# In Memoriam Roger Needham

When, last summer, a meeting of editorial board members of *The Computer Journal* was proposed for January 2003, one of those planning to attend was Roger Needham. His participation was not to be. Following a struggle with cancer, Roger Needham passed away on 28 February 2003.

Roger was one of *les grands* of computer science. His work ranged over an incredibly wide swathe of computer science. His early work at Cambridge in the 1950s included cluster analysis and information retrieval. In the 1960s, he carried out pioneering work on computer architecture and system software. In the 1970s, his work involved distributed computing. In later decades, he devoted considerable attention to security.

His many honours included Fellow of the Royal Society, Fellow of the Royal Academy of Engineering, and CBE. In 1997, Roger set up and became first director of Microsoft Research in Cambridge. He was in this position when he died. He is survived by his wife, Karen Spärck Jones.

Roger's career and his life's work are an inspiration for all in computer science and related fields. His vision, well represented in his work and publications, was panoramic. His management and leadership were exemplary. *Abiit ad plures* now. His achievements will be remembered.

FIONN MURTAGH
*Editor-in-Chief*

# Capsule Reviews

**The Capsule Reviews are intended to provide a short succinct review of each paper in the issue, in order to bring the content to a wider readership. This issue's Capsule Reviews were compiled by Fairouz Kamarreddine. Professor Kamarreddine is an Associate Editor of *The Computer Journal* and is based in the Department of Computing and Electrical Engineering at Heriot-Watt University, Edinburgh, UK.**

**Representation of Web data in a Web warehouse.**
S. S. BHOWMICK, S. MADRIA AND W. K. NG

The Internet and the World Wide Web will always stand out as two of the greatest revolutions of the twentieth century. They have entered our lives from numerous directions and they will continue to influence all sorts of aspects of our daily routine, no matter what profession we have. The availability of the Internet, although with many benefits, requires sophisticated techniques in order to effectively manage Web data. This paper realises this need and proposes a data warehouse of Web data, where relevant hyperlinked Web documents are represented and stored effectively for further querying and manipulation. Previous approaches by information retrieval technology when applied to the Internet have been quite inefficient. The Web warehouse approach used in this paper is a collection of Web tables each of which consists of a set of Web tuples. Every Web tuple is a set of directed graphs where each such graph consists of a set of nodes and links, and satisfies a Web schema that contains useful meta information. A set of Web algebraic operators facilitate the manipulation of data. This paper treats the node and link objects as first-class citizens, and exploits the tree structure of XML to represent and manipulate data. This paper is interesting and provides a useful application of database technology.

**A type-passing approach for the implementation of parametric methods in Java.** M. VIROLI

Java was designed as a simple language and in this aspect it does not support generic programming in a straightforward manner. Although the inclusive polymorphism of the Java inheritance supports a reasonable degree of genericity, this approach has its limitations and this has led to several approaches to extend Java with generics. Many such approaches exploit parametric polymorphism to support generic programming. Parametric polymorphism allows the programmer to abstract a piece of code from one or more types, parameterizing the code on the so-called type parameters. This allows code reuse in different contexts by simply providing different instantiations for the type parameters. There are two implementation approaches to parametric polymorphism: the extensions approach and the translation approach. This paper concentrates on the translation approach, which claims to be easier to understand. The translation approach covers the existing basic implementation strategies for parametric polymorphism: type-erasure, code-expansion and type-passing. Until now, only type-erasure and code-expansion have been exploited for the implementation of parametric methods, but they suffer from limitations in the sense that type-erasure is unable to support parametric types at run-time, whereas code-expansion leads to the increase of disk and memory overheads. This paper studies a type-passing approach for the implementation of parametric methods in Java, which avoids the limitation of type-erasure and code-expansion.

**On the minimality of stream X-machines.** F. IPATE
The use of formal specification and verification methods and models leads to high-quality software and facilitates formal and automated analysis. Finite-state machines have been a successful approach in specifying software. In particular, a form of finite-state machine, called a stream X-machine, has already been described and has notable advantages, especially since it combines the dynamic features of finite-state machines with data structure. The advantages of using a stream X-machine as a specification method have been demonstrated in several published articles. Furthermore, tools, extensions and refinements of stream X-machines have been developed, all pointing to the usefulness of this concept. Just as the case for finite-state machines, the question of minimality arises for stream X-machines. That is, just as the research literature has investigated the minimal finite-state machines, the strongly minimal sequential machines, etc., for stream X-machines it would be interesting to investigate the existence of a minimal specification for a required functionality. This paper deals with this question and investigates the minimality issue for deterministic stream X-machines. The paper addresses two types of minimality with respect to a set of processing functions the machine uses: a state-minimal stream X-machine and a minimal cover. Both constructions are useful and have important implications for practical analysis and testing of software systems.

**Generating candidates when testing a deterministic implementation against a non-deterministic finite-state machine.** R. M. HIERONS
In the computational paradigm, non-determinism aids abstraction, whereas implementations are typically deterministic. A number of computational systems and protocols are described using non-deterministic finite-state machines, while their implementations are deterministic. With such a state of affairs, it is highly desirable to check an implementation against a non-deterministic finite-state machine. This paper therefore considers the problem of testing a deterministic implementation system $I$, which behaves like some unknown deterministic finite-state machine with at most $m$ states, against a non-deterministic finite-state machine $M$. This paper proposes a new approach to adaptive testing with two phases. In the first phase, an adaptive test procedure is applied in order to generate from $M$, a candidate $M^C$ that is a deterministic finite-state machine such that (under the test hypotheses) $I$ conforms to $M$ if, and only if, $I$ is equivalent to $M^C$. In other words, the candidate $M^C$ has the property that under the test hypotheses used, the implementation is correct if, and only if, it is equivalent to the candidate. Once this candidate $M^C$ is generated, the second phase carries out tests that could be generated from this candidate. These tests could be significantly smaller than those generated from $M$. It is easier to test from a deterministic finite-state machine than from a non-deterministic one. This paper focuses on the first phase and defines provably correct algorithms to generate a candidate deterministic finite-state machine where the implementation is deterministic, whereas the specification is a non-deterministic finite-state machine.

**LR-tree: a logarithmic decomposable spatial index method.** P. BOZANIS, A. NANOPOULOS AND Y. MANOLOPOULOS
Currently, there is a big demand for efficient manipulation of massive sets of geometric objects and hence, spatial database techniques have been an important area of development in the past few years. Spatial databases must be able to answer a wide range of geometric queries. This wide range of queries, combined with the massive volume of datasets involved, have led to the use of general-purpose data structures in real-world applications. This explains the success of the so-called R-trees as one of the most practical and well-behaved data structures for accommodating dynamic massive sets of geometric objects and conducting a very diverse set of queries on such datasets in real-world applications. Due to their flexibility, R-trees have been extended in different ways. Many variants of R-trees have been proposed, each aiming at improving the performance by tuning some parameters. The most prominent extension of R-trees is the so-called R*-tree, which is said to achieve the best performance. R*-trees, however, still face problems related to performance and the worst-case query time complexity cannot be avoided. The problem with performance in R*-trees comes from the repeated forced re-insertions and such repeated re-insertions do not guarantee a good tree in terms of query performance. The paper proposes the LR-tree instead of the R*-tree. The LR-tree consists of a number of component sub-structures, called blocks. Each block is organized as a weak R-tree which is a deletion-only R-tree. Insertions are dealt with by the blocks. The algorithms for block construction provide good performance during query search. This paper examines performance by both describing theoretical bounds and giving detailed experimental results.