

# The e-Demand project: A Summary

Paul Townend, Jie Xu, Erica Yang, Keith Bennett<sup>1</sup>, Stuart Charters<sup>2</sup>,  
Nick Holliman<sup>2</sup>, Nik Looker<sup>2</sup>, Malcolm Munro<sup>2</sup>

School of Computing,  
University of Leeds,  
LS2 9JT, UK

<sup>1</sup> School of Engineering,  
University of Durham,  
DH1 3LE, UK

<sup>2</sup> Department of Computer Science,  
University of Durham,  
DH1 3LE, UK

{ pt, jxu, ericay } @ comp.leeds.ac.uk

{ keith.bennett, s.m.charters, n.s.holliman,  
n.e.looker, malcolm.munro } @ durham.ac.uk

## Abstract

*The e-Demand project is a recently completed, three year joint collaborative e-Science project between the Universities of Durham and Leeds, together with experts from industry (Sun Microsystems, Sharp, and Sparkle Computer Technology). The goal of e-Demand has been to investigate fundamental dependability issues underlying large-scale coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations, as well as to provide support for stereoscopic visualisation amongst e-Science applications. Services have been developed to facilitate attack-tolerant information retrieval systems, the provision of multi-version-design based fault tolerant web services, fault injection testing of Grid software and web services, and stereoscopic visualisation applications. This paper summarises the achievements of the project, and describes the services that have been developed in more detail, together with details of planned future work.*

## 1. Introduction

The e-Demand project is a recently completed, three year joint collaborative e-Science project between the Universities of Durham and Leeds, together with experts from industry (e.g. Sun Microsystems, Sharp, and Sparkle Computer Technology). The project was coordinated under the guidance of the North East Regional e-Science Centre, and consists of four investigators, two research assistants, and two PhD students.

The overall goal of e-Demand has been to investigate fundamental dependability issues underlying large-scale coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations, as well as to provide support for stereoscopic visualisation amongst e-Science applications. To achieve the goal, four research themes were derived: Attack-Tolerant Information Services (ATIR), Fault Tolerance for Web Services (FT-Grid), Fault Injection for Web Services (WS-FIT), and Stereoscopic visualisation. ATIR aims to resolve a fundamental security problem associated with large-scale Grid applications: secure information retrieval. FT-Grid and WS-FIT offer solutions for developing dependable Grid services in an error-prone network environment. The visualisation strand supports complex visualisation pipelines including those where branching and re-

combination is required. This paper summarises what the project has achieved during its lifetime.

## 2. Attack-tolerant Information Services

### 2.1 Motivations

The fast-growing reliance of our daily life upon online information services often demands an appropriate level of privacy protection as well as highly available service provision. Most existing security solutions have attempted to address these problems separately. However, the unique characteristics of the Grid pose significant security challenges that demand new security solutions that can provide satisfactory answers to both requirements. In the Grid service infrastructure, the collaboration relationship among participating parties can be formed on the fly. As a result, only limited trust can be placed in any nodes within a dynamic Grid environment. Any trust relationship among Grid nodes should remain valid only within the lifetime of a submitted job. The transient and dynamic nature of Grid based interactions demand new solutions to security problems arise in the Grid environment.

Because we have much less control over remote Grid resources, the level of security assurance that is required by Grid users naturally becomes much higher than that by users of conventional distributed systems. With a particular emphasis on the users' security, this research focuses on two

key security challenges centred on the Grid trust issue: protecting the intention (privacy) of users against untrusted Grid nodes and detecting job tampering against malicious attacks.

## 2.2 What is ATIR?

Attack-Tolerant Information Retrieval (ATIR) is a distributed query technique that has been developed and implemented as part of the e-Demand project. Based on an extension of existing theoretical results for Private Information Retrieval (PIR) [CHO95], ATIR is developed with particular emphasis on its practicability in real applications. ATIR exploits replicated resources/services to protect a user's privacy and to ensure service availability. In particular, ATIR can tolerate any collusion of up to  $t$  servers for privacy violation and up to  $f$  faulty (either crashed or malicious) servers in a system with  $k$  replicated servers, provided that  $k \geq t + f + 1$  where  $t \geq 1$  and  $f \leq t$ . In contrast to other related approaches, ATIR relies on neither enforced trust assumptions, such as the use of tamper-resistant hardware and trusted third parties, nor an increased number of replicated servers. This technique is particularly appealing for certain security critical applications in the Grid and Web Services environments, where unknown information resources are discovered on the fly and are exploited with certain level of security requirements.

ATIR aims to achieve two goals: i) to protect the privacy of users against accidentally or purposefully information disclosure by information owners; and/or ii) to ensure the correctness of information retrieval against malicious attacks, such as the occurrence of corrupted results.

## 2.3 ATIR Schemes

The ATIR technique comprises of three closely linked techniques: privacy protection, error detection and attack tolerance. Here, we describe their principles. By hiding the intention of retrieval operations, the privacy of users can be protected. Hence, the risk of targeted attacks can be reduced. Through restricting the range of valid results, errors may be detected and corrupted servers may therefore be identified. Finally, attack tolerance is achieved through the introduction of a form of redundancy – replication, a classic and well-known fault tolerance technique for tolerating faults. As a whole, all three techniques complement to each other and together they provide a solution for the ATIR problem.

We have developed two ATIR schemes: deterministic ATIR (d-ATIR) and probabilistic ATIR (p-ATIR) were developed. Both ATIR schemes consist of three basic algorithms: a query

algorithm, an answer algorithm, and a reconstruction algorithm, and a result verification algorithm. The basic algorithms are based on the polynomial-interpolation PIR schemes presented in [CHO95]. There are two types of result verification algorithms: one for the pATIR scheme and the other for the dATIR scheme. Both verification algorithms are used to identify correct results and eliminate incorrect ones. The result verification algorithms are executed by the client program after executing the query and reconstruction algorithm whilst the answer is executed by the server program.

Both ATIR schemes satisfy the following five requirements: efficiency, privacy, availability, safety, and liveness. The efficiency requirement means that ATIR schemes should have a non-trivial communication complexity. The privacy requirement means that ATIR schemes should maintain the privacy of the user in an information theoretic sense. The availability requirement means that ATIR schemes should ensure the availability of a correct result. The safety requirement means that ATIR schemes should ensure the correctness of an output if there is an output. The liveness requirement means that ATIR schemes eventually terminate.

## 2.4 Implementations

ATIR has been implemented based on a client-server architecture where a user utilises an information service through submitting a query to the ATIR client program. Together with a set of randomly generated queries, the client produces a set of query requests which are sent to remote servers to execute respectively. Based upon the replies, the client identifies and recovers the desired result.

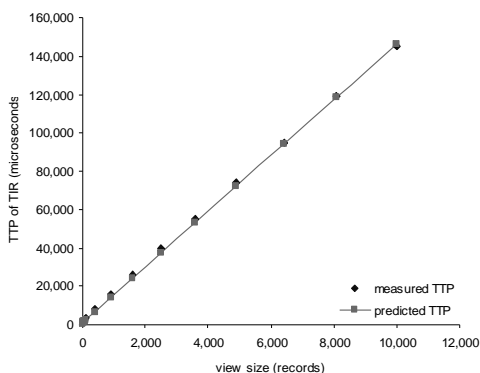
We have implemented both ATIR schemes in Java and C programming languages based on the implementation model presented in [YAN03]. This model has two parts: client side and server side. A user interacts with a client, which can be the user's own machine or a trusted computing base chosen by the user. There are two types of messages exchanged between the user and the client: INPUT and RESULT messages. The server side comprises of  $k$  replicas. Each of them connects to a backend database. There are two types of messages exchanged between the client and each individual replica: QUERY and ANSWER messages.

## 2.5 Performance Evaluation

In order to examine the impacts of varying key performance factors of the system, linear regression based analytical methods have been applied to the evaluation of the implemented

ATIR system. Various analytic models have been developed to predict the performance of key operations and the system as a whole. These models have been validated by extensive empirical results, and the results have shown that predicted values match closely with the actual measurements.

Figure 1 shows the measure and predicted total processing time (TPT) of the pATIR implementation against view sizes. The view size is the number of database records involved in the server side computation. The view size used in an ATIR service is an indication of the level of privacy protection achieved. As the view size grows, the protection level of the user's privacy increases. As the number of records involved in an ATIR computation grows, the server has less information about the actual record that the user is interested in. As indicated in the figure, the TPTs clearly increase linearly as the view sizes increase. The figure also shows that there is little difference between the prediction and the actual measurements of TPTs of pATIR. The coefficient of determination is 99.96%. A similar pattern of performance result is shown in the predicted and measured TPTs of the dATIR implementation.



**Figure 1: predicted and measured TPTs of pATIR schemes**

## 2.6 Applications

In the e-Science community, the practicability of ATIR has also been demonstrated through local and Internet deployments. The ATIR technique has been applied to a bio-informatics GRID application where privacy protection is one of the primary concerns of (resource) users. Specifically, ATIR has been integrated as a plug-in into Taverna to enable secure information retrieval in a realistic application setting. The end result has been a privacy-preserving query service for Taverna users (e.g. biologists) to access a wide

range of biological data sets available in the public domain. For further details of the integration, readers are referred to the companion paper [YAN05].

## 3. Fault Tolerance for Web Services

### 3.1 Motivations

The fault tolerance strand of the e-demand project is concerned with providing the means for increasing dependability in service-based systems that feature potentially unreliable, insecure or otherwise untrusted nodes. A traditional way to increase the dependability of distributed systems is through the use of *fault tolerant* techniques [AND90]. The approach of design diversity - and especially multi-version design (MVD) - lends itself to service-oriented architectures (SOAs - application architectures within which all functions are defined as independent services with well-defined invocable interfaces, which can be called in defined sequences to form business processes), as the potential availability of multiple functionally-equivalent services should allow a multi-version system to be dynamically created at much lower cost than would traditionally be the case. At the same time, service-orientation promises to reduce the cost of developing and maintaining any in-house services, as well as the cost of integrating multiple services together [LIN03].

A potential problem of this approach, however, is that in the traditional SOA model, the implementational details of a service are hidden from a client; the only information available to a client is the service's interface and - possibly - some Quality of Service (QoS) metadata. This may be an issue when developing an MVD system using functionally-equivalent services, as although these services may initially seem disparate (for example, they may be developed or hosted by different organisations), they may - during the course of their execution - invoke one or more identical, "shared" services. Should one of these shared services fail, then the failure may propagate back to the calling services, and result in a common-mode failure (CMF). A CMF occurs when independent or non-independent faults lead to similar errors between versions of an MVD system. Such failures are a "worst case" scenario in a fault-tolerant system, as such failures may be passed through the system undetected; it is often safer to return no result, and alert an operator and/or place a system in a safe state, than it is to allow an undetected error occur.

A solution is offered in the form of provenance, which is the documentation of the process that leads to a result. By using a provenance system to



encouraging results when subjected to vigorous fault injection-based testing. In addition, our research has also for the first time explored the use of provenance data to provide topological information which can be used during the voting process of an MVD scheme, and has resulted in an initial proof-of-concept for the approach. The empirical data we have generated is a valuable first step in evaluating the effectiveness of such an approach. Future work will include investigation into obtaining QoS indicators from the metadata of each service in an MVD channel's workflow – possibly facilitated through actor provenance – and applying these to the weighting algorithm. We also intend to investigate the relationship between shared components and common-mode failure in more detail, in order to more finely tune our voting scheme

## 4. Fault Injection for Web Services

### 4.1 Motivations

The proliferation of Web Service technology within the domains of e-commerce and e-science has made quality of service (QoS) issues a high priority. Our research into dependability assessment has produced a method and tools that have wider applicability for all areas of QoS with relation to SOAP-based SOAs.

### 4.2 WS-FIT

Previous research in the field of service testing via fault injection has concentrated on tightly coupled, RPC-based distributed systems. In defining a testing method for web services, new sets of challenges are faced which require different solutions. Key differences that are encountered when testing web services are: (1) greater chance for network failure, (2) higher levels of security and encryption, (3) more generic nature of the platform and the need to support multiple programming languages, (4) timing constraints and the asynchronous nature of web service operations. This is due to the loosely coupled nature of typical SOAP-based systems that implement services.

As part of the e-Demand project, we have developed an innovative fault injection tool called WS-FIT (Web Services Fault Injection Technology). WS-FIT allows network level fault injection to be used to test web service systems (including OGSA systems).

Normally network level fault injection is based upon the more or less random corruption of bytes within a network packet; our method extends this method to make meaningful perturbation to a SOAP message, e.g. our method can target a single parameter within an RPC message

sequence, and hence it can simulate API level fault injection. Also, since our method is script based it is possible to manipulate message flows in other ways; for instance, latencies can easily be added to message transfers and thus affect throughput.

An instrumented SOAP API is used that includes two small pieces of hook code. One hook intercepts outgoing messages, transmits them via a socket to the fault injector engine and receives a modified message from the fault injector. This modified message is then transmitted normally to the original destination via the original protocol stack being signed and encrypted as part of this process. There is a similar hook for incoming messages. By instrumenting the SOAP stacks on strategic machines, this method can be used as part of the certification testing for individual components within a production system, without the need for a test harness.

Whilst a number of existing fault injectors could be used to do this, notably DOCTOR [HAN95] and Orchestra [DAW96], these tools are designed for general purpose protocol testing. WS-FIT was designed around an engine to decode SOAP messages and presents an interface at the script API level, with the information included in a SOAP RPC easily accessible. WS-FIT uses a script to provide both a trigger for fault injection and also the fault injection itself. The WS-FIT GUI can be used to create skeleton scripts by parsing the WSDL for a web service and allowing the user to set triggers on messages and parameters. The user can then complete the test script by entering fragments of script to perform fault injection. The GUI can generate a complete test script from this information.

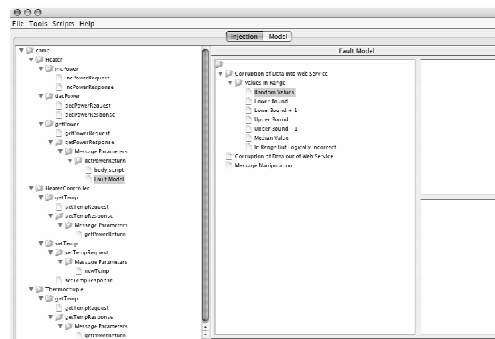


Figure 3: The WS-FIT GUI

To use WS-FIT, hook code must be installed on the server where faults are to be injected. By strategically positioning this hook code on servers not running the service under test, WS-FIT can be used as part of the certification process for

individual components, since the platform running that service will not be altered. WS-FIT uses a script to provide both a trigger for fault injection and also the fault injection itself. The WS-FIT GUI, shown in figure 3, can be used to create skeleton scripts by parsing the WSDL for a web service and allowing the user to set triggers on messages and parameters. The user can then complete the test script by entering fragments of script to perform fault injection. The GUI can generate a test script from this information.

### 4.3 Evaluation

WS-FIT has been used to evaluate both example web service systems [LOO04] and OGSA Grid middleware systems [LOO03]. Our research using WS-FIT has shown that it can effectively be used to detect defects in system designs by exercising seldom used code pathways via fault injection techniques.

### 4.4 Future Work

Our future work will concentrate on three main areas. Firstly, we will conduct experiments on more complex systems. This will allow us to not only evaluate and enhance our method and tools further, but will also provide us with more metrics on constructing test scripts using network-level fault injection techniques. Secondly, we will examine and enhance our real-time RPC visualization method included in WS-FIT. Our preliminary experiments with visualization have provided promising results that indicate its potential usefulness. Finally, we will investigate a method that will allow us to construct test scripts for our tool. Currently, our method and tool provide assistance for users writing test scripts. This includes generation of skeleton scripts from WSDL definitions and the generation of triggers from graphical input, but the tools still require the user to enter code fragments to complete the test scripts. Ideally, our new method would be automated and would generate complex test scripts to give some defined level of coverage.

## 5. Visualisation

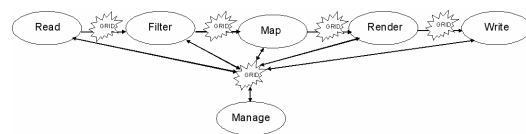
### 5.1 Background

The visualisation strand of the e-Demand project examined the use of services to allow visualisation on the grid. Current visualisation software is designed to run on a single machine or a local cluster of machines. This method of operation limits the problem size that can be dealt with using the software and requires a machine that is at least as powerful as required for the most complex operation to be performed. Requiring a machine which may not be fully utilised at all times is undesirable and expensive. Where other

research projects have looked to grid enable existing visualisation software, the opportunity in this project has been taken to look at the visualisation process and visualisation software anew and to start with a 'blank canvas' when approaching the issue of visualisation on the grid.

### 5.2 Architecture:

The traditional visualisation architecture is a one way pipeline, where data flows through various stages of processing, filtering, mapping and rendering to produce a final visualisation. The grid visualisation architecture [CHA04] maintains the traditional visualisation pipeline but augments it by making each stage in the pipeline into a service and allowing them to be distributed on different resources and also by adding a manage service which acts as a user proxy and is designed to manage context across the pipeline and to facilitate collaboration. The architecture is shown in Figure 4.



**Figure 4: Distributed Grid Based Service Oriented Visualisation Architecture**

The advantages of the service oriented architecture over the traditional pipeline and other visualisation software are:

- Independently scalable stages of the pipeline.
- Ability to handle larger problems.
- Use of specialized resources for specialist tasks.
- Ability to use resources with limited computational power to control and view visualisations.
- No need for client side software.

### 5.3 Implementation:

The visualisation grid architecture has been implemented using the Java programming language and Web Service technology including Apache Tomcat as a container environment. Each service has an interface defined in WSDL, three such interfaces exist, one for read services, one for write services and one to cover the map, filter and render services.

The Java programming language was chosen for the implementation as it is platform independent and Web Service technology was employed to

allow platform agnostic description of services and communication between services.

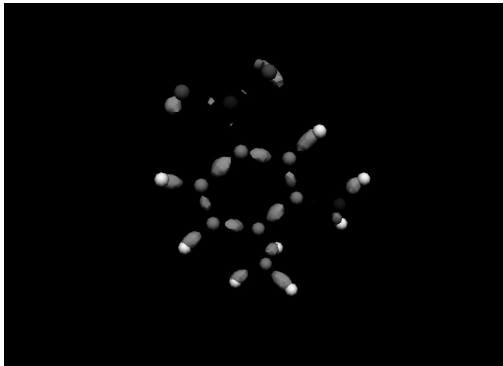
The Java Commodity on Grid (COG) Toolkit [LAS01] from Globus was leveraged to provide a streaming Data Transport mechanism between services.

The current implementation of the architecture does not include the manage service or allow for stateful services, as such the pipeline must be executed each time a revision to the output is require as if it were a new pipeline.

#### 5.4 Results

The architecture was tested with two scenarios, the first scenario whose results are shown in Figure 5 is that of X-Ray Crystallography where X-Rays are used to examine crystals at extremely low temperatures.

The pipeline for this scenario was a multiple branch pipeline, with multiple filter and map services. One branch of the pipeline produced the model showing the location of atoms in the crystal. The other branch produced the isosurface model of the electric field strength in the crystal. The render service merged the output of these two branches and provided the output for the write service to display on screen.

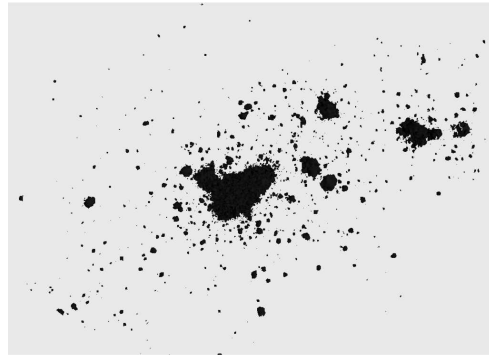


**Figure 5: X-Ray Crystallography Scenario**

The second scenario is a simulation of the spread of dark matter through the evolution of the universe from the big bang onwards. The results of this scenario are shown in Figure 6. This scenario contained a straight forward pipeline with only one branch. However this scenario has a much larger data set to produce the visualisation from.

Several experiments were run on the scenarios showing the impact of distributing services and the data for the pipeline, the results showed that transfer of data across a network increased computation time and for a multiple branch

pipeline the visualisation time to completion was the time that the slowest branch took. The experiments also showed that heavily loaded resources had a much more variable effect on the runtime of the visualisation. It was also noted that the larger the amount of data transfer the more effect that the network transfer had on the total run time.



**Figure 6: Dark Matter Scenario**

It was also observed however that the network distance, that is the remoteness of the data, did not have a significant impact on the transfer time when compared to a local data set being transferred over the network

#### 5.5 Conclusion

The current implementation highlights several useful areas for future work and highlights that whilst simply distributing the visualisation pipeline using non-scalable services increases the implementation time, it does not increase that time prohibitively and suggests that distribution using highly scalable services has the potential to decrease the total runtime of the visualisation.

These scalable services would also allow for much bigger problems to be examined and visualised. For scalable services to be most effective however it has been discovered that the format of data used in visualisation needs to be manipulated so that data can be easily split into small quantities for analysis.

#### 5.6 Future Work:

The future work is to develop services that have state and combine these with a manage service to co-ordinate the operation of the visualisation pipeline. The manage service would also allow support for collaborative visualisation to be incorporated into the pipeline.

The development of scalable services and tools to analyse the branches of a visualisation pipeline

to determine where resources should be deployed would provide the potential to visualise bigger problems and also to decrease the total runtime of the visualisation.

## 6. Other collaborations

In addition to academics at the universities of Durham and Leeds, the e-Demand project has benefited from contributions from industry, including Sun Microsystems, Sparkle Technology, and Sharp. In addition to this, the project has benefited from a collaboration with the IBHIS project (<http://www.co.umist.ac.uk/ibhis/>). The IBHIS project, funded as part of the EPSRC Distributed Information Management (DIM) programme, seeks to explore how to integrate information when it is obtained from a set of autonomous and independent organisations, stored in a multiplicity of formats (which may themselves change), and subject to both national and local access rules. By collaborating with e-Demand, both projects have benefited greatly.

## 7. Conclusion

This paper gives an overview of the contributions that have been made by the e-Demand e-Science project, which concluded in April 2005. We surmise our work with Attack-tolerant Information retrieval systems, fault tolerance, fault injection and stereoscopic visualisation, and list our achievements with each. For an extensive list of e-Demand related publications, please visit the e-Demand website at <http://www.comp.leeds.ac.uk/edemand/>.

## 8. Acknowledgements

The e-Demand project was funded by the EPSRC/DTI e-Science Core Programme grant number THBB/C008/00112C.

## 9. References

- [AND90] T. Anderson and P. Lee, *Fault Tolerance: Principles and Practice*. New York: Springer-Verlag, 1990
- [CHA04] S.M. Charters, N.S. Holliman, M. Munro, "Distributing Stereoscopic Scientific Visualisation Across the Grid", *Proceedings of the 7<sup>th</sup> IASTED Conference on Computer Graphics and Imaging*, Hawaii, August 17-19 2004
- [CHO95] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval", in *Proc. 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, Milwaukee, Wisconsin, USA, 23-25 Oct. 1995, pp. 41-51. *Journal version*: *J. of the ACM*, vol. 45, no. 6, 1998, pp. 965-981.
- [DAW96] S. Dawson, F. Jahanian, T. Mitton, "ORCHESTRA: A Probing and Fault Injection Environment for Testing Protocol Implementations", presented at *International Computer Performance and Dependability Symposium*, Urbana-Champaign, USA, 1996
- [HAN95] S. Han, K. G. Shin, H. A. Rosenberg, "DOCTOR: An IntegrateD Software Fault InjeCTiOn EnviRonment for Distributed Real-Time Systems", presented at *International Computer Performance and Dependability Symposium*, Erlangen, Germany, 1995
- [KIM95] K. H. Kim, "The Distributed Recovery Block Scheme," in *Software Fault Toleranmce*, M. R. Lyu, Ed. Chichester: John Wiley & Sons, pp. 189-210, 1995
- [LAS01] G. von Laszewski, I. Foster, J. Gawor, P. Lane, "A Java Commodity Grid Kit", *Concurrency and Computation: Practice and Experience*, pages 643-662, Volume 13, Issue 8-9, 2001.
- [LIN03] Z. Lin, H. Zhao, and S. Ramanathan, "Pricing Web Services for Optimizing Resource Allocation - An Implementation Scheme," presented at *2nd Workshop on e-Business*, Seattle, December 2003
- [LOO03] N. Looker and J. Xu, "Assessing the Dependability of SOAP RPC based Web Services by Fault Injection", in *IEEE International Workshop on Object-Oriented, Real-Time and Dependable Systems*, Capri Island, Oct. 2003
- [LOO04] N. Looker, M. Munro, and J. Xu, "Assessing Web Service Quality of Service with Fault Injection," presented at *Workshop on Quality of Service for Application Servers, SRDS, Brazil*, 2004
- [PRO05] *Provenance Recording for Services (PreServ)*, <http://www.pasoa.org>
- [TOW05a] P. Townend, P. Groth, N. Looker, J. Xu, "FT-Grid: A Fault-Tolerance System for e-Science", 4th UK e-Science All-Hands Meeting, Nottingham, September 2005
- [TOW05b] P. Townend, P. Groth, J. Xu, "A Provenance-Aware Weighted Fault Tolerance Scheme for Service-Based Applications", in *Proceedings of 8th IEEE International Symposium on Object-oriented Real-time distributed Computing*, Seattle, May 2005.
- [YAN03] E. Y. Yang, J. Xu, and K. H. Bennett, "Sharing with Limited Trust: An Attack-Tolerant Service in Durham e-Demand Project", in *Proc. UK eScience 2nd All-Hands Meeting*, Simon J. Cox Eds., Sept. 2nd-4th, 2003, Nottingham, U.K., ISBN 1-904425-11-9.
- [YAN05] Erica Y. Yang and Jie Xu, "Integrating an Attack Tolerant Information Service with Taverna", in *Proc. UK eScience All-Hands Meeting 2005*.