Published by Oxford University Press on behalf of The British Computer Society. All rights reserved.
For Permissions, please email: journals.permissions@oupjournals.org
doi:10.1093/comjnl/bxh123

Capsule Reviews

FAIROUZ KAMAREDDINE

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 Kamareddine. Professor Kamareddine is an Associate Editor of *The Computer Journal* and is based in the School of Mathematical and Computer Sciences at Heriot-Watt University, Edinburgh, UK.

A Dynamic Component and Aspect-Oriented Platform. MONICA PINTO, LIDIA FUENTES AND JOSÉ MARÍA TROYA
Both component-based software development (CBSD) and aspect-oriented software development (AOSD) aim to improve the modularity and evolution of distributed systems by plugging in independent and reusable entities. By representing a significant advance towards assembling systems through plugging together independent and reusable components, CBSD aims to reduce time, cost and effort, and to improve flexibility, reliability and maintainability. By introducing the aspect dimension to separate 'cross-cutting' concerns (like synchronization and coordination), AOSD solves the code tangling problem. The paper argues that CBSD and AOSD are complementary technologies and proposes a component-aspect model (CAM) which combines their benefits. In CAM, both components and aspects are first-order entities. The underlying infrastructure supporting the CAM model is a component-aspect platform referred to as DAOP (dynamic aspect oriented platform), where the plugging of software aspects into components is performed at runtime. The main characteristics of both the CAM model and the DAOP platform are illustrated through a running example, and a CAM/DAOP prototype is implemented. The main advantages of this approach include an increase in the reusability of components, a reduction in the time needed to build complete applications and a better gap bridge between design and implementation.

Branch-Coverage Testability Transformation for Unstructured Programs. R. M. HIERONS, M. HARMAN AND C. J. FOX
Testing is both an important and an expensive part of software verification. For this reason, the used test criterion specifies when a set of test input is sufficient, and the challenge is usually to find test data which satisfy this criterion. Since the generation by hand of such test data is expensive and error-prone, automated test data generation has replaced manual test data generation. However, unstructured control flow and unstructured programs (i.e. programs which contain explicit jump statements like 'goto', 'exit' or 'break') present problems for the existing automated test data generation techniques, in particular, with respect to the presence of loops in these unstructured programs which create the additional problem of determining where the implicit loop boundaries are. Furthermore, the presence of many 'gotos' makes it hard to determine how close a test input comes to hitting a desired branch because the flow of control is not obvious. Moreover, in a test data generation system, the generator will attempt to force the loop to terminate either normally or through other means like using the 'exit' statement. But, since the problem of determining which paths are feasible is not decidable, the test data generator will not know which terminating way is likely to lead to the desired outcome. This paper focuses on the 'exit' statements as the most common source of unstructuredness and considers the problem of transforming a program p to a structured program p' such that any set of test data that provides full coverage for p' also provides full coverage for p. The paper shows that there are programs with 'exit' statements that are not path equivalent to any structured programs and hence removing 'exist' statements may lead to the loss of path equivalence. For this reason, the paper uses the so-called testability transformation which although does not preserve the traditional meaning of the program, preserves the test adequacy of sets of input data. The paper defines the so-called branch-coverage equivalence which gives rise to new transformations and new proof of correctness obligations. The transformation algorithm which takes a (unstructured) program p containing one or more 'exit' statements to a (structured) branch-covering equivalent program p' (which does not contain any exit statements) is shown to be correct and to have low-order polynomial complexity.

Java Memory Allocation with Lazy Worst Fit for Small Objects. HYUNG-KYU CHOI, YOO C. CHUNG AND SOO-MOOK MOON
Memory allocation in programming languages should be fast, space efficient and dynamic. Moreover, since fragmentation (which involves used and unused memory being interleaved) leads to waste of memory in managed heaps, finding free memory chunks to accommodate requested memory may be difficult even if the requested memory is smaller than the used memory. There are different approaches to memory allocation which permit different degrees of fragmentation and allocation speeds. This paper is concerned with a common approach which maintains a linked list of free memory chunks, called the free list, and searches the free list for a chunk that satisfies the memory allocation request.

A certain fitting policy is used when searching the free list for a suitable memory chunk. The fitting policy may be based on the so-called first fit where the list is searched sequentially and the first suitable free memory chunk is used, or the fitting policy may be based on the so-called best fit where the smallest suitable memory chunk is used. At the other end of the scale, the worst fit strategy uses the largest suitable memory chunk. The approach of this paper is to use lazy worst fit for memory allocation in Java which exploits pointer-incrementing memory allocation with free lists. This reduces the search time for the free lists and allows to discard inadequate free space. Lazy fits are evaluated on a working Java virtual machine using non-trivial Java programs.

Area Conscious State Assignment with Flip-Flop and Output Polarity Selection for Finite State Machine Synthesis—A Genetic Algorithm. SANTANU CHATTOPADHYAY
The controllers of digital circuits are often specified in the form of a finite state machine (FSM) consisting of a number of states and transitions between these states. Synthesis of an FSM plays an important role in digital circuit design and consists of three subproblems:

- state encoding and assignment where binary codes are assigned to individual states,
- flip-flop selection to store the state bits and
- polarity selection for state bits and primary output.

Genetic algorithms are stochastic optimization search algorithms based on the mechanics of natural selection and natural genetics which can find good state assignments and are effective in performing flip-flop selection and flip-flop polarity selection. An earlier work of the author presented a genetic algorithm approach integrating state assignment and flip-flop selection to store the state bits. This paper starts from that earlier work and presents an integrated approach based on the use of genetic algorithms to solve the problems of:

(i) assignment of codes to the states;
(ii) polarity assignment to the flip-flops;
(iii) polarity assignment for primary outputs.

Extensive experiments are reported which demonstrate various advantages. In particular, the paper shows that a proper choice of state codes and polarities can result in significant improvement in the area requirements of an FSM.

BLRL: Accurate and Efficient Warm-up for Sampled Processor Simulation. LIEVEN EECKHOUT, YUE LUO, KOEN DE BOSSCHERE AND LIZY K. JOHN
Microarchitectural research relies heavily nowadays on architectural simulation to model the cycle-level behaviour of modern microprocessors. However, since cycle-level simulations work at a fairly detailed level, they are time-consuming (they can take up days or even months). One proposed solution to reduce the total simulation time is the so-called sampled simulation where a selected number of execution intervals (samples) are simulated from a complete benchmark execution. However, sampled simulation suffers from two problems:

(i) how to select the samples so that all major phases are represented in the sampled execution and
(ii) the cold-start problem which refers to the unknown hardware state at the start of each sample.

Solutions to (i) and (ii) have been proposed in the literature. In particular, the cold-start problem has been avoided by warming up hardware structures. Different warm-up strategies have been used in the literature and are reviewed in the paper. The paper then proceeds to propose an accurate and efficient warm-up strategy called the boundary line reuse latency (BLRL). Like an earlier proposed memory reference reuse latency (MRRL) warm-up strategy, BLRL is based on reuse latencies. However, BLRL considers reuse latency for memory references originating from instructions in the sample only and which cross the presample/sample boundary line. These cross boundary reuse latencies point to the memory locations that need to be warmed up. Experimental results conducted by the authors show that BLRL outperforms MRRL.

Pyramidic Clustering of Large Scale Microarray Images. PAUL O'NEILL, KARL FRASER, ZIDONG WANG, PAUL KELLAM, JOOST N. KOK AND XIAOHUI LIU
Clustering is a learning technique used for classification and segmentation problems in exploratory data analysis. Current clustering techniques are limited in the scale of data they can process. In order to deal with this scaling issue (and hence with large datasets), current clustering approaches require that the data be sampled, summarized or constrained. This paper aims to make traditional clustering techniques applicable to large datasets while still producing results comparable to their unscalable relatives when tested on small data. The paper focuses on the application of clustering techniques to microarrays and instead of discarding the wealth of spatial image information, it harnesses it producing an amalgamated result from a consensus of the previous layers. By scaling so the existing techniques to larger datasets, the paper provides extra information about the internal decision process of the clustering methods employed. This results in the capture of contextual information that traditional methods usually lose. The paper achieves this using a novel image analysis technique based on the pyramidic layering concept, the so-called Copasetic clustering, which enables the review of historical information and helps improve the final output. The proposed solution is illustrated using a detailed example and is further evaluated through a set of applications.

A Robust Multiparty Key Agreement Protocol Resistant to Malicious Participants. YUH-MIN TSENG
A multiparty key agreement protocol involves all participants cooperatively establishing a common key to be used

to encrypt/decrypt transmitted messages amongst the participants over an open channel. Multiparty key agreement protocols can be either authenticated or non-authenticated. A non-authenticated protocol cannot provide participant and message authentication and so must make use of some form of authentication (e.g. through an authenticated network channel or some alternative schemes). This paper concentrates on two well-known approaches which use authenticated network channels and shows that these approaches are unable to withstand malicious participant attacks. The paper then proposes a new multiparty key agreement protocol which is resistant to malicious participant attacks and shows that this new protocol:

- is round-efficient (i.e. requires only two simultaneously broadcast rounds when no malicious participant is detected, and otherwise, other participants will know these malicious participants and will proclaim them as cheaters);
- is secure (against malicious participant attacks under the random oracle model); and
- has message size which is independent of the number of participants.

Generating *T*-ary trees in linked representation.
JAMES F. KORSH
Binary and t-ary trees have been generated in the literature using various methods which include sequences of integers and linked representation. However, in only one linked representation method, the next tree is generated from its predecessor. This method is owing to Knuth who adopted Sharbek's algorithm for generating all ordered trees on n nodes and showed that this method required only $8.75-9.37/[n + O(n^{-2})]$ memory references per generated tree. This was carried out for binary trees and Knuth posted the problem of generating it to 3-ary trees. This paper presents a new algorithm for the generation of all n node t-ary trees in linked representation. This new algorithm (which is implemented in C++) is based on the colex listing of the sequences representing all n node t-ary trees (i.e. when each of the sequences is reversed, the listing is in lexicographic order).