Arjuna

Computing Laboratory
University of Newcastle upon Tyne
Newcastle upon Tyne, Tyne and Wear
NE1 7RU
England

Personnel

Principal investigator	Prof. Santosh K. Shrivastava	Santosh.Shrivastava@newcastle.ac.uk
Project members	Dr. Graham D. Parrington	Graham.Parrington@newcastle.ac.uk
	Dr. Stuart M. Wheater	Stuart.Wheater@newcastle.ac.uk
	Mr. Mark C. Little	M.C.Little@newcastle.ac.uk
	Mr. Daniel L. McCue	Dan.McCue@newcastle.ac.uk
	Mr. Steve Caughey	S.J.Caughey@newcastle.ac.uk
	Mr. Joao Geada	J.M.C.Geada@newcastle.ac.uk
	Mr. Luiz Eduardo Buzato	L.E.Buzato@newcastle.ac.uk
	Mr. Raimundo J. D. A. Macedo	R.J.D.A.Macedo@newcastle.ac.uk

Contacts

Graham D. Parrington or Stuart M. Wheater
Computing Laboratory
University of Newcastle upon Tyne
Newcastle upon Tyne, Tyne and Wear
NE1 7RU
England

Project Description

Arjuna is an object-oriented programming system that provides a set of tools for the construction of fault-tolerant distributed applications. A prototype version written in C++ has been designed and implemented to run on a collection of Unix workstations connected by a local area network. Arjuna provides nested atomic actions (nested atomic transactions) for structuring application programs. Atomic actions operate on objects, which are instances of abstract data types (C++ classes), by making use of remote procedure calls (RPCs).

The design and implementation goal of Arjuna was to provide a state of the art programming system for constructing fault-tolerant distributed applications. In meeting this goal, three system properties were considered highly important:

(i) Integration of mechanisms: a fault-tolerant distributed system requires a variety of system functions for naming, locating and invoking operations upon local and remote objects and also for concurrency control, error detection and recovery from failures etc. These mechanisms must be provided in an integrated manner such that their use is easy and natural.

(ii) Flexibility: these mechanisms should also be flexible, permitting application specific enhancements, such as type-specific concurrency and recovery control, to be easily produced from existing default ones.

(iii) Portability: the system should be easy to install and run on a variety of hardware configurations.

The computational model of atomic actions controlling operations upon persistent objects provides a natural framework for incorporating integrated mechanisms for fault-tolerance and distribution. In Arjuna, these mechanisms have been provided through a number of C++ classes; these classes have been organised into a class/type hierarchy in a manner which will be familiar to the developers of 'traditional' (single node) centralised object-oriented systems. Arjuna is novel with respect to other fault-tolerant distributed systems in taking the approach that every major entity in the system is an object. Thus, Arjuna not only supports an object-oriented model of computation, but its internal structure is also object-oriented. This approach permits the use of the type inheritance mechanism of object-oriented systems for incorporating the properties of fault-tolerance and distribution in a very flexible way, permitting the implementation of concurrency control and recovery for objects in a type specific manner. Arjuna has been implemented without any changes to the underlying operating system (Sun Unix), making it quite portable.

Current Status and Future Plans

At the time of writing (June 91), a version of Arjuna is fully functional, and consists of a C++ stub generator (through which distribution transparency is achieved) and a number of C++ classes mentioned earlier. This version supports nested atomic actions and is available to interested users for research purposes.

Several enhancements to this version are in the pipeline. Supports for replication is currently being implemented, based on the design described in [15]. It will then be extended to include support for long running transactions using the concepts of coloured actions [11, 13]. Work has also begun to port Arjuna on to different platforms. This includes porting onto distributed memory multiprocessor systems (transputers running the Helios operation system) and the distributed computing platform (ANSAWARE) produed by the ESPRIT-ISA project.

Funding

The project is funded in part by Science and Engineering Research Council (SERC) grant No. GR/F 06494: The Design of High Performance Distributed Fault-Tolerant Systems, SERC grant No. GR/F 38402: Fault-Tolerant Multiprocessor Systems and ESPRIT Project 2267: Integrated Systems Architecture (subcontract from APM Ltd., Cambridge).

Report Availability

The Arjuna Info Service is a simple information retrieval system that is accessed via electronic mail. The Info Service has information about a number of topics related to the Arjuna System. The information on a particular topic can be obtained by sending message whose "Subject:" line is the topic you are interested in. For a complete list of all the topics, ask for "index".

Technical Questions	Arjuna@newcastle.ac.uk
Information Service	Arjuna-Info@newcastle.ac.uk
Information Service Administration	Arjuna-Info-Admin@newcastle.ac.uk

Arrangments are currently being made to make the Arjuna software widely available via FTP.

References

[1] S. K. Shrivastava, G. N. Dixon, G. D. Parrington. *Objects and Actions in Reliable Distributed Systems.* IEE Software Engineering Journal, Vol. 2, No. 5, pp. 160-168, September 1987.

[2] F. Panzieri, S. K. Shrivastava. *Rajdoot: A Remote Procedure Call Mechanism Supporting Orphan Detection and Killing.* IEEE Transactions on Software Engineering, Vol. SE-14, No. 1, pp. 30-37, January 1988.

[3] G. N. Dixon. *Object Management for Persistence and Recoverability.* University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, No. 276, December 1988. (Ph.D. Thesis)

[4] G. D. Parrington. *Management of Concurrency in a Reliable Object-Oriented System.* University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, No. 277, December 1988. (Ph.D. Thesis).

[5] G. N. Dixon, S. K. Shrivastava. *Exploiting Type Inheritance Facilities to Implement Recoverability in Object Based Systems.* Proceedings of Sixth Symposium on Reliability in Distributed Software and Database Systems, Williamsburg, pp. 107-114, March 1987.

[6] G. N. Dixon, S. K. Shrivastava, G. D. Parrington. *Managing Persistent Objects in Arjuna: A System for Reliable Distributed Computing.* Proceedings of Workshop on Persistent Object Systems, Persistent Programming Research Report, No. 44, Department of Computational Science, University of St. Andrews, August 1987.

[7] S. K. Shrivastava, L. Mancini, B. Randell. *On the Duality of Fault-Tolerant System Structures.* Proceedings of Workshop on Experiences with Distributed Systems, Kaiserslautern, West Germany, September 1987. (Lecture Notes in Computer Science, Vol. 309, Experiences with Distributed Systems, Springer-Verlag, pp. 19-37, September 1987.)

[8] G. D. Parrington, S. K. Shrivastava. *Implementing Concurrency Control in Reliable Distributed Object-Oriented Systems.* Proceedings of the Second European Conference on Object-Oriented Programming, ECOOP88, Oslo, Norway, August 1988. (Lecture Notes in Computer Science, Vol. 322, Springer-Verlag, pp. 233-249, 1988.).

[9] S. K. Shrivastava, G. N. Dixon, M. C. Little, G. D. Parrington, F. Hedayati, S. M. Wheater. *The Design and Implementation of Arjuna.* University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, No. 280, March 1989.

[10] G. N. Dixon, G. D. Parrington, S. K. Shrivastava, S. M. Wheater. *The Treatment of Persistent Objects in Arjuna.* Proceedings of the Third European Conference on Object-Oriented Programming, ECOOP89, University of Nottingham, pp. 169-189, July 1989. (Also in: The Computer Journal, Vol. 32, No. 4, pp. 323-332, August 1989.)

[11] S. K. Shrivastava, S. M. Wheater. *Objects and Multi-Coloured Actions.* Proceedings of Third Workshop on Large Grained Parallelism, SEI, Pittsburgh, October 1989.

[12] G. D. Parrington. *Reliable Distributed Programming in C++: The Arjuna Approach.* Second Usenix C++ Conference, San Fransisco, pp. 37-50, April 1990.

[13] S. K. Shrivastava, S. M. Wheater. *Implementing Fault-Tolerant Distributed Applications Using Objects and Multi-Coloured Actions.* Proceedings of Tenth International Conference on Distributed Computing Systems, Paris, France, pp. 203-210, May 1990.

[14] D. L. McCue, S. K. Shrivastava. *Structuring Fault-Tolerant Object Systems for Portability.* Fourth ACM SIGOPT Workshop, September 1990, Bologna.

[15] M. C. Little, S. K. Shrivastava. *Replicated K-Resilient Objects in Arjuna.* Proceedings of IEEE Workshop on the Management of Replicated Data, Houston, Texas, pp. 53-58, November 1990.

[16] S. M. Wheater. *Constructing Reliable Distributed Applications Using Actions and Objects.* University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, No. 316, June 1990. (Ph.D. Thesis)

[17] S. K. Shrivastava, G. N. Dixon, G. D. Parrington. *An Overview of the Arjuna Distributed Programming System.* IEEE Software, pp. 66-73, January 1991.