Publish/Subscribe Systems on Node and Link Error Prone Mobile Environments

Sangyoon Oh[1,2], Sangmi Lee Pallickara[2], Sunghoon Ko[1],
Jai-Hoon Kim[1,3], and Geoffrey Fox[1,2]

[1] Community Grids Lab., Indiana University, Bloomington, IN. U.S.A
{ohsangy, leesangm, suko, jaikim, gcf}@indiana.edu
[2] Department of Computer Science, Indiana University,
Bloomington, IN. U.S.A
[3] Graduate School of Information and Comminucations,
Ajou University, Suwon, S. Korea
jaikim@ajou.ac.kr

Abstract. Publish/subscribe model is appropriate in many push based data dissemination applications such as data dissemination services, information sharing, service discovery, etc. Recently, theses types systems are rapidly adopted in mobile and ubiquitous environment. However, mobile and ubiquitous environments are error prone due to wireless link disconnection, power exhaustion on mobile devices, etc. We analyze performance and effectiveness of publish/subscribe systems on the failure of client and server nodes and disconnection of communication links, which is common in mobile environments. We also perform experiments on our test bed to verify correctness and usefulness of our analysis.

1 Introduction

Publish/subscribe model [1] is appropriate in many applications such as data dissemination services [2], information sharing [3], service discovery [4], data dissemination services, information sharing, service discovery, etc. As these kinds of services are popular in mobile and ubiquitous environments, publish/subscriber model will be more widely used. Fig.1 depicts system configurations of publish/subscribe systems for error prone mobile environments. Publish/subscribe system consists of publisher (ES: Event Source), server (EBS: Event Brokering System), and subscriber (ED: Event Displayer). After publisher publishes data (events) asynchronously to a server, the server disseminates the data (events) to subscribers which registered its interest on the server. Main advantages of publish/subscribe systems include decoupling of publishers and subscribers in time, space, and synchronization [5-7]. Subscriber can access published data asynchronously anytime and anywhere at its own convenience. Also publisher, server, and subscriber can continue its operation even some parts of publish/subscribe system fails. These kinds of characteristics are main advantages of publish/subscriber model and make it useful in mobile ubiquitous environments [8].

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 576 – 584, 2005.

Many researches have been performed so far to propose architecture, add useful functions, and improve performance of publish/subscribe systems include Siena [9], Gryphon [10], JEDI [11], Rebeca [12], Elvin [13]. However, to our best knowledge, research of performance modeling for publish/subscriber system including node and communication link failures has not been announced yet. As mobile and ubiquitous service environments include many kinds of mobile devices and sensors which are connected via wireless networks, failure (or unavailability) rate of device and communication links are much higher than conventional wired environments. Thus, it is essential to consider how the failure and unavailability of mobile devices and wireless networks affect on performance and use of service. We analyze the influence of errors on mobile devices, servers, and wireless links, and compare to other interaction-based models such as client-server model and polling models. We can estimate performance in error prone environment and effectively adopt publish/subscribe systems by using our analysis. The results of our analysis show that publish/subscribe system is more durable than client/server models in error prone mobile and ubiquitous environments including mobile devices and wireless networks. We also perform some experiment on test bed to verify correctness of our analysis.

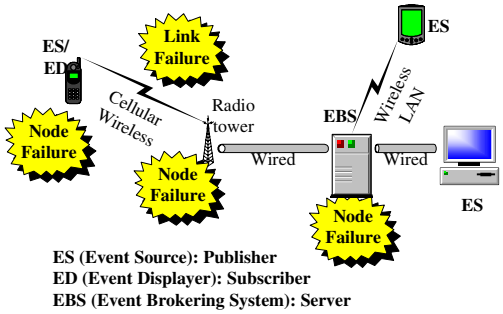

Fig. 1. Pub/Sub System Configurations

2 Cost Model

2.1 System Models

In this subsection, we propose cost analysis model for publish/subscribe systems.
We assume following basic system parameters to analyze cost.

- α : publish rate
- β : subscriber's access rate of published events, or request rate of client in the client/server models
- c_{ps} (α) : publish/subscribe cost per event, c_{pub} (cost for publish events) + c_{sub} (cost for subscribe events)
- c_{rr} (β) : cost per request and reply in client-server model
- $c_{poll}(\alpha,T)$: cost of periodic publish or polling (We assume function of α and T (ex. $c_{pub}\alpha T$, c_{poll}), where T is length of period.)
- s(n) : effect of sharing among n subscribers, e.g., server can deliver events with low cost when it broadcasts event to many subscribers.
- t_{ps} : time delay for publish/subscribe, t_{pub} (time delay for publish) + t_{sub} (time delay for subscribe)
- t_{rr} : time delay for request and reply in request-reply (client-server) model

- $t_{poll}(\alpha, T)$: time delay for periodic publish
- $t_{proc}(\alpha, T)$: amount of processing time on server
- λ, λ_s, λ_c : failure rate of communication link, server, and client, respectively
- μ, μ_s, μ_c : recovery rate of communication link, server, and client, respectively.

2.2 Cost Analysis

We analyze cost of four different models, (1) publish/subscribe, (2) event(broker) message based request/reply, (3) request/reply, and (4) periodic polling models including failure (disconnection) and recovery (reconnection) of communication links as well as failure and recovery of node of server and client. Four different models can be categorized as shown in Table 1. Most implementations of publish/subscribe model and some request/reply models (we call it as "event message based request/reply") include *persistent file* and *durable database*, for fault-tolerance of server and client, respectively. *Persistent files* for events publish and subscribe are sharable among servers. When a server fails, the another server can take over the role and publish/server model can continue its transactions. *Durable database* saves events log and provides clients events history after clients recovers from failures. Without the durable database, clients could not receive events occurred during failure.

Table 1. Characteristics on Failures

Models	Types	Link failures	Server failures	Client failures
publish/subscribe	publish/ subscribe	Wait until link reconnection, t ransaction is preserved	Server is replicated an d transaction can be c ontinued with sharabl e *persistent file*	After recovery, client can access events occurred dur ing failure using events lo g on *durable database*
Event message ba sed request/reply	request/ reply	Wait until link reconnection, t ransaction is preserved	Server is replicated an d transaction can be c ontinued with sharabl e *persistent file*	After recovery, client can access events occurred dur ing failure using events lo g on *durable database*
request/reply (RPC)	request/ reply	Wait until link reconnection, t ransaction needs to restart	Transaction needs to r estart after recovery	Lost event or data during c lient failure
periodic polling	any	Wait until link reconnection	Depends on system	Depends on system

Cost metric in this analysis is time delay to transfer message and additional time required due to failure of communication links and nodes of server and client.

Publish/Subscribe Model

Failure of communication link: Besides cost to transfer message without disconnection (c_{pub} + n s(n)c_{sub}), additional cost (delay time) is required due to failure of communication links (disconnection). If communication link is disconnected during publish or subscribe, data transfer is delayed until the link is reconnected. There are two cases in disconnection: (1) communication link was connected but disconnected during message transfer for publish or subscribe is performed (probability is $\mu/(\lambda+\mu)$ $(1-\varepsilon^{-(t_{pub}+t_{sub})\lambda})$), and (2) communication link was disconnected (probability is $\lambda/(\lambda+\mu)$). When disconnection occurs (probability is $\mu/(\lambda+\mu)(1-\varepsilon^{-(t_{pub}+t_{sub})\lambda})$+

$\lambda/(\lambda+\mu)$) and subscriber accesses events before reconnection (probability is $\beta/(\mu+\beta)$), average delay time due to link disconnection is $1/\mu$. Thus, cost of publish/subscribe model for each event publish and subscribe in the view point of a subscriber is:

$$t_{pub} + t_{sub} + \{\mu/(\lambda+\mu)\,(1-\varepsilon^{-(t_{pub}+t_{sub})\lambda}) + \lambda/(\lambda+\mu)\}\,\{\beta/(\mu+\beta)\}(1/\mu). \quad (1)$$

Failure of server: When a server (broker) fails, the another server (broker) can take over the role and publish/server model can continue its transactions without any interruption. We ignore the cost required for task transition from failed server to backup server. Thus, cost is:

$$t_{ps} = t_{pub} + t_{sub}. \quad (2)$$

Failure of client: After a client recovers from failure, client can obtain any event at anytime if durable database exists, which logs data (or events) to be used by client after recovery from failure. However, size of durable database is limited. If durable database can log maximum of n_{log} data (events), a client can be provided up to n_{log} recent events occurred before recovery. However, data is not available to the client after its recovery from a failure when the client requires more prior event than n_{log}-th recent event. The probability that i events occurred between failure and recovery is:

$$\left(\frac{\alpha}{\alpha+\mu_c}\right)^i \frac{\mu_c}{\alpha+\mu_c} \quad (3)$$

If $i \geq n_{log}+1$, i - n_{log} events are lost due to exceeding limitation of capacity for event log. Thus, an average number of lost events per client failure is:

$$\sum_{i=n_{log}+1}^{\infty}\left(\frac{\alpha}{\alpha+\mu_c}\right)^i \frac{\mu_c}{\alpha+\mu_c}(i-n_{log}) = \left(\frac{\alpha}{\alpha+\mu_c}\right)^{n_{log}} \frac{\alpha}{\mu_c} \quad (4)$$

Event (Broker) Message Based Request/Reply Model

Failure of communication link: Besides cost (delay time) without disconnection (t_{rr}), additional cost (delay time) is required due to disconnection of communication link. If communication link is disconnected during request and reply, a transaction is delayed until the link is reconnected. There are two cases in disconnection: (1) communication link was connected but disconnected during a transaction (sending request message, processing the request on a server, and receiving reply) is performed (probability is $\mu/(\lambda+\mu)(1-\varepsilon^{-t_{rr}\lambda})$), and (2) communication link was disconnected (probability is $\lambda/(\lambda+\mu)$). When disconnection occurs (probability is $\mu/(\lambda+\mu)(1-\varepsilon^{-t_{rr}\lambda})+\lambda/(\lambda+\mu)$), average delay time is $1/\mu$. Thus, cost of request/reply model is :

$$t_{rr} + \{\mu/(\lambda+\mu)(1-\varepsilon^{-t_{rr}\lambda})+\lambda/(\lambda+\mu)\}(1/\mu) \quad (5)$$

Failure of server: When we assume that persistent file and durable database exist in event (broker) message based request/reply model, request/reply model can continue its transaction. The roles of persistent file and durable database are similar to those of publish/subscribe model. Persistent files in publish/subscribe model are sharable

among servers. When a server fails, the another server can take over the role and receives requests from clients and replies to clients. Thus request/reply model can con tinue its transactions without any interruption. We also ignore the cost required for task transition from failed server to backup server. Thus, cost is:

$$t_{rr} \tag{6}$$

Failure of client: Durable database saves information. Thus, a client can receive reply any information at anytime after it recovers from a failure up to n_{log}-th recent data. Thus, an average number of lost data item per client failure is the same as Equ. (3).

RPC Based Request/Reply Model (Non-persistent and Non-durable)

Failure of communication link: In addition to waiting time for link recovery, transaction of request and reply procedure will be restarted after the link recovers if persistent file does not exist. Thus, following additional costs (time delay) are required:

- useless computation: If a server fails a transaction of during request/reply process, a client needs to request the server again after recover. Thus, the previous request/reply process until the server failure is useless (lost). We call it useless computation. The amount useless computation is the maximum of $t_{rr} + t_{proc}$.
- recovery: When a server recovers from a failure, some amount of time is required for recovery procedure. In this analysis, recovery time is ignored like the other models (publish/subscriber and event (broker) message based request/reply models).

We analyze cost (time) delay of request/reply model on a link failure in three cases: (1) Link does not fail (probability is $1-\mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})-\lambda/(\lambda+\mu)$): Required cost is t_{rr} , (2) Link was failed (probability is $\lambda/(\lambda+\mu)$): Cost is needed to wait until link recovers ($1/\mu$) and restart transaction (c_{rr}), and (3) Link fails during a transaction (probability is $\mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})$): In addition to useless computation ($E(t_{rr}+t_{proc},\lambda)$), costs to wait until link recovers ($1/\mu$) and restart transaction (c_{rr})) are necessary. Thus, cost is:

$$\begin{aligned} c_{rr} &= [1-\{\mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})+\lambda/(\lambda+\mu)\}]t_{rr} \\ &\quad + \mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})\{E(t_{rr}+t_{proc},\lambda)+1/\mu+c_{rr}\}+\lambda/(\lambda+\mu)\{(1/\mu)+c(\beta)\} \\ \Rightarrow c_{rr} &= [\,[1-\{\mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})+\lambda/(\lambda+\mu)\}]t_{rr} \\ &\quad + \mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})\{E(t_{rr}+t_{proc},\lambda)+1/\mu\}+\lambda/(\lambda+\mu)(1/\mu)] \\ &\quad /[1-\{\mu/(\lambda+\mu)(1-\mathcal{E}^{-(t_{rr}+t_{proc})\lambda})+\lambda/(\lambda+\mu)] \end{aligned} \tag{7}$$

$E(x, \lambda)$ denotes the amount of time, during an interval of length x, before a failure occurs with rate of λ, given that failure rate is λ and the failure occurs during the interval. Then,

$$E(x, \lambda) = \int_0^x t \frac{\lambda \varepsilon^{-\lambda t}}{1-\varepsilon^{-\lambda t}} dt = \lambda^{-1} - \frac{x \varepsilon^{-\lambda x}}{1-\varepsilon^{-\lambda x}} \tag{8}$$

Failure of server: When persistent file does not exist, transaction of request/reply model is delayed on a server failure until the server recovers from the failure. Moreover, transaction of request and reply procedure will be restarted after the server recovers. Cost analysis is similar to that of failure of communication link (Equ. (5)). We analyze cost (time) delay of request/reply model on a server failure in three cases: (1) Server does not fail (probability is $1-\mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})-\lambda_s/(\lambda_s+\mu_s)$): Required cost is t_{rr} , (2) Server was failed (probability is $\lambda_s/(\lambda_s+\mu_s)$): Cost is needed to wait until server recovers ($1/\mu_s$) and restart transaction (c_{rr})), and (3) Server fails during a transaction (probability is $\mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})$): In addition to useless computation ($E(t_{rr}+t_{proc},\lambda_s)$), costs to wait until server recovers ($1/\mu_s$) and restart transaction (c_{rr})) are necessary. Thus, cost is:

$$\begin{aligned} c_{rr} &= [1-\{\mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})+\lambda_s/(\lambda_s+\mu_s)\}]t_{rr} \\ &\quad +\mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})\{E(t_{rr}+t_{proc},\lambda)+1/\mu_s+c_{rr}\}+\lambda_s/(\lambda_s+\mu_s)\{(1/\mu_s)+ c_{rr}\} \\ \Rightarrow c_{rr} &= [\,[1-\{\mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})+\lambda_s/(\lambda_s+\mu_s)\}]t_{rr} \\ &\quad + \mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})\{E(t_{rr}+t_{proc},\lambda)+1/\mu_s\}+ \lambda_s/(\lambda_s+\mu_s)(1/\mu_s)] \\ &\quad /[1-\{ \mu_s/(\lambda_s+\mu_s)(1-\varepsilon^{-(t_{rr}+t_{proc})\lambda_s})+\lambda_s/(\lambda_s+\mu_s)] \end{aligned} \tag{9}$$

Failure of client: As we assume that a server of RPC based request/reply model contains a current data (or event) only without logging data during a failure of client, data (event) occurred during the failure is lost.

Periodic (Polling) Model

In the periodic model, the period of publish (or polling) is delayed until communication link is reconnected once communication link is disconnected. Cost of periodic model due to disconnection is function of delayed probability and time. When link was disconnected or is disconnected during message delivery (probability of $\mu/(\lambda+\mu)\,\varepsilon^{-t_{poll}(\alpha,T)\lambda} + \lambda/(\lambda+\mu)$), period is delayed by $1/\mu$ on average. Thus time delay is:

$$t_{poll}(\alpha,T) + \{\mu/(\lambda+\mu)\,\varepsilon^{-t_{poll}(\alpha,T)\lambda} + \lambda/(\lambda+\mu)\}(1/\mu) \tag{10}$$

If we consider conceptual cost:

$$\begin{aligned} &c_{poll}(\alpha,T) + [1-\{\mu/(\lambda+\mu)\,\varepsilon^{-t_{poll}(\alpha,T)\lambda} +\lambda/(\lambda+\mu)\}]c_{delay}(\alpha, T) \\ &+ \{\mu/(\lambda+\mu)\,\varepsilon^{-t_{poll}(\alpha,T)\lambda} +\lambda/(\lambda+\mu)\}c_{delay}(\alpha,T+1/\mu) \end{aligned} \tag{11}$$

We also assume that persistent file and durable database exist in persistent and durable periodic model. Costs are basically same as Equ's (5) and (6) in the case of server failure of communication links and server, respectively, and Equ (3) in client failure.

3 Performance Comparisons

In this section, we describe performance comparisons by parametric analysis. System parameters are setup as in the Table 2. Curves in Fig.2 show time delay when communication links are transiently disconnected and reconnected. As publisher and subscriber are decoupled in time and space, publish/subscribe system is more tolerable than request/reply system from failure. Publish/subscribe system requires less communication delay in compare to request/reply system. We also measure communication delay and additional service time delay per transaction due to sever failure. As shown in Fig.3, publish/subscribe system (curve "pub/sub") and event message based request/reply system (curve "req/reply") always require only communication delay since backup server takes over failed server without transition time. However, RPC based request/reply system (curve "RPC") requires additional service time delay and lost

Table 2. Parameters

Param.	Values
α, β	0.5
c_{ps}, c_{rr}	2
c_{pub} , c_{sub}	1
$c_{poll}(\alpha, T)$	1 or α T
$c_{delay}(\alpha,T)$	0, T, or α T
s(n)	1/n - 1
λ , λ_s, λ_c	0.0001 – 0.5
μ, μ_s	0.1
μ_c	0.05 - 0.1
t_{ps} , t_{rr}	1
t_{proc}	1 or 5
$t_{poll}(\alpha, T)$	1, T, or α T

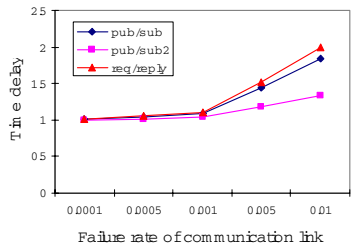

Fig. 2. Time delay per transaction links (α=0.5, s(n)=1, t_{ps}=1, t_{rr}=1, μ=0.1, and β=0.5 for pub/sub and req/reply and β=0.2 for pub/sub2)

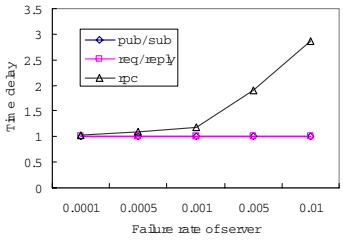

Fig. 3. Time delay per transaction (α=0.5, t_{ps}=1, t_{rr}=1, β=0.5, and μ_s=0.1)

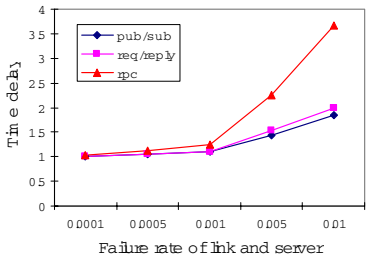

Fig. 4. Time delays per transaction (α=0.5, t_{ps}=1, t_{rr}=1, t_{poll}=1, β=0.5, μ=0.1, μ_s=0.1)

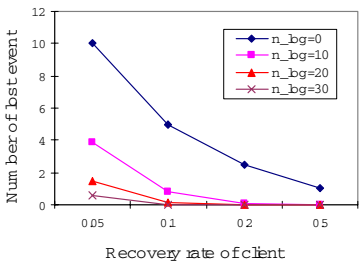

Fig. 5. Number of lost events per client failure by varying recovery rate of client

Table 3. Experimental results: Delay time of sending message

	Wireless	Wired	Total (msec.)
RPC	1290.7 (client – gateway)	39.9 (gateway – server)	1330.6 (t_{rr})
Pub/Sub	1448.4 (ED – EBS)	89.7 (EBS – ES)	1538.1 (t_{ps})

computation due to server failure. We also measure performance considering both failures of communication link and server. Fig.4 depicted results. As we expected, performance of "RPC" is the worst. Fig.5 shows effectiveness of durable database which logs events for the failure of client. Without logging, a client loses events occurred during its failure. As shown in Fig.5, number of lost events can be reduced as maximum number of log increases. We can consider "nlog=0" curve as publish/subscribe system without durable database.

Table 3 shows our experimental results (time delay) of sending short message (10 bytes) in a practical environment. The experiment environment consists of NaradaBrokering [14] system where a HHMS (HandHeld Message Service) Proxy [15] plugged in, mobile clients, and conventional PC applications. We have an experiment with a J2ME echo client application and a simpleserver that sends 10 byte message. The round trip time is measured 2000 times and the median value is taken. We use Treo 600 with Sprint PCSVision service as a client device. In our analysis, we set parameters $t_{ps} = t_{rr} = 1$, which is comparable to our experimental results ($t_{rr} = 1.33$ sec. and $t_{ps} = 1.53$ sec.). We need to adjust various parameters for each system because these parameters depend on systems and application environments.

4 Conclusion

Publish/subscribe model has many advantages in push based mobile applications. However, mobile and ubiquitous environments are error prone due to wireless link disconnection, power exhaustion on mobile devices, etc. We analyze performance and effectiveness of publish/subscribe systems on the failure of client and server nodes and disconnection of communication links. The results of our analysis show that publish/subscribe system is more durable than client/server models in error prone mobile and ubiquitous environments including mobile devices and wireless networks.

References

1. P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," ACM Computing Surveys, vol. 35, no. 2, Jun. 2003, pp. 114-131.
2. G. Muhl, A. Ulbrich, K. Herrmann, and T. Weis, **"**Disseminating Information to Mobile Clients Using Publish-Subscribe," Proc. of the IEEE Internet Computing, Vol.8, No. 3, 2004
3. Sagar Chaki, Pascal Fenkam, Harald Gall, Somesh Jha, Engin Kirda, and Helmut Veith, "Integrating Publish/Subscribe into a Mobile Teamwork Support Platform," Proc. of the 15th International Conference on Software Engineering and Knowledge Engineering 2003

4. Zhexuan Song, Yannis Labrou and Ryusuke Masuoka, "Dynamic Service Discovery and Management in Task Computing," Proc. of the 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services, August 22-25, 2004
5. L. Fiege, F. Gartner, O. Kasten, and A. Zeidler, "Supporting Mobility in Content-Based Publish/Subscribe Middleware," Middleware 2003, LNCS 2672, pp. 103 – 122, 2003.
6. M. Caporuscio, A. Carzaniga, and A. Wolf, "Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications," IEEE Transactions on Software Engineering, vol. 29, no. 12, pp. 1059 – 1071, Dec. 2003.
7. U. Farooq, E. Parsons, and S. Majumdar, "Performance of Publish/Subscrive Middleware in Mobile wireless Networks," Proc. of WOSP'04, pp. 278-289, Jan. 2004.
8. S. Lee, S. Ko, G. Fox, K. Kim, and S. Oh, "A Web Service Approach to Universal Accessibility in Collaboration Services", Proc. of ICWS, Las Vegas June 2003.
9. A. Carzaniga, D.Rosenblum, and A. Wolf, "Design and evaluation of a wide-area event notification service," ACM Transactions on Computer Systems, 2001
10. M. Aguilera, R. Strom, D. Sturman, M.Astley, and T. Chandra, "Matching events in a content- based subscription system," Proc. of ACM Symp. on Principles of Distributed Computing, 1999
11. G. Cugola, E. Di Nitto, and A. Fuggeta, "The JEDI Event-based infrastructure and its Application to the Development of the OPSS WFMS," IEEE Trans. of Software Engineering, 2001
12. L. Fiegen, G. Muhl, and F. Gartner, "A Modular Approach to Building Event-Based Systems," Proc. of ACM Symposium on Applied Computing, 2002
13. B. Segall, D. Arnold, J. Boot, M. Henderson and T. Phelps, "Content Based Routing with Elvin4," Proc. of AUUG2K, 2000
14. Shrideep Pallickara and Geoffrey Fox, "NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids," Proc. of ACM/IFIP/USENIX International Middleware Conference Middleware, pp 41-61, 2003.
15. Sangyoon Oh, Geoffrey C. Fox, Sunghoon Ko, "GMSME: An Architecture for Heterogeneous Collaboration with Mobile Devices", Proc. of MWCN 2003, Singapore, Oct., 2003.