) responsible for the completion of the Activity A_i;

Measuring the Reliability of the Workflow system.

Next, we examine the effect of MWP unavailability and redundancy of work allocation in terms of Execution Time and Reliability on Workflow Process and Activity Instances. For the sake of simplicity, the Workflow Process Activities have a common deadline ***d*** (i.e. d_i =d for each i) for each experiment, and participating MWP fail with a common probability ***p*** (i.e. p_j =p for each j). For these experiments, a Workflow Instance is considered failed if it does not meet the deadline.

We provide the following definitions:

Def. 3: *The* ***Performance Index (PI_r)*** of the system is measured as the average number of delays per Workflow Process Instance where MWPs fail with a common probability **p** and the Degree of Redundancy is set to **r**.

Def. 4: ***Workflow Process Reliability$_r$ (PS_r)*** is the percentage of Workflow Process Instances without any workflow overall deadline (*od*) timeouts where MWPs fail with a common probability **p** and the Degree of Redundancy is set to **r**.

Def. 5: ***Workflow Activity Reliability$_r$ (AS_r)*** is the percentage of Workflow Activity Instances (i.e. out of $A_2..A_N$ since we assume A_1 is always competed successfully by the initiator) of a Workflow Process, i.e. without any deadline (d_i) timeouts of each A_i where MWPs fail with a common probability **p** and the Degree of Redundancy is set to **r**.

In these experiments, we construct 10,000 Process Instances of a simple mobile journalism workflow comprising 5 Activities ('start workflow', 'commission article', 'receive submission', 'edit submission', 'publish article'), i.e. where N=5. We run several experiments with different parameter settings (Table III) and we measure for each experiment the values of ***TP_r***, ***AS_r*** and ***PS_r***.

TABLE III. EXPERIMENT PARAMETER SETTINGS

r	1 ($r_1..r_5$=1, 1, 1, 1, 1), 1.4 ($r_1..r_5$=1, 2, 1, 2, 1), 2 ($r_1..r_5$=1, 2, 3, 2, 2)
p	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
d	2, 4

As expected, the results shown in figures 2 and 3, indicate that the more redundancy we use the better performance (i.e. TP_r , AP_r PS_r) results we obtain.

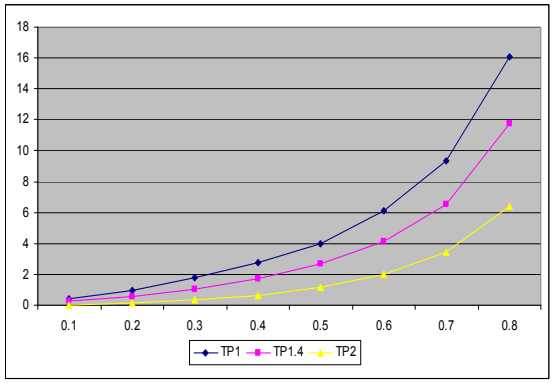

Figure 2. TP_1, $TP_{1.4}$, TP_2. Average Delays per Workflow Process Instance.

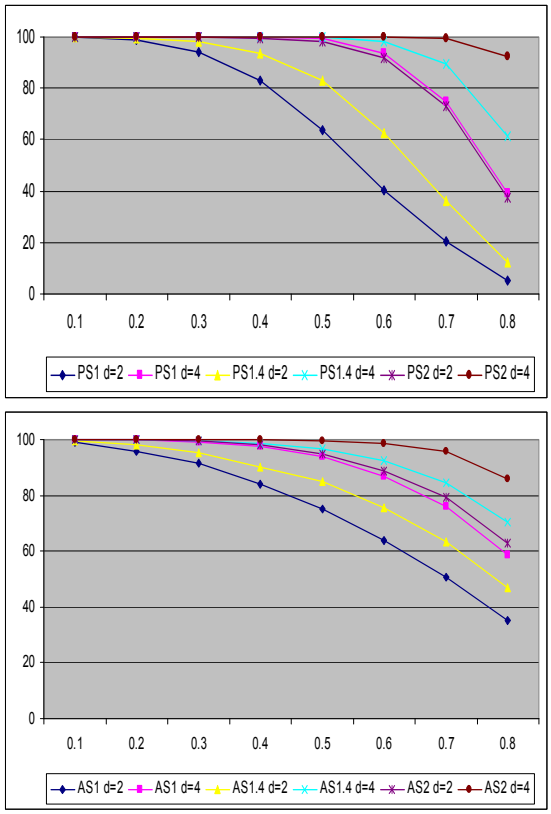

Figure 3. PS_1, $PS_{1.4}$, PS_2 (top) and AS_1, $AS_{1.4}$, AS_2 for d=2 and d=4 (bottom)

Calculating the parameters of the peer redundancy policy.

Next, we calculate how mobile peer redundancy can reduce or even eliminate Workflow Activities and Processes deadlines timeouts. In these experiments, we used a Workflow Process with 5 Activities (i.e. N=5) with deadlines 1, 4, 5, 3 and 1 time units (e.g. hours) respectively, and calculate the Workflow Process and Activity Reliability for 2 different values for π_i, (i.e. 0.9 and 0.7) and 2 different values for p (i.e. 0.5 and 0.8). Again, we ran 10,000 Workflow Process Instances for each experiment for the 4 different possible WPDs. Table IV shows the final WPDs for the 4 different combinations of values for π_i, and p.

Def 6: ***Workflow Activity Reliability (AS_i)*** *is the percentage of survivable Workflow Activity Instances of a Workflow Activity A_i, i.e. without any deadline (d_i) timeouts where MWPs fail with a common probability* ***p***.

Def 7: ***Workflow Process Reliability (PS)*** *is the percentage of Workflow Process Instances i.e. without any workflow overall deadline (od) timeouts where MWPs fail with a common probability* ***p***.

TABLE IV. THE WPDS FOR THE DIFFERENT VALUES OF π_i, AND P.

p	π_i	$WPD=\{WAD_1(1, d_1, r_1, \pi_1)..WAD_N(N, d_N, r_N, \pi_N)\}$:
0.5	0.9	(1,1,1,0.9), (2,4,1,0.9), (3,5,1,0.9), (4,3,2,0.9), (5,1,4,0.9)
0.5	0.7	(1,1,1,0.7), (2,4,1,0.7), (3,5,1,0.7), (4,3,1,0.7), (5,1,2,0.7)
0.8	0.9	(1,1,1,0.9), (2,4,3,0.9), (3,5,3,0.9), (4,3,4,0.9), (5,1,11,0.9)
0.8	0.7	(1,1,1,0.7), (2,4,2,0.7), (3,5,2,0.7), (4,3,2,0.7), (5,1,6,0.7)

Table V gives a summary of the simulation results for each activity A_i. AT_{Ai} is the average time unit required for the execution of each Activity A_i; which, as expected, is much lower than d_i, because it is the r_i that implicitly determines the longest time needed for the completion of the activity, and not the AT_{Ai}.

TABLE V. EXPERIMENTAL RESULTS FOR EACH WORKFLOW ACTIVITY

		p= 0.5, π_i= 0.9			p= 0.5, π_i= 0.7		
A_i	d_i	r_i	AT_{Ai}	AS_i	r_i	AT_{Ai}	AS_i
1	1	1	1	100	1	1	100
2	4	1	2.0	93.7	1	2.0	93.6
3	5	2	1.9	97.0	1	1.9	97.0
4	3	3	1.3	98.3	1	2.0	87.3
5	1	4	1.0	93.3	2	1.3	74.7

VI. CONCLUSIONS AND FURTHER WORK

This paper presented a novel Mobile Grid Workflow Management Architecture that utilizes a proxy pattern and reliability calculation techniques, to facilitate the reliable execution of workflows that involves mobile peers.

Critical issues that need to be further investigated include more accurate peer reliability calculation techniques. We also plan to develop a process definition tool that will make it easier for workflow managers to create and distribute automatically the workflow definitions to the mobile peers.

In addition, since the activity scheduling policy used in this work is rather static, we plan to investigate other scheduling policies based on reliability calculations that employ dynamic information from the peers as well as the environment, for example, network condition. Finally, we would like to apply the proposed architecture in other environments where reliability is a critical issue and participating mobile peers are mutable; e.g. emergency response systems.

REFERENCES

[1] J. Yan, Y. Yang and G. K. Raikundalia. A Data Storage Mechanism for peer-to- peer Based Decentralised Workflow Systems, Proc. 15th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE2003), San Francisco, USA, July 2003, 354-358.

[2] J. Yan, Y. Yang and G. K. Raikundalia. SwinDeW - A peer-to- peer based Decentralised Workflow Management System, IEEE transactions on systems, man, and cybernetics—part a: systems and humans, vol. 36, no. 5, September 2006

[3] G. Fakas and B. Karakostas. A peer to peer Architecture for Dynamic Workflow Management, Information and Software Technology, Elsevier, Volume 46, Issue 6, pp 423-431, 2004.

[4] G. Fakas and B. Karakostas. A peer to peer Dynamic Workflow Management Based on Web Services, Proceedings of the First International Conference on Web Services, ICWS'03, CSREA Press, pp 428-431, Las Vegas, Nevada, USA, 2003.

[5] G. Alonso, C. Mohan, R. Gunthor, D. Agrawal, A. El Abbadi and M. Kamat. Exotica/FMQM: a Persistent Message-Based Architecture for Distributed Workflow Management, Proc. IFIP Working Conf. Information System for Decentralized Organisations, Trondheim, 1995.

[6] G. Alonson, C. Hagen, D. Agrawal, A. E. Abbadi, C. Mohan. Enhancing the Fault tolerance of Workflow Management Systems, IEEE Concurrency, July-September 2000, pp 74-81, 2000.

[7] K. Ranganathan, A. Iamnitchi, and I. Foster. Improving Data Availability through Dynamic Model-Driven Replication in Large peer-to- peer Communities, CCGrid2002, IEEE Computer Society, pp. 376-381, 2002.

[8] B. Benatallah, M. Dumas, Q. Z. Sheng. Facilitating the Rapid Development and Scalable Orchestration of Composite Web Services, Distributed and Parallel Databases, 2005 Springer Science + Business Media, 17, pp. 5–37, 2005.

[9] B. Benatallah, Q. Z. Sheng, M. Dumas. The Self-Serv Environment for Web Services Composition, IEEE Internet Computing, 7(1), pp. 40-48, 2003.

[10] J. Miller, D. Palaniswami, A. Sheth, K. Kochut and H. Singh. WebWork: METEOR 2 's Web-Based Workflow Management System, Journal of Intelligent Information Systems, 10(2), pp. 185-215, 1998.

[11] Das, S., Kochut, K., Miller, J., Sheth, A., Worah, D. ORBWork: A Reliable Distributed CORBA-based Workflow Enactment System for METEOR, Technical Report, Department of Computer Science, University of Georgia, USA, 1997.

[12] J Cardoso, Z Luo, J Miller, A Sheth, K Kochut. Reliability Architecture for Workflow Management Systems, Proceedings of the 39th Annual ACM Southeast Conference, Athens, GA. pp. 207-216, 2001.

[13] X. Gu, K. Nahrstedt, B. Yu. SpiderNet: An Integrated peer-to- peer Service Composition Framework, Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC' 04), pp. 110-119, 2004.

[14] G. Alonson, C. Hagen, D. Agrawal, A. E. Abbadi, C. Mohan. Enhancing the Fault tolerance of Workflow Management Systems, IEEE Concurrency, July-September 2000, pp 74-81, 2000.

[15] Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison-Wesley, 1995.