doi:10.1093/comjnl/bxl077

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. The Capsule Reviews were compiled by Fairouz Kamareddine. Professor Kamareddine is an Associate Editor of *The Computer Journal* and is based in the Department of Mathematical and Computer Sciences at Heriot-Watt University, Edinburgh, UK.

An Inline Detection and Prevention Framework for Distributed Denial of Service Attacks. Z. CHEN, Z. CHEN AND A. DELIS

Contemporary Distributed Denial of Service (DDoS) attacks are very sophisticated which make the tracing and detecting of their activities extremely challenging. It is a daunting fact that many DDoS attack stages are even automated making these attacks effortless for the malicious actors and painful for the affected parties. A number of efforts have taken place to curb the effects of DDoS and *The Computer Journal* has been an important medium for the publication of research on these and related security issues (e.g. on the Denial of Service attacks in [1] and authenticated key agreement protocols in [2]). The paper under discussion proposes a so-called DDoS container which is a network-based detection/preservation framework which inspects every passing packet and blocks any DDoS traffic in real-time. The framework monitors the progress of all connections and performs data correlation and inspection. If a DDoS session is identified, the framework can take a number of options, such as terminating the session or blocking the connection or causing packet dropping, etc. The framework is described in detail and the paper discusses the analysers for two very common contemporary DDoS networks (Stacheldracht and the Tribe Flood Network 2000 TFN2K) as well as for the DNS amplification attacks. The paper also reports on the implementation and experimental evaluation of the proposed DDoS container. The experimental evaluation demonstrates that the framework is both efficient and effective in a large number of settings.

REFERENCES

[1] Sharafat, A. and Fallah, M. (2004) A framework for the analysis of denial of service attacks. *The Computer Journal*, **47**, 179–192.
[2] Tseng, Y. (2006) A secure authenticated group key agreement protocol for resource-limited mobile devices. *The Computer Journal*, **50**, 41–52.

A Secure Authenticated Group Key Agreement Protocol for Resource-Limited Mobile Devices. Y.-M. TSENG

Since technologies applicable to the security of wired networks are not suitable for wireless networks, and since wireless networks are more vulnerable to security threats, it is a challenging task to develop secure technologies for mobile devices. One particular important secure technology is the design of a secure group key establishment protocol. Such a group key establishment protocol would allow users to construct a group key to encrypt/decrypt messages over an open channel. The key establishment protocol can either be 'a group key distribution' where an elected entity generates the group key or a 'group key agreement' where all users collectively generate the key. This paper, which builds on an earlier one by the author [1], is concerned with the development of a secure group key agreement protocol for asymmetric mobile wireless devices. This protocol requires only two rounds to construct a group key and is suited to mobile devices with low-power computing capabilities. After presenting the model and notations used, the author gives the new authenticated group key agreement protocol and shows a number of its properties. In particular, the author shows that the protocol is well-suited for low-power mobile devices, is a real contributory group key agreement protocol, is secure against passive adversaries under the decision Diffie-Hellman assumption (i.e. an attacker is unable to obtain the established group key by eavesdropping messages) and is secure against impersonators attacks under the decision Diffie-Hellman and the discrete logarithm assumptions (i.e. the malicious attacker cannot generate valid formulae that pass verification). The author also shows that the proposed protocol provides implicit key authentication and forward secrecy (i.e. compromise of a long-term key cannot result in the compromise of previously established keys). The computational complexity and the communication cost of the proposed protocol are analysed. It is shown that the proposed protocol is efficient in terms of the computational cost of each low-power node and the number of rounds. It is also shown that the communication cost is larger than the cost required by Bresson *et al.*'s protocol; however, this is compensated by the

fact that this proposed protocol provides more security properties (e.g. Bresson *et al.*'s protocol does not enjoy forward secrecy, or implicit key authentication or is a contributory group key agreement).

REFERENCE

[1] Tseng, Y. M. (2005) A robust multi-party key agreement protocol resistant to malicious participants. *The Computer Journal*, **48**, 480–487.

The Alpha of Indulgent Consensus. R. GUERRAOUI AND M. RAYNAL

Reaching consensus among processes on a common decision (which may be based on different proposals) is key to distributed computing. The consensus problem is that each process proposes a value and each correct process (i.e. a process which does not crash) has to decide on the same value v, where v is one of the proposed values. It is well known that in an asynchronous system, an algorithm which ensures validity and agreement may not always terminate and, hence, solving consensus in an asynchronous system is impossible even if only one system may crash. In practice, distributed systems are synchronous and have timing bounds on process speeds and communication delays. These bounds are sometimes violated during the so-called instability periods. Indulgent algorithms preserve the safety of consensus during instability periods and are partially synchronous. Failure detector devices such as Omega (not to be confused with the OMEGA system of *The Computer Journal* paper of [1]) enable one to design different forms of indulgent consensus, although they may not themselves implement the consensus. This paper attempts to build a complement to systems such as Omega where the consensus is implemented in an indulgent way. The result is a safety-oriented high-level abstraction, Alpha, where indulgent consensus algorithms decouple safety and liveliness properties. Alpha is a simple abstraction which factors out the way processes communicate and cooperate. After setting out the process models, the paper presents the generic framework where the Alpha abstraction is presented and followed by a generic consensus algorithm based on Alpha and Omega, which is shown through a sketch to be correct. Then, four sections are devoted to instantiate and implement the generic framework on four different communication systems. First, Alpha is implemented in a shared memory system where the algorithm and its correctness proof are given. Then, the same process is followed for a shared disk model, for a message passing model and for active disks. Then the paper discusses the possibility where infinitely many processes are present.

REFERENCE

[1] Gitzel, R., Ott, I. and Schader, M. (2006) Ontological extension to the MOF metamodel as a basis for code generation. *The Computer Journal*, **50**, 93–115.

An Adaptive Materialization Method for Globally Optimized Query Evaluation in a Hierarchical Mediator System. K. H. JOO AND W. S. LEE

A mediator is a middleware that allows a local user to assess a single server by hiding the heterogeneous environment from the user as much as possible. Usually, a so-called wrapper is used to hide some information so that a server can be presented in a new form without changing the internal structure of the server. In a distributed database, a number of local schemata are integrated in a global schema. Since the local schemata can change, it is difficult to maintain a unique global schema. A mediator handles the requirements of a global user so that only relevant local information is integrated. To evaluate a global query on a number of distributed information sources, the query is transformed into a number of sub-queries each corresponding to one of the information sources. The result of a sub-query in a mediator is obtained by either 'modification' or 'materialization'. In a hierarchical mediator system, the performance of query evaluation can be improved if the materialization method is used. The reason for this improvement is due to the fact that materialized results of sub-queries in a mediator can also be used by its parent mediators. The paper proposes a method for choosing the optimized set of materialized queries in each mediator of a hierarchical mediator system such that available storage in each mediator can be highly utilized at any time. To do so, the paper proposes an algorithm to generate the optimized 'implementation plan' (IP) used to describe how the sub-queries of a mediator are evaluated and describes how to maintain the IP to be optimized. The paper gives the method for query transformation in a hierarchical mediator system and defines the query evaluation cost of a mediator. Several experiments are reported.

A Fast Radix-4 Floating-Point Divider with Quotient Digit Selection by Comparison Multiples. H. NIKMEHR, B. PHILIPS AND C.-C. LIM

This article starts by stating that most recent microprocessors and digital signal processors perform all four arithmetic operations (addition, subtraction, multiplication and division) but that division is not performed as fast as the other three operations. The authors propose an algorithm for a more robust radix-4 Floating-Point (FP) division which increases the efficiency of division by increasing the parallelism among the functional units. Most VLSI implementations of FP division are based on the so-called SRT algorithm whose speed depends on the complexity of the Quotient Division Selection (QDS) function which is usually implemented using a lookup table. The paper proposes an implementation of the QDS function using the comparison multiples method where the quotient digit is calculated using a sign and magnitude format. The paper presents this new comparison multiples method which is judged to halve the number of comparators and comparison sign detectors, to lead to less delay to FP division and

to enable the Partial Reminder (PR) to operate in parallel to the rest of the QDS function. All this makes the QDS function faster. Calculations by logical synthesis show that the comparison multiples based radix 4-FP divider is up to 22% faster than the conventional implementation.

Ontological Extension to the MOF Metamodel as a Basis for Code Generation. R. GITZEL, I. OTT AND M. SCHADER
There exist two applications of metamodeling: the ontological and the linguistic. The ontological metamodeling uses metamodels to describe domain-specific hierarchies whereas the linguistic metamodeling uses a metamodel to describe the language syntax without a concrete real-world mapping. Ontological metamodeling needs metamodel hierarchies with a flexible number of layers whereas many linguistic applications do not require more than four model layers. Hence, domain-specific languages enjoy a strong 'ontological' aspect whereas purely ontological metamodeling lack any such support. This paper proposes the so-called Ontological Metamodel Extension for Generative Architectures (OMEGA) where the standard Meta Object Facility (MOF) which exhibits a four layer maximum is extended to allow ontological metamodel hierarchies and non-linear metamodeling. (Note that OMEGA must not be confused with the system Omega used in *The Computer Journal* paper in [1].) The paper starts with a discussion of linear versus non-linear metamodel hierarchies. A number of options that need to be addressed when designing a non-linear hierarchy are discussed, including the number of layers in a metamodel hierarchy, deep versus shallow instantiation and the instantiation semantics, and the choice of the model elements (strictness ordering, the number and nature of distinct elements, whether methods or operations need to be included in the hierarchy, the static and dynamic relationships that need to be included and whether the top layer is recursively defined or modeled as an axiomatic layer). With all the design issues and options considered, the paper sets out to describe the OMEGA framework and its contribution to code generation. A code generation example scenario is discussed as well as the prototype implementation which is available for download. Ample discussion of the limitations, open questions and related work is given.

This paper belongs to the active area of research concerned with improving a number of features in metamodeling. See, for instance, the recent paper [2] in *The Computer Journal*.

REFERENCES

[1] Guerraoui, R. and Raynal, M. (2006) The Alpha of indulgent consensus. *The Computer Journal*, **50**, 53–67.
[2] Almendros-Jimenez, J. and Iribarne, L. (2006) Describing use case relationships with sequence diagrams. *The Computer Journal*, **50**, 116–128.

Describing Use-Case Relationships with Sequence Diagrams. J. M. ALMENDROS-JIMÉNEZ AND L. IRIBARNE
Use cases specify required usages of a system. Each use case specifies some behavior which can be performed in collaboration with some actors (users/systems which interact with the system in question). There are many objections against UML's semantics of use cases and their relationships. For this reason, this paper describes use cases by means of sequence diagrams where sequence diagram relationships are used for identifying and defining use-case relationships. In particular, for each use case, a sequence diagram is built and then, from the existing sequence diagrams, use-case relationships are identified. Furthermore, the sequence diagrams can be refined leading to new use cases and hence new use case relationships can be discovered from the refined sequence diagrams. This process can be applied iteratively allowing the developer to discover further behavioural details or to better describe already existing ones.

This paper belongs to the active area of research concerned with improving a number of features in metamodeling. See, for instance, the recent paper [1] in *The Computer Journal*.

REFERENCE

[1] Gitzel, R., Ott, I., and Schader, M. (2006) Ontological extension to the MOF metamodel as a basis for code generation. *The Computer Journal*, **50**, 93–115.